
SZERKESZTETTE: MARTIN F. WOLTERS

D. BUSSE
J. TWIEHAUS

**KULCS
A SZÁMÍTÓ-
GÉPHEZ**

HARDVER

SZOFTVER

SZERKESZTETTE: MARTIN F. WOLTERS

KULCS A SZÁMÍTÓGÉPHEZ

HARDVER
SZOFTVER

2. kiadás

D. BUSSE

KULCS A SZÁMÍTÓGÉPHEZ

HARDVER

2. kiadás

SZERKESZTETTE:
MARTIN F. WOLTERS

MŰSZAKI KÖNYVKIADÓ, BUDAPEST, 1988

Az eredeti mű:

D. Busse: Der Schlüssel zur Computer — Hardware

Herausgegeben: Martin F. Wolters

Copyright by Econ Verlag, Düsseldorf, 1981

Lektorálta:

Dr. Sima Dezső okl. villamosmérnök

Hungarian translation

© Dr. Büky Péterné, Pótzty Péter, 1984

ETO: 681.3

ISBN: 963 10 5694 5

963 10 5849 2

963 10 7595 8

Előszó

Kiadónk gondozásában tíz éve jelent meg a „Kulcs a számítógéphez” első változata. A könyv világszerte sikert aratott: számos nyelvre lefordították*, összesen több mint 250 000 példányban jutott el az olvasókhoz.

Mostani négykötetes vállalkozásunkkal ezt a hagyományt szeretnénk folytatni.

A számítástechnika rohamos fejlődése: az újszerű műszaki megoldások a hardver vonatkozásában; teljesen új utak a programozásban, amelyek a szoftvertechnológia kialakulásához vezettek; és a szervezés terén bekövetkezett változások, amelyek a számítástechnika alkalmazását kísérik, szükségessé tették, hogy a számítástechnika alapismereteinek megismertetéséhez teljesen új oktatási anyagot dolgozzunk ki.

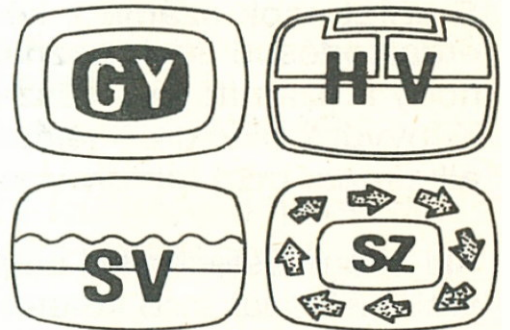
Ezért az ismereteket a következő négy kötetre osztottuk fel:

Kulcs a számítógéphez — Gyakorlat.

Kulcs a számítógéphez — Hardver.

Kulcs a számítógéphez — Szoftver.

Kulcs a számítógéphez — Szervezés (Orgver).



E felosztás és a kötetek anyagának összeállítása számítástechnikai és oktatási szakemberek hosszú közös munkájának eredménye.

A szerzők az ismeretek tárgyalása során most is arra törekedtek, hogy az érdeklődő laikus a szöveget azonnal megérthesse, a legfontosabb tudnivalókat könnyen elsajátíthassa.

* Többek között 1972-ben és 1974-ben két kiadásban is magyar nyelven. (Szerk.)

A „Kulcs a számítógéphez — Hardver” c. kötet a számítógép műszaki oldalát mutatja be.

Ebben a könyvben bemutatjuk

- a számítógép-technika és a kommunikációtechnika egyre határozottabb összefonódását és ennek következményeit;
- a számítógép-rendszerek fajtáit, és ezek teljesítőképességét;
- a számítógépek részegységeit és perifériáit, valamint azok működését;
- a konstrukció és a modern félvezető-technika ismereteire támaszkodva az utasítások megértésének és végrehajtásának logikai és műszaki vonatkozásait.

E kötet végén példán keresztül mutatjuk be, hogy hogyan épülnek fel a számítógépek utasításai, hogyan lehet azokat dekódolni, valamint hogy a számítógép egyes részegységeire milyen feladatok hárulnak, és ezek hogyan működnek együtt.

E kötet azok számára készült, akik hivatásuknál fogva vagy egyéni érdeklődésből foglalkoznak számítógépekkel, és tudni akarják, hogy a számítógép műszakilag milyen technikán alapul.

Könyvünk didaktikai felépítése olyan, hogy magántanuláshoz ugyanúgy alkalmas, mint tanfolyamok kísérő segédleteként.

Aki számítástechnikai alapismeretekkel rendelkezik, ami a sorozat első kötetéből „Kulcs a számítógéphez — Gyakorlat” könnyen megszerezhető, és kellő érdeklődést tanúsít, az a kötet gondosan feldolgozott tudásanyagát gyorsan és biztosan magáévá teheti.

München, 1981. szeptember

Martin F. Walters

A kötet célja

A könyv gondos áttanulmányozásával ismereteket szerez arról, hogy

- hogyan függnek össze egymással a kommunikációtechnika és a számítástechnika, és hogyan lehet a hagyományos készülékekből, mint pl. a telefonból, írógépből vagy tv-készülékből terminálokat kialakítani, és ezeket egymással vagy pedig az adatfeldolgozó berendezésekkel összekötni;
- milyen számítógéptípusok léteznek, ezek kiépítettségüktől függően milyen teljesítményre képesek, és milyen lehetőségek vannak a számítógépek vagy részegységeik összekapcsolására;
- milyen egységekre és perifériákra van szüksége a számítógépnek ahhoz, hogy funkcióit betölthesse, és milyen technikai sajátosságai vannak az egyes készülékeknek;
- milyen logikai összefüggések szerint működik a számítógép hardverje, és ez a logika miként valósítható meg a félvezető-technikában;
- hogyan hajt végre egy utasítássorozatot a központi egység.

Tanulási módszer

E sorozat könyvei a szerzőmunkacsoport által kifejlesztett, jól bevált oktatási módszeren alapulnak.

E módszerrel készült művek különösen alkalmasak az önálló tanulásra, mert:

- a magyarázathoz hétköznapi nyelvet használ, így a szöveg jól érthető, és az ismeretek logikusan, gondolati ugrások nélkül épülnek egymásra;
- a fontosabb meghatározásokra, ismeretekre nemcsak a szövegbeli nyomdatechnikai kiemelések, hanem az ábrák is felhívják a figyelmet a jobb rögzítés érdekében;
- minden fejezet végén feladatok segítik:
 - a lényeges ismeretek fontosságának felismerését,
 - az olyan összefüggések megértését, amelyek a szövegben nem mindig dolgozhatók ki,
 - az újonnan tanultak összekapcsolását a már meglévő ismeretekkel.

Mielőtt könyvünk áttanulmányozását megkezdené, a következőket kell tudnia:

A feladatok nem az Ön vizsgáztatására készültek.

Ezért nem is fontos, hogy a feladatokat „kapásból” meg tudja oldani.

A kérdések a lapok egyik oldalán, míg a válaszok a túloldalon találhatóak.

A kérdéseken való elgondolkozás és a válaszok helyességének ellenőrzése didaktikailag igen fontos.

Ha valamelyik választ nem helyesen adta meg,

vagy nem érti, hogy miért a megadott válasz a helyes,



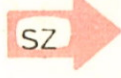
akkor olvassa el újra a kérdésekhez tartozó, megfelelő oldalakat:

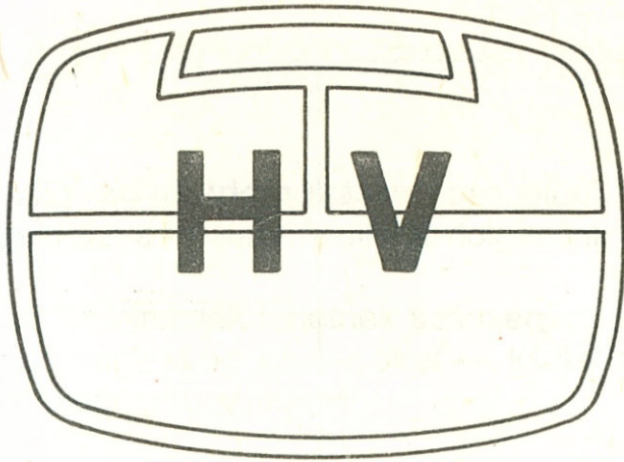
a kérdéseket tartalmazó lap felső részén mindig megadjuk,

hogy azok mely oldalakon található szövegrészekre vonatkoznak.

A szövegben többször utalásokat fog találni sorozatunk többi kötetére, amelyekben a szóban forgó témát részletesebben tárgyaltuk.

Azért, hogy ez szembetűnő legyen, a kötetekre való utalást a következőképpen jelöltük:

- | | | |
|-----------------------|---|---------------------|
| Kulcs a számítógéphez |  | Gyakorlat. |
| Kulcs a számítógéphez |  | Szoftver. |
| Kulcs a számítógéphez |  | Szervezés (Orgver). |



Adatfeldolgozás
és kommunikáció

Számítógépek
és alkalmazásuk

A számítógépek
funkcionális
egységei

Hogyan
működik
a számítógép?

Ebben a részben megismeri,
hogyan

- a kommunikáció és az adatfeldolgozás hogyan függnek össze egymással,
- a kommunikáció hagyományos eszközeinek teljesítményét miként növelték a számítógép-technika alkalmazásával, és
- milyen adatátviteli vonalakon és hálózatokon tudnak az előfizetők egymással kommunikálni.

Számítógép
és kommunikáció-
technika

Személyi kommunikáció

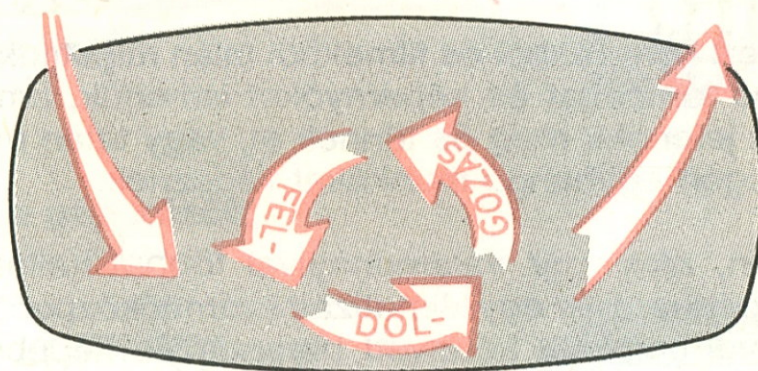
Adatkommunikáció

Átviteli vonalak

Számítógép és kommunikációtechnika

Mindig **adattfeldolgozásról** beszélünk, ha adatokat vagy — általánosan fogalmazva — információkat dolgozunk fel.

Ilyen információk lehetnek számjegyek vagy betűk — függetlenül attól, hogy leírjuk vagy kimondjuk-e őket —, különleges jelek: pl. &, %, § stb., vagy ezek tetszőleges kombinációi.



A feldolgozás sokféle lehet: pl. másolás, rendezés, számítás, összehasonlítás, azonosítás, tárolás vagy hozzárendelés.

Ha a feldolgozás elektronikusan, tehát számítógép segítségével történik, akkor azt **elektronikus adattfeldolgozásnak** — röviden **EAF**-nek — nevezzük.

Nem is nagyon régen, ha elektronikus adattfeldolgozásról hallottunk, képzeletünkben még nagy számítógépek jelentek meg. Az elektronikus építőelemek egyre növekvő mértékű miniatürizálása lehetővé tette, hogy teljes értékű számítógépet* egyre kisebb térben, pl. egyetlen áramköri elemként realizáljanak. E miniatűr számítógépek természetesen tetszőleges készülékekbe is beépíthetők.

Az elektronikus adattfeldolgozás tehát ma már nem csak nagy számítógépre korlátozódik. A számítógépek egyre jobban elterjednek az élet minden területén, pl. többek között akkor is, ha két előfizető telefonon beszélget egymással, vagy egyikük táviratot küld a másiknak. A számítógépet tehát nemcsak adattfeldolgozásra, hanem kommunikációra is használják, és már nincs messze az az idő, amikor a számítógép minden távbeszélő- és táviróforgalomban is jelen lesz.

* Pontosabban fogalmazva azok központi egységét. (Lektor.)

Számítógéphardver

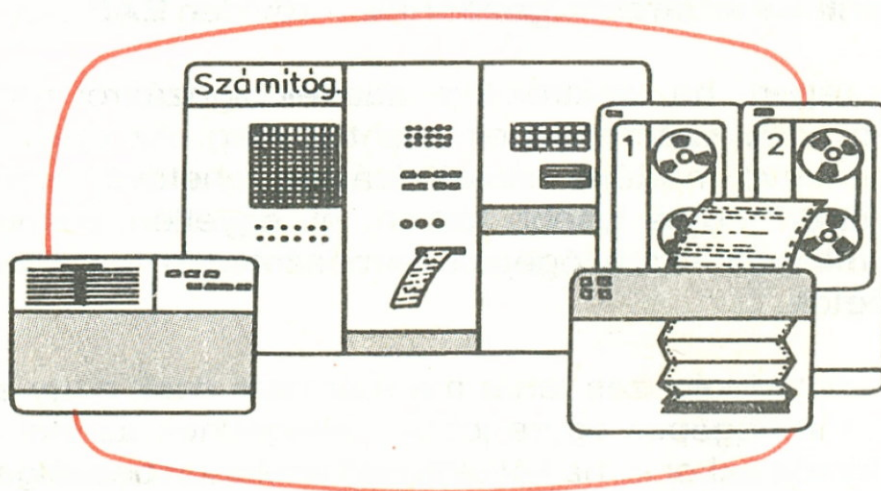
A **számítógéphardver** olyan fogalom, amely bizonyára nem ismeretlen azok számára, akik könyvünket kezükbe veszik. A „hardware” angol eredetű szó, és tulajdonképpen szó szerint „fémáru, kemény áru” jelent. A hardver mindazt magában foglalja, ami számítógépben a „kemény”-technikának nevezhető. Hardverhez soroljuk pl. a számítógépet alkotó berendezéseket, de ugyanígy ide tartoznak a kábelek vagy a csatlakozók is. A szoftver kifejezés az angol software: „lágú áru” kifejezésből ered és az adatok feldolgozásához nélkülözhetetlen programok együttesét jelenti.



A számítógéphardvert könyvekből és filmekből talán mint titokzatosan világító készülékeket, villogó lámpákat és képernyőket ismerjük, amelyen misztikus színes minták, jelek jelennek meg és tűnnek el, vagy talán mint „robotokat”, amelyek a futószalagon egyhangú munkákat végeznek.

Mások képzeletében, akik már valamennyivel többet hallottak a számítógépekről, a számítógéphardver egy klimatizált számítóközpontban elhelyezett, nagy tárákkal és nyomtatókkal körülvett berendezés képében jelenik meg. Aki már tud valamit a távfeldolgozásról, az talán terminálokra is gondol, amelyek a központi számítógéptől távol helyezkednek el.

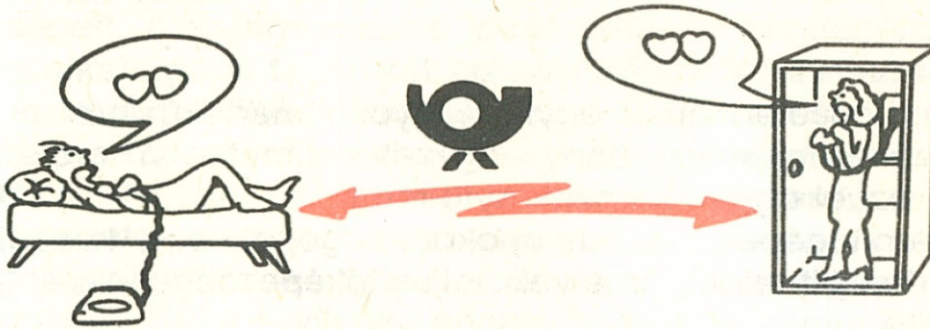
Mindezek az elképzelések helyesek. Ez mind számítógéphardver.



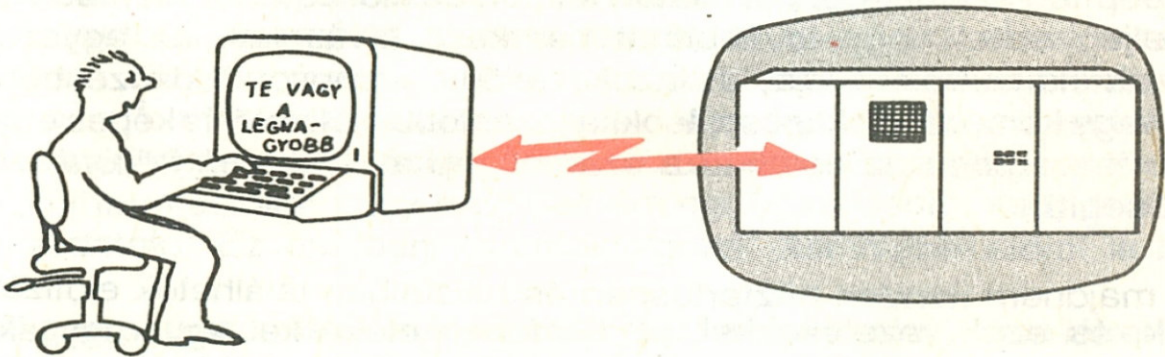
De egy egészen hétköznapi készülék is lehet számítógéphardver, mint pl. a telefon, a tv-készülék, az írógép vagy a táviró.

Mindezeket a készülékeket a hírközléshez, a **kommunikációhoz** használják:

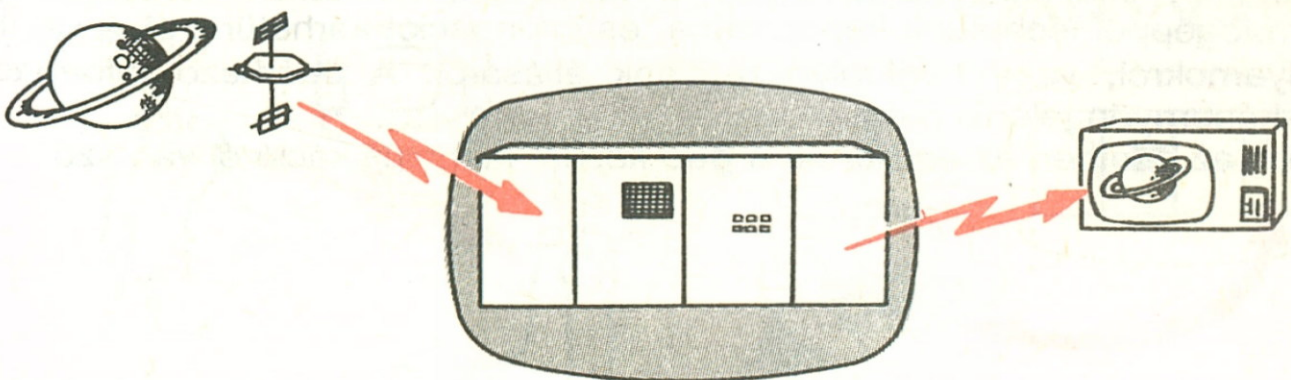
■ az emberek között,

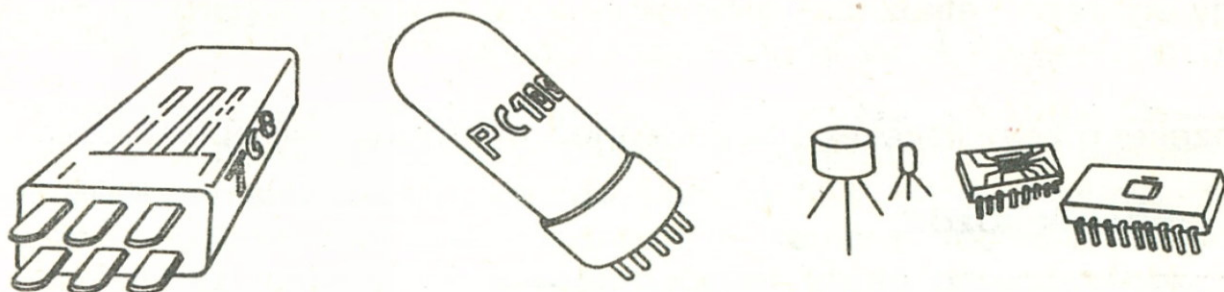


■ az ember és a gép között,



■ a gépek között.





A számítógép áramköri elemeinek egyre nagyobb mértékű *miniatürizálása* lehetővé tette a számítógépek néhány négyzetcentiméternyi területen való megvalósítását, és ezzel egy új korszak nyílt meg.

Ilyen kis méretű számítógépek már telefonokba, írógépekbe, tv-készülékekbe vagy távirókba is beépíthetők, amelyek teljesítőképesége ezzel jelentős mértékben fokozható.

Példák:

Egy beépített számítógéppel ellátott telefon az előfizető távollétében is fogadhat, feljegyezhet vagy továbbíthat beérkező hívásokat, az egyszer bevitt hívószámokat tárolni tudja, és azokat előprogramozott időközökben felhívhatja, vagy konferenciakapcsolásokkal akár több előfizetőt is képes egymással összekötni. Ebben az esetben a számítógép az *emberek közötti* kommunikációt segíti.

Mivel majdnem minden háztartásban és hivatalban található előfizetői végpontok, és ezek vezetékekkel, pl. telefonvezetékekkel egy vagy akár több számítógéphez is csatlakoztathatók, lehetővé válik a számítógép teljesítményének minden háztartásban vagy minden hivatalban való hasznosítása.

Ha egy telefonhoz megfelelő kiegészítő berendezésen keresztül egy tv-készüléket csatlakoztatunk, akkor a távbeszélő-hálózaton keresztül egy számítógéppel léphetünk kapcsolatba, és információt kérhetünk pl. a valutaárfolyamokról, vagy bankfolyószámlánk állásáról. A beérkező információ a tv-képernyőn jelenik meg.

Ebben az esetben az *ember és a gép közötti* kommunikációról van szó.

Kommunikáció és kommunikációtechnika

A **kommunikáció** tehát az információknak beszéd, kép, szöveg vagy adatok formájában való kicserélése személyek, személyek és gépek, vagy csak gépek között.

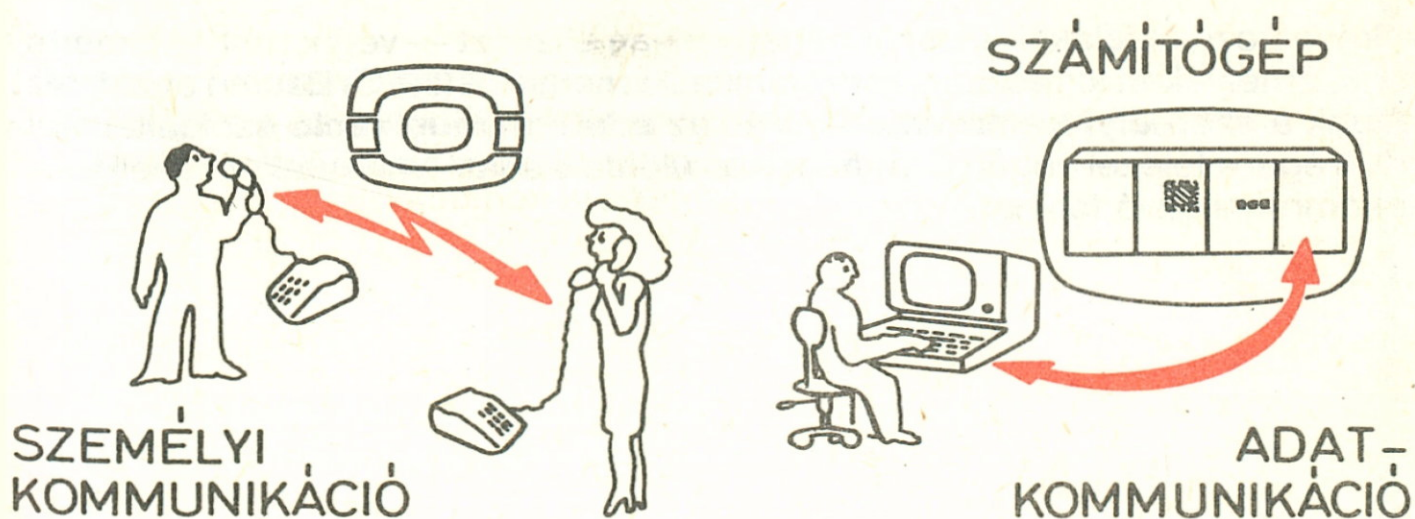
A mikroelektronika fejlődése nagymértékben kiterjesztette a kommunikáció lehetőségeit. A kommunikáció létrehozására, segítésére, ill. kényelmesebbé tételére alkalmazott technikát **kommunikációtechnikának** nevezzük.

Mivel a **kommunikációtechnikában** nagyjából ugyanazt a technikát, azaz ugyanazokat az elektronikus építőelemeket alkalmazzuk, mint a **számítástechnikában**, és mert a kommunikációtechnika berendezései ma már olyan feladatokat teljesítenek, amelyekre régebben csak az adatfeldolgozó berendezések voltak képesek, azért e két terület élesen már nem választható el egymástól. Ez is egyik oka annak, hogy a „kommunikációtechnika” témáját a számítógéphardvert tárgyaló kötetünkben tárgyaljuk.

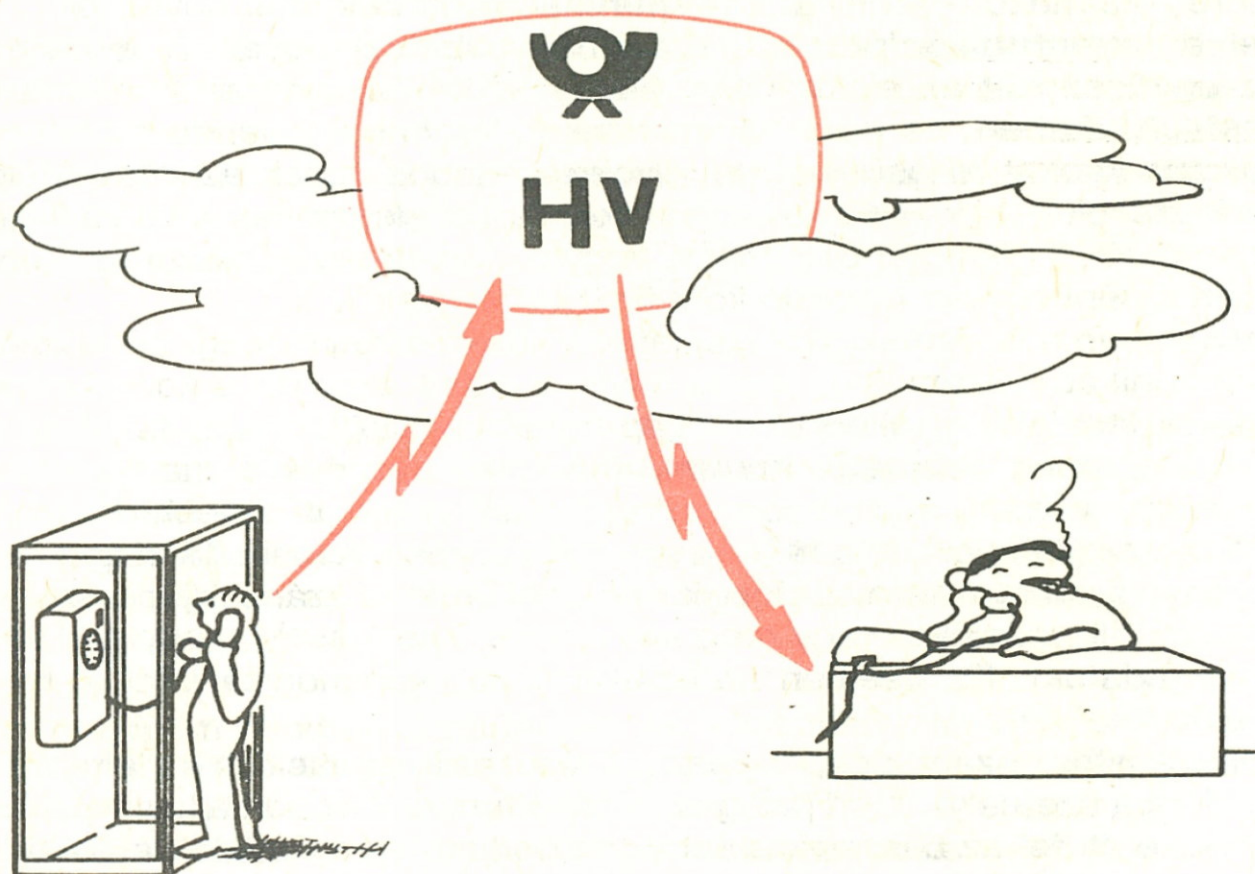
Könyvünkben a kommunikáció két fajtáját különböztetjük meg: egyik esetben a kommunikáció személyek között folyik, a másik esetben a kommunikációs partnerek legalább egyike számítógép (tehát a „gép”).

Az első esetben **személyi kommunikációról** beszélünk, mert a technika az emberek közötti információcserét „csupán” elősegíti, a technikának *nem feladata* az információ megértése vagy feldolgozása. A második esetet hagyományosan **adatkommunikációnak** nevezzük, mert a számítógépnek a hozzá beérkező információkat vagy *meg kell értenie* (utasítások), vagy *fel kell dolgoznia* (adatok). Ez esetben „adatkommunikáció” kommunikációról is beszélhetnénk.

Mindkét esetben számítógépek vesznek részt az információcsere létrehozásában. Mind a személyi, mind pedig az adatkommunikációban a kommunikációtechnika építi fel az összeköttetést és valósítja meg az adatok átvitelét.



A kommunikációtechnika a kommunikáló partnereknek az információátvitelhez kapcsolatfelépítést és átvitelt biztosít. Ez a technika azonban a felhasználó előtt rejtve marad, aki rendszerint csak a *végpontokat* látja, amelyek az információk bevitelére és kivitelére használatosak. Hiszen pl. egy telefonbeszélgetés során a beszédet „be kell vinni”, azaz bele kell beszélni a kagylóba. Ugyanígy az átviteli lánc végén a kagyló hallgatóján át „kivitelre”, hangra van szükség az „egymással beszélgetés” megvalósításához.



A következő oldalakon először néhány olyan eszközt — végpontot — ismertetünk, amelyeken keresztül a kommunikáció megvalósítható. Ezután áttekintést adunk a **személyi kommunikáció** és az **adatkommunikáció** szolgáltatásairól. Végül a fejezet végén bemutatjuk a különféle adatátviteli utakat, amelyeken a kommunikáció folyhat.



1. A következő berendezések közül melyek tartoznak a számítógéphardverhez?

- | | | | | | |
|-----------|-----------------------|---------|-----------------------|-----------|-----------------------|
| Tár. | <input type="radio"/> | Másoló. | <input type="radio"/> | Szoftver. | <input type="radio"/> |
| Nyomtató. | <input type="radio"/> | Távíró. | <input type="radio"/> | Képernyő. | <input type="radio"/> |
| Telefon. | <input type="radio"/> | Kábel. | <input type="radio"/> | Írógép. | <input type="radio"/> |

2. A kommunikációra mely megállapítások igazak?

A kommunikáció az információk kicserélése személyek között, ill. gépek vagy személyek és gépek között.

A kommunikációt technikai segédeszközök segíthetik.

Nincs olyan kommunikáció, amelyben ne venne részt számítógép.

Az olyan kommunikációt, amelynél minden partner ember, személyi kommunikációnak nevezzük.

Az olyan kommunikációt, amelynél legalább egy partner számítógép, adatkommunikációnak nevezzük.

Nincs olyan kommunikáció, amelynél legalább egy partner ne ember lenne.

A kommunikáció végpontjai nagyon gyakran a számítógéphardver részét képezik.

3. Hogy nevezzük azt a technikát, amelyet a kommunikáció létrehozására, segítésére, ill. kényelmesebbé tételére alkalmazunk?

.....



Válaszok

1. A szoftver kivételével minden megnevezett berendezés lehet számítógéphardver:

- | | | | | | |
|-------------------------------------|-----------|-------------------------------------|---------|-------------------------------------|-----------|
| <input checked="" type="checkbox"/> | Tár. | <input checked="" type="checkbox"/> | Másoló. | <input type="checkbox"/> | |
| <input checked="" type="checkbox"/> | Nyomtató. | <input checked="" type="checkbox"/> | Távíró. | <input checked="" type="checkbox"/> | Képernyő. |
| <input checked="" type="checkbox"/> | Telefon. | <input type="checkbox"/> | Kábel. | <input type="checkbox"/> | Írógép. |

2. A kommunikációra a következő megállapítások igazak:

- A kommunikáció az információk kicserélése személyek között, ill. gépek vagy személyek és gépek között.
- A kommunikációt technikai segédeszközök segíthetik.
-
- Az olyan kommunikációt, amelynél minden partner ember, személyi kommunikációnak nevezzük.
- Az olyan kommunikációt, amelynél legalább egy partner számítógép, adatkommunikációnak nevezzük.
-
- A kommunikáció végpontjai nagyon gyakran a számítógéphardver részét képezik.

3. Azt a technikát, amelyet a kommunikáció létrehozására, segítésére, ill. kényelmesebbé tételére alkalmazunk, **kommunikációtechnikának** nevezzük.

Személyi kommunikáció

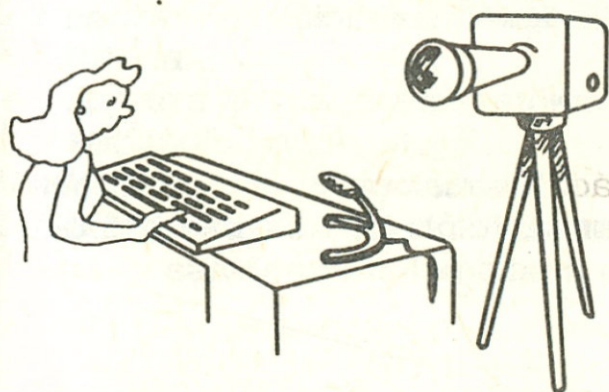
Adatátviteli végpontok

A **végpontok** feladata az információk felfogása és bevitele, valamint ezen információk rendelkezésre bocsátása vagy kivitele olyan formában, amely a kommunikációs partnerek számára érthető. A személyi kommunikáció esetén *emberek* a kommunikációs partnerek és gyakran az adatkommunikáció egyik partnere is az ember. Ezért a be-, ill. kiviteli eszközöket az *emberi érzékszervekhez illeszkedő* módon (tapintás, látás és hallás) kell kialakítani.

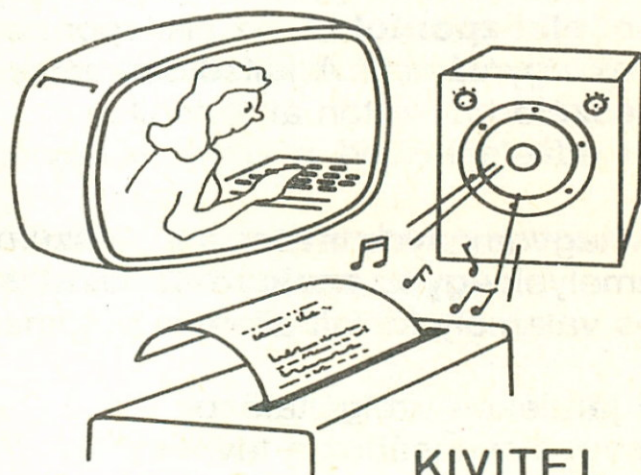
Ennek megfelelően a bevitelre és kivitelre használatos kommunikációs eszközöket rendszerint a következőkkel látják el:

A **bevitelhez** billentyűzet,
kamera és
mikrofon.

A **kivitelhez** nyomtató,
képernyő és
hangszóró.



BEVITEL



KIVITEL

Azt, hogy ezek az eszközök miként használhatók a kommunikációra, a következőkben ismertetjük.

Beszédkommunikáció

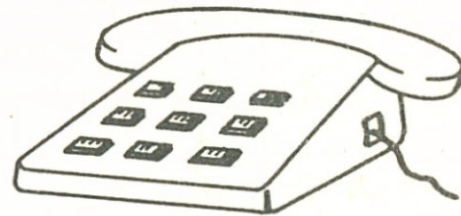
A kommunikáció legegyszerűbb és legtermészetesebb formája kétségtelenül az emberek közötti **közvetlen beszélgetés**, technikai segédeszközök nélkül. Ennek a közvetlen párbeszédnek az az előnye, hogy rövid idő alatt sok információ cserélhető ki. Ezek az információk nemcsak a beszédre szorítkoznak, hanem a beszélgető partnerek mimikáját, gesztikulálását és egyéniségének megnyilvánulásait is tartalmazzák.



Távbeszélés

Gyakran azonban a közvetlen beszélgetés térbeli vagy időbeli okokból nem lehetséges, ezért a beszédkommunikációt **távbeszélő útján**, tehát telefon igénybevételével végezzük.

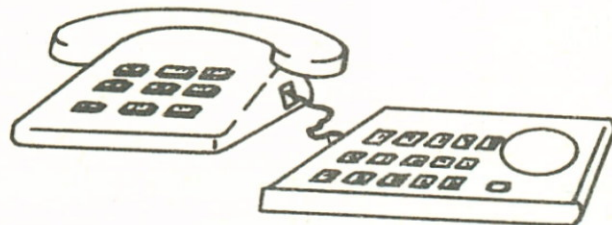
A magánéletben ehhez csak egy szokványos telefonra van szükségünk, amely **a nyilvános távbeszélő-hálózatba** van bekapcsolva, és amelyet egy számtárcsával vagy számbillentyűzettel működtetünk.



A hivatalokban gyakran alakítanak ki magánkommunikációs rendszereket, ún. **alközpontokat**. Az alközpont belső használói díjmentesen beszélgethetnek egymással. A külső személyekkel való kommunikáció a nyilvános távbeszélő-hálózaton át történik.

A hagyományos távbeszélő-készülékeken kívül vannak olyan berendezések is, amelyek **egyéb szolgáltatásokat is nyújtanak**, mint pl. hívószámok tárolása, és valamely foglalt állomás automatikusan megismételt újrarahívása.

E járulékos szolgáltatások megvalósításához a távbeszélő-berendezésekbe egy kisméretű számítógépet, helyesebben **számítógép-részegységeket** építenek be.



A következő példák betekintést nyújtanak azokba a lehetőségekbe, amelyek a járulékos szolgáltatások által a távbeszélő-forgalomban kínálóznak.

Automatikus hívásismétlés

Ha valamely hívott előfizető száma foglalt, akkor a készülék ezt az állomást bizonyos időközökben újra hívja, mindaddig, amíg a kapcsolat megvalósul.

Billentyűs hívás

A gyakran hívott számok tárolhatók, és adott nyomógombokhoz rendelhetők. Ez esetben elégséges a kiválasztott nyomógomb megnyomása, és a „telefon-számítógép” a teljes számot tárcsázza — esetleg az idegen ország vagy város hívószámával együtt.

Hangos telefon

Felhíváskísérleteknél nem kell a fülre szorított hallgatóval várni, amíg a hívott fél jelentkezik. Ehelyett a hívó fél a kagyló visszahelyezése után tovább dolgozhat. Még a partner jelentkezése után sem kell a hallgatót leemelni, mert a beszélgetés a mikrofonon és a hangszórón keresztül folytatható.

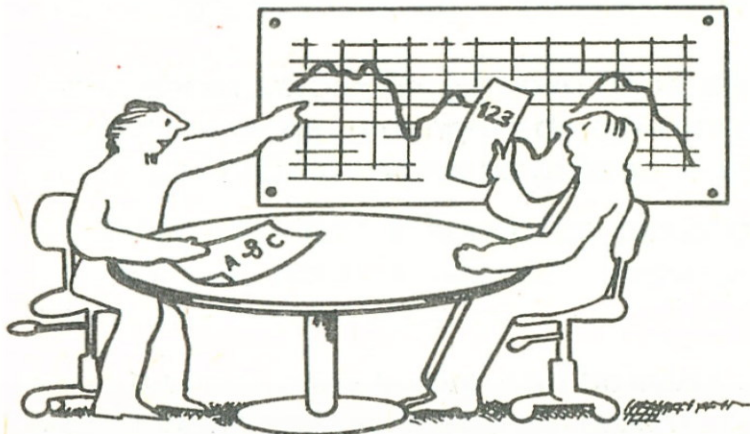
Egyéb szolgáltatások

További kényelmet nyújtanak azok a berendezések, amelyek

- minden érkező hívást automatikusan egy másik mellékállomáshoz továbbítanak, vagy
- az arra jogosult előfizetőknek megengedik, hogy a beszélgetésbe bekapcsolódjanak, vagy
- konferenciakapcsolásokat tesznek lehetővé, azaz olyan beszélgetéseket, amelyekben több előfizető egyidejűleg vesz részt.

Képkommunikáció

A közvetlen, szemtől szembe való beszélgetés során a kommunikációban nem csupán a beszédnek, hanem a beszélgető partnerek mimikájának és gesztikulálásának is igen fontos szerepe van. Ezenkívül megmutathatnak egymásnak pl. feljegyzéseket, leveleket, grafikákat, könyvoldalakat stb., amivel információt közölnek.



A telefonbeszélgetés során mindez nem lehetséges, legfeljebb ha mindkét partnernek pl. ugyanazok az írásos dokumentumok vannak éppen a kezében, akkor ezek konkrét részeivel kapcsolatosan kérdéseiket tisztázni tudják. Ha azonban nem ez a helyzet, akkor ezekről a dokumentumokról másolatot kell a partnerhez gyorsan eljuttatni, ez viszont a hagyományos közlési eljárásokkal túl lassú.

Távmásolás (TELEFAX)

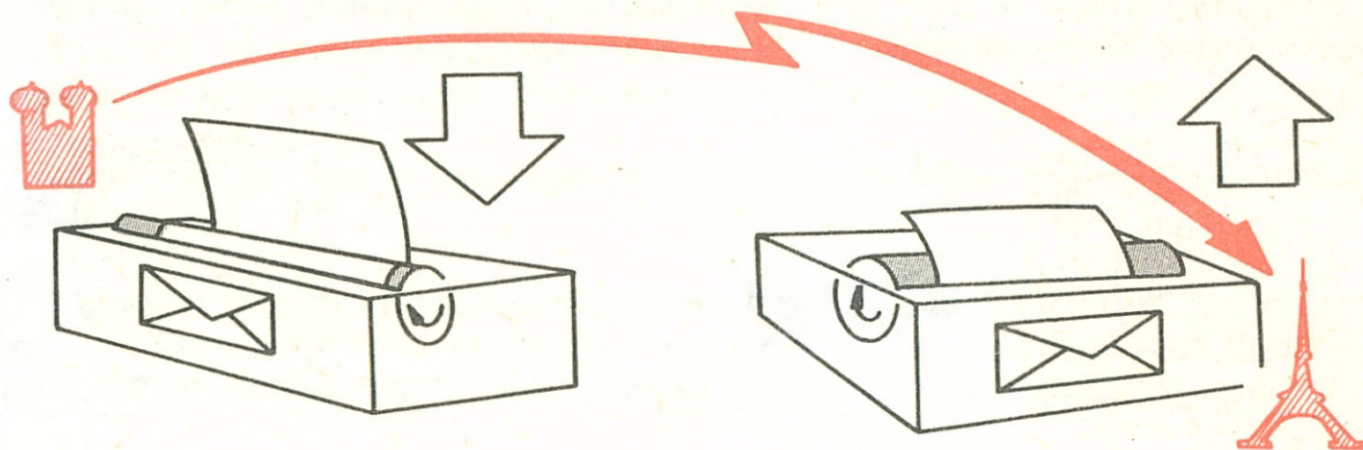
A 70-es évek óta beszerezhetőek olyan készülékek, amelyek egy DIN-A4-oldalt jó minőségben, rövid idő alatt a normál távbeszélő-hálózaton át tudnak közvetíteni. Ezek a készülékek a **távmásolók**.

A távmásolás, amelyet **TELEFAX**-nak is neveznek („fax”: faksimile), egy távközlési szolgáltatás, amely a tiszta beszédkommunikáció* kiegészítéseként tekinthető.

A távmásoló olyan készülék, amely mind adásra, mind vételre használható. A leadandó eredeti rajzot vagy írott anyagot pl. egy hengerre rögzítjük. A szerkezet ezt a henger forgása közben pontról pontra letapogatja, és az így kapott információt a vevőhöz továbbítja, amely a másolatot egy papíron — amely ugyancsak egy forgó hengeren van elhelyezve — pontról pontra rekonstruálja.

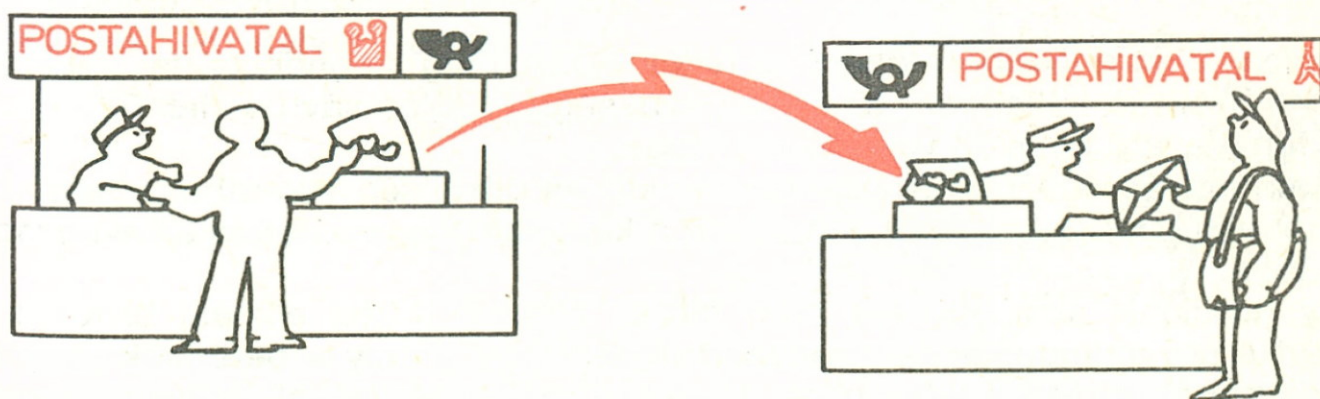
* A tisztán szöveges információk továbbítására az ún. TELETEX rendszer szolgál, l. a 32. oldalon. (Lektor.)

Egy tipikus alkalmazási eset táblázatoknak vagy grafikának a telefonbeszélgetés alatti azonnali közvetítésére:



A távmásolás azonban egy önálló új telekommunikációs forma is, mert egyre nagyobb mértékben fogja a hosszú telefonbeszélgetéseket, leveleket, és főként az expressz leveleket pótolni. A távmásoló felszerelése rendkívül egyszerű, mert a készüléket csak a távbeszélő-hálózathoz kell csatlakoztatni. A távbeszélő-hálózat technikai továbbfejlesztésével a távmásolatok átvitelének időtartama is egyre rövidebbé válik.

Aki nem akar a saját hivatalában távmásolót felállítani, az igénybe veheti a posta **telelevél-szolgálatát** a postahivatalban létesített távmásolón keresztül. Ott bármely írásunk távmásolatát elkészíttethetjük, amely azonnal továbbításra kerül a címzett postahivatalba, ahonnan a borítékolt másolatot a posta küldönccel kézbesíti a címzettnek.

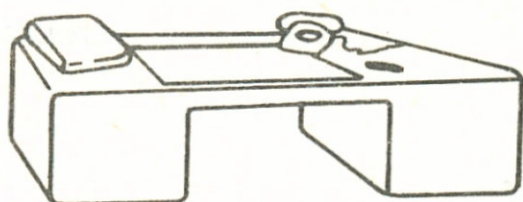


Képtelefon

A **képtelefon** azt jelenti, hogy használója hall és lát, és őt is hallják és látják. A készülék, amely mindezt lehetővé teszi, egy olyan telefon, amely kamerával és képernyővel van ellátva.



A képtelefon segítségével videokonferenciák is rendezhetők, miközben több résztvevő beszélhet egymással és láthatja egymást. Ezenkívül ügyszerzők, tervek, grafikák stb. is megjeleníthetők a képernyőn. Az alkalmazott képernyők nagysága különböző lehet. A munkahelyi beépített kis képernyők mellett esetenként nagy képernyőket is alkalmaznak, amelyeknek az az előnye, hogy egyidejűleg többen láthatják.



A képtelefon a jövőben főként az üzleti életben fog meghonosodni, azonban meglehetősen költséges, az ehhez szükséges széles sávú átvitel miatt (ami pl. fényvezető kábellel valósítható meg).

Alkalmazásával azonban jelentékeny idő és útiköltség takarítható meg.

A következő, gyakorlatból vett *példa* a képtelefon egy másféle alkalmazását mutatja.

Egy repülőtér segélyhelyén nem tudtak állandó orvosi ellátást biztosítani, ezért egy kommunikációs rendszert létesítettek, amely a betegek vagy az ápolónők lehetővé tette, hogy a helybeli klinika orvosával optikai és akusztikai kapcsolatba léphessen. Ezzel biztosították az esetenként szükséges kezelés orvosi ellenőrzését.

Szöveges kommunikáció

TELEX

Szöveges kommunikációnak nevezzük az információknak betűk, számjegyek és különleges jelek, tehát szöveg alakjában való átvitelét.

A **távíró** segítségével ilyen információk morzekódban már régóta továbbíthatók.

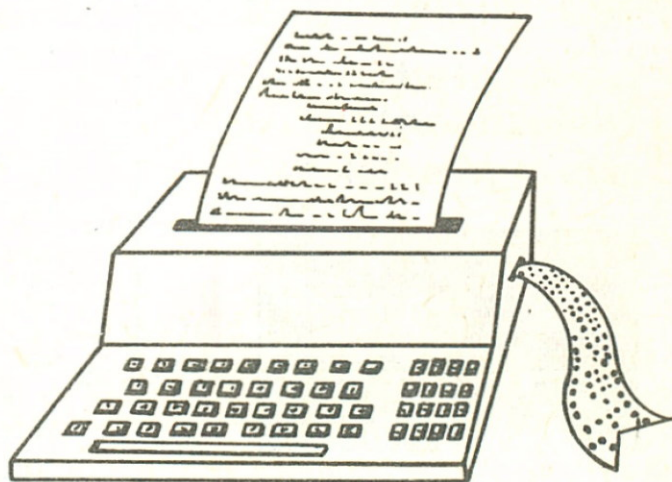
Azonban a telexhálózaton keresztül — amelyet már évtizedek óta használnak — gyorsabban és egyszerűbben bonyolítható le a szöveges kommunikáció. A **telexhálózatba** 150 országban 1,2 millió előfizető kapcsolódott be.

A telex rövid közlések vételét, tárolását és a címzethez való juttatását jelenti.

A **távgépíró** egy adó-vevő készülék, amelynek írógépbillentyűzete és nyomtatója, valamint kapcsolatlétesítő berendezése van. A legtöbb távgépíróhoz lyukszalaglyukasztó is csatlakozik. A lyukszalagon a billentyűzet segítségével lekopogott szöveg egyidejűleg tárolható is. Így a szöveg nyugodtan megfogalmazható. Alkalmos időpontban az elkészített lyukszalagot beolvassák, és a rajta levő hírt nagyobb sebességgel a címzettnek leadják.

A távgépírókat eddig rendszerint külön helyiségben, a telexszobában helyezték el, és pedig azért, mert a mechanikus telexgépek rendkívül zajosan dolgoznak.

Az újonnan kifejlesztett és egyre inkább elterjedő elektronikus távgépírók csendes működésűek, így hivatalokban is elhelyezhetők. A „hivatali távírással” a következő fejezetek egyikében foglalkozunk.



Szövegrendszer

Levelezés

A szöveggommunikáció egyik legrégebbi formája a közlemények postai úton való továbbítása.

Az **írógép** hosszú ideig az egyetlen technikai segédeszköz volt magán- és hivatalos leveleknek gyors és szép formában való leírására. Az írógép hátránya, hogy a szövegben előforduló hibák utólag nehezen javíthatók, és ha változtatásra van szükség, az egész levelet újra kell írni.

Néhány éve a **szövegkezelés** és a **szövegfeldolgozás** új eszközei jelentek meg a piacon, amelyeket **speciális alkalmazású számítógépeknek** nevezhetünk.

A szövegkezelést és a szövegfeldolgozást a következők szerint különböztetjük meg.

Szövegkezelésen a szövegek rögzítését (írását), tárolását és kinyomtatását értjük. Egy tárolt szöveg a kinyomtatás után is mindenkor rendelkezésre áll, és az igényeknek megfelelően javítható, átdolgozható és megváltoztatható, majd ezek után újra nyomtatható.

Szövegfeldolgozáson a már rögzített és tárolt szövegelemek felhasználásával új szöveg összeillesztését értjük. A szövegfeldolgozást mindenekelőtt üzleti levelek, körlevelek vagy jogi szerződések fogalmazásakor alkalmazzák, amelyek a személyes részletek kivételével szokásos szövegelemekből állíthatók össze.

Példa:

Előkészített
(változatlan)
szövegelemek

Az Ön által megrendelt alkatrészeket a kívánt határidőre, sajnálatunkra, nem tudjuk leszállítani.

A szállítás legkorábbi időpontjaként

jelölhetjük meg. Kérjük annak szíves közlését, hogy ez az időpont Önnek megfelel-e?

Szívélyes üdvözlettel:
FA-SUPER

FASUPER

N. - úrnak
Postafiók
1000 Budapest Pécs, 83. 3. 30.

Az Ön 1983. márc. 12-én kelt megrendelésére:
Igen tisztelt Uram!

Az Ön által megrendelt alkatrészeket a kívánt határidőre, sajnálatunkra, nem tudjuk leszállítani.

A szállítás legkorábbi időpontjaként
1984. jan. 6-át

jelölhetjük meg. Kérjük annak szíves közlését, hogy ez az időpont Önnek megfelel-e?

Szívélyes üdvözlettel:
FA-SUPER

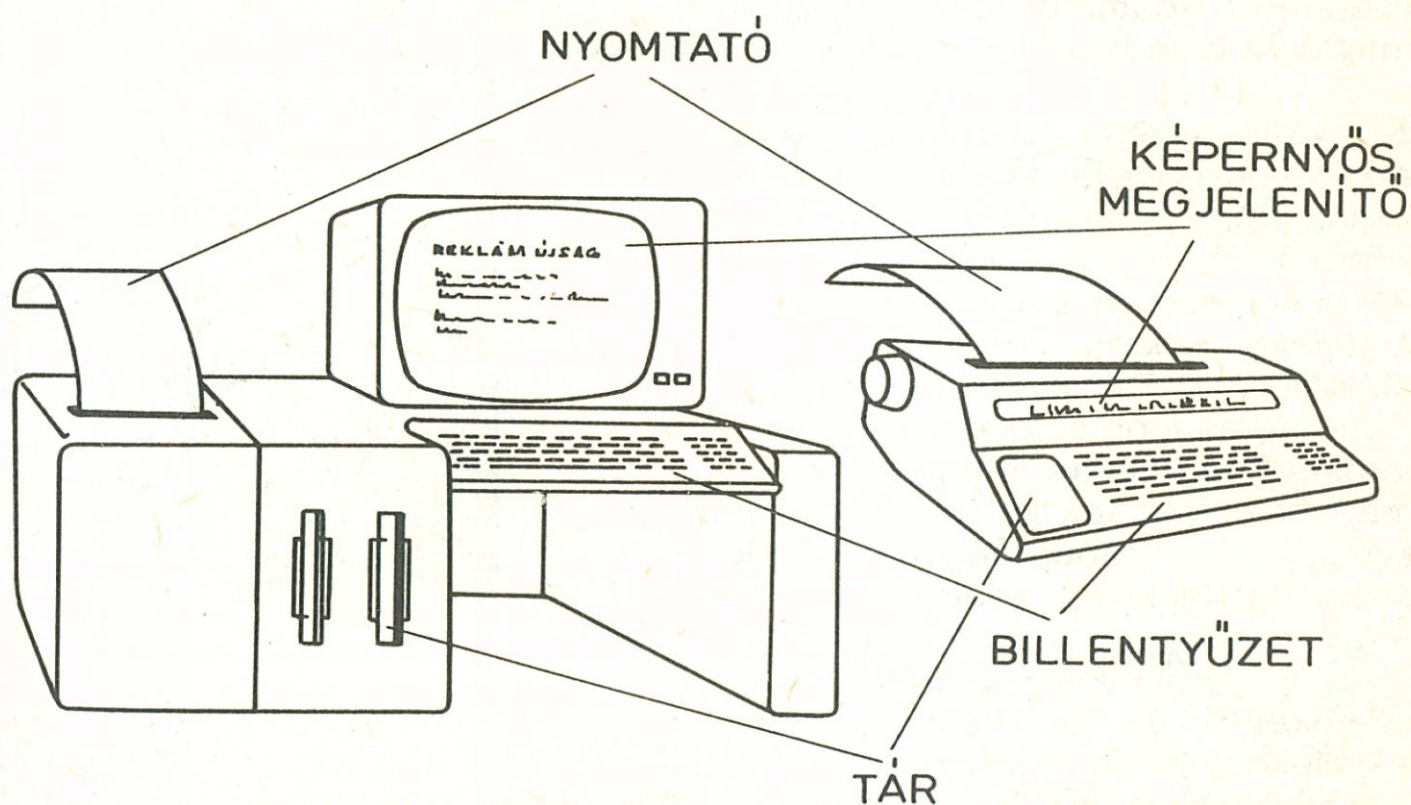
alkalmanként
változó

alkalmanként
változó

A szövegkezelés és szövegfeldolgozás eszközeit **szövegrendszereknek** vagy **szövegautomatáknak** nevezzük. Az egyszerűbb szövegrendszereket **tároló írógépeknek** is nevezik, mert ezek külsőleg alig különböztethetők meg az elektromos írógépektől.

E készülékek közös tulajdonsága, hogy velük a szöveg

- billentyűzettel rögzíthető,
- adattárban tárolható,
- képernyőn megjeleníthető,
- nyomtatón kinyomtatható.



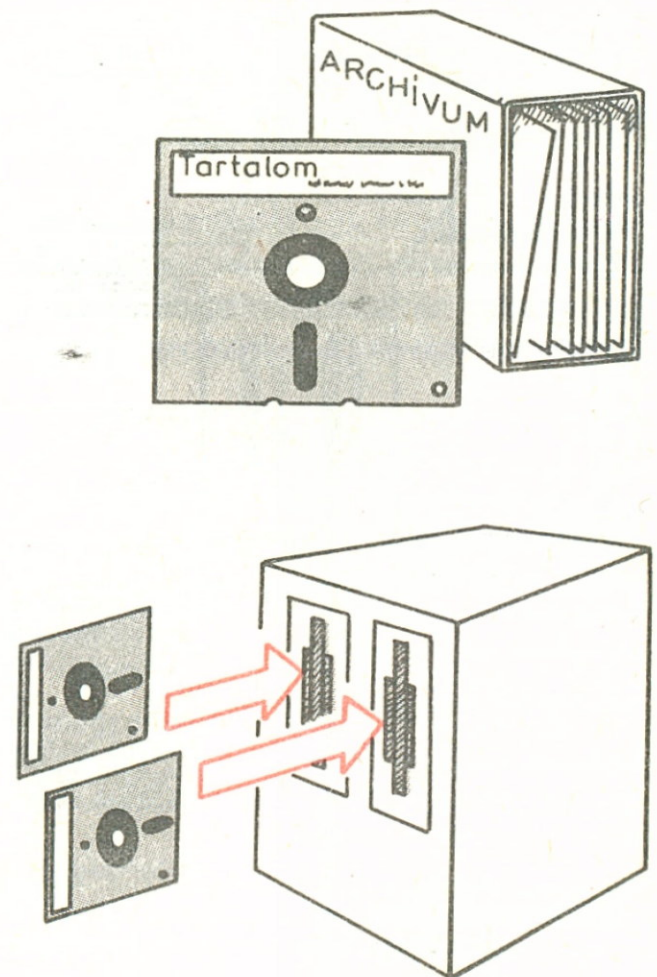
Tárak

Egy szövegrendszer tárában a szövegeket időlegesen tárolják azért, hogy azok javíthatók, átdolgozhatók vagy többször nyomtathatók legyenek. Azonban ha a szövegrendszernek nagy kapacitású tára van, akkor a tár *elektronikus* lerakatként (archívum) is felhasználható.

A **tároló írógépeknek** többnyire aránylag kis táruk van. Ezek félvezető-elemekből épülnek fel, és be vannak építve a szövegrendszerbe. Kapacitásuk néhány oldaltól néhány tucatnyi DIN A4 oldalig terjed.

A **fejlettebb szövegrendszerek** esetén a beépített félvezetőtár mellett cserélhető adathordozók is használhatók, ezáltal az adattár kapacitásának majdnem határtalan fokozására van lehetőség.

Szövegrendszerekhez a mágneses adathordozók alkalmazása terjedt el leginkább, amelyeknél a tárolandó információt egy mágnesezhető anyagra viszik rá. Leggyakrabban a *hajlékony mágneslemezt* alkalmazzák. Ez egy hanglemezhez hasonlít, és több százezer karakter tárolására alkalmas. A hajlékony mágneslemezt egy szövegrendszerhez kapcsolt meghajtóegységbe helyezik, ahol a tárolni kívánt adatokat a gyorsan forgó hajlékony lemezre írják, ill. igény esetén a tárolt adatokat kiolvassák. Használat után a hajlékony lemez a meghajtóegységből kivehető, és — mivel kis méretű és könnyen kezelhető — jól tárolható vagy elküldhető. Ha a tárolt szövegre később szükség van, akkor a lemeznek a meghajtóegységbe való helyezése után a szöveget a szövegrendszer képernyőjére vagy nyomtatójára vihetjük ki.



Megjelenítő (display)

A **megjelenítők** a szövegek papír nélküli megjelenítését, ellenőrzését és kijavítását teszik lehetővé.

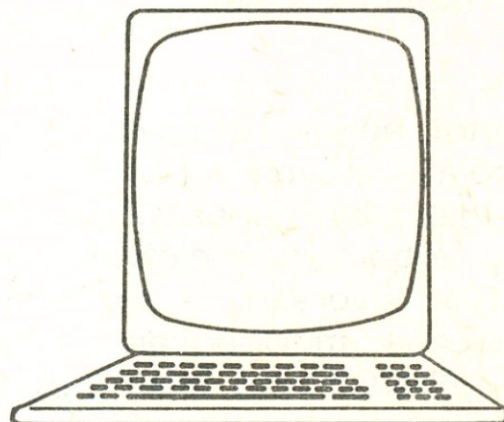
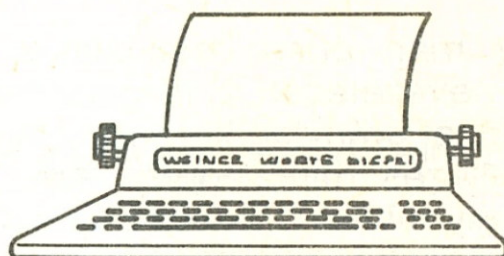
A különféle szövegrendszerek nagymértékben különböznek egymástól. Egy egyszerű megjelenítő pl. csak kevés szót tud megjeleníteni, míg a fejlettebb szövegrendszereknek olyan képernyőjük van, amelyen teljes szabványos A4-es oldalak jelenhetnek meg.

Olyan szövegrendszerek is vannak, amelyeknek a megjelenítői a szöveget ugyanúgy ábrázolják, ahogyan az később a papíron nyomtatásban megjelenik.

Ezek a következőket nyújtják:

- különféle betűtípusok (félvastag, kurzív),
- különböző betűknek megfelelő különböző betűszélesség,
- lapszél- (margó-) kiegyenlítés,
- automatikus szótagelválasztás és sok más egyéb.

Ezáltal a kéziratok a képernyőn nyomtatásra kész állapotba hozhatók, ami a nyomdákban eddig bonyolult munkamenettel járt.



A szövegrendszerek — mint már említettük — rendszerint különleges célra alkalmazott számítógépek, amelyeknek szigorúan megadott feladatokat — a szövegkezelést és szövegfeldolgozást — kell megoldaniuk, kezelésüket is ez határozza meg. Használójuk ezért gyakran észre sem veszi, hogy számítógéppel dolgozik.

Vannak azonban olyan szövegrendszerek is, amelyek használójuk által „programozhatók”, azaz más feladatokra is alkalmazhatók.

Másrészt sok számítógép részére léteznek olyan programcsomagok, amelyek ezeket a sokoldalúan használható számítógépeket átmenetileg szövegrendszerré teszik.

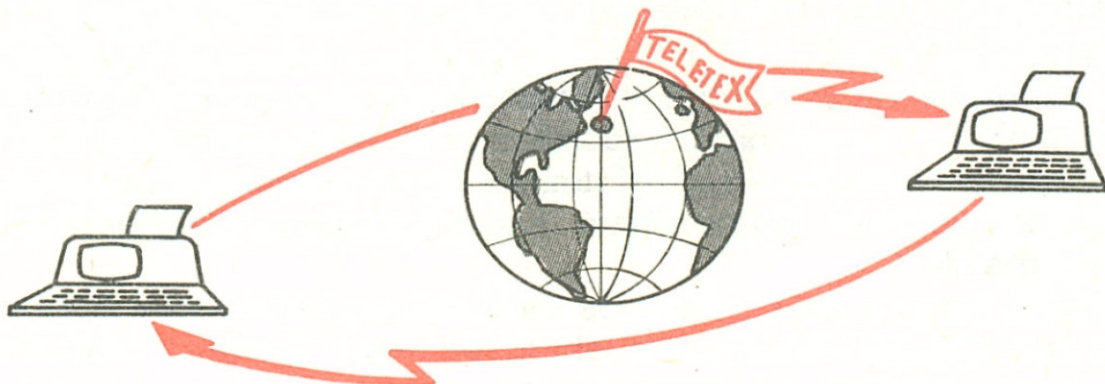
Ebből kitűnik, hogy a szövegkezelés és a szövegfeldolgozás alapjában véve egyike a számítógép alkalmazási lehetőségeinek. Gyakran csak értelmezés kérdése, hogy valamely készüléket szövegrendszernek vagy irodai számítógépnek nevezünk-e — a hardver lényegében azonos.

TELETEX

A szövegkezelés és szövegfeldolgozás fejlődése egyszerűbbé, olcsóbbá és komfortosabbá tette azokat a munkákat, amelyeket azelőtt sok írógéppel, sok munkaóra alatt és nagyon sok papír felhasználásával lehetett csak elvégezni. A modern technikának azonban azonnal vége szakadt, mielőtt a legmodernebb eszközökkel megszerkesztett szövegeket a címzetthez kellett továbbítani. Ekkor ugyanis a szövegeket ki kellett nyomtatni és postára adni, vagy esetleg más adathordozón, pl. hajlékony mágneslemezen elküldeni. Az elektronikus készülékek közötti összekötő tag a papír vagy valamely más információhordozó volt. Kézenfekvő, hogy meg kellett keresni a szövegek *elektronikus továbbításának* módját.

Bár már régen használatos volt az elektronikus továbbítás egy formája — nevezetesen a távgépírásnál alkalmazott módszer —, azonban a telexhálózat aránylag kis sebességű, és a távgépíróval továbbított szövegek nem levelezés minőségűek. Ami hiányzott, az egy olyan szövegállomás volt, amely minden irodában felállítható, és amely a szövegrendszerben levelezés minőségben előállított szöveget elektronikus úton a címzett szövegállomáshoz tudja továbbítani.

A 80-as évek elején létrehoztak egy ilyen célú nyilvános világméretű szolgálatot, amely TELETEX néven ismeretes. A TELETEX szolgálat végpontjai speciális szövegrendszerek, amelyekkel a szövegeket nem csak írni, javítani és változtatni lehet. Olyan hívóberendezéssel is rendelkeznek, amellyel egy vagy több teletex-előfizetőt felhívhatnak, amelyeknek azután a szövegeket nagyon gyorsan — oldalanként néhány másodperces teljesítménnyel — levelezés minőségben leadhatják. Átmenetként a hagyományos telexszolgáltatáshoz a telex-előfizetőkkel való levelezés is lehetséges, azonban ebben az esetben csak „telex minőségben”, és sokkal kisebb sebességgel.





1. Kik a kommunikációs partnerek a személyi kommunikációnál?

Emberek. Gépek. Emberek és gépek.

2. Milyen elemekkel kell a személyi kommunikáció végpontjainak rendelkezniük, hogy megfeleljenek az emberi érzékszervi észleléseknek?

A bevitelre

A kivitelre

Tapintás:

Látás:

Beszéd:

Hallás:

Látás:

3. Nevezzen meg néhány járulékos szolgáltatást, amelyeket a telefonok a beszédkommunikáció részére nyújtanak:

a) ha azokba megfelelő számítógép-részekységek vannak beépítve?

.....

.....

.....

.....

b) Ha kamerával és képernyővel vannak összekapcsolva?

.....

.....



Válaszok

1. Kommunikációs partnerek a személyi kommunikációnál:

 Emberek.

Bár a kommunikációs folyamatban gépek is részt vesznek, csak közvetítő szerepük van, így azok nem kommunikációs partnerek.

2. A személyi kommunikáció végpontjainak a következő elemekkel kell rendelkezniük azért, hogy megfeleljenek az emberi érzékszervek észlelőképességének:

Bevitelre:

Kivitelre:

Tapintás: billentyűzet.

Látás: nyomtató képernyő.

Beszéd: mikrofon.

Hallás: hangszóró.

Látás: kamera.

3. Járulékos szolgáltatások, amelyeket telefonok a beszédkommunikációnak nyújthatnak:

a) ha azokba megfelelő számítógép-részekységek vannak beépítve:

hívószám-tárolás,

foglalt állomás automatikus újrAhívása,

billentyűs hívás,

beérkező beszélgetések mellékállomásra való átkapcsolása,

bekapcsolódás beszélgetésekbe,

konferenciakapcsolások stb.

b) Ha kamerával és képernyővel vannak összekötve:

képtelefon.



4. A szövegkommunikáció említett lehetőségei közül melyikkel lehet a leggyorsabban szövegeket szerkeszteni és továbbítani?

- | | | | |
|--|-----------------------|--------------|-----------------------|
| Telex. | <input type="radio"/> | Morzetávíró. | <input type="radio"/> |
| Szövegszerkesztés
szövegrendszerekkel,
szövegtovábbítás
távmásolóval. | <input type="radio"/> | | |
| | | Teletex. | <input type="radio"/> |

5. A szövegkommunikáció mely berendezései szorítják ki a használatból növekvő mértékben az írógépet?

.....

6. a) Melyek a szövegrendszerek elemei?

- | | | |
|--|---------------|-----------------------|
| | Tár. | <input type="radio"/> |
| | Megjelenítő. | <input type="radio"/> |
| | Nyomtató. | <input type="radio"/> |
| | Telefon. | <input type="radio"/> |
| | Billentyűzet. | <input type="radio"/> |
| | Mikrofon. | <input type="radio"/> |

b) A teletexnél mely berendezés jön még ezekhez?

.....



Válaszok

4.



A teletex a szövegkommunikációnak az a lehetősége, amellyel szövegek a leggyorsabban szerkeszthetők és továbbíthatók.

5. A szövegrendszerek vagy tároló írógépek szorítják ki a hagyományos írógépet növekvő mértékben a használatból.

6. a) A szövegrendszerek elemei:



Tár.



Megjelenítő.



Nyomtató.



Billentyűzet.



b) A teletex a *szövegek továbbítására* szolgáló berendezéssel egészül ki.

Adatkommunikáció

Az adatkommunikáció sajátosságai

Míg a személyi kommunikáció esetén az információk emberektől indulnak ki és emberekhez érkeznek, addig az adatkommunikáció esetén legalább az egyik kommunikációs partner a számítógép.

Adatkommunikációról van szó, ha pl. az ember kérdést intéz a számítógéphez, vagy utasítást ad annak, és a számítógép a kérdést megválaszolja, vagy a végrehajtott műveletek eredményét az emberrel közli. De adatkommunikáció az is, ha két számítógép kommunikál egymással.

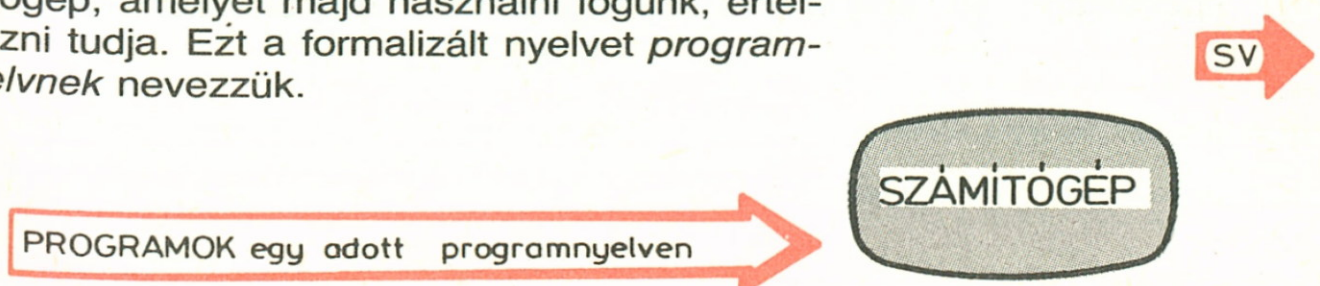
Az adatkommunikáció esetén olyan információk cserélődnek ki, amelyeket a számítógépnek kell feldolgoznia. Ezek az információk lehetnek: **adatok** vagy **programok**.

Adatok lehetnek pl. matematikai változók értékei, a számítógép által megoldandó egyenletek, de lehetnek a számítógépben tárolt szövegek és képek is, amelyeket a számítógép valamilyen utasításra kiad. A programok olyan utasítások, amelyek a számítógéppel közlik, hogy az adatokkal mit kell tennie.



Az adatkommunikációnál kicserélt információknak tehát két fajtája van: **adatok** és **programok**.

Az emberek a szóbeli érintkezés során a *természetes* köznyelvet használják. Akkor azonban, amikor az ember a számítógéppel akar valamit közölni, ezt csak egy korlátozott szókinccsel teheti meg. Ahhoz, hogy a számítógépnek utasításokat adhassunk, egy formalizált nyelvre van szükségünk, azaz egy olyan nyelvre, amelynek minden kifejezése szigorúan egyértelmű, és azt a számítógép, amelyet majd használni fogunk, értelmezni tudja. Ezt a formalizált nyelvet *programnyelvnek* nevezzük.

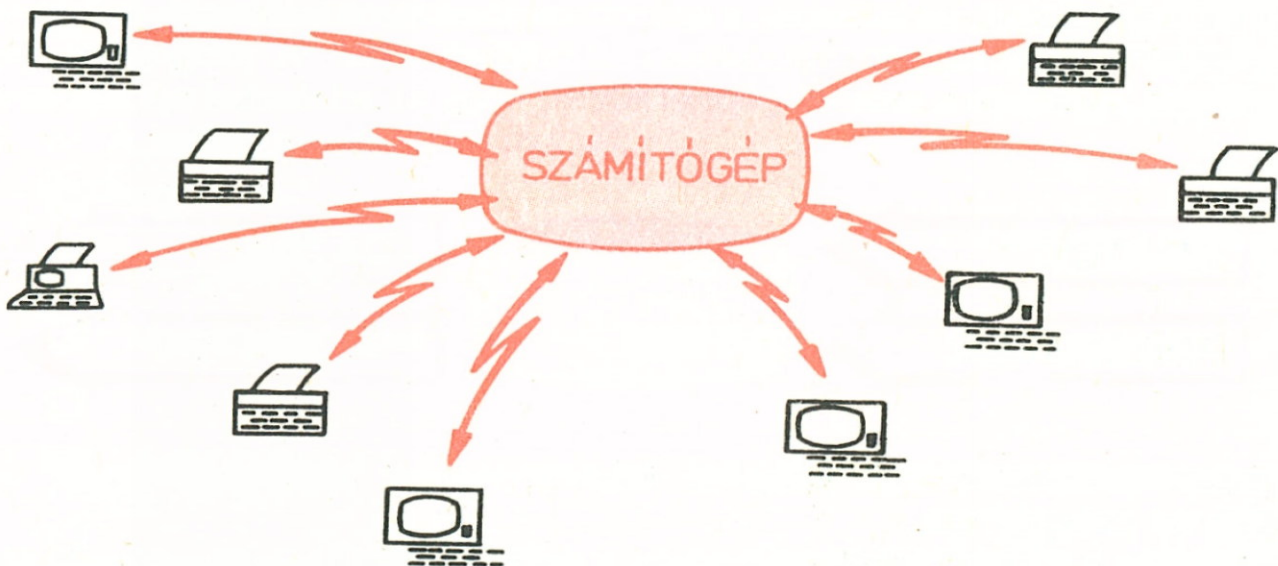


Az adatkommunikáció formái

Távfeldolgozás

A számítástechnika kezdeti időszakában a számítógép felhasználójának kellett a géphez mennie, hiszen az adatok és programok be- és kivitelét végző összes berendezést és a számítógép további egységeit egy *központi számítóközpontban* helyezték el.

Mivel azonban sokkal célszerűbb az adatokat ott bevinni vagy kiadni, ahol azok keletkeznek és ahol azokra szükség van — azaz a hivatalokban és a munkahelyeken —, azért a fejlődés eredményeként a **számítógép-terminálokat**, azaz azokat a készülékeket, amelyeken keresztül a számítógéppel érintkezésbe léphetünk, ott állították fel, ahol azokra szükség van, és ezeket vezetékekkel csatlakoztatták egy központi számítógéphez.



Így a számítógéppel mintegy „helyközi távbeszélést” folytathatunk. Az adatkommunikáció e fajtáját **távfeldolgozásnak** nevezzük.

A távfeldolgozás esetén a géptől távol dolgozó jogosultak részére lehetővé válik a számítógép felhívása és a gép használata a távbeszélő-hálózaton vagy speciálisan az adattovábbításra létesített adathálózaton keresztül.

Számítógép-hálózat

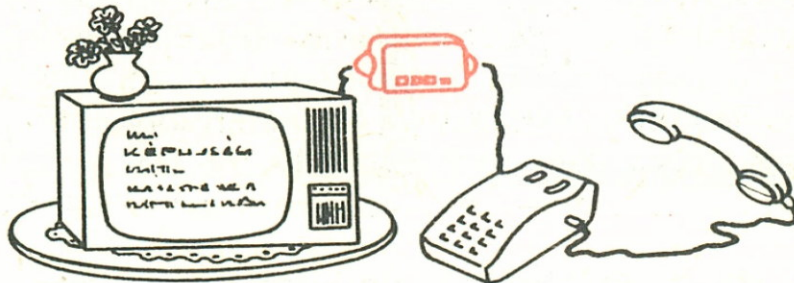
Adatkommunikációról — mint már említettük — nemcsak akkor beszélünk, ha az információk egy terminál és egy számítógép között cserélődnek ki, hanem akkor is, ha számítógépek állnak egymással összeköttetésben, amelyek a feldolgozás érdekében egymásnak adatokat továbbítanak. Mivel ez esetben a számítógépek vezetékekkel vannak egymással összekötve, az adatkommunikációnak ezt a formáját **számítógép-hálózatnak** nevezzük.

A számítógépek összekapcsolásának különböző okai és megvalósítási formái vannak. A számítógép-hálózatok témájával részletesebben egy későbbi fejezetben foglalkozunk. Itt csak azt állapítjuk meg, hogy egy vagy több számítógépnek adatfeldolgozás céljából (adatok és programok kicserélése vagy számítógép-szolgáltatások rendelkezésre bocsátása) való összekapcsolása ugyancsak egy formája az adatkommunikációnak.



Teledata*

Egy **telefon** és egy közönséges **tv-készülék****, melyek egy **segédberendezés***** segítségével vannak egymással összekötve, alkalmassá tehetők arra, hogy az előfizető egy számítógépet felhívjon, és a számítógéptől információkat kérjen a képernyőre.



* E szolgáltatás elnevezésére a CCITT az „interactive videotex”, míg az IEC a „videotex” fogalomnevet ajánlja. A „teledata” fogalomnév az ajánlott magyar megnevezés. (Lektor.)

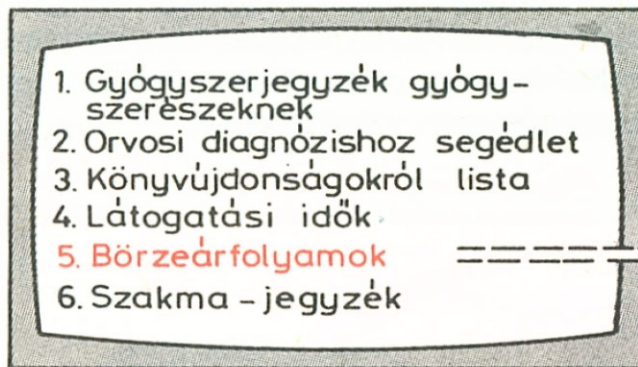
** A szóban forgó szolgáltatáshoz valójában egy megfelelő dekódolóval ellátott tv-készülékre van szükség. (Lektor.)

*** Modem. (Lektor.)

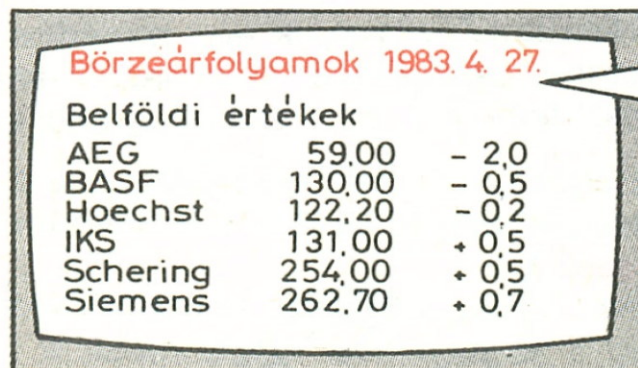
E célra egy információ-központban számítógépeket szerelnek fel, amelyeknek egyedüli feladata, hogy információkat kezeljenek, és azokat kívánságra rendelkezésre bocsássák. E központilag kezelt információkat **információbázisoknak** vagy **adatbázisoknak** nevezzük.

Példa:

Egy előfizető az aktuális börzeárfolyamokat szeretné látni. A telefonbillentyűzet vagy számtárca segítségével felhívja a számítógépet. Ez egy elektronikus lapozható tartalomjegyzék formájában áttekintést ad az általa felkínált szolgáltatásokról.

- 
1. Gyógyszerjegyzék gyógyszerészeknek
 2. Orvosi diagnózishoz segédlet
 3. Könyvújdonságokról lista
 4. Látogatási idők
 5. **Börzeárfolyamok**
 6. Szakma - jegyzék

Ebből az előfizető azt a kulcsszót választja ki, amely alapján a kívánt információt feltehetően megkaphatja, és a tv-készülékkel vezeték nélküli kapcsolatban álló kézi billentyűzeten megadja a megfelelő azonosítószámot, esetünkben az 5-öt. A kívánt információk (esetünkben a börzeárfolyamok) néhány másodperc múlva megjelennek az előfizető képernyőjén.



Börzeárfolyamok 1983. 4. 27.

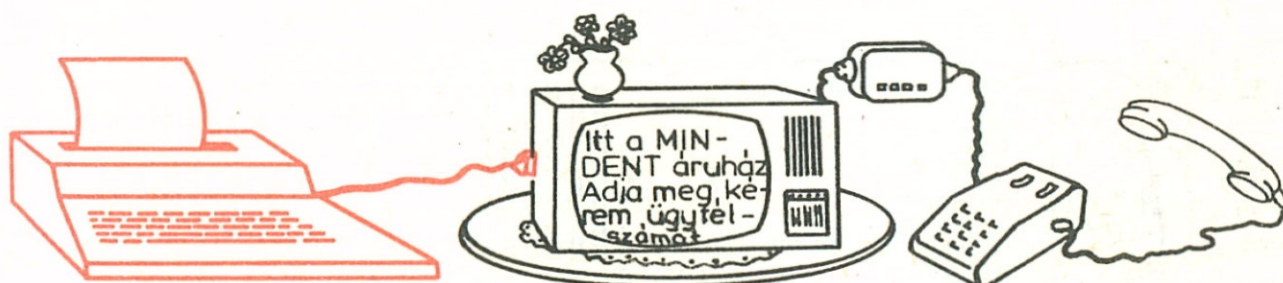
Belföldi értékek		
AEG	59,00	- 2,0
BASF	130,00	- 0,5
Hoechst	122,20	- 0,2
IKS	131,00	+ 0,5
Schering	254,00	+ 0,5
Siemens	262,70	+ 0,7

Egy előfizető így sok számítógéppel és adatbázissal létesíthet kapcsolatot, vagyis a felkínált információmennyiség számára majdnem határtalannak mondható.

Minden előfizetőnek információkat kínálhatnak: a sajtóügynökségek, a hatóságok, az utazási irodák, a közlekedési vállalatok, a fogyasztói szervezetek, a biztosítási társulatok, a bankok, és általában az ipar és a kereskedelem. Az előfizetők érdeklődhetnek többek között: a politika legújabb híreiről, a közgazdaságról, a sportról, a helyi eseményekről és rendezvényekről, a menetrendekről, az időjárásról, a teszteredményekről, a piaci árakról, az apróhirdetésekről stb. De tárolhatók pl. az áruházak katalógusai, kézikönyvek vagy ételreceptek is, és ezek minden előfizető által bármikor lehívhatók.

Emellett létezhetnek olyan információk is, amelyek csak különleges előfizetőknek állnak rendelkezésre. Pl. gyógyszerjegyzékhez csak gyógyszerészek férhetnek hozzá, jogi információt csak jogászok hívhatnak le, egyesületek belső ügyeiről csak az egyesület tagjai informálódhatnak.

Mindezeket az értesüléseket az előfizető otthon kaphatja meg, ha van telefonja, tv-készüléke és segédberendezése. Ez az együttes egyszerűen kibővíthető pl. egy szám- és betűbillentyűs távműködtetővel, vagy egy kis *személyi számítógéppel*, és ezzel máris tetszőlegesen kiterjeszthetők egy számítógép szolgáltatásainak lehetőségei.



Minden előfizető, akinek van ilyen járulékos berendezése, kialakíthat saját **személyes adatbankot** is, amelyet vagy csak saját maga, vagy az általa feljogosított személyek használhatnak. Így pl. készíthet kimutatásokat folyószámlái állásáról, személyes határidőkről, tárolhat érdekes újságcikkeket, és ezek szükség esetén bármikor lehívhatók. Ezzel a kibővített rendszerrel nemcsak információk lehívása lehetséges, hanem futtatható *saját program* is, és *saját adatok* is feldolgozhatók. Pl. kamatokat, adókat, járadékokat ki lehet számítógéppel számítani, ill. oktató- vagy játékprogramok is lehívhatók.

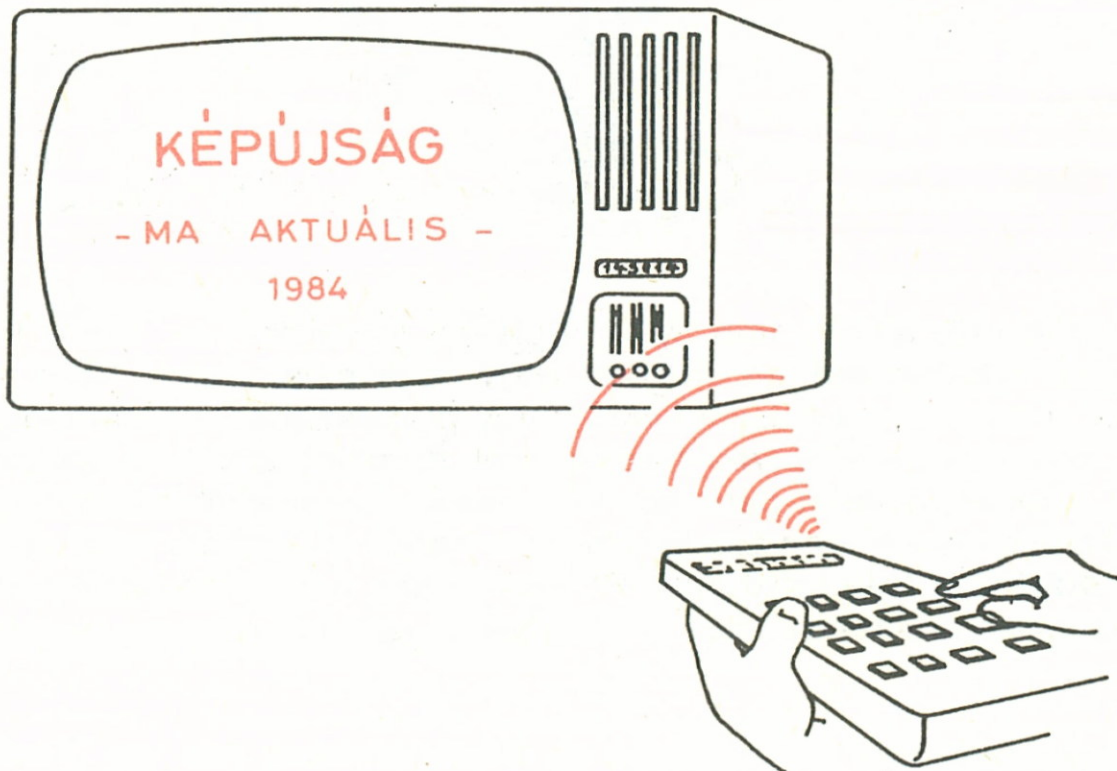
Képújság*

A képújság (videotext) szolgáltatást főként magánhasználatra szánják. Ehhez egy tv-vevőkészülékre és egy kiegészítő berendezésre (dekódolóra) van szükség.

A dekódoló gondoskodik arról, hogy a tv-félképek között kisugárzott szövegek a képernyőn megjeleníthetők legyenek.

Míg a teledata szolgáltatás esetén az információk vezetéken jutnak a vevőkészülékhez, addig a képújságnál azokat tv-jelekkel együtt sugározzák ki: a telefoncsatlakozásra itt nincs szükség.

A képújság a Német Szövetségi Köztársaságban a nyolcvanas évek közepétől áll rendelkezésre. A távműködésű kézi billentyűzet segítségével néhány általánosan is hozzáférhető értesülés szerezhető be: pl. börzehírek, menetrend vagy színházprogramok. Ezek ugyanúgy kapcsolhatók, mint amikor az 1., 2. vagy 3. tv-program között választunk. Egyéni információkat azonban — mint pl. a bankfolyószámla állása — nem kaphatunk. Ezt a lehetőséget csak a teledata szolgáltatás nyújtja.



* E szolgáltatás elnevezésére a CCITT a „broadcast videotex”, míg az IEC a „teletext” fogalomnevet ajánlja. A „képújság” a hazánkban 1982. nov. 30-án megkezdett ilyen irányú szolgáltatás időközben elfogadottnak tekinthető megnevezése. (Lektor.)

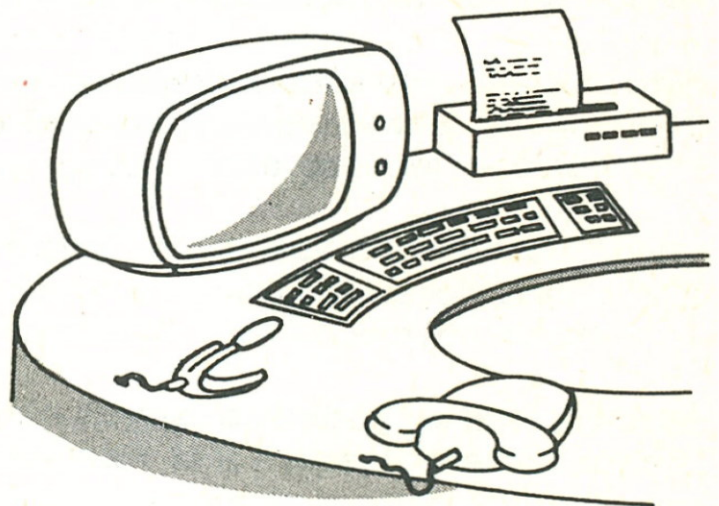
Integrált irodai rendszerek

Ahogy egy telefon, tv-készülék és néhány kiegészítő berendezés segítségével egy vagy több számítógép erőforrásai igénybe vehetők, ugyanúgy ezek az erőforrások természetesen egy hivatal bármely munkahelyéről is elérhetők, ha a hivatalnak erre van távközlési lehetősége. Általános törekvés, hogy a hivatalokban minden kommunikációs készüléket egyetlen készülékké egyesítsenek, függetlenül attól, hogy a személyi kommunikáció vagy az adat-kommunikáció eszközeiről van-e szó. Ezt az univerzális kommunikációs eszközt egy kisméretű számítógéppel szerelik fel, és összekapcsolják a legkülönbözőbb távközlési hálózatokkal.

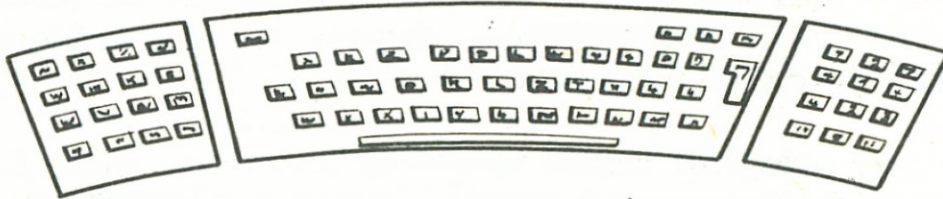
Az eredmény egy **integrált irodai rendszer** (electronic mail: EMS), amely a munkahelyen a dolgozó valamennyi információigényét kielégíti.

Használója lényegében az alábbiakkal dolgozik:

- billentyűzet,
- képernyő,
- mikrofon és hangszóró (telefon).



Míg a különféle hagyományos hivatali készülékeknek, mint az írógép, mechanikus működésű számológép, távíró, számítógép-terminál és telefon külön-külön saját billentyűzetük van, amelyek még különböző elrendezésűek is, addig a jövő integrált munkahelyének csak **egyetlen billentyűzete** lesz, amely egy számjegy-, egy betű- és egy funkcióbillentyű-mezőből áll, amelyen az összes szükséges bevitel elvégezhető.

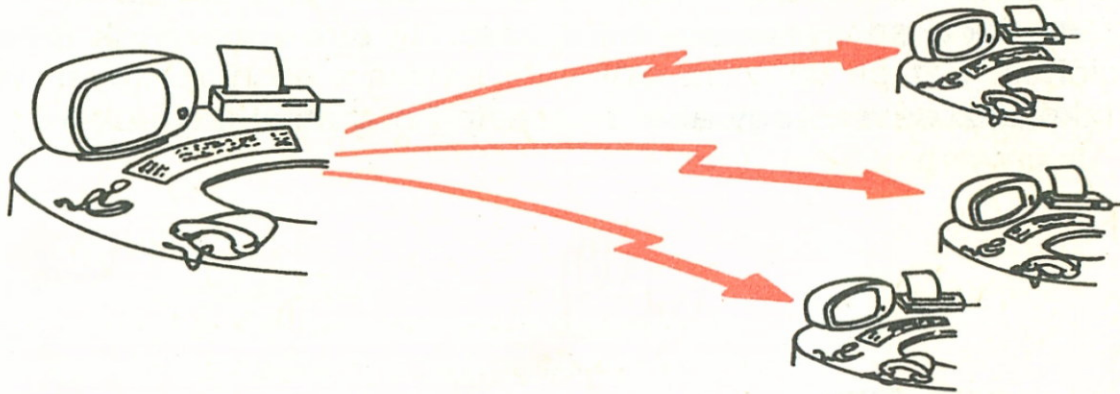


Így ezzel a billentyűzettel nemcsak leveleket lehet gépelni, hanem az adatok a hálózatra kapcsolt számítógépbe is bevihetők. Ezenkívül a billentyűzet segítségével tetszőleges telefonszámok is megadhatók, vagy más csatlakoztatott integrált munkahelyek és számítógépek is felhívhatók.

Az integrált munkahely **képernyőjének** ugyancsak több funkciója van: segítségével a begépelte információk ellenőrizhetők, és az összes beérkező információ — lehet az akár szöveg, akár kép — megjeleníthető.

Az integrált munkahelyen majdnem minden hivatali munka elvégezhető.
Néhány példa:

- Különböző **levelek** írhatók, attól függően, hogy milyen lehetőségei vannak a szövegautomatának, ezenkívül azok egy vagy több címzetthez elektronikusan továbbíthatják.

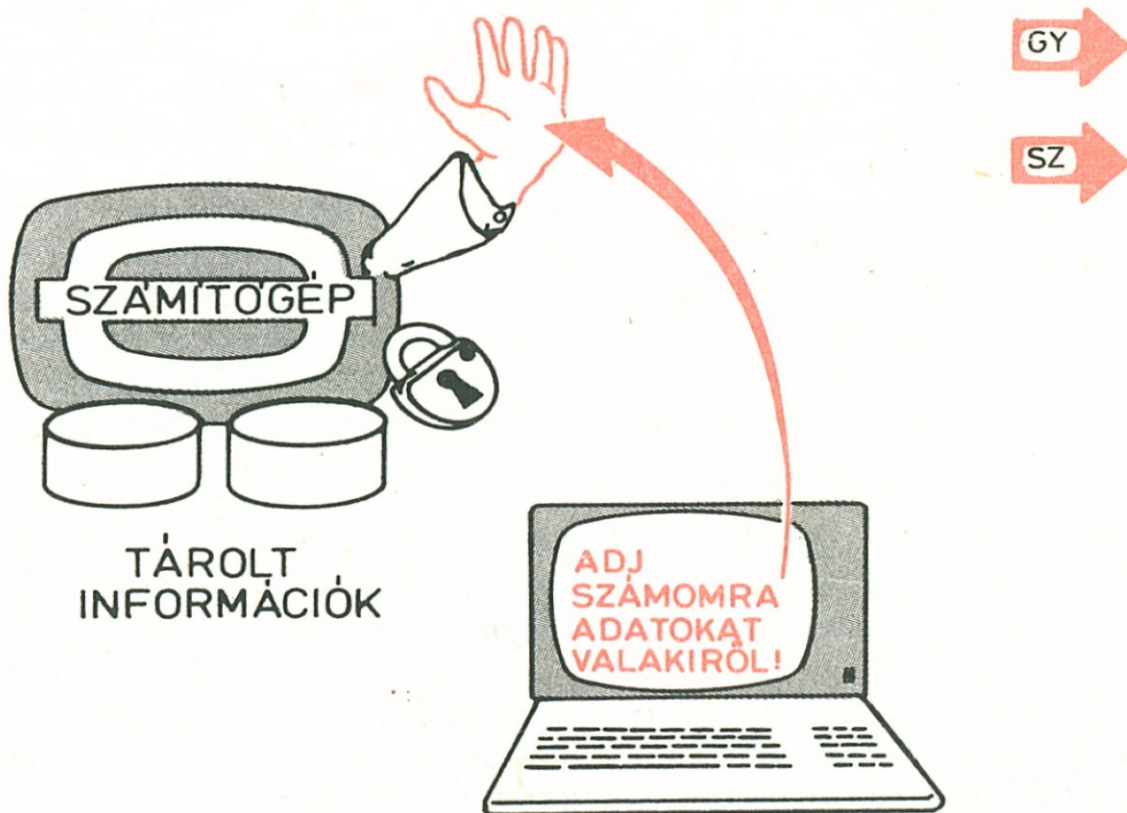


- Különböző **információanyagok**, amelyek az integrált munkahelyhez érkeznek — legyenek azok a vállalat minden dolgozójához szóló rövid értesítések, üzleti levelek stb. — automatikusan tárolhatók, és alkalmas időpontban lehívhatók. Ez egy elektronikus postaszekrény funkciójának felel meg.
- **Iráttár**, amelyhez nincs szükség papírra, mivel egy kommunikációhálózatra kapcsolt számítógép segítségével a tárolt és keresett ügydarabot néhány kulcsszó alapján gyorsan megtalálja, és a képernyőn megjeleníti.
- A billentyűzet és a képernyő **szokványos számítógép-terminálként** is használható. Tehát információk hívhatók le a kapcsolt adatbázisokból, saját adatállomány építhető fel, saját programok is kifejleszthetők, és azok lefuttathatók vagy létesíthető a megjegyezni valókról jegyzet, határidőnapló vagy címjegyzék.
- **Telexként** vagy **teledataként** is használhatók.

Igen nagy mennyiségű adatot csak számítógéppel tudunk kezelni. A számítógép segít az adatok átvitelében, ha az átvitelhez használt vezetékben hibásodás lép fel más átviteli utat létesít, ellenőrzi a beérkező információkat, és adott esetben újra kéri azokat.

Megkönnyíti az integrált munkahely dolgozójának munkáját, amennyiben segít a levelek megírásában és korrigálásában, és átveszi az „elektronikus irattár” szerepét. Sok adminisztrációs munkától mentesíti használóját.

De a számítógép arra is ügyel, hogy csak az arra illetékesek juthassanak információhoz. Ezt pl. vagy az előfizetői sorszám ellenőrzésével, vagy egy kódszó lekérdezésével, vagy akár beszéd felismeréssel tudja elérni.



Az integrált irodai rendszerek hálózata nemcsak a papírádatot korlátozza, hanem kiküszöböli az ún. *adathordozó-változásokat* is: a munkaidő igen nagy részét ma ui. arra fordítják, hogy az írásbeli kommunikációkat egy más kommunikációs közegre vigyék át, pl. a képernyőről űrlapokra, űrlapokról írógéppel írt lapokra, írógéppel írt lapokról távíróra és onnan lyukkártyára, hogy végül az információk ismét képernyőn legyenek megjeleníthetők.



1. Mikor beszélünk adatkommunikációról?

Ha két ember kommunikál egymással.

Ha egy ember egy számítógéppel kommunikál.

Ha két számítógép elektronikus adatfeldolgozás céljából kommunikál egymással.

2. Jelölje meg kereszttel, hogy a következő kommunikációféleségek melyikénél beszélhetünk adatkommunikációról!

Teledata.

Számítógép-hálózat.

Képújság.

Szövegkezelés.

Teletex.

Képtelefon.

3. Az információk mely fajtái cserélhetők ki adatkommunikáció során?

..... és

4. a) Az adatkommunikáció mely fajtája valósul meg, ha egy Rómában levő munkahelyen elhelyezett terminál használója egy Wiesbadenben levő számítógéphez intéz kérdéseket?

.....

b) Mely hálózatokon jöhet létre ilyen kapcsolat?

..... vagy



Válaszok

1. Adatkommunikációról van szó:



Ha egy ember egy számítógéppel kommunikál.



Ha két számítógép kommunikál egymással elektronikus adatfeldolgozás céljából.

2. A következő kommunikációfajták esetén beszélhetünk adatkommunikációról:



Teledata.



Számítógép-hálózat.



Képújság.



3. Az adatkommunikáció során **adatok** és **programok** cserélhetők ki.

4. a) Ha egy Rómában levő munkahelyen elhelyezett terminál használója egy Wiesbadenben levő számítógéphez intéz kérdéseket, akkor **távfeldolgozás** valósul meg.

b) Ilyen kapcsolat a **távbeszélő-hálózaton** vagy az **adathálózaton** keresztül jöhet létre.



5. a) Hogyan nevezzük a központilag kezelt információkat, amelyekhez minden jogosult előfizető hozzájuthat, akinek megfelelő előfizetői készüléke van?
.....
- b) Mely előfizetői készülékekre van a magán-előfizetőnek szüksége (egy megfelelő segédberendezés mellett), hogy a teledata szolgáltatásban részt vehessen?
.....
- c) Mely információk hívhatók le az adatbázisokból?
- Valamennyi információ, amely az adatbázisban tárolva van.
- Valamennyi információ, amelyre igény merül fel.
- Csak az adatbázisban tárolt olyan információk, amelyek lehívására valamely előfizető jogosult.
6. Képűjság és teledata; mindkettő az adatkommunikáció egy formája. Egészítse ki a következő mondatot:
- A esetén az információk a telefonvezetéken érkeznek a tv-vevőkészülékhez.
- A esetén azokat a tv-antennával veszik.
7. Sorolja fel, hogy a hagyományos irodai munkák az integrált munkahelyen milyen berendezéssel végezhetők!
Távmásolás 1; elektronikus irattár 2; központi adatbázis igénybevétele 3; szövegszerkesztés valamely szövegrendszerrel 4.
- Levelezés iratrendezőbe helyezése.
- Levélírás írógéppel.
- Szakmai dolgozatok kölcsönvétele és lemásolása.
- Rajzok postán történő továbbítása.



Válaszok

5. a) Információbázisoknak vagy adatbázisoknak nevezzük azokat a központilag kezelt információkat, amelyekhez minden jogosult előfizető hozzájuthat, akinek van megfelelő előfizetői készüléke.
- b) A megfelelő segédberendezésen kívül az előfizetőnek *telefonra* és *tv-vevőkészülékre* van szüksége, hogy a teledata szolgáltatásban részt vehessen.
- c) Az adatbázisokból lehívhatók:
-
-
- Csak az adatbázisban tárolt olyan információk, amelyek lehívására valamely előfizető jogosult.
6. Az információk teledata esetén a telefonvezetéken érkeznek, a képűtség esetén azokat a tv-antennával veszik.
7. A hagyományos irodai munkák az integrált munkahelyeken a következő berendezésekkel végezhetők:
A számjegyek jelentése: távmásolás 1; elektronikus irattár 2; központi adatbázis igénybevétele 3; szövegszerkesztés valamely szövegrendszerrel 4.
- 2 A levelezés iratrendezőbe helyezése.
- 4 Levélírás írógéppel.
- 3 Szakmai cikkek kölcsönzése és másolása.
- 1 Rajzok postai továbbítása.

Átviteli vonalak

Az előzőekben a kommunikáció különféle fajtáit mutattuk be, és bemutattuk azokat a lehetőségeket, amelyeket a számítástechnika alkalmazása ad a kommunikáció elősegítésére.

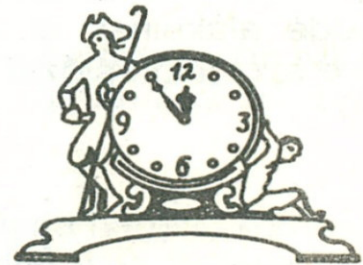
A következőkben megvizsgáljuk, hogy milyen lehetőségek vannak az információk átvitelére, milyen hálózatok állnak ehhez rendelkezésre, és hogyan építhetők ki összeköttetések.

Analóg és digitális átvitel

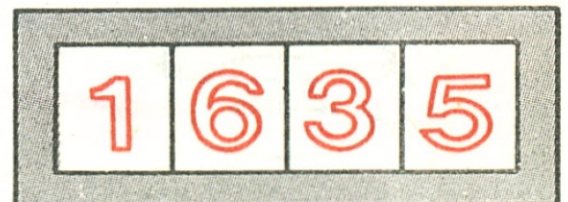
Az „analóg” és „digitális” jelzőket nemcsak ott használjuk, ahol információk átviteléről, azaz összeköttetésekről van szó, ezekkel a fogalmakkal gyakran találkozhatunk mindazokon a területeken, ahol a számítógépet alkalmazzák. Az **analóg** szó jelentése hasonló, valaminek bizonyos szempontból megfelelő. A **digitális** szó jelentése számjegyes (latinul digitus: ujj, angolul digit: számjegy).

Az „analóg” és „digitális” azt a módot jelentik, ahogyan az **információk ábrázolhatók**. Az információkat ábrázolnunk kell, annak érdekében, hogy rögzíthessük, átvihezzük, feldolgozhassuk, terjeszthessük, sokszorosíthatjuk stb. őket.

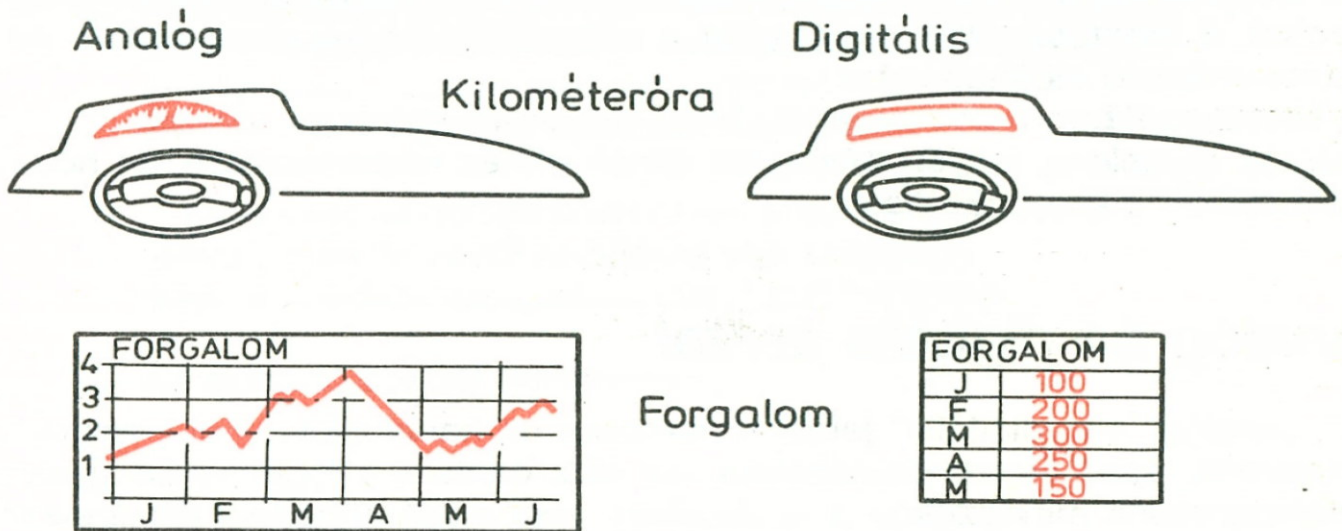
Egy óra, amelynek mutatója van, az *analóg* ábrázolás tipikus példája: az ilyen óra úgy mutatja az időt, hogy mutatója a múlt idővel folyamatosan halad az időjelölések felett.



Egy digitális óra ezzel szemben számjegyekkel, tehát *digitálisan* mutat egy határozott értéket. Egy bizonyos időegység, pl. egy másodperc elmúltával egy másik időértéket mutat.

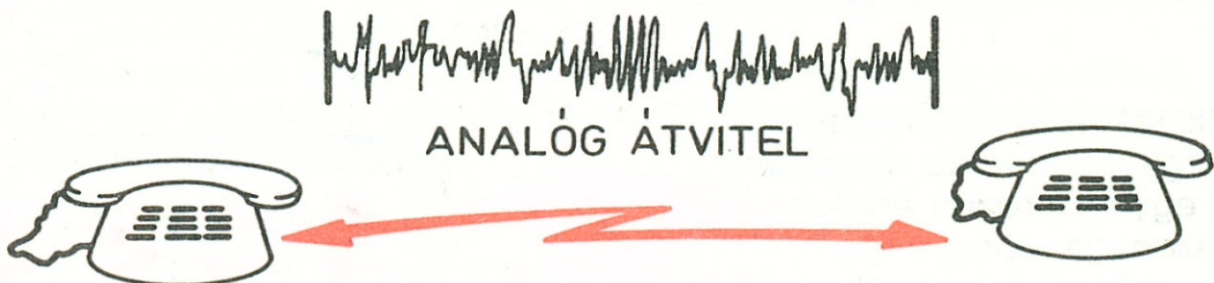


További példák: analóg és digitális tachométerek, vagy valamely vállalat forgalmának ábrázolása grafikongörbével (analóg), vagy táblázattal (digitálisan).



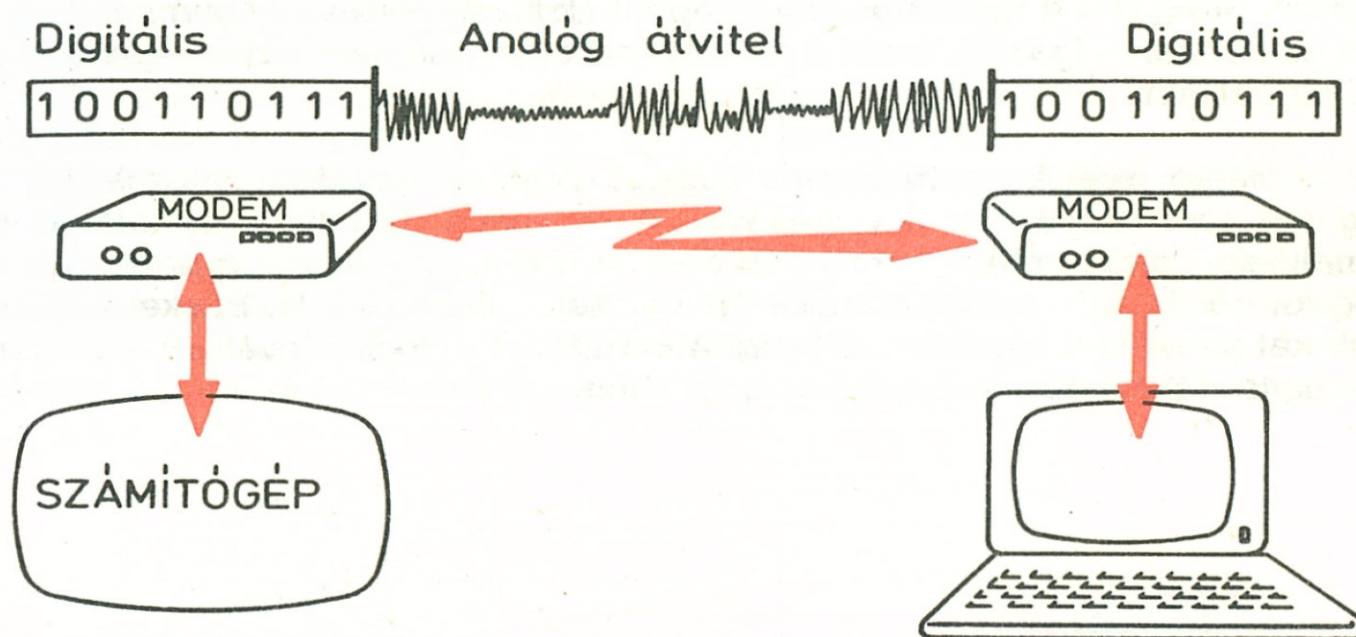
Az **információátvitel** vonatkozásában az analóg vagy digitális ábrázolásnak különös jelentősége van:

Ha a hagyományos távbeszélő-hálózaton egy telefonbeszélgetést közvetítünk, akkor az emberi beszéd hangjait (rezgéseit) analóg *elektromos rezgésekre* alakítjuk át. Ezeket az elektromos rezgéseket a *távközlési hálózaton* továbbítjuk, és a hívott fél készülékében ezek ismét hangrezgésekre, tehát emberi beszéddé alakulnak át. A hagyományos távbeszélő-hálózat tehát *analóg* átvitelre szolgáló hálózat.



A legtöbb számítógép azonban digitális információkkal dolgozik (egyesekkel és nullákkal). Ha az egyik számítógépből a másikba az analóg távközlőhálózaton akarunk adatokat átvinni, akkor a digitális adatokat előbb analóg információkra kell *átalakítanunk*.

Ezeket az átalakítást végző készülékeket **modemeknek** nevezzük (modulátor—demodulátor), és az adatfeldolgozó berendezések és az átvivővezeték közé kell kapcsolni.



Mivel a hívott félnek az adatokra rendszerint digitális formában van szüksége, ezért nála is kell hogy legyen egy modem, hogy az analóg adatok ismét átalakíthatók legyenek digitális adatokra.

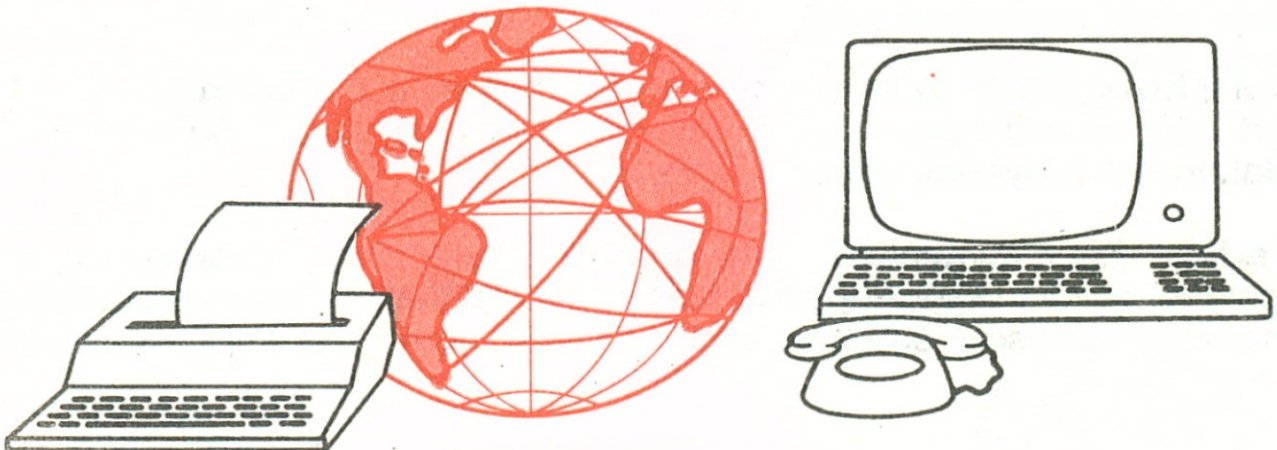
Az adatoknak a továbbítás miatti többszöri átalakítása nagyon körülményes. Ezenkívül gondot okoz még az is, hogy a távközlőhálózat az átvitelrel szemben támasztott minőségi és sebességi igényeket nem tudja teljesen kielégíteni.

Egyszerűbb, gyorsabb, olcsóbb és biztonságosabb a digitális információk **digitális átvitele**. Ezért létesítenek digitális hálózatokat, amelyek mindenképp előtt azokat a követelményeket elégítik ki, amelyeket az adatkommunikáció a hálózatokkal szemben támaszt.

Általánosan ismert példa a digitális összeköttetésre a **távíróhálózat**, ahol a távíró billentyűzetén legépelte betűket számjegyes kóddá (digitális) alakítva továbbítják.

Először **magán digitális vonalakat** építettek ki, amelyek azonban nem egységes kóddal dolgoztak, ezért a felhasználók gyakran nem tudtak egymással kommunikálni.

A probléma **nyilvános** hálózattal oldható meg, a szükségletnek megfelelő digitális távközlőhálózat létesítésével. A nyilvános digitális **telexhálózatot**, amely egyébként már a harmincas években alakult ki, és ma már az egész világot behálózta, a **digitális adathálózattal** egészítették ki. Ezeket a hálózatokat a jövőben egy **integrált digitális hálózattal** fogják leváltani, amelyen — digitalizálás után — a beszéd is továbbítható lesz.



Kapcsolt és bérelt vonalak

Kapcsolt vonalakon megvalósuló összeköttetés esetén a hálózatba bekapcsolt összes előfizető kapcsolóberendezéseken keresztül léphet egymással összeköttetésbe. Ha az előfizető egy összeköttetést akar létesíteni, akkor egy telefon tárcsájával vagy billentyűzetével felhívhatja a kívánt előfizetőt.

A kapcsolt hálózatra példák:

- nyilvános távbeszélő-hálózat (analóg) a telefon-, telefax- és teledataforgalomhoz,
- távíróhálózat (digitális) a telexforgalomhoz,
- adathálózat (digitális) a teletex- és adatforgalomhoz,
- széles sávú összeköttetés a képtelefon-forgalomhoz (ma még ritka).

Datex elnevezéssel a Német Szövetségi Posta egy különleges szolgáltatást nyújt az adatkommunikáció céljára. Egy terminálról az előfizető felhívja a kívánt számítógépet. A hívást könnyíthetik segédeszközökkel, pl. a rövid hívással vagy direkthívással, amikor a kapcsolatot egy hívóbillentyű lenyomására létrejön.



Még egyszerűbb, ha a számítógép saját maga végzi a hívást. Egy program vezérelte alatt felhívja a kívánt előfizetőt, megvárja annak jelentkezését, és megkezdí az adatközvetítést az előírányzott terminálhoz.

A kapcsolt hálózati összeköttetésekkel ellentétben bérelt vonalak esetén az előfizetők **állandó kapcsolatban** vannak. Ezt majdnem mindig csak adatátvitel céljából létesítik.

Példa:

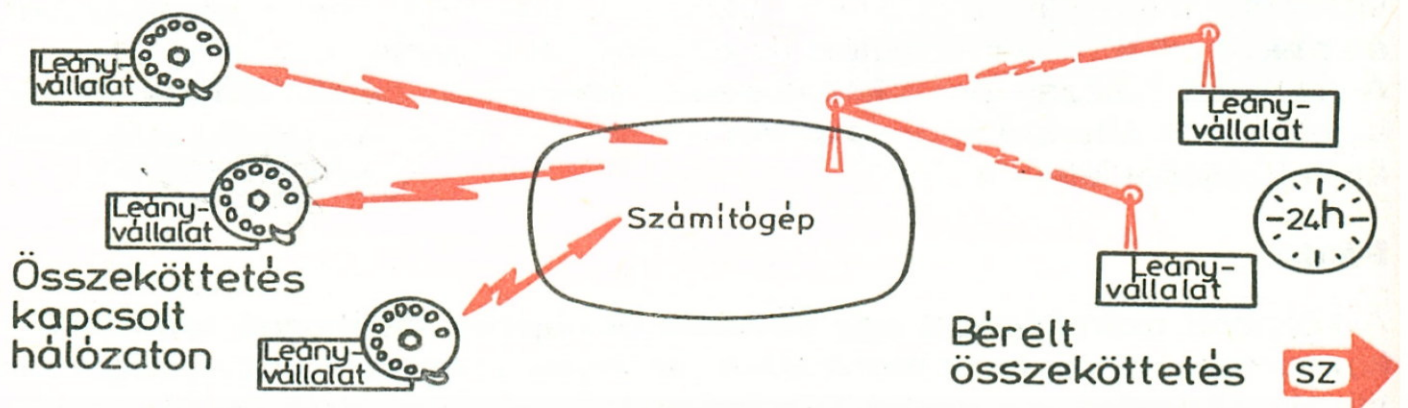
A központi számítógépet egy irodaház pincéjében létrehozott számítógépközpontban állítják fel, a terminálokat, amelyek az egyes irodahelyiségekben állnak, közvetlen vonalakkal kapcsolják össze a számítógéppel. A felhasználók bármikor dolgozhatnak a számítógéppel, mivel a számítógéppel való kapcsolat állandó.

Egy központi számítógéppel akkor is létesíthető **állandó összeköttetés**, ha a terminálok távolabb, pl. egy másik városrészben vagy egy másik városban vannak. Ekkor az összeköttetést az illetékes távközlési hatóságtól bérelt vonalak formájában kell megvalósítani. Ez kb. az az eset, mint amikor valakit egy másik városban felhívánk, és a hallgatót nem helyeznénk vissza, így bármikor beszélhetünk.

Az állandó összeköttetésnek az az előnye, hogy az kizárólag az előfizető rendelkezésére áll, míg kapcsolt összeköttetés esetén az előfizetőnek esetenként várnia kell (ha a felhívott állomás foglalt). Azonban a számítógépfelhasználóknak az adatátvitelhez gyakran több hívószámuk van, így a foglaltság miatti várakozási idők csökkenthetők.

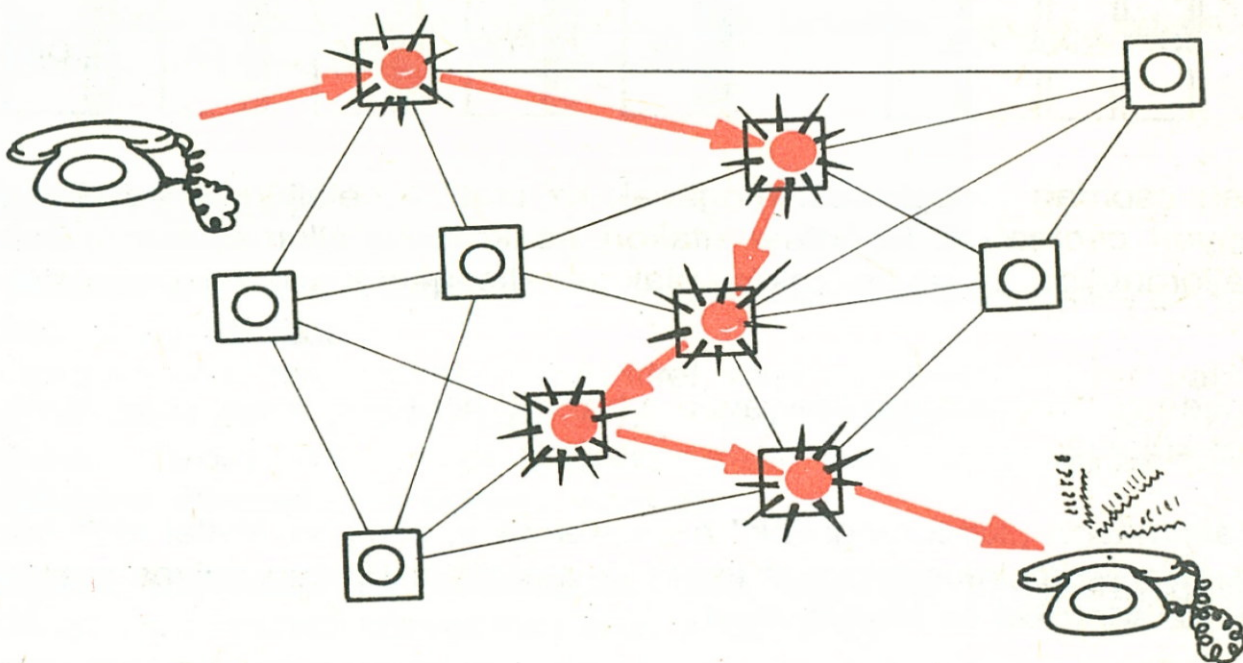
A bérelt nyilvános vezetéken megvalósított állandó összeköttetésért bérleti díjat kell fizetni, függetlenül attól, hogy ténylegesen milyen gyakran használták azt adatátvitelre. Azért kell fizetnünk, hogy a kapcsolat mindig „fennálljon”. Ezért az ilyen kapcsolat csak akkor gazdaságos, ha azt valóban ki is használják. Ezzel szemben a kapcsolt hálózati összeköttetések esetén a távbeszélő-hálózathoz hasonlóan külön hívásra jön létre a kapcsolat, és ezt hívásonként kell fizetni. Az adathálózatban az előfizetők egymástól való távolsága többnyire különösebben nem hat ki a költségekre. Sokkal fontosabb az összeköttetés időtartama és az átvitt információk mennyisége.

Annak eldöntése, hogy a bérelt és a kapcsolt összeköttetés közül melyik a gazdaságosabb, attól függ, hogy mennyi információt kell közvetíteni, és hogy mennyire fontos az azonnali átvitel lehetősége. Mivel a kapcsolt hálózatok rugalmasabbak és nagy előfizetői létszámot tesznek lehetővé, ezért a kapcsolt összeköttetés a gyakoribb.



Vonalkapcsolás és csomagkapcsolás

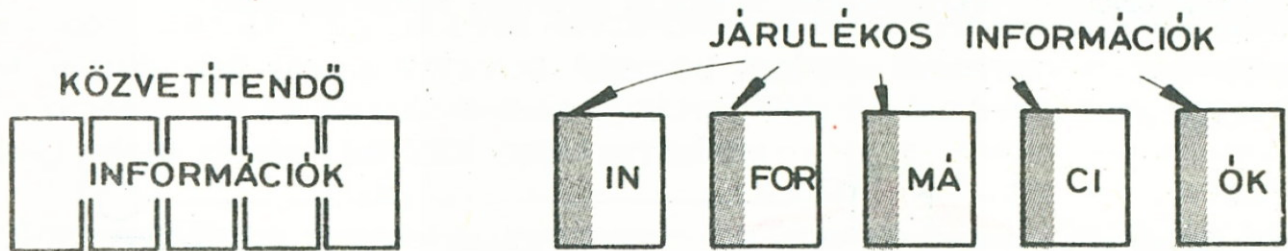
Vonalkapcsolt összeköttetés esetén az előfizetőket egy **fizikailag átkapcsolt** vezeték köti össze oly módon, mintha az előfizetőket megszakítás nélkül egy kábel kötné össze. Miután valamely terminálon a számhívás megtörtént, az ebből a készülékből kifutó vezeték a kapcsolóberendezések — a hálózat egymást követő csomópontjait összekötő szabad vezetékek igénybevételével — összeköti a hívott állomás termináljával.



Mihelyt ez az összeköttetés felépül, megkezdődhet a bevitt információk átvitele: ha az előfizető a telefonkagylóba belebeszél, hangja megszólal a hívott előfizető hallgatójában.

A vonalkapcsolt hálózatban az előfizetők közötti *kapcsolat fennmarad*, és más előfizető részére *nem használható* mindaddig, amíg azt a hívó és hívott előfizetők lefoglalják. Ezért előfordulhat pl., hogy a távbeszélő-hálózatban már a körzetszám hívásakor felhangzik a foglaltjelzés, nevezetesen akkor, ha minden intervonal foglalt. Helyi beszélgetés azonban többnyire ekkor is lehetséges, mert a városi vezetékhalózati tényegesen nagyobb átbocsátóképességű, mint az interhálózat.

A vonalkapcsolásos átvitel ellentétben a **csomagkapcsolás** esetén az információk nem egy kapcsolt, csak egy előfizető részére fenntartott vezetékben futnak a feladótól a címzettig, hanem a következőkben ismertetett módon. Az előfizető által bevitt adatokat kisebb részekre, ún. **csomagokra** tagolják, és minden csomagot ellátnak megfelelő kiegészítő információkkal (adatok hossza, típusa, sorrendje, hívott azonosítója stb.). Az egyes „csomagokat” egyenként továbbítják a hívott előfizetőhöz.

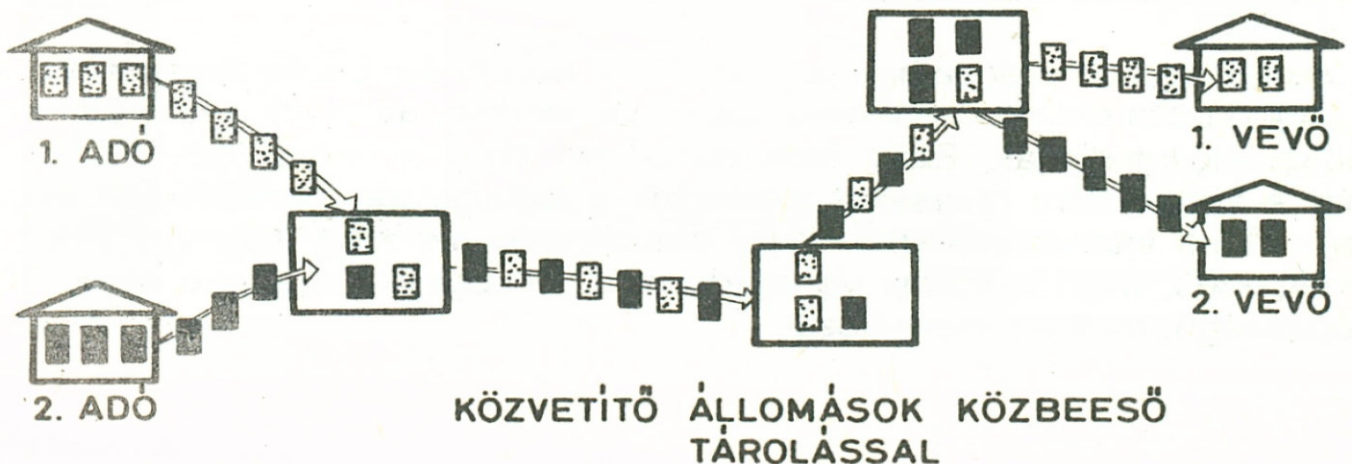


Minden csomag „megkeresi magának az utat” a célállomáshoz. Eközben az egyes csomagok különféle utakon és közvetítőállomásokon juthatnak a célállomáshoz. Ez az összekötő utak terheltségének mindenkorai mértékétől függ.

Az egyes csomagok címzése révén a hálózat pontosan tudja, hogy hová kell azokat küldeni.

A célállomáson „kicsomagolják” a csomagokat, azaz az átvitel előtt hozzáfűzött kiegészítő információkat ismét leválasztják. Ezután helyes sorrendbe teszik az adatokat és kiadják őket.

Mivel előfordulhat, hogy egyes vonalak átmenetileg túl vannak terhelve, vagy hogy a címzett nem fogadóképes, azért a közvetítőállomásoknak **tárolni** kell tudniuk a beérkező csomagokat, és a zavar elhárítása után azokat továbbítaniuk kell. Ezért nevezik a csomagkapcsolást tárolt átvitelnek.



Csomagkapcsolás* esetén az átviteli hálózat — a vonalkapcsolásos átvittel szemben — gazdaságosabban használható ki, mert

- az egyes részvonalakat sok előfizető egyidejűleg használhatja, mivel nincs átmenő kapcsolat a feladótól a címzettig, ami a vezetéket más előfizetők előtt elzárná;
- az információcsomagok (az ún. protokollok által) szabványosítottak, mégpedig úgy, hogy különböző adatfeldolgozó rendszerek is „meg tudják egymást érteni”;
- ellenőrzési mechanizmusok által sokkal nagyobb átviteli biztonság érhető el;
- az átviteli hálózat egyenletesebben van terhelve, mert az egyes információk különféle utakon továbbíthatók.

Üzem módok

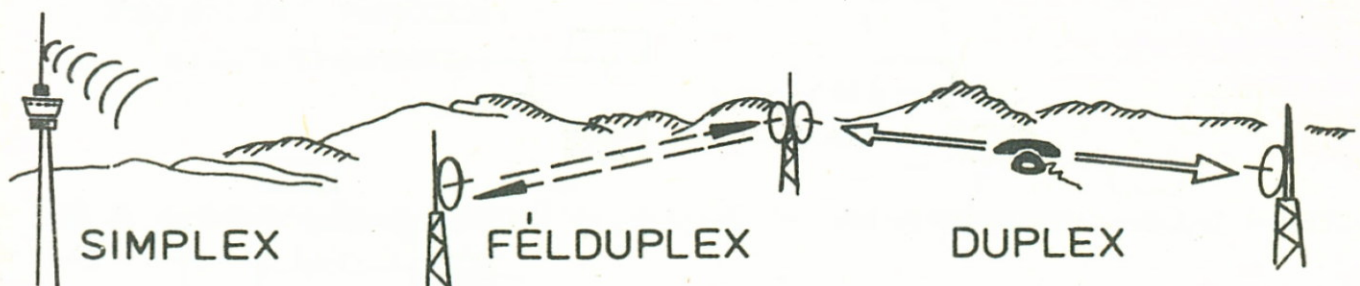
Az említett átviteli módokon kívül egy további jellemző is van az összeköttetésre vonatkozóan.

A kapcsolatokat technikailag kábelekkel vagy rádiócsatornákkal valósítják meg, amelyeknek üzemeltetése a használatától függően különböző lehet. Általában három üzemmódot különböztetünk meg: a **szimplex üzemet**, a **félduplex üzemet** és a **duplex üzemet**.

Szimplex üzemben az információk csak egyetlen irányban küldhetők. Példa erre a tv-műsor közvetítése vagy a képűjság.

Félduplex üzemben az információk felváltva az egyik vagy másik irányba közvetíthetők. Példa a telexforgalom vagy a teledata.

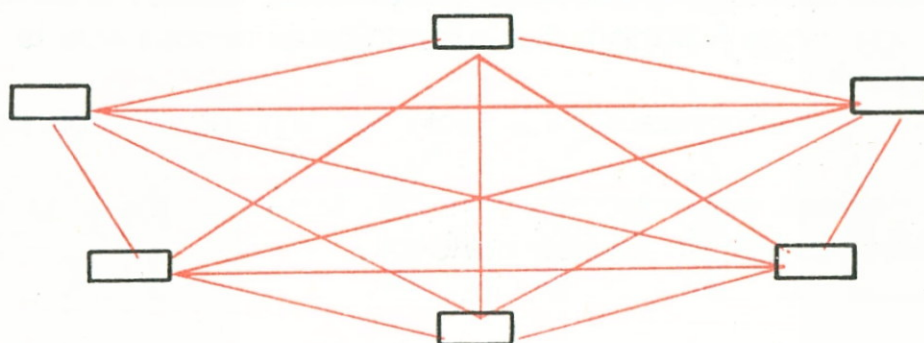
A **duplex üzem** egyidejűleg kétirányú információtovábbítást tesz lehetővé. A távbeszélő-hálózat pl. duplex üzemben dolgozik.



* A Német Szövetségi Köztársaságban az adatátvitelre szolgáló nyilvános hálózatok (datex) elnevezése: datex-L (vonalkapcsolt hálózat) és datex-P (csomagkapcsolt hálózat).

Hálózattopológiák

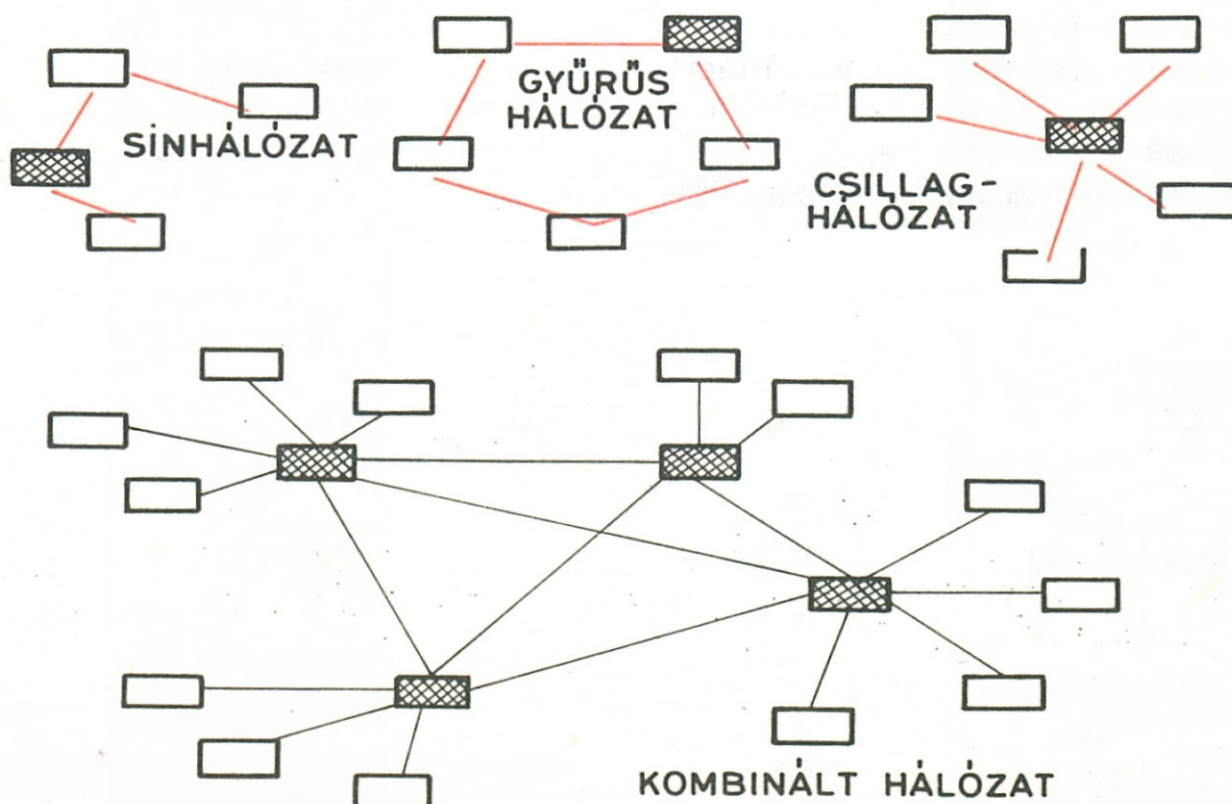
Egy átviteli hálózat az összeköttetések szempontjából akkor optimális, ha az *minden előfizetőt minden más előfizetővel összeköt.*



Egy ilyen topológiájú összekötő hálózatot **teljes csatlakozású** (vagy hálós) hálózatnak nevezünk, amely azonban a szükséges összekötő vonalak nagy száma miatt csak részlegesen valósítható meg.

A gyakorlatban ezzel szemben egyszerűbb hálózattopológiák és ezek kombinációi terjedtek el.

Példa:



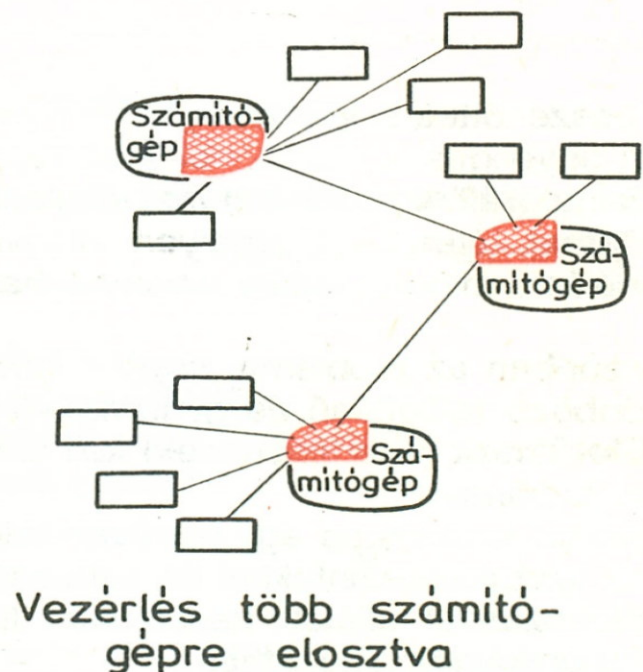
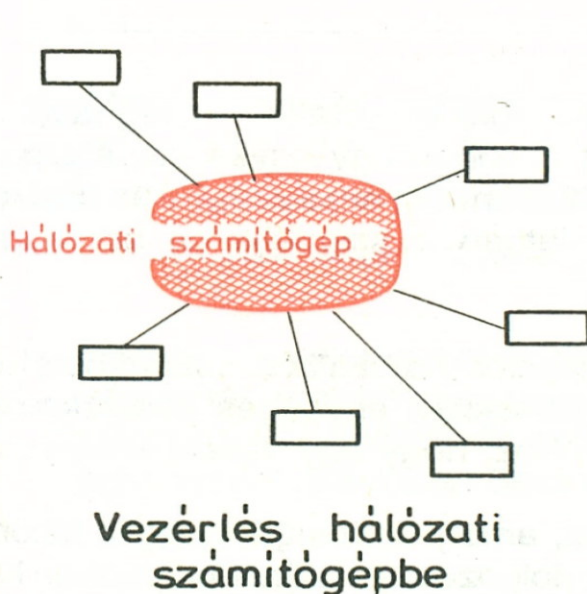
A hálózat topológiájának kiválasztását a következő tényezők befolyásolják: az előfizetők száma és területi elhelyezkedése, az összeköttetés gyakorisága, a rendszer üzembiztonsága, a hálózat bővíthetősége stb.

A gyűrűs vagy a teljes csatlakozású hálózat pl. messzemenően független a vezetékmeghihibásodásoktól, mert ilyen esetben az átvitel kerülőúton is megvalósítható.

Ebben az összefüggésben az a kérdés is érdekes, hogy hol történjék a teljes átviteli hálózat **vezérlése** és **felügyelete**, amit a **korszerű digitális hálózatok** esetén rendszerint egy számítógép végez.

A vezérlés és felügyelet feladatai: kapcsolatok létesítése, információk átvitele, információk átmeneti tárolása, helyességük és teljességük ellenőrzése. Ha csonka információk érkeznek, akkor azok a feladónál újra lehívhatók.

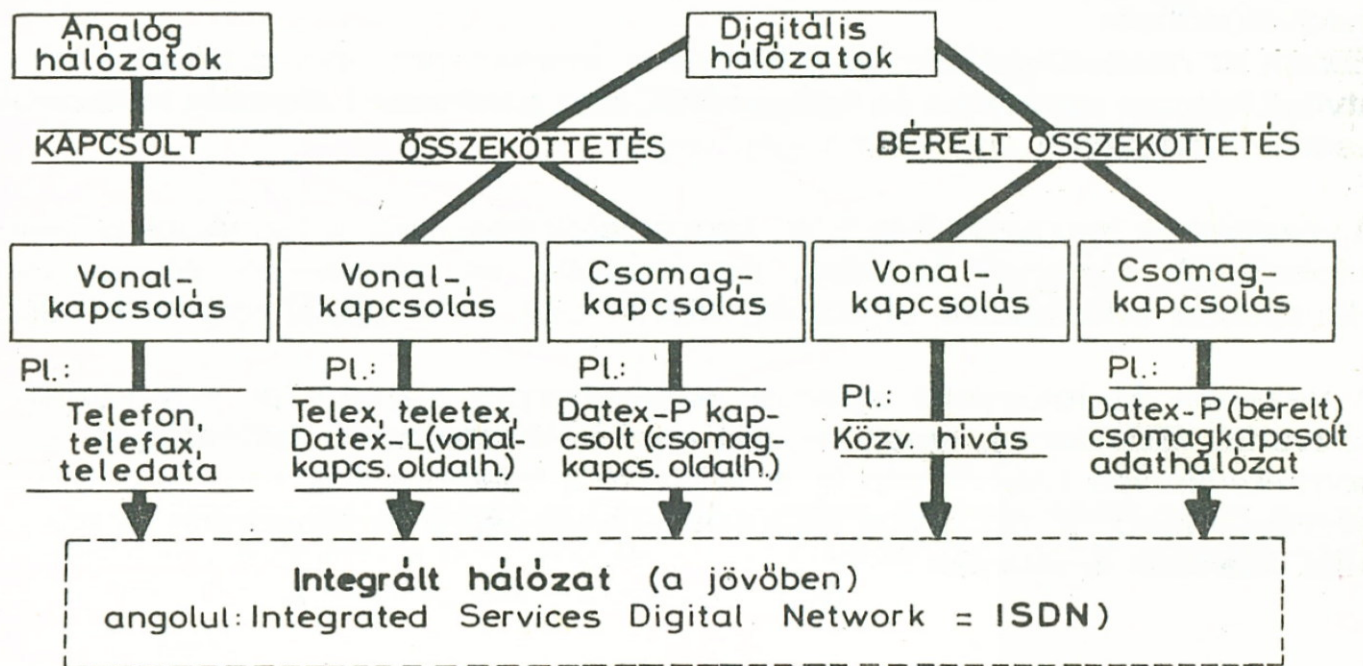
A vezérlés és felügyelet egyetlen számítógépre bízható (pl. egy központi közvetítő állomáson), amely ezután mint **hálózatvezérlő számítógép** vagy **kommunikációs számítógép** kizárólag erre a feladatra illetékes. Ez a feladat azonban több, a hálózatra kapcsolt számítógépre is elosztható, amelyek más feladatot is ellátnak.



Magát a számítógépet, annak egységeit és működésének módját később még részletesen leírjuk.

Áttekintés az összeköttetési lehetőségekről

A személyi és adatkommunikációhoz ma a nyilvános hálózatok és szolgáltatások egész sora használható, amelyek részben az egész világra kiterjednek.



Az összeköttetési lehetőségek — mint azt az ábra is mutatja — még távolról sem egységesek. Az utóbbi évek fejlődése azonban egy egységes, szolgáltatásintegrált **digitális hálózat** kifejlesztése felé mutat, amely **digitális összeköttetést** létesít, és amelyen világviszonylatban a személyi és adatkommunikáció minden fajtája lebonyolítható.

Itt azonban az probléma, hogy a kommunikációs hálózathoz csatlakoztatott különböző felépítésű és gyártmányú készülékeknek egységes **csatlakozófelületüknek** (interfészüknek) kell lennie ahhoz, hogy egymással kommunikálni tudjanak.

Egyrészt szükséges egy **hardver-interfész**, amely biztosítja, hogy a különböző feszültségszintekkel és sebességgel dolgozó és különféle mechanikai csatlakozással ellátott készülékek fizikailag csatlakoztathatók legyenek a kommunikációs hálózathoz.

Másrészt egy ún. **szoftver-interfészt** is egyértelműen definiálni kell, pl. annak megadására, hogy az információkat milyen kódban kell kifejezni, vagy az adatok milyen sorrendben kerüljenek átvitelre. Ezt ún. protollok szabályozzák.



1. Milyen nyilvános átviteli hálózatokon továbbítanak információkat analóg módon?
.....
2. Milyen eszközök használhatók átviteleknél analóg információk digitális információkká való átalakítására?
.....
3. Jelölje meg, hogy analóg információkat mikor kell digitális információkká átalakítani!
 - Ha egy olyan készülék fogadja azokat, amely csak a digitális információkat „érti”.
 - Ha az átvitelre csak digitális hálózatok állnak rendelkezésre.
4. Mely hálózatok esetén lehet a kapcsolatot egy másik előfizető felhívásával létrehozni?

Távbeszélő-hálózat. <input type="radio"/>	Közvetlen hívó hálózat. <input type="radio"/>
Távíróhálózat (telex). <input type="radio"/>	Datex-L adathálózat. <input type="radio"/>
5. Hogyan nevezzük azt az összeköttetést, amelynél végponti készülékek egymás között, végponti készülékek és számítógépek vagy számítógépek egymással állandóan össze vannak kötve?
.....
6. Az átvitel melyik fajtája esetén marad a vezeték a kapcsolat tartamára lefoglalva úgy, hogy más előfizető a vezetéket nem használhatja?
.....



Válaszok

1. Információkat analóg módon csak a távbeszélő-hálózaton továbbítanak.
2. **Modemekkel** lehet analóg információkat átviteleknél digitális információkká átalakítani.
3. Analóg információkat akkor kell digitális információkká átalakítani:
 - Ha a vevőkészülék csak digitális információkat tud „megérteni”.
 - Ha az átvitelre csak digitális hálózatok állnak rendelkezésre.
4. Az alábbi hálózatok esetén lehet kapcsolatot létrehozni egy másik előfizető felhívásával:
 - Távbeszélő-hálózat.
 - Távíróhálózat (telex). Datex-L adathálózat.
5. **Bérelt összeköttetésnek** nevezzük azt a kapcsolatfajtát, amelynél a végponti készülékek egymással, végponti készülékek és számítógépek vagy két számítógép egymással állandóan össze vannak kötve.
6. A **vonalkapcsolt átvitel** esetén marad a vezeték a kapcsolat teljes tartamára lefoglalva úgy, hogy más előfizető a vezetéket nem használhatja.



7. Az alábbi megállapítások melyike igaz a csomagkapcsolt átvitelre?

A csomagkapcsolt átvitelt vonalkapcsolt átvitelnek is nevezik.

Az előfizető által bevitt információkat kisebb egységekké tördeljük és azokat egyenként továbbítjuk.

Minden egyes információcsomagot adatok hosszára, sorrendjére, a hívott és a hívó előfizetőre vonatkozó kiegészítő információkkal látunk el.

Egy vezetéken különböző feladók információcsomagjai továbbíthatók.

A csomagkapcsolt átvitel esetén az átviteli hálózatot gazdaságosabban használjuk ki, mint a vonalkapcsolt átvitel esetén.

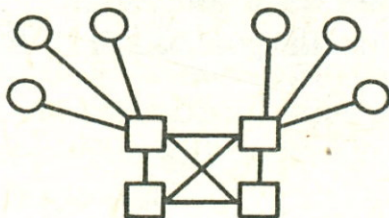
8. Hogyan nevezzük azt az összeköttetési típust, amelynél az információk csak egy irányban küldhetők?

.....

9. A jövőben valamennyi átviteli hálózatot egyetlen nagy hálózattá kívánjuk egyesíteni, amelyen a személyi és adatkommunikáció minden fajtája lebonyolítható. Milyen néven ismeretes ez a hálózat?

.....

10. a) Mely hálózattopológia látható a rajzon?



.....

.....

b) Kitűnik-e ebből az ábrából, hogy az összeköttetés analóg vagy digitális összeköttetés-e?

Igen. Nem.

Kapcsolt vagy bérelt összeköttetésről van-e szó?

Igen. Nem.



Válaszok

7. A csomagkapcsolt átvitelre az alábbi megállapítások igazak:

Az előfizető által bevitt információkat kisebb egységekké tördeljük és azokat egyenként továbbítjuk.

Minden egyes információcsomagot pl. az adatok hosszára, sorrendjére, a hívott és a hívó előfizetőre vonatkozó kiegészítő információkkal látunk el.

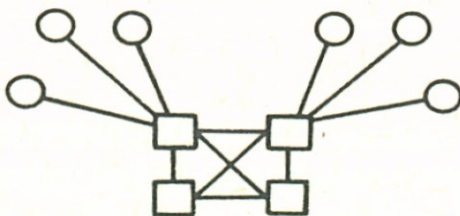
Egy vezetéken különböző feladók információcsomagjai továbbíthatók.

A csomagkapcsolt átvitel esetén az átviteli hálózatot gazdaságosabban használjuk ki, mint a vonalkapcsolt átvitel esetén.

8. **Szimplex üzemnek** nevezzük azt az összeköttetési típust, amelynél az információk csak egy irányban küldhetők.

9. **Szolgáltatásintegrált** digitális hálózatnak (ISDN) nevezzük azt az egységes, világszerte használt hálózatot, amelyen a személyi és adatkommunikáció minden fajtája lebonyolítható.

10. a)



Ez a hálózat

egy **teljes csatlakozású**
és két **csillaghálózatból** áll.

b)



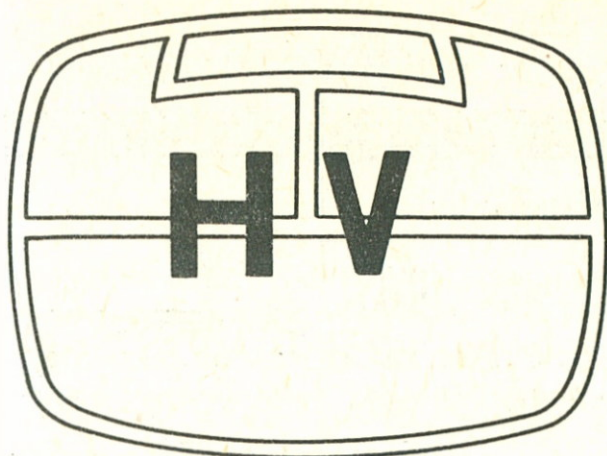
Nem.

Ebből az ábrából

csak a hálózat topológiája ismerhető fel,
az összeköttetés típusa nem.



Nem.



Adatfeldolgozás
és kommunikáció

Számítógépek
és alkalmazásuk

A számítógépek
funkcionális
egységei

Ebben a fejezetben megtudja,
hogyan osztályozzuk
a számítógépeket, valamint
hogyan a különböző kategóriájú
és kiépítettségű számítógépek
mire képesek.

Ezenkívül megtanulja,
hogyan milyen lehetőségek vannak
a számítógépek vagy
részegységeik összekapcsolására,
és hogyan egy számítógép-hálózat
milyen előnyöket nyújt
a felhasználók számára.

Hogyan működik
a számítógép?

**A számítógépek osztályozása
és teljesítményviszonyai**

A számítógépek üzemmódjai

A számítógépek osztályozása és teljesítményviszonyai

Az első fejezetből láthattuk, hogy napjaink kommunikációja a számítógéptechnika nélkül már elképzelhetetlen. Minden ember hasznát látja ennek, jóllehet nem is mindig tud róla. A számítástechnika és a kommunikációtechnika egyre jobban közeledik egymáshoz.

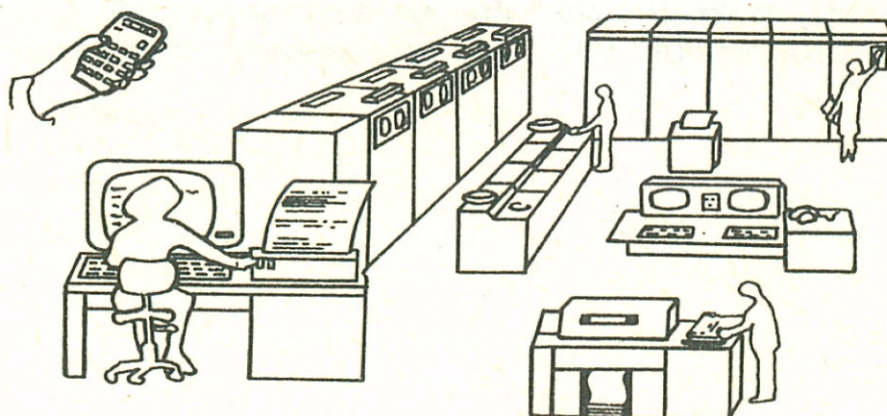
Miután az első fejezetben kifejtettük, hogy a hagyományos kommunikációs eszközök és az átviteli vonalak teljesítménye a számítástechnika alkalmazásával hogyan növelhető, most és könyvünk további fejezeteiben magát a számítógépet kívánjuk részletesen leírni.

A számítógépek osztályozása méretük és teljesítményük szerint

Különböző nagyságú és kiépítésű számítógépek vannak. A széles választék az egyszerű mikroszámítógépektől a nagyszámítógépekig terjed.

A számítógépek következőkben bemutatott fogalmi osztályozása főként teljesítőképességük szerint terjedt el, amit azonban nem használnak egységesen. Eszerint megkülönböztetünk:

**mikroszámítógépet,
miniszámítógépet és
nagyszámítógépet.**



Mikroszámítógépek

A **mikroszámítógép** a számítógépek között a „törpe”. A mikroszámítógép méretei lehetnek esetenként csak néhány centiméteresek, így az még egy karórában is elfér.

Megfelelő kiépítésben egy teljes értékű számítógép teljesítménye természetesen kisebb, mint a következő méretosztálybeli „nagy testvéreié”.

A mikroszámítógépek gyártását a mikroelektronika fejlődése tette lehetővé. Az integrált áramköri technikával a számítógépek építőelemeinek (tranzisztorok, diódák, ellenállások stb.) méretét oly mértékben lehetett csökkenteni, hogy pl. egy négyzetcentiméternél is kisebb felületű szilíciumlemezre manapság már több százezer elem helyezhető el.

Egy ilyen kis szilíciumlapka a számítógépnek már mindazon részeit tartalmazhatja, amelyek a műveletvégrehajtáshoz szükségesek, és így az képes a bevitt adatok és tárolt programok feldolgozására.

Azonban, ha kívülről akarunk adatokat vagy programokat bevinni és látni szeretnénk a feldolgozás eredményét, akkor megfelelő kiegészítő be- és kiviteli egységekre van szükségünk. A beviteli egység lehet pl. egy billentyűzet, a kiviteli egység pedig pl. egy folyékony kristályos kijelző vagy nyomtató.

Ebből már látható, hogy egy számítógép méretét, amelynek feldolgozóegysége egy kisméretű szilíciumlapkán megvalósítható, lényegében a be- és kiviteli egységek és emellett természetesen az áramellátás módja, pl. az elemek nagysága határozza meg.

A szilíciumlapkák előállításának módját, a rajta levő kapcsolások működését könyvünk másik fejezetében tárgyaljuk részletesebben.

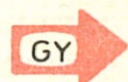
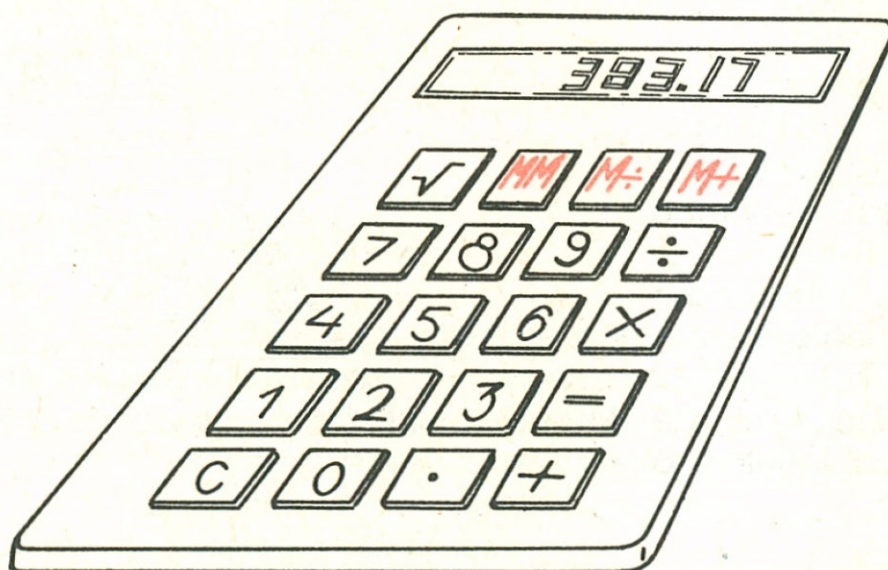
Vegyük először szemügyre a legkisebb teljesítményű számítógépeket. Ezek a **zseb-** vagy **asztali számoló- és számítógépek**, amelyeknek többnyire beépített programjuk van, és amelyeknek ezért csak korlátozott a teljesítményük.

Fix programú zseb- és asztali számológépek

A **zsebszámológépekbe kis táruk** vannak beépítve, amelyek a számológép funkcióinak végrehajtásához szükséges programokat tartalmazzák. Ezeket a programokat a gép gyártója véglegesen írja be a tárba, azok nem törölhetők, nem változtathatók. Külső táruk rendszerint nincsenek.

A változó adatokat kis kapacitású **adattárakban** tárolják. Kikapcsoláskor az adatok törlődnek, hacsak a gépben nincs „nem törlődő” tár, mint ahogyan ez sok gyors regiszteres zsebszámológép esetében előfordul. Miként a nagyszámítógépeknek, úgy a zsebszámológépeknek is van egy **központi egysége**, amelyben az adatok a program szerint feldolgozásra kerülnek.

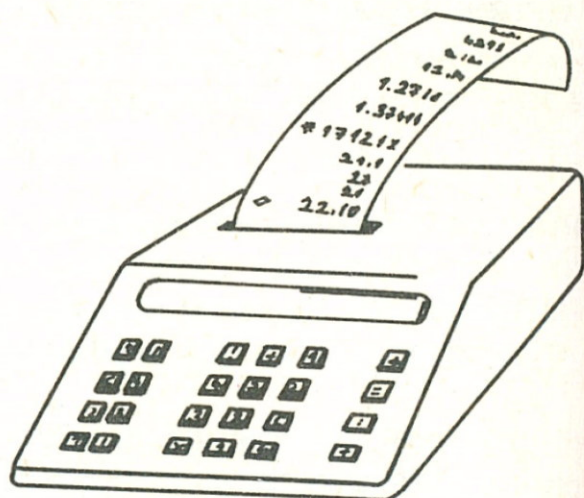
Ennek teljesítménye azonban sokkal kisebb, mint a nagyszámítógépeké.



A zsebszámológépek a négy számtani alpművelet mellett (összeadás, kivonás, szorzás, osztás) típustól függően más funkciókat is végezhetnek. Ezek speciális tudományos vagy kereskedelmi feladatok lehetnek (szinusz-, négyzetgyök-, százalékszámítás stb.), vagy olyan szolgáltatások, amelyek a használnak a legkülönbözőbb információkat adják. Például idő és naptári nap kijelzése, stopperóra, ébresztőóra, jegyzettár, határidőnaplár és sok minden más is.

A bevitt a **billentyűzettel** végezzük. Nemcsak a feldolgozandó adatokat (pl. az összeadandó számokat), hanem a programokat is bevisszük*, pl. a + billentyű lenyomásával az „összeadás” programját. Egy kijelzőmezőben megjelennek a bevitt számok és a számítás eredménye is, tehát a kijelzőmező ábrázolja mind a bemenő, mind pedig a kimenő adatokat.

A zsebszámológép és asztali számológép közé éles határ nem húzható. Az asztali számológép nagyobb, és a funkciók és adatok bevételére is több a billentyű. Az asztali számológépek esetében a bemenő és kimenő adatok rendszerint mind a kijelzőmezőben, mind pedig papírszalagon is megjeleníthetők.



Egyes zsebszámológépek átalakíthatók asztali számológéppé, ha egy papírszalag-nyomtatóval látják el őket.

* Valójában nem a programok bevételére, hanem a műveleti jelek bevételére kerül sor. (Lektor.)

Programozható zseb- és asztali számológépek

Ha egy zseb- vagy asztali számológép tárolt programja nem változtatható, akkor a teljesítménye adott, és azzal kell megelégednünk.

Ha pl. egy műveletsorozatot többször egymás után végre akarunk hajtani, akkor a megfelelő funkcióbillentyűket ugyanabban a sorrendben ismételten le kell nyomnunk.

Vannak azonban **programozható** zseb- és asztali számológépek is, amelyekben, ha a funkciók egy sorozatát tároljuk, akkor ezeket a funkciókat a később bevitt adatokkal akárhányszor megismételhetjük anélkül, hogy az ezeknek megfelelő billentyűket újra és újra le kellene nyomnunk.

Példa:

Fahrenheit- (angol—amerikai, nem SI rendszer) fokban kifejezett hőmérséklet-adatokat Celsius-fok értékre akarunk átalakítani. Az átszámítást a következő összefüggés szerint kell végezni:

Hőmérséklet Fahrenheit-fokban:

mínusz 32
osztva 9-cel
szorozva 5-tel
= a hőmérséklet Celsius-fokban.

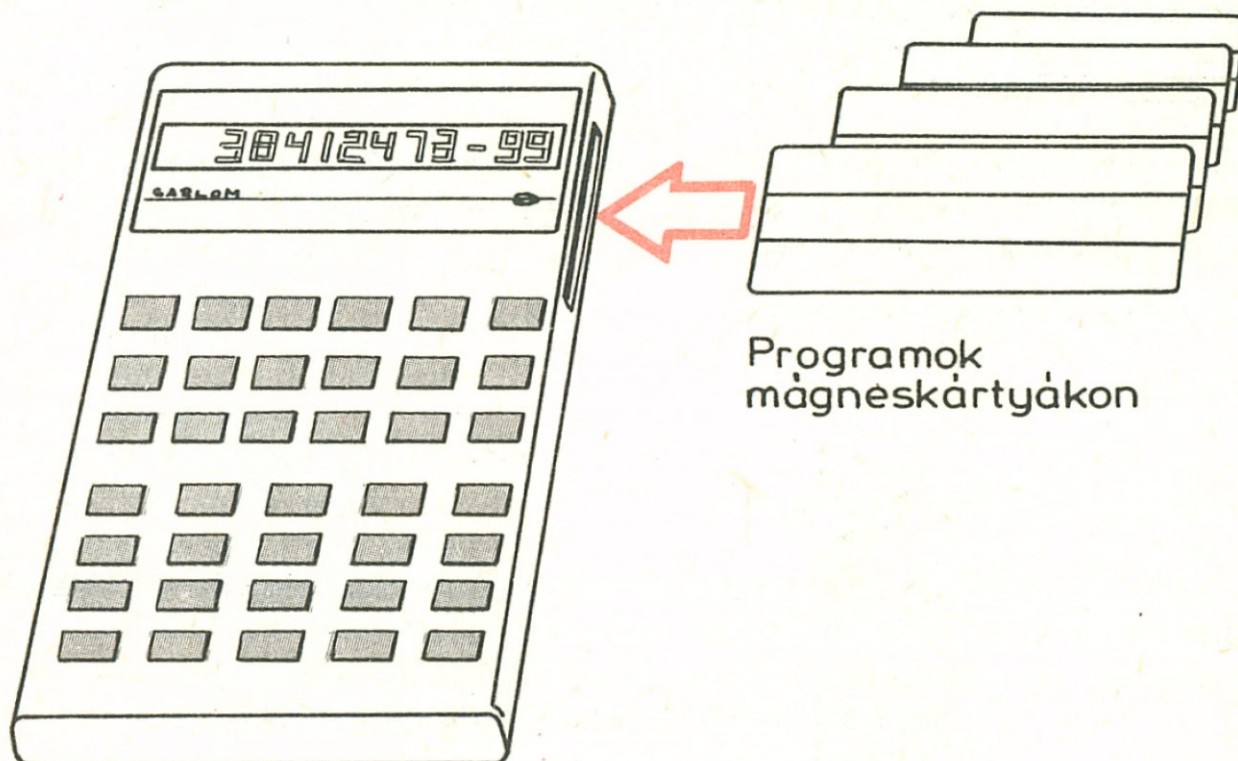
$$\left(C = \frac{F - 32}{9} \times 5 \right)$$

Egy nem programozható számológépben minden egyes hőmérséklet-átszámításkor a következő beviteli műveleteket kellene elvégezni:

1. lépés:	Fahrenheit-fok értékét bevinni.
2. lépés:	– billentyűt lenyomni.
3. lépés:	32-t bevinni.
4. lépés:	÷ billentyűt lenyomni.
5. lépés:	9-et bevinni.
6. lépés:	× billentyűt lenyomni.
7. lépés:	5-öt bevinni.
8. lépés:	= billentyűt lenyomni.

Egy programozott zsebszámológép esetében ezeket a lépéseket csak egyetlen egyszer kell megtenni. Ezek után ezek a beviteli lépések programként tárolódnak. Most már csak egy tetszőleges Fahrenheit-értéket kell begépelni, és a megfelelő funkcióbillentyűt lenyomni. A számológép ezzel az értékkel a 2—8. feldolgozási lépéseket automatikusan elvégzi, és az eredmény megjelenik a kijelzőmezőben.

A **programozható zseb- és asztali számológépek** a programokat, amelyeket a gép használója készített vagy speciális tárukban őrzik meg, amelyek a gép kikapcsolásakor sem törlődnek, vagy pedig a felhasználó a programokat kis mágneskártyákra írja, és ha újra használni akarja, a számológépbe visszatölti. Ilyen esetben, amikor a mágneskártyán levő programok futtatására alkalmas a zsebszámológép, a kereskedelemben kapható programok is használhatók hozzá.

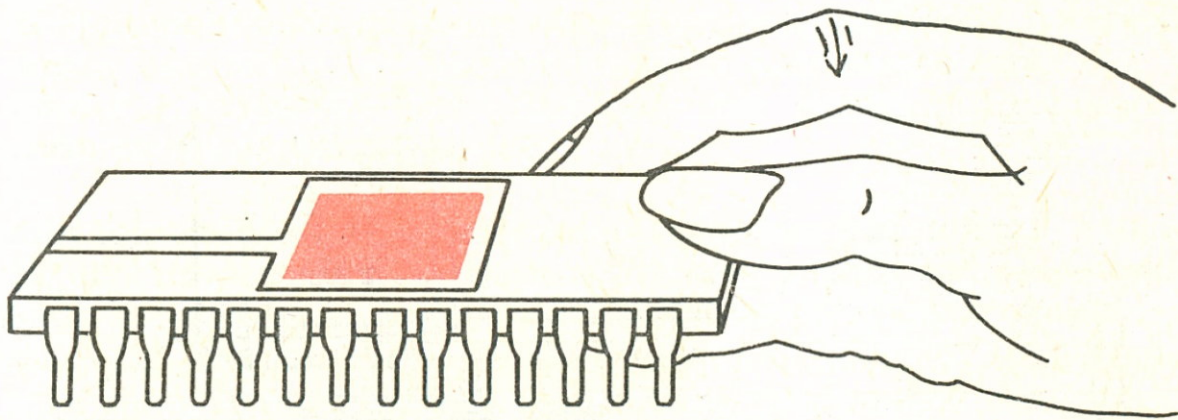


Szabadon programozható mikroszámítógépek

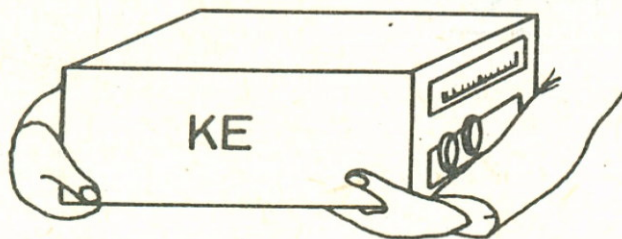
A legtöbb szolgáltatást azok a mikroszámítógépek adják, amelyek valamely programnyelven **szabadon programozhatók**. A szabadon programozható mikroszámítógépekkel nemcsak tárolt programok sorozata állítható össze, hanem személyes igényeket kielégítő programok is kifejleszthetők.

Ezekkel a mikroszámítógépekkel tehát az adatfeldolgozás minden fajtája elvégezhető. A tulajdonképpeni számítások mellett a következőket végezheti: szövegek tárolását, megváltoztatását, újraserkesztését és kinyomtatását — statisztikák készítését és kiértékelését, grafikonok rajzolását és még sok minden mást. A **szabadon programozható mikroszámítógép** teljes értékű számítógép, csak éppen kisebb méretű és teljesítményű, mint egy nagyszámítógép.

A központi egységnek az adatok feldolgozását végző részeit többnyire egyetlen félvezető építőelemben: a **mikroprocesszorban** egyesítik.



A központi egység többi részét többnyire néhány félvezető építőelemen valósítják meg úgy, hogy az áramellátást is magában foglaló teljes központi egység egy táskairógép nagyságú.



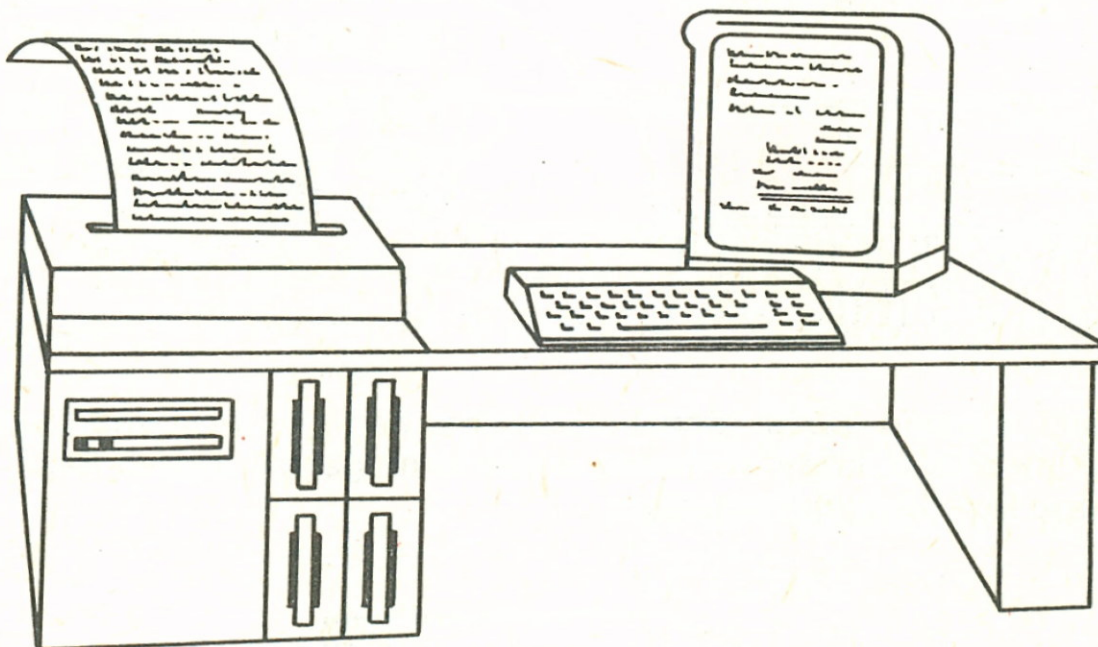
A központi egység felépítését és az adatfeldolgozásban játszott szerepét könyvünk egy későbbi fejezetében részletesen leírjuk.

A központi egységet rendszerint egy képernyős terminállal, egy háttértárral (mint pl. hajlékony lemezes tár vagy mágneskazettás egység) és egy nyomtatóval látják el.

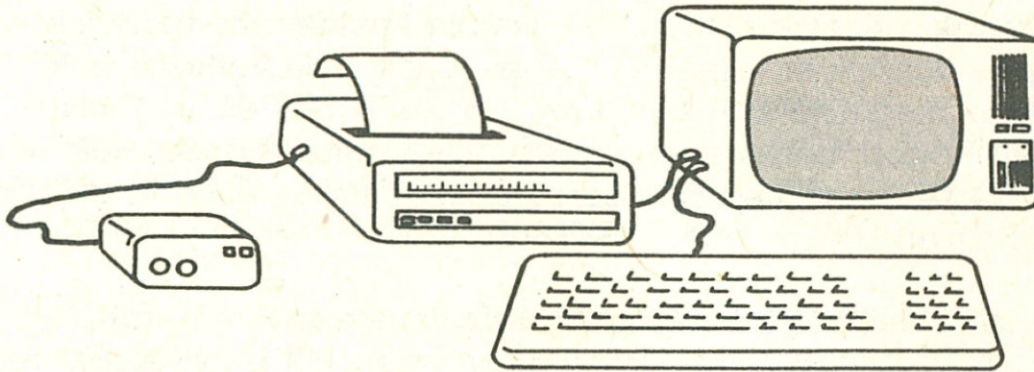
A gazdasági életben a szabadon programozható mikroszámítógépeket olyan kiépítésben szállítják, hogy az alkalmas legyen pl. kis- és középüzemek bérelszámolására, laboratóriumban végzett tudományos kísérletek ellenőrzésére stb.

A szabadon programozható számítógépek másik alkalmazási területe a szövegfeldolgozás és a szövegszerkesztés.

A **szövegautomatának** vagy **szövegrendszernek** nevezett ilyen mikroszámítógépek programjait rendszerint egy háttértáron (pl. hajlékony mágneslemezen) tárolják, amelyet a gyártó cég a géppel együtt szállít.

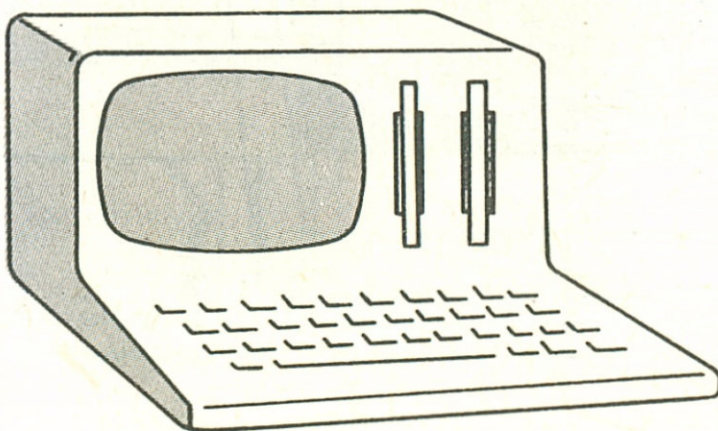


Olcsó perifériákkal, pl. egyszerű billentyűzettel és egyszerű nyomtatóval, valamint a normál tv-készülékhez való csatlakoztatási lehetőséggel a szabadon programozható mikroszámítógépek **személyi számítógépként** használhatók*. Ezeket gyakran otthoni, házi, háztartási, családi, asztali vagy hobbi-számítógépnek is nevezik.



Olyan terminálok — főként képernyős terminálok —, amelyekben van mikroprocesszor is, az adatfeldolgozás egy részét már maguk is elvégezhetik és részben függetleníthetik magukat attól a nagyszámítógéptől, amellyel a terminál össze van kötve. Ez tehermentesíti a számítógép-rendszert. Ezeket a terminálokat, amelyek tulajdonképpen már önálló mikroszámítógépek, **intelligens termináloknak** nevezzük.

Intelligens terminálokat pl. a programfejlesztéshez használnak, ahol a programozók egy programot a terminálon „off-line”, tehát a nagyszámítógéptől függetlenül megírhatnak, és az utasításokat formális hibák vonatkozásában (írás hibák és programnyelv szabályainak megsértése) ellenőriztethetik. Ezek után már az ún. szintaktikailag hibátlan programot viszik át a nagyszámítógépre, ahol a fordításra és a logikai helyességtesztre kerül sor.

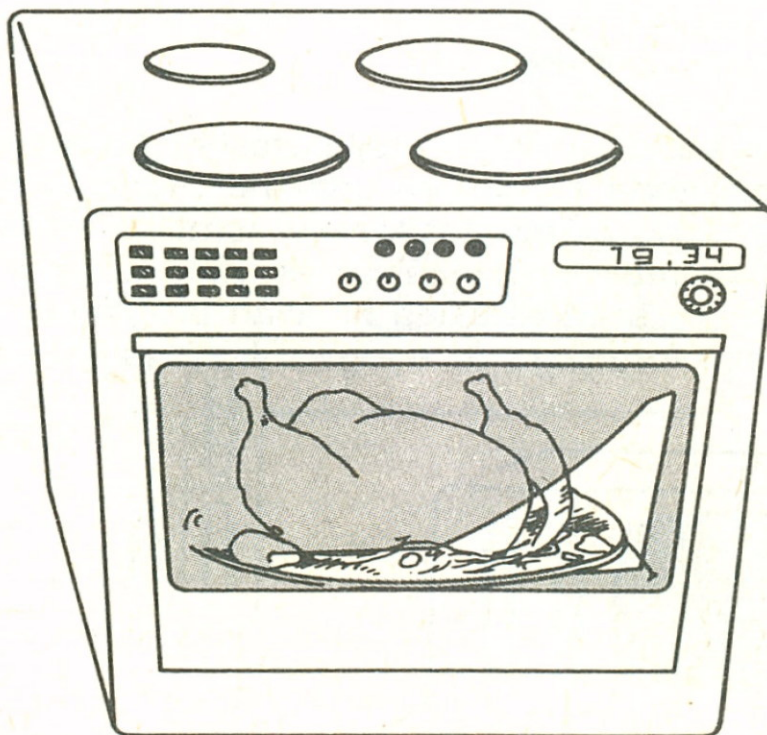


* Napjainkban a hobbi célú, olcsóbb kivitelű személyi számítógépek mellett egyre nagyobb jelentőségre tesznek szert az ún. professzionális személyi számítógépek, amelyek nagy teljesítményű perifériákkal felszerelve üzleti, gazdasági, pénzügyi vagy műszaki feladatok széles körének megoldására alkalmasak. (Lektor.)

A mikroszámítógépet az említett alkalmazási esetek mellett a mosógéptől a lökhajtásos repülőgépekig a legkülönbözőbb készülékek vezérlésére használják. Az ilyen mikroszámítógépeknek rendszerint fix, változtathatatlan programjuk van, mivel ismételten ugyanazokat a műveleteket kell vezérelniük és ellenőrizniük.

A számítógépek, háztartási készülékek közül példaként nézzük meg a mikroszámítógépes tűzhelyet. Használójának csak a következő billentyűket kell megnyomnia: megsütendő ételfajta, annak súlya és a tálalás időpontja. A mikroszámítógép beindítja a sütési folyamatot a megfelelő időpontban, minden sütési fázisban beállítja az optimális hőmérsékletet, és a kívánt időpontra kész az étel.

További példák mikroszámítógépes készülékekre: pénztárgépek, árszámító mérlegek, szerszámgépek, elektronikus orgonák, HIFI- és videoberendezések, fedélzeti számítógépek gépjárműveken stb.

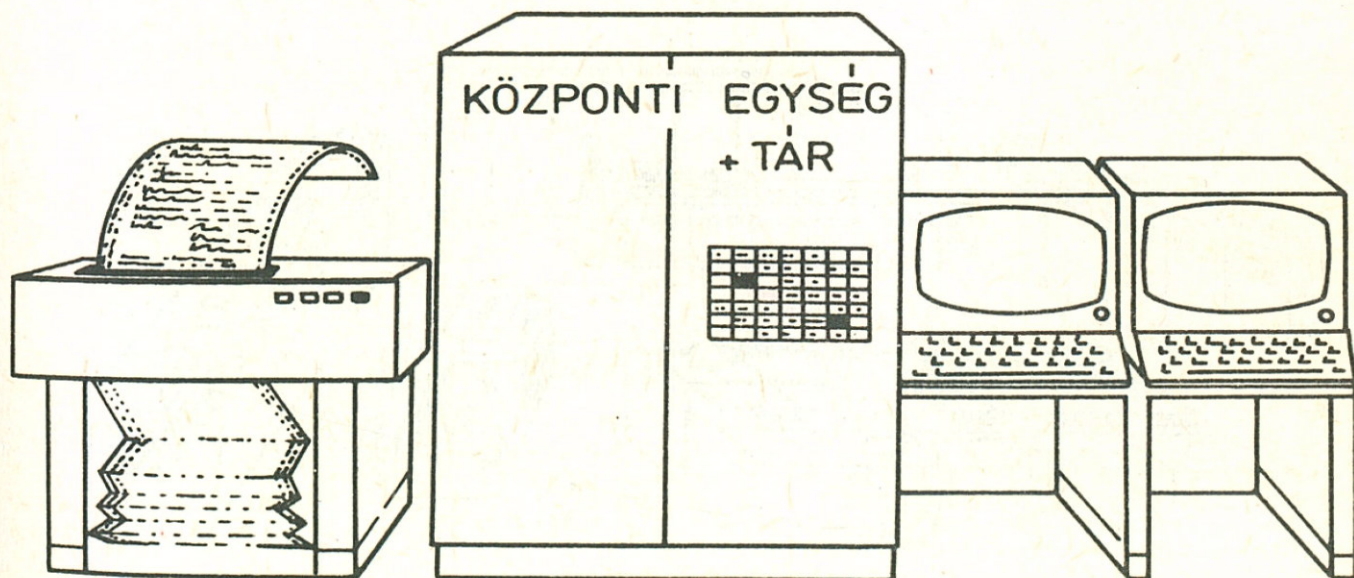


Kisszámítógépek

A mikroszámítógép után következő számítógép-kategóriát **kisszámítógépnek** (miniszámítógépnek) nevezzük. Használják az irodai számítógép elnevezést is, mert a kisszámítógépeket gyakran kombinálják számlázó- és könyvelőgépekkel, és így többnyire a hivatalokban találhatók. Az adatok feldolgozását azok keletkezési helyén, tehát a munkahelyen teszik lehetővé.

A kisszámítógépek a mikroszámítógépekhez hasonlóan programozhatók. Azonban a számítógépeknek ezt a kategóriáját a legkülönbözőbb alkalmazási területekre (pl. bérelszámolás, raktárnyilvántartás, könyvelés) kifejlesztett típusprogramokkal (ún. tipizált programokkal) látják el, mivel általában a vevőknek nincs mindig megfelelően iskolázott személyzetük, akik a programokat saját maguk meg tudnák írni, az egyedi programozás pedig túlságosan költséges volna.

A mikroszámítógéppel ellentétben a kisszámítógépnek többnyire nagyobb a tárkapacitása és a feldolgozási sebessége, továbbá nagyobb műveleti sebességű perifériák csatlakoztathatók hozzá. A gyors mágneslemeztárakkal, képernyős terminálokkal és nyomtatóval ellátott kisszámítógépek nagy teljesítményű számítógépek, amelyek univerzálisak, azaz nem csak egy bizonyos alkalmazási esetben használhatók.



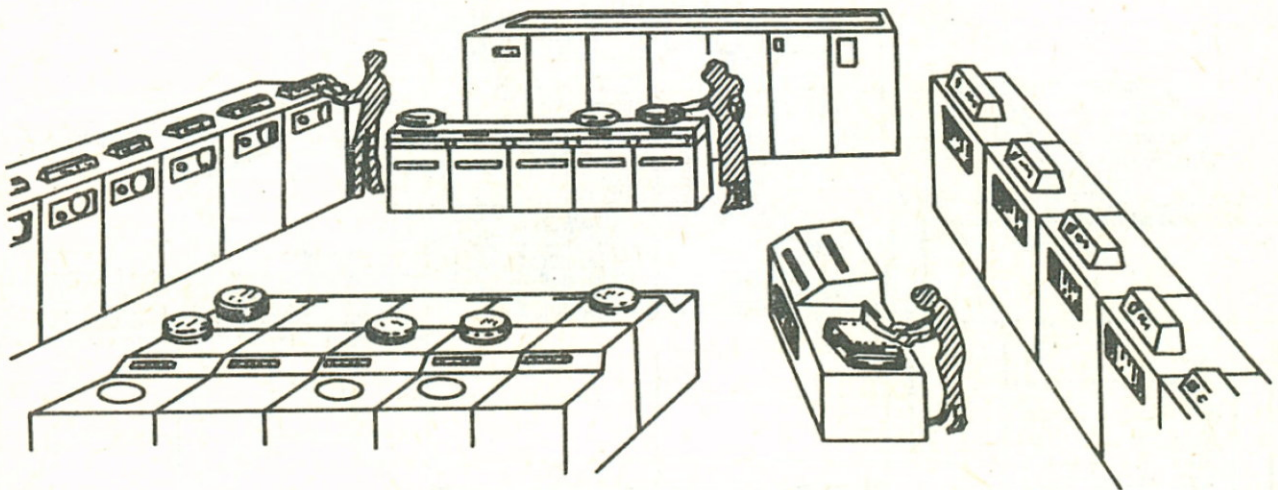
Nagyszámítógépek

Ha számítógépekről beszélünk, akkor többnyire egy számítóközpontban elhelyezett **nagyszámítógépre** gondolunk. Ott rendszerint több mágneslemezes tárral, mágnesszalagos tárral és nyomtatóval, valamint sok más perifériával ellátott nagy teljesítményű központi egységek találhatók. Az ebbe a kategóriába tartozó számítógépek állandó hőmérsékletet és légnedvességtartalmat igényelnek. Ezért klimatizált **számítógéptermekekben** (számítóközpontokban) helyezik el őket.

Nagyszámítógép-berendezésekhez száz vagy annál is több terminál kapcsolható. A terminálok nem a számítóközpontban vannak elhelyezve, hanem használóik munkahelyén, és telefon- vagy adatvonalakkal vannak a számítóközponttal összekötve. E számítógépek nagy feldolgozási sebessége a felhasználóknak azt az érzést kölcsönzi, mintha egyedül ők dolgoznának a géppel, holott mindegyikük esetenként csak a másodperc tört részéig használhatja a központi egységet, mielőtt a következő felhasználó sorra kerül.

A nagyszámítógépeknek rendszerint több millió byte kapacitású operatív tára van. Háttértáron nagy adatállományok (adatbázisok) és az alkalmazási esetek széles spektrumát lefedő programok (programkönyvtár) tárolhatók.

A következő ábra egy nagyszámítógéppel és perifériákkal igen jól ellátott számítóközpontot mutat be.



Nagyszámítógépeket ott alkalmazunk, ahol különösen *sok* adatot kell feldolgozni, vagy rendkívül *gyorsan* akarjuk megkapni a feldolgozás eredményét.

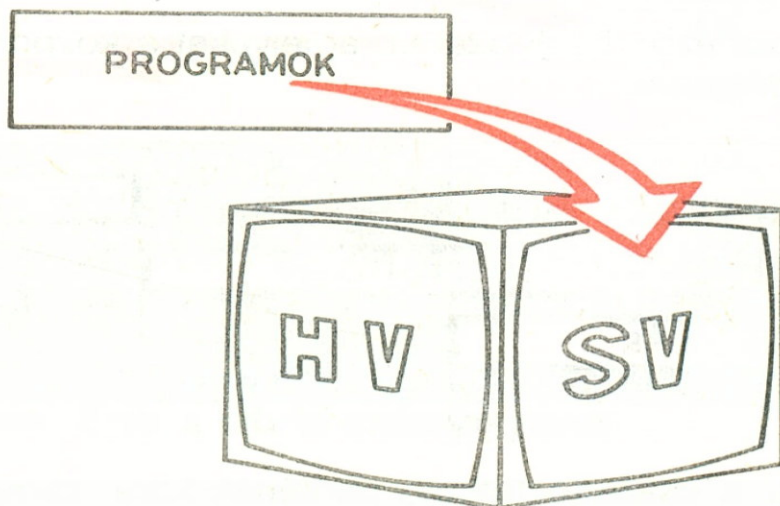
A nagyszámítógépekkel nagy tömegű adatok kezelhetők. Ilyen nagy tömegű adatok feldolgozására a hatóságoknál, kórházakban, energiaellátó vállalatoknál, pénzügyintézeteknél, biztosító intézetekben, áruházakban, ipari vállalatoknál stb. lehet szükség.

Az ipar a termelési folyamatok tervezéséhez és ellenőrzéséhez, a feladatok kidolgozásához, laboratóriumi kísérleteknél bonyolult számítások elvégzésére és a bérelszámolásokhoz alkalmazza a nagyszámítógépeket.

De a tudományos életben is megtalálhatók a nagyszámítógépek. Elsődlegesen ott, ahol bonyolult számításokat kell elvégezni, esetleg olyanokat, amelyek elvégzésére egy emberi élet sem volna elegendő. A modern űrhajózás a nagyszámítógépek nélkül elképzelhetetlen volna.

A nagyszámítógépek alkalmazási területei szinte végtelenek! A sokféle feladat mindegyike ugyanazzal a nagyszámítógéppel megoldható.

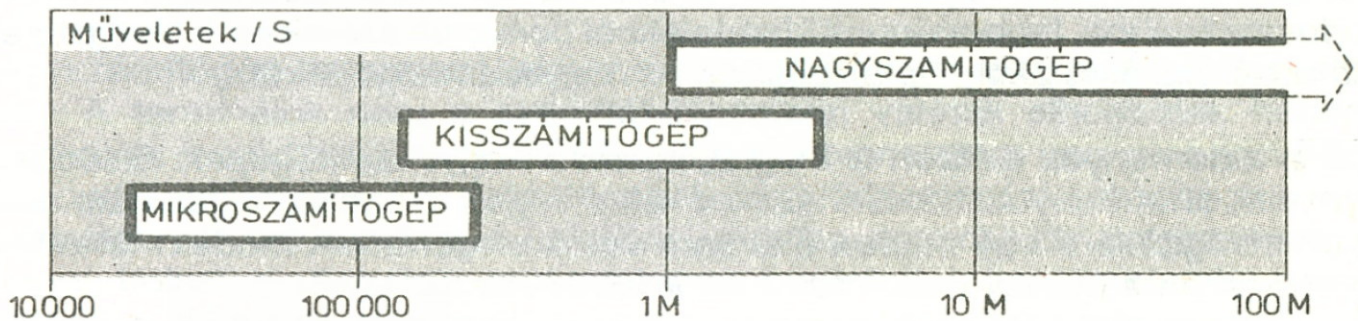
A nagyszámítógépeket majdnem mindig **univerzális számítógépként** használják, vagyis azokkal a legkülönbözőbb feladatok oldhatók meg a hardver változtatása nélkül. A megfelelő **szoftvernek** viszont természetesen meg kell lennie, azaz azoknak a programoknak, amelyek beolvasása után a számítógép az adatokat az előírtak szerint feldolgozza.



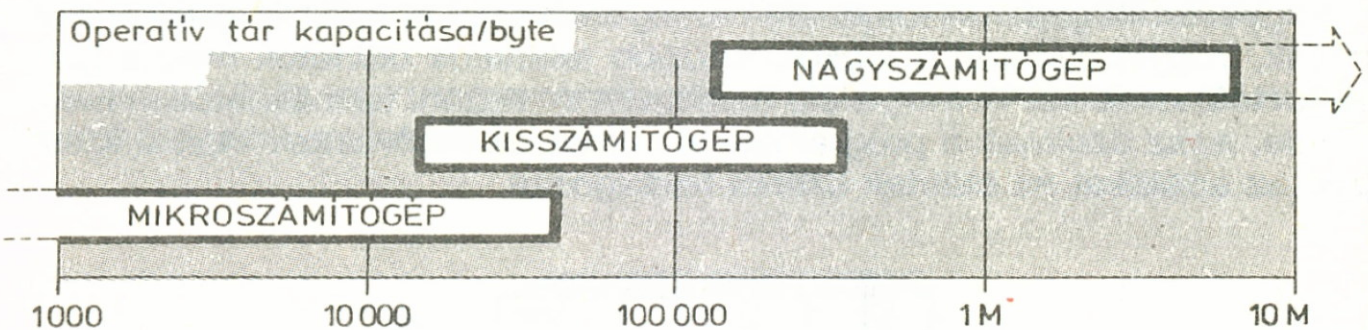
A mikro-, a kis- és a nagyszámítógépek között nincsenek éles határok. A teljesítményjellemzők az egyes számítógép-kategóriákra mégis közelítőleg megadhatók.

A legfontosabb teljesítményjellemzők*:

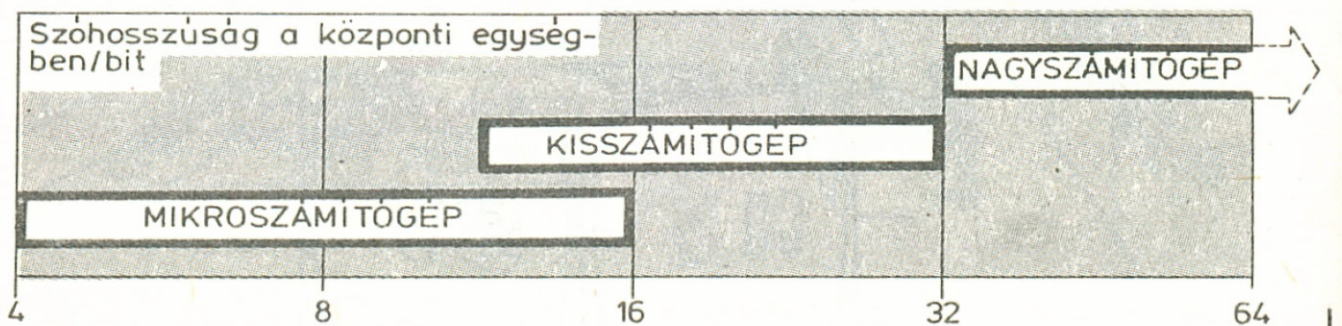
- a **feldolgozási sebesség**, azaz a központi egység által másodpercenként végrehajtott utasítások száma;



- a **tárkapacitás**, amely megadja, hogy a számítógép operatív tára hány byte-ot képes tárolni;



- a **szóhossz**, azaz azon bitek száma, amelyeket a központi egység egyidejűleg tud feldolgozni.



További jellemzők: a csatlakoztatható perifériák köre, kompatibilitás, azaz csereszabotosság más számítógépekkel, az adatátviteli sebesség, és természetesen a vételár.

* A megadott teljesítményjellemzők kb. az 1980. évi viszonyokat tükrözik, értékük természetesen évről évre tovább nő. (Lektor.)

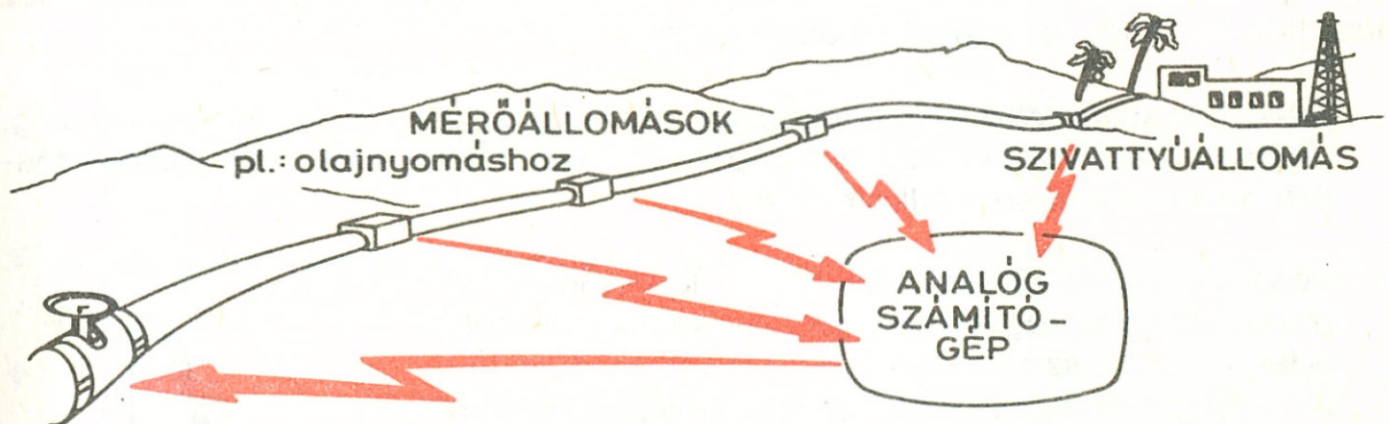
Digitális és analóg számítógépek

Az adatfeldolgozás területén szinte mindenütt **digitális számítógépek** vannak. Ebben a kötetben majdnem kizárólag digitális számítógépekről van szó. Felépítésüket és a hardver működését a következő fejezetekben ismertetjük.

Mégis, most egy rövid áttekintést adunk az analóg számítógépekről, amelyek használata különleges esetekben célszerű, pl. kémiai folyamatok vezérléséhez stb.

Az **analóg számítógépek**, mint nevük is mutatja, **analóg** adatokat dolgoznak fel*. Ezek nem számjegyekkel (digitálisan) ábrázolt értékek, hanem fizikai mérési értékek, mint pl. feszültségek, áramok, ellenállások, amelyek a mért rendszer hőmérsékletével, nyomásával, sebességével, szögeivel stb. arányosak.

Ezeket a mérési értékeket az analóg számítógép elektromos vagy pneumatikus módon dolgozza fel, az eredményeket folyamatosan megjeleníti, vagy azonnal a mért rendszerek szabályozására használja fel, amennyiben pl. egy bizonyos helyen a hőmérséklet vagy a nyomás megváltozik.



Az analóg számítógépek elsődlegesen olyan alkalmazásokhoz előnyösek, ahol mind a bemenő adatok, mind pedig az eredmények analóg ábrázolásúak, mint pl. a folyamatirányítás esetében.

Olyan számítógépek is vannak, amelyek mind digitálisan, mind pedig analóg módon dolgoznak. Ezek a **hibrid számítógépek**.

* Az analóg értékeket alapvetően az jellemzi, hogy azok egy adott értéktartományon belül tetszőleges értéket vehetnek fel és így folytonosan változtathatók. (Lektor.)

Univerzális és speciális rendszerek

Ha a számítógépeket alkalmazásuk szempontjából vizsgáljuk, akkor univerzális és speciális rendszereket szokás megkülönböztetni.

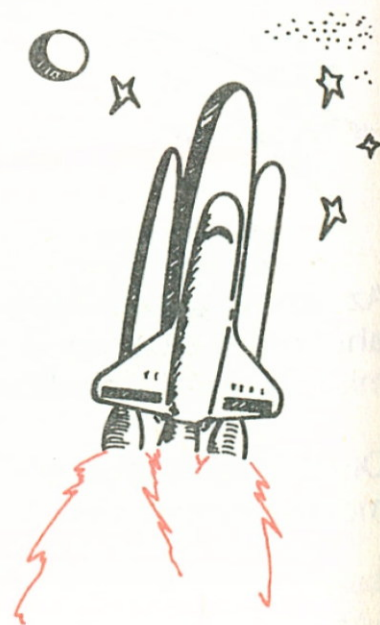
Univerzális rendszereknek nevezzük azokat a nagyobb adatfeldolgozó berendezéseket, amelyek technikai—tudományos és kereskedelmi célokra egyaránt alkalmasak. Az univerzális rendszerek a feladatok legkülönbözőbb fajtáit látják el (pl. valamely vállalat bérelszámolása, raktárkezelés, számlázás, tájékoztatási rendszer, mérlegkészítés, elszámolás stb.).



Speciális rendszereknek viszont azokat a rendszereket nevezzük, amelyekben az adatfeldolgozó berendezéseket bizonyos feladatok megoldásához alakítjuk ki. Ilyen speciális rendszerek pl.:

- a **folyamatirányító számítógépek**, amelyek alkalmazásuk szerint analóg, digitális vagy hibrid számítógépek, egyedi folyamatok, pl. közlekedés-áramlás irányítására alkalmasak;
- **vektorszámítógépek**, amelyek különösen a *nagy műveletvégző sebességet* követelő feladatokhoz használatosak**.

A vektorszámítógépnek, amelyet párhuzamos gépnek is nevezünk, sok processzora van, amelyhez saját táruk tartoznak. Az adatok bevitelét és kivitelét eközben egy univerzális számítógép végzi, amely vezérlőegységként működik. Így a feladatok tized- vagy századannyi idő alatt számíthatók ki, mint amennyire egy hagyományos számítógépnek erre szüksége lenne.



* Kissé pontosabban fogalmazva: a vektorszámítógépek vektorokkal végzendő műveletek nagy sebességű végrehajtására képesek.



1. Mely megállapítások igazak a zseb- és asztali számológépekre, és melyek a programozható mikroszámítógépekre?

	Zseb- és asztali számoló- gépek	Progra- mozható mikro- számító- gépek
Intelligens terminálokba vannak beépítve.	<input type="radio"/>	<input type="radio"/>
Központi egységük van.	<input type="radio"/>	<input type="radio"/>
Programnyelven programozhatók.	<input type="radio"/>	<input type="radio"/>
Szövegek feldolgozására képesek.	<input type="radio"/>	<input type="radio"/>
A központi egység többnyire képernyőterminállal és háttértárral, pl. hajlékony mágneslemezes tárral van ellátva.	<input type="radio"/>	<input type="radio"/>
Készülékeket, pl. pénztárgépeket, fedélzeti számítógépeket, háztartási eszközöket vezérelnek.	<input type="radio"/>	<input type="radio"/>
Személyi számítógépként is használatosak.	<input type="radio"/>	<input type="radio"/>

2. Mi a különbség a mikro- és a kisszámítógépek között?

- A kisszámítógépnek nagyobb a tárkapacitása.
- A kisszámítógép feldolgozási sebessége általában nagyobb.
- A kisszámítógép perifériaválasztéka nagyobb.

3. Minek nevezik azokat a speciális számítógépeket, amelyek vektorokkal igen gyors műveletvégzésre képesek?

.....

4. Sorolja fel a számítógépek néhány teljesítményjellemzőjét!

.....

.....



Válaszok

1. A következő megállapítások igazak a zseb- és asztali számológépekre, ill. a programozható mikroszámítógépekre:

Zseb- és asztali számoló- gépek	Progra- mozható mikro- számító- gépek	
<input type="radio"/>	<input checked="" type="radio"/>	Intelligens terminálokba vannak beépítve.
<input checked="" type="radio"/>	<input checked="" type="radio"/>	Központi egységük van.
<input type="radio"/>	<input checked="" type="radio"/>	Programnyelven programozhatók.
<input type="radio"/>	<input checked="" type="radio"/>	Szövegfeldolgozásra képesek.
<input type="radio"/>	<input checked="" type="radio"/>	A központi egység többnyire képernyőterminállal és háttértárral, pl. hajlékony mágneslemezes tárral ellátva.
<input type="radio"/>	<input checked="" type="radio"/>	Készülékeket, mint pl. pénztárgépeket, fedélzeti számítógépeket, háztartási eszközöket vezérelnek.
<input type="radio"/>	<input checked="" type="radio"/>	Személyi számítógépként is használatosak.

2. A mikroszámítógép és a kisszámítógépek közötti különbségek:

- A kisszámítógépnek nagyobb a tárkapacitása.
- A kisszámítógép feldolgozási sebessége általában nagyobb.
- A kisszámítógép perifériaválasztéka nagyobb.

3. Vektorszámítógépnek nevezzük azokat a speciális számítógépeket, amelyek vektorokkal igen gyors műveletvégzésre képesek.

4. Számítógépek teljesítményjellemzői:

- **feldolgozási sebesség,**
- **tárkapacitás,**
- **szóhosszúság,**
- **csatlakoztatható perifériák köre,**
- **adatátviteli sebesség.**



5. Mely kijelentések igazak a nagyszámítógépekre?

A nagyszámítógépek rendszerint klimatizált számítóközpontokban vannak elhelyezve.

A nagyszámítógéphez száz vagy ennél is több terminál kapcsolható.

A nagyszámítógépeket főként egy meghatározott feladatra alkalmazzák.

A nagyszámítógépeknek rendszerint több byte kapacitású operatív tára van.

A nagyszámítógépeket akkor alkalmazzuk, ha különösen sok adatot kell feldolgozni, vagy ha az eredményeket különösen gyorsan akarjuk megkapni.

A nagyszámítógépek majdnem mindig univerzális gépek.

6. Mely adatok dolgozhatók fel analóg (A), digitális (D) vagy hibrid (H) számítógépekkel digitál—analóg átalakítás nélkül?

Feszültségek

Számjegyekkel ábrázolt értékek

Nyomás

7. A következő adatfeldolgozó berendezések esetén rendszerint univerzális (U) vagy speciális (S) rendszerről van-e szó?

Adatfeldolgozás a közigazgatásban.

Számítógépek a közlekedés irányítására.

Vektorszámítógépek.



Válaszok

5. A nagyszámítógépekre a következő megállapítások igazak:

- Rendszerint klimatizált számítóközpontokban vannak elhelyezve.
- A nagyszámítógépekhez száz vagy még ennél is több terminál csatlakoztatható.
-
- A nagyszámítógépeknek rendszerint több millió byte kapacitású operatív tára van.
- A nagyszámítógépeket akkor alkalmazzuk, ha rendkívül sok adatot kell feldolgozni, vagy ha különösen rövid idő alatt akarjuk az eredményt megkapni.
- A nagyszámítógépek majdnem mindig univerzális gépek.

6. A felsorolt adatokat a következő számítógépek dolgozzák fel:

A H Feszültségek.

D-H Számjegyekkel ábrázolt értékek.

A H Nyomás.

7. A következő rendszerekről van szó:

- U** Adatfeldolgozás a közigazgatásban.
- S** Számítógépek a közlekedés irányítására.
- S** Vektorszámítógépek.

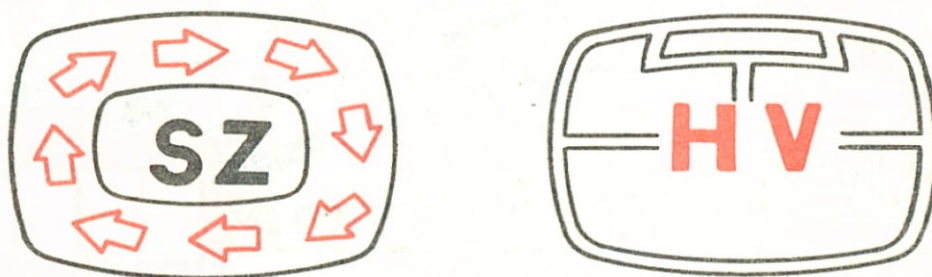
A számítógépek üzemmódjai

Függetlenül attól, hogy valamely feladat megoldásához mikro-, kis- vagy nagyszámítógépet alkalmazunk-e, az adatfeldolgozó rendszer tervezésére illetékesnek meg kell határoznia az adatfeldolgozó berendezés üzemmódját.

Ez elsősorban attól függ, hogy

- hol, hogyan és mikor állnak a feldolgozandó adatok rendelkezésre és
- hol, hogyan és mikor kell az eredményeknek rendelkezésre állniuk.

Ez mindenekelőtt *szervezési probléma*, amelyet e sorozat Kulcs a számítógéphez — Szervezés kötetében tárgyalunk, de egyben hardverprobléma is, ha a számítógépek alkalmazásának technikai lehetőségeiről van szó a gyakorlatban.



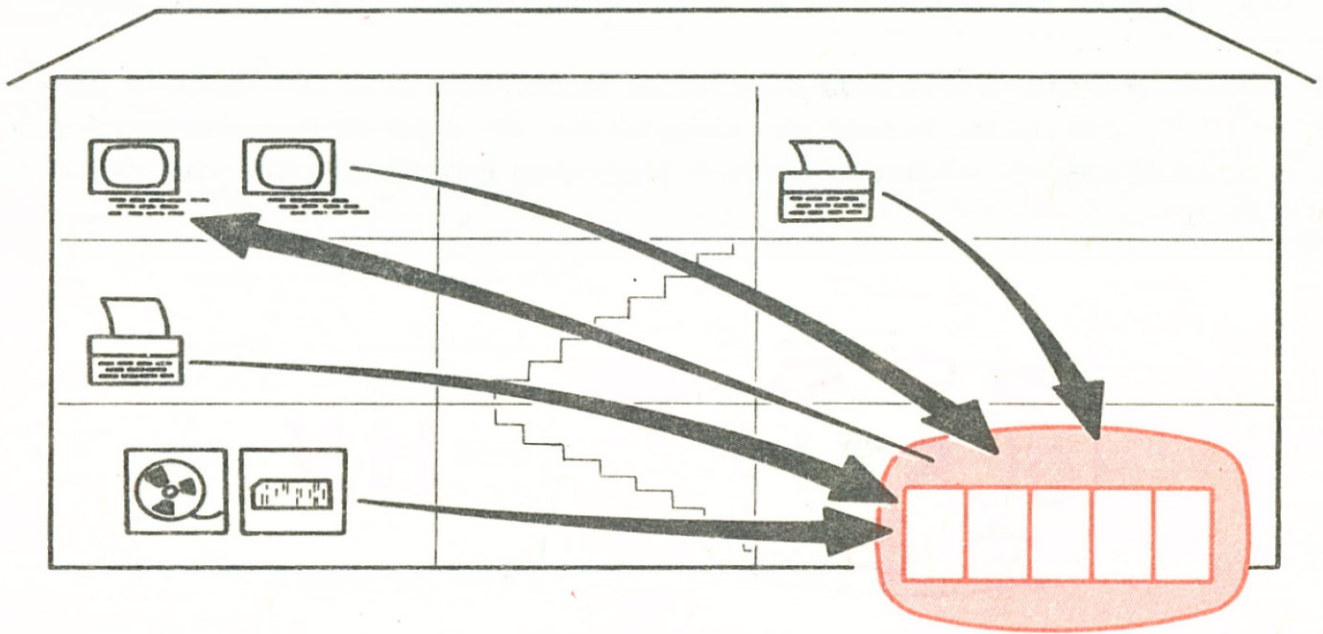
Az adatfeldolgozás kétféleképpen hajtható végre:

- mint **központi adatfeldolgozás** vagy
- mint **decentralizált adatfeldolgozás**.



Központi adatfeldolgozás

Központi adatfeldolgozásról akkor beszélünk, ha egy számítógép valamennyi csatlakoztatott perifériával egy (központi) helyen van telepítve, miközben a feldolgozandó adatokat a számítógéphez kell továbbítani és az eredmények is onnan vihetők el. A számítógépet ez esetben többnyire egy klimatizált számítóközpontban helyezik el.

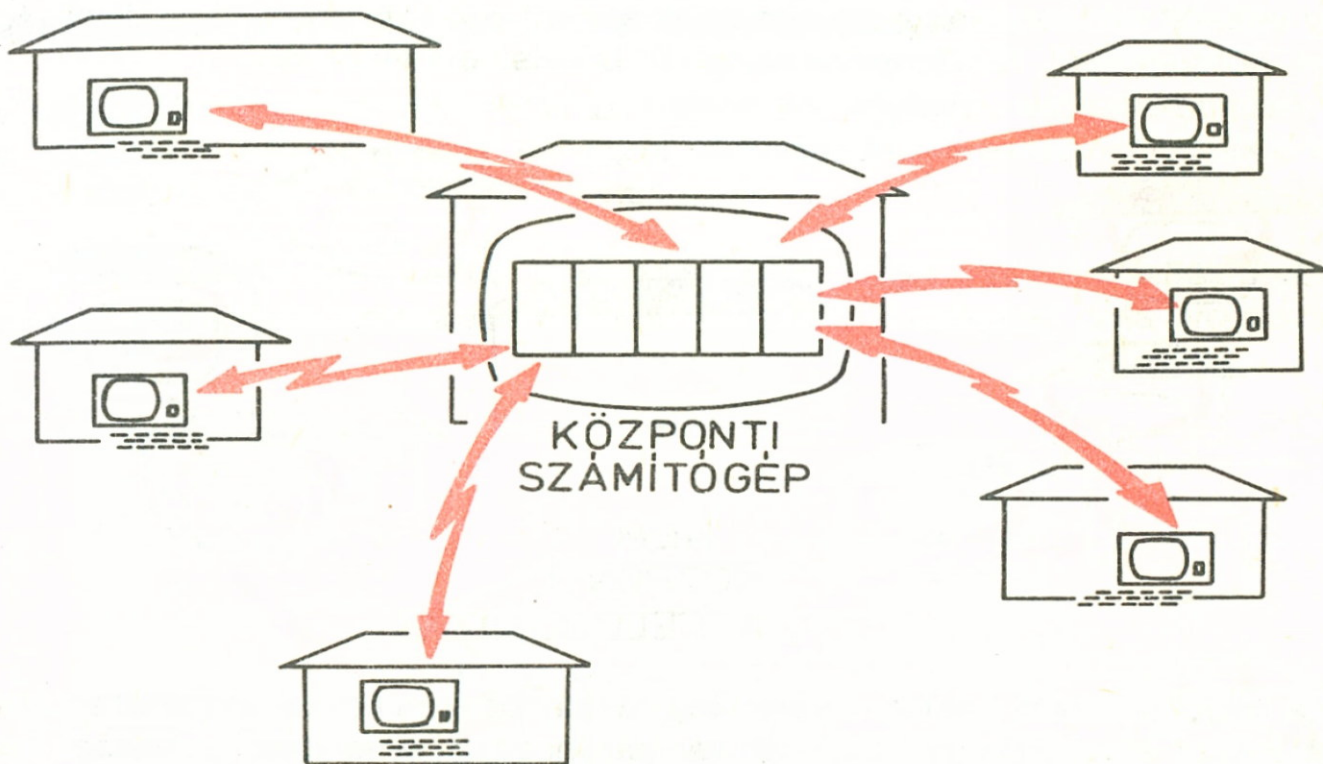


Az adatok rögzítése, tehát az információknak gépi olvasásra alkalmas adathordozókra (mágnesszalagok, mágneslemezek vagy lyukkártyák) vitele vagy ott történik, ahol az adatok keletkeznek, vagy központi adatrögzítő helyiségekben, amelyek a számítóközponthoz tartoznak.

Mivel az adatfeldolgozás a közgazdaság és közigazgatás majdnem minden területén elterjedt, így a központi adatfeldolgozás e tiszta formájának már nincs olyan nagy jelentősége. Az **adatátvitel** segítségével lehetővé vált valamely központi számítógép szolgáltatásának terminálok útján a felhasználó munkahelyére való vitele.

A feldolgozandó adatok rögzítése és bevitele, valamint az eredmények kivitele ebben az esetben ott történik, ahol az adatok keletkeznek és ahol az eredményekre szükség van.

Példa erre egy vállalat, amelynek leányvállalatai terminálokkal vannak összekötve a központi nagyszámítógéppel.

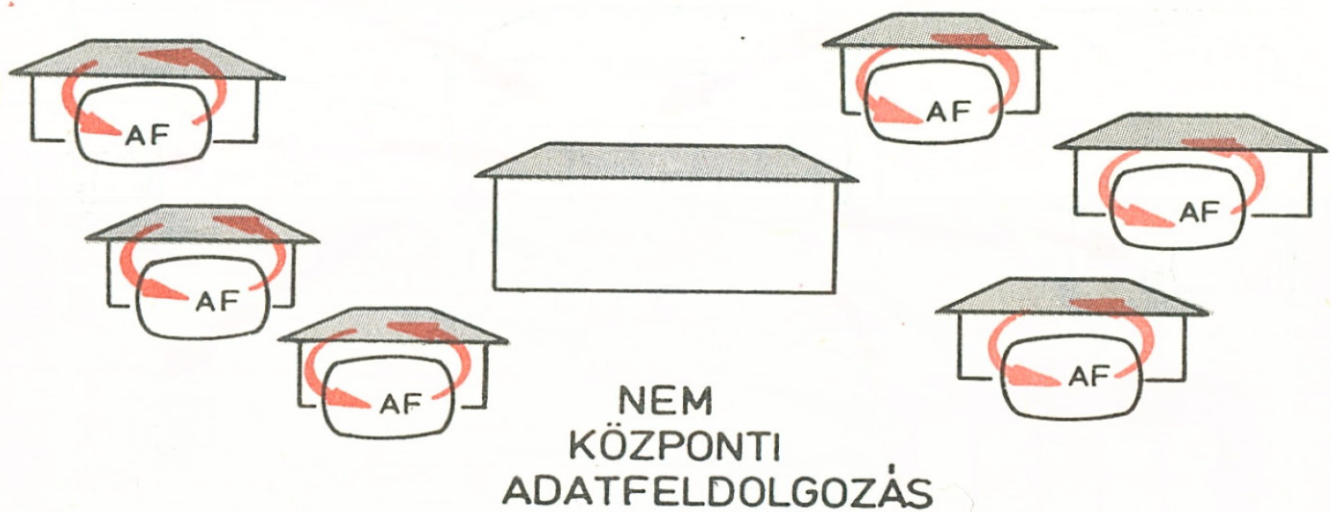


Ebben az esetben is központi adatfeldolgozásról beszélünk, mert bár a számítógép szolgáltatásait a központtól távoli helyekre juttatják, a tulajdonképeni adatfeldolgozás kizárólag a központi számítóközpontban történik.

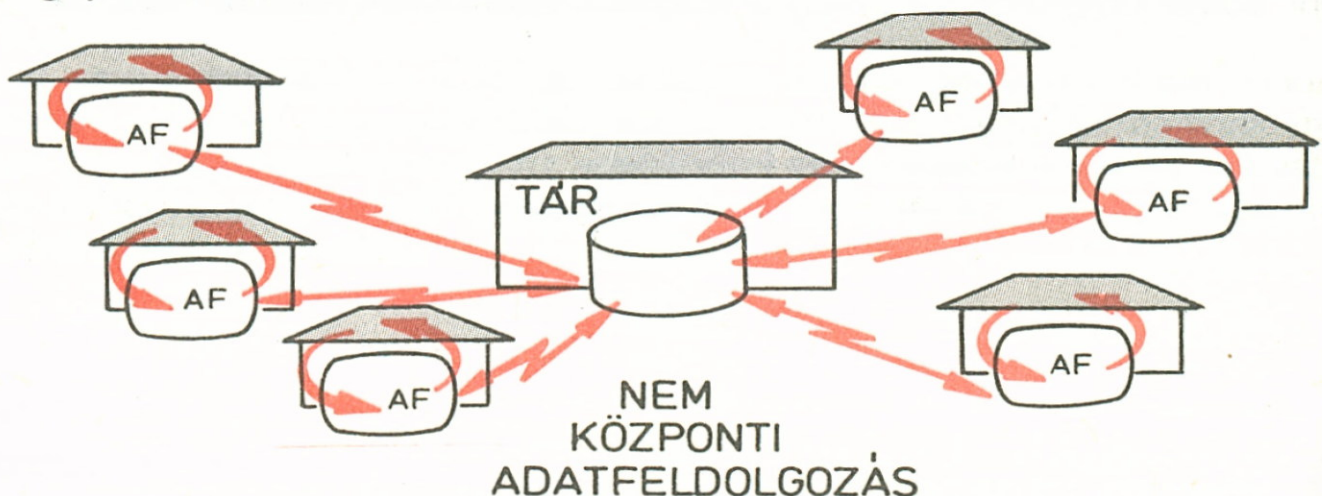
Decentralizált adatfeldolgozás

Míg a központi adatfeldolgozás esetén rendszerint egy számítóközpontban telepített nagyszámítógép nyújtja a szolgáltatásokat, addig a **decentralizált adatfeldolgozás** esetén többnyire több kisebb számítógépet (kis- vagy mikro-számítógépet) alkalmaznak, amelyek közönséges irodai környezetben (klimatizálás nélkül) tudnak dolgozni. A decentralizált adatfeldolgozás jellegzetesége tehát az, hogy a *feldolgozást azokra a helyekre helyezik át, ahol a bevitelre váró adatok keletkeznek és ahol az eredményre szükség van.*

Egy vállalat minden leányvállalata ezek szerint saját kis számítógépével végzi el pl. könyvelését, raktárnyilvántartását, bérelszámolását stb.



Ha a feldolgozandó adatokra valamely vállalat összes fiókvállalatánál szükség van, akkor célszerű egy **központi tár** létrehozása, amelyhez az összes helyi számítógép hozzáférhet. Ennek az az előnye, hogy az információkat csak *egyszer kell rögzíteni és tárolni* (pl. cikkszámokat, aktuális árakat). Módosításokat — amelyek pl. árváltozások esetén válnak szükségessé — szintén csak egyszer kell átvezetni. Az adatok feldolgozása továbbra is a *helyi számítógépen* történik.



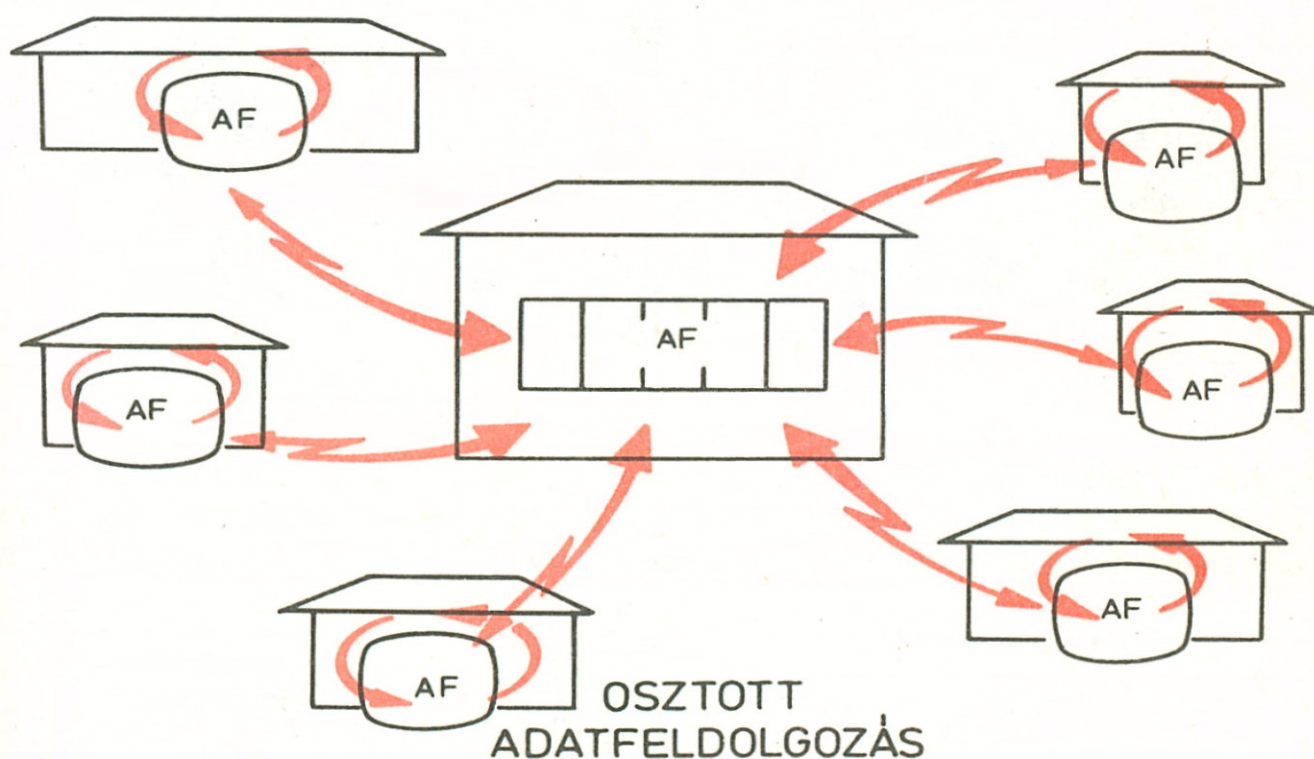
Osztott adatfeldolgozás

Az adatfeldolgozás egy másik formája a *helyi kishámítógépek egy központi nagyszámítógéppel* való összekötése.

Ezzel lehetővé válik az egyszerűbb feldolgozási folyamatoknak a helyi kishámítógépekkel való decentralizált elvégzése a helyszínen, és a központi számítógépnek csak nehéz, számításigényes feladatok esetén való igénybevétele. A központi számítógép használatára viszont szükség van pl. az egész vállalatot érintő kiértékelések vagy olyan folyamatok esetén, amelyek a központi adattár gyakori igénybevételét teszik szükségessé.

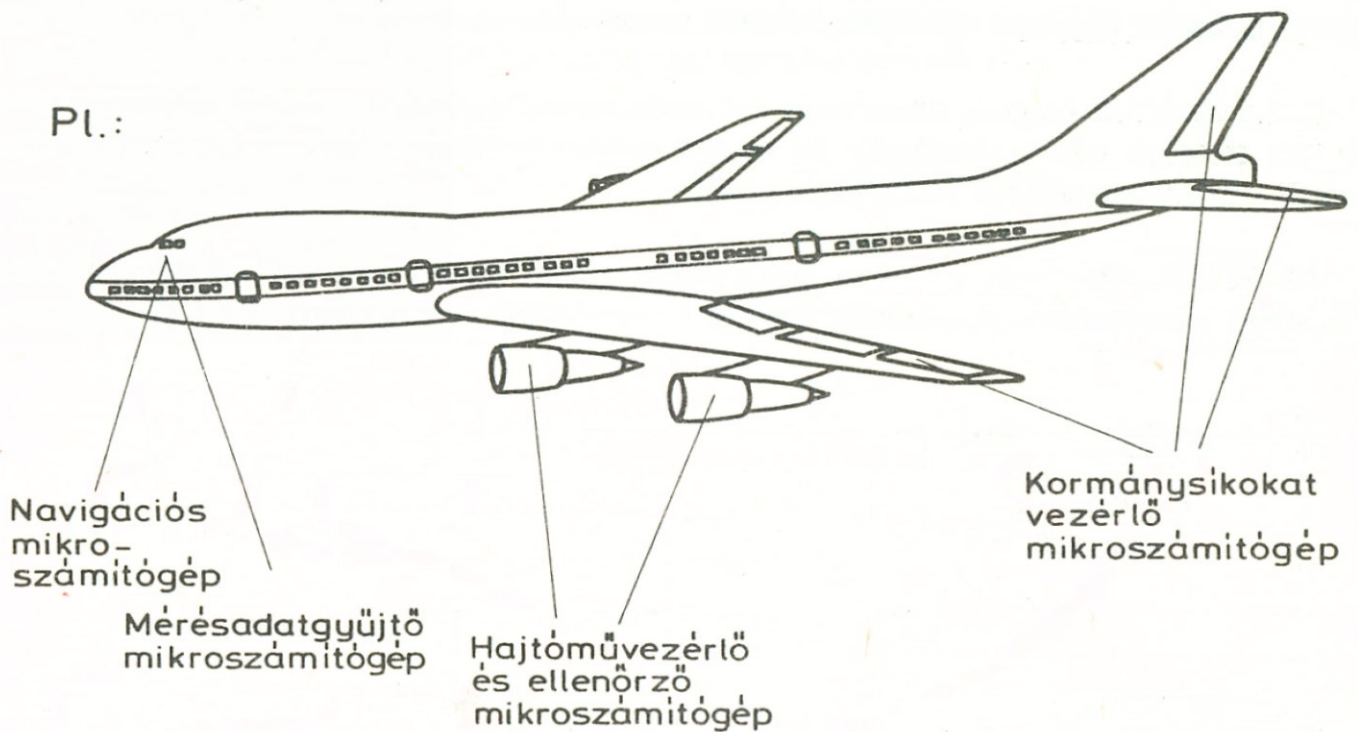
A helyi számítógépek átvehetik az adatoknak egyfajta „előfeldolgozását” is, pl. az adatok osztályozását, és ezzel nagymértékben tehermentesítik a központi számítógépet a rutinmunkáktól.

A funkciók előbbiek szerinti felosztása miatt ezért ezt az adatfeldolgozást **osztott adatfeldolgozásnak** nevezik (angolul: distributed data processing).



A gyakorlatban nemcsak irodai számítógépeket kötünk össze egy központi számítógéppel, hanem irodai számítógépek egymással vagy irodai és nagyszámítógépek tetszőleges kombinációban is összeköthetők.

További példa az osztott adatfeldolgozásra a technika területén: egy modern utasszállító repülőgép vezérlése több decentralizált mikroszámítógéppel, amelyek önállóan végzik részfeladataikat, azonban egymással kapcsolatban állnak és folyamatosan jelentéseket küldenek saját „kezelési területük” mindenkori állapotáról a vezetőfülkében elhelyezett mikroszámítógépnek, amely a műszerfalon adja jelzéseit.

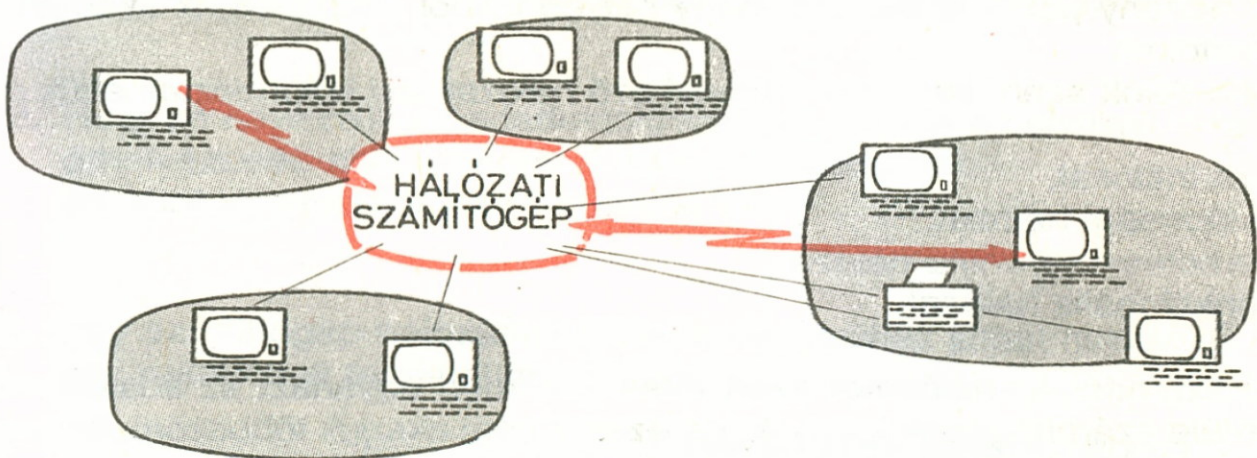


Számítógép-hálózatok

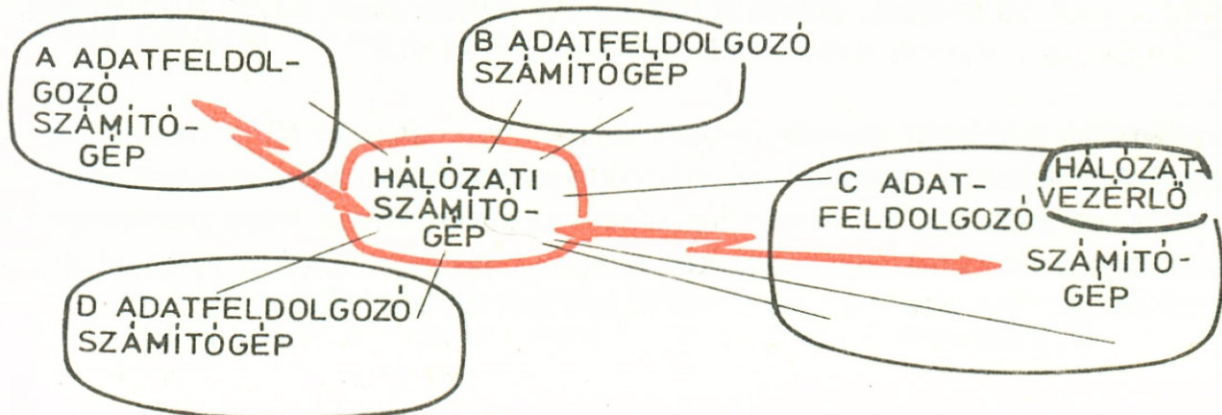
Számítógép-hálózatról akkor beszélünk, ha több számítógép több adatállomással (terminállal) egy átviteli hálózaton keresztül van összekötve, amely a szükséges átviteli berendezésekből és egy vagy több hálózati számítógépből áll.

Egy ilyen rendszerhez kapcsolt terminál felhasználója nincs egy bizonyos számítógépre korlátozva, és a rendszerhez tartozó összes berendezést igénybe veheti.

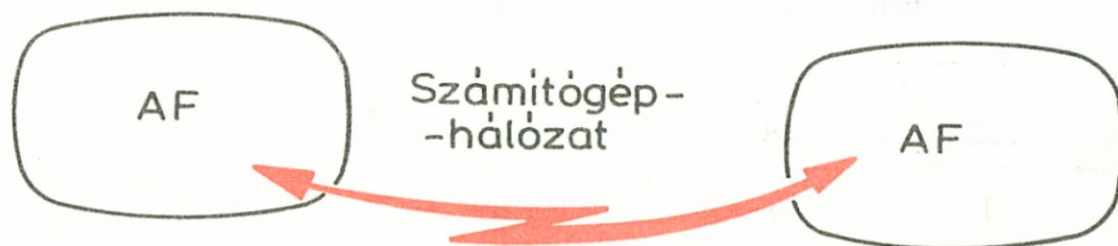
Tehát kiválaszthat egy tetszőleges számítógépet, amellyel programjait feldolgoztatni kívánja, egyik számítógéptől a másikhoz vitethet át adatokat és programokat — természetesen csak akkor, ha erre fel van jogosítva —, és végül információkat küldhet saját termináljától egy másik kapcsolt terminálhoz.



Az egész rendszer vezérlését és felügyeletét a csatlakoztatott *feldolgozó számítógépek* egyike végezheti, vagy a feladat egy hálózati számítógépre bízható, amelyet a rendszer vezérlésére és felügyeletére specializálunk.



A számítógép-hálózat további lehetőségeket nyújt a **gép—gép kommunikációban**. Ha valamely számítógépben futó programnak olyan adatokra van szüksége, amelyek egy másik számítógépben vannak tárolva, akkor a két számítógép *emberi beavatkozás nélkül* önállóan kapcsolatba léphet egymással, és a kívánt adatokat ki tudja cserélni. Amennyiben ezek az adatok nem nyilvánosak, akkor ebben az esetben is automatikus jogosultsági ellenőrzés történik.



Ez a néhány példa is mutatja, hogy milyen célból létesíthetünk számítógép-hálózatokat.

Az elvi célok szempontjából a következő hálózatokat különböztethetjük meg:

- adathálózat,
- biztonsági hálózat,
- terhelés kiegyenlítő hálózat,
- funkcionális hálózat.

Ha a számítógép-hálózatot azért létesítjük, mert ugyanazt az adatállományt különféle számítógépek igénylik, de ezeket nem akarjuk többszörösen tárolni, akkor **adathálózatról** beszélünk.

A **biztonsági hálózatnak** ezzel szemben az a célja, hogy valamely számítógép kiesése esetén más számítógép vehesse át a fontos feladatok elvégzését.

A **terhelés kiegyenlítő hálózat** arra alkalmas, hogy a rendszer teljes számítási kapacitását kihasználjuk, mivel a felmerülő feladatokat azzal a számítógéppel végeztethetjük, amelynek még szabad kapacitása van.

A **funkcionális hálózat** esetén olyan számítógépeket kötünk össze, amelyeknek egyéni feladatkomplexumok megoldásához speciális hardver- és szoftver-eszközeik vannak (pl. adatokat közvetlenül mikrofilmre kiíró perifériájuk stb.). A számítógép-hálózat útján ezeket a speciális eszközöket más előfizetők is használhatják, így ezek az eszközök jobban kihasználhatók.



1. Mit jelent a központi adatfeldolgozás kifejezés?

Jelölje meg az igaz megállapításokat kereszttel!

Az adatok rögzítése csak központi helyiségekben történik.

Az adatfeldolgozó berendezés egy központi helyen levő számítóközpontban van felállítva.

Az adatok bevitele csak a számítóközpontban történik.

Központi adatfeldolgozás esetén a számítógép szolgáltatásai decentralis helyekre is átvihetők.

Központi adatfeldolgozás esetén a tulajdonképpeni feldolgozás kizárólag a számítóközpontban történik.

2. Hogyan nevezzük az adatfeldolgozásnak azokra a helyekre történő áthelyezését, ahol a bemenő adatok keletkeznek, és ahol az eredményekre szükség van?

3. Egy számítógép-hálózat mely berendezéseihez juthat hozzá egy rákapcsolt terminál használója?

Csak ahhoz a számítógéphez, amelyre saját terminálja kapcsolódik.

Csak saját feldolgozó számítógépéhez és a hálózati számítógéphez.

A rendszerhez kapcsolt valamennyi berendezéshez.

4. Mely hálózati elveket ismert meg?

.....

5. Melyik hálózati elvet valósítjuk meg, ha mindig azzal a számítógéppel végeztetünk feladatokat, amelynek még van szabad kapacitása?

.....



Válaszok

1. A központi adatfeldolgozásra a következő megállapítások igazak:

Az adatfeldolgozó berendezés egy központi helyen levő számítóközpontban van felállítva.

A központi adatfeldolgozás esetén a számítógép szolgáltatásai decentralis helyekre is átvihetők.

A központi adatfeldolgozás esetén a tulajdonképpeni feldolgozás kizárólag a számítóközpontban történik.

2. Az adatfeldolgozás áthelyezését azokra a helyekre, ahol a bemenő adatok keletkeznek és ahol az eredményekre szükség van, **decentralizált adatfeldolgozásnak** nevezzük.

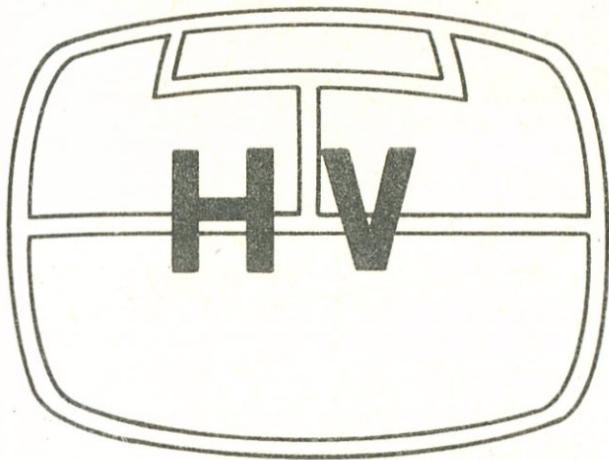
3. Egy számítógép-hálózatra kapcsolt terminál használója a következő berendezésekhez juthat hozzá:

A rendszerhez kapcsolt valamennyi berendezéshez.

4. A megismert hálózati elvek a következők:

adathálózat, biztonsági hálózat, terhelés kiegyenlítő hálózat, funkcionális hálózat.

5. A terhelés kiegyenlítő hálózat elvét valósítjuk meg, ha mindig azzal a számítógéppel végeztetünk feladatokat, amelynek még van szabad kapacitása.



Adatfeldolgozás
és kommunikáció

Számítógépek
és alkalmazásuk

A számítógépek
funkcionális
egységei

Ebben a fejezetben bemutatjuk azokat az egységeket, amelyekre a számítógépnek az adatbevitel, adatkivitel, adattárolás és adatfeldolgozás funkcióinak teljesítéséhez szüksége van.

Megismerheti a bevitelt, kivitelt, tárolást és feldolgozást megvalósító berendezések technikai és gyakorlati jellegzetességeit, és azokat a rendszereket, amelyek segítségével a számítógép központi egysége a perifériákkal kapcsolatban áll.

Hogyan működik
a számítógép?

A bevitel

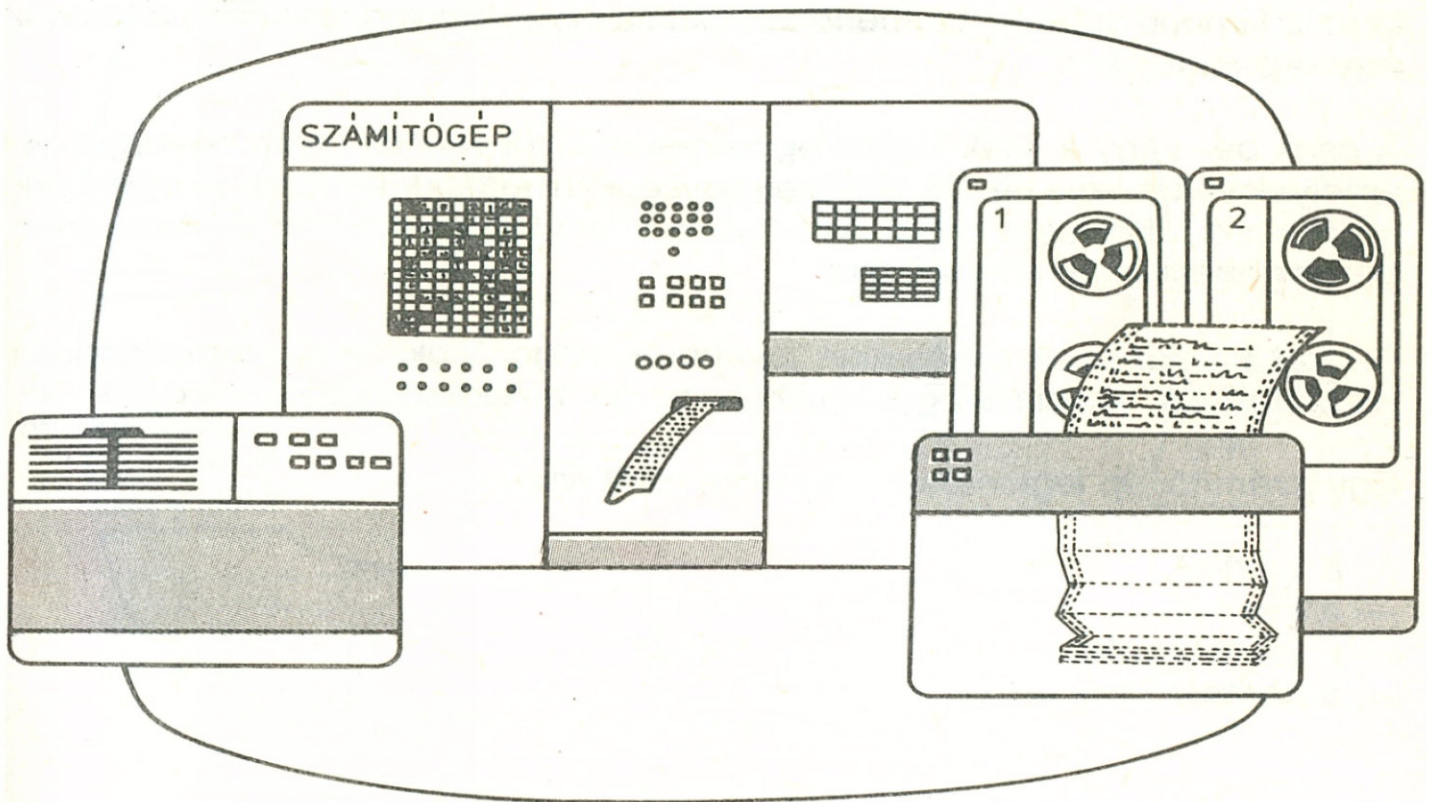
A kivitel

A tárolás

A feldolgozás

Adatátviteli berendezések

A számítógépek olyan egységekből vannak felépítve, amelyek az adatfeldolgozás folyamán meghatározott funkciókat teljesítenek. Ebben a fejezetben a számítógép legfontosabb **funkcionális egységeit** mutatjuk be.



A számítógép fő feladata az adatfeldolgozás. E feldolgozási funkciót a számítógép **központi egysége** látja el. Az adatokat és programokat, amelyek a feldolgozáshoz szükségesek, természetesen előzőleg be kell vinni a számítógépbe. Az adatok és programok bevitele a **beviteli egységek** segítségével történik.

Minden adatfeldolgozás értelme és célja a feldolgozás végén kapott eredmény, azaz a kimenő adatok. A kimenő adatokat az adatfeldolgozás végén a **kiviteli egységek** adják ki.

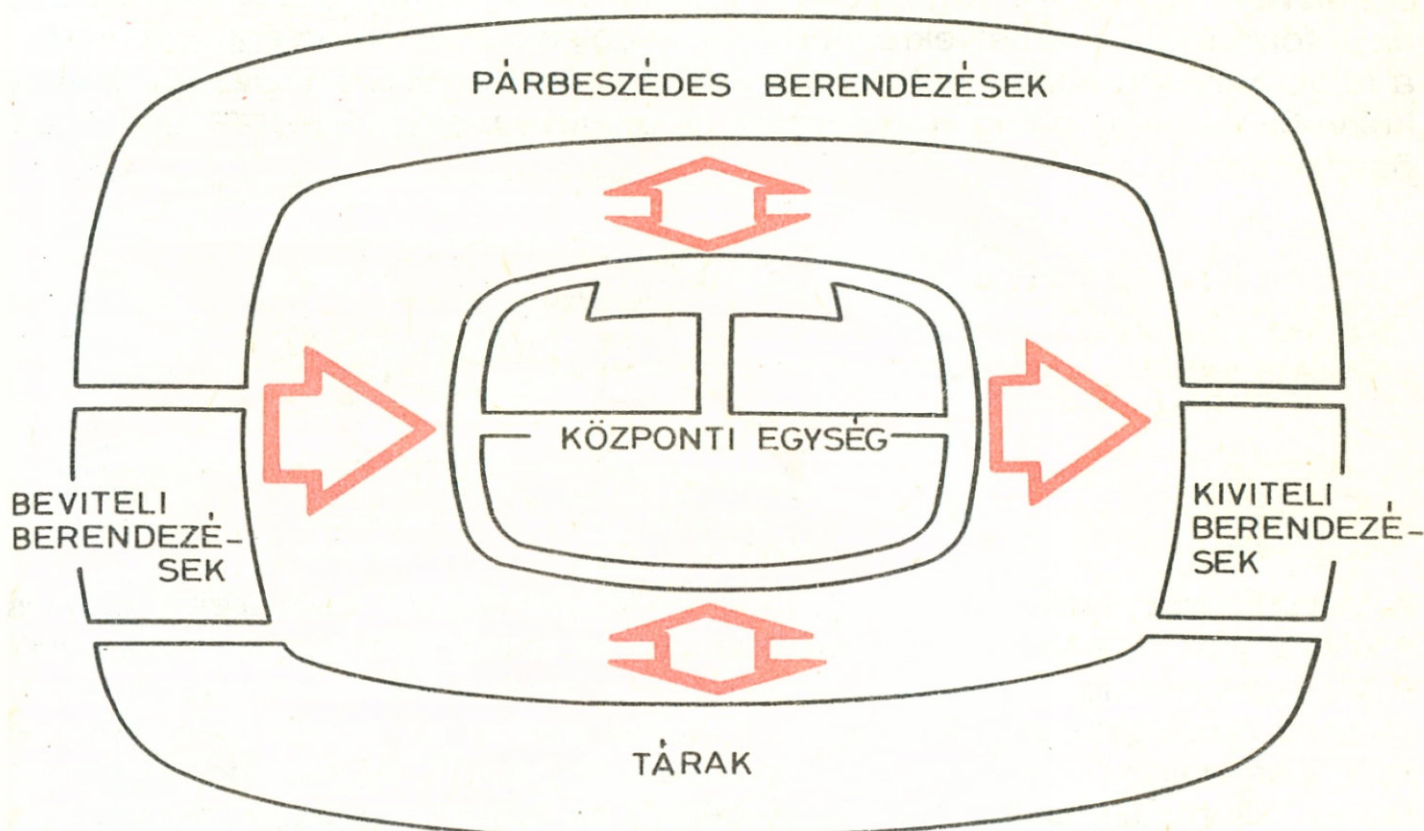
A csak be- vagy a csak kiviteli egységeken kívül vannak olyan egységek is, amelyek váltakozva be-, ill. kimeneti egységként működnek. Ezek az egységek párbeszéd folytatását teszik lehetővé a számítógéppel, ezért **párbeszédés berendezéseknek** nevezik őket.

Azokat az adatokat, amelyeket további feldolgozásokhoz a számítógépben visszatartunk, speciális egységekben, az ún. **tárakban** őrizzük meg.

Egy számítógép legfontosabb funkciói ezek szerint:

- a **bevétel,**
 - a **tárolás,**
 - a **feldolgozás,**
 - a **kivitel.**
-

Ezeket a funkciókat a számítógép egységei végzik, amelyeket az alábbi ábrában tüntettünk fel. Könyvünk, valamint e sorozat más könyvei is, *hardver vonatkozásában* ezeket írják le.

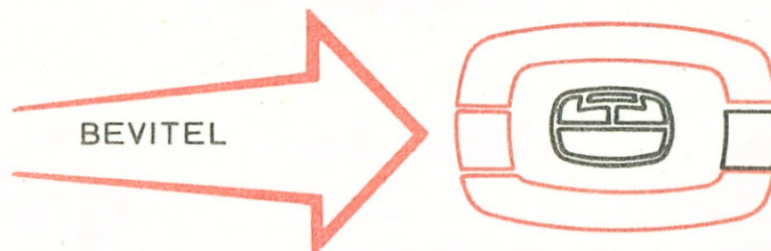


Az egyes egységeket, az egységek részeit és jellegzetességeit a következőkben részletesebben vesszük szemügyre. Azokat az egységeket, amelyek több funkciót is el tudnak látni — pl. a tárolóeszközök, a tárolás mellett a bevitel és kivitel funkcióit is elvégzik —, csak egyszer mutatjuk be részletesen, mégpedig azoknál a funkcionális egységeknél, amelyekhez az eredeti funkciójuk szerint tartoznak. Más egységeknél csak a teljesség kedvéért említjük meg őket.

Mindazokat az egységeket, amelyeknek nem a tulajdonképpeni adatfeldolgozás a feladata, hanem pl. a bevitel, a kivitel és a tárolás egységei, **perifériáknak** nevezzük.

A bevitel

A **beviteli egységek** segítségével visszük be a számítógépbe mindazokat az információkat, amelyekre annak a feldolgozáshoz szüksége van: tehát a feldolgozandó adatokat és a programokat. Mindezek a berendezések vagy közvetlenül, vagy pedig vezetékeken keresztül vannak a központi egységgel összekötve.



A beviteli egységeken kívül gyakran még **adatrögzítő berendezésekre** is szükség van, amelyek a beviendő adatokat az ember által olvasható formából a gép által olvasható formába hozzák, hogy azok a beviteli egységek segítségével feldolgozhatóvá váljanak. Például azokat az adatokat, amelyeket mágnesszalagos egységen keresztül akarunk bevinni, az adatrögzítő berendezés billentyűzetén be kell „gépelnünk”, és azokat így a mágnesszalagra átvinni. Az adatrögzítő egységeket — ellentétben a beviteli egységekkel — a számítógép központi egységétől függetlenül működtetjük.

Amennyiben a különféle beviteli formákhoz adatrögzítő berendezésekre van szükség, azokat a megfelelő beviteli egységeknél említjük meg.

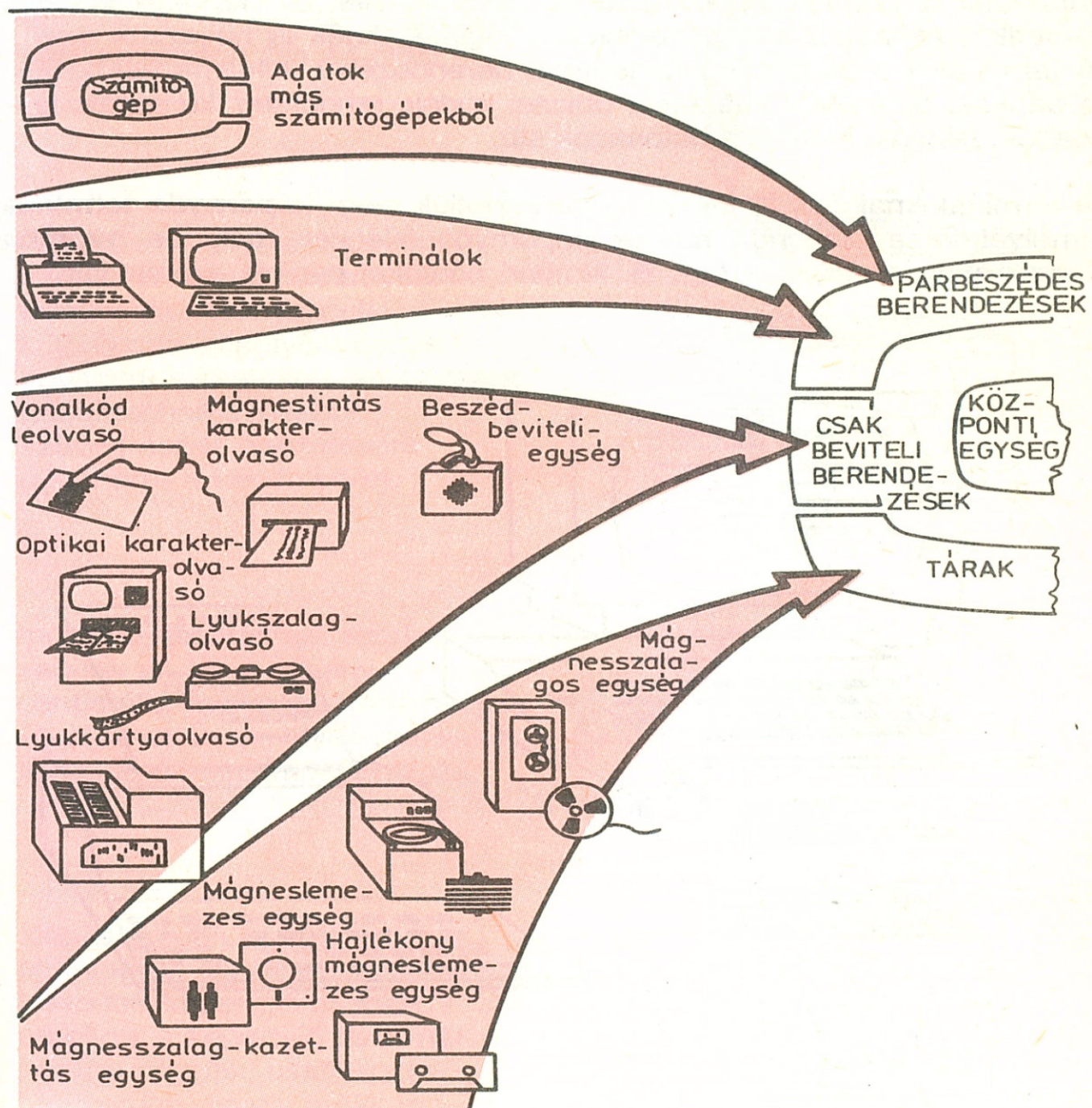
A piacon ma sok olyan berendezés van forgalomban, amelyekkel *adatok vihetők be a számítógépbe*. Ezek egyrészt speciális beviteli egységek, amelyek kizárólag adatok bevitelét végzik. Ezekhez tartoznak a lyukkártyaolvasók, lyukszalagolvasók, beszédbeviteli egységek, kézírásolvasók, mágneses írás-olvasók, vonalkód-leolvasók.

Másrészt vannak olyan berendezések is, amelyekkel adatok bevihetők és tárolhatók is. Ilyenek pl. a képernyőterminálok, nyomtatóterminálok, mágneslemezes egységek, mágnesszalagos egységek, hajlékony mágneslemezes egységek (diskettek), mágnesszalagos kazetták.

Az adatbevitelre még más lehetőségek is vannak. Egy számítógép közvetlenül is átvihet adatokat, pl. akkor, ha mindkét számítógép ugyanahhoz a számítógép-hálózathoz tartozik.

A következő ábra áttekintést ad a bevitel legfontosabb lehetőségeiről.

Példa:



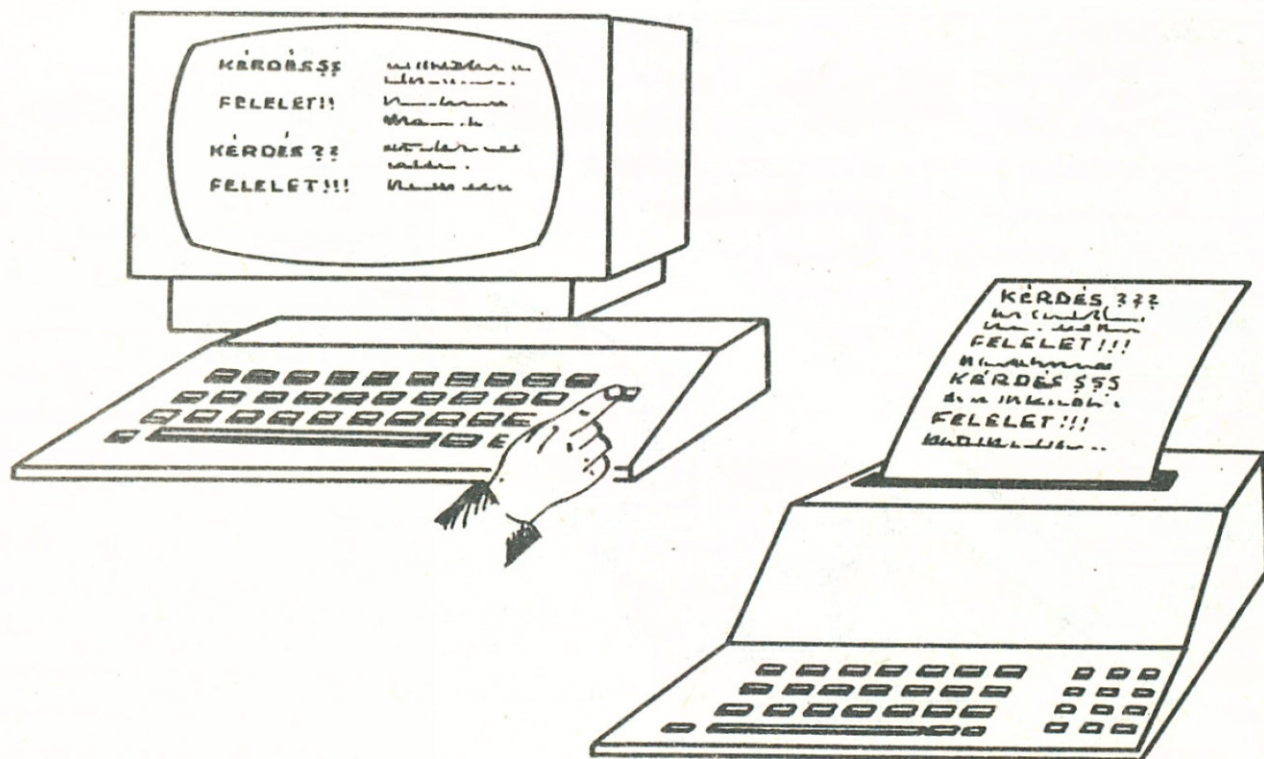
A következőkben röviden ismertetjük az egyes beviteli egységeket.

Terminálok

A **terminálok** olyan berendezések, amelyek segítségével párbeszédet folytathatunk a számítógéppel, azaz az adatbevitel és -kivitel váltakozva követik egymást. Ezért **párbeszédés** egységeknek is nevezzük őket.

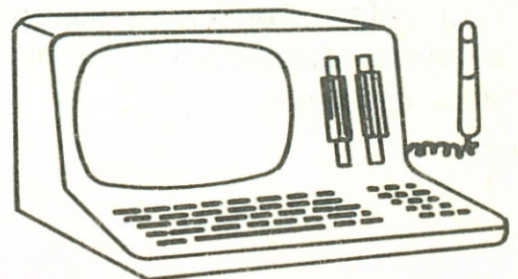
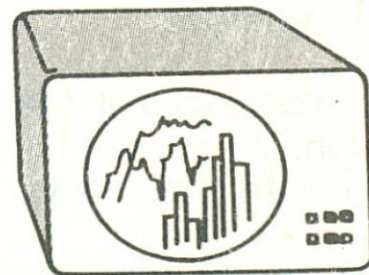
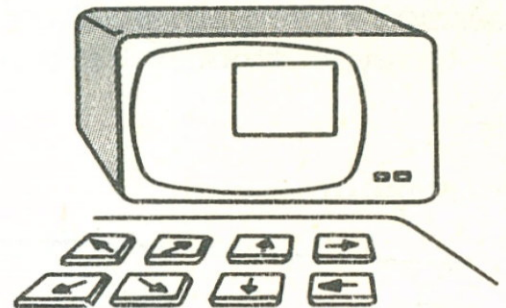
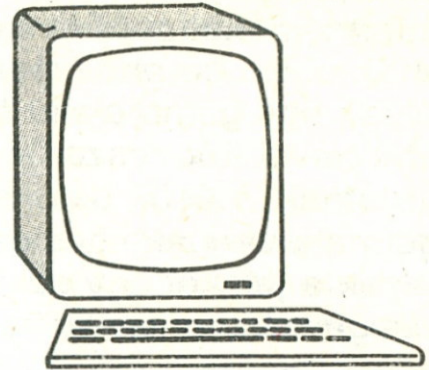
A terminálok a bevitel legelterjedtebb berendezései. Alkalmazási helyük pl. a bankok, biztosítási intézetek, utazási irodák, rendőrség, kórházak, könyvkiadók, iskolák, különféle hatóságok stb.

A termináloknak két fő fajtáját különböztetjük meg: **képernyős terminálok**, amelyeknél a beviendő adatok képernyőn jelennek meg, és **nyomtatós terminálok**, amelyek a be-, ill. kimenő adatokat papírra nyomtatják.



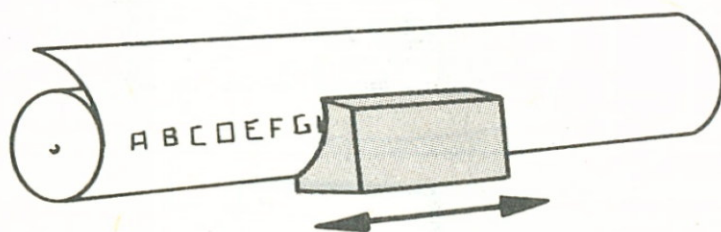
A képernyős terminálok osztályozhatók:

- a megjelenítési formátum szerint vannak pl. képernyők, amelyekre egy nyomtatott Din A4 oldal összes karaktere ráfér,
- használati kényelem szerint: vannak pl. olyan képernyők, terminálok, amelyeknek különleges billentyűi vannak egyszerű szövegkorrekció céljára (ablaktechnika további szövegrészek beiktatására ugyanarra a képernyőre),
- műszaki kialakításuk szerint: vannak pl. csak néhány centiméter mélységű képernyők is, és olyanok is, amelyek grafikákat több színben tudnak ábrázolni,
- kiépítettségük szerint: pl. integrált diszkettállomások, beépített képernyőmásolók, amelyek a képernyő tartalmát papírra rögzítik, csatlakoztató fényceruza, amely grafikus bevitelt tesz lehetővé azáltal, hogy a fényceruzát a képernyő bizonyos helyére vezetjük és ott gombnyomással aktivizáljuk. Ezzel a módszerrel pl. a megadott lehetőségek egyike kiválasztható („menütechnika”).



A **nyomtatós terminálok** lényegében nyomtatási technikájukban különböznek egymástól, amellyel az információkat papírra nyomják. Vannak olyan berendezések, amelyek egy gömb alakú fej segítségével írógépszerűen nyomtatnak, mások betűkerékkel dolgoznak, vannak olyanok, amelyek a jeleket egy színes tintasugárral „spriccelik” a papírra, mások hőre érzékeny papírra égetik be a jeleket, és ezért nagyon csendesen dolgoznak.

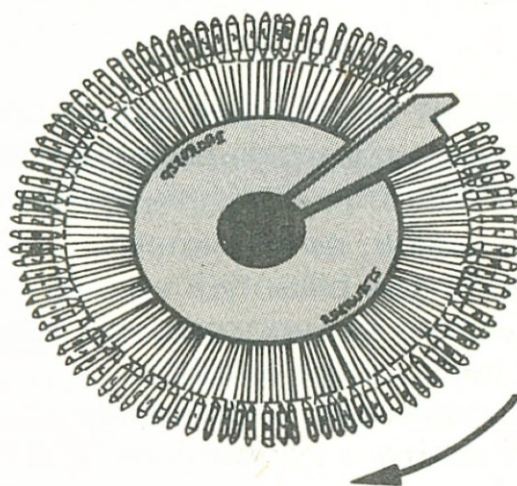
Hőnyomtató



Gömbfej



Betűkerék



A nyomtatós terminálok alkalmazásának előnye, hogy a lefolytatott párbeszéd papíron, *jegyzőkönyvszerűen* rögzítve van.

A képernyős terminálok ezzel szemben a bevitt szövegek *egyszerű korrekcióját* teszik lehetővé.

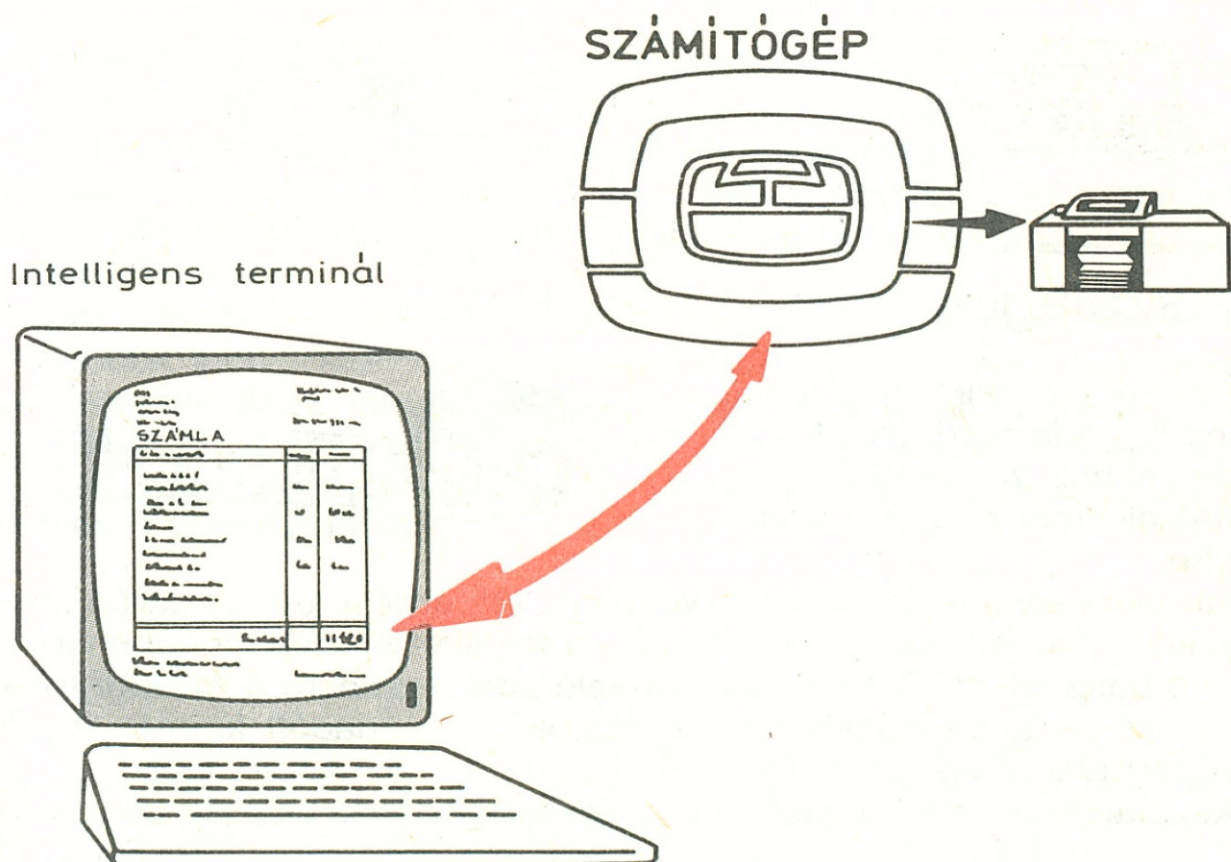
A terminálok mindkét fajtája esetén az adatok a begépelés után először a képernyőn, ill. a papíron jelennek meg, és használójuk csak a megfelelő ellenőrzés után továbbítja azokat egy billentyű lenyomásával a számítógépbe.

Tehát az adatok **rögzítése** egy terminál billentyűzetén történő begépeléssel alig választható el a tulajdonképpeni **adatbeviteltől**. Az adatbevitel e módja esetén természetesen feleslegessé válik egyéb adatrögzítő berendezések alkalmazása.

Gyakran használunk **intelligens terminálokat** is, amelyek — mivel proceszoruk és adattáruk is van — a számítógéppel való összeköttetés nélkül is képesek adatfeldolgozásra. Az intelligens terminálok tulajdonképpen már számítógépeknek tekinthetők, amelyek mind terminálként, mind pedig önálló számítógépként saját programmal üzemeltethetők.

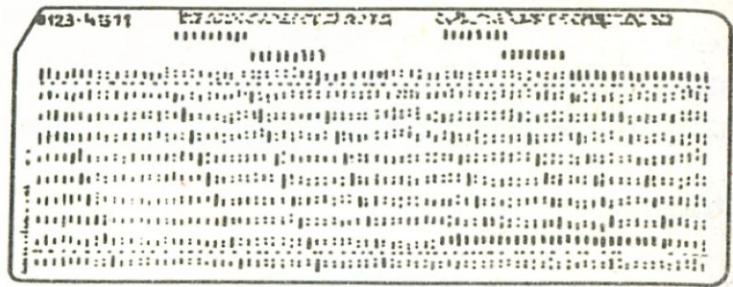
Példa:

Egy számlát kell kiállítani. Az intelligens terminál képernyőjén egy űrlapot, pl. egy számlanyomtatványt jelentettünk meg. Ebbe az űrlapba visszük be az adatokat az ügyfelek számláinak megszerkesztésére. A terminálban lefutó program az egyes tételeket összeadja, így megkapja a számlaösszeget, miközben pl. az értéktöbbletadót is kiszámítja. Csak a kész számlának a képernyőn való megjelenése után kerül át a hozzákapcsolt számítógépbe, amely az összes számlákat összegyűjti, és azokat naponta egyszer egy gyorsnyomtatóval kinyomtatja.



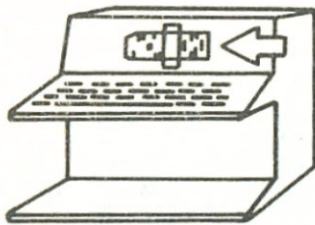
Lyukkártyaolvasók

Ma már elég ritkán alkalmaznak lyukkártyákat mint gép által olvasható adathordozókat, mert ezeknek van néhány hátránya a korszerű mágnesréteges adathordozókkal — pl. a mágneslemezekkel és mágnesszalagokkal — szemben. A kártyánkénti 80 karakter információsűrűség nagyon kicsi, az egyszer rávitt (lyukasztott) adatokat többé nem lehet korrigálni, és a lyukkártyakötegek tárolása sok helyet igényel. Ezenkívül a kártyák lyukasztása a speciális rögzítőberendezéssel, a **kártyalyukasztóval**, sok időt vesz igénybe és a bevitel a **lyukkártyaolvasóval** a számítógép belső feldolgozási sebességéhez viszonyítva rendkívül lassú.

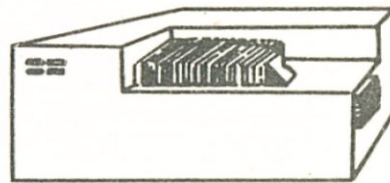


Lyukkártya

Rögzítőberendezés: kártyalyukasztó

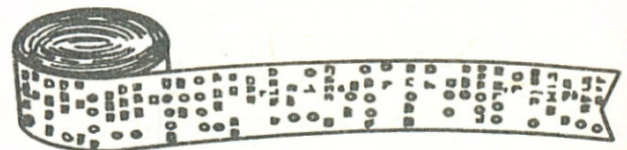


Beviteli berendezés: lyukkártyaolvasó



Lyukszalagolvasók

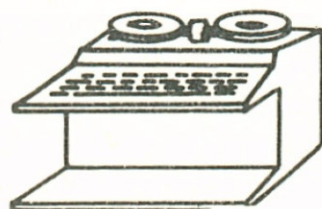
Lyukszalagokat mint gép által olvasható adathordozókat, a lyukkártyákhoz hasonlóan, alig használnak ma már az adatbevitelre.



Ma már csak elvétve találkozunk velük, pl. ott, ahol a távirás során „melléktermékként” lyukszalagok keletkeznek, amelyeket a számítógépbe való adatbevitelre használunk. A lyukszalagra való adatrögzítésre a **szalaglyukasztó** szolgál, amely a papírszalagon az adatokat a megadott kódnak megfelelő lyukasztással ábrázolja.

A lyukszalagon rögzített adatokat a **lyukszalagolvasó** olvassa be a számítógépbe.

Rögzítőberendezés:
szalaglyukasztó



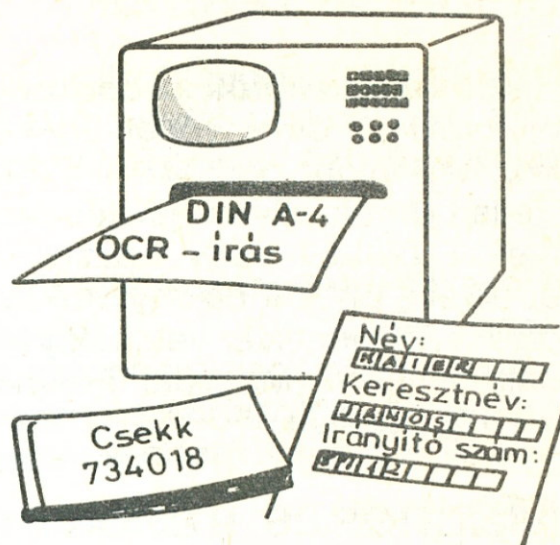
Beviteli berendezés:
lyukszalagolvasó



Optikai karakterolvasók

Az **optikai karakterolvasó** egyes típusai különösen az írógéppel írt vagy nyomtatott hosszabb szövegek bevitelére alkalmasak. Gyakran azonban csak bizonyos szabványos írások olvashatók velük.

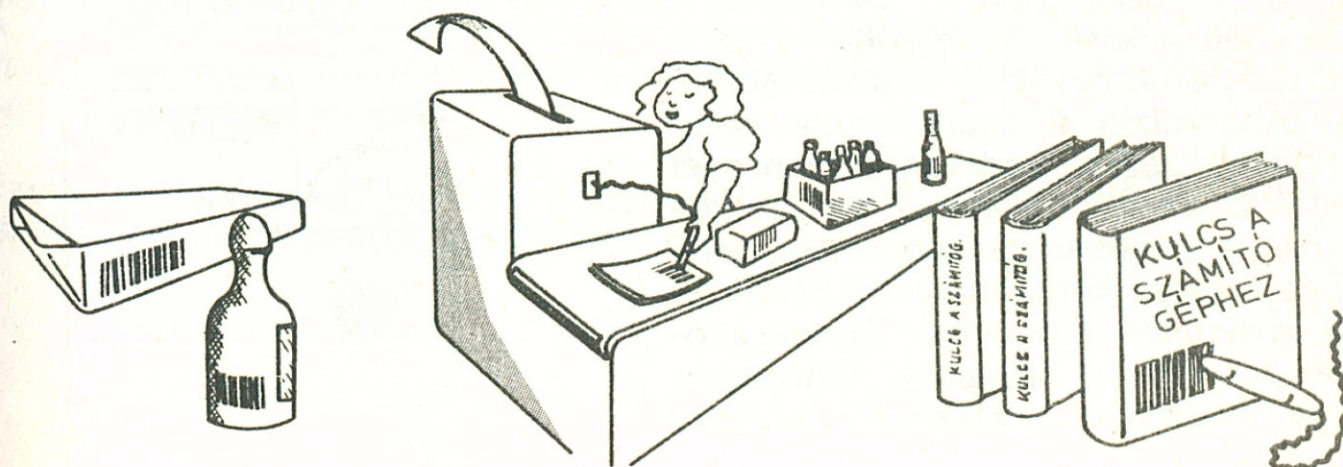
Más modellek ezzel szemben egyforma bizonylatok, pl. csekkürlapok vagy más adatlapok számítógépbe vitelére alkalmasak, amelyek esetleg részben gondos kézírással is kitölthetők.



Vonalkód-leolvasók

A vonalkód-leolvasókat főként a kereskedelemben használják, ahol sok különböző árucikket a vételártól függetlenül kell tudni azonosítani, és a cikkszámokat az eladásnál egyszerűen és gyorsan meg kell tudni állapítani. A cikkszámhoz tartozó aktuális ár és a cikk megnevezése a bevitel után azonnal kiolvasható a számítógép tárából, és a vevő részére kinyomtatható vagy megjeleníthető.

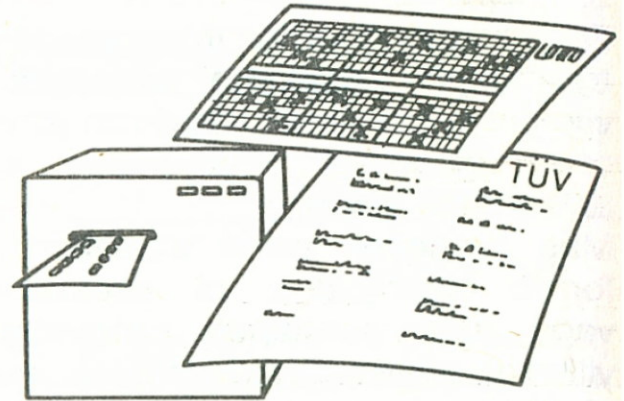
A vonalkód-leolvasókat a könyvtárakban a könyvek kiadásánál is használják. A vonalkódok leolvasására és bevitelére különféle technikájú berendezések vannak használatban: lényegében ezek vagy olyan készülékek, amelyeknél a vonalkódokat (pl. egy vonalkóddal megjelölt árucikket) egy leolvasólemezzre helyezünk, vagy olyan berendezések, amelyeknél egy leolvasóceruzát kell végighúzni a vonalkód fölött.



Jelölésérzékelők

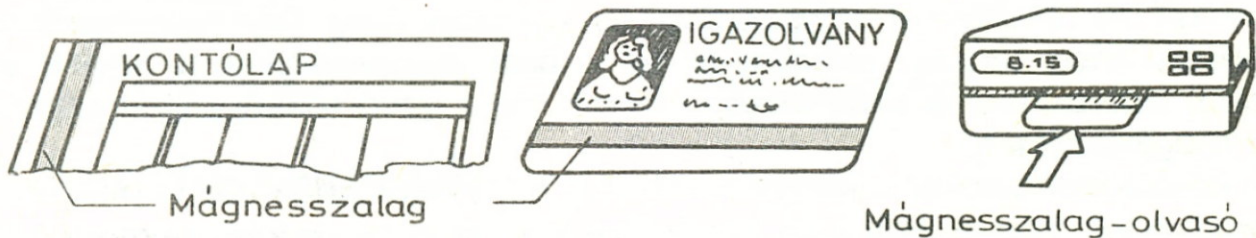
A **jelölésérzékelők** az optikai karakter-olvasókhoz és a vonalkód-leolvasókhoz hasonlóan optikai elven működnek. Felismernek olyan jelöléseket, amelyeket géppel vagy kézzel vittek fel speciális optikai bizonylatokra. Példák erre: a lottószelvények, vagy az NSZK-ban a TÜV (Műszaki Felülvizsgálat) hiányjegyzékei.

Jelölésérzékelő



Mágneses írásolvasók

Ma már sok bizonylatra és okmányra mágnescsíkot is tesznek, hogy az ember által olvasható információk mellett még gép által olvasható adatokat is lehessen rajtuk tárolni. Példák erre: mágneses kontókártyák, az igazolványok minden fajtája, kölcsönlapok stb. Az így tárolt adatok bevitelére különféle típusú **mágneses írásolvasók** vannak használatban.



Beszédbeviteli egységek

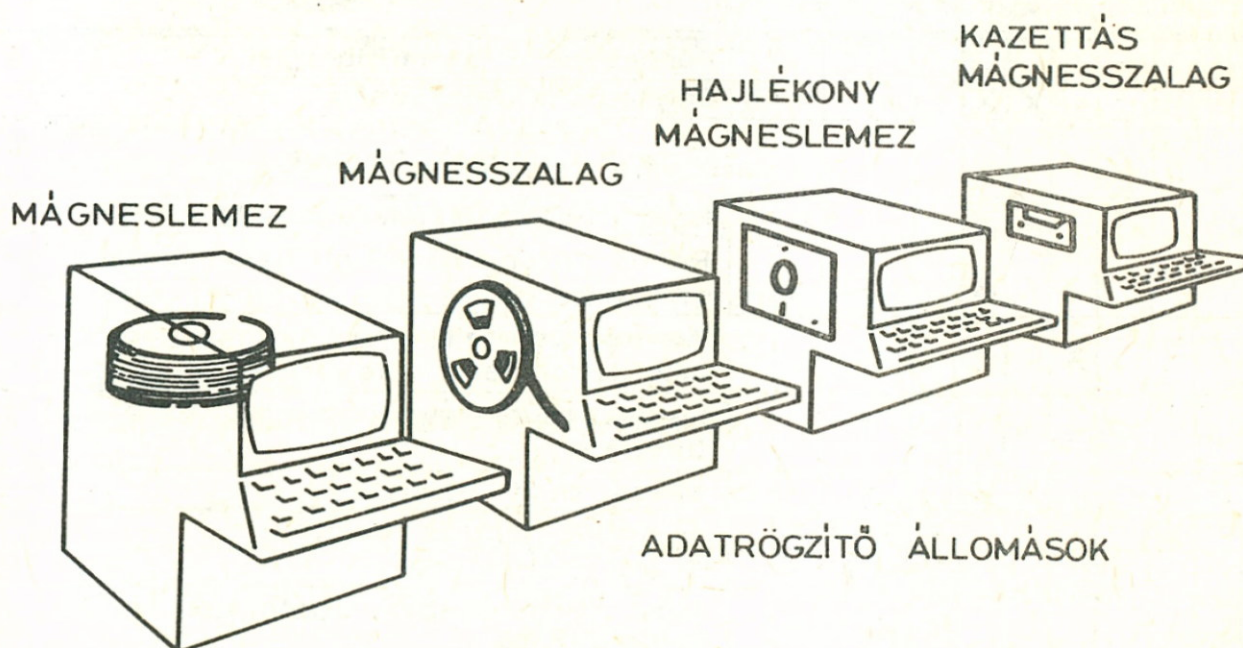
Mindenhol, ahol a készülékek kezelésében járatlan személyeknek kell adatokat bevinni, vagy ahol a kezelőnek mindkét kezét igénybe veszi munkája (pl. a futószalagnál), a **beszédbevitel** kínálkozik megfelelő eszközként. A kereskedelemben a következő rendszerű berendezésekkel találkozhatunk: beszédanalízis a személyazonosság megállapítására, korlátolt szókinccsel működő beszédfelismerő készülékek (számok bevitele beszéd által), nagy teljesítményű beszédbeviteli berendezés lexikai analízissel és nagyobb szókinccsel (pl. információk lehívására nagy adatbázis-rendszerek esetén).



Bevitel tárokon keresztül

A **tárok** nagy mennyiségű adatot tudnak nagy sebességgel a számítógép központi egységétől átvenni, ill. igény esetén a kivitt adatok újra beolvashatók a központi egységbe. Ezért a legtöbb tár eredeti feladata, az adattárolás mellett, *adatok bevitelére* is alkalmas.

Ehhez az adathordozókra (mágneslemez, mágnesszalag, kazetta és hajlékony mágneslemez) speciális berendezésekkel rögzítjük az adatokat. Ilyen adattörzítő berendezések: a mágneslemezes, a mágnesszalagos, a mágnesszalag-kazettás vagy a hajlékony mágneslemezes adattörzítők. Az adatok rögzítése azonban a számítógépben futó program eredményeként is végbe-mehet.



A bevitel a mindenkori adathordozónak megfelelően *mágnesszalagos, mágneslemezes, mágnesszalag-kazettás* vagy *hajlékony mágneslemezes berendezésekkel* történik.

Ezeket a tárokat egy következő fejezetben — amelyben a tárolás funkcióját ismertetjük — tárgyaljuk részletesebben.

A beviteli berendezések leírása után befejezésül áttekintést adunk az egyes berendezések legfontosabb tulajdonságairól, amelyből látható, hogy azok legjobban milyen célra alkalmasak.

BEVITELI BERENDEZÉSEK										Különösen alkalmas: ha ...	
Képernyőterminal	Nyomtatóterminal	Optikai karakter	Lyukkártyaolvasó	Lyukszalagolvasó	Mágnesolvasó	Jelölésolvasó	Mágneslemezolvasó	Hajlékony mágneslemezolvasó	Közvetlen mágneslemezolvasó		Beszédbeviteli berendezés
X	X										Párbeszéd a számítógéppel
				X	X				X		Egyszerű kezelhetőség
		X									Gépell szövegek bevitele
			X	X	X						Egyes árucikkszámok bevitele
					X						Rövid információk pl. igazolványok bevitele
			X								Adatok bevitele lyukszalagra, amelyeket telex közvetített
		X									Lyukkártyán lévő adatok bevitele
X	X			X	X				X		Kevés adat bevitele
		X				X	X	X	X		Tömeges adatok bevitele
						X	X	X	X		Gyors adatbevétel
						X	X	X	X		Más számítógépek adatainak további feldolgozása
X	X							X	X		Jutányos beszerzési árú berendezések
											esetén
											A berendezés alkalmas:
		X	X	X	X	X			X		Kizárólag bevételre
X	X										Kivételre is (párbeszédés berendezések)
						X	X	X	X		Tárolásra és kivételre



1. Nevezze meg azon beviteli egységeket, amelyek adatok be- és kiviteléhez használhatók!

.....

.....

.....

2. Tételezzük fel, hogy a következő fajta adatokat kell a számítógépbe bevinnie, és a bevitelre a következőkben felsorolt berendezések állnak rendelkezésére:

Mely adatokat milyen berendezéssel vinne be?

Írja be a megfelelő betűket a téglalapokba!

Árucikkszámok vonalkóddal: A.

Adatok, amelyek távirókódban érkeztek: B.

50 szabvány írásban írt gépírási oldal: C.

Árucikk-lyukkártyák: D.

Adatok, amelyek egy feldolgozás következtében már névlegesen fel vannak jegyezve: E.

Lottószelvény: F.

Beszélt mondatok hanganalíziséhez: G.

Jelölésérzékelő.

Lyukkártyaolvasó.

Optikai karakterolvasó.

Lyukszalagolvasó.

Vonalkód-leolvasó.

Tár.

Beszédbeviteli egység.

3. A következő beviteli egységek közül melyeknél kell a beviendő adatokat először gép által olvasható adathordozón rögzíteni?

Nyomtató terminál.

Lyukszalagolvasók.

Lyukkártyaolvasó.

Optikai karakterolvasó.

Hajlékony mágneslemezes
beviteli egység.

Beszédbeviteli
egység.

Mágnesszalag-kazettás
egység.

Képernyős
terminál.



Válaszok

1. A következő beviteli egységek használhatók adatok be- és kivitelére:

Nyomtatóterminálok, képernyős terminálok, mágnesszalagos egységek, mágneslemezes egységek, hajlékony mágneslemezes egységek, mágnesszalag-kazettás egységek, de egy másik számítógép is lehet beviteli berendezés, pl. egy számítógép-hálózat esetén.

2.

 F

Jelölésérzékelő.

 D

Lyukkártyaolvasó.

 C

Optikai karakterolvasó.

 B

Lyukszalagolvasó.

 A

Vonalkód-leolvasó.

 E

Tár.

 G

Beszédbeviteli egység.

3. A következő beviteli egységeknél kell a beviendő adatokat előbb gép által olvasható adathordozón rögzíteni:

Lyukszalagolvasó.

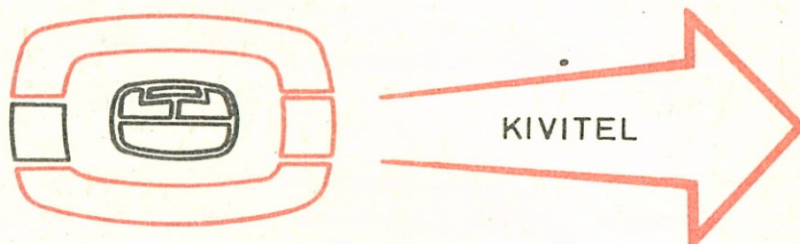
Lyukkártyaolvasó.

Hajlékony mágneslemezes beviteli egység.

Mágnesszalag-kazettás egység.

A kivitel

Azoknak a berendezéseknek a piaci választéka, amelyek az adatfeldolgozás eredményeinek kiadására használatosak, ugyanolyan nagy, mint a beviteli berendezéseké.

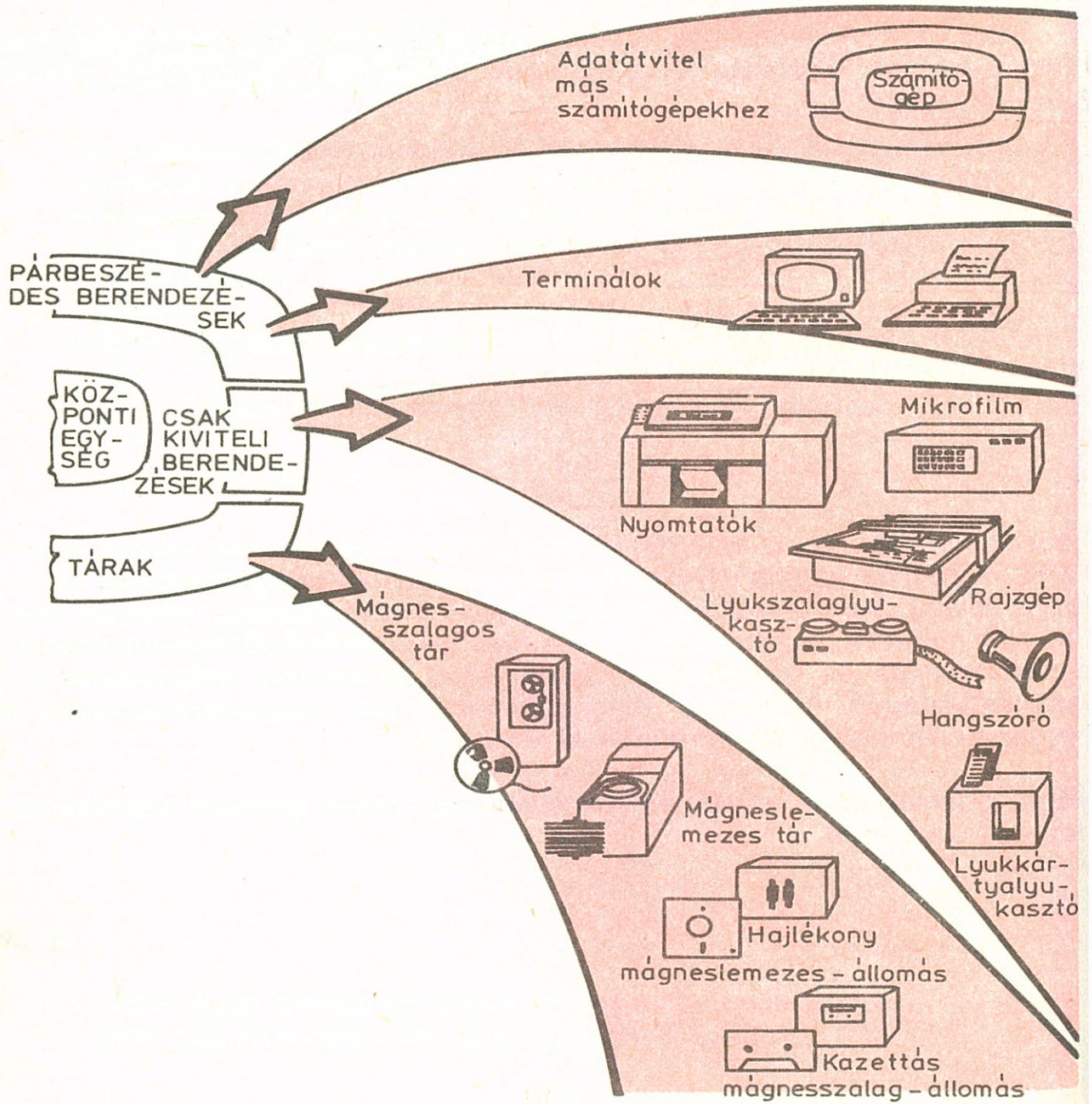


Vannak berendezések, amelyek *kizárólag* *kivitelre* alkalmasak, mint pl. nyomtatók, rajzgépek (plotter), lyukkártyalyukasztók, lyukszalaglyukasztók, mikrofilmre kivivő berendezések, továbbá olyan berendezések, amelyek más funkciók mellett az *adatok kivitelének* funkcióját is betöltik. Ezekhez tartoznak: a párbeszédés és tárolóberendezések, tehát terminálok, mágneslemezes, mágnesszalagos, hajlékony mágneslemezes és mágnesszalag-kazettás egységek, amelyek által az adatok ki is vihetők.

Végül lehetséges az adatoknak közvetlenül egy *másik kapcsolt számítógéphez* való kivitele is.

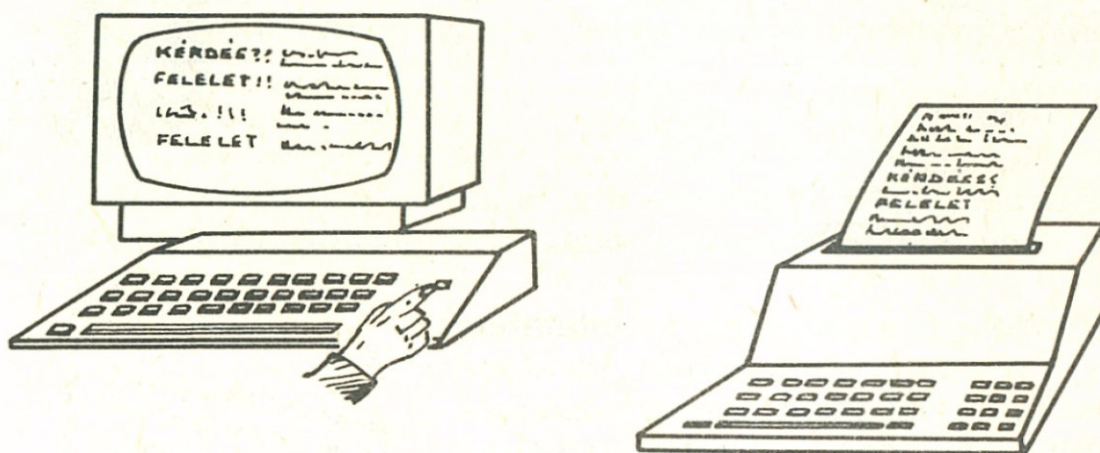


A kiviteli egységek részletes leírása előtt előbb nézzük az ábrát:



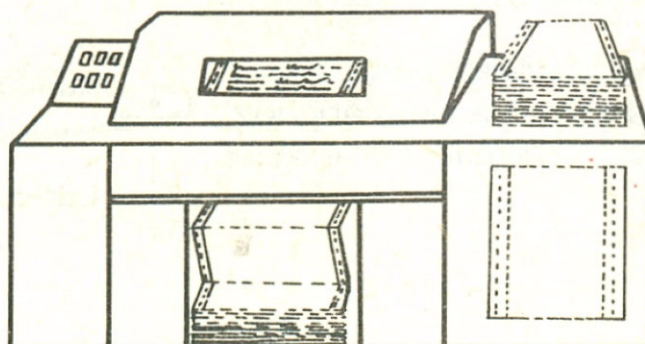
Terminálok

A **terminálok** párbeszédese berendezésekként már a beviteli berendezéseknél leírtuk. Adatkivitel esetén a képernyős terminálok előnye, hogy használatukkal hosszabb szövegrészek lapozhatók át, azaz a képernyőn megjelenő szövegeket egymás után végig lehet olvasni. Ez különösen akkor előnyös, ha egy bizonyos szövegrészt keresünk. Ezáltal nem kell az egész szöveget kinyomtatni, ami nagy időmegtakarítással jár.



Nyomtatók

A **nyomtató** egyike a leggyakrabban használt kiviteli egységnek. A gyártók a legkülönbözőbb modelleket kínálják, amelyeknek nyomástechnikája és teljesítőképessége a legkülönfélébb.



A nyomtatókat aszerint osztályozhatjuk, hogy munkamenetenként hány karaktert tudnak a papírra felvinni.

Vannak modellek, amelyek minden nyomtatandó karaktert bizonyos számú raszterpontból állítanak össze. Ezeket **mátrixnyomtatóknak** nevezzük.

Más nyomtatók egyenként teljes karaktereket nyomnak, hasonlóan az írógépekhez, egy gömbfej vagy egy betűkerék segítségével. Ezeket **karakternyomtatóknak** hívjuk.

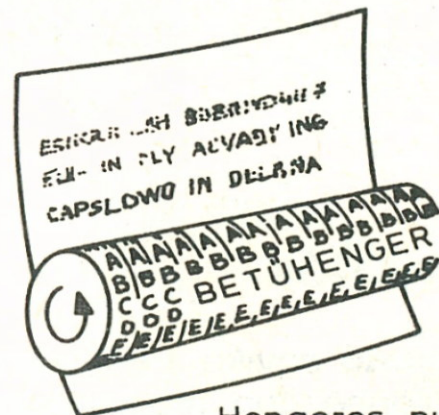
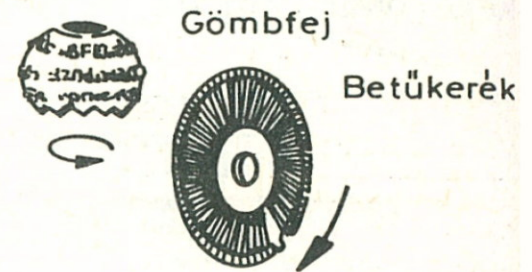
A **sornyomtatók** ezzel szemben teljes nyomtatott sorokat tudnak egy munkamenetben nyomni. Eközben a papír többnyire egy betűhengerhez vagy betűlánchoz nyomódik.

A **betűhengerek** egy sor minden nyomtatandó pozícióján minden nyomtatható karaktert tartalmaznak. A 132 jeles sorszélességnek tehát megfelel összesen: 132-szer az összes nyomtatható betű, szám és különleges karakter. A betűhenger állandóan nagy sebességgel forog. Minden nyomtatási pozícióhoz egy nyomtatókalapács tartozik, amely pontosan abban a pillanatban nyomja a papírt a hengerhez, amikor a nyomtatandó karakter a forgó hengeren vele szemben megjelenik.

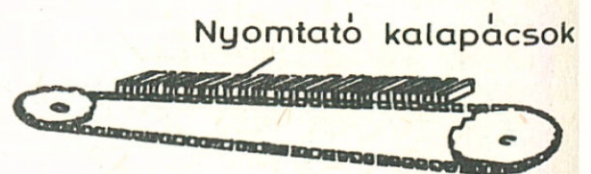
Ehhez hasonlóan működnek a **láncos nyomtatók** is.

Itt is többnyire 132 nyomtatókalapács abban a pillanatban nyomja a papírt a betűlánchoz, amikor a nyomtatandó karakterek előttük „elhaladnak”.

Mátrixnyomtató

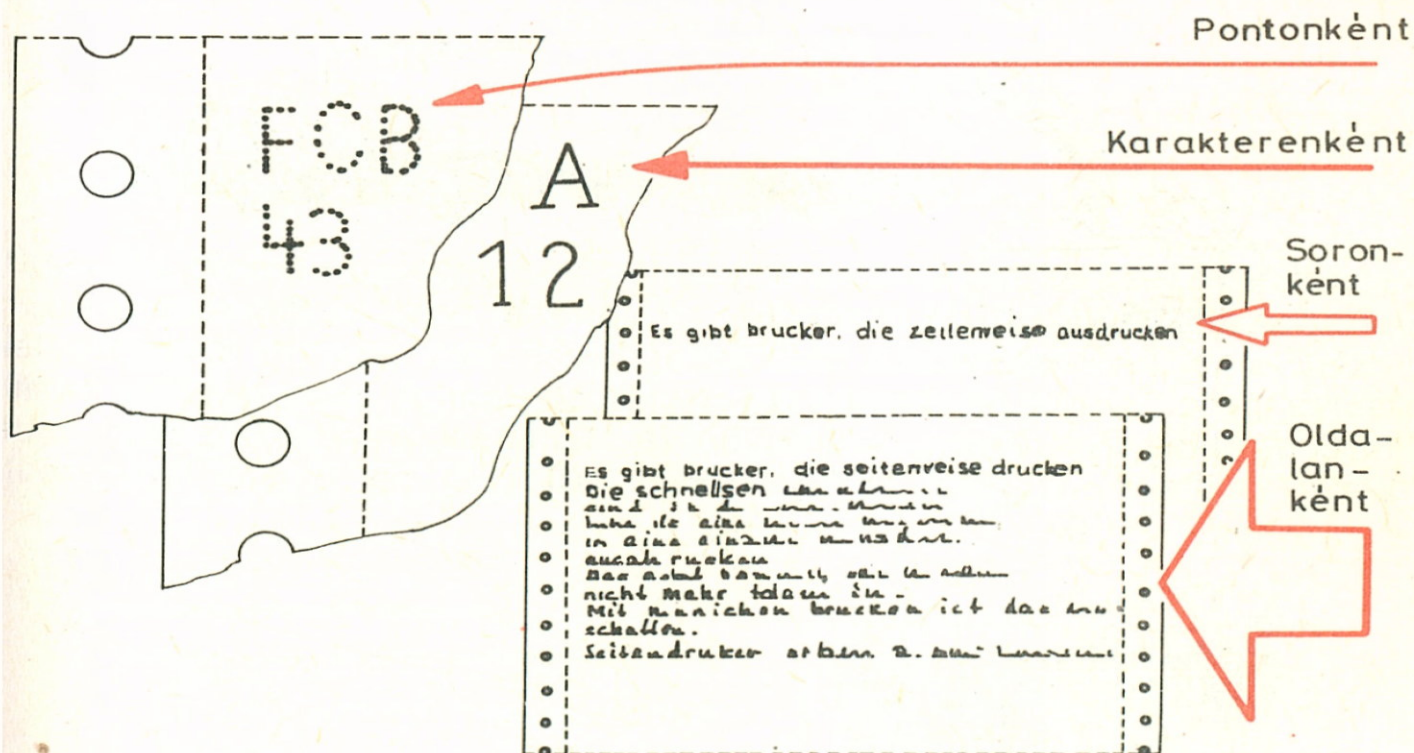
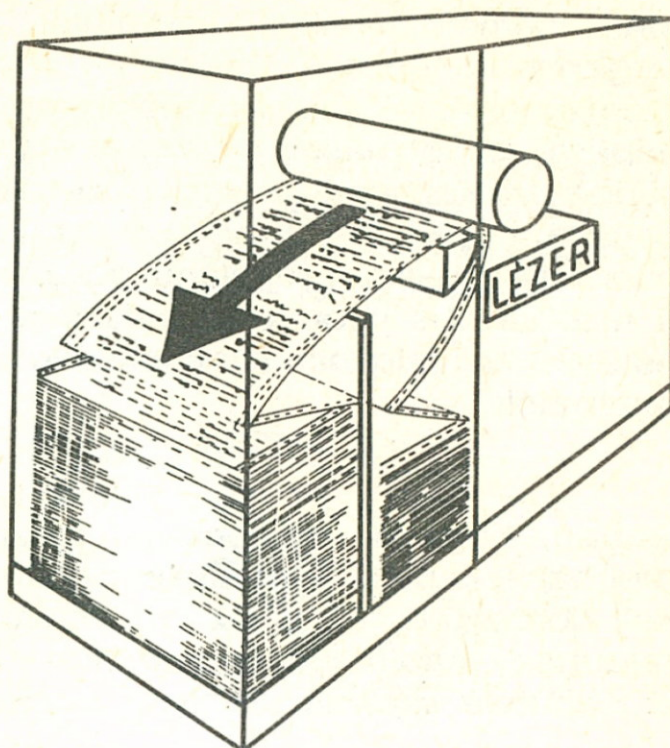


Hengeres nyomtató



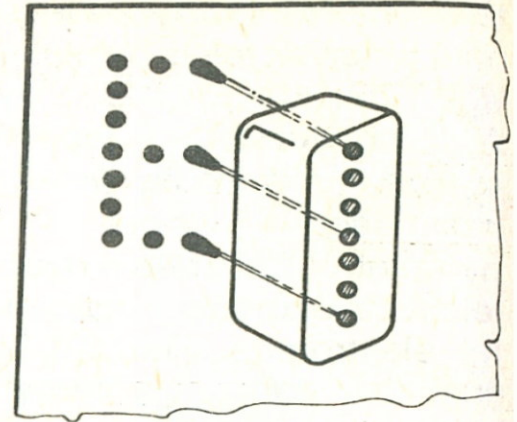
Láncos nyomtató

A mátrix-, karakter- és sornyomtatókon kívül **lapnyomtatók** is vannak, amelyek teljes oldalakat tudnak egy munkamenetben nyomtatni. A leggyorsabb lapnyomtatók: a **lézernyomtatók**, óránként egymillió sornál többet is képesek nyomtatni! Eközben a nyomtatandó jeleket a lézersugár pontonként viszi fel elektromos töltéssel egy fényérzékeny dobra úgy, hogy az oda-juttatott színezőpor pontonként odatapad. Ezt követőleg a dobról papírra másolódik át a szöveg. A lapnyomtatók az eddig bemutatott berendezésekkel ellentétben a **nem-mechanikus nyomtatókhöz** tartoznak.

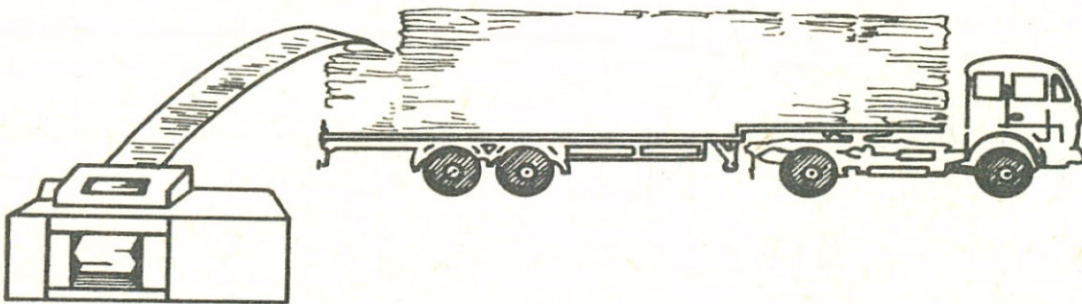


Más nemmechanikus nyomtatók **festék-sugárral** dolgoznak, amely — mint pl. a mátrixnyomtatók — tintapontokat szór a papírra. Ismét mások **termikus alapon** működnek, miközben a karaktereket hőérzékeny papírra égetik be. Egyes nemmechanikus nyomtatók teljesen zajtalanul dolgoznak, míg a mechanikus nyomtatók zajszintje magas, és ezért hivatalokban csak feltételesen alkalmazhatók.

Tintas nyomtató

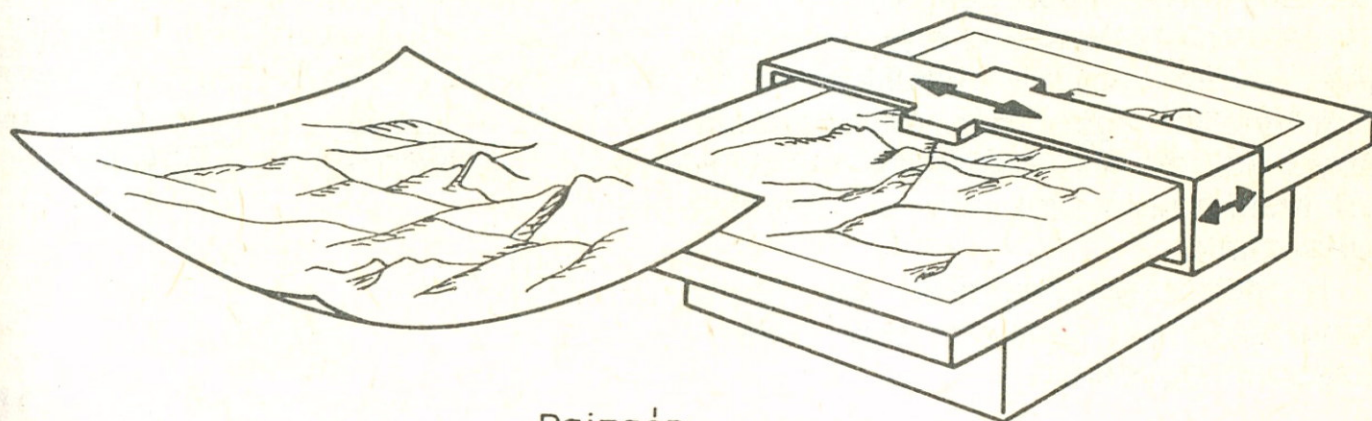


Abban, hogy a lassabban dolgozó mechanikus vagy a gyorsabb nemmechanikus nyomtatót alkalmazzuk-e, a gazdaságosság szempontjai szerint kell döntenünk. Pl. a lézernyomtatót — amely naponta egy tehergépkocsi-rakomány papírt tud nyomtatni — csak akkor alkalmazzuk ésszerűen, ha a különféle információk ilyen mennyiségére rövid időn belül szükség van.



Rajzgépek

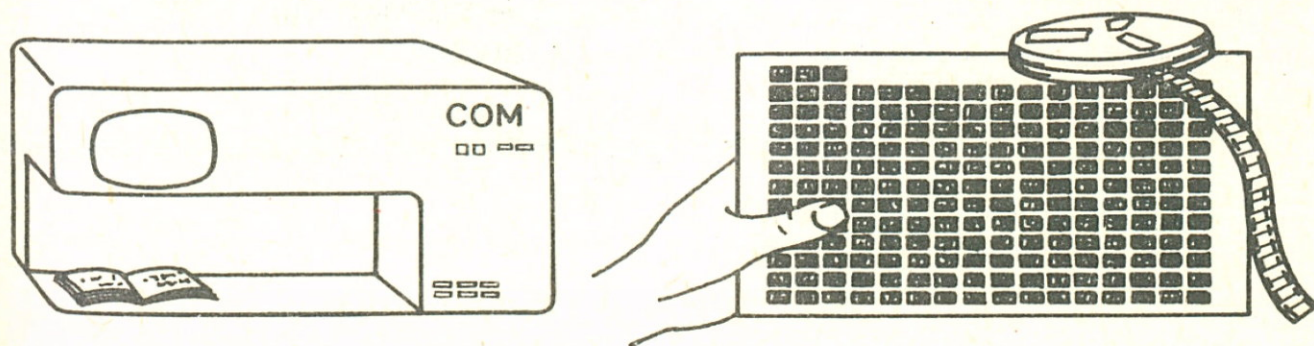
A rajzgépek speciális berendezések, amelyet akkor alkalmazunk, ha *grafikus formájú kimenő* adatra van szükségünk, pl. rajzok, tervek, görbék és diagramok alakjában. A mechanikus rajzgépek esetén egy vagy több (részben több színű) íróhegyet vezetnek a papíron. Az elektronikus rajzgépek esetén a grafika egy képcső segítségével kerül mikrofilmre, amelyet ezután felnagyítanak.



Rajzgép

Mikrofilmes adatkiviteli egységek

Speciális kiviteli egységek segítségével a számítógép kimenő adatait közvetlenül mikrofilmre lehet felvenni. Ez különösen kis helyigényű archívumok létesítésénél előnyös, mert feleslegessé teszi a bizonylatok utólagos mikrofilmre való felvételét. A kiviendő adatok egy képcsővön jelennek meg, ezután foto úton mikrofilmre kerülnek. A mikrofilmes adatkivitel, amelyet COM (computer output on microfilm) mozaikszóval jelölünk, rendkívül gyors.

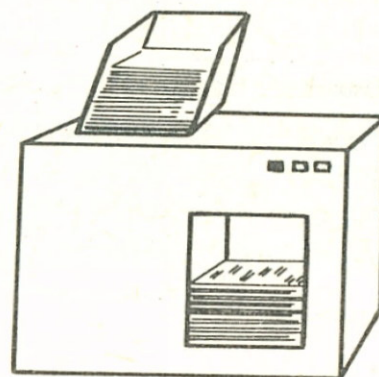


Lyukkártyalyukasztók

A **lyukkártyalyukasztók** a kiviendő adatokat elektronikus vezérléssel lyukkombinációk formájában lyukasztják a lyukkártyákba. A lyukkártyát és a lyukkártya-feldolgozó berendezéseket azonban a fejlett hardvertechnika már majdnem teljesen kiszorította. A lyukkártyalyukasztókat pl. már csak ott használják, ahol az árucikkekhez előállításuknál lyukkártyákat mellékelnek, amelyek mind ember által, mind pedig gép által olvasható információkat tartalmaznak, és amelyeket az árucikkek eladásakor csak a számítógép lyukkártyaolvasójába kell bevinni.

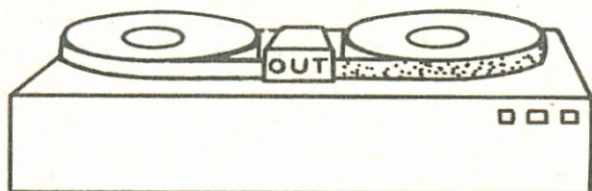
A lyukkártyalyukasztók egy csoportja másolni is tudja a lyukkártyákat, amennyiben egy, már kilyukasztott kártya adatait átveszik, és a mindenkor lyukkombinációt egy új kártyába lyukasztják.

A lyukkártyalyukasztókat és lyukkártyaolvasókat gyakran egy berendezésben egyesítik.



Lyukszalaglyukasztók

A **lyukszalaglyukasztók** egy számítógép központi egységéből érkező kimenő adatokat lyukkombinációkként közvetlenül egy lyukszalagra lyukasztják. A lyukszalagokat a tárolt adatok további feldolgozása végett ismét egy másik számítógépbe olvashatjuk be.

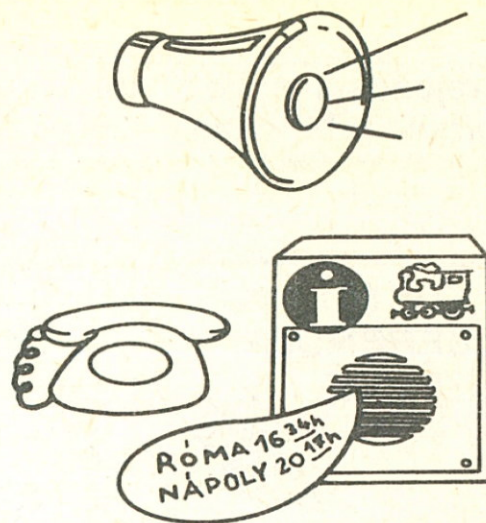


Ha az információkat távirókodekban lyukasztjuk, akkor azok minden táviró-berendezésen át — pl. több címzettnek — átvihetők.

Beszédkivitel

A **beszédkivitel** jelentősége egyre fokozódik. Itt a kiviendő adatokat bonyolult eljárásokkal hangrezgésekké alakítják át, és hangszórókon keresztül kisugározzák. Az akusztikai kivitelnek más eljárásokkal szembeni nagy előnye, hogy az adatfeldolgozásban járatlan laikus minden járulékos berendezés nélkül közérthető információkat kaphat a számítógéptől, pl. telefonon keresztül.

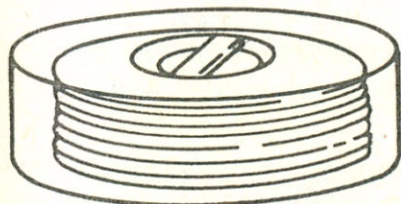
Már több beszédbeviteli és -kiviteli rendszer van használatban, pl. egy vonatmenetrend-tájékoztatói rendszer is, amely telefonérdeklődésre a kívánt vonatösszeköttetést megállapítja, és azt az utassal beszédkivittel közli.



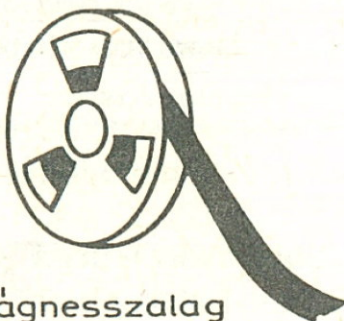
Kivitel táraakra

Táraakra akkor viszünk ki adatokat, ha a program eredményeit nem kell az ember részére láthatóvá tennünk, és a kiviteli adatokat egy másik számítógép általi további feldolgozás céljából készenlétben akarjuk tartani.

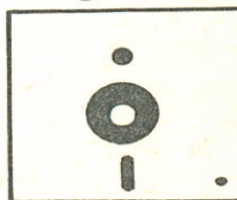
Mágneselem



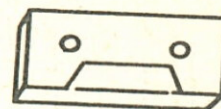
Mágnesszalag



Hajlékony
mágneselem



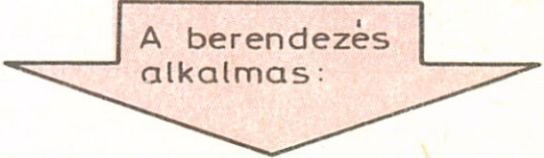
Kazettás
mágnesszalag



A táraakkal a következő fejezetben foglalkozunk.

Befejezésül a következő ábrán áttekintést adunk az egyes berendezések legfontosabb tulajdonságairól:

KIVITELI BERENDEZÉSEK										
Képernyős terminál										
Nyomtató terminál										
Rajzgep										
Akusztikus kiviteli										
Kiviteli mikrofilmre										
Magnelemezes tár										
Hajlékony-magnelemezes tár										
Kazettás magnesszalagos tár										
X	X									Párbeszédre
			X							Egyszerű kezeléshez
		X	X	X	X	X	X			Adathordozó manuális továbbítására
		X	X	X	X	X	X			Dokumentációra
	X	X	X							Papírra való kivitelnél
X	X	X	X							Rövid adatsorhoz
		X		X	X	X				Hosszú összefüggő adatsor esetén
		X		X	X	X	X			Tömeges adatokhoz
X	X						X	X		Jutányos beszerzési árú berendezésekhez
				X	X	X				Különösen gyors kivitelnél
				X						Helytakarékos raktározáshoz
					X	X	X	X		Továbbfeldolgozáshoz számítógépekben
		X	X	X	X					Kizárólag kivitelnél
X	X									Kivitelnél és bevitelnél
					X	X	X	X		Bevitelnél, kivitelnél és tárolásra





1. Feltételezzük, hogy a következő kiviteli egységek állnak rendelkezésünkre: karakternyomtató, lapnyomtató, képernyős terminál, rajzgép, mikrofilm-es kiviteli egység, lyukkártyalyukasztó, lyukszalaglyukasztó, beszédkiviteli berendezés, tár.

Melyik kiviteli egységet választja:

időjárás-térképek kinyomtatására:

valamely könyvtár

összes könyvcímeinek

katalogizálására:

a munkahelyen a számítógéptől

rövid információk ismételt

lekérdezésére:

ha a kiviteli adatokat további

feldolgozásra készenlétbe kell

helyezni:

egy feldolgozás eredményeinek

távíron, az egész világon való

szétküldésére:

árucikk-kártyák nyomtatására

és az áruhoz való mellékelésre:

2. Jelölje meg azokat a kiviteli berendezéseket, amelyek bevitelre is alkalmasak!

Képernyős terminál.

Nyomtató terminál.

Mágnesszalagos egység.

Mágneslemezes egység.

Hajlékony mágneslemezes (diskett) egység.

Mágnesszalag-kazettás egység.



Válaszok

1. A következő kiviteli egységeket kellene választani időjárás-térképek kinyomtatására: **rajzgépet,**

valamely könyvtár

összes könyvcímeinek

katalogizálására: **mikrofilmes kiviteli egységet,**

a munkahelyen a számítógéptől

rövid információk ismételt

lekérdezésére: **képernyős terminált,**

ha a kiviteli adatokat további

feldolgozásra készenlétbe kell

helyezni: **tárat,**

egy feldolgozás eredményeinek

távíróon az egész világon való

szétküldésére: **lyukszalaglyukasztót,**

árucikk-kártyák nyomtatására

és az áruhoz való mellékelésre: **lyukkártyalyukasztót.**

2. Valamennyi felsorolt kiviteli egység bevitelre is alkalmas:

Képernyős terminál.

Nyomtatós terminál.

Mágnesszalagos egység.

Mágneslemezes egység.

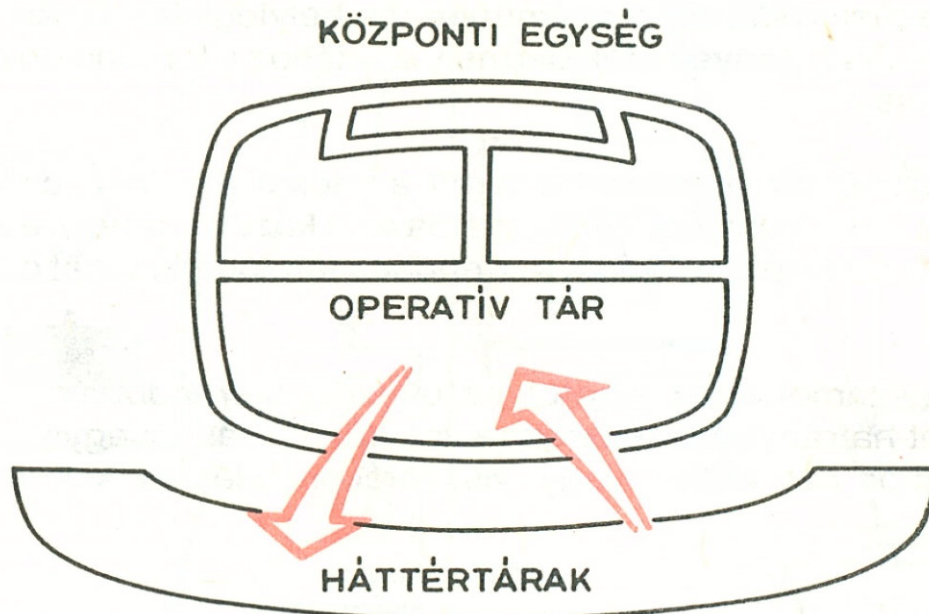
Hajlékony mágneslemezes (diszkett) egység.

Mágnesszalag-kazettás egység.

A tárolás

A **háttértárak** a számítógép központi egységével közvetlen kapcsolatban álló egységek, amelyek nagy mennyiségű adatot képesek a központi egységtől átvenni és azokat ugyanolyan gyorsan a központi egységnek átadni.

A háttértáraknak az a rendeltetése, hogy a központi egység tárát — az operatív tárát — tehermentesítsék azoktól a programoktól és adatoktól, amelyekre éppen nincs szükség, azokat azonban egy esetleges feldolgozás céljára készenlétben kell tartani. A háttértárakon tároljuk pl. az adatbázisok információit, amelyek az operatív tárban nem férnek el, de azoknak mégis állandóan rendelkezésére kell állniuk.



A tárok egy további funkcióját már megemlítettük: ezeket — ha cserélhető adathordozójuk van — gyakran *adatok bevitelére és kivitelére* is használják.

A táraikat általában az adatok **elérhetőségének módja** szerint osztályozzuk, azaz annak alapján, hogy mennyi idő alatt és milyen sorrendben lehet a tárolt adatokat kiolvasni.

Vannak olyan táraik, amelyeken a tárolt információk bármelyike közel azonos, rövid idő alatt elérhető. Ezek a **közvetlen hozzáférésű** táraik.

Egy általánosan ismert közvetlen hozzáférésű tár a hanglemezt, amelyre a tüt a lemez bármely pontjára ráhelyezhetjük.

Egy számítógépes feldolgozáshoz akkor van szükség közvetlen hozzáférésű tárra, ha a tárolt adatok nem dolgozhatók fel feljegyzésük sorrendjében.

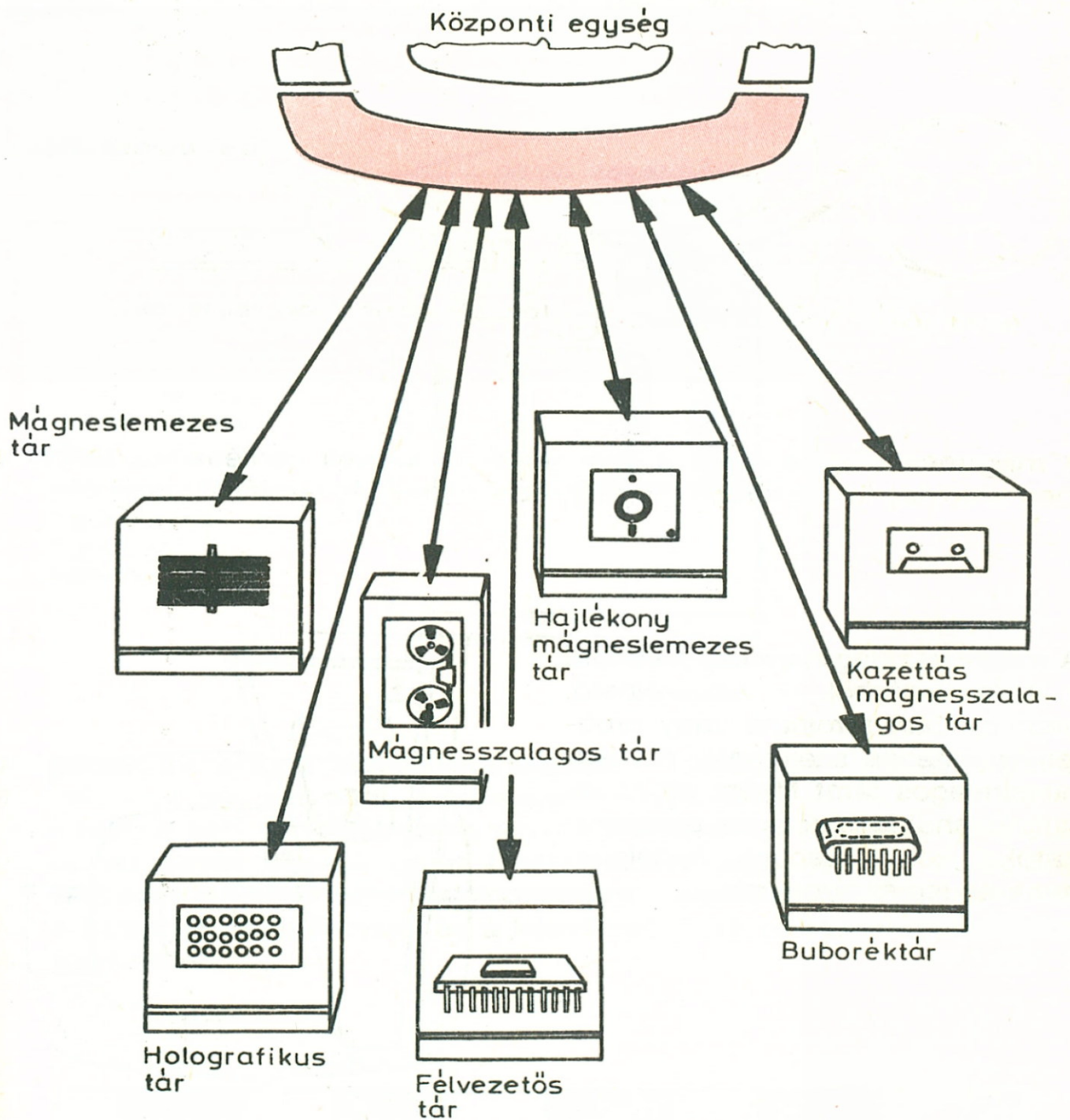
Példa:

Valamely utazási irodában a számítógépes helyfoglalás során az ügyintézőnek az utasok által igényelt utak bármely adatához a terminálon azonnal hozzá kell tudni férnie.

A **soros elérésű** tár esetében a tárolt adatokat azonban csak a felvételük sorrendjében lehet kiolvasni. Erre jó példa az a kazetta, amelyre zeneszámokat vettünk fel, és azok elejétől végig sorrendben tárolódtak, tehát csak így hallgathatók vissza.

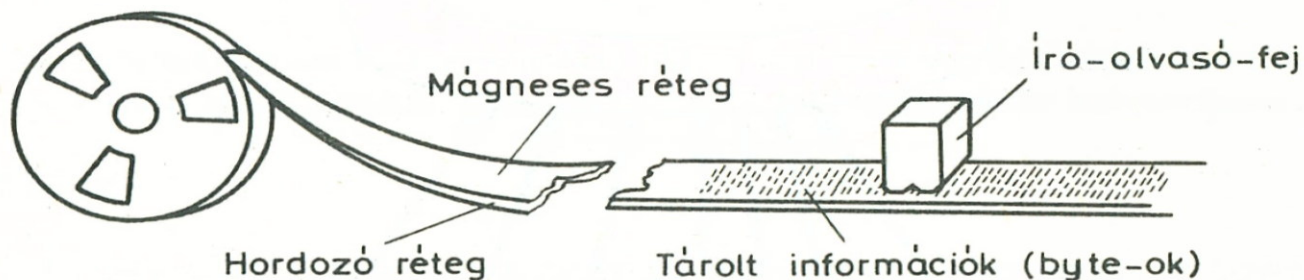
Amíg a zeneszámokat az elsőtől az utolsóig sorrendben akarjuk hallgatni, ez nem jelent hátrányt. Ha azonban ezek közül csak az egyiket akarjuk meghallgatni, akkor azt előre- vagy visszatekeréssel kell megkeresnünk.

A következőkben ismertetjük a legfontosabb adattárakat, előbb azonban tekintsük át ezeket egy ábrán.



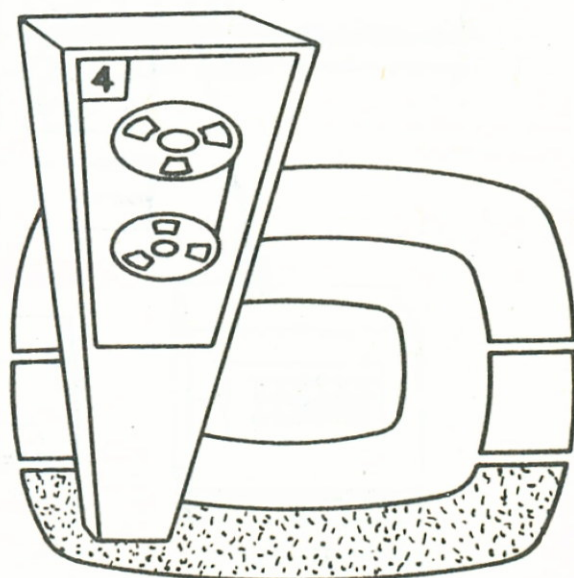
Mágnesszalagos tár

Egy mágnesszalagon az adatok szorosan helyezkednek el. Az információkat a szalagra felvitt vékony mágnesezhető rétegen tároljuk, és többnyire 9, egymás melletti mágnesfejjel ugyanannyi sávon olvassuk és írjuk azokat.

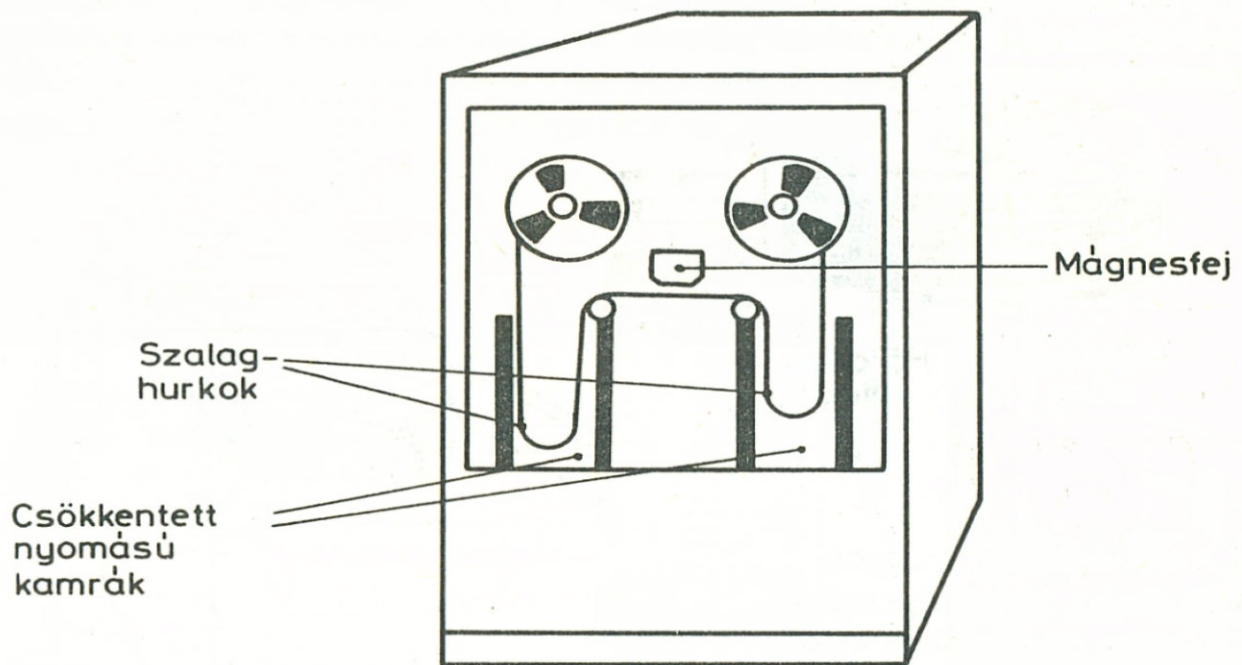


A **mágnesszalagos táruk** a számítógépek kedvelt perifériáihoz tartoznak, mert aránylag kevés költséggel nagy adatmennyiség tárolását teszik lehetővé.

A mágnesszalag bármikor kivehető a berendezésből, és kicserélhető, archívumban tárolható vagy problémamentesen szállítható. A mágnesszalagos tárat főként ott használjuk, ahol folytatólagos adatsorozatokot kell olvasnunk, feldolgoznunk és ismét tárolnunk.

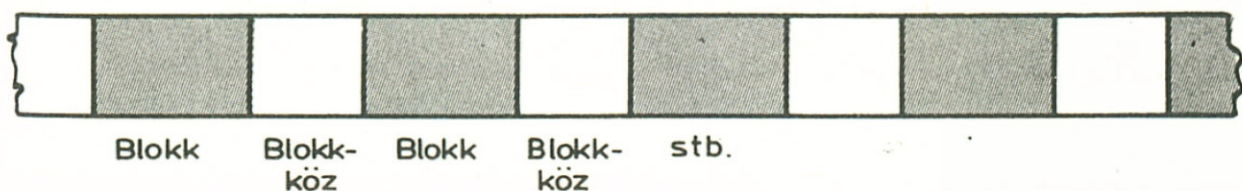


A mágnesszalagos egységek olyan felépítésűek, hogy lehetővé teszik a nagy sebességgel haladó szalag gyors megállítását és ismételt indítását. Ebből a célból a szalag nem közvetlenül halad orsóról orsóra, hanem közöttük szalag-előtárolókat képeznek ki, amelyekben a szalag hosszát elektronikus berendezés tapogatja le és vezérli ennek megfelelően a szalag meghajtását. A legkülönbözőbb berendezések léteznek, működési elvük azonban azonos.



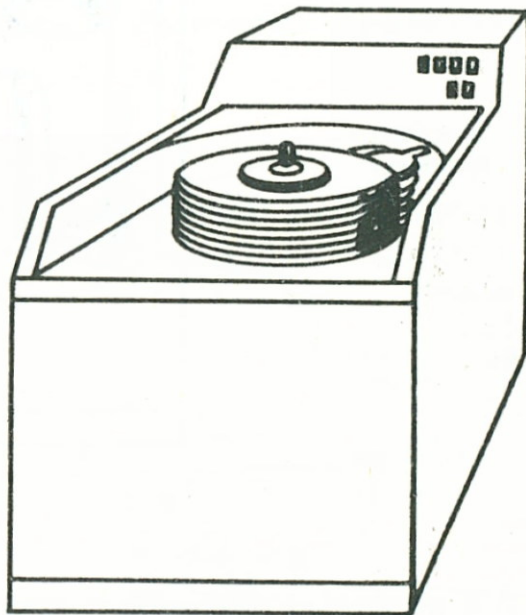
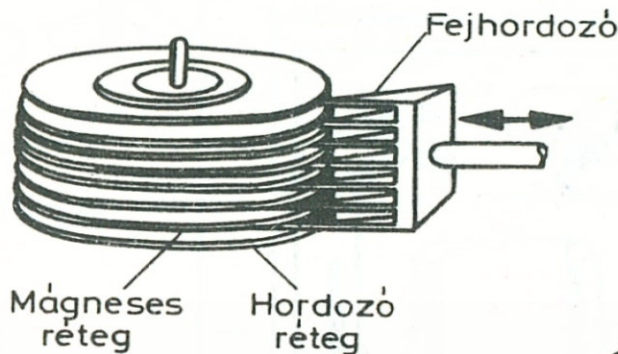
Az adatokat a szalagon nagyobb egységekbe — blokkokba — foglaljuk össze, amelyek — pl. egy cím alatt — egyenként olvashatók.

A legkisebb megcímezhető egység tehát a **blokk**, amely átlagosan 1000 byte hosszú. Az egyes blokkok között blokk-közöket hagyunk, hogy az egység a tárolt adatok blokkonkénti feldolgozásakor – a mindenkori blokk leolvasása után a szalagot megállíthassa és a következő blokk olvasásához azt ismét felgyorsíthassa.



Mágneslemezes tár

A **mágneslemezes tár** adathordozója többnyire több lemezből áll, amelyeknek mágneses rétege információkat tud tárolni. Több, egymás fölött elhelyezett lemezt *lemezkötegnek* nevezünk. A mágneslemezek felületei több pályára vannak beosztva, amelyek lemezoldalanként egy író-, ill. olvasófejjel írhatók vagy olvashatók. A pályánkénti cím segítségével minden pálya adatai közvetlenül hozzáférhetők, a mágneslemezes tár tehát *közvetlen hozzáférésű*.



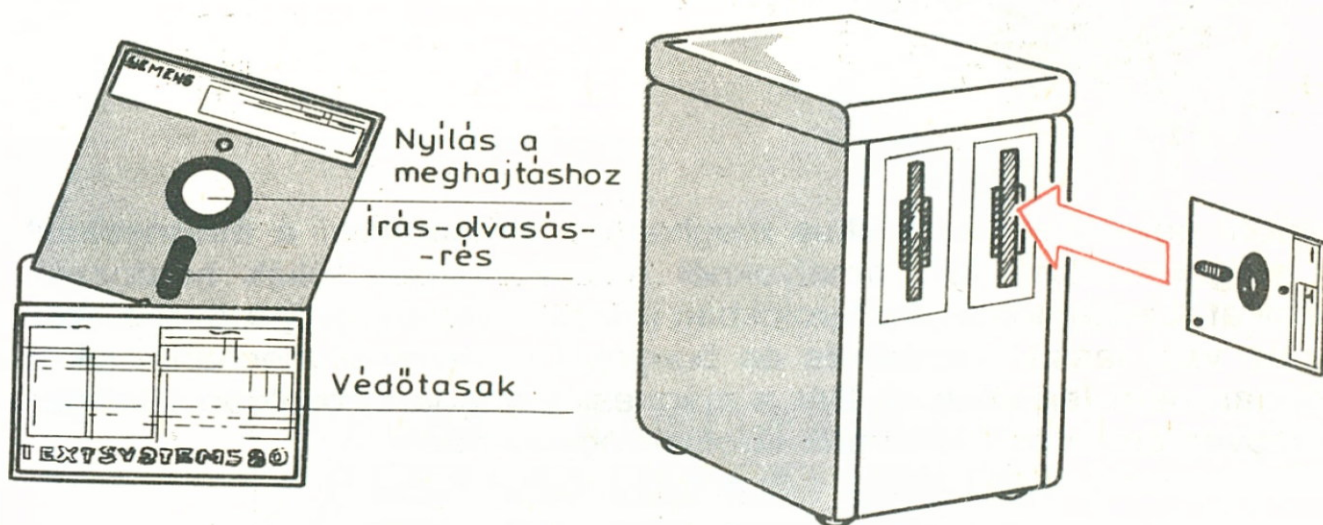
Mivel a lemezköteg nagyon gyorsan forog, azért egy pálya írására vagy olvasására igen rövid idő szükséges.

Vannak rögzített és cserélhető lemezkötegű mágneslemezes táruk. Nagy kapacitásuk és az adatok gyors hozzáférhetősége miatt főként a közepes és nagyobb számítógépekhez terjedtek el.

Hajlékony mágneslemezes tár

A hajlékony **mágneslemezes tár** (floppy disk) a mágneslemezes táruk különleges típusa. Ez egy hajlékony kör alakú lemez, amely a modelltől függően egyik vagy mindkét oldalán mágnesezhető réteggel van bevonva. A lemezt plasztik burok védi, amelyet a be-, ill. kiviteli egységbe — a hajlékony mágneslemezes tárba — történő behelyezéskor nem távolítunk el.

Bár a hajlékony mágneslemezes tár kapacitása és átviteli sebessége a mágneslemezes tárhoz képest lényegesen kisebb, az könnyen kezelhető, és nem utolsósorban olcsóbb, ezért főként kisebb adatfeldolgozó berendezésekhez, szövegautomatákhoz és az adatrögzítéshez terjedt el.

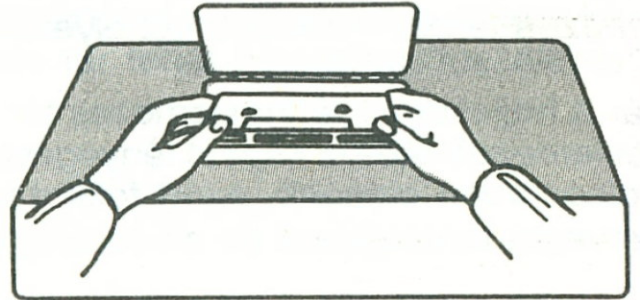


Térfogatszükséglete kicsi, szerkezete az íróasztal alsó részében könnyen elhelyezhető, és a helyiség klímájával, valamint a karbantartással kapcsolatosan nem különösen igényes.

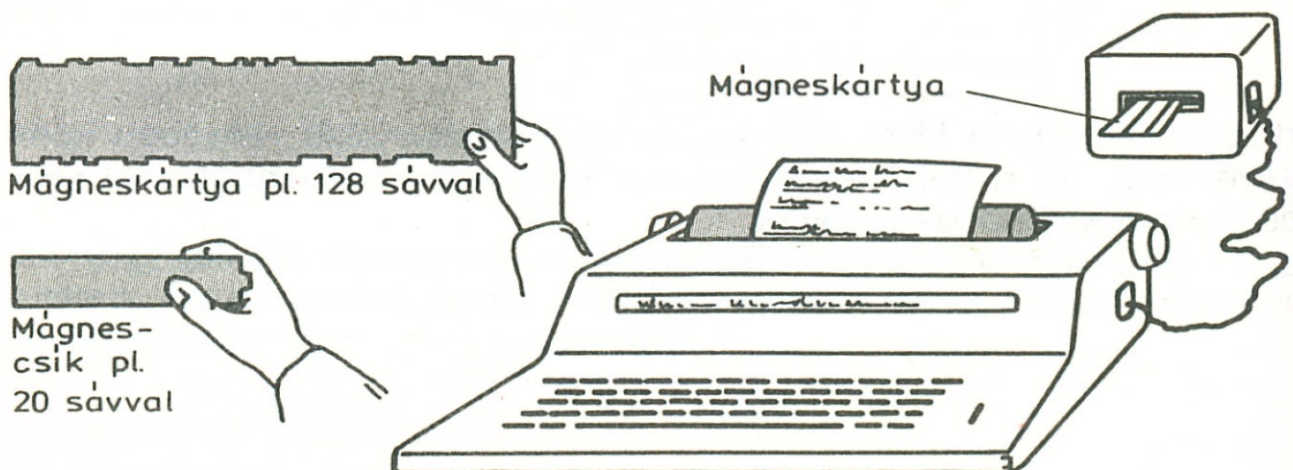
Mágnesszalag-kazettás tár

A **mágnesszalag-kazettás tár** — néha mágnesszalag-kazettás állomásnak is nevezik — adathordozóként a mágnesszalag különös formáját, a szóráskoztatóiparból jól ismert mágnesszalag-kazettát alkalmazza.

Viszonylag olcsó, ezért főként kisebb számítógépekhez és az adatrögzítéshez terjedt el. Pl. sok könyvelő-automata van mágnesszalag-kazettával felszerelve.



További példák a mechanikus meghajtású adattárokra: a **mágneskártyás** és **mágnescsíkos tár**, amelyeknél mágnesréteges fóliák hordozzák az adatokat. Ezeket többnyire tárolókban (magazinokban) gyűjtik össze. Az adatokhoz való lassú hozzáférés és bonyolult bevezetési mechanikájuk miatt azonban nem terjedtek el. Bár a mágneskártyás tárat gyakran alkalmazzák pl. szövegtárolásra a különféle tárolós írógépekhez.

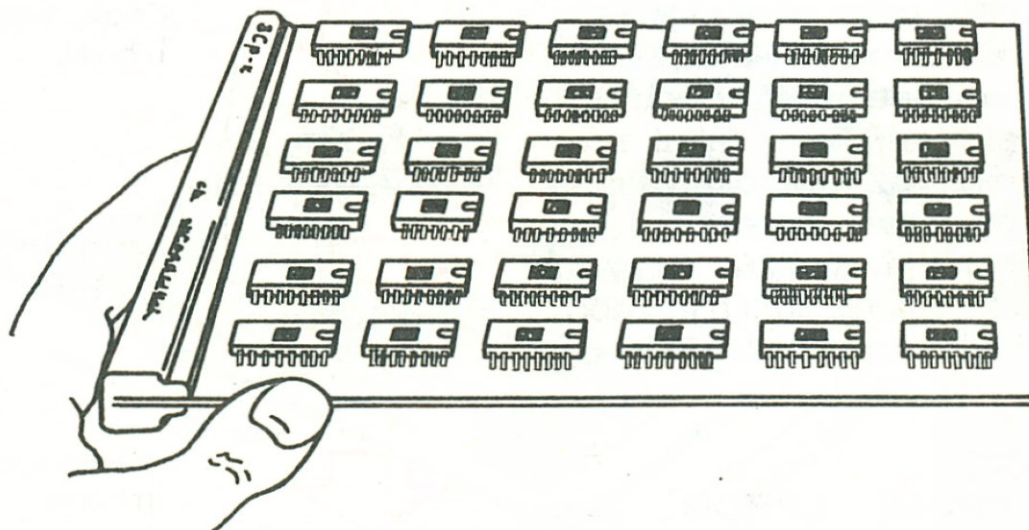
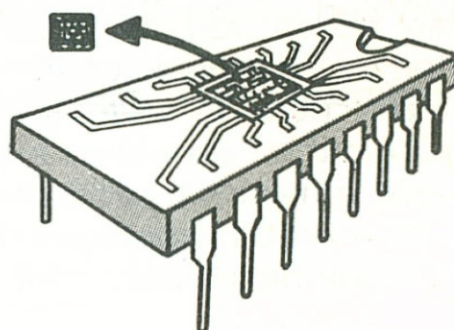


A mechanikus tárokon kívül vannak *elektronikus* alapon dolgozó tárok is, tehát mozgó alkatrészek nélküliek, amelyek ennek következtében rövidebb idejű hozzáféréssel működnek.

Félvezetős tár

A **félvezetős tár** *nemmechanikus tár*. A félvezetők általános működésével és információtároló képességükkel később foglalkozunk. A legfontosabb táruk felsorolásakor azonban a félvezetős tár nem maradhat ki, ezért röviden itt is foglalkozunk velük.

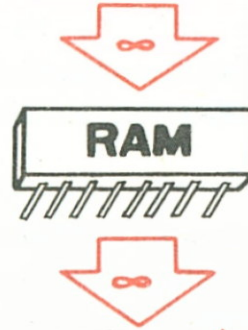
Egyetlen 5 cm-nél rövidebb félvezető építőelem százezernél is több byte-ot tud tárolni. Az építőelemeket nyomtatott áramkörü kártyákra helyezik, és azokból a szükségletnek megfelelően nagy táruk alakíthatók ki.



A számítógépek *operatív táruk* többnyire félvezetős táruk, mert közvetlen és rövid idejű hozzáférésűeknek kell lenniük.

A félvezető táruk típusai:

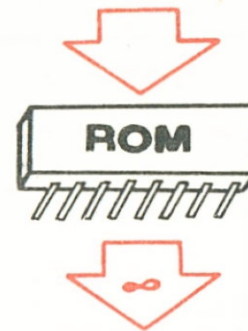
— **Írható—olvasható táruk, RAM (Random Access Memory)**, amelyeknél minden tárolóhely címének megadásával tetszőlegesen sokszor olvasható, törölhető vagy beírható. Áramkimaradás esetén az információk elvesznek.



Tetszőlegesen sokszor írható

Tetszőlegesen sokszor olvasható

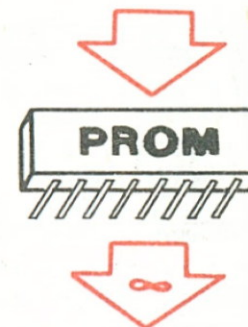
— **Csak olvasható tár, ROM (Read Only Memory)**, amelynek tartalmát a gyártó az ügyfél kívánsága szerint az előállítás során rögzíti. Áramkimaradás esetén az információk nem vesznek el, de azok nem is változtathatók meg. A ROM-okban azokat az információkat tároljuk, amelyekre ismételten szükség van, pl. a zsebszámológép programjait. A ROM-ok olcsóbbak, mint a RAM-ok.



Tartalmát a gyártó írja be

Csak olvasható tetszőlegesen sokszor

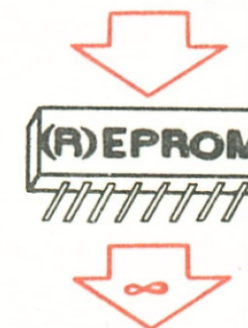
— **Programozható, csak olvasható tár, PROM (Programmable ROM)**. Ez olyan ROM, amely egy speciális programozó berendezéssel írható be. Ezáltal a felhasználó azokat az adatokat, amelyekre egy speciális feladathoz szüksége van, tárolhatja, anélkül, hogy az információkat ki kellene adni a kezéből. A PROM-ok drágábbak, mint a ROM-ok.



Csak egyszer írható

Tetszőlegesen sokszor olvasható

— **Törölhető PROM, EPROM vagy REEPROM (Erasable PROM vagy Reprogrammable ROM)**, amely vagy ultraviola fényvel, vagy elektromos jelekkel törölhető*, és egy speciális berendezéssel többször újraírható.



Néhányszor írható

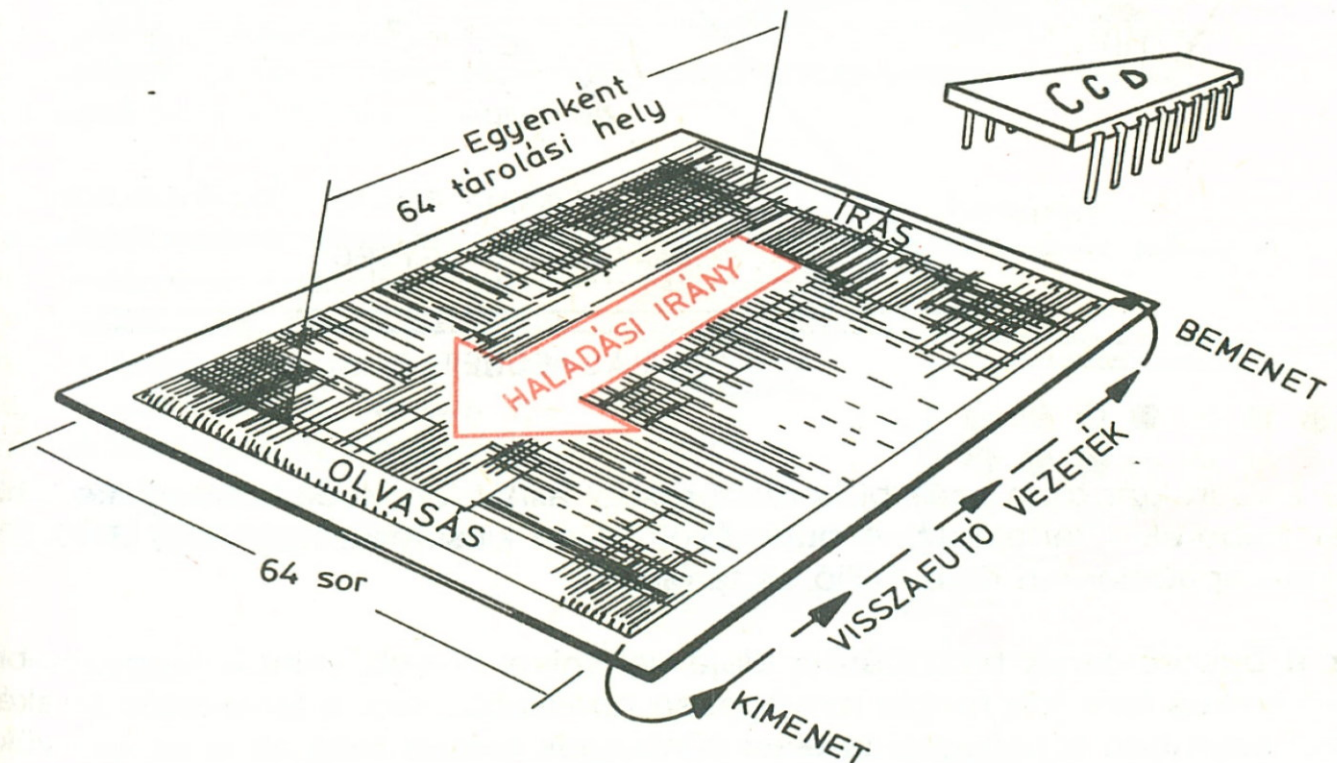
Tetszőlegesen sokszor olvasható

A RAM, ROM, PROM vagy EPROM típusú félvezetős táruk *közvetlen hozzáférésűek*.

Vannak azonban *soros hozzáférésű* félvezetős táruk is, amelyek félvezető-áramkörökből állnak, működésük az elektromos töltések lépésről lépésre való továbbításán alapul. Ezeket **töltéscsatolt eszközöknek** nevezzük: **CCD** (Charge Coupled Devices).

Ezekben az áramkörökben a tárolt bitek állandóan egy zárt körben mozognak és elhaladnak az *írasi* és *olvasási helyek* előtt. Beírásuk a bemenetnél történik, a beírt adatok azután egyenletes ütemben az egyik tárolási helyről a következőhöz továbbítódnak, amíg saját soruk végét eléri. Csak akkor olvashatók vagy írhatók át. Azonban, ha az adatoknak tovább is tárolva kell maradniuk, akkor a visszafutó vezetéken keresztül az adatokat a belépési helyen ismét be kell írni.

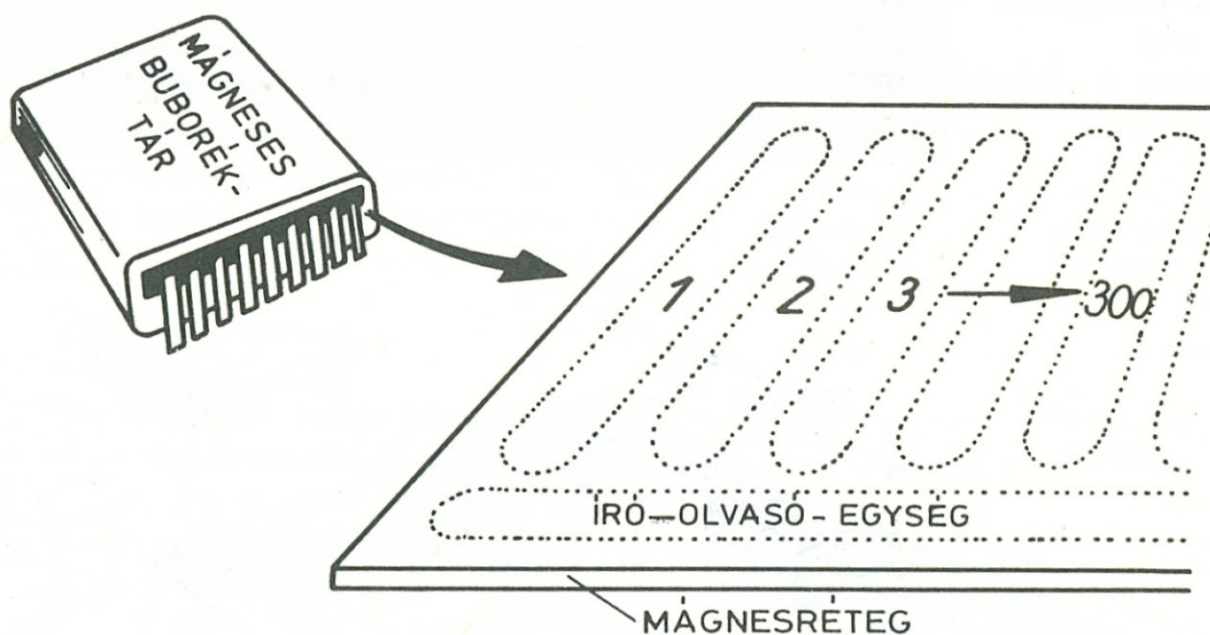
Az adatok e „körforgását” nem szabad megszakítani, mert különben azok elvesznek.



Buboréktár

Egy további nemmechanikus tárberendezés a **buboréktár** (magnetic bubble memory). Külső megjelenésre hasonlít a félvezető tár elemekhez, és azokhoz hasonlóan nyomtatott áramköri kártyákra kerülnek.

A buboréktechnikánál egy vékony mágneses anyagrétegen mikroszkopikus méretű kis „szigeteket” — buborékokat — képeznek. Mesterségesen előállított mágneses mezőben a buborékok körforgást végeznek, és eközben egy író—olvasó egységet érintenek, amelyen adatok írhatók be vagy olvashatók ki. Egy buborék jelenléte az **1** információnak, hiánya pedig a **0** információnak felel meg.



Egy buboréktár több száz buborékhurokból állhat, amelyek mindegyike több ezer buborékot tartalmaz. A buborékok olyan kicsinyek, hogy egy 3×3 cm méretű építőelemen több millió bit tárolható.

Bár a buboréktárak hozzáférési ideje valamivel kisebb, mint a leggyorsabb mechanikus táré, de mégis lényegesen hosszabb, mint a félvezető táruké. Ezzel szemben a nagyobb tárolási sűrűségük mellett még az is az előnyük, hogy áramkimaradás esetén az információk nem vesznek el.

Optikai táruk

További nemmechanikus táruk az **optikai táruk**. Magnetooptikai és holografikus tárukat különböztethetünk meg.

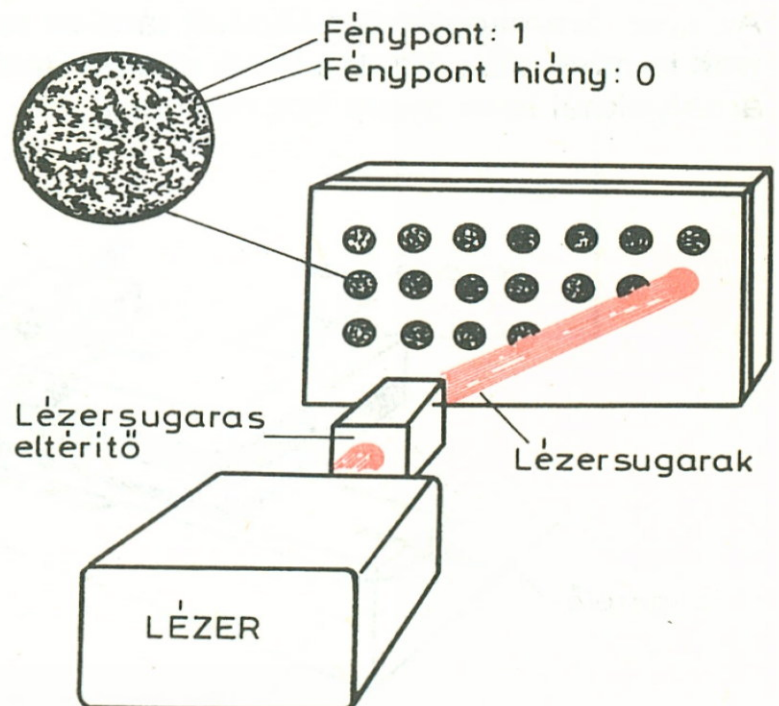
A **magnetooptikai táruk** az információkat lézersugár segítségével tárolják egy mágnesezett rétegen. A lézersugár felhevíti a mágnese réteget azokat a helyeket, amelyeknek az információkat fel kell venniük. Ezek a helyek ezáltal elvesztik mágneseességüket, és ezután egy mágnese mező által a kívánt mágnesezési irányba hozhatók (**0** vagy **1**).

Más optikai táruk esetén lézersugárral nagyon kis lyukakat olvasztanak az adathordozó anyagba, amely pl. egy mágneselemezhez hasonló speciális fémfilmel bevont, kör alakú lemez lehet. Egyetlen ilyen lemez néhány milliárd lyukat vehet fel, miközben egy lyuk pl. **1**-et, míg a lyuk hiánya **0**-t jelenthet. Az optikai tár adatainak olvasására gyenge lézersugarat használnak, amely a forgó lemezt letapogatja.

Az optikai táruk hátránya, hogy csak egyszer írhatók, és ezért elsősorban terjedelmes archívumok létesítésére alkalmasak.

A **holografikus táruk** a tárgyak háromdimenziós ábrázolását teszik lehetővé, de digitális adatok tárolására is használhatók. E táruk esetén azonban az egyes biteket nem pontszerűen, hanem az egész hologramfelületen elosztva tárolják. E felület fényérzékeny rétegből áll.

A holografikus tárolás nagy zavarbiztonságú. Ugyanis, ha a tár valamely részében zavar keletkezik, ez nem semmisíti meg a teljes információt, hanem csak csökkenti az egyes bitminták kontrasztját. Holografikus tárolással egymilliárd bit/cm² rögzítési sűrűség érhető el.



Alacsony hőmérsékletű táruk

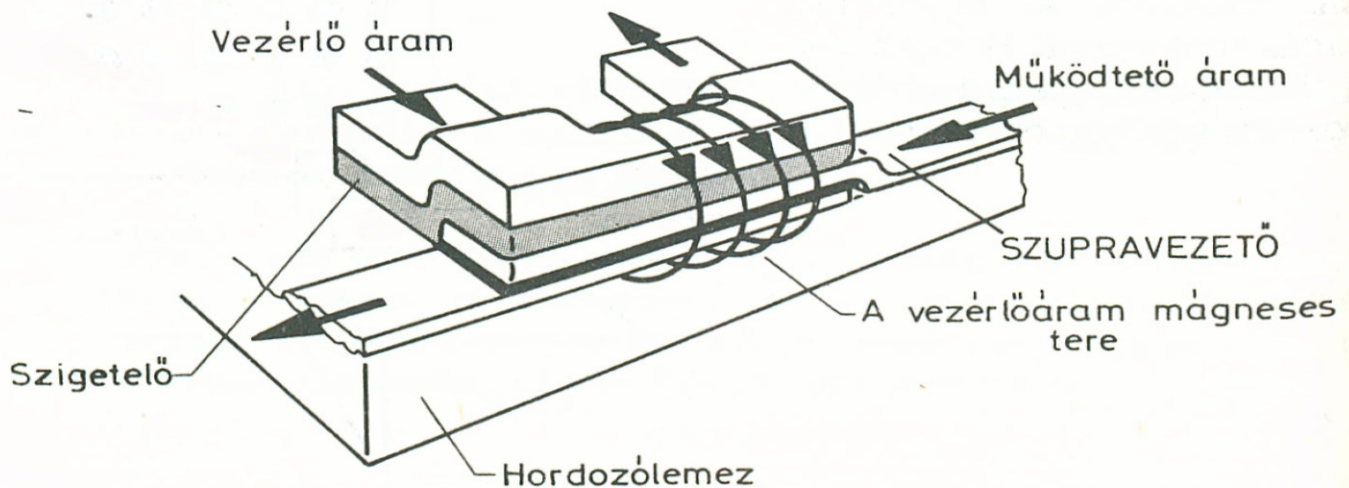
Egy mai gyorszámítógép áramkörei másodpercenként egymilliárd kapcsolást is végezhetnek (számjegyekkel kifejezve: 1 000 000 000), a kapcsolási idő tehát az ns tartományba esik. De kísérleteznek olyan áramkörökkel is már, amelyek s-onként kb. százmilliárd kapcsolásra képesek (számjegyekkel: 100 000 000 000).

Azonban van egy fizikai határ: egy elektromos jel egy ns alatt néhány cm utat tesz meg. Ha ns alatti kapcsolási időket akarunk kihasználni, akkor egy központi egység áramkörei legfeljebb néhány mm távolságban lehetnek egymástól. Ez azt jelenti, hogy az építőelemeket tovább kell miniatürizálni. A félvezető-áramkörök üzeme során azonban hő fejlődik, amely az építőelemek még nagyobb sűrűsége esetén már nem vezethető el.

Tehát olyan áramköröket kellett kifejleszteni, amelyek nagyon kevés energiát igényelnek, és ezért kevés hőt fejlesztenek. Ilyen áramkörök kifejlesztése Josephson nevéhez fűződik, az általa kifejlesztett áramköröket **Josephson-áramköröknek** nevezik.

A Josephson-áramkörök szupravezető fémrétegekkel dolgoznak, azaz olyan fémekből készülnek, amelyek az abszolút nullpont közelében (mínusz 273 °C) az átfolyó árammal szemben már nem tanúsítanak ellenállást. Így rendkívül kis áramigényűek. A szükséges kis hőmérséklet elérésére az építőelemeket folyékony héliumba ágyazzák.

Az ilyen áramkörökből felépített tárukat ezért **alacsony hőmérsékletű táruk-nak** is nevezzük. Ezek a táruk olyan tárolási feladatok elvégzésére hivatottak, amelyeknél igen gyors hozzáférésre van szükség.



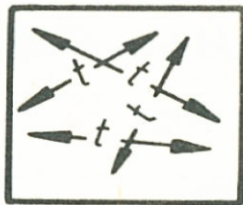
Befejezésül megadjuk az ismertetett táruk néhány jellemzőjét, annak érdekében, hogy az egyes táruk tulajdonságairól áttekintést adjunk.

Félvezetős és buboréktáruk 100 millió, lemeztáruk egymilliárd, optikai táruk tízmilliárd, mágnesszalagos táruk százmilliárd byte-ot tudnak tárolni.

A félvezetős táruk hozzáférési ideje egymilliomod másodperc alatt van, a buborék-, lemez- és optikai táruk hozzáférési ideje néhány ezred másodperc, a mágnesszalagos táruké pedig néhány másodperc.

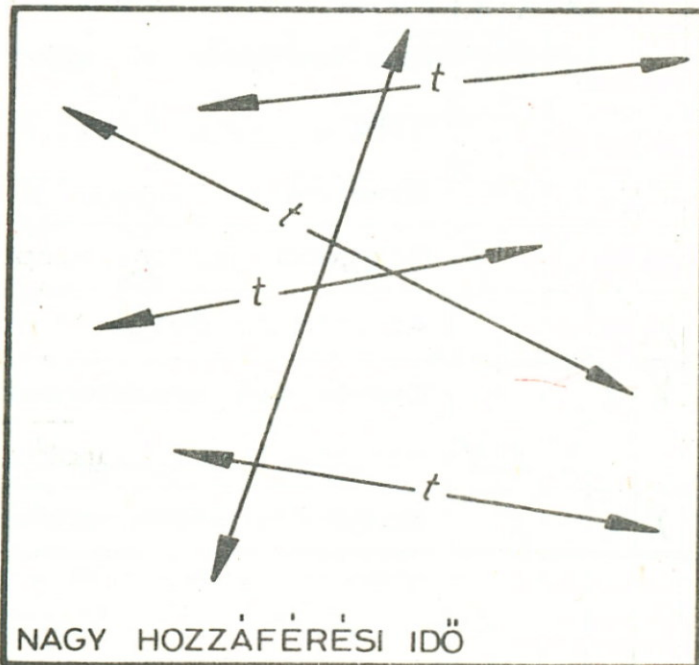
Ebből kitűnik, hogy nagy kapacitású és rövid hozzáférési idejű ideális tár nem létezik. A nagy kapacitás hosszú hozzáférési idővel jár együtt.

KIS KAPACITÁS



KIS
HOZZÁFÉRÉSI
IDŐ

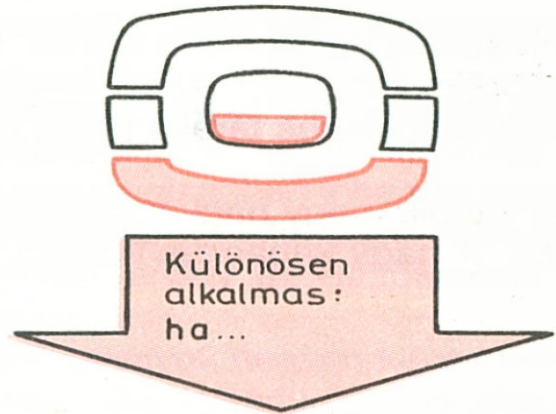
NAGY KAPACITÁS



NAGY HOZZÁFÉRÉSI IDŐ

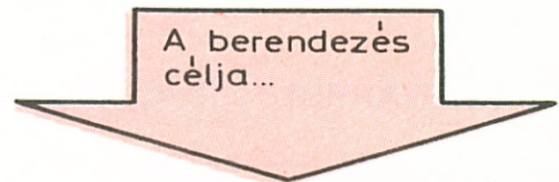
A háttértárak áttekintésének befejezéseként ismét összeállítást adunk az egyes tárolási technikák legfontosabb jellemzőiről, alkalmazásuk szempontjából.

TÁRAK							
							Mágneselemes-tár
							Magneszalagos-tár
							Kazettás mágneselemes-tár
							Hajlékony mágneselemes-tár
							Magneskártya-tár
							Felvezőtár
							Buborektár
							Optikai tár



X	X					X	X	X	Nagy adatmennyiségek
X						X	X	X	Kis hozzáférési idő
	X	X							Szekvenciális hozzáférés
		X	X	X	X	X	X		Közepes és kisebb számítógépek
X	X					X		X	Nagyobb számítógépek
	X	X	X						Az adathordozók továbbítása
X	X	X	X	X					Cserélhető adathordozók
						X	X	X	Helytakarékos tárolás
X	X	X	X	X		X			Nagy biztonság áramkimaradás

esetén



				X	X	X	X		Kizárólag adattárolás
X	X	X	X						Kivitel és bevétel is



1. Milyen hozzáféréssel használhatók

a mágneslemezes táruk:

a mágnesszalagos táruk:

2. Hogyan találunk meg egy adatot
a közvetlen hozzáférésű táruk esetén?

Egy cím segítségével.

Az adathordozó nagyon gyors végigkeresésével.

3. A következő táruk melyikének adathordozója cserélhető?

Mágnesszalagos tár.

Rögzített lemezes tár.

Hajlékony mágneslemezes
tár.

Mágnesszalag-kazettás
tár.

Mágneskártyás tár.

Félvezetős tár.

Buboréktár.

Holografikus tár.

4. Hogyan nevezzük azokat a félvezetős tárukat,
amelyekre az alábbi megállapítások igazak?

Mindegyik tárolóhely egy cím megadásán keresztül
tetszőlegesen sokszor olvasható, írható, törölhető:

Az adatokat a gyártó írja be,
és azok csak olvashatók:

Az adatokat a felhasználó egyszer írhatja be,
és azok tetszés szerint olvashatók:

Az adatok néhányszor törölhetők
és újraírhatók:



Válaszok

1. **Közvetlen hozzáféréssel** használható a mágneslemezes tár,

soros vagy szekvenciális hozzáféréssel használható a mágnesszalagos tár.

2. Közvetlen hozzáférésű táruk esetén egy adat megtalálható:

Egy cím segítségével.

3. Az adathordozó a következő táruk esetén cserélhető:

Mágnesszalagos tár.

Hajlékony mágneslemezes tár. Mágnesszalagkazettás tár.

Mágneskártyás tár.

4. Az említett megállapítások a következő félvezető tárukra igazak:

RAM mindegyik tárolóhely egy cím megadásán keresztül tetszőleges sokszor olvasható, írható, törölhető,

ROM az adatokat a gyártó írja be, és azok csak olvashatók,

PROM az adatokat a felhasználó egyszer írhatja be, és azok tetszés szerint olvashatók,

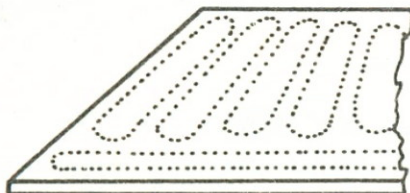
(R)EPROM az adatok néhányszor törölhetők és újraírhatók.



5. A következő tárok melyike dolgozik mechanikus (M), melyike nemmechanikus (N) alapon?

- | | | | |
|------------------------------|--------------------------|---|--------------------------|
| Mágnesszalagos tár. | <input type="checkbox"/> | Mágneslemezes tár. | <input type="checkbox"/> |
| Hajlékony mágneslemezes tár. | <input type="checkbox"/> | Félvezetős tár közvetlen hozzáféréssel. | <input type="checkbox"/> |
| CCD. | <input type="checkbox"/> | Buboréktár. | <input type="checkbox"/> |
| Holografikus tár. | <input type="checkbox"/> | Magnetooptikai tár. | <input type="checkbox"/> |

6.



Melyik technikát mutatja az ábra?

.....

7. Nevezze meg azokat a tárokat, amelyeknél az információk áramkimaradás esetén nem vesznek el!

.....

8. A következő tárok melyike ad közvetlen hozzáférési lehetőséget?

- | | |
|------------------------------|-----------------------|
| Mágneslemezes tár. | <input type="radio"/> |
| Mágnesszalagos tár. | <input type="radio"/> |
| Hajlékony mágneslemezes tár. | <input type="radio"/> |
| Félvezetős tár. | <input type="radio"/> |
| Buboréktár. | <input type="radio"/> |

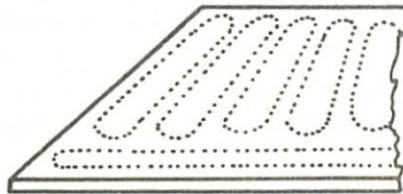


Válaszok

5. A következő táruk dolgoznak mechanikus (M), ill. nemmechanikus (N) alapon:

- | | | | |
|----------------------------|------------------------------|----------------------------|---|
| <input type="checkbox"/> M | Mágnesszalagos tár. | <input type="checkbox"/> M | Mágneslemezes tár. |
| <input type="checkbox"/> M | Hajlékony mágneslemezes tár. | <input type="checkbox"/> N | Félvezetős tár közvetlen hozzáféréssel. |
| <input type="checkbox"/> N | CCD. | <input type="checkbox"/> N | Buboréktár. |
| <input type="checkbox"/> N | Holografikus tár. | <input type="checkbox"/> N | Magnetooptikai tár. |

6.



Az ábra a buboréktár-technikát mutatja.

7. Táruk, amelyeknél az információk áramkimaradás esetén nem vesznek el:

valamennyi **mágnesréteges tár**, a **ROM**, a **PROM**, az **EPROM** és a **buboréktárak**.

8. A mágnesszalagos tár kivételével az összes felsorolt tár közvetlen hozzáférésű!

- Mágneslemezes tár.
-
- Hajlékony mágneslemezes tár.
- Félvezetős tár.
- Buboréktár.

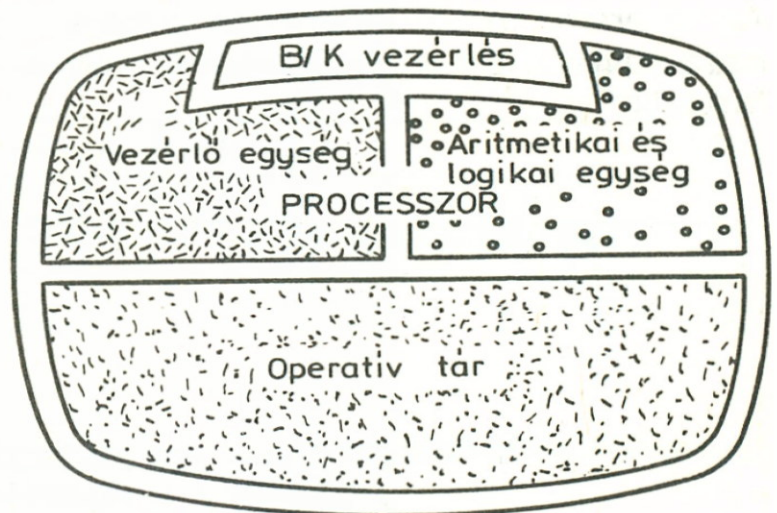
A feldolgozás

Míg a számítógép eddig megbeszélt funkcionális egységei a be-, ill. kiviteli egységek és a tárok — alapjában véve csak kiszolgáló tevékenységet folytatnak, hiszen feladatuk a feldolgozandó adatokat és a programokat a központi egységbe bevinni, azokat hosszabb ideig tárolni és az eredményeket kivinni, addig a **központi egység** az adatok tulajdonképpeni **feldolgozását** végzi.

A központi egység vagy **CPU (Central Processing Unit)** a számítógép „szíve”. Az adatfeldolgozó berendezés minden feldolgozási folyamatát ez vezérli és ellenőrzi. Még azok az adatok is áthaladnak a központi egységen, amelyeket a beviteli egységek beolvasnak, egy háttértárra kerülnek, majd változás nélkül, a kiviteli egységeken keresztül kiíródnak. A központi egység ezeket az adatokat feldolgozza, még ha a feldolgozás csak abból áll is, hogy az adatokat a beviteli egységekből átveszi és egy rendelkezésre álló tárnak átadja, ill. azokat a tárból a kiviteli egységhez továbbítja.

A számítógép központi egysége lényegében a következő részekből áll:

- vezérlőegység,
- aritmetikai és logikai egység,
- operatív tár,
- be-, ill. kiviteli vezérlés.



A vezérlő-, valamint az aritmetikai és logikai egységet közös néven **proceszszornak** nevezzük.

A vezérlőegység, az aritmetikai és logikai egység, az operatív tár, valamint a be-, ill. kiviteli vezérlésen túl a központi egységnek további részei is vannak, amelyeket majd a „Központi egység további egységei” c. pontban írunk le.

A vezérlőegység

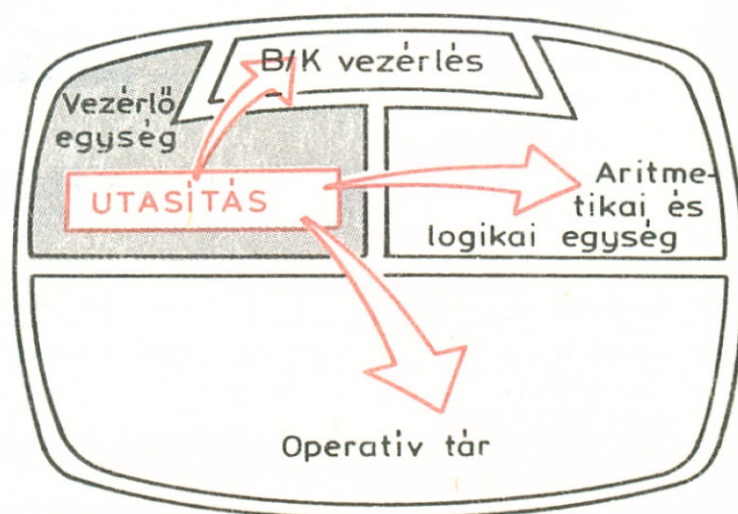
A **vezérlőegység** vezérli és koordinálja mindazokat a tevékenységeket, amelyek a központi egységben lefolynak. Az angol—amerikai szakirodalomban „control unit”-nak nevezik.

A vezérlőegység hívja be a programot — amelyben pontosan rögzítve van, amit a számítógépnek tennie kell — utasításról utasításra a helyes sorrendben az operatív tárból, értelmezi az egyes utasítások jelentését, és megfelelő vezérlőimpulzusokat ad a központi egység egyéb részeinek.

Ha pl. egy számítási utasításról van szó (pl. „add hozzá a következő mennyiséget az eddigi összeghez”), akkor a vezérlőegység impulzusokat küld az operatív tárnak ahhoz, hogy a számításhoz szükséges adatokat („mennyiség” és „összeg”) behívja és az aritmetikai és logikai egységbe továbbítsa. Ezt követően a vezérlőegység vezérlőjeleket ad ki az aritmetikai és logikai egység részére annak érdekében, hogy az a két adatot összegezze.

Eközben a vezérlőegység ügyel arra, hogy ne adjon ki vezérlőjeleket a központi egység ama részeihez, amelyek még más feladatokkal vannak elfoglalva.

Az operatív tárból kihozandó adatok címeit gyakran több címrészből kell összeállítani. Ez a cím kiszámítás ugyancsak a vezérlőegység feladataihoz tartozik.



Az aritmetikai és logikai egység


Az aritmetikai és logikai egység, amelyet ALU-nak (**A**rithmetic and **L**ogic **U**nit) is neveznek, végzi az összes számítási műveletet.

Számítási műveleten nemcsak az aritmetikai alpműveleteket (összeadás, kivonás, szorzás, osztás), hanem az összehasonlítások végrehajtását, a logikai műveletek elvégzését vagy a számértékek kerekítését is értjük.

Ilyen műveleteket aránylag könnyen végez el a számítógép. A bonyolult számítások pedig gyakran sok egyszerű műveletre bontva végezhető el. Így pl. a szorzás ismételt összeadásként is végrehajtható.

Az eredmény gyorsabb elérésére az aritmetikai egység géporientált számítási eljárásokat alkalmaz. Példa erre a számjegyek eltolásával végzett szorzás vagy osztás. Mint tudjuk, a számítógép, a túlnyomórészt tízes számrendszert használó emberrel ellentétben, kettes számrendszerben* számol. Ez a rendszer a számok ábrázolására csak a **0** és az **1** számjegyeket használja. Ha egy kettes számrendszerben adott számot egy számjegypozícióval balra eltolunk, akkor minden számjegypozíció helyértéke** kétszeresére nő, ami 2-vel való szorzásnak felel meg.

Bináris szám								Decimális szám	
Aritmetikai és l.e. (í28)									
(64)	(32)	(16)	(8)	(4)	(2)	(1)			
0	0	0	0	1	1	0	1	$= (8+4+1) = 13$	
0	0	0	1	1	0	1	0	$= (16+8+2) = 26$	



Miként a példa is mutatja, a bináris szám egy számjegypozícióval balra való eltolása 2-vel való szorzásnak felel meg.

Két számjegypozícióval való eltolás 4-gyel, három számjegypozícióval való eltolás 8-cal, négy számjegypozícióval való eltolás 16-tal való szorzásnak felel meg stb.

A 2, 4, 8, 16 stb. számokkal való osztást a bináris szám 1, 2, 3, 4 stb. számjegypozícióval való jobbra tolásával végezhetjük.



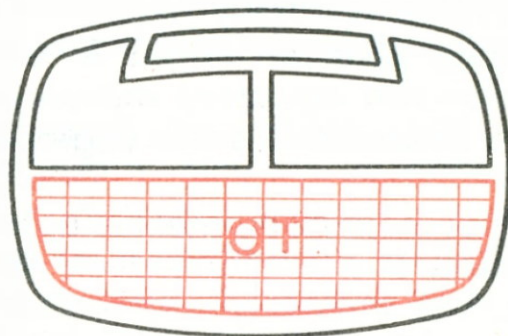
* A számok kettes számrendszerbeli ábrázolását bináris, ritkábban duális ábrázolásnak is nevezik.

** A helyérték a számjegypozícióhoz tartozó érték.

Az operatív tár

Az **operatív tár**, amelyet központi, primer vagy főtárnak (angol nyelven: main memory) is neveznek, a számítógép „memóriája”.

Az operatív táiban tároljuk a közvetlenül feldolgozásra kerülő adatokat és programokat.



Az operatív tárat az információk elhelyezésére alkalmas sok rekeszes állvány-nak is képzelhetjük.

Minden rekesznek saját címe van. A tárolt információ e címen közvetlenül lehívható, azaz a tárolandó információk bármely rekeszben közvetlenül elhelyezhetők és azok e rekeszből ismét kivehetők, miközben bármely cím tartalma egyenlő idő alatt hozzáférhető. Az operatív tár tehát **közvetlen hozzáférésű**.

Információknak az operatív táiban történő „elhelyezését” írásnak vagy írási folyamatnak nevezzük. Az írás esetén azokat az információkat, amelyek az **írási folyamat** előtt a megfelelő tárolási helyen voltak tárolva, átírjuk, és ezek ezzel elvesznek.

Az előbbi ellenkezője az **olvasási folyamat**. Eközben a kívánt információkat a tár megadott címén keressük és az operatív tárból kivisszük. Az olvasott információk azonban továbbra is az operatív táiban maradnak tárolva, mert sok információra változatlan formában ismételtlen szükség van.

Némely operatív tár felépítése olyan, hogy az információk kiolvasását közvetlenül azok automatikus újrainírása követi.

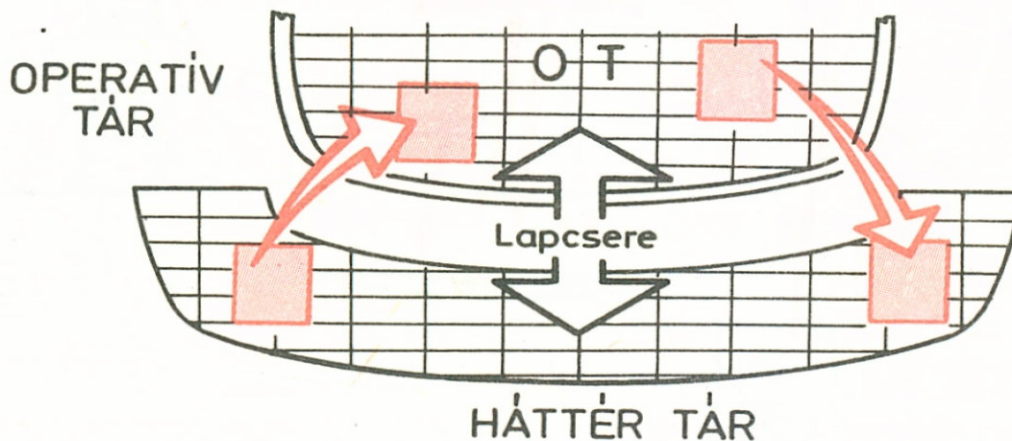
Egy operatív tár teljesítőképességét annak kapacitása és sebessége, tehát a **hozzáférési idő** határozza meg. Egy nagy teljesítményű tár sok millió karaktert tud befogadni, és közelítőleg csak egy milliommód másodpercre van szüksége egy adott című adat kiolvasásához.

Vannak olyan számítógépek, amelyekkel sok olyan felhasználó tart kapcsolatot, akik egymástól függetlenül és gyakorlatilag egyidejűleg futtatják saját külön programjaikat és dolgoztatják fel adataikat. Ebben az esetben még a több millió byte befogadóképességű operatív tár kapacitása sem elégséges.

Ahhoz, hogy mindezeket az adatokat mégis hozzáférhetővé tegyük, egy gyors háttértárat kapcsolunk hozzá, többnyire egy mágneslemezes tárat mint „kibővített operatív tárat”, és a programoknak és adatoknak mindig azokat a részeit visszük rá a mágneslemezes tárra, amelyekre a központi egységnek a feldolgozáshoz éppen nincs szüksége.

Ha a mágneslemezes táron levő adatok valamelyikére szükség van, akkor az villámgyorsan átvihető az operatív tárba.

Mivel ettől az operatív tár látszólag megnagyobbodik, ezért ilyenkor **látszólagos** vagy **virtuális tárolásról** beszélünk. Azért, hogy a virtuális tárolás során a programokat és adatokat minden további nehézség nélkül az operatív tárba be- és kivinni lehessen, ezeket egyenlő nagyságú egységekre osztjuk fel (pl. 4096 byte-ot tartalmazókra), amelyeket **lapoknak** is nevezünk. Ennek megfelelően a lapoknak az operatív tárból a mágneslemezes tárba történő átvitelét, vagy ennek megfordítottját **lapváltásnak** (paging) nevezzük.



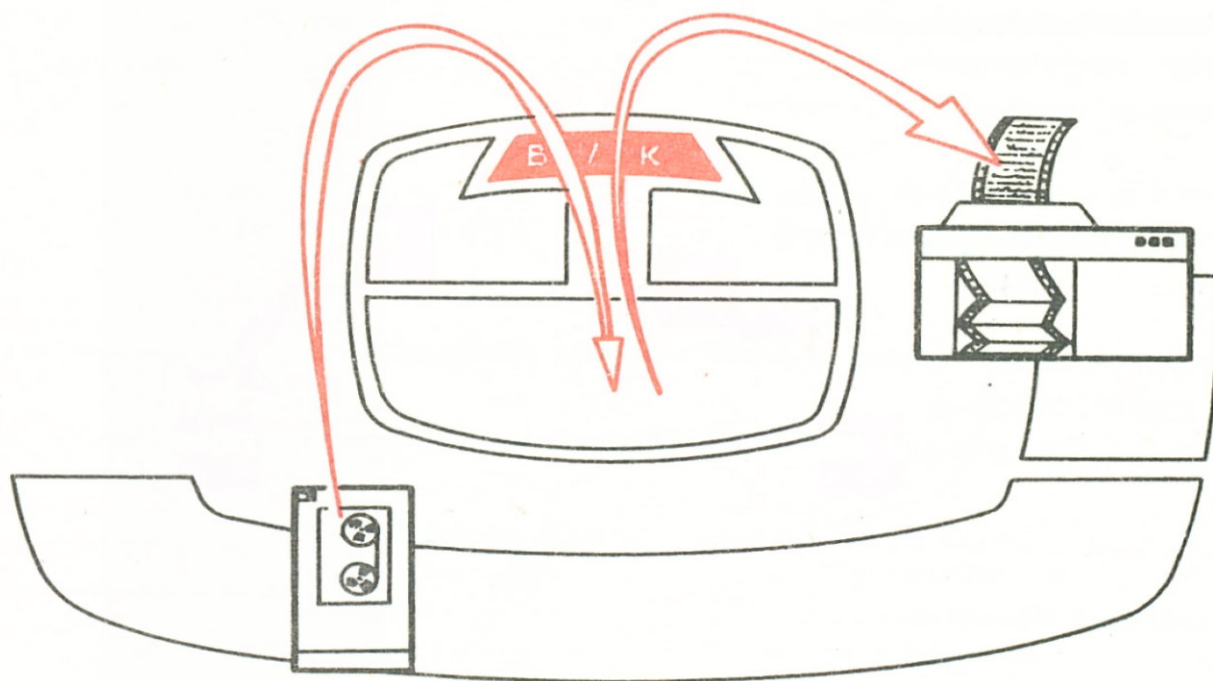
Természetesen arról gondoskodni kell, hogy az éppen feldolgozandó lapok az operatív tárban legyenek. Ezt a szervezési munkát, valamint a lapokra való felosztást is, speciális rendszerprogramok veszik át, amelyek a szokásos alkalmazói programok fölé vannak rendelve, és amelyek a felhasználók előtt észrevétlenül futnak.

A mai számítógépek operatív tárai többnyire *félvezető-technikával* készülnek. Mivel az operatív tárnak író—olvasó tárnak kell lennie, természetesen RAM-ról van szó. A félvezetők működését és a tárelemek megvalósítását e technikával könyvünk egy külön fejezetében írjuk le.

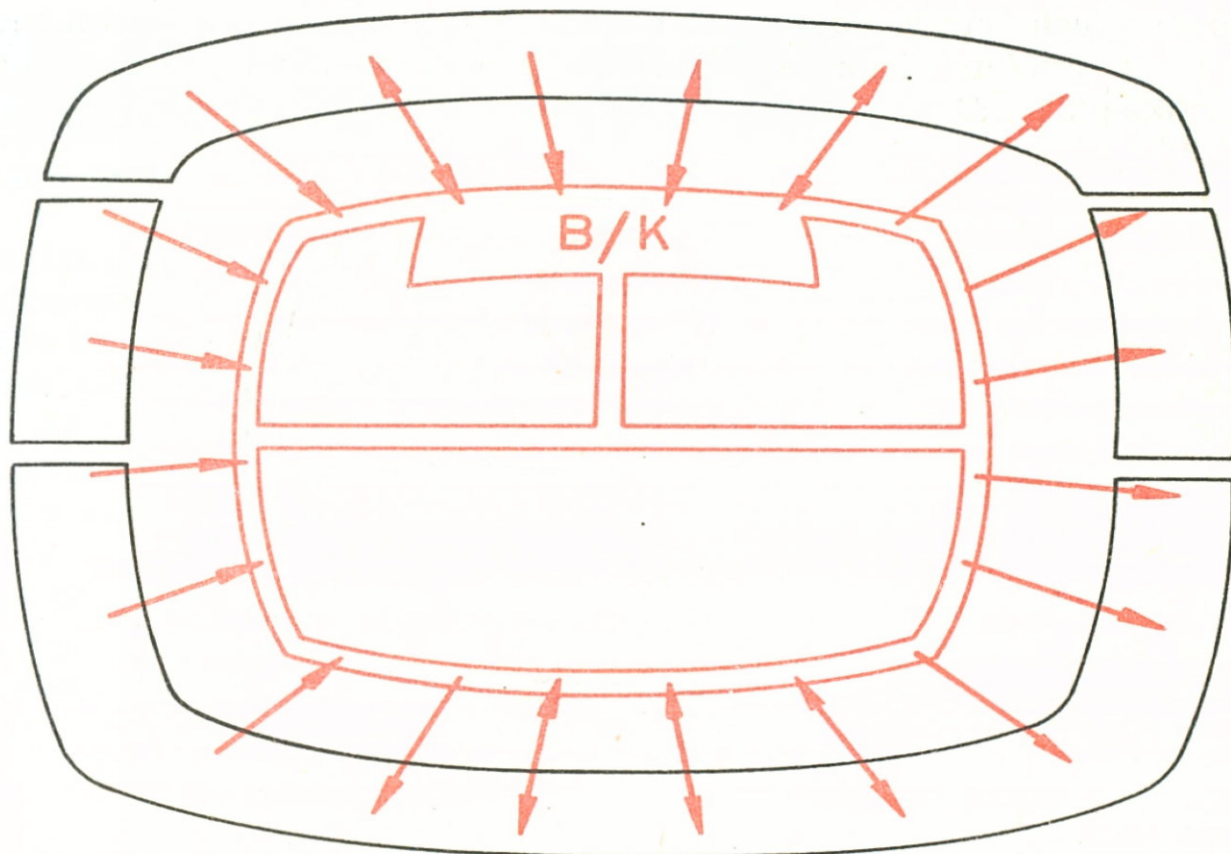
A be-, ill. kiviteli vezérlés

A **be-, ill. kiviteli vezérlésnek** — a vezérlőegység irányítása alatt — az a feladata, hogy a szükséges kapcsolatokat a központi egység és a perifériák között létrehozza és az adatátvitelt végrehajtsa.

A be-, ill. kiviteli vezérlés a vezérlőegység felszólítására kiválasztja azt a ki-, ill. beviteli csatornát, amelyhez az a periféria kapcsolódik, amellyel az adatátvitelnek meg kell történnie. A be-, ill. kiviteli vezérlés ezek után gondoskodik arról, hogy az adatok abban a mennyiségben és ütemben kerüljenek átvitelre, ami a kiválasztott periféria műszaki paramétereinek megfelel. Eközben az átvitt információkat teljesség és helyesség szempontjából folyamatosan ellenőrzi.



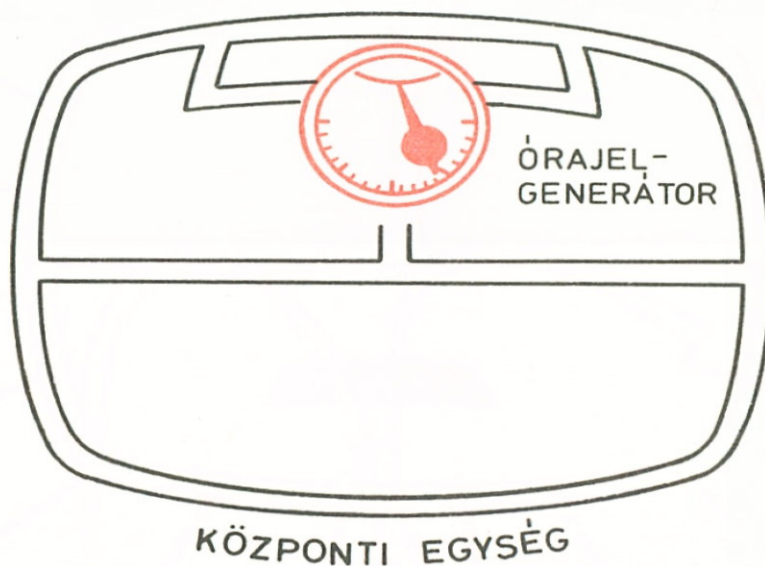
A *be-, ill. kiviteli vezérlés* a lefutó program alapján a vezérlőegységtől kapja meg a parancsot, és ezek után többnyire önállóan hajtja végre a *be-, ill. kiviteli műveleteket* úgy, hogy a vezérlőegység a következő utasítások végrehajtására felszabadul. Azt a *be-, ill. kiviteli vezérlést*, amely a számítógép processzorával párhuzamosan autonom dolgozik, **be-, ill. kiviteli processzornak** is nevezik.



A központi egység egyéb egységei

A központi egység áramköreinek működését elképzelhetetlen gyorsasággal egymás után következő óraimpulzusok szinkronizálják. A számítógép típusa szerint ez egy másodperc alatt az órajelek milliárdjait jelentheti — és ez még mindig nem jelenti a technikai fejlődés végét.

Ezeket az órajeleket egy ugyancsak a központi egységben elhelyezett egység hozza létre, amelyet **órajelgenerátornak** (clock) neveznek. Az órajelek feladatát, alkalmazását részletesebben könyvünk utolsó fejezetében ismertetjük.



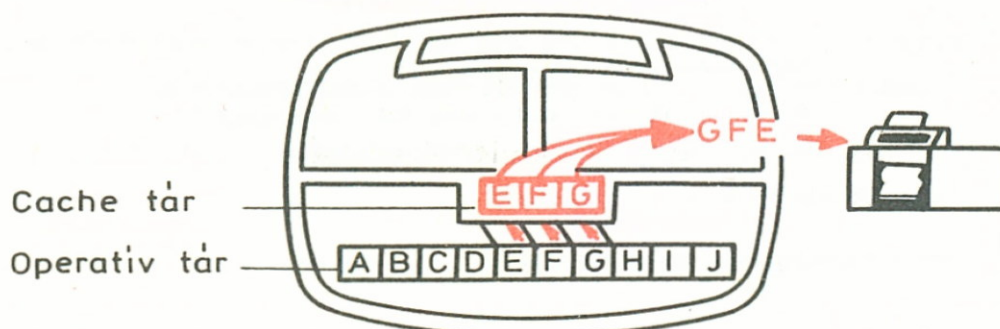
Az operatív tár mellett további más táruk is vannak a központi egységben, amelyeknek a feladata az adatok *rövid idejű* tárolása.

Ezek lehetnek egyrészt **gyorstárak**, amelyek rendkívül gyors működésű regiszterekkel dolgoznak, és amelyeket főként közbeeső részeredmények rövid idejű tárolására használunk. Ezeket a kis kapacitású gyorstárakat angolul *scratch-pad memory*-nak nevezik*.

Másrészt lehetnek közbenső táruk vagy **puffertárak** is, amelyeket kis kapacitású és gyors átmeneti tárként az operatív tár elé kapcsolnak. A kis kapacitású puffertár hozzáférési ideje sokkal rövidebb, mint a nagyobb kapacitású operatív táré.

Ha a központi egység ilyen puffertárral van ellátva, akkor az operatív tár tartalmának a puffertárba való átírásánál nemcsak azokat az adatokat írják át a puffertárba, amelyekre közvetlenül szükség van, hanem mindjárt a szomszédos tárolóhelyek tartalmát is, mert elég nagy annak a valószínűsége, hogy az operatív tár következő elérésekor e közelben tárolt információk valamelyikére lesz szükség, és ezel az operatív tár újbóli olvasása fölöslegessé válik.

Ezt a központi egységben elhelyezkedő puffertárat **cache-tárnak** (cache-memory) is nevezik. [Úgy kell kiejteni, mint az angol „cash” szót, azonban a francia „cacher”-ből (eldugni) származik.]



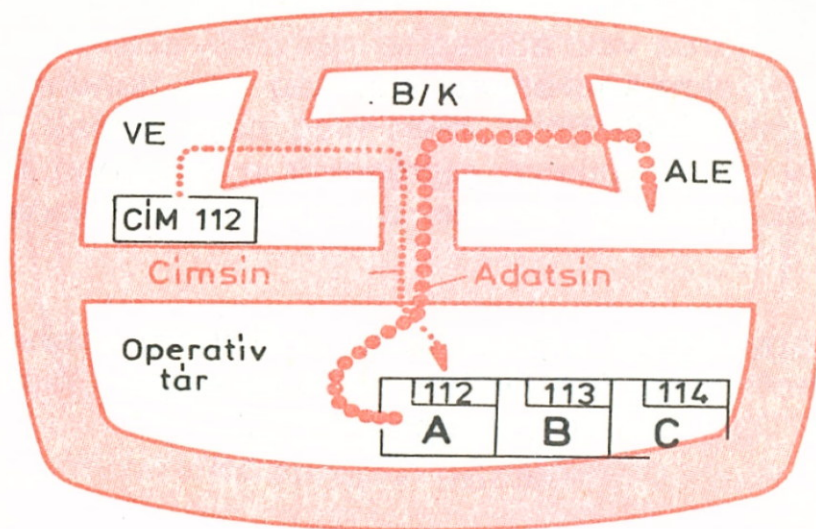
A puffertárak egyébként olyan esetekben is használatosak, amikor a számítógép gyorsabb egységei lassúbbakkal dolgoznak együtt, pl. ha a központi egység nagy sebességgel ad ki adatokat egy nyomtatónak, amelyeket ez csak aránylag lassan tud kinyomtatni. Ebben az esetben a puffertár az adatokat nagy sebességgel veszi át, tárolja azokat, majd azzal a sebességgel adja át a tárolt adatokat a nyomtatónak, amellyel a nyomtató az adatokat ki tudja írni.

* Néha a jegyzettömbtár elnevezést is használják. (Lektor.)

A központi egység részeinek ismertetése során nem térünk rá azokra a vezetésekre, amelyek a központi egység különböző részeit összekötik, és amelyeken az információk kicserélődnek.

Az ilyenfajta vezetéseket többnyire **buszoknak (síneknek)** nevezzük (bus). A központi egységnek legtöbbször egy **címbusza** (addressbus) és egy **adatbusza** (databus) van. Az előbbi az operatív tár bizonyos tárolóhelyeinek eléréséhez, az utóbbi az adatoknak az operatív tárból az egyéb egységekhez való átviteléhez használható.

A számítógép konstrukciója szerint a címbusz és az adatbusz bizonyos számú vezetékből áll, pl. egy nagyszámítógépnél az adatbusz 64 vezeték lehet.



VE: vezérlő egység

ALE: aritmetikai-logikai egység

B/K: be-, ill. kiviteli egység



1. A következő feldolgozási folyamatok melyikében nem vesz részt a központi egység?

Az adatokat a beviteli egység átveszi, és átadja egy tárnak.

Egy háttértáron levő adatokat rendezik.

Egy beviteli berendezésbe rendezetlen adatokat viszünk be, és ezek betűrendben nyomtatott listaként kerülnek kinyomtatásra.

2. A következő tevékenységek melyikét végzi a vezérlőegység (VE), melyiket az aritmetikai és logikai egység (ALE), melyiket az operatív tár (OT) és melyiket a be-, ill. kiviteli vezérlés (BIK)?

Az utasításokat behívja az operatív tárból.

Értelmezi az egyes utasításokat.

Összead, kivon, szoroz, oszt.

Vezérlőimpulzusokat ad a központi egység más részeinek.

Sok millió karaktert tud tárolni és a megcímzett adatot a másodperc tört része alatt ki tudja olvasni.

Kapcsolatot létesít a központi egység és a perifériák között.

Végrehajtja az adatátvitelt.

3. A központi egység mely más egységeit ismeri Ön a vezérlőegység, az aritmetikai és logikai egység, az operatív tár, és a be-, ill. kiviteli vezérlés mellett?

.....

.....

.....



Válaszok

1.

Egyik lehetőséget sem kell megjelölni,
mert a központi egység minden feldolgozási folyamatban
részt vesz.

2.

 VE

Az utasításokat behívja az operatív tárból.

 VE

Értelmezi az egyes utasításokat.

 ALE

Összead, kivon, szoroz, oszt.

 VE

Vezérlőimpulzusokat ad
a központi egység más részeihez.

 OT

Sok millió karaktert tud tárolni, és a megcímzett adatot
a másodperc tört része alatt ki tudja olvasni.

 BIK

Kapcsolatot létesít a központi egység
és a perifériák között.

 BIK

Végrehajtja az adatátvitelt.

3. A központi egység egyéb egységei a vezérlőegység,
az aritmetikai egység, az operatív tár és a be-, ill. kiviteli vezérlés
mellett: az **órajel-generátor**, a **gyorstár**, a **puffertár**, a **cache-tár**,
az **összekötő-vezetékek**, **buszok**.

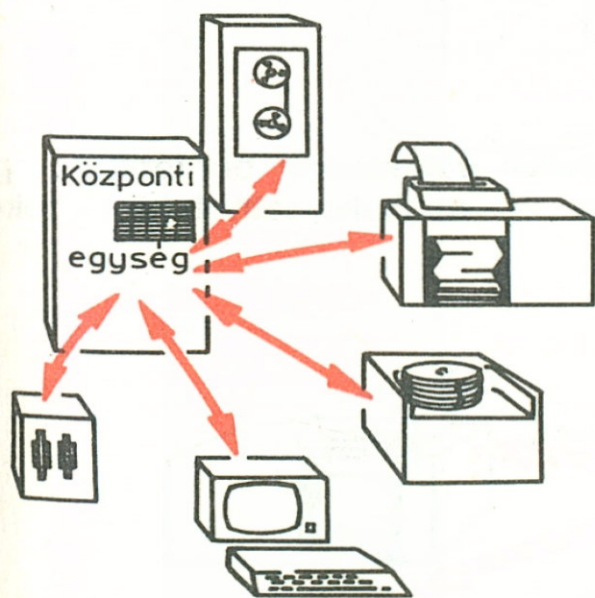
Adatátviteli berendezések

A központi egységet és a perifériákat egymással össze kell kapcsolni, hogy közöttük az adatok cseréje lehetővé váljék.

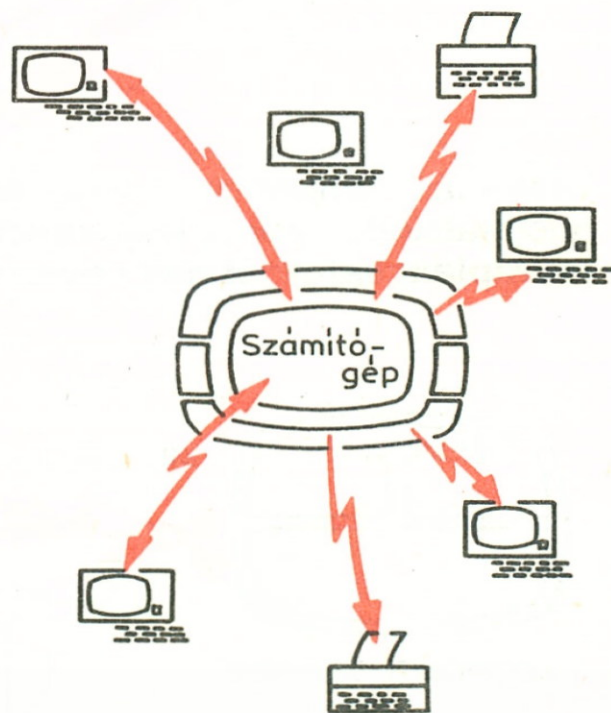
Ha egy periféria a központi egység közelében áll, amint ez pl. a közepes és nagyobb gépeknél a számítóközpontokban gyakran előfordul, akkor a berendezéseket **közeli kapcsolattal** csatlakoztatjuk a központi egységhez. A kis távolságon belül lefolytatott adatforgalmat **közeli forgalomnak** nevezzük.

Néhány perifériával ellátott kisebb számítógép esetén a központi egységet és a perifériákat gyakran úgy csatlakoztatjuk egymással, hogy azok egyetlen dobozban (házban) elhelyezhetők. Ez főként a személyi számítógépek esetében (personal computer) szokásos.

Azokat a perifériákat, amelyek nem a számítógép mellett vannak felállítva, hanem ott, ahol az adatok keletkeznek, **távoli kapcsolattal** kötjük a számítógéphez. Példa erre az utazási irodák könyvelési rendszere, amikor is a regionális irodákban álló terminálok a nagyvárosban levő számítógéppel távoli kapcsolatban állnak. Ebben az esetben **távfeldolgozásról** beszélünk.



Közeli adatforgalom

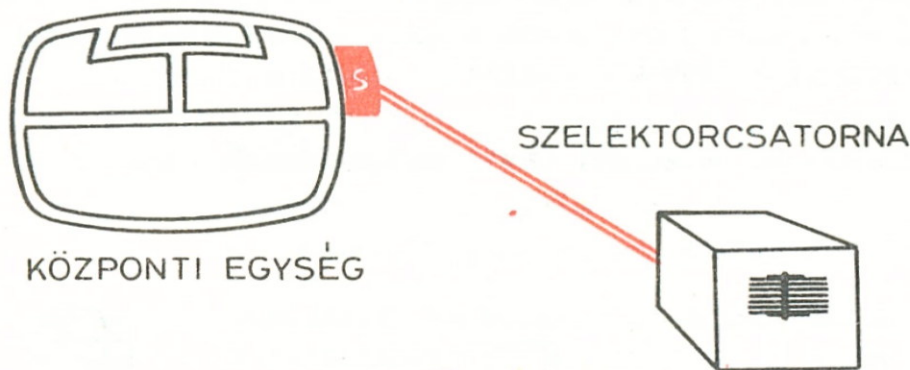


Távfeldolgozás

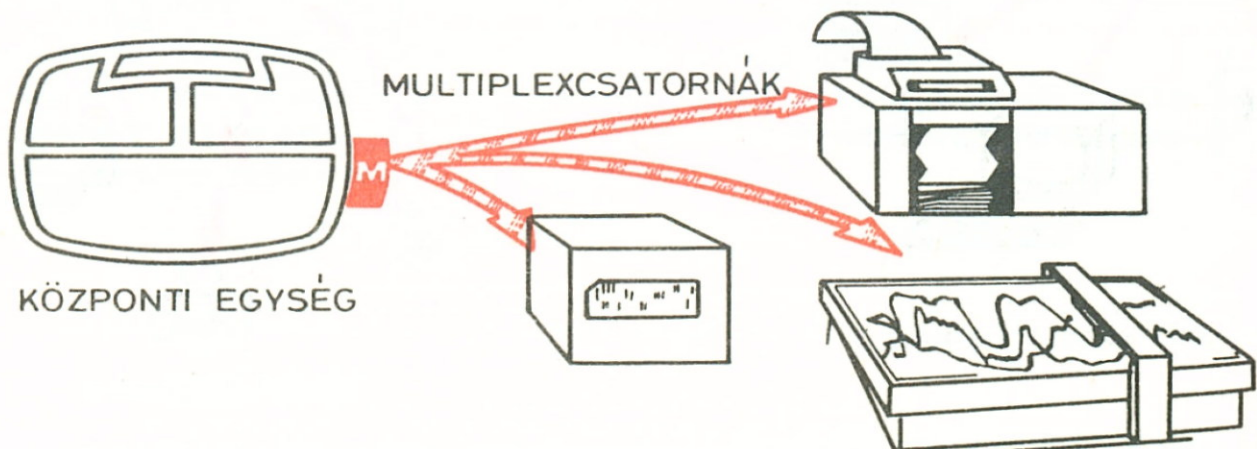
Közeli kapcsolatok

A központi egység és a közeli perifériák közötti adatforgalom az egy-egy géptípus vagy gépcsalád részére **egységesített szabványos csatlakozáson** (szabványos interfész) keresztül bonyolódik le. Ezek az egységes csatlakozások, amelyeknél a csatlakozóelemek típusa is szabványos, a perifériák egyszerű cseréjét vagy járulékos csatlakoztatását teszik lehetővé. Azt a berendezést, amely a beolvasandó vagy a kiírandó adatokat képes tárolni és más egységeknek továbbítani, **csatornának** nevezzük. A be-, ill. kiviteli vezérlésnek több ilyen csatornája van, amelyek két csoportba sorolhatók:

— csatornák, amelyek adatok gyors továbbítását teszik lehetővé a központi egység és egyetlen gyorsperiféria — pl. egy mágneslemez tároló — között; ezeket **szelektorcsatornáknak** nevezzük,



— csatornák, amelyekhez több kisebb átviteli sebességű perifériát — pl. nyomtatókat, lyukkártyaberendezéseket — csatlakoztathatunk, ezeket **multiplexcsatornáknak** nevezzük.

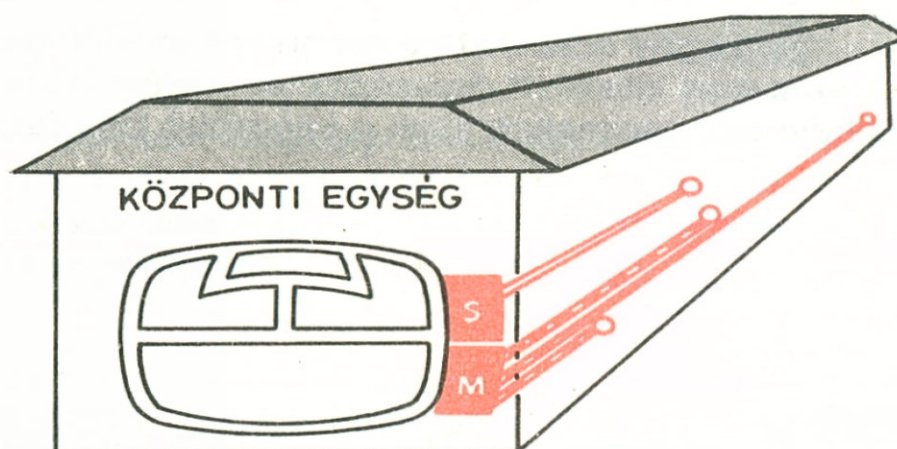


Míg egy periféria a szelektorcsatornán keresztül állandóan össze van kötve a központi egységgel, addig a multiplexcsatornára kapcsolt perifériák esetenként csak rövid időre foglalják le a csatornát.

Példa:

Egy nyomtató multiplexcsatornán keresztül csatlakozik a központi egységhez. A kapcsolat rövid ideje alatt bizonyos mennyiségű adat érkezik a nyomtatóhoz, amelyet egy puffertár vesz fel. A puffertár a nyomtató üzemi sebességéhez alkalmazkodó sebességgel adja át az adatokat a nyomtatónak.

Korszerű adatfeldolgozó berendezések esetén a csatornák mind szelektor-, mind pedig multiplexcsatornákkal el vannak látva. Az adatforgalom e csatornákon át azonban csak korlátozott távolságon belül lehetséges, ez a távolság a technikától és a periféria fajtájától függően néhány métertől néhány száz méterig terjedhet. Nagyobb távolságokon a távadatátvitel technikáját kell alkalmazni.



Távoli összeköttetések

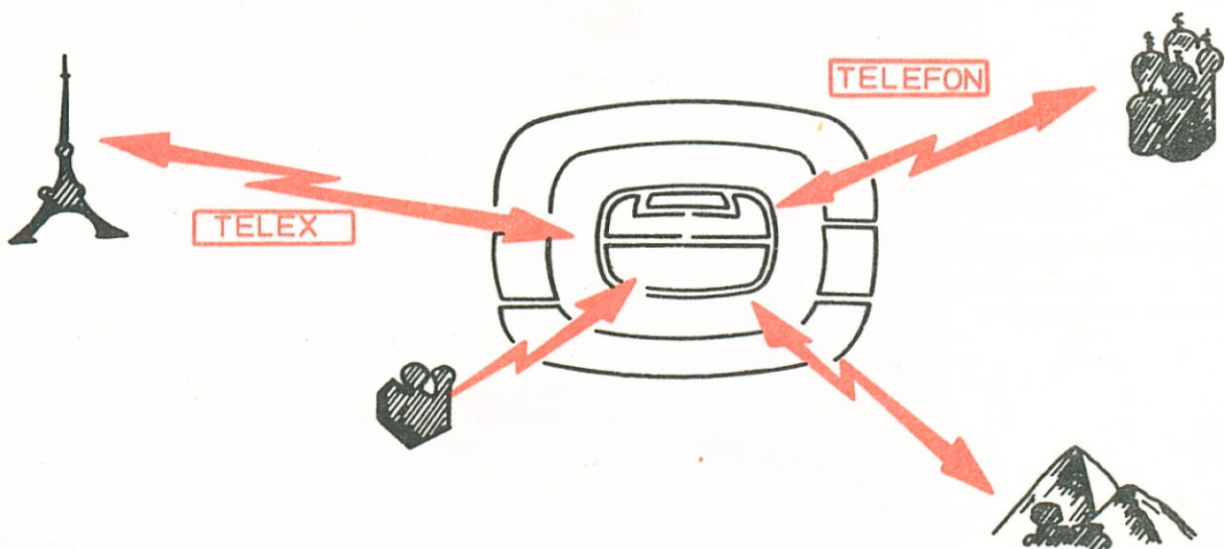
Ha a számítógép központi egysége és perifériái nem ugyanazon a helyen, hanem egymástól távoli helyeken vannak felállítva, akkor az adatokat **táv-adatátvitellel** kell kicserélni.

Adatátviteli hálózatok

A „Kommunikációtechnika” c. részben az adatátvitelre szolgáló különféle hálózatokat — telex-, adat-, távbeszélő- és közvetlen hívásos hálózatot — már megemlítettük. Az, hogy ezek közül az adatátvitelhez melyiket választjuk, az adatmennyiségtől, az átviteli gyakoriságtól és a sürgősségtől függ, azaz attól, hogy igény esetén a kapcsolat azonnal rendelkezésre álljon-e.

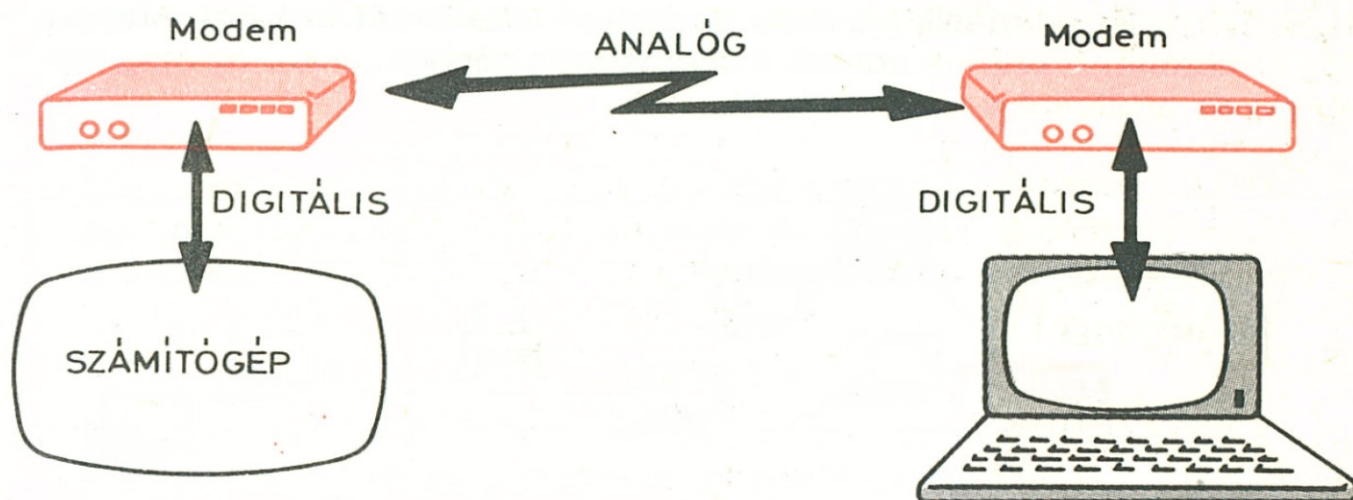
Ha nagy adatmennyiségek gyakran kerülnek átvitelre vagy a sürgősségnek nagy jelentősége van, akkor erre a közvetlen hívású hálózat **állandó vezeték**ei (bérelt vezeték) a megfelelőek, amelyekért rendszerint átalánybért kell fizetni.

Azonban ha kevésbé sürgős adatokat ritkábban kell átvinni, gazdaságosabb, ha a **kapcsolt hálózaton** létesített összeköttetés mellett döntünk, amelynek használatáért a havi alapbér mellett csak a kapcsolat időtartama szerint kell fizetnünk.

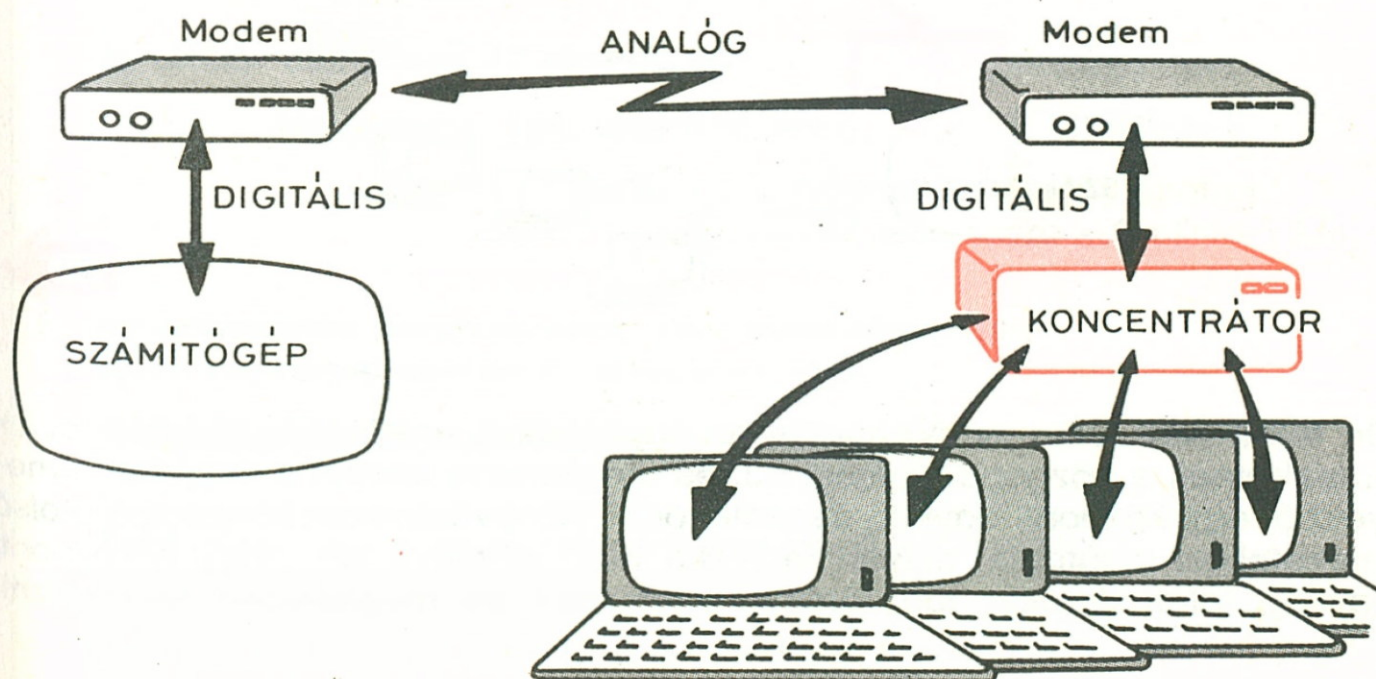


A távoli kapcsolatok melletti adatcseréhez különleges **adatátviteli berendezésekre** van szükségünk, ilyenek: a modemek, a koncentrátorok és a csomóponti gépek.

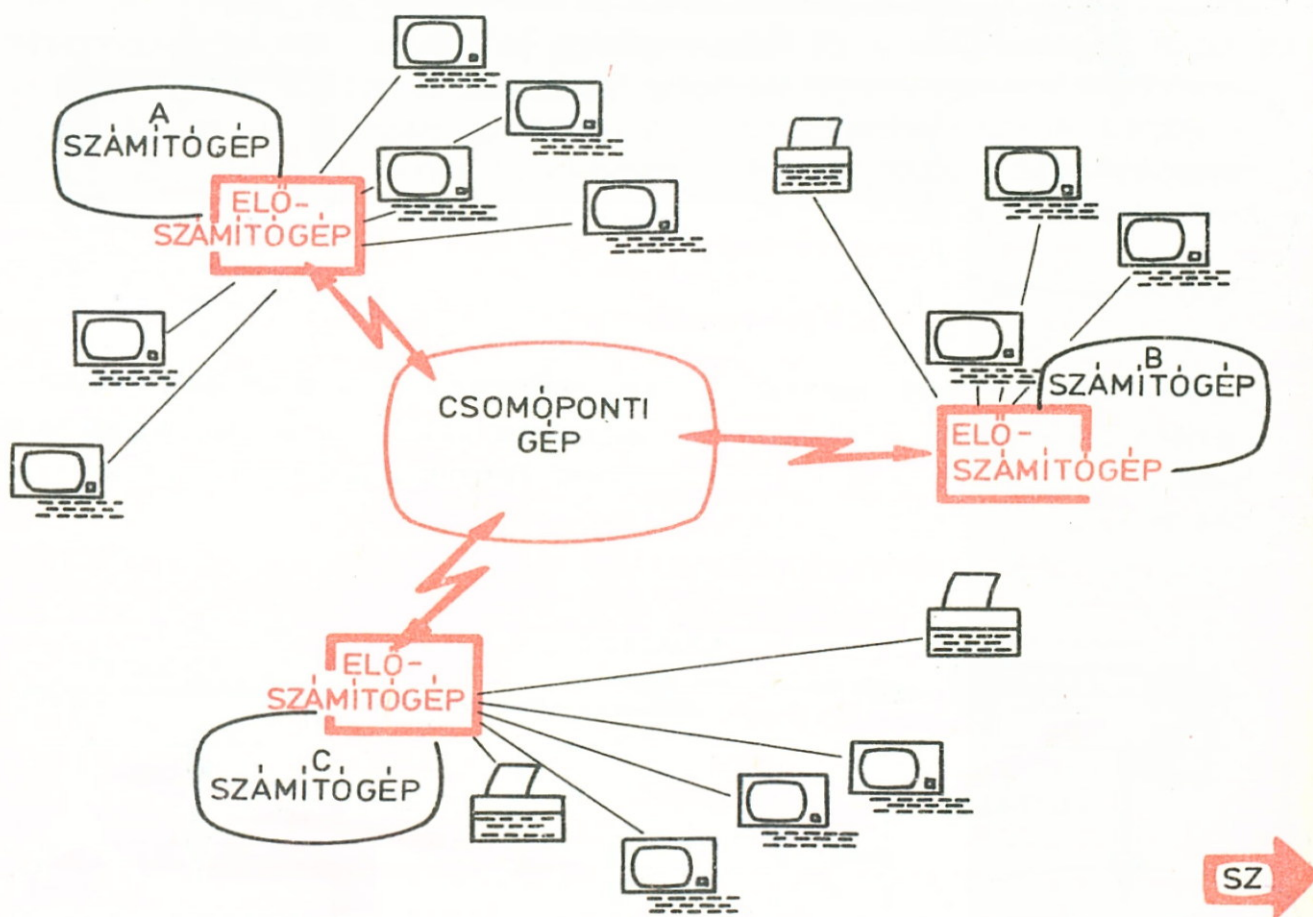
A **modem**, amelyet a „Kommunikációtechnika” c. részben ugyancsak bemutatunk már, egy olyan berendezés, amely a digitális adatokat úgy alakítja át, hogy azok egy analóg hálózaton — pl. a távbeszélő-hálózaton — átvihetők.



Ha több terminált kell valamely távoli kapcsolathoz csatlakoztatni, akkor **koncentrátorokat** kell alkalmazni, amelyek feladata a beérkező adatok gyűjtése és továbbítása. Így nem minden terminálnak van szüksége külön vezetékre a számítógéphez.



Az ún. **csomóponti gépeket** az átviteli hálózat csomópontjain alkalmazzuk, amelyek az átviteli hálózatba kapcsolt egységek között bonyolítják le az adatforgalmat. Ezek lehetnek: terminálok, egyéb perifériák vagy az adatátvitelre szolgáló speciális előszámítógépekkel ellátott teljes *adattfeldolgozó berendezések*. A csomóponti gépek képesek pl. az átviendő adatokat átmenetileg tárolni, és szabad vezetékkapacitás esetén a címzethez továbbítani. Meg tudják állapítani az optimális átvitel útját is egy távoli előfizetőhöz, vagy hálózati zavar esetén kerülő úton összeköttetést tudnak létesíteni. A csomóponti gép másik feladata az *adatok hibátlan átvitelének* ellenőrzése. Ha pl. hibásan érkeznek adatok a csomóponti géphez, akkor ez még egyszer lehívja az adatokat a feladóhelyről.



Őn tehát ezek után ismeri azokat az egységeket, amelyeken át információk érkehetnek a központi egységhez, és megismerte azokat a részeket, amelyekből egy központi egység összetevődik. Könyvünk most következő utolsó fejezetében megtudja, milyen technika teszi lehetővé azt, hogy a központi egység információkat tud tárolni, értelmezni és megfelelően feldolgozni.



1. A kapcsolatok mely fajtájával vannak a perifériák a központi egységhez kapcsolva:
 Ha a periféria a központi egység közelében áll?

 Ha a periféria a központi egységtől távoli helyen áll?

2. Minek nevezzük az egy-egy gép vagy gépcsalád részére egységesített csatlakozást, amelyen át az adatforgalom a központi egység és a közeli periféria között lebonyolódik?

3. Melyik csatorna teszi lehetővé az adatok gyors továbbítását a központi egység és egyetlen gyorsperiféria között?
 Multiplexcsatorna.
 Szelektorcsatorna.
4. a) Hogyan nevezünk egy közvetlen hívású hálózaton lefoglalt (bérelt) vezetékét?

 b) Mikor előnyösek az állandó vonalak?
 Ha gyakran kell nagy mennyiségű adatot továbbítani.
 Ha az összeköttetésnek igény esetén *azonnal* rendelkezésre kell állnia.
5. Az adatátviteli berendezések mely fajtái alkalmasak távoli kapcsolat esetén az adatcseréhez?
 A digitális adatok átalakítására analóg jelekké.

 Több terminál csatlakoztatására egy távoli kapcsolathoz.

 Egy átviteli hálózat vezérlésére és ellenőrzésére.



Válaszok

1. A perifériák a következő kapcsolattal csatlakozhatnak a központi egységhez:

Közeli kapcsolat, ha a periféria a központi egység közelében áll.

Távoli kapcsolat, ha a periféria a központi egységtől távoli helyen áll.

2. **Szabványos csatlakozásnak** nevezzük az egy-egy géptípus vagy gépcsalád részére egységesített csatlakozást, amelyen át az adatforgalom a központi egység és a közeli periféria között lebonyolódik.

3. Az adatok gyors továbbítását a központi egység és egyetlen gyorsperiféria között lehetővé teszi:

Szelektorcsatorna.

4. a) Az ilyen lefoglalt (bérelt) összeköttetést **állandó vonalnak** nevezzük.

b) Az állandó vonalak előnyösek:

Ha gyakran kell nagy mennyiségű adatot továbbítani.

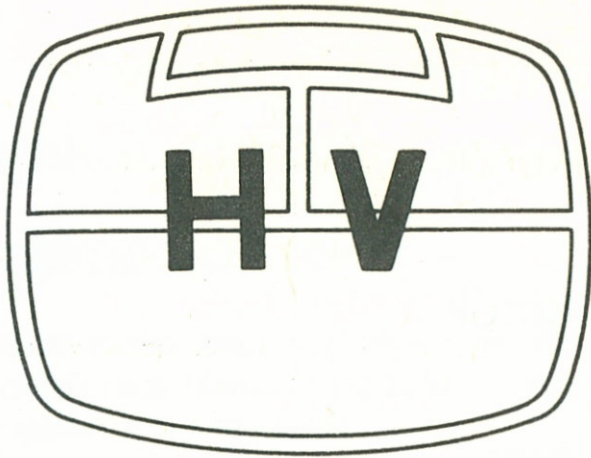
Ha az összeköttetésnek igény esetén azonnal rendelkezésre kell állnia.

5. Távoli kapcsolat esetén az adatcseréhez a következő adatátviteli berendezések alkalmasak:

Modem digitális adatok átalakítására analóg jelekké.

Koncentrátor több terminál csatlakoztatására egy távoli kapcsolathoz.

Csomóponti gép egy átviteli hálózat vezérlésére és ellenőrzésére.



Adatfeldolgozás
és kommunikáció

Számítógépek
és alkalmazásuk

A számítógépek
funkcionális
egységei

Ebből a fejezetből megtudja,
hogyan lehetséges az,
hogy a számítógép
az utasításokat megérti
és végre tudja hajtani.

Ehhez megismeri
a számítógép-áramkörök
logikai működését
és műszaki megvalósítását,
és egy példán láthatja,
hogyan kell elképzelnünk
egy utasítássorozat
feldolgozását a számítógépben.

Hogyan működik
a számítógép?

A számítógép-áramkörök
logikai alapjai

A számítógép-áramkörök
megvalósítása

A hardver és a szoftver
kapcsolata

Visszatekintés a múltba
és előretekintés a jövőbe

A számítógép-áramkörök logikai alapjai

Alapkapcsolások

A számítógép alapjában véve csak olyan okos, hogy az elektromos áram meglétét vagy hiányát meg tudja különböztetni. Az áram jelenlétéhez az **1**-et, hiányához a **0**-t rendeljük.

A számítógép ezt a megkülönböztetést valószínűtlen gyorsasággal tudja elvégezni, és — amennyiben nem alkotó feladatokról van szó — ebben rejlik nagy előnye az emberi aggyal szemben.

A számítógép azért képes erre, mert lényegében rendkívül gyors működésű **elektronikai építőelemekből** áll, amelyek viszont néhány digitális alapkapcsolásból tevődnek össze.

Ezek az egyszerű alapkapcsolások igen változatos kombinációs lehetőségeik által a legcsodálatosabb dolgokat tudják létrehozni. Az alapkapcsolásokat **ÉS**, **VAGY** és **NEM** kapuknak nevezzük, ezek legjobban egy billenőkapcsoló kapcsolási rajzával magyarázhatók meg.

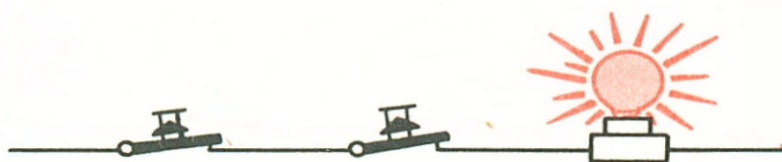
Az **ÉS** kapu kapcsolási vázlata:

Az **ÉS** kapu:

Ha két sorba kapcsolt billenő-kapcsoló nincs bekapcsolva, akkor a vezetékben nem folyhat áram. Akkor sem folyhat áram, ha csak az egyik kapcsoló van zárva.



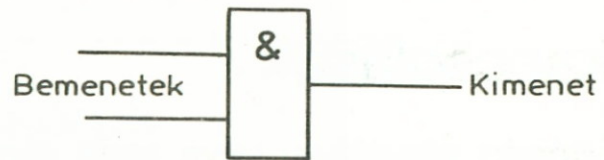
Csak ha mindkét kapcsolót zárjuk, akkor folyhat áram.



A gyakorlatban ezeket a kapcsolásokat természetesen nem mint billenőkapcsolókat ábrázoljuk. Ehelyett egyszerű áramköri jelképeket használunk.

Egy kétbemenetű ÉS kapu jelképe:

A billenőkapcsoló-modellben áram folyhat, ha mindkét kapcsolót zárjuk.



Ennek megfelelően az ÉS kapu kimenetén csak akkor folyik áram, ha mindkét bemenet zárva van.

A kétbemenetű ÉS kapunál négy kombináció lehetséges:

vagy egy bemeneten sincs feszültség (A),

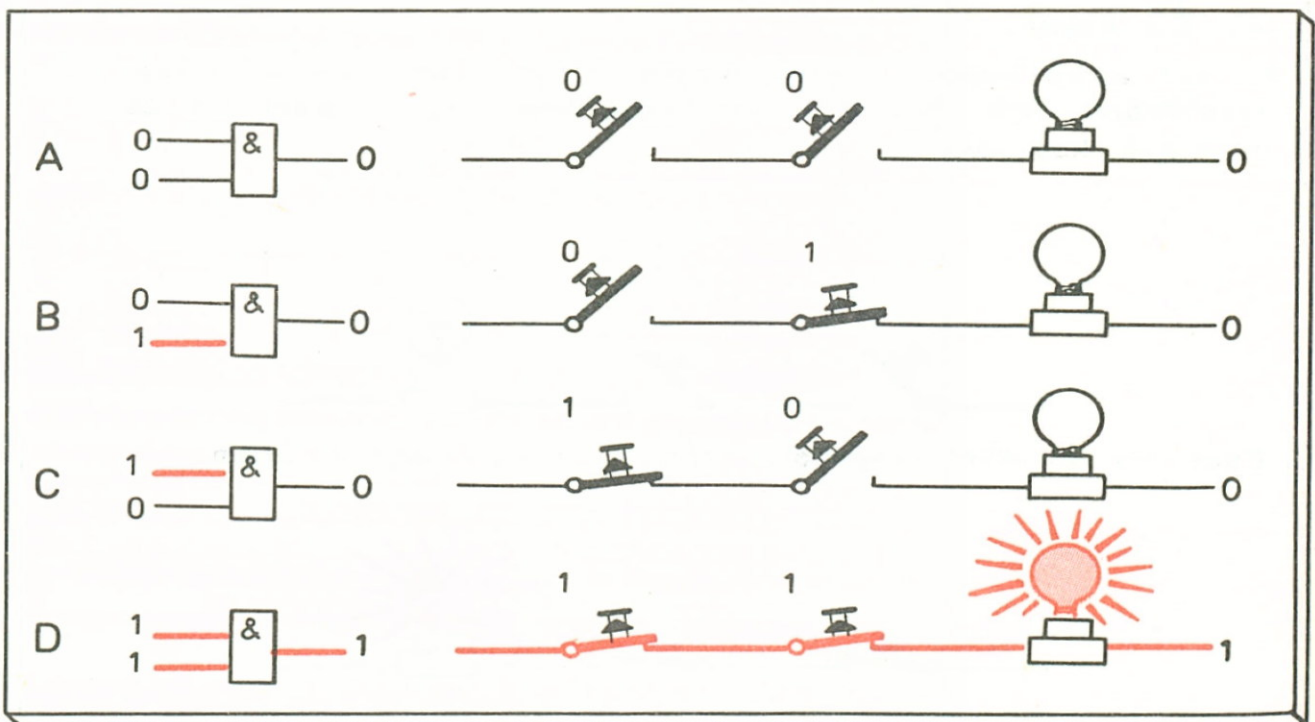
vagy a két bemenet egyikén van feszültség (B, C),

vagy mindkét bemeneten van feszültség (D).

Ha az ÉS kapu egyik bemenetén van feszültség (amit 1-gyel jelölünk), akkor az a billenőkapcsoló egy zárt kapcsolóállásának felel meg.

A jelképben a bemenetek száma és a billenőkapcsoló kapcsolóinak száma megegyezik.

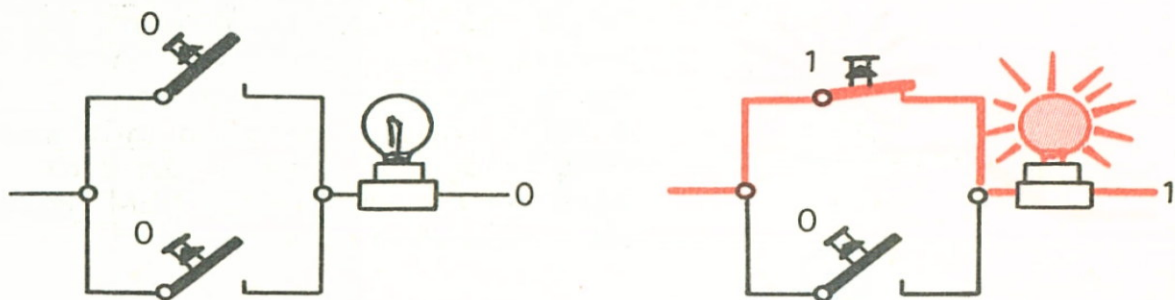
Az ÉS kapu bemeneteinek száma különféle lehet. Azonban mindig csak egyetlen kimenet van.



Csak ha valamennyi bemeneten feszültség van, ha tehát a billenőkapcsoló-modell minden kapcsolója zárva van, akkor folyik a kimeneten is áram.

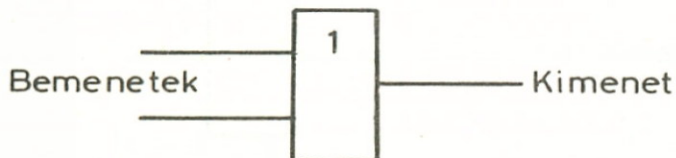
A VAGY kapu:

Egy VAGY kapu kimenetén akkor folyik áram, ha legalább az egyik bemenetén áram van. Billenőkapcsoló modellként egy VAGY kaput így ábrázolhatunk:



Ha a párhuzamosan kapcsolt kapcsolóknak legalább egyikét zárjuk, csak akkor folyhat áram.

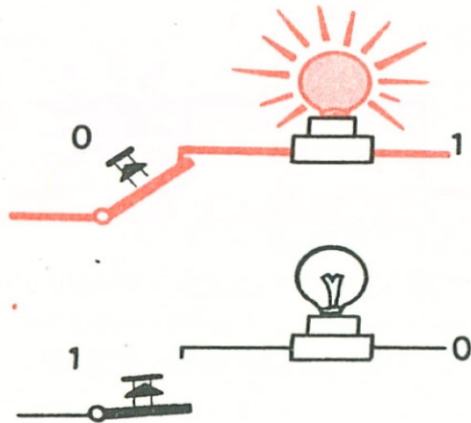
A VAGY kapu ábrázolására a következő jelképet használjuk:



Ha a kapcsolásnak legalább egyik bemenete áramot vezet, akkor a kimeneten is áram folyik.

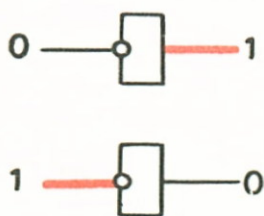
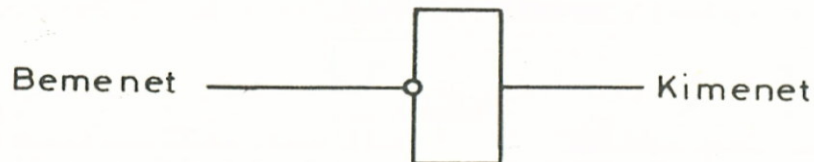
A NEM kapu:

a NEM kapu esetén az érkező jelet invertálja. Ha a bemeneten 0 van, akkor a kimeneten 1 jelenik meg, ha ezzel szemben a bemeneten 1 van, akkor a kimeneten 0 jelenik meg.



A billenőkapcsoló-modellben áram folyik, ha a kapcsolót nem működtetjük, és nem folyik áram, ha a kapcsolót működtetjük. A NEM kapura példa a hűtőszekrény világítás, amely az ajtó zárásakor -- ami egy NEM kapcsolót működtet -- kiálszik.

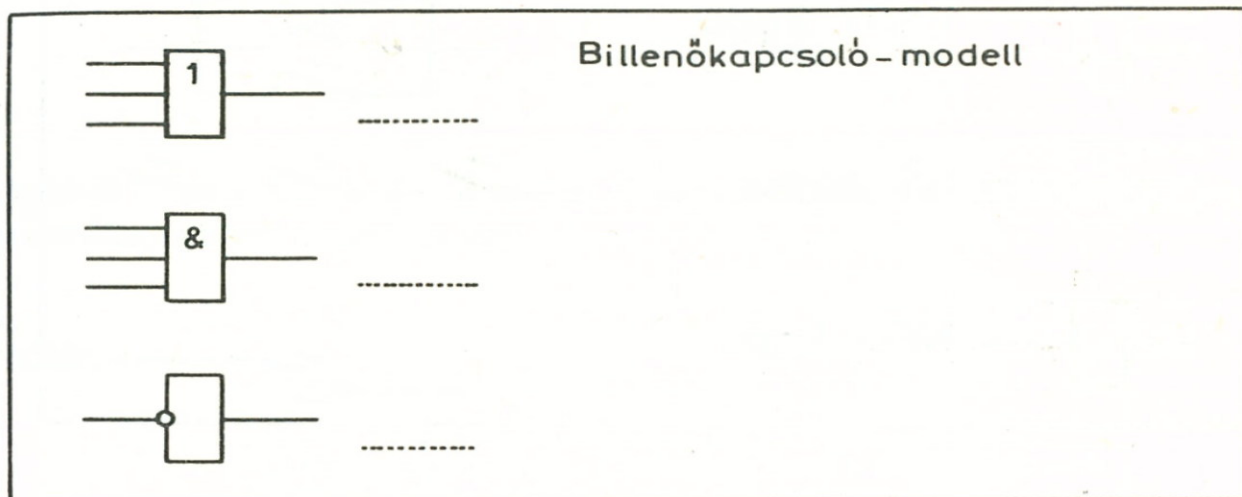
A NEM kapu jelképe:



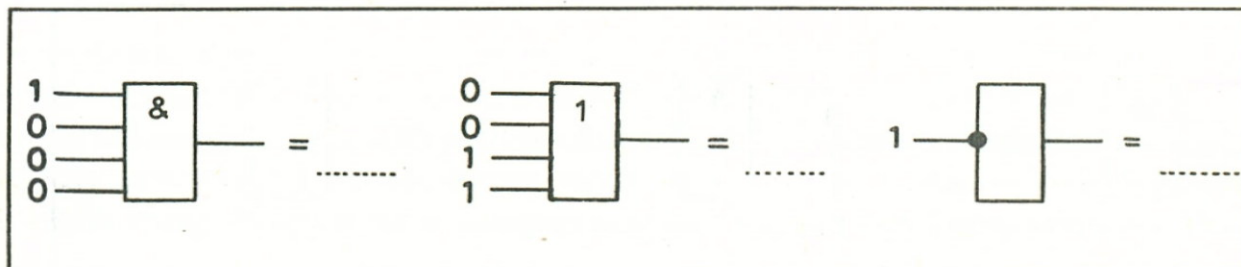
A bemenő jelet a NEM kapu invertálja. 0-ból 1 és 1-ből 0 lesz.



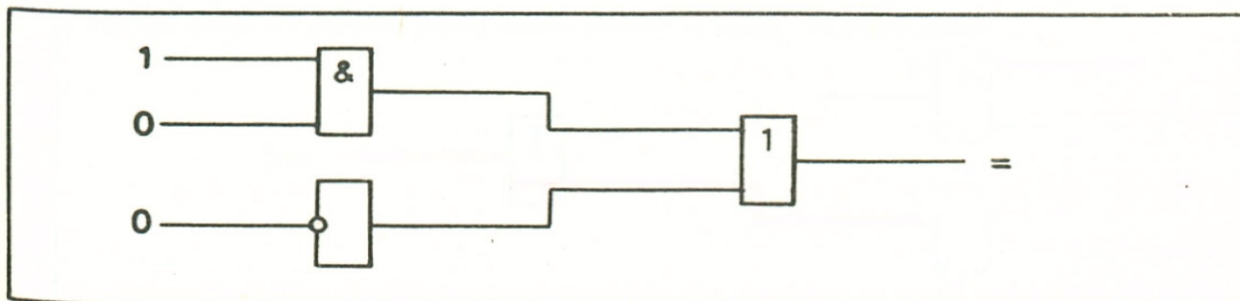
1. A következő kapcsolási jelképek melyike ábrázol **ÉS**, melyike **VAGY** és melyike **NEM** kaput?
Rajzolja be a megfelelő billenőkapcsolókat is a jelképek mellé!



2. Az alábbi kapuk kimenetein van (1) vagy nincs (2) áram?
Ennek megfelelően írjon egy **1**-est vagy **0**-t a kapuk kimenetéhez!



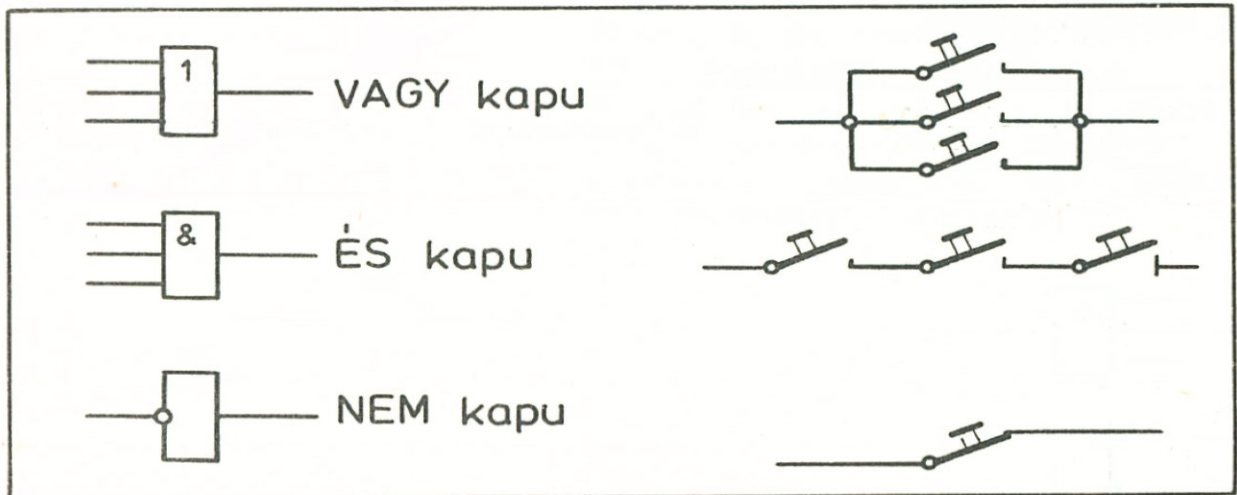
3. Milyen jel jelenik meg az alábbi kapu kimenetén?



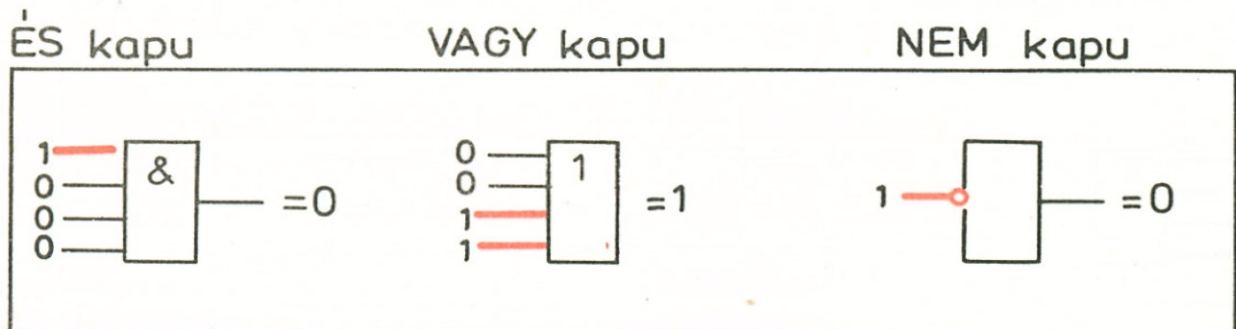


Válaszok

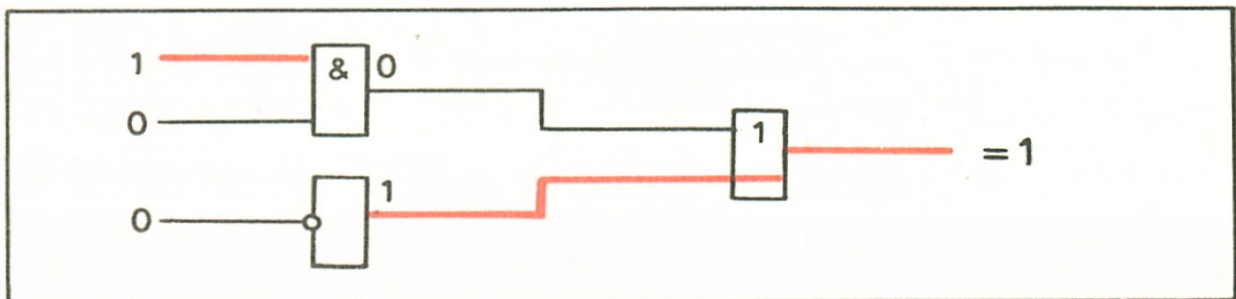
1.



2. A jelképek a következők:

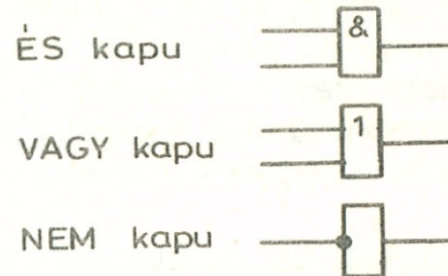


3. A kapu kimenetén van áram (1).



Összetett kapcsolások

E három alapkapu kombinációjával



bonyolult kapcsolások építhetők fel, ahogyan azokra az adatfeldolgozás során a számítógépben szükség van.

Példák:

Dekódoló kapcsolások
Összehasonlító - (komparátor) - kapcsolások
Összeadó kapcsolások
Tároló kapcsolások

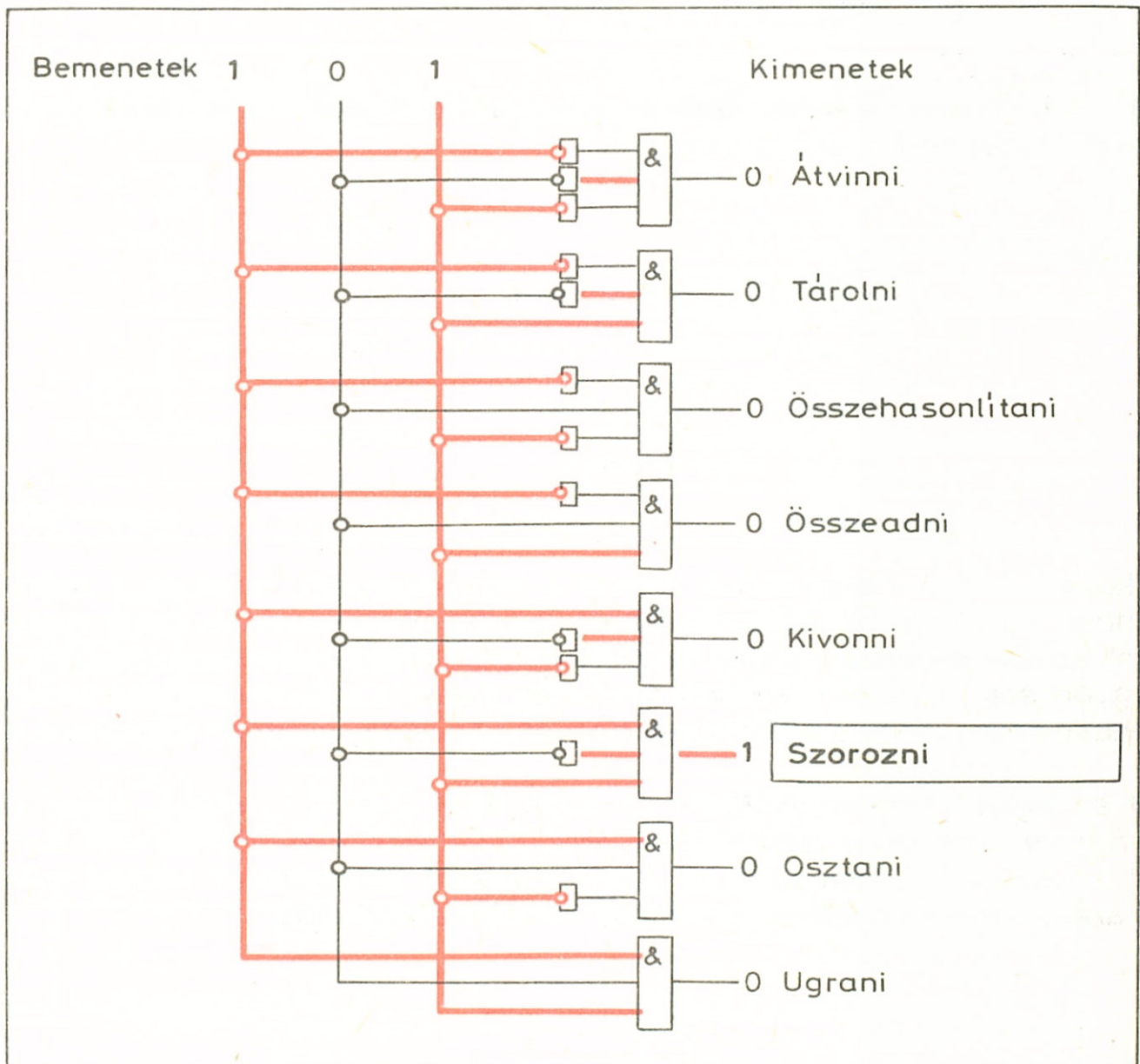
A következőkben bemutatjuk azokat a kapcsolásokat, amelyek a számítógépben különféle helyeken és különböző céllal előfordulnak, hogy rávilágítsunk arra, hogy a számítógép rendkívüli szolgáltatásai alapján véve nagyon-nagyon sok egyszerű lépésből tevődnek össze. Csak ennek felismerésével érthetjük meg, hogy a számítógép hogyan képes egy program végrehajtására.

Ha az egyes kapcsolásokban az áram útját követni szeretné, akkor csak a három alapkapu előzőekben leírt törvényeit kell figyelembe vennie. Az egyszerűség kedvéért az áramot vezető vezetékeket vastag (piros), az árammentes vezetékeket vékony vonallal ábrázoltuk.

Dekódolókapcsolásokra van pl. szükségünk a központi egység vezérlőegységében, a 0-kból és 1-esekből álló utasítások megfejtésére és vezérlőimpulzusokká való átalakítására.

Példa:

Az **101** gépi utasítást dekódolni (megfejtani) kell. A NEM kapcsolások és a három bemenetű ÉS kapcsolások célszerű kombinációjával elérjük, hogy mindig csak az egyik kimeneten van áram.

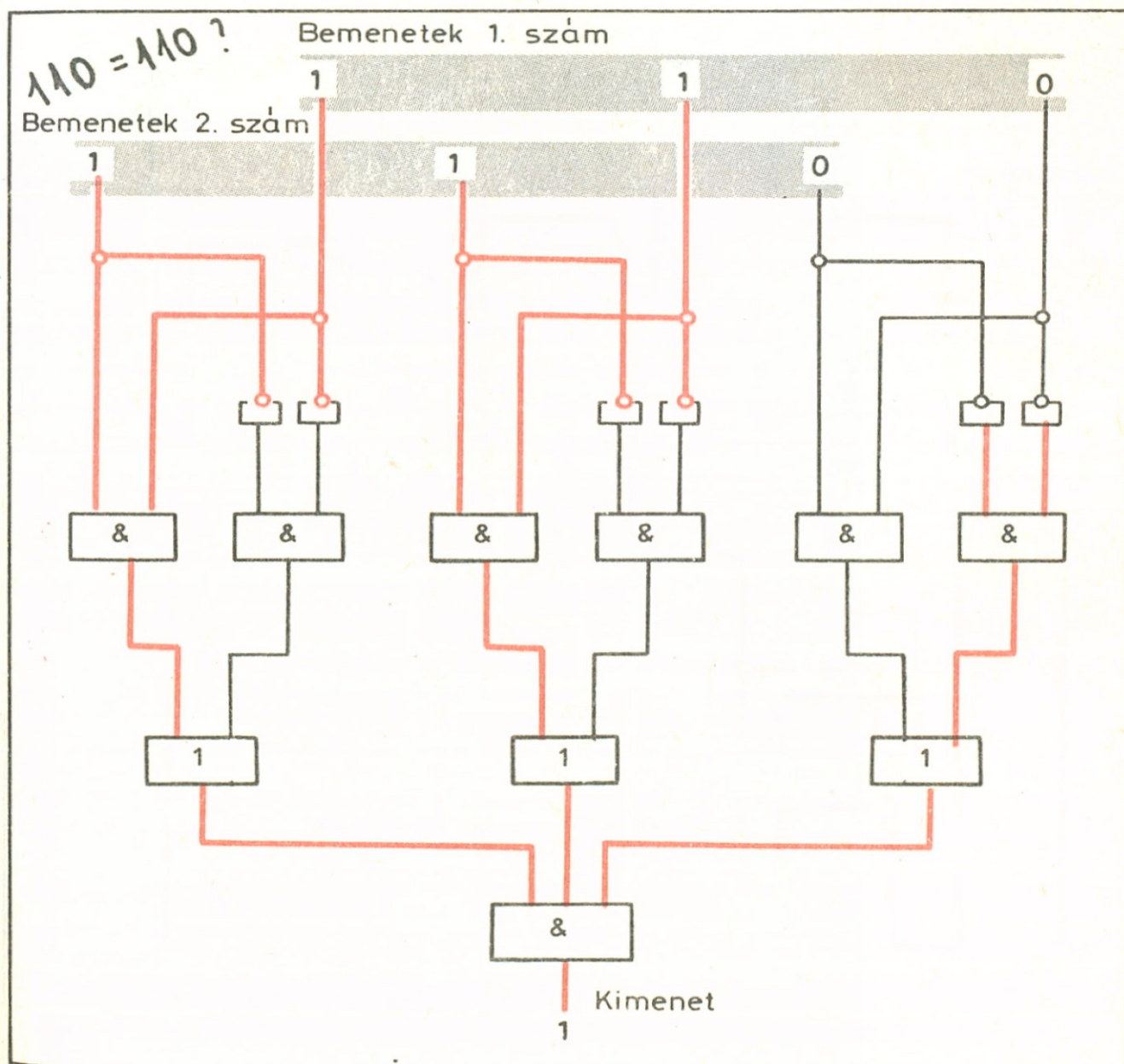


Ebben a dekódolókapcsolásban csak a hatodik ÉS kapcsolás kimenetén van áram, mert az ÉS kapcsolás mindhárom bemenetén feszültség van. Ez azt jelenti, hogy a megfejtendő **101** utasítás egy *szorzási utasítás*.

Összehasonlító-kapcsolásokra pl. a központi egység aritmetikai és logikai egységében van szükség, a döntések meghozatalához. Egy utasítás pl. (mindennapi nyelvi fogalmazásban) így szólhatna: „Hasonlítsd össze az 1. értéket a 2. értékkel, ha egyenlők, fejezd be a programot!” A két értéket az aritmetikai és logikai egység összehasonlítja. Ha egyenlők, akkor egy impulzus megy a vezérlőegységbe, amely a programot befejezi. Ha nem egyenlők, akkor az impulzus elmarad.

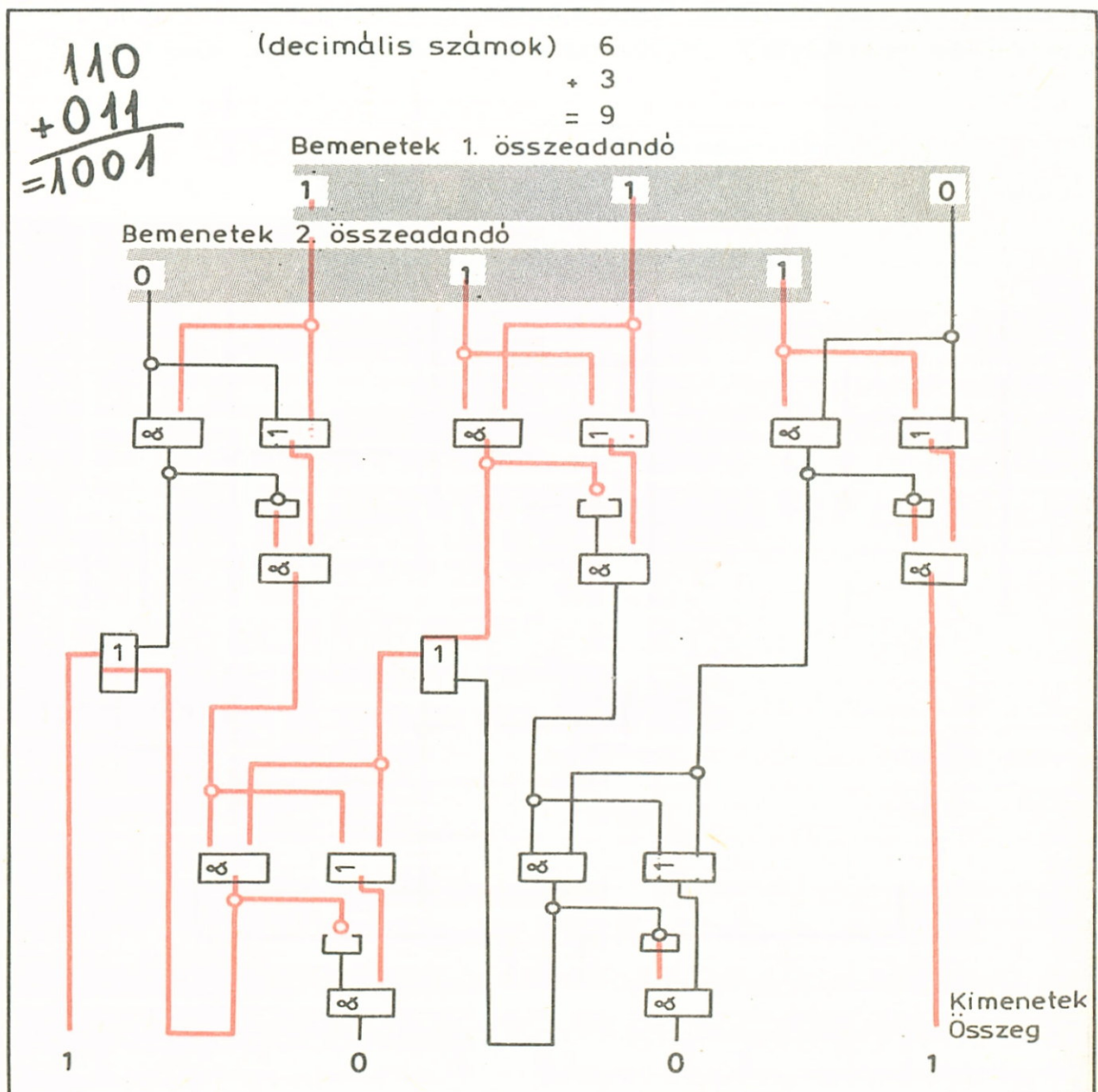
Példa:

Az **110** és **110** bináris számokat kell összehasonlítani. Az összehasonlító-kapcsolás úgy van felépítve, hogy a kimeneten áram folyik (**1**), ha a számok egyenlők, és nem folyik áram (**0**), ha a számok nem egyenlők.

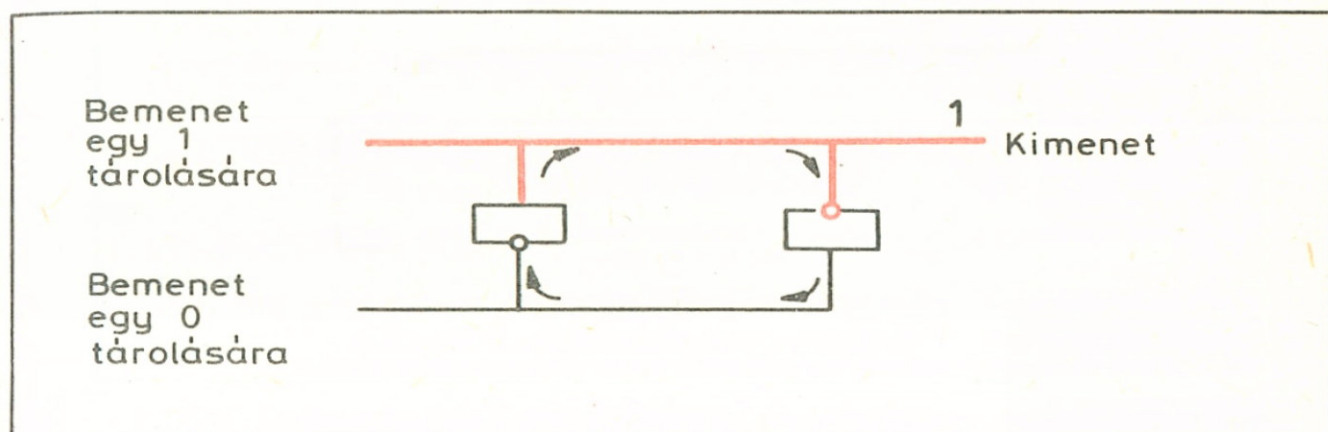


Az **összeadó-kapcsolásokra** számítások esetén van szükség. Elsősorban a központi egység aritmetikai egységében találhatók. Az összeadó-kapcsolások nemcsak összeadni tudnak, de a többi alpműveletet is el tudják végezni. A kivonás pl. egy negatív szám hozzáadásával történik, a szorzás pozitív számok többszörös összeadásával, az osztás negatív számok többszöri összeadásával.

A következő ábra egy egyszerű összeadó-kapcsolást mutat két, 3 számjegy-pozíciós bináris szám részére:

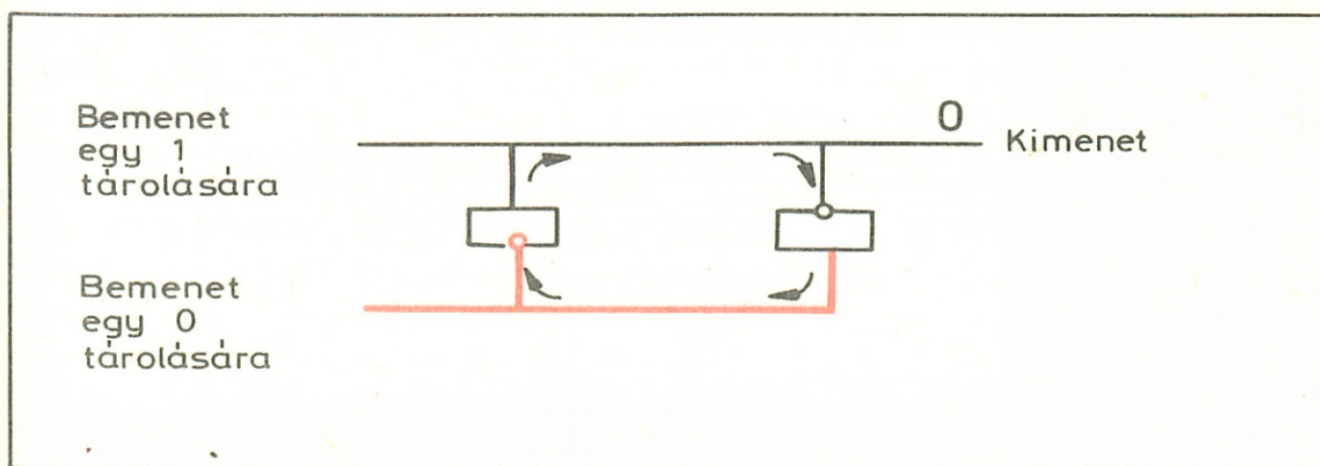


A **tárolókapcsolásoknak** az a tulajdonságuk, hogy egy állapotot (pl. áram megléte, ill. áram hiánya) rögzíteni tudnak. Erre a tulajdonságra ott van szükség, ahol adatokat kell tárolni, pl. az operatív tárban. Legegyszerűbb formája elvileg két, egymás mögé kapcsolt NEM kapuból állhatna, amelyek egy jelet átalakítanak (**1-et 0-ra és 0-t 1-re**), és azt körben mindig a másik NEM kapcsoláshoz továbbadják. A kimeneten mindig ugyanaz az állapot mérhető. Csak a két bemenet egyikére érkező rövid idejű impulzus tudja megváltoztatni a tárolt állapotot.



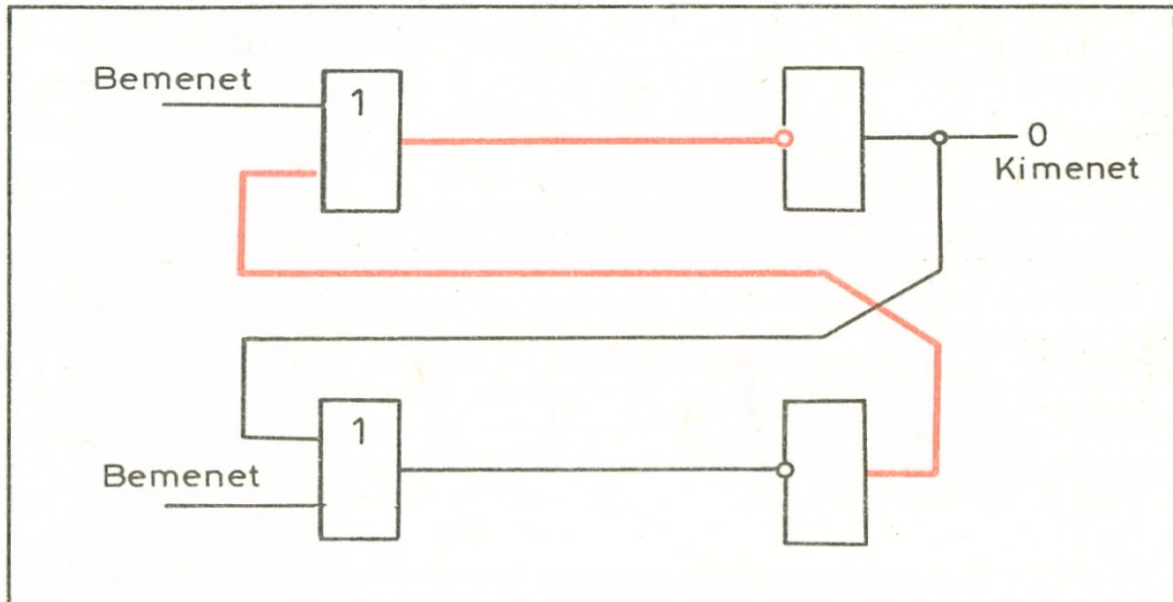
Az ábrázolt kapcsolásban az **1** állapot van tárolva, azaz a kimeneten áram folyik.

Csak ha „a bemenetre egy **0** tárolása céljából” rövid ideig egy impulzust adunk, a tárolókapcsolás „átbillen” (ahol ezelőtt nem folyt áram, ott most folyik, és megfordítva), és a kimeneten **0** jelenik meg. Az „átbillenés” tulajdonsága miatt a tárolókapcsolást **flip-flopnak** vagy **bistabil áramkörnek** nevezzük.



Ennyit a tárolókapcsolás elvéről, amelyet didaktikai okokból egy kapcsolás-modellen mutattunk be.

A gyakorlatban a tárolókapcsolásokat kissé másképpen ábrázoljuk:



Ebben az ábrázolásban a flip-flop csak akkor billen, ha az alsó bemenetére egy impulzus érkezik.

Kísérelje meg a kapcsolásfunkciók átgondolását ezen, a gyakorlatnak megfelelő kapcsolási rajzon, amely ugyanúgy működik, mint a modellbeli leegyszerűsített ábrázoláson!

Ezek után a következő fejezetben leírjuk, hogy ezek a kapcsolások műszakilag hogyan valósíthatók meg.

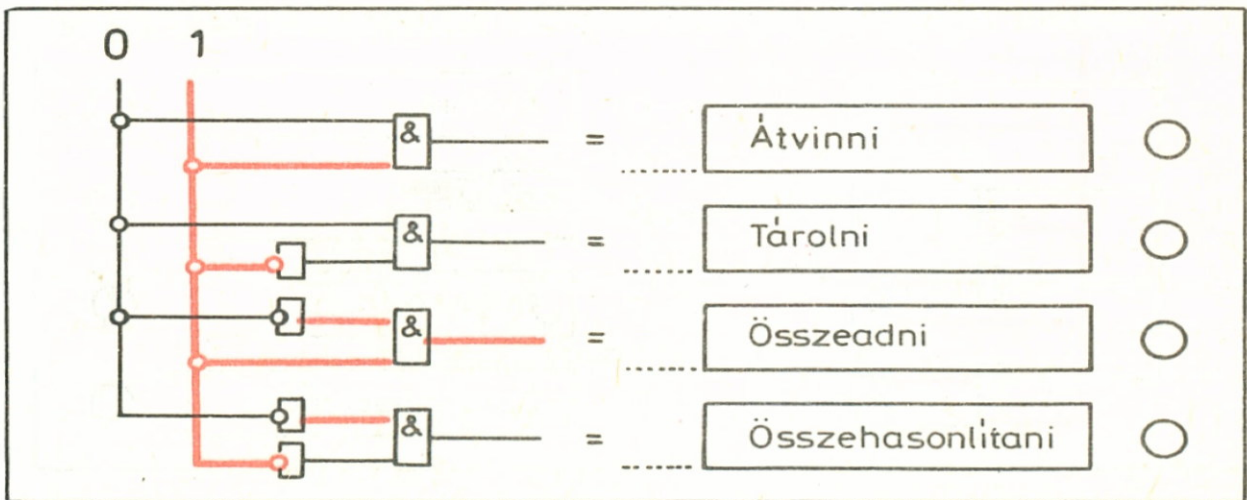


1. A központi egység mely helyein van szükség főként a következő kapcsolásokra:

- Dekódolókapcsolás.
- Összehasonlító kapcsolás.
- Tárolókapcsolás.
- Összeadó-kapcsolás:

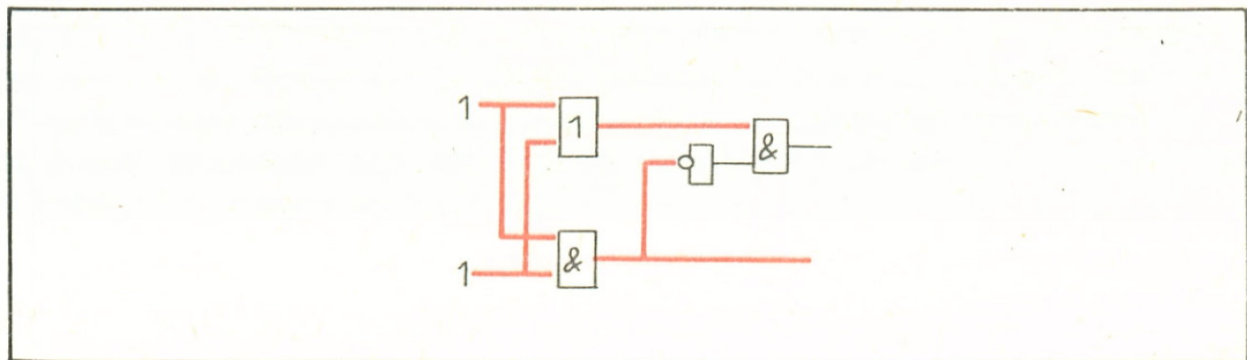
2. Milyen kapcsolást mutat a következő ábra

és melyik utasítást jelöli ki
 a **01** kódolás?



3. Végezze el az ábrázolt kapcsolás segítségével két bináris szám összeadását!

$1 + 1 = \dots\dots\dots$





Válaszok

1. A központi egység alábbi helyein van szükség a következő kapcsolásokra:

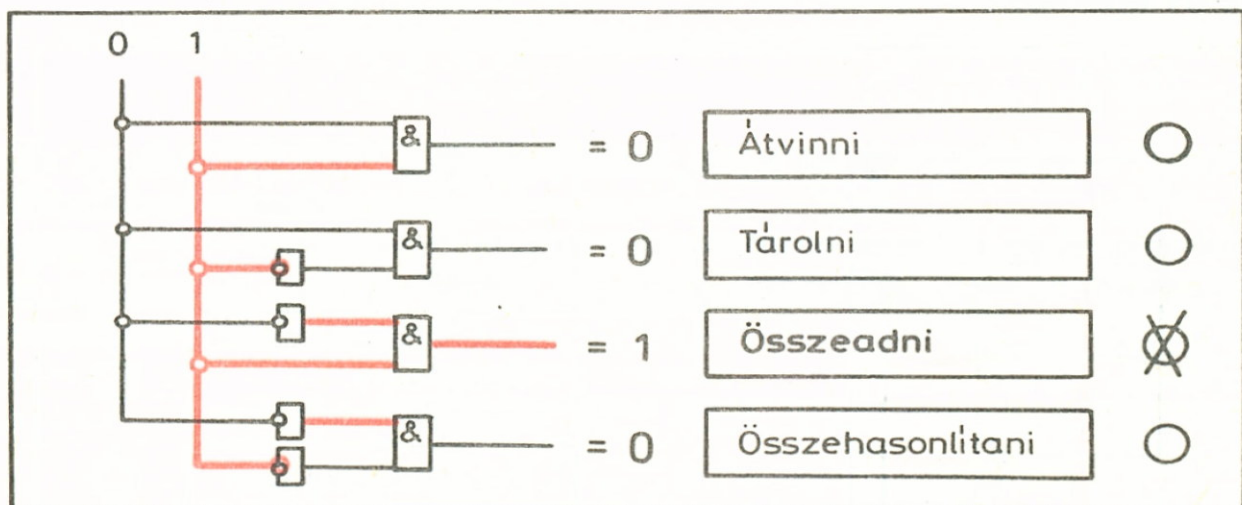
Dekódolókapcsolás: **Vezérlőegység.**

Összehasonlító-kapcsolás: **Aritmetikai és logikai egység.**

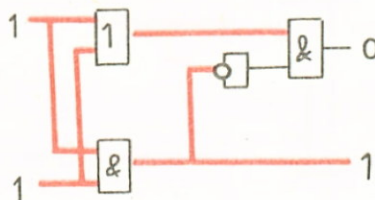
Tárolókapcsolás: **Operatív tár, regiszterek.**

Összeadók-kapcsolás: **Aritmetikai és logikai egység.**

2. A következő ábra egy dekódolókapcsolást mutat, és a **01** kódolással egy összeadási utasítást jelölünk ki.



3. Az ábrázolt kapcsolás segítségével egyszerűen követhető, hogy a két bináris szám összeadása: $1 + 1 = 10$ eredményt ad (decimális számokkal: $1 + 1 = 2$).



A számítógép-áramkörök megvalósítása

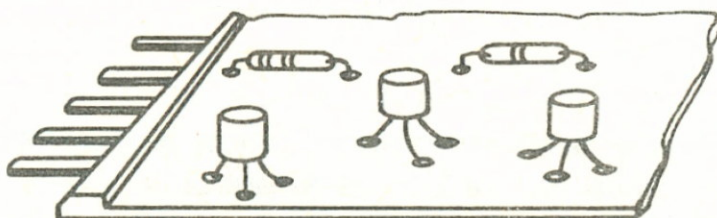
Az ismertetett kapcsolásokkal a számítógép többek között a következőkre képes: utasítások értelmezésére, összehasonlítások végrehajtására, számítások elvégzésére és tárolásra. Alapjában véve mást nem is kell tudnia a programok végrehajtása érdekében.

A következőkben leírjuk, hogy ezeket a kapcsolásokat hogyan lehet megvalósítani.

Az áramkörök felépítése

A számítógépben nincsenek mechanikus kapcsolók. A kapcsolási funkciókat a **tranzisztorok** veszik át. Egy VAGY kapcsolást pl. három tranzisztorral lehet megvalósítani.

A 60-as években használatos technika ezt szemléletesen mutatja:



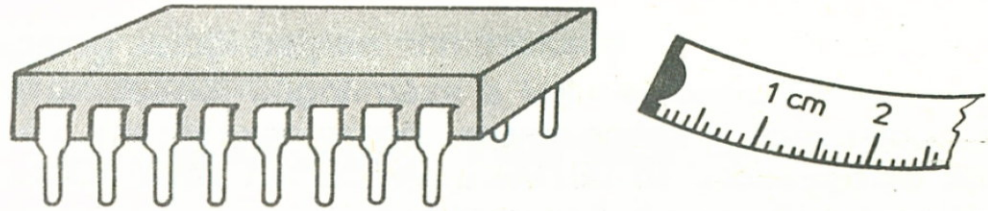
Tranzisztoros áramkör

Az ábrázolt elektronikus alkatrészek elhelyezéséhez legalább kb. 2×2 cm felületre, tehát egy egyszerű összeadó-kapcsolás több mint 50 tranzisztorának elhelyezéséhez kb. 10×10 cm területre volt szükség.

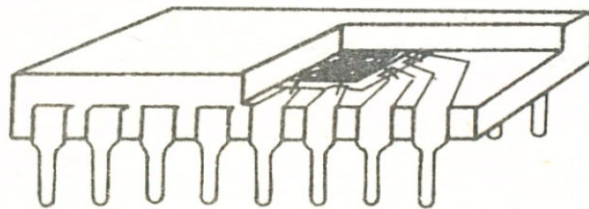
A mai félvezető-technikában az egyes tranzisztorok már nincsenek önálló házba beépítve. Bonyolult gyártási eljárással ma már a *tranzisztorok száz-ezreit* tudjuk egy négyzetcentiméternél kisebb területen elhelyezni.

Mivel a sok áramkört egy kis lapkán (chip) sűrítjük össze — integráljuk —, ezért **integrált áramkörökről** beszélünk.

Ez a kis lapka, amelyen a tranzisztorok százezrei vannak integrálva, egy néhány centiméter hosszúságú elektronikus alkatrészbe (integrált áramkör) van biztonságosan beépítve. Bizonyára Ön is látott már ilyet.



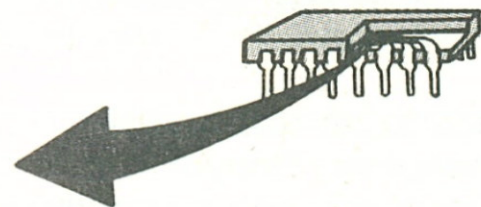
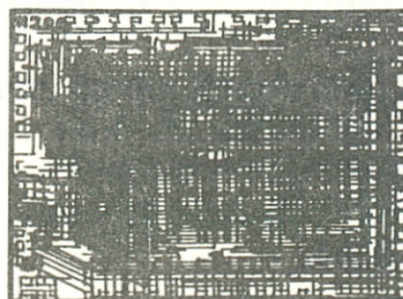
Ha ezt az elektronikus alkatrészt felvágnánk, akkor egy fél milliméter vastag és egy négyzetcentiméternél is kisebb szilíciumlapocska — az ún **lapka (chip)** — válna láthatóvá. Ezen a lapkán van a sok tranzisztor, dióda és ellenállás megvalósítva és a vezetékek bonyolult rendszerével egymással összekötve. A lapkát és a tok „lábait” rendkívül vékony aranyfonalak kötik össze. Az elektronikus alkatrész lábainak végei a lapka kimenetei, ill. bemenetei.



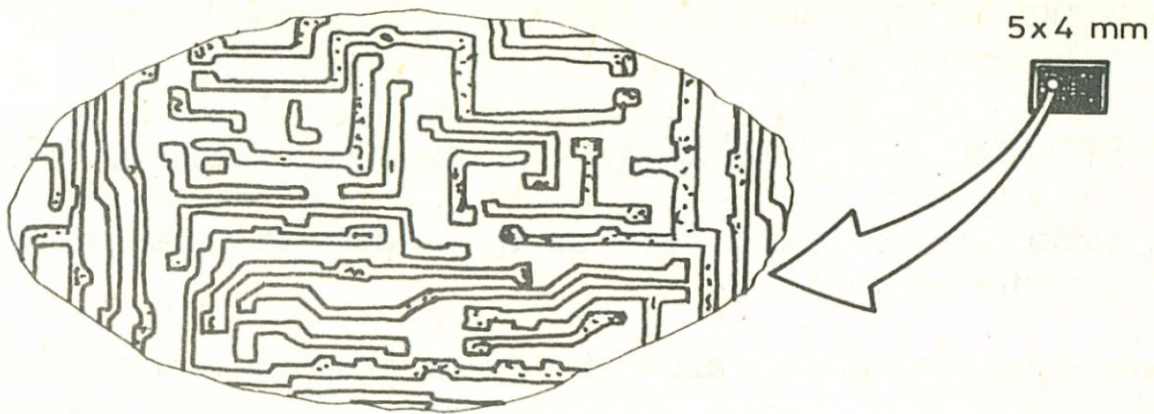
Ha egy lapkát mikroszkóp alatt vizsgálunk, akkor a vezetékek és foltok sokaságát láthatjuk, amelyek alatt az egyes kapcsolási elemek rejtőznek, ezek lényegében azok a tranzisztorok, amelyek az említett kapcsolási funkciókat elvégzik.

Nehezen elképzelhető, hogy egy lapka kb. 4×5 mm területén ma már több mint 100 000 áramköri elemet tudunk elhelyezni!

Eredeti nagyság 5x4 mm

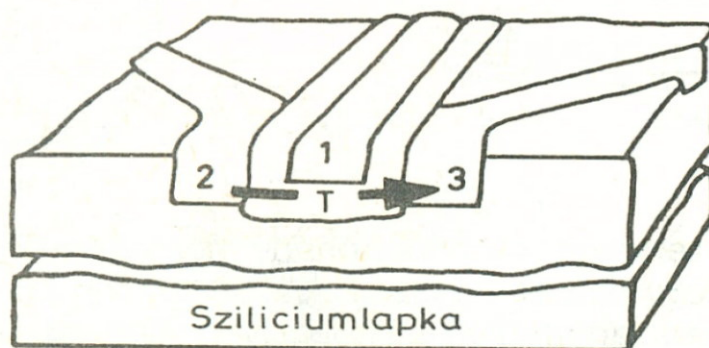


Még nagyobb mikroszkopikus nagyítással pl. az ábrázolt részletet láthatnánk, amely csak néhány tranzisztort és azok összekötését mutatja.



Ha átvágnánk a lapkát, és a vágási felületet nagy mikroszkopikus nagyítással szemlélnénk, akkor azt is felismerhetnénk, hogy a fél milliméter vastag lapka legfelső rétege alatt még további, sok vezetékkel átszőtt rétegek helyezkednek el.

A következő ábra a lapka egyetlen tranzisztorának metszetét mutatja erős nagyításban:



A tranzisztor tulajdonképpen csak három áramvezetőből áll, amelyek között **félvezetőréteg** (F) van.

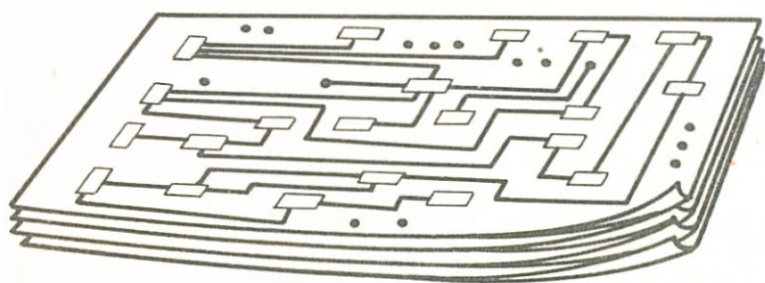
A félvezető-technika „titka” lényegében abban áll, hogy egy vezetéken, a *bázison* áramot vezetünk át (1). Ezáltal megváltozik a félvezetőréteg (F) áramvezető képessége, és az a réteg, amely előzőleg nem vezette az áramot, most vezetővé válik. Ez teszi lehetővé, hogy a második elektródától, az *emittertől* a harmadikhoz, a *kollektorhoz* (3) áram folyjék. A tranzisztor így az áram hozzávezetése által átveszi a leírt kapcsolási funkciót. A két elektróda (az emitter és a kollektor) közötti áramot a bázisba vezetett nagyon kis nagyságú árammal vezéreljük.

Ha abból indulunk ki, hogy egy alapkapcsolás felépítéséhez (ÉS—VAGY—NEM kapuk) az alkalmazott technika szerint átlagosan három tranzisztor szükséges, akkor egy összeadó-kapcsoláshoz, amely két, 16 számjegyes bináris számot tud összeadni, 400-nál több ilyen tranzisztor kell.

Az áramkörök gyártása

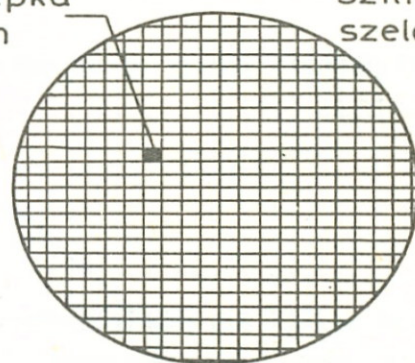
A félvezetőlapkák előállítását csak az új, *modern tervezési és gyártástechnológia* tette lehetővé.

Egy félvezetőlapka előállítása azoknak a kapcsolásoknak a megtervezésével kezdődik, amelyeket a lapkán el kell helyezni. Az áramköri terveknek ezt követő kifejlesztése és megrajzolása számítógép segítségével történik, amely az egyes áramkörök optimális elrendezését kiszámítja, és a több négyzetméter nagyságú terveket egy rajzgéppel megrajzoltatja.



Egy lapka
5x4 mm

Szilícium
szelet

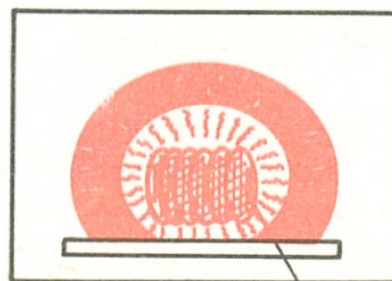


Kapcsolás

Ezeket az áramköri terveket ezután néhány négyzetmilliméternyi felületre lekicsinyítik, és *fotolitográfiai eljárással* egy szilíciumlemezre (vagy valamely más félvezetőre) átviszik, amely 100-nál több ilyen „leendő lapkát” tud felvenni.

A lapkák tulajdonképpeni előállítása az integrált áramkörök egyes rétegeinek felviteléből áll, ami bonyolult gyártási eljárásokkal, mint pl. maratással, diffúzióval, ionimplantálással stb. történik, amit itt részletesebben nem kívánunk leírni.

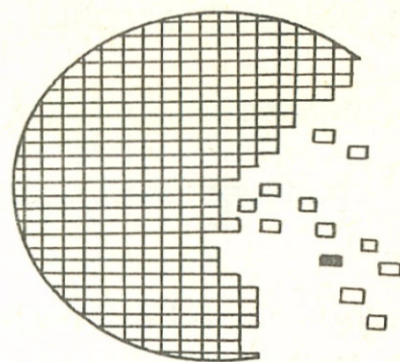
Bonyolult gyártási eljárás



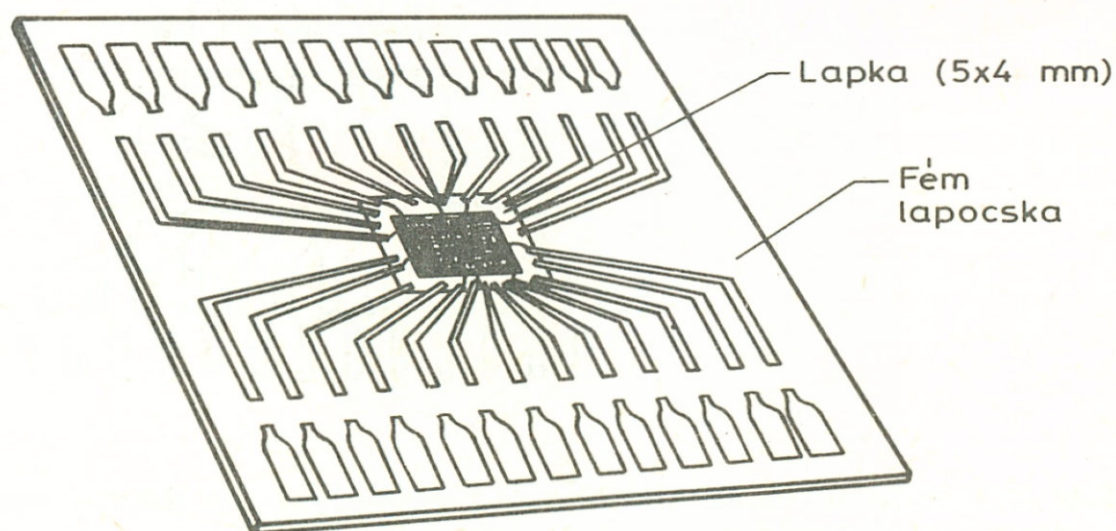
Szilícium szeletek

A szilíciumszeleten (wafer) elhelyezkedő lapkákat ezután egyenként ellenőrzik. Ezt is számítógép végzi, mert lapkánként számtalan funkciót kell ellenőrizni.

Ezután következik a szilíciumszeleteknek az egyes lapkáknak megfelelő szétvágása.

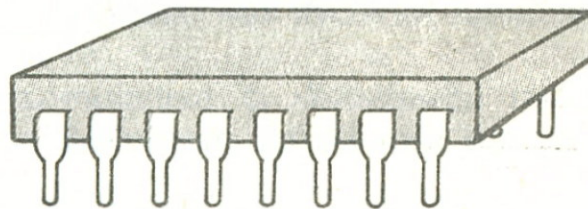


A használható lapkákat egy kis préselt lemezre ragasztják, amelyen a lapka be- és kimenetei már felismerhetők. A lapka ki- és bemeneteit és a lábait vékony aranyhuzalokkal *elektromosan összekötik*, ami a rendkívül kis méretek miatt a legnagyobb fokú precizitást követeli meg.

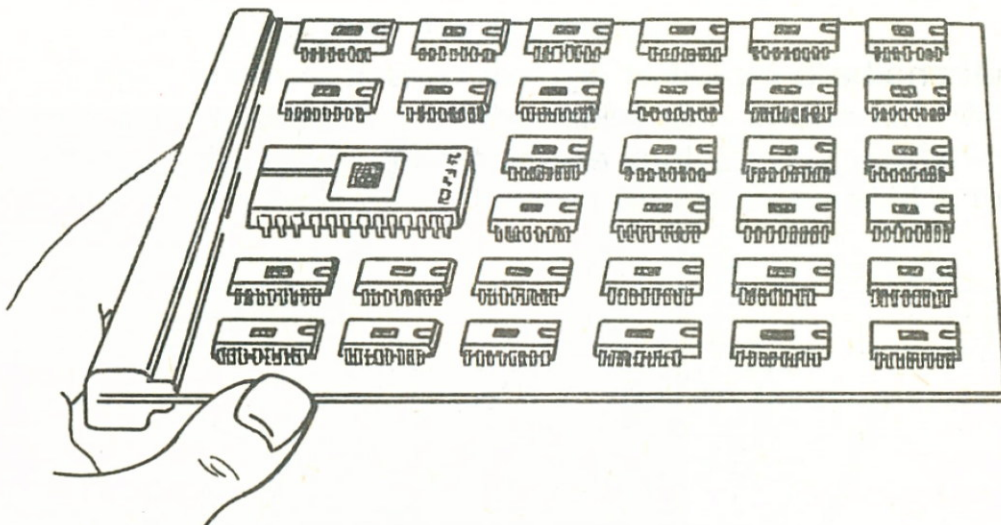


Egy további munkamenetben a lapkát műanyag vagy kerámia házba foglalják, és a lábait szögben lehajlítják. Ezáltal egy félvezető-alkatrész keletkezett.

Egy befejező vizsgálat után az alkatrészek felhasználhatók.



Több ilyen alkatrészt a kivezetéseikkel nyomtatott áramköri kártyára forrasztanak. Ilyen kártyákat helyezünk el ezután a számítógép „házába”.

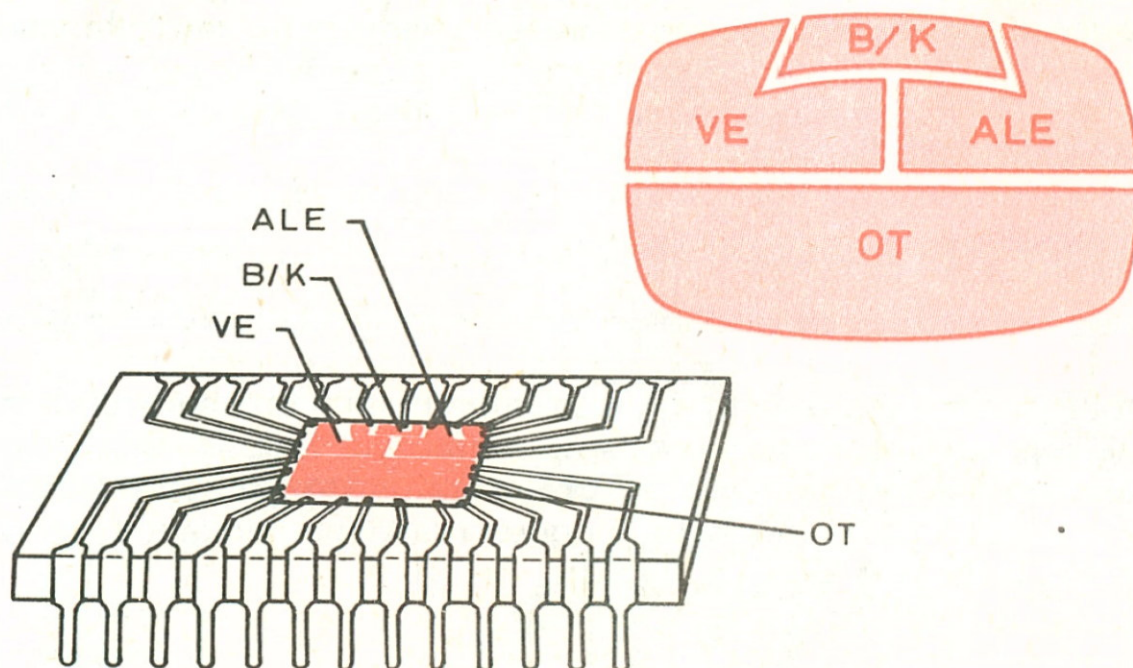


A számítógép műszaki felépítése

A számítógép központi egysége rendszerint félvezető-alkatrészekből áll. A processzor — tehát a vezérlőegység, valamint az aritmetikai és logikai egység — funkcióit eközben a gép nagyságától függően csak egyetlen vagy kevés számú félvezető-alkatrész látja el.

Az *operatív tár* ezzel szemben a konstrukciótól függően többnyire néhány vagy akár nagyon sok önálló, különálló elektronikus alkatrészből állhat.

Vannak azonban mikroszámítógépek, amelyek csak egyetlen lapkából állnak. Az ilyen **egylapkás számítógépen** a központi egység egységei azonban mégis megkülönböztethetők. Tehát figyelmesen szemlélve felismerhető a vezérlőegység, az aritmetikai és logikai egység, a be- és kiviteli vezérlőegysége és az operatív tár.



Egyébként ennek az egylapkás számítógépnek a felépítéséből fejlesztettük ki a központi egység hardverének könyvsorozatunkban használt, és azóta biztosan már Ön által is jól ismert *szimbólumát* is.

A központi egységben *dekódoló-, összeadó-, összehasonlító- és tárolókapcsolásokra* van szükség, amelyek a félvezető-technikában mint integrált áramkörök vannak realizálva. A következő oldalon az ábra ezeket a kapcsolásokat mutatja be, nagymértékben leegyszerűsítve és a központi egység egyes részeihez rendelve.

Dekódolókapcsolásokra főként a vezérlőegységben van szükség az utasítások megfejtéséhez, de az operatív tárban is a tárcímek dekódolásához.

Összeadó-kapcsolásokat túlnyomó részben a központi egység aritmetikai és logikai egységében találunk.

Összehasonlító kapcsolásokra is szükség van az aritmetikai és logikai egységben.

A központi egységnek a *regiszterek* és más *gyorstárak* részére sok **tárolókapcsolást** kell tartalmaznia. Az operatív tár is nagyszámú tárolókapcsolásból épül fel. Mivel minden tárolandó bit részére egy tárolóelem kell, ezért pl. egy 8 millió byte kapacitású nagy operatív tárnak 64 millió tárolóelemből kell állnia.

A központi egység egyes részeit vezetékek, az ún. **adatbuszok** és **címbuszok** kötik egymással össze.

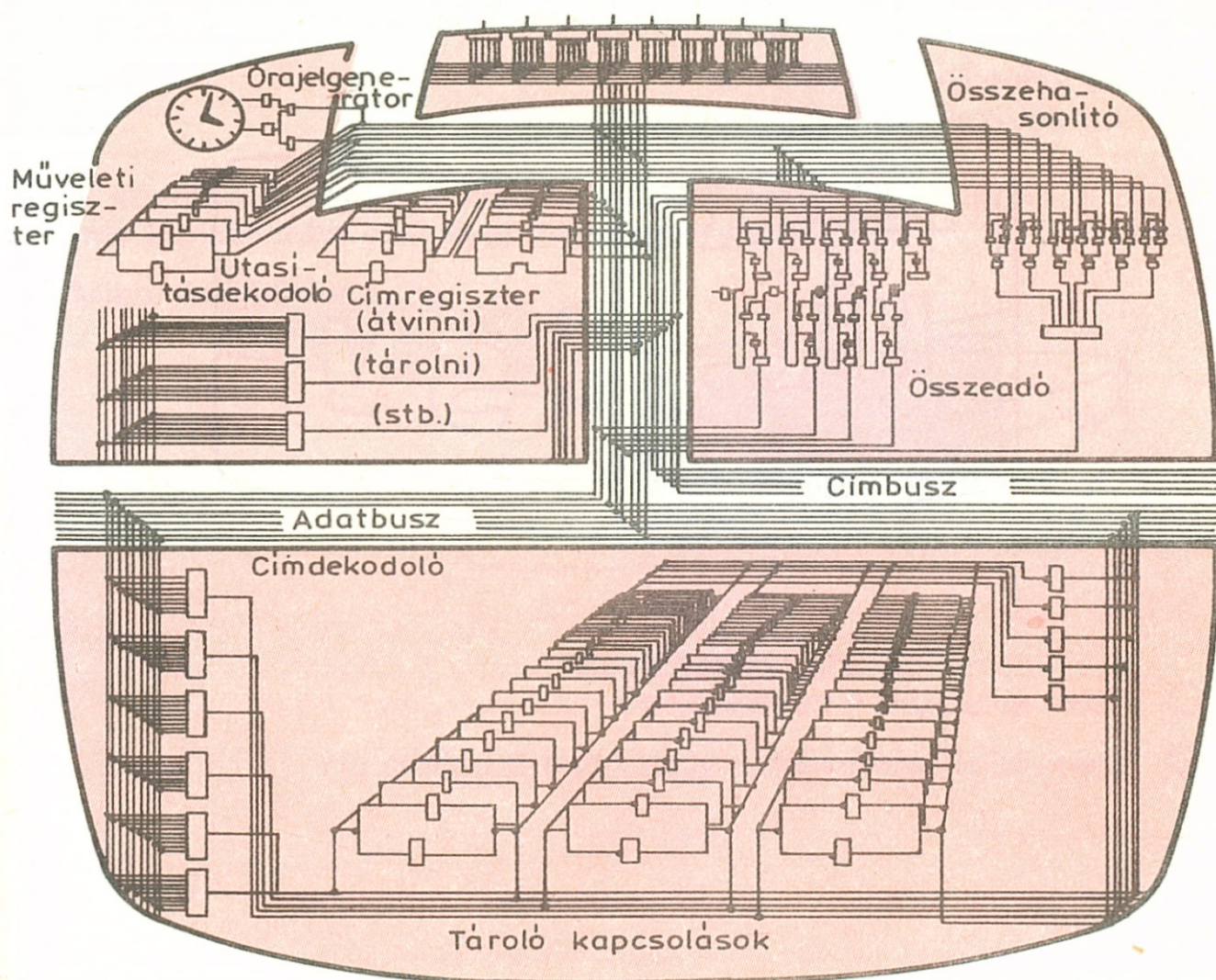
Nézzük meg most ennek az első tekintetre bonyolult hálózatnak a működését!

A hálózatban semmi esetre sem folyik folyamatosan áram, a központi egység minden akciója sokkal inkább **áramimpulzusokon** alapul, tehát rövid időtartamú áramlökéseken, amelyek az **órajel-generátortól** származnak. Ez a generátor a szóban forgó modellről függően másodpercenként néhány millió impulzust hoz létre.

Az ember részére elképzelhetetlenül rövid ütemperiódus alatt pl. egy byte az operatív tárból egy 8 vezetéket tartalmazó adatbuszon keresztül az aritmetikai és logikai egységbe kerül.

Ez a másodperc milliomod részénél is rövidebb idő elegendő pl. az utasításdekóderben egy utasítás megfejtésére, vagy egy összehasonlítás végrehajtására, ill. egy összehasonlítás eredményének a vezérlőegységbe való átvitelére.

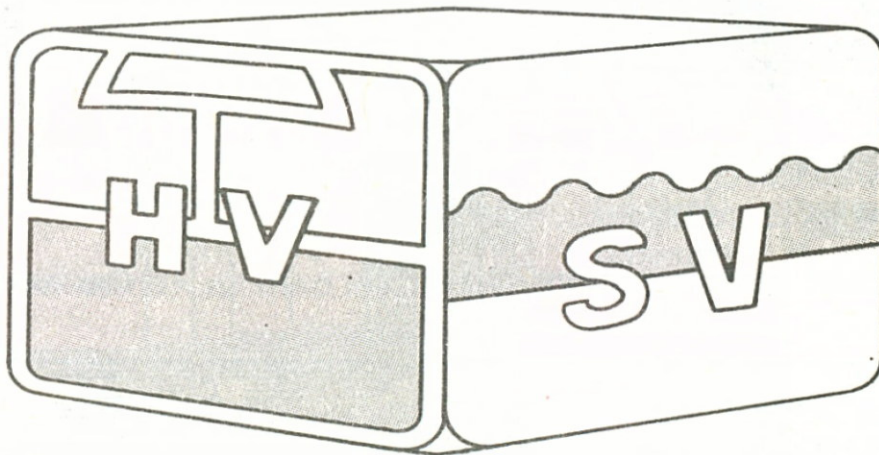
Ez ismét azt mutatja, hogy a számítógép szolgáltatásai sok-sok egyszerű műveletből tevődnek össze, amelyek elképzelhetetlen gyorsasággal követik egymást.



Ennyit a számítógép műszaki felépítéséről. Némi fantáziával most már elképzélhetjük, hogy a számítógép áramimpulzusok segítségével hogyan képes adatok értelmezésére (dekódolására), adatokkal való számításokra, adatok összehasonlítására, és ezzel azok felismerésére, hogyan tud adatokat rendezni és tárolni.

Felmerül azonban a kérdés, hogy az ember az utasításokkal — a programmal — miként tudja a számítógépet adatok feldolgozására készíteni.

Ehhez röviden meg kell vizsgálnunk a hardver és a szoftver kapcsolatát.





1. Melyik megállapítás helyes a félvezetőlapkára?

Egy lapkán tranzisztorok százezrei helyezhetők el.

A lapka egy olyan szilíciumlapocska, amely 1 centiméter vastagságú, és éppen 10 négyzetcentiméter felületű.

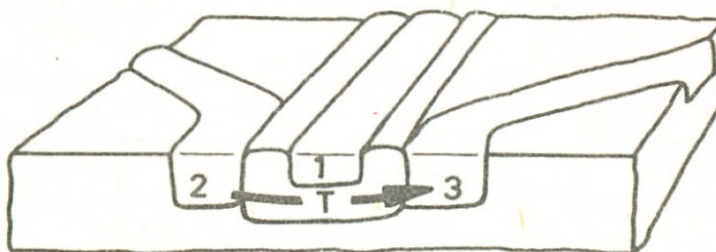
Minden lapka egy elektronikus alkatrészbe van beépítve, amelynek tokján lábak találhatók.

A lábak a lapka be- és kimenetei, amelyekkel azok pl. rendkívül vékony aranyvezetékekkel vannak összekötve.

A lapka több rétegből áll, amelyekben nagyszámú vezeték fut.

2. A következő ábra egyetlen tranzisztor metszetét mutatja, egy lapkán több ezer ilyen tranzisztor van.

Mikor folyhat áram?



Ha az 1-esen áramot vezetünk át és ezáltal az F vezetővé válik.

Ha csak a 2-esen vagy a 3-ason folyik át áram.

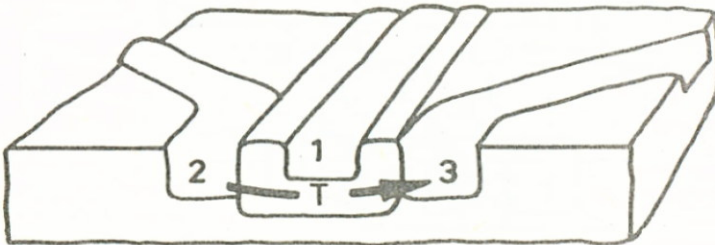


Válaszok

1. A félvezetőlapkára vonatkozó következő megállapítások igazak!

- Egy lapkán tranzisztorok százezrei helyezhetők el.
-
- Minden lapka egy elektronikus alkatrészbe van beépítve, amelynek tokján lábak találhatók.
- A lábak a lapka be- és kimenetei, amelyekkel azok rendkívül vékony aranyvezetékekkel vannak összekötve.
- A lapka több rétegből áll, amelyekben nagyszámú vezeték fut.

2. Az ábrán feltüntetett esetben, amely egyetlen tranzisztor metszetét mutatja (amelyből százezrek lehetnek egy lapkán), akkor folyik áram:



- ha az 1-esen áramot vezetünk át, és ezáltal F vezetővé válik.
-



3. Nevezze meg egy félvezető-alkatrész előállításának egyes lépéseinek helyes sorrendjét úgy, hogy az 1...4 számjegyeket a megfelelő téglalapba beírja!

Az integrált áramkörök rétegeinek felvitele bonyolult gyártási folyamatokkal egy szilíciumszeletre.

A lapkák ellenőrzése és a szilíciumszelet szétdarabolása az egyes lapkákra.

Minden áramkör megtervezése, megrajzolása és kicsinyítése, amelyeket a lapkára fel kell vinni.

Minden használható lapkát egy fémlemezre felragasztanak, annak be- és kimeneteit aranyvezetékekkel a tok lábaihoz forrasztják, és azt műanyag házba foglalják.

4. Hogyan nevezzük azokat a vezetékeket, amelyek a központi egység részeit egymással összekötik?

Ha azon adatokat továbbítanak:

Ha azon címeket továbbítanak:

5. a) Mi a feladata a számítógép órajel-generátorának?

.....

- b) Másodpercenként hány impulzust tudnak a legnagyobb teljesítményű, leggyorsabb órajel-generátorral generálni?

Max 10 000 impulzust másodpercenként.

Max 100 000 impulzust másodpercenként.

Max 1 000 000 impulzust másodpercenként.

Több mint 1 000 000 impulzust másodpercenként.



Válaszok

3. Egy félvezető-alkatrész előállításának lépéseinek helyes sorrendje:

- 2 Az integrált áramkörök rétegeinek felvitele bonyolult gyártási folyamatokkal egy szilíciumszeletre.
- 3 A lapkák ellenőrzése és a szilíciumszelet szétdarabolása az egyes lapkákra.
- 1 Minden áramkör megtervezése, megrajzolása és kicsinyítése, amelyeket a lapkára fel kell vinni.
- Minden használható lapkát egy fémlemezre felragasztanak, annak be- és kimeneteit aranyvezetékekkel a toklábakhoz forrasztják, és azt műanyag házba foglalják.
- 4

4. A központi egység részeit összekötő vezetékeket a következőképpen nevezzük:

Adatbusz, ha az adatokat továbbít.

Címbusz, ha az címeket továbbít.

5. a) Az órajel-generátor feladata: az órajelek előállítása.

b) A leggyorsabb ütemadók:

Másodpercenként több mint 1 000 000 impulzust tudnak generálni*.

* Valójában nem az órajel-generátorok teljesítőképessége korlátozza a számítógépek teljesítőképességét, hanem a processzor realizálható min. ciklusideje. Ehhez igazítják az órajel-generátor frekvenciáját. Jelenleg a legkisebb ciklusideje a CRAY-I-nek van, ennek értéke 12,5 ns. (Lektor.)

A hardver és a szoftver kapcsolata

Utasítások nélkül egy számítógép nem működik. A számítógép elvégzendő tevékenységeit az utasításokból álló **program** jelöli ki. Ezt rendszerint egy programozó írja egy adott **programnyelven** — ami a természetes emberi beszéd és a gépi nyelv (amit a számítógép „megért”) közötti nyelv.



Egy számítógép **gépi nyelve** a gép típusának függvényében száznál több különféle gépi utasításból is állhat. Az utasítások összességét a számítógép **utasításkészletének** nevezzük.

A programnyelv, amelyben a programozó a megoldandó feladat megoldását megfogalmazza, ezzel szemben csak kevés és könnyen érthető utasításból áll (miközben ezek mindegyike a gépi utasítások sorozatát váltja ki), feltételezve, hogy a programozó korszerű, problémaorientált nyelven programoz.

Utasítások a központi egységben

A kiválasztott programnyelven megírt programot a gép által olvasható formában **rögzítik**, és **beolvassák** a számítógép központi egységébe. Az utasítások ekkor egymást követően kerülnek az operatív tárba. Ahhoz azonban, hogy a központi egység az utasításokat megérthesse és végrehajthassa, azokat gépi nyelvre, tehát gépi utasításokra kell fordítani. Erre a célra speciális fordítóprogramokat készítenek, amelyek indításuk után önállóan lefutnak a számítógépben.

A fordítás után az utasítások gépi nyelven állnak az operatív tárban*. Két további feldolgozási lépés után: a szerkesztés után, amely különféle programrészeket egy egységbe foglal össze, és a betöltés után, amelynek során a program futásra készen az operatív tárba kerül, megindítható a tulajdonképpeni **végrehajtás** (produktív futás). Ez azt jelenti, hogy a felhasználói program utasításait a vezérlőegység — egymás után — lehívja az operatív tárból, és gondoskodik azok végrehajtásáról. A feldolgozásra szánt adatokat a számítógép eközben a program logikájának megfelelően feldolgozza.



* A lefordított program, amit tárgyprogramnak is neveznek, gyakran a háttértárra kerül. (Lektor.)

Valamely utasítás *gépi nyelven* a 0-ák és 1-esek sorozatából áll. Az utasításbitek kszáma és jelentése az utasítás és a számítógép típusának függvényében változik.

A következő ábra egy 48 bit hosszúságú gépi utasítást ábrázol.

MIT	HOSSZ	HOVA	HONNAN	
1 1010010	00111001	01010000	00010110	01010100 11 011001
Műveleti rész	Operandusok hossza 1 2	1. operandus címe	2. operandus címe	

Az utasítás első része az elvégzendő **művelet kódját** tartalmazza, pl. add össze, hasonlítsd össze, ugorj egy adott címre, tárolj stb. Az utasításnak ezt a részét **műveleti résznek** nevezzük.

Az utasítás második része a feldolgozandó adatok (az operandusok) **hosszát** tartalmazza, azaz azt, hogy hány biten vannak az operatív tárban ábrázolva.

Az utasítás harmadik és negyedik részében közöljük a számítógéppel, hogy **honnan** jönnek az adatok, amelyekkel valamit végezni kell, azaz az operatív tárban milyen címen található a feldolgozandó adatok. Az 1. cím még rendszerint megadja, hogy hová kerüljenek az adatok. Ha pl. egy utasítás a következő: „add össze az 1. mező és 2. mező tartalmát”, akkor az eredmény az első cím alatt tárolódik. Vannak azonban utasítások, amelyek csak egy címet igényelnek, mint pl. „ugorj az x. címre”*.

* Valójában számos további ún. utasításformátum létezik. E bevezető könyv célja nem lehet ezek részletes bemutatása. (Lektor.)

A következő ábra egy leegyszerűsített gépi utasítást mutat. A jobb érthetőség érdekében a sok bináris 1-est és 0-át (bináris számjegyek) ember által olvasható kifejezések (decimális számok és betűk) helyettesítik. Ennek és még néhány következő utasításnak a bemutatásával ismertetjük a központi egységben lezajló feldolgozás egymás utáni szakaszait.

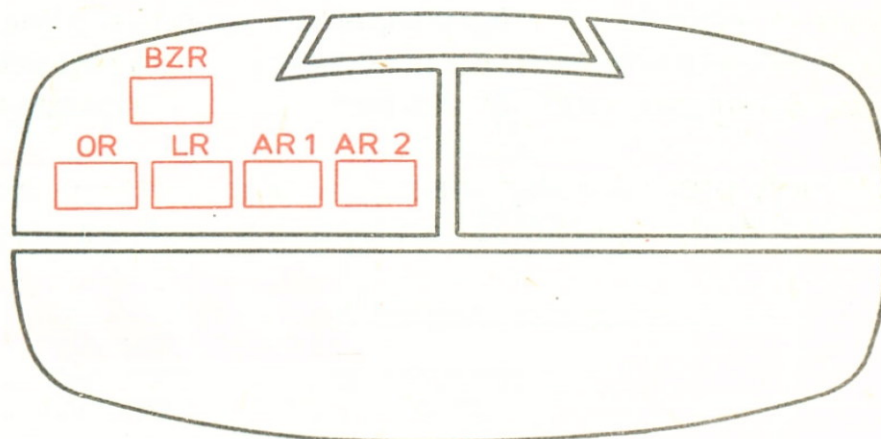
MIT	HOSSZ	HOVA	HONNAN
ADD	2 2	1480	1482
Műveleti rész	Operandumok hossza 1 2	1. operandus címe	2. operandus címe

Az utasítás legyen a következő:

„Add hozzá az 1. operandushoz, amely az operatív tárban az 1480 cím alatt található és 2 byte hosszúságú (tehát az 1480. és 1481. című tárolási helyen van) a 2. operandust, amely az operatív tárban az 1482. cím után következik és ugyancsak 2 byte hosszúságú, majd tárold az eredményt az 1480—1481. címeken”!

A számítógép vezérlőegységének felépítése és az utasítás szerkezete között jól látható pármegfeleltetés van. A vezérlőegység egyszerűsített ábrázolásában lényegében a következő regisztereket tartalmazza:

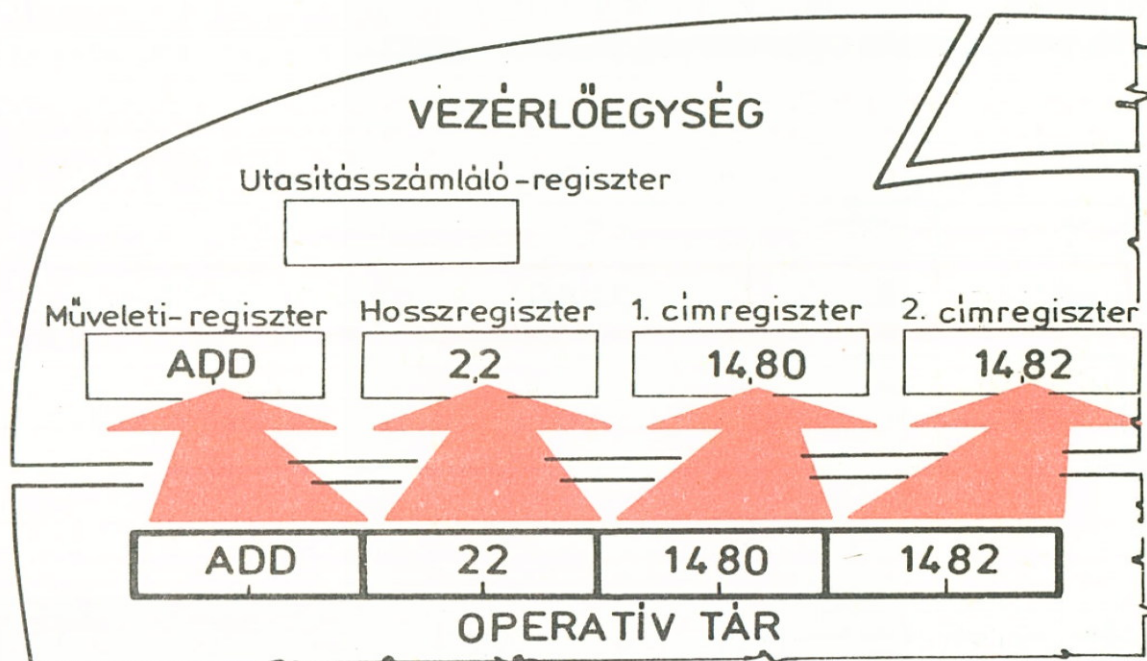
Műveleti regiszter (OR)
Hosszregiszter (LR)
1. címregiszter (AR 1)
2. címregiszter (AR 2)
Utasításszámláló-
regiszter (BZR)



A műveleti, a hossz- és a címregisztereknek kell az utasítás egyes részeit felvenni és kiértékelni.

A regiszterek, mint már említettük, kis kapacitású táruk, amelyek nagyon gyors tárolóáramkörökből épülnek fel.

Az utasítás végrehajtásához a vezérlőegység az utasítást először változatlan alakban veszi át az operatív tárból.



A műveleti kód tehát a műveleti regiszterbe, a hosszúságadatok a hosszregiszterbe és a két cím a címregiszterekbe kerül.

Különleges helyet foglal el az **utasításszámláló** regiszter, amely azt a címet tárolja, amelyen a végrehajtandó utasítás az operatív tárból megtalálható. Miután az utasítást a vezérlőegység átvette és végrehajtotta, az utasításszámláló regiszter tartalma a végrehajtott utasítás hosszúságával növekszik.

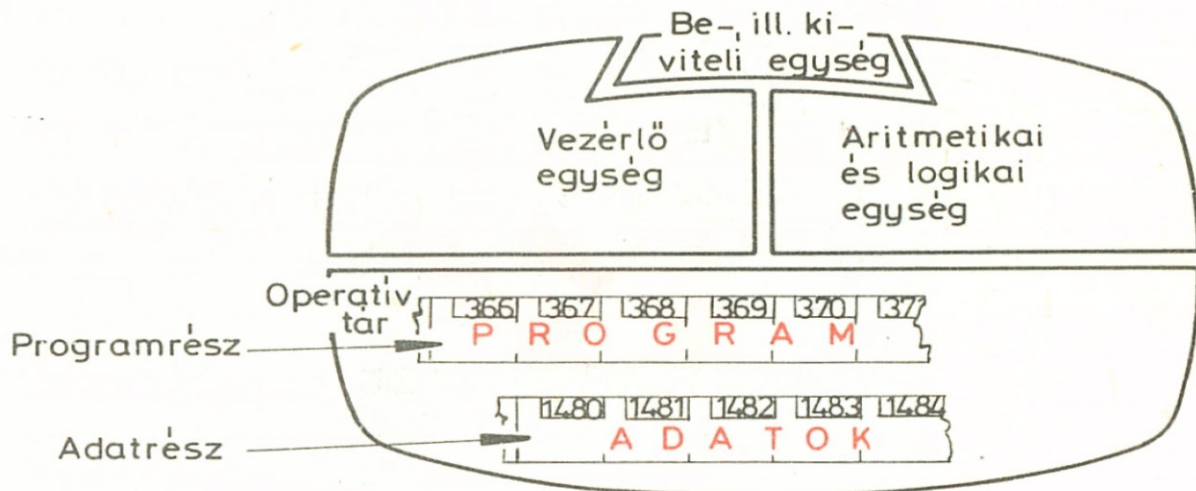
Ezzel a következő végrehajtandó utasítás címe jelenik meg az utasításszámláló regiszterben. Az utasításszámláló regiszter tehát mintegy *megjegyzi a programnak azt a helyét, amely legközelebb végrehajtásra kerül.*

Valamely program első utasításának *címét* egyébként a program betöltésekor a számítógép operációs rendszere adja át az utasításszámláló regiszternek.

Most bemutatjuk egy példán, hogy a program egyes utasításai a központi egységben hogyan kerülnek végrehajtásra.

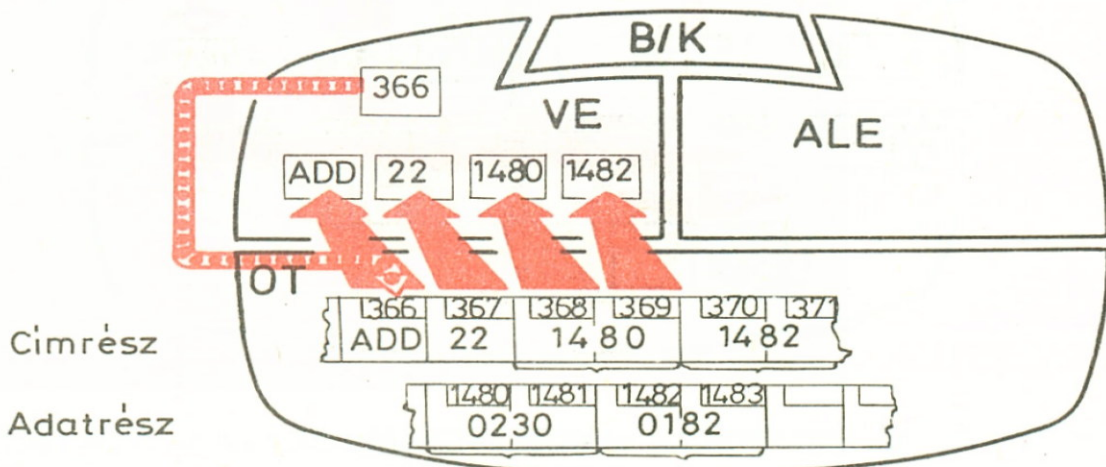
A következő helyzetből indulunk ki:

A (lefordított) program gépi nyelven van az operatív tárnak azt a részét, amely a programutasításokat felveszi, **programrésznek** nevezük. Ezzel szemben az operatív tár **adatrészében** a program által feldolgozandó adatok állnak.

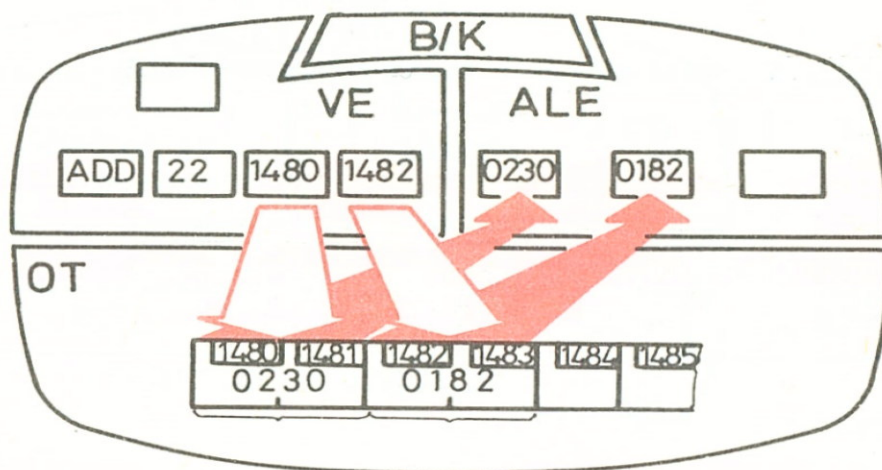


Az utasításszámláló regiszter, mint már említettük, mindig a következő végrehajtandó utasítás címét tartalmazza. A program betöltésénél az utasításszámláló regiszter megkapta az első programutasítás címét, példánkban ez a 366. cím, azaz a 366. címtől kezdődően található a végrehajtandó utasítások az operatív tárnak.

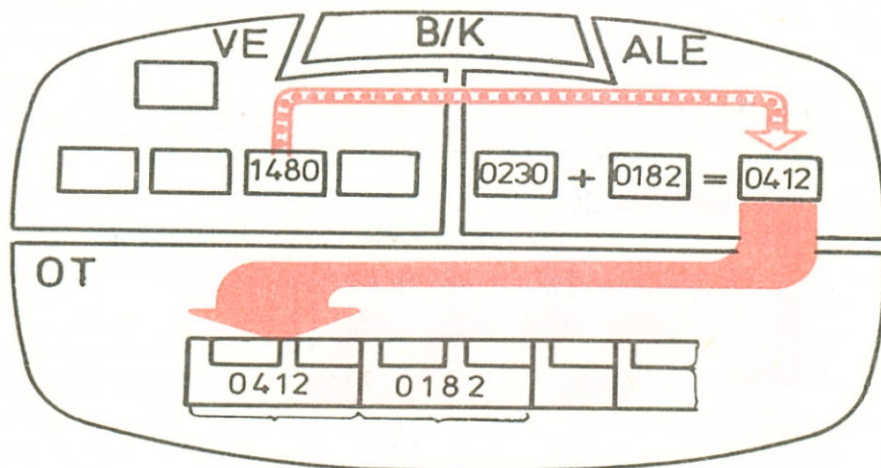
A vezérlőegység most átveszi ezt az első, pl. hat byte hosszúságú utasítást az operatív tárból.



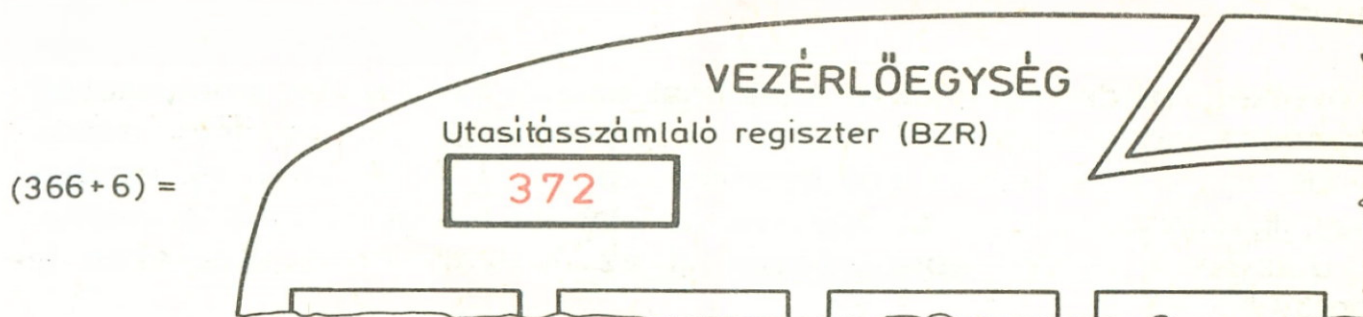
A műveleti regiszter tartalmát a vezérlőegység értelmezi, ami egy terjedelmes *dekódolókapcsolás* segítségével történik. Felismeri, hogy összeadási utasításról van szó. Ezért gondoskodik az összeadandó adatok lehívásáról az operatív tárból és átvitelükről az aritmetikai és logikai egységbe. Ezek az adatok: egyrészt az 1480. és 1481. című tárolási helyek tartalma (a hosszregiszter első részében a 2 hosszúságú adat értelmében), másrészt az 1482. és 1483. tárolási helyek tartalma (a hosszúságregiszter második részében a 2 hosszúságú adat értelmében).



Az aritmetikai és logikai egység *összeadó-kapcsolások* segítségével végrehajtja az összeadást. A vezérlőegység elrendeli, hogy az eredményt az 1. címregiszterben megadott címen kell tárolni.



Most tehát az első utasítás végre van hajtva. Mivel ez az operatív tárban 6 byte-ot foglal el, tehát 6 byte hosszúságú, ezért az *utasításszámláló regiszter* tartalma 6-tal megnövekszik, és ezért a 372. címet tartalmazza.



Az operatív tárban a 372. címen a soron levő következő programutasítás van tárolva. Ez egy **ugró utasítás**, azaz a program folyamán e hely elérésekor a program egy másik helyén kell a műveletet folytatni.

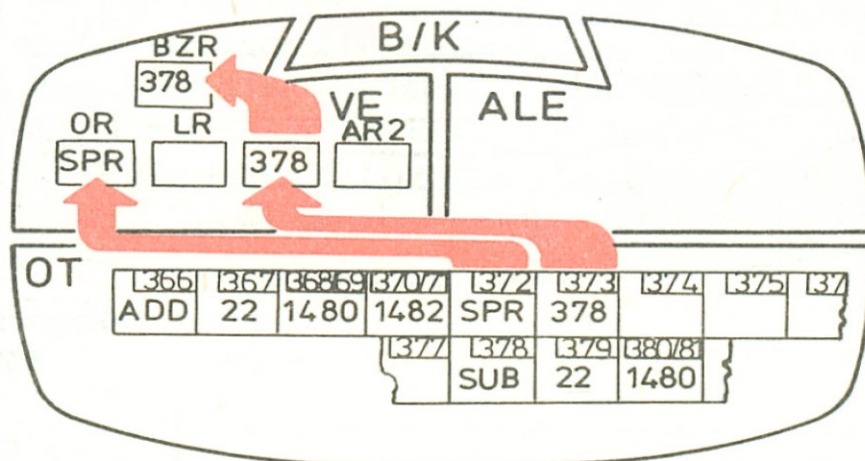
Az ugró utasítás a következő:

Ugrási utasítás:



azaz: ugorj a 378. címre, és hajtsd végre azt az utasítást, amely ott van tárolva!

Ez az utasítás a vezérlőegységbe továbbítódik, amely azt a *dekódoló-kapcsolás* segítségével mint ugró utasítást azonosítja. Ezután a következő végrehajtandó utasítás címe az *utasításszámláló regiszterbe* kerül.

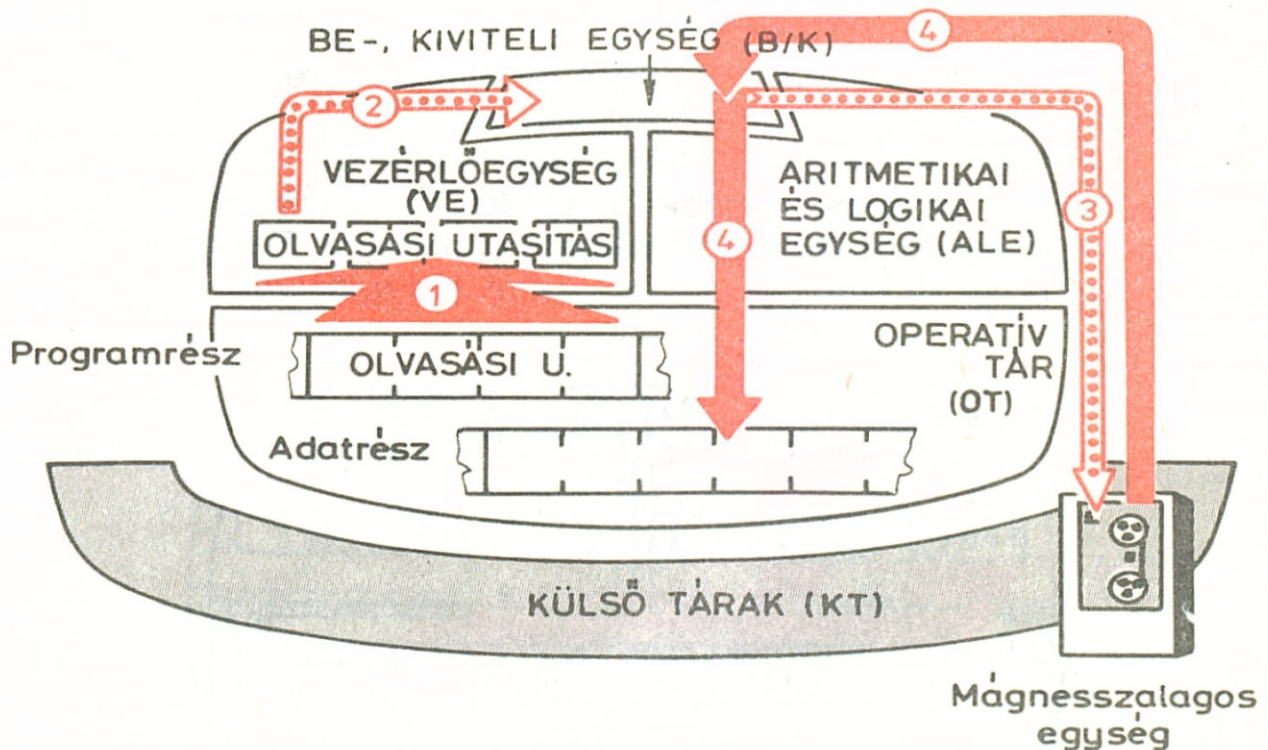


Következő lépésként azt az utasítást, amely az operatív tárban a 378. címen található, a vezérlőegységbe viszi a számítógép, és végrehajtja azt.

A program további utasításai is ehhez hasonlóan kerülnek feldolgozásra. Ha a számítógép perifériáit érintő utasításokról van szó, mint pl. a *be-, ill. kiviteli utasítások*, akkor a vezérlőegység aktiválja a be-, ill. kiviteli vezérlést, amely a perifériákhoz továbbítja a szükséges adatokat, vagy azokat onnan lehívja.

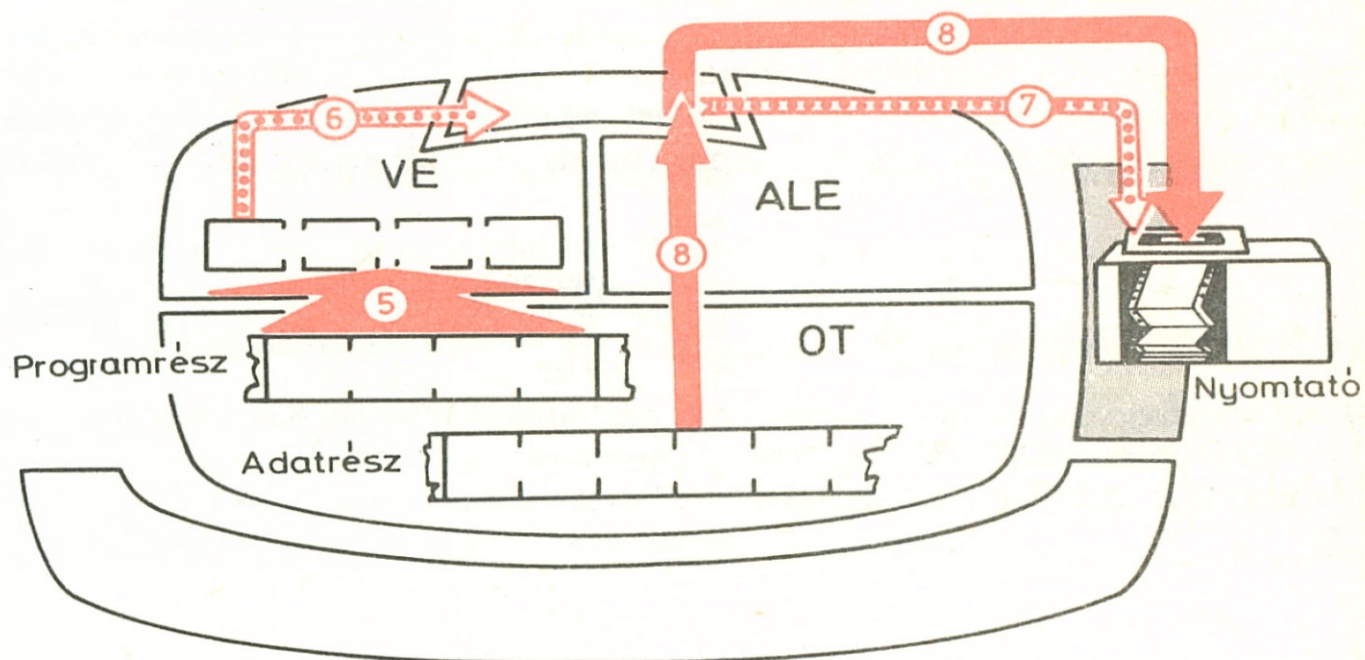
A következő példán azoknak az adatoknak az útját mutatjuk be, amelyeket egy mágnesszalagos egység az operatív tárba beolvas, és ezt követően minden további feldolgozás nélkül egy nyomtatónak ad ki. A két utasítás, amelyek a be-, ill. a kivitelt rendelik el, egy olvasási utasítás és egy nyomtatási utasítás. Az utasítások egzakt ábrázolásáról itt az áttekinthetőség érdekében lemondunk.

Az **olvasási utasítást** a számítógép az operatív tár programrészéből a **vezérlőegységbe** viszi. Ez többek között azt is tartalmazza, hogy az adatokat melyik egységből kell kiolvasni (1).



A vezérlőegység aktiválja a *be-, ill. kiviteli vezérlést* (2). Ez kapcsolatot létesít a kívánt perifériával, amely pl. egy mágnesszalagos egység lehet (3). Az igényelt adatok ezután a mágnesszalagról leolvasásra kerülnek, és a be-, ill. kiviteli vezérlésen át az *operatív tár adatrészébe* továbbítódnak (4).

Az olvasási utasítás végrehajtása ezzel befejeződött, és a **nyomtatási utasítást** a gép a vezérlőegységbe juttatja (5).



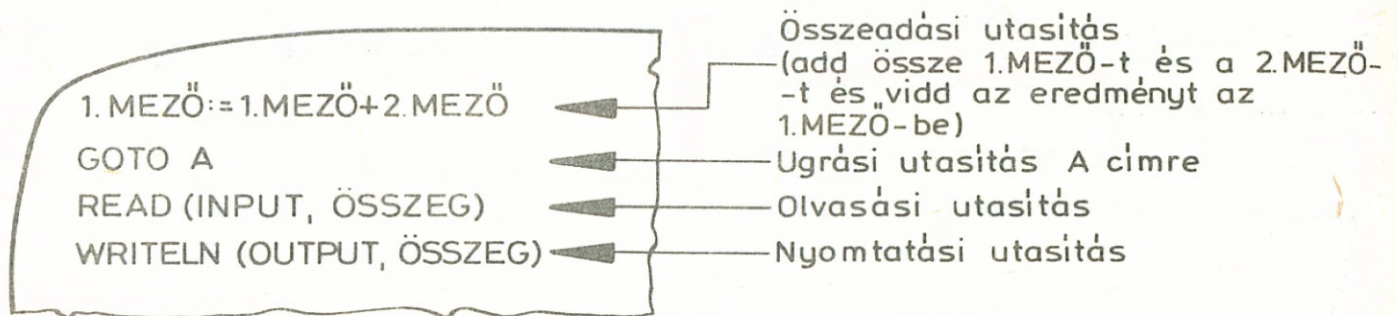
A vezérlőegység felismeri a nyomtatási utasítást, és aktiválja a *be-, ill. kiviteli vezérlést* (6), amely kapcsolatot létesít a nyomtatóval (7). Mivel a nyomtató aránylag lassú periféria, azért azt többnyire a multiplexcsatornán keresztül működtetjük. Az adatok tehát az operatív tárból a *be-, ill. kiviteli vezérlésen* keresztül a multiplexcsatornán át kerülnek a nyomtatóba (8).

Ezek a példák mindenekelőtt a *vezérlőegységnek* azt a funkcióját mutatják be, amelyet az mint a számítógép koordinátora tölt be. Természetesen magát a vezérlőegységet is vezérelni és ellenőrizni kell. Ezt olyan fölérendelt programok végzik, amelyekre a felhasználónak nincs befolyása: ezek a **rendszerprogramok**.* De ezek már szoftverkérdések, amelyekkel sorozatunk „Kulcs a számítógéphez — Szoftver” c. kötetében foglalkozunk.



* Itt a szerzők egy rendkívül leegyszerűsített képet adnak a rendszerprogramok feladatairól. (Lektor.)

Egy program feldolgozásának pontos belső folyamatával azonban a felhasználónak általában nem kell különösképpen törődnie. Azt, hogy az előzőekben leírt utasítások a programozó részére milyen egyszerű formában jelentkeznek, a következő példa mutatja, amelyben az utasítások egy adott programnyelven vannak megfogalmazva. A programozónak a *címek megnevezésére* nem kell számokat használnia, sokkal inkább magaválasztott nevekkel (szimbolikus címekkel) nevezi meg a tárolási helyeket, és a megfelelő abszolút címek megtalálását a számítógépre bízta. Példánkban az „1. mező”,



„2. mező” és az „összeg” szimbolikus címeket választottuk. A kész program alkalmazójának még a programozónál is könnyebb dolga van, akit — ha a programozó ezt így tervezte — a számítógép udvariasan felszólít egy összeg bevitelére, vagy megkérdezi, hogy az eredményt ki kell-e nyomtatni.

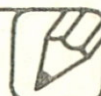
ÍRJA BE A KÖVETKEZŐ ÖSSZEGET:
AZ EREDMÉNY KINYOMTATANDÓ (IGEN/NEM)



Könyvünkben eddig eljutva Ön tulajdonképpen már nagyon sokat tud a számítógép-hardverről. Tudja, hogy:

- mely egységek, részek és vezetékek tartoznak hozzá;
- elvileg hogyan működnek ezek az egységek, részek és vezetékek;
- a számítógép az ember által megfogalmazott utasításokkal hogyan vezérli a hardvert, és hogyan tud feladatokat megoldani.

Befejezésül — tehát könyvünk végén — felsoroljuk azokat a fontos történelmi állomásokat, amelyek a mai számítástechnikához vezettek. Egy pillantást vetünk a jövőbe, és talán megsejthetjük, hogy a következő években és évtizedekben ez a fejlődés hová vezethet.



1. Hány gépi utasítást tud értelmezni egy számítógép?

Annyit, amennyit az utasításkészlete tartalmaz.

Annyit, amennyit az alkalmazott programnyelv tartalmaz.

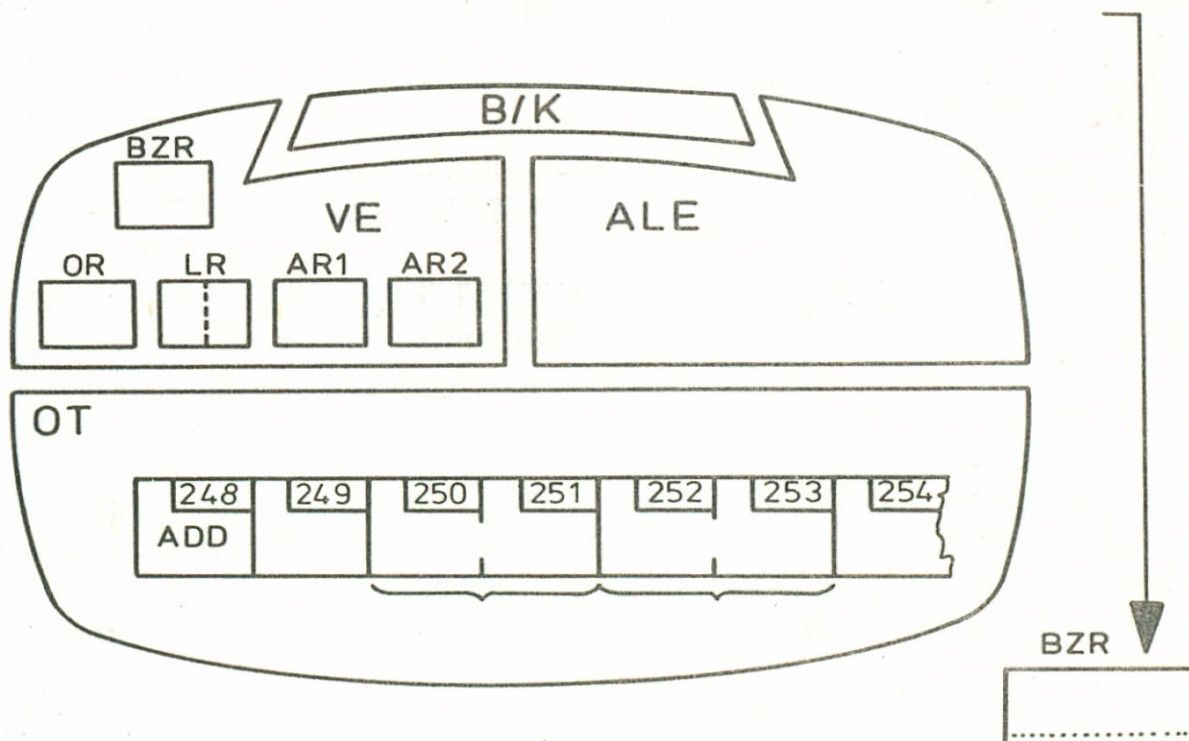
2. A 248. utasítást kell a központi egységben feldolgozni.

Ez egy 6 byte hosszúságú összeadási utasítás.

Az összeadandó adatok az 1010...1013. tárolási helyeken állnak, és egyenként 2 byte hosszúságúak.

a) Rajzolja be a központi egység alábbi vázlatán az operatív tár programrészébe és a vezérlőegységbe azt az állapotot, ami az utasítás lehívása után kialakult!

b) Melyik cím áll az utasítás végrehajtása után az utasításslámlálóban?



3. Milyen utasítások hatására aktivizálja a vezérlőegység a be-, ill. kiviteli vezérlést?

Aritmetikai utasítások.

Be-, ill. kiviteli utasítások.

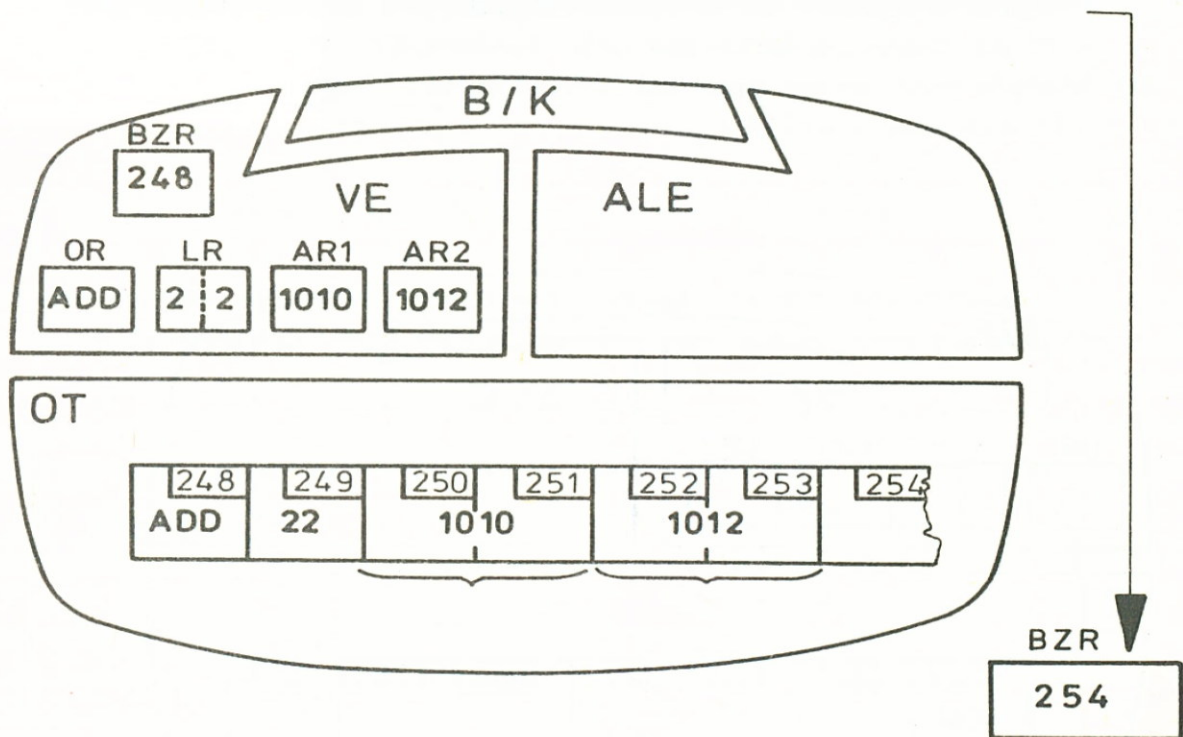


Válaszok

1. Egy számítógép annyi gépi utasítást tud értelmezni,
 amennyit utasításkészlete tartalmaz.

2. a)

b) Következésként a **254.** utasítás áll az utasításslámláló regiszterben.



3. A vezérlőegység a be-, ill. kiviteli vezérlést:

A be-, ill. kiviteli utasítások hatására aktivizálja.

Visszatekintés a múltba és előretekintés a jövőbe

A technikának talán egy területe sem ért meg az utóbbi években olyan gyors fejlődést, mint a számítástechnika. Emellett a gépi segítséggel végzett adatfeldolgozás egyáltalán nem századunk találmánya.

Még az i. e. 82. évből származik Antikythera planetárium, egy bronzból készült fogaskerék-szerkezet, amelyet a régi görögök a csillagászati és hajózási számításokhoz használtak. Ez a planetárium egy analóg számítógép első ismert régészeti lelete, amelynek technológiája azonban egy hosszú, egészen Babilonig visszamutató fejlődésről tanúskodik.

Kb. ugyanebben az időben tűnt fel a digitális számítógép legegyszerűbb formája: az *abakus*. Ezzel a számolótáblával elvégezhető volt a négy alapművelet. A 13. században Raimundus Lullus, egy spanyol misztikus felismerte egy olyan gép elvét, amely logikai következtetések levonására volt alkalmas. A 17. században Wilhelm Schickard és Blaise Pascal egymástól függetlenül felfedezték az összeadó- és kivonógépet. Gottfried Wilhelm Leibnitz filozófus és a differenciál- és integrálszámítás megalkotója a négy alapművelet elvégzésére alkalmas számológépet szerkesztett, és kidolgozta a különféle számrendszerek alapjait, közöttük a bináris számrendszert.

1822-ben Charles Babbage leírta egy programvezérlésű digitális számítógépet. A 19. század vége felé Hermann Hollerith lyukkártya-gépekkel dolgoz fel tömeges adatokat.

1941-ben Konrad Zuse felépíti az első programvezérlésű relés digitális számítógépet. Néhány évvel később Neumann nyilvánosságra hozza a tárolt programozás elvét. A Pennsylvanai Egyetemen megszületik az első elektroncsöves elektronikus számítógép (ENIAC), és már a hatvanas évek közepe táján megkezdődik az integrált áramkörös adatfeldolgozó számítógépek építése.

Az új gyártási eljárások egyre kisebb építőelemek készítését teszik lehetővé, úgyhogy ma egy teljesen működőképes számítógép már néhány négyzetcentiméternyi területen elfér.

A kommunikációnak is hosszú fejlődéstörténete van. Ismeretes, hogy már az antik világban is alkalmaztak nagy távolságokon optikai jelátvitelt.

A 18. században Hooke javaslatot tesz hangjeleknek egy kifeszített huzalon való továbbítására, és Chappe optikai telegráfvonalakat épít ki olyan jelzőkészülékekkel, amelyek karjaik helyzetével tudnak jeleket továbbítani.

A korszerű kommunikációtechnika alapjait 1820-ban Oersted és Ampère az elektromagnetizmus leírásával rakja le. Ugyanebben az évszázadban Henry és Morse a távíró, Reis és Bell pedig a telefont fedezik fel. Maxwell és Hertz kidolgozza az elektromágneses hullámok terjedésének elméletét, és Marconi kiépíti az első radiotelegrafikus összeköttetést.

A rádió-, távíró- és tv-technika világviszonylatban csak századunkban terjedt el. A moduláció alapjait Nyquist, az információelméletet Shannon, a kibernetikát Wiener, és a rendszerelméletet Küpfmüller fejlesztette ki.

Előre látható, hogy az utóbbi évek viharos fejlődése mind a kommunikációtechnika, mind pedig az adatfeldolgozás területén továbbtart.

Könyvünknek ebben a kötetében olyan technikák sorát mutattuk be, amelyek jelenleg még a bevezetési fázisban vannak. Ide tartoznak a csomagkapcsolt, teletex-, teldata- és képújság-szolgáltatások, vagy az integrált irodarendszerek, amelyek az egyre fokozódó munkamegosztás és a munkafolyamatok egyre szorosabb összefonódásának keretében a mind inkább összetettebbé váló feladatok megoldásának gerincét alkotják. Megemlítettük továbbá, hogy a nyilvános hálózatok egy egységes szolgáltatásintegrált digitális hálózattá fognak egyesülni, amelyen a személyi és adatkommunikáció minden fajtája lebonyolítható lesz.

A fejlődésben lényeges lökést fog adni az optikai kábelek bevezetése, amelyek majd megszüntetik az információátvitel és különösen a képek átvitele esetén mutatkozó jelenlegi szűk keresztmetszetet. Arra azonban csak tíz-húsz év múlva számíthatunk, hogy az eddigi vörösréz kábeleket széles körben helyettesíti.

Nincs még teljesen megoldva az adatoknak az ember általi bevitele a gépbe. Az ember ma még a billentyűzetre van utalva, a korlátozott szókinccsel történő automatikus beszédbevitel terén még csak az első lépéseknél tartunk. A teljes megoldás — beszédbevitel tetszőleges személyek által, tetszőleges szókinccsel — a közeljövőben valószínűleg még nem lesz megvalósítható.

Az adatfeldolgozásban a tárolás szűkebb területén lényeges lépések várhatók a holografikus táruk alkalmazásával, és a félvezető-technikában a lapkák továbbfejlesztésével. Itt még nem érték el a műszaki lehetőségek határai miatti korlátokat. Ezzel szemben a jövő „papír nélküli irodájának” álma — információfeldolgozás és -átvitel képernyővel ellátott számítógéppel — műszakilag lényegében már megoldott.

Az automatikus beszédbevitelhez hasonló probléma a más nyelvre történő automatikus fordítás kérdése. Ezen a téren is vannak már jelentős eredmények, azonban a tökéletes megoldás — még egyszerű szövegek esetében is — még néhány évet igénybe fog venni.

Ez utóbbi két, érintett terület szorosan összefügg a mesterséges intelligencia kérdésével. Vajon intelligens-e a számítógép?

Korlátozott értelemben már ma is intelligensnek mondhatjuk, amennyiben a tanulékonytságot, a tapasztalatok figyelembevételével, sok információból logikai következtetések levonását korlátozott intelligenciaként definiáljuk: gondoljunk csak pl. a sakkozó számítógépekre!

Ez a kérdés azonban már nehezebben válaszolható meg, ha megkíséreljük a számítógépnek az emberi aggyal való összehasonlítását.

Ha a számtalan szempont közül csak a tárolóképességet ragadjuk ki, akkor mai ismereteink kszerint az agyvelőben a tároló sejtek (synapsis) számát kb. tízmilliárdnak vehetjük. Egy nagyszámítógép operatív tára ma közel egymilliárd bitet tud tárolni, tehát kezdi megközelíteni az emberi agy tárolóképességét, és néhány év múlva meg is fogja azt haladni.

Az emberi agy azonban az osszociációval lényegesen meg tudja növelni tárolóképességét. Ez nem más, mint az elképzések összekapcsolásának lehetősége, miközben egy elképzelés számos más elképzélést idéz fel. Más szavakkal annak lehetősége, hogy a tároló sejtek tartalmát nem a címek megadásával, hanem csak a tartalmuk egyes részeinek megadásával is le tudjuk hívni. Ilyen képességei a számítógépnek ma még nincsenek*.

Ha tehát az ember intelligenciáját vesszük mértékül, arra a kérdésre, hogy a számítógép mikor fogja az emberi agy képességeit elérni, az erre a kérdésre adható spekulatív válasz így hangzik: vagy a következő évezredben, vagy soha!

Ebben a kötetben az adatfeldolgozást csak egy szemszögből: a hardver szemszögből vizsgáljuk.

Ahhoz, hogy teljes képet kapjunk a számítástechnikáról, a szoftver és a szervezés témákkal is foglalkoznunk kell. E témákkal való könnyű megbarátkozásra és az adatfeldolgozás alapismereteibe való általános bevezetésre ajánljuk e sorozat további köteteit.

Kulcs a számítógép-
hez - Gyakorlat



Kulcs a számítógép-
hez - Szoftver



Kulcs a számítógép-
hez - Szervezés (Orgver)



* Legújabbán már léteznek ún. asszociatív számítógépek is (pl. PEPE, STARAN, ECAM, ALAP). (Lektor.)

Tárgymutató

A, Á

adatátvitel 90
 adatátviteli berendezések 165
 — hálózatok 164
 — végpontok 21
 adatbázis 40
 adatbusz 158
 adatfeldolgozás 13
 adathálózat 96
 adathordozó-változások 46
 adatkommunikáció 17, 37
 adatok 37
 adatrögzítő berendezés 104
 alacsony hőmérsékletű táruk 142
 alközpont 22
 állandó összeköttetés 56
 — vezetékek 164
 ALU 151
 analóg 51
 — számítógép 83
 aritmetikai és logikai egység 151
 asztali számológép 70
 automatikus hívásismétlés 23

B

be-, ill. kiviteli processzor 155
 —, — vezérlés 154
 bérelt vonalak 55
 beszédbevitel 112
 beszédbeviteli egységek 112
 beszédkivitel 125
 beszédkommunikáció 22
 betűhenger 120
 beviteli egység 102, 104
 billentyűs hívás 23
 bistabil áramkör 181
 biztonsági hálózat 96
 blokk 133
 buboréktár 141
 busz 158

C

cache-tár 157
 CCD 139
 chip 186
 címbusz 158
 címregiszter 201
 COM 123
 CPU 149

Cs

csak olvasható tár 138
 csatlakozófelület 62
 csatorna 162
 csomagkapcsolás 58
 csomagok 58
 csomóponti gépek 166

D

Datex 55
 datex-L 59
 datex-P 59
 decentralizált adatfeldolgozás 92
 dekódoló kapcsolások 178
 digitális 51
 — adathálózat 54
 — átvitel 54
 — hálózat 62
 — számítógép 83
 display 31
 duplex üzem 59

E, É

EAF 13
 egylapkás számítógép 191
 egységesített szabványos csatlakozás 162
 elektronikai építőelemek 171
 elektronikus adatfeldolgozás 13
 EMS 43
 EPROM 138
 ÉS kapu 171

F

fejlettebb szövegrendszerek 30
 feldolgozási sebesség 82
 félduplex üzem 59
 félvezetőrétteg 187
 félvezetős tár 137
 félvezető-technika 187
 fix programú asztali számológépek 71
 — — zsebszámológépek 71
 — — adattára 71
 — — billentyűzete 71
 — — központi egysége 71
 flip-flop 181
 floppy disk 135
 folyamatirányító számítógépek 84
 funkcionális hálózat 96

G, Gy

gép—gép kommunikáció 96
 gépi nyelv 199
 gyorstárok 157

H

hajlékony mágneslemez 30
 — mágneslemezes tár 135
 hálózatvezérlő számítógép 61
 hangos telefon 23
 hardver 14
 hardver 14
 hardver-interfész 62
 hardware 14
 háttértárok 129
 hibrid számítógép 83
 holografikus tárok 141
 hosszregiszter 201
 hozzáférési idő 152

I

információátvitel 52
 információbázis 40
 információk ábrázolása 51
 integrált áramkörök 185
 — digitális hálózat 54
 — irodai rendszer 43
 intelligens terminálok 77, 109
 interfész 62
 írógép 28
 írható—olvasható tár 138

J

jelölésérzékelők 112
 Josephson-áramkörök

K

kapcsolt hálózat 164
 — vonalak 55
 karakternyomtató 120
 kártyalyuksztó 110
 képernyős terminálok 106, 107
 képkommunikáció 24
 képtelefon 26
 képűjság 42
 kétbemenetű ÉS kapu 172
 kisszámítógépek 79
 kivitel 117
 kiviteli egység 102, 118
 kommunikáció 17
 kommunikációs számítógép 61
 kommunikációtechnika 17
 koncentrátorok 165
 közeli forgalom 161
 — kapcsolat 161

központi adatfeldolgozás 90
 — egységek 102, 149
 — tár 92
 közvetlen beszélgetés 22
 — hozzáférésű tárok 130

L, Ly

láncos nyomtató 120
 lapka 186
 lapnyomtatók 121
 lapok 153
 lapváltás 153
 látszólagos tárolás 153
 lemezköteg 134
 lézernyomtatók 121
 levelezés 28
 lyukkártyalyuksztók 124
 lyukkártyaolvasó 110
 lyukszalaglyukasztók 124
 lyukszalagolvasók 110

M

mágnescsíkos tár 136
 mágneses írásolvasók 112
 mágneskártyás tár 136
 mágneslemezes tár 134
 mágnesszalag-kazettás tár 136
 mágnesszalagos tár 132
 magnetooptikai tárok 141
 mátrixnyomtató 120
 megjelenítő 31
 mikrofilmes adatkivitel 123
 mikroprocesszor 75
 mikroszámítógép 70
 miniatürizálás 16
 miniszámítógépek 79
 modemek 53, 165
 multiplexcsatorna 162
 művelet kódja 200
 műveleti regiszter 201
 — rész 200

N, Ny

nagyszámítógépek 80
 NEM kapu 174
 nemmechanikus nyomtatók 121
 nyilvános távbeszélő-hálózat 22
 nyomtatási utasítás 207
 nyomtató 119
 nyomtatós terminálok 106, 108

O, Ő

olvasási utasítás 206
 operatív tár 129, 152
 — — adatrésze 203

— — programrésze 203
 optikai karakterolvasó 111
 — táruk 141
 órajel-generátor 156
 összeadó-kapcsolások 180
 összehasonlító-kapcsolások 179
 összetett kapcsolások 173
 osztott adatfeldolgozás 93

P

paging 153
 párbeszédés berendezés 102
 — egységek 106
 perifériák 103
 processzor 149
 professzionális személyi számítógépek 77
 programnyelv 37, 199
 programok 37, 199
 programozható asztali számítógépek 73
 — csak olvasható tár 138
 — zsebszámológépek 73
 PROM 138
 protokollok 59
 puffertár 157

R

RAM 138
 rajzgépek 123
 regiszterek 201
 rendszerprogramok 207
 REEPROM 138
 ROM 138

S

scratch-pad memory 157
 sín 158
 software 14
 sornyomtató 120
 soros elérésű táruk 130
 speciális rendszerek 84

Sz

szabadon programozható
 mikroszámítógépek 75
 szalaglyukasztó 110
 számítógép-hálózat 39, 95
 számítógéphardver 14
 számítógép-terminálok 138
 szelektorcsatorna 162
 személyes adatbank 41
 személyi kommunikáció 17
 — kisszámítógépek 77
 szimbolikus címek 208
 szimplex üzem 59
 szoftver 14, 81

— interfész 62
 szóhossz 82
 szövegautomaták 29, 76
 szöveges kommunikáció 27
 szövegfeldolgozás 28
 szövegkezelés 28
 szövegrendszerek 28, 29, 76

T

táruk 102, 113
 tárgyprogram 199
 tárkapacitás 82
 tároló írógépek 29, 30
 tárolókapcsolás 181
 tárolt átvitel 58
 távadatátvitel 164
 távbeszélés 22
 távfeldolgozás 38, 161
 távgépíró 27
 távíró 27
 távíróhálózat 54
 távmásolás 24
 távmásoló 24
 távoli kapcsolat 161
 teledata 39
 TELEPAX 24
 telelevél-szolgálat 25
 TELETEX 32
 TELEX 24, 27
 telexhálózat 27
 teljes csatlakozású hálózat 60
 terheléskiegyenlítő hálózat 96
 terminálok 106, 119
 töltéscsatolt eszközök 139
 törölhető PROM 138
 tranzisztorok 185

U

ugró utasítás 205
 univerzális rendszerek 84
 — számítógép 81
 utasításkészlet 199
 utasításszámláló regiszter 202

V

VAGY kapu 173
 végpontok 18, 21
 vektorszámítógépek 84
 vezérlőegység 150
 videotext 42
 virtuális tárolás 153
 vonalkapcsolt összeköttetés 57
 vonalkód-leolvasók 111

Zs

zsebszámológép 70, 71

Tartalom

Előszó	5
A kötet célja	7
Tanulási módszer	8
Adatfeldolgozás és kommunikáció	11
Számítógép és kommunikációtechnika	13
Számítógéphardver	14
Kommunikáció és kommunikációtechnika	17
Személyi kommunikáció	21
Adatátviteli végpontok	21
Beszédkommunikáció	22
— Távbeszélés	22
Képkommunikáció	24
— Távmásolás (TELEFAX)	24
— Képtelefon	26
Szöveges kommunikáció	27
— TELEX	27
— Szövegrendszerek	28
— TELETEX	32
Adatkommunikáció	37
Az adatkommunikáció sajátosságai	37
Az adatkommunikáció formái	38
— Távfeldolgozás	38
— Számítógép-hálózat	39
— Teledata	39
— Képújság	42
— Integrált irodai rendszerek	43
Átviteli vonalak	51
Analog és digitális átvitel	51
Kapcsolt és bérelt vonalak	55
Vonalkapcsolás és csomagkapcsolás	57
Üzem módok	59
Hálózattopológiák	60
Áttekintés az összeköttetési lehetőségekről	62
Számítógépek és alkalmazásuk	67
A számítógépek osztályozása és teljesítményviszonyai	69
A számítógépek osztályozása méretük és teljesítményük szerint	69
— Mikroszámítógépek	70

— Kiszámítógépek	79
— Nagyszámítógépek	80
Digitális és analóg számítógépek	83
Univerzális és speciális rendszerek	84
A számítógépek üzemmódjai	89
Központi adatfeldolgozás	90
Decentralizált adatfeldolgozás	92
Osztott adatfeldolgozás	93
Számítógép-hálózatok	95
A számítógépek funkcionális egységei	99
A bevitel	104
Terminálok	106
Lyukkártyaleolvasók	110
Lyukszalagleolvasók	110
Optikai karakterolvasók	111
Vonalkódleolvasók	111
Jelölésérzékelők	112
Mágneses írásolvasók	112
Beszédbeviteli egységek	112
Bevitel táraikon keresztül	113
A kivitel	117
Terminálok	119
Nyomtatók	119
Rajzgépek	123
Mikrofilmes adatkiviteli egységek	123
Lyukkártyalyuksztók	124
Lyukszalaglyuksztók	124
Beszédkivitel	125
Kivitel táraikra	125
A tárolás	129
Mágnesszalagos tár	132
Mágneslemezes tár	134
Hajlékony mágneslemezes tár	135
Mágnesszalag-kazettás tár	136
Félvezetős tár	137
Buboréktár	140
Optikai táraik	141
Alacsony hőmérsékletű táraik	142
A feldolgozás	149
A vezérlőegység	150
Az aritmetikai és logikai egység	151
Az operatív tár	152

A be-, ill. kiviteli vezérlés	154
A központi egység egyéb egységei	156
Adatátviteli berendezések	161
Közeli kapcsolatok	162
Távoli összeköttetések	164
— Adatátviteli hálózatok	164
Hogyan működik a számítógép?	168
A számítógép-áramkörök logikai alapjai	171
Alapkapcsolások	171
Összetett kapcsolások	177
A számítógép-áramkörök megvalósítása	185
Az áramkörök felépítése	185
Az áramkörök gyártása	188
A számítógép műszaki felépítése	191
A hardver és a szoftver kapcsolata	199
Utasítások a központi egységben	199
Visszatekintés a múltba és előrettekintés a jövőbe	211
Tárgymutató	215

JEGYZET

JEGYZET

J.TWIEHAUS

KULCS A SZÁMÍTÓGÉPHEZ

SZOFTVER

2. kiadás

SZERKESZTETTE:
MARTIN F. WOLTERS

MŰSZAKI KÖNYVKIADÓ, BUDAPEST, 1988

Az eredeti mű:

J. Twiehaus: Der Schlüssel zur Computer — Software

Herausgegeben: Martin F. Wolters

Copyright by Econ Verlag, Düsseldorf, 1981

Lektorálta:

Dr. Sima Dezső okl. villamosmérnök

Hungarian translation

© Dr. Büky Péterné, Pótzty Péter, 1984

ETO: 681.3

ISBN: 963 10 5694 5

963 10 5850 6

963 10 7594 x

Előszó

Kiadónk gondozásában tíz éve jelent meg a „Kulcs a számítógéphez” első változata.

A könyv világszerte sikert aratott:

számos nyelvre lefordították*,

összesen több mint 250 000 példányban jutott el az olvasókhöz.

Mostani négykötetes vállalkozásunkkal

ezt a hagyományt szeretnénk folytatni.

A számítástechnika rohamos fejlődése:

az újszerű műszaki megoldások a hardver vonatkozásában;

teljesen új utak a programozásban,

amelyek a szoftvertechnológia kialakulásához vezettek;

és a szervezés terén bekövetkezett változások,

amelyek a számítástechnika alkalmazását kísérik,

szükségessé tették,

hogy a számítástechnika alapismereteinek megismertetéséhez

teljesen új oktatási anyagot dolgozzunk ki.

Ezért az ismereteket a

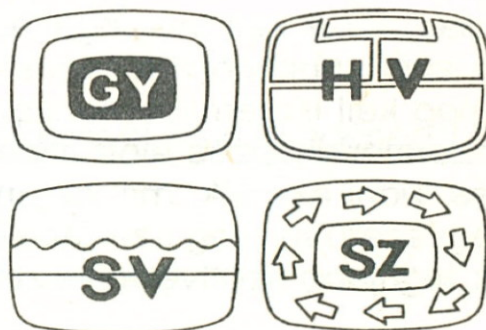
a következő négy kötetre osztottuk fel:

Kulcs a számítógéphez — Gyakorlat.

Kulcs a számítógéphez — Hardver.

Kulcs a számítógéphez — Szoftver.

Kulcs a számítógéphez — Szervezés (Orgver).



E felosztás és a kötetek anyagának összeállítása

számítástechnikai és oktatási szakemberek

hosszú közös munkájának eredménye.

A szerzők az ismeretek tárgyalása során most is arra törekedtek,

hogy az érdeklődő laikus a szöveget azonnal megérthesse,

a legfontosabb tudnivalókat könnyen elsajátíthassa.

* Többek között 1972-ben és 1974-ben két kiadásban is magyar nyelven. (Szerk.)

A „Kulcs a számítógéphez — Szoftver” c. kötet a nem szoftverspecialisták számára ad áttekintést a legkorszerűbb szoftvertechnológiáról.

E kötetben logikusan felépített, könnyen érthető tananyag ismerteti meg az olvasóval, hogy

- mi a szoftver, mire használható, és mi jellemzi a magas színvonalon megírt szoftvert;
- milyen lépésekből áll a szoftverfejlesztés folyamata a tervezéstől az alkalmazásig;
- milyen intézkedések és eszközök segítik a szoftverfejlesztő munkáját;
- milyen feladatai vannak a programozónak a szoftverfejlesztés során;
- mi a feladata a számítógép operációs rendszerének (amely, mint látni fogjuk, maga is szoftver) a szoftver fejlesztésében és felhasználásában.

E — szoftver-alapismereteket összefoglaló — kötet azoknak készült, akiknek a számítástechnikáról eddig csak általános ismeretei voltak, most pedig:

- maguknak kell megfogalmazniuk a számítástechnika segítségével megoldandó feladataikat;
- meg kell ismerniük munkatársaik és beosztottaik munkáját;
- szoftverkiképzés előtt állnak, és tudni akarják, mit várjanak attól;
- egyszerűen meg szeretnék érteni a legújabb szoftverfejlesztésekkel foglalkozó publikációkat.

A könyv a didaktikus fölépítése, a sok ábra és a gyakorlati példák sokasága miatt szoftvertanfolyamokhoz oktatási segédletként is igen jól használható.

Aki a számítástechnikai alapismeretek mellett — amelyek e sorozat „Kulcs a számítógéphez — Gyakorlat” c. összefoglaló kötetéből egyszerűen megszerezhetők — még komolyabban is érdeklődik a számítástechnika iránt, az e könyvből a szoftverrel kapcsolatos kérdéseket mélyebben megismerheti.

München, 1981. szeptember

Martin F. Walters

A kötet célja

A kötetet gondosan áttanulmányozó olvasó:

- tisztában lesz a szoftver fogalmával, a jó minőségű szoftver jellemzőivel, a szoftver osztályozási szempontjaival, a tervszerű és rendszeres szoftverfejlesztés fontosságával;
- megismeri a szoftverfejlesztési folyamat lépéseit, a különböző fázisokban jelentkező feladatokat, a megoldásokhoz alkalmazott szervezési módszereket;
- megismeri a fejlesztő munkáját — a szoftver tervezését, specifikálását, ellenőrzését, tesztelését és dokumentálását támogató eljárásokat és segédeszközöket;
- áttekintést kap a programozó munkájáról, a programfejlesztési, -tesztelési, -dokumentálási tevékenységről; sőt egyszerűbb feladatokat önállóan is képes lesz a számítógép nyelvén megfogalmazni;
- megismerkedik az operációs rendszer felépítésével és működésével, a programfuttatás vezérlésével, a berendezések üzemmódjának fogalmával.

Tanulási módszer

E sorozat könyvei a szerzőcsoport által kifejlesztett, jól bevált programozott oktatási módszeren alapulnak.

E módszerrel készült művek különösen alkalmasak az önálló tanulásra, mert:

- a magyarázathoz hétköznapi nyelvet használnak, így a szöveg jól érthető, és az ismeretek logikusan, gondolati ugrások nélkül épülnek egymásra;
- a fontosabb meghatározásokra, ismeretekre nemcsak a szövegbeli nyomdatechnikai kiemelések, hanem az ábrák is felhívják a figyelmet a jobb rögzítés érdekében;
- minden fejezet végén feladatok segítik:
 - a lényeges ismeretek fontosságának felismerését,
 - az olyan összefüggések megértését, amelyeket a szövegben nem mindig lehet részletesen kifejteni,
 - az újonnan tanultak összekapcsolását a már meglévő ismeretekkel.

Mielőtt e könyv áttanulmányozását megkezdené, a következőket kell tudnia:

A feladatok nem az Ön vizsgáztatására készültek.

Ezért nem is fontos, hogy a feladatokat „kapásból” meg tudja oldani.

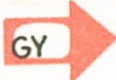


A kérdések a lapok egyik oldalán, míg a válaszok a túloldalon találhatóak.

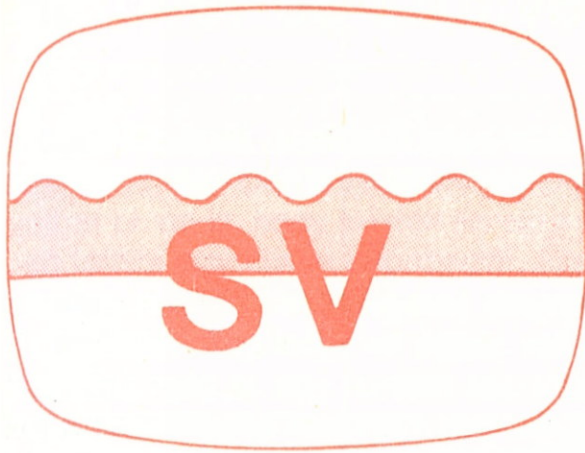
A kérdéseken való elgondolkozás és a válaszok helyességének ellenőrzése didaktikailag igen fontos.

Ha valamelyik választ nem helyesen adta meg, vagy nem érti, hogy miért a megadott válasz a helyes, akkor olvassa el újra a kérdésekhez tartozó, megfelelő oldalakat: a kérdéseket tartalmazó lap felső részén mindig megadjuk, hogy azok mely oldalakon található szövegrészekre vonatkoznak.

A szövegben többször utalásokat fog találni sorozatunk többi köteteire, amelyekben a szóban forgó témát részletesebben tárgyaltuk.

Azért, hogy ez szembeűnő legyen, a kötetekre való utalást a következőképpen jelöltük:

- | | | |
|-----------------------|---|---------------------|
| Kulcs a számítógéphez |  | Gyakorlat. |
| Kulcs a számítógéphez |  | Hardver. |
| Kulcs a számítógéphez |  | Szervezés (Orgver). |



Ebben a fejezetben megtanulja, hogy:

- mit is jelent a szoftver,
- a szoftvert tervszerűen és módszeresen kell fejleszteni és ez hogyan történhet,
- milyen fajtái vannak a szoftvernek, és mit kell érteni egy számítógép operációs rendszerén,
- mik a jó minőségű szoftver jellemzői.

Mi a szoftver?

Hogyan fejlesztik a szoftvert?

Mi segítheti a szoftvertervezést és -fejlesztést?

Mit csinál a programozó?

Milyen szolgáltatásai vannak az operációs rendszernek?

Hardver és szoftver

Szoftvertechnika

A szoftver fajtái

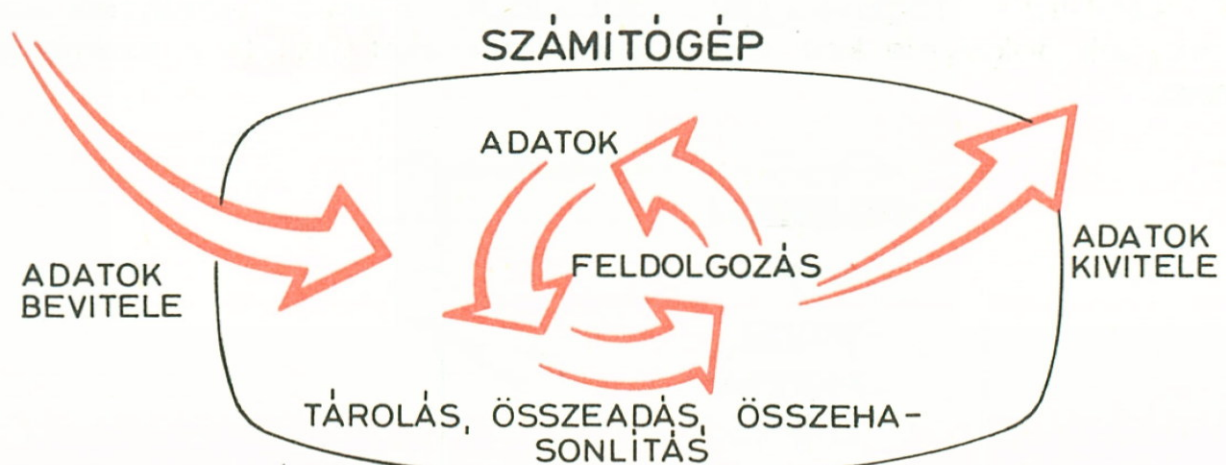
A szoftver minősége

Hardver és szoftver

A számítógépek rendkívüli teljesítményekre képesek. Segítségükkel nemcsak rakétákat vagy ipari berendezéseket lehet irányítani; alkalmazásukkal lehetővé válhat pl. az is, hogy adatok ezreiből a másodperc tört része alatt meg lehessen találni egy kívánt információt. Egy számítógép képes rá, hogy pl. a talált tünetek alapján betegségeket diagnosztizáljon és terápiás javaslatot adjon. Számítógép segítségével egész vállalatokat lehet vezetni és irányítani. Nézzünk meg pl. egy konkrét számítógép-alkalmazást! Egy árucikk eladásakor a számítógép megírja a számlát, egyidejűleg módosítja a raktárkészletet, figyeli az ellenérték befolyását, szükség esetén figyelmeztetéseket küld, kellő időben gondoskodik a nyersanyagról, vezérli a szerszámgepeket a termelésben, statisztikát készít az eladásokról, kezeli a személyzeti és az ügyféladatokat, végrehajtja a bérszámfejtést, és elkészíti az átutalási megbízásokat. Ugyanez a gép tárol mindenfajta dokumentumot, rendezi és kiértékeli ezeket, és egy kulcsszó megadására előkeresi a dokumentumot a felhasználó számára. Mindezen különböző feladatokat egy és ugyanaz a számítógép képes ellátni!

Ehhez legelőször is szükség van a **műszaki berendezésre**, a számítógépre, amelyet műszakilag úgy kell megalkotni, hogy alkalmas legyen:

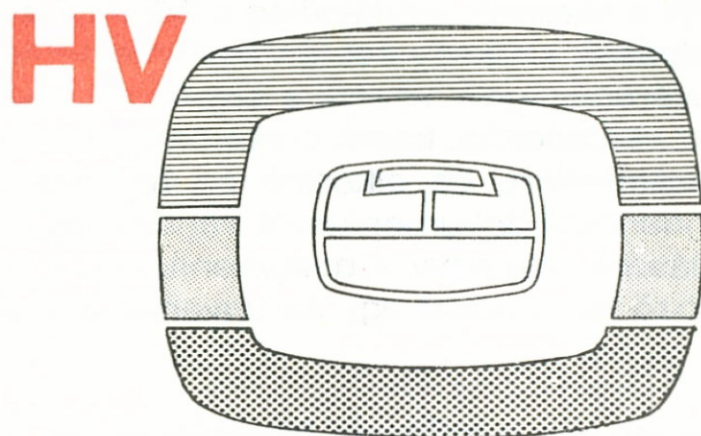
- az adatok vételére;
- az adatok feldolgozására (pl. tárolásra, összeadásra, összehasonlításra);
- a feldolgozás befejezése után az adatok szolgáltatására (pl. a számítási eredményeknek nyomtatott lista alakjában, vagy folyamatirányításhoz impulzusok formájában való kiadása stb.).



E műszaki berendezések összességét nevezik **hardvernek**. A szó az angol nyelvből származik (hardware: kemény [fém-] áru). Egy számítógép hardverje lényegében a következőkből áll:

- a központi egységből, ahol a feldolgozás megtörténik;
- a perifériákból, amelyek az adatok és programok beviteléhez, kiviteléhez és tárolásához szükségesek.

A hardver kutató- és fejlesztőintézetekben tervezett és gyárakban, üzemekben előállított termék. A hardver ábrázolására sorozatunk könyveiben ezt a jelképet fogjuk használni:

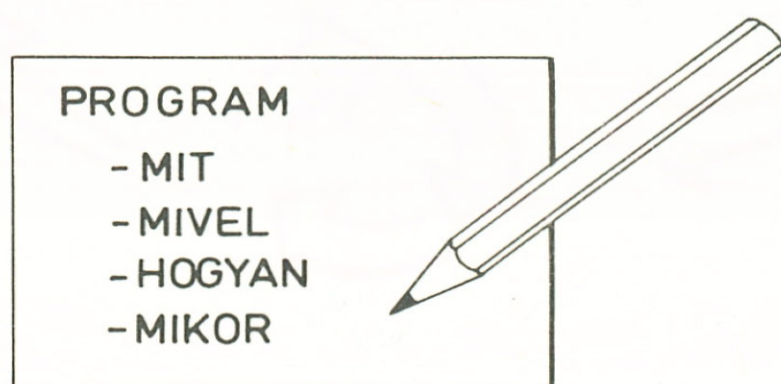


Ahhoz azonban, hogy a hardverrel valamilyen feladatot meg lehessen oldani, a számítógépet programozni kell: azaz meg kell határozni azt, hogy a hardver

- *mit,*
- *mely adatokkal,*
- *milyen sorrendben és*
- *milyen eredménnyel*

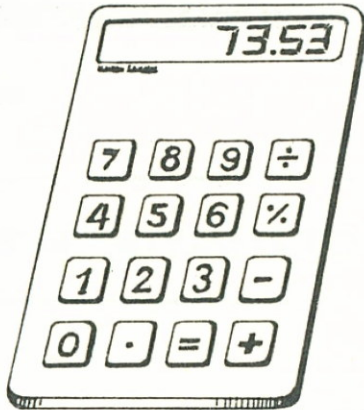
hajtson végre.

Elő kell tehát írni — rögzíteni kell — azt a logikai menetet, amelynek alapján a hardvernek dolgoznia kell. Ezt a logikai menetet írja elő a **számítógép-program**.



A hardver két alapvetően eltérő módon programozható. Egyik esetben a hardver olyan áramköri elemeket tartalmaz, amelyekben előre meghatározott célra szolgáló programot lehet rögzíteni. A hardver ilyen esetben természetesen csak az előre meghatározott célra használható.

Példa:

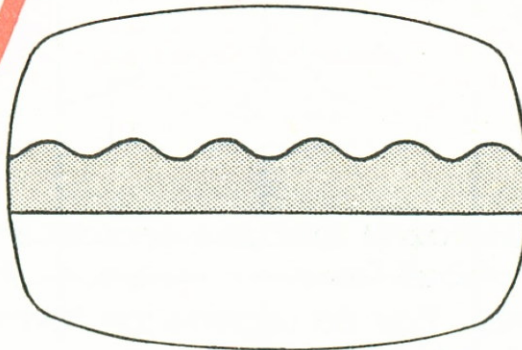


Egy egészen egyszerű zsebszámológépben egy olyan rögzített program van tárolva, amely a négy alapműveletet tartalmazza.

Ha a második esetben abból indulunk ki, hogy a hardvert nemcsak egyetlen célra akarjuk alkalmazni, hanem sok különböző feladatot akarunk vele megoldani, így nem célszerű a hardvert megváltoztathatatlanul programozni, hanem tetszőlegesen megválaszthatóan kell azokat az utasításokat kialakítani, amelyek szerint a hardvernek mindenkor dolgoznia kell. A hardver vezérlését tehát mindig új programoknak kell átvennie.

A számítógépprogramokat — a hardver szó párjaként — **szoftvernek** nevezik. A szó az angol nyelvből származik (a software elnevezés a hardware analógiájára képződött). A szoftvert könyveinkben a következő jelképpel ábrázoljuk:

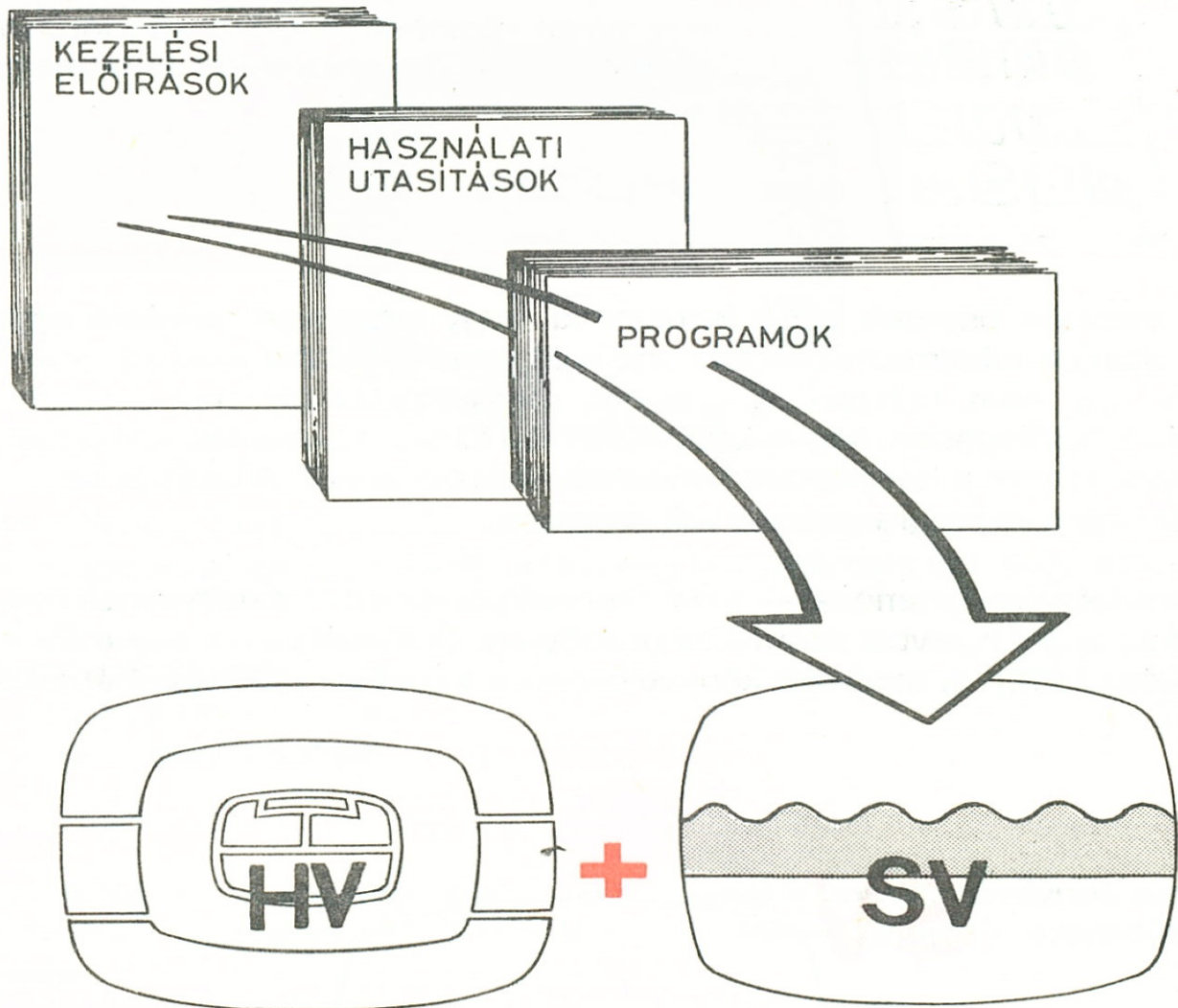
SV



A szoftver fejlesztése tervszerű, rendszerezett eljárást kíván meg, és hasonló ráfordítással jár, mint egy új hardvertermék kifejlesztése.

■ A szoftverfejlesztés eredménye: a végrehajtható **programok**.

A szoftver elnevezés gyűjtőfogalom valamennyi számítógépprogram megjelölésére, beleértve azokat az írott anyagokat is, amelyek ahhoz kellnek, hogy a programokat használni lehessen, vagyis a használati utasításokat, a kezelési előírásokat is.



Egy program egy meghatározott feladat megoldásához szükséges tevékenységek logikus sora. Különböző feladatok megoldásához egymástól különböző programokra van szükség. Egy és ugyanazon hardver számára nagyon sok különböző programot lehet kidolgozni.

A program egyes tevékenységi lépései az **utasítások**.

■ A program az **utasítások logikus sorozata**.

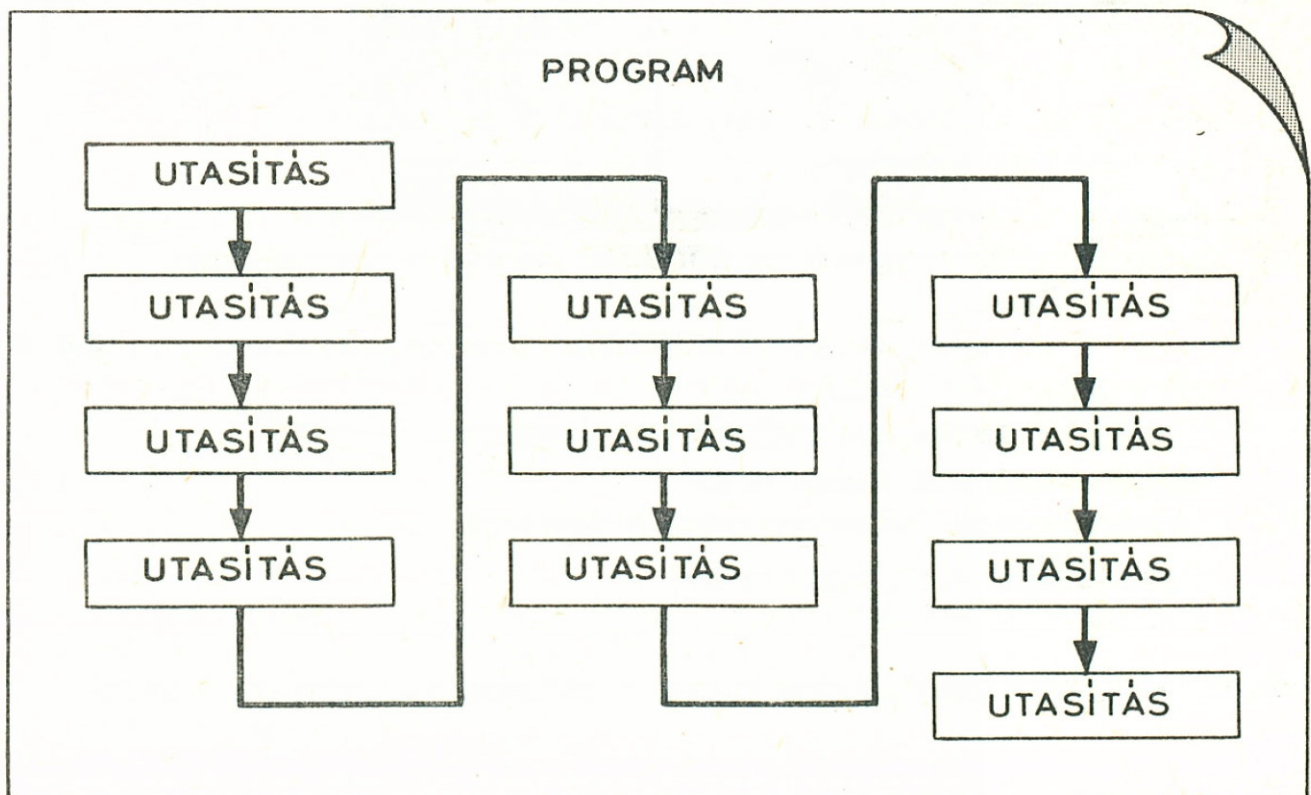
Egy utasítás pontosan megmondja a számítógépnek, hogy részleteiben mit kell tennie: pl. add össze, rendezz, hasonlíts össze! Egy utasítás információkat tartalmaz arról, hogy mit kell csinálni, és hol lehet megtalálni a központi tárban a feldolgozandó anyagokat.

Példa:

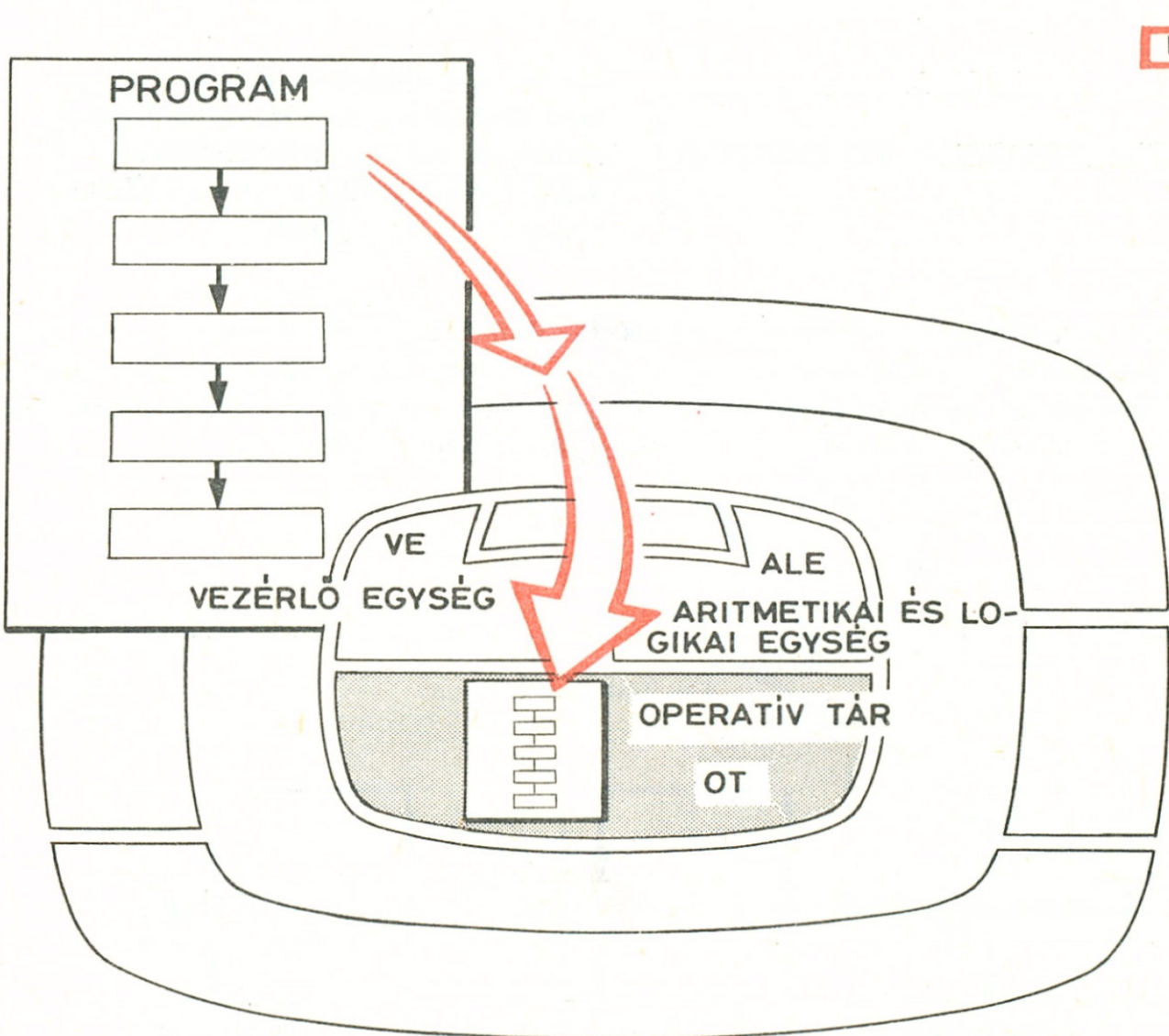
ADD MENNYISÉG TO ÖSSZEG

Ez az utasítás azt jelenti:
add össze azokat az értékeket,
amelyek az operatív tárban
a **MENNYISÉG** és az **ÖSSZEG**
néven vannak tárolva.

Minden számítógép csak egy meghatározott számú utasítást „ért meg”. Az utasítások együttese a számítógép szótára. Ezt a számítógép utasításkészletének is nevezik. A programozó feladata, hogy a probléma megoldását ennek az utasításkészletnek a segítségével megfogalmazza.



Minden programot, annak végrehajtása előtt, be kell tölteni a központi egység **operatív tárába**. A végrehajtáshoz a vezérlőegység az utasításokat egymás után lehívja az operatív tárból. A vezérlőegység megállapítja, hogy melyik utasításról van szó és kiadja a megfelelő vezérlőjeleket a hardver részére. Ennek alapján az indított hardveregységek végrehajtják a kívánt feldolgozási műveleteket. Egy összeadási utasítás esetén pl. a vezérlőegység az operatív tárat az összeadandó adatok rendelkezésre bocsátására készíti, majd az aritmetikai és logikai egységnek vezérlőjeleket ad, amelyek hatására az adatok összeadása létrejön.





1. Mit rögzítenek a számítógépprogramok?

Jelölje meg a helyes állításokat!

Azt, hogy mit hajtson végre a hardver,
pl. összeadást, összehasonlítást, bevitelt?

Azt, hogy mely adatokkal kell műveletet végezni,
pl. az ügyfélállománnyal, Ft-összegekkel stb.?

Azt, hogy milyen sorrendben kell
a műveleti lépéseket végrehajtani?

Azt, hogy milyen alakban várjuk
az eredményt?

2. Mi teszi lehetővé, hogy egy és ugyanazzal a hardverrel egy vállalat bérszámfejtését és raktári nyilvántartását egyaránt végre lehet hajtani?

.....

3. Mely állítások igazak a szoftverre, ill. a fejlesztésére?

A szoftver egy gyűjtőfogalom, amely a programok összességét
jelenti, beleértve a kezelési és használati utasításokat is.

A szoftver fejlesztéséhez tervszerű, rendszerezett
munkára van szükség.

A szoftverfejlesztés költségei összemérhetők egy új
hardvertermék kifejlesztéséhez szükséges ráfordításokkal.

4. Mit nevezünk egy számítógép utasításkészletének?

Azokat a programokat,
amelyek egy géphez rendelkezésre állnak.

Azon utasítások összességét,
amelyek a programokban előfordulhatnak.

Azon utasítások összességét,
amelyek a számítógép utasításkészletében vannak.

5. Hol kell egy program utasításait tárolni ahhoz, hogy a program végrehajtható legyen?

.....



Válaszok

1. A számítógépprogramok rögzítik, hogy:

- mit hajtson végre a hardver, pl. összeadást, összehasonlítást, bevitelt;
- mely adatokkal kell műveletet végezni, pl. az ügyfélállománnyal, Ft-összegekkel stb.;
- milyen sorrendben kell a műveleti lépéseket végrehajtani;
- milyen alakban várjuk az eredményt.

2. **Cserélhető programok** teszik lehetővé azt, hogy egy és ugyanazon hardvert a legkülönbözőbb feladatok elvégzésére lehessen alkalmazni.

3. Valamennyi állítás igaz a szoftverre!

- A szoftver gyűjtőfogalom, amely a programok összességét jelenti, beleértve a kezelési és használati utasításokat is.
- A szoftver fejlesztéséhez tervszerű, rendszerezett munkára van szükség.
- A szoftverfejlesztés költségei összehasonlíthatók egy új hardvertermék kifejlesztéséhez szükséges ráfordításokkal.

4. Egy számítógép utasításkészlete alatt értjük:

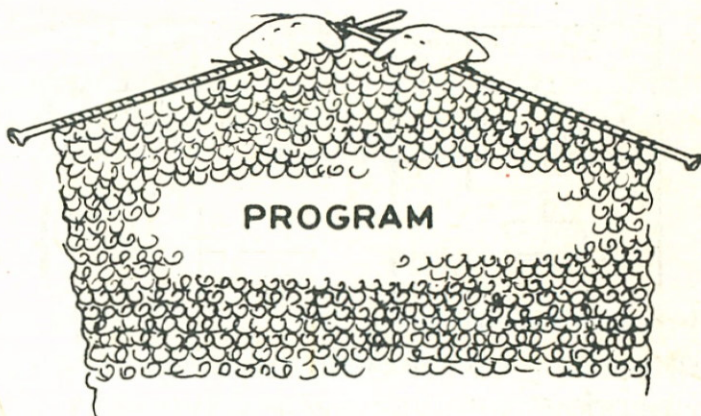
- azon utasítások összességét, amelyek a programokban előfordulhatnak.
- azon utasítások összességét,
- amelyek a számítógép utasításkészletében vannak.

5. A központi egység **operatív tárában** kell a program utasításait tárolni ahhoz, hogy a program végrehajtható legyen.

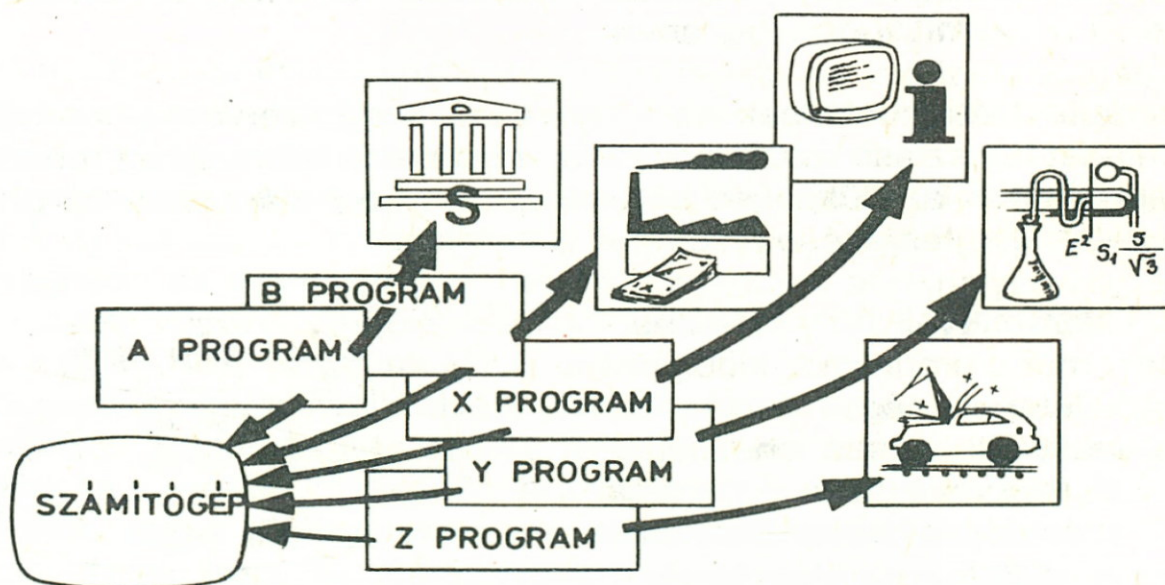
Szoftvertechnika

A számítástechnika kezdeti időszakában a számítógépet először csak egyedi feladatok megoldására használták. Ehhez egyedi programokat dolgoztak ki, tekintet nélkül arra, hogy azok más felhasználók számára érthetők vagy netán használhatók voltak-e.

Pl. egy műszaki-tudományos alkalmazói program programozója gyakran saját programjának egyetlen alkalmazója is volt. A számítástechnika széles körű elterjedése és újabb alkalmazási területek kibontakozása összetettebb feladatmegfogalmazásokat tettek kszükségessé, amelyekhez az egyedi problémák egyéni megoldásának módszere többé már nem volt megfelelő.

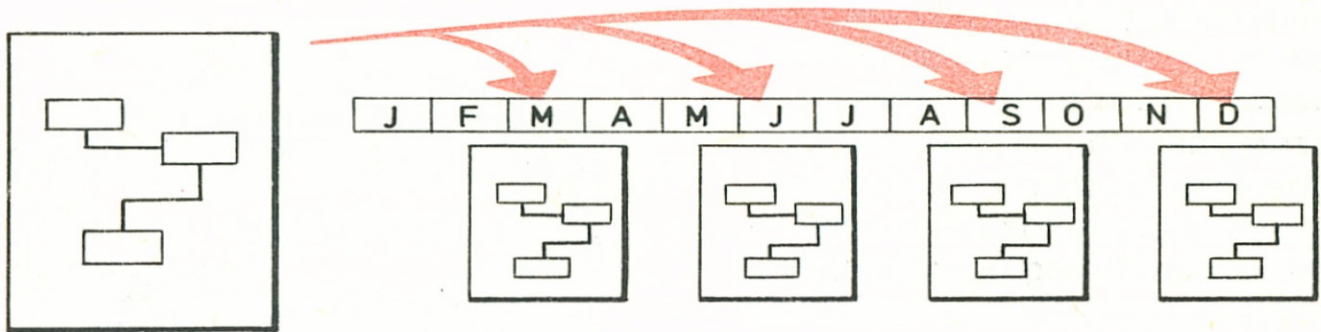


A **szoftverfejlesztés** során nemcsak az egyes programok többé-kevésbé esetleges kidolgozására van szükség.



A szoftverfejlesztés átfogja egy számítógépes eljárás teljes **tervezési, kivitelezési és bevezetési folyamatát.**

Egy **eljárás** teljesen általános értelemben egy feladat vagy feladatláncolat elvégzésének olyan általános szabályozása, melynek hatására az egyes tevékenységeknek mindig azonos módon kell lezajlani. Ha a feladatok lebonyolításához „segédeszközként” számítógépet használnak, akkor azt **számítógépes eljárásnak** nevezzük.

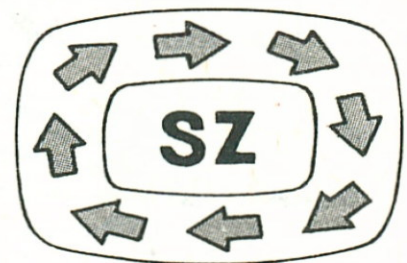


Számítógépes eljárás lehet pl.: egy számítógéppel segített személynyilvántartás, egy központi adatbank kiépítése, vagy szerszámgépek elektronikus vezérlése.

A szoftverfejlesztés hosszadalmas fejlesztési folyamat, amelynek végtermékei a **számítógépprogramok**.

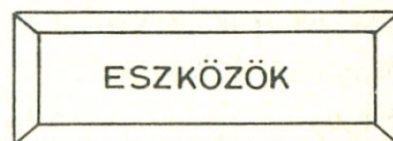
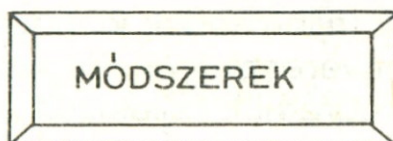
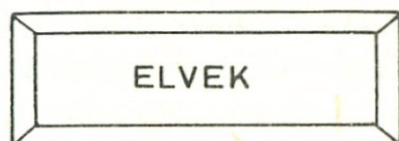
A szoftverfejlesztés munkáinak nagy része kizárólag szervezési jellegű, pl.: a teljes feladat felosztása részfeladatokra, a feladatok személyhez rendelése, az adatáramlás és a szükséges berendezések meghatározása, az űrlapok és bizonylatok megtervezése stb.

Azonban valamennyi ilyen szervezési intézkedést már a programok kidolgozása előtt meg kell tenni azért, hogy a munkát értelmesen lehessen elkezdni. A szervezéssel e sorozatnak a Kulcs a számítógéphez — Szervezés c. kötete foglalkozik részletesebben.



A szoftvertermék olyan, mint a többi ipari termék, ezért azokhoz hasonlóan ezt is **tervszerűen** és **rendszeresen** kell fejleszteni.

Ahogy az ipari termékek tervezéséhez, fejlesztéséhez és gyártásához elveket (pl. a feladatszerkezeti felosztásával kapcsolatosan) dolgoznak ki, segédeszközöket, módszereket (pl. futószalagos gyártás, szerkesztési rajzok) és szerszámokat (pl. kalapács, esztergapad) használnak, amelyek nélkül ma az ipari termelés már elképzelhetetlen, úgy a szoftver „termeléséhez” is **elveket, módszereket** és **eszközöket** kellett alkotni, olyanokat, amelyek ésszerű, gazdaságos és állandóan ellenőrizhető fejlesztést tettek lehetővé számítógépes eljárásokhoz is.



A számítógépes eljárások kifejlesztését speciálisan erre a célra kidolgozott elvek, módszerek és eszközök segítségével **szoftvertchnikának** — software engineering-nek — nevezzük. Ez a szó is, mint annyi más a számítástechnikában, az angol nyelvből származik, és bizonyára abból a gondolatból született meg, hogy fenti értelemben vett módszeres szoftverfejlesztést a mérnöki fejlesztőtevékenységhez lehet hasonlítani. Mindkét esetben az ötlettől a „termék” jövőbeli bevezetéséig gondosan előre meg kell tervezni a fejlesztőmunkát, mielőtt hozzákezdünk a fejlesztéshez és a gyártáshoz.

Elvek, módszerek és eszközök a szoftverfejlesztés valamennyi stádiumában léteznek; vannak, amelyek különösen a szervezési munkát támogatják, és mások, amelyek a kivitelezési munkákban segítenek.

A szoftvertchnika egyik elve szerint pl. a **fejlesztési folyamatot szakaszokra** (fázisokra) kell tagolni, amelyeknek a tartalmát és a szükséges tevékenységeket világosan és egyértelműen meg kell határozni, és amelyek végrehajtását egy szakasz végén a projektvezetésnek ellenőriznie kell. Bővebben erre egy következő fejezetben térünk ki.

A szoftvertchnika további elve a **projekteken végzett munka** összehangolása. Egy számítógépes eljárás fejlesztésekor a számítástechnikai szakemberek, a szerzők, a megbízó és a jövőbeli felhasználók szorosan együttműködnek, és ez az ismeretek optimális kihasználásához vezet.

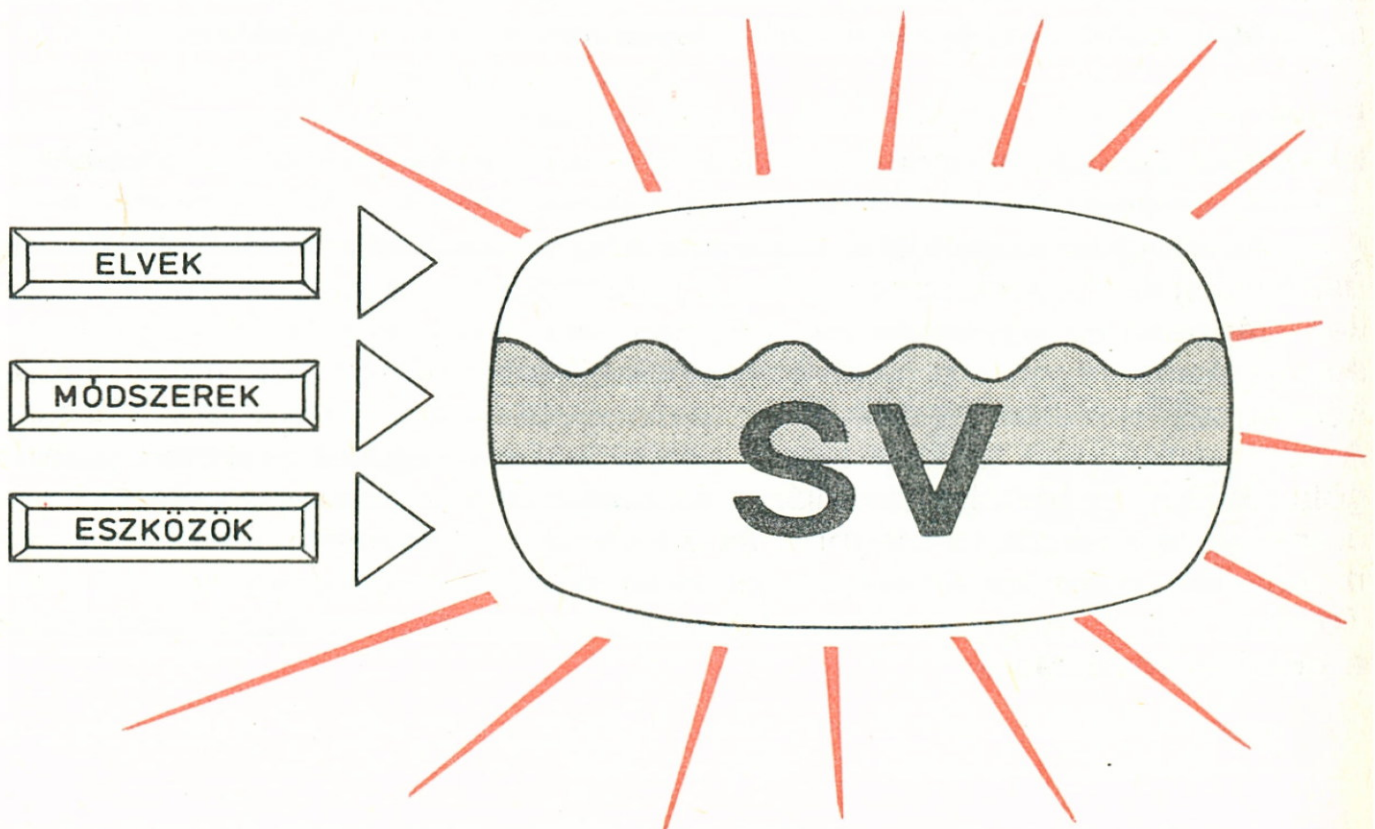
További elv pl., hogy az eljárás fejlesztési szakaszait állandóan **követni kell a dokumentációkkal**.

Azok a **módszerek**, amelyek a számítógépes eljárások fejlesztése során alkalmazhatók, igen sokrétűek. Vannak pl. olyan **tervezési és fejlesztési módszerek**, amelyeket a műszaki élet és a szervezés területéről vettek át (pl. a modultechnika a tv-gyártás területéről, vagy a hálótervezés módszere), és amelyeket a szoftverfejlesztés speciális igényeihez igazítottak. Továbbá léteznek olyan **grafikus módszerek**, amelyek az ipari termékek szerkesztési terveihez hasonlítanak. Ezeket speciálisan a számítástechnikai igényekhez alakították ki, és lehetővé teszik a szoftver konstrukciós elveinek szemléletes ábrázolását (pl. program-folyamatábrák, struktrogramok stb.).

A szoftverfejlesztés valamennyi szakaszában léteznek olyan **eszközök** is, amelyeket külön arra a célra fejlesztettek ki, hogy egyszerűsítsék a szoftverfejlesztési folyamatot. A szoftvereszközök maguk is **számítógépprogramok**. Ezek támogathatják pl. a programok fejlesztőit a kidolgozandó program tervezésében és tesztelésében.

A lehetséges módszerekből és eszközökből természetesen projektenként kell kiválasztani a megfelelőket.

Ezen eszköztár célzatos alkalmazásával lehet megteremteni a **jó minőségű szoftver** fejlesztésének alapjait. A legfontosabb szoftvertechnikai elvek, módszerek és eszközök leírásával a következő fejezetek foglalkoznak.





1. Mi a szoftverfejlesztés? Keresztezze be a helyes állításokat!

A szoftverfejlesztés egy számítógépes eljárás tervezésének, fejlesztésének és bevezetésének teljes folyamata.

A szoftverfejlesztés egy számítógépes eljárás keretén belül a programok fejlesztését jelenti.

A szoftverfejlesztés programok írása.

2. Hogyan nevezzük a szoftver módszeres fejlesztését az erre a célra kidolgozott elvek, módszerek és eszközök tudatos alkalmazásával?

.....

3. Nevezze meg a szoftvertechnika két fontos elvét!

.....

.....

4. Mik a szoftvertechnika eszközei?

Számítógépprogramok.

Folyamatábrák.

Adatfeldolgozó berendezések.



Válaszok

1. A szoftverfejlesztés:

- A szoftverfejlesztés egy számítógépes eljárás tervezésének, fejlesztésének és bevezetésének teljes folyamata.

2. Az erre a célra kidolgozott elvek, módszerek és eszközök tudatos alkalmazásával történő szoftverfejlesztést **szoftvertechnikának** nevezzük.

3. A szoftvertechnika két fontos elve:

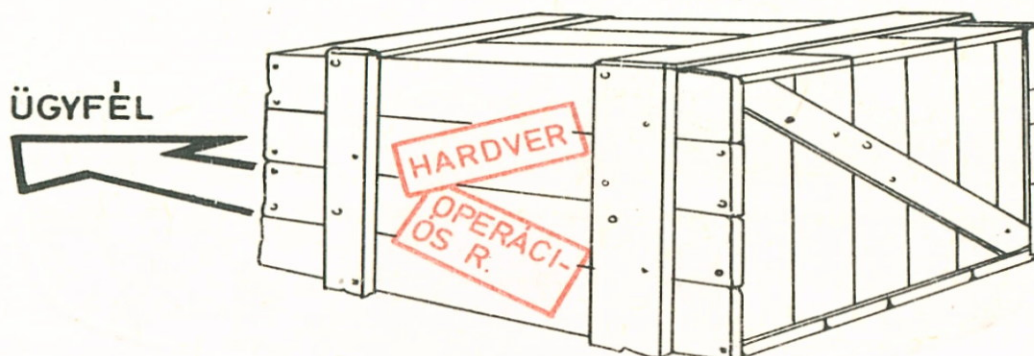
- a **fejlesztési folyamat szakaszokra bontása** és
- a **projektrendszerű munkavégzés**.

4. A szoftvertechnika eszközei:

- Számítógépprogramok.

A szoftver fajtái

A számítógépgyártók a hardverrel együtt a szoftver egy részét is szállítják — és pedig a szoftvernek azt a részét, amely a gép üzeméhez elengedhetetlen és alapvető tartozéknak számít.



Ezt az alapszoftvert, amely nélkül a gép nem működtethető, **rendszer-szoftvernek** nevezzük.



A rendszerszoftverhez tartozik valamennyi olyan program, amely a számítógépek közvetlen működtetését segíti azzal, hogy a számítógép egyes részeinek együttműködését lehetővé teszi, azt vezérli és ellenőrzi.

Rendszerszoftver nélkül nem lehetne a számítógépbe adatokat bevinni, tárolni, a feldolgozást vezérelni és adatokat kivinni.

A rendszerszoftver azonban nem alkalmas arra, hogy egy adott — a felhasználó által megfogalmazott — feladatot megoldjon.

Ezért van szükség **alkalmazói szoftverre**.

Az alkalmazói szoftverhez tartoznak mindazon programok, amelyeket azért fejlesztettek ki, hogy a számítógépek felhasználóinak — a „végfelhasználóknak” — a speciális feladatait megoldják.



Alkalmazói programok pl. a bérszámfejtő programok, tájékoztatórendszerek, folyamatirányító programok stb. Ezeket csak részben készítik maguk a felhasználók. Az irányzat az, hogy általános alkalmazási esetekre (pl. bérszámfejtés) központilag készítenek programokat, majd ezeket forgalmazzák. A számítógépek gyártói a hardverrel és a rendszerszoftverrel együtt az alkalmazói programok jelentős részét is szállítják. Ezután a programokat már csak illeszteni kell a felhasználó saját igényeihez.

A rendszerszoftver mellett, amely ahhoz szükséges, hogy a számítógép egyáltalán működni tudjon, valamint az alkalmazói szoftver mellett, amely a felhasználó speciális feladatainak megoldására készül, létezik egy harmadik szoftverfajta: a **rendszerközeli szoftver**.



A rendszerközeli szoftver funkciójában az alkalmazói szoftver és a rendszerszoftver között áll.

A felhasználónak és a programfejlesztőnek egyaránt kényelmesebbé teszi a munkáját.

Egy berendezés alkalmazója számára a rendszerközeli szoftver olyan programokat tartalmazhat, amelyek pl. elvégzik az adatok rendezését és egyesítését, vagy olyan programokat, amelyek jegyzik a felhasznált gépidőket stb. A programfejlesztő számára a rendszerközeli szoftver olyan programok, együttese, amelyek az alkalmazói programok megírását, tesztelését és futását megkönnyítik. Minden szoftverfejlesztő eszköz a rendszerközeli szoftverhez tartozik.

A szoftverre utaló jelkép, amely az egész könyvben végig megtalálható, minden esetben a rendszerszoftver és az alkalmazói szoftver között elhelyezkedő rendszerközeli szoftvert jelenti, azért, mert ez segíti a felhasználót abban, hogy a számítógép lehetőségeit jobban kihasználhassa.

A számítógép műszaki kiépítettségétől (konfigurációjától) és a gép tervezett felhasználási területétől függően kell a rendszerszoftver és a rendszerközeli szoftver választéka felől dönteni, amelynek azután állandóan a felhasználó rendelkezésére állanak. Ez a speciális szoftverválaszték a számítógép **operációs rendszere**. Egy és ugyanazt a számítógépet az alkalmazási területtől függően különböző operációs rendszerekkel lehet ellátni.



A szoftver fejlesztését — függetlenül attól, hogy rendszerszoftverről, rendszerközeli szoftverről vagy alkalmazói szoftverről van-e szó — mindig a szoftvertechnika által ajánlott elvek, módszerek és eszközök alkalmazásával kell elvégezni, annak érdekében, hogy növekedjék a jó szoftver létrejöttének valószínűsége.

Azt, hogy mit értünk jó szoftveren, a következő fejezetben ismertetjük.



1. Melyik szoftverfajta-hoz tartoznak a következő programok?

- Rendező- és egyesítőprogramok
- Alkalmazói programok utasításait interpretáló (értelmező)
és a hardvert ennek megfelelően vezérlő
programok
- Az értéktöbbletadót kiszámító program
- A tesztadatokat generáló (előállító) program
- A tájékoztatórendszer
információ-visszakereső programja

2. Miben segíthet a rendszerközeli szoftver?

Jelölje meg a jó válaszokat!

- Alkalmazói szoftver készítésében.
- Rendszerszoftver készítésében.
- Rendszerközeli szoftver készítésében.
- Adatok rendezésénél.
- A számítógép használatában.

3. A következő állítások közül melyek igazak az operációs rendszerre?

- Egy számítógépre csak egy speciális operációs rendszer létezik.
- Az operációs rendszer rendszerszoftver
és rendszerközeli szoftver elemekből áll.
- Az operációs rendszert a gyártó rendszerint
a számítógéppel együtt szállítja.

Válaszok

1. A programok a következő szoftverfajtákhoz tartoznak:

A rendező- és egyesítőprogramok **rendszerközeli szoftver**.

Az alkalmazói programok utasításait interpretáló és a hardvert ennek megfelelően vezérlő programok: **rendszerszoftver**.

Az értéktöbbletadót kiszámító program: **alkalmazói szoftver**.

A tesztadatokat generáló (előállító) program: **rendszerközeli szoftver**.

A tájékoztatórendszer információ-visszakereső programjai: **alkalmazói szoftver**.

2. *A rendszerközi szoftver valamennyi felsorolt tevékenységben segíthet, mert függetlenül a fejlesztendő szoftvertől az alkalmazónak és a programfejlesztőnek is kényelmesebbé teszi a munkáját.*

- Alkalmazói szoftver készítésében.
- Rendszerszoftver készítésében.
- Rendszerközeli szoftver készítésében.
- Adatok rendezésénél.
- A számítógép használatában.

3. Az operációs rendszerre a következő válaszok helytállóak.

- Az operációs rendszer rendszerszoftver és rendszerközeli szoftver elemekből áll.
- Az operációs rendszert a gyártó rendszerint a számítógéppel együtt szállítja.

A szoftver minősége

A szoftver elkészítése összehasonlítható egy prototípus elkészítésével a műszaki életben. Mindkét esetben igen nagy figyelemmel járnak el a tervezés és a kivitelezés során. Az egyszer kifejlesztett termék többszörös reprodukálása ezek után viszonylag problémamentes. Az elérendő cél tehát az, hogy már első menetben **jó minőségű szoftvert** állítsunk elő.

A szoftver akkor jó minőségű, ha a következő jellemzői vannak:

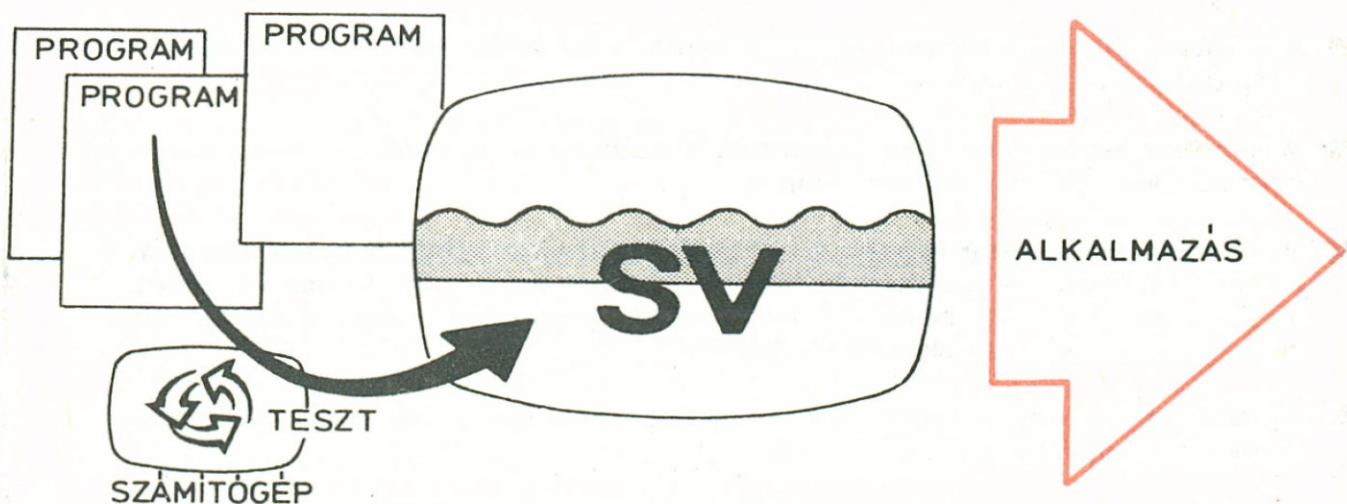
- **A szoftver legyen megbízható**, azaz minden lehetséges bemenő adatra a legnagyobb valószínűséggel a helyes eredményeket kell adnia.
- **A szoftver legyen „életerős”**, azaz ismerje fel a szoftver- és a hardverhibákat, a gépkezelői vagy az átviteli hibákat, és ezek negatív hatásait amennyire lehet hárítsa el.
- **A szoftver legyen könnyen módosítható és bővíthető**, azaz azok a programozók is, akik nem vettek részt a fejlesztésben, azt megváltoztatott vagy kibővített feladatokra is alkalmassá tudják tenni.
- **A szoftver legyen átvihető (portabilis)**. A szoftver akkor átvihető (portabilis), ha lehetőleg kevés illesztési ráfordítás árán más számítógépeken és más operációs rendszer alatt is lehet használni.
- **A szoftver legyen könnyen karbantartható**, azaz a hiányosságok a fejlesztő közreműködése nélkül is könnyen felismerhetők és elháríthatók legyenek.
- **A szoftver legyen könnyen kezelhető**. A szoftvert az azzal dolgozókhöz (ügyintéző, operátor) kell igazítani és nem fordítva.
- **A szoftver terjedelme legyen a szolgáltatásainak megfelelő**. A szoftvert úgy kell elkészíteni, hogy a feladat megoldásához szükséges egyik fontos funkció se hiányozzék, de ugyanakkor az csak olyan terjedelmű legyen, hogy elfogadható idő alatt a rendelkezésre álló eszközökkel elkészíthető legyen.
- **A szoftver legyen hatékony**, azaz lehetőleg kevés operatív-tár-kapacitást igényeljen, és lehetőleg gyorsan fusson le.

Az, hogy ezeknek a jellemzőknek mekkora jelentőséget tulajdonítunk, a megoldandó feladattól és azon peremfeltételektől függ, amelyek mellett a szoftvert alkalmazni kell. A szoftver nem tehet eleget egyenlő mértékben valamennyi követelménynek, mivel ezek egy része ellentmondó. Pl. egy olyan programnak, amelyik a változtathatóság és bővíthetőség érdekében jól áttekinthetőnek készült, tárhely-igénye és futási ideje rendszerint nagyobb.

Jó minőségű szoftvert az alábbi **szabályok** betartásával érhetünk el.

- A szoftvertermékek áttekinthető tagolása.
- Állandó dokumentációs követés a fejlesztés minden szakaszában.
- Megfelelő és jól kiképzett programozók alkalmazása, és ezek jól működő csoportokká szervezése.
- A megoldandó feladatok optimális szétosztása a munkatársak között.
- A munkatársak ösztönzése jó minőségű szoftver előállítására.
- Már meglévő programok vagy programrészletek felhasználása.
- A szoftvertechnika korszerű elveinek, módszereinek és eszközeinek alkalmazása.
- Valamennyi szabály betartásának ellenőrzése a projektvezetés által.

Ha a számítógépes eljáráshoz tartozó valamennyi programot a felsorolt szabályok betartásával fejlesztették, tesztelték, és ha valamennyi programnak elkészült a pontos leírása és futtatási utasítása, csak akkor beszélhetünk **kész, használatba vehető szoftverről**.





1. A szoftvernek lehetőleg minél több most felsorolt minőségi jellemzője legyen. A szoftver legyen megbízható (1), „életerős” (2), módosítható és bővíthető (3), átvihető (4), karbantartható (5), könnyen kezelhető (6), terjedelme a szolgáltatásainak megfelelő (7), hatékony (8).

Melyik minőségi jellemzőre vezethető vissza, ha egy szoftverről elmondhatók az alábbiak:

Könnyen illeszthető kibővíthető feladatokhoz.

A hibái gyorsan elemezhetők és javíthatók.
Problémamentesen futtatható más számítógépen,
más operációs rendszer alatt is.

Azt teljesíti, amit várnak tőle.
Bármelyik számítástechnikai szakember képes használni,
és le tudja futtatni.

Tárkapacitás-igénye és futási ideje kicsi.

2. Milyen intézkedésekkel lehet jó minőségű szoftvert fejleszteni?
Soroljon fel annyi szabályt amennyit tud!

.....

.....

.....

.....

.....

3. Lehetséges-e valamennyi követelményt egyidejűleg kielégíteni?

Igen.

Nem.



Válaszok

1. A szoftver jellemzői a következők:

- 3 Könnyen illeszthető kibővített feladatokhoz.
- 5 A hibák gyorsan elemezhetők és javíthatók.
- 4 Problémamentesen futtatható más számítógépen, más operációs rendszer alatt is.
- 7 Azt teljesíti, amit várnak tőle.
- 6 Bármelyik számítástechnikai szakember képes használni és le tudja futtatni.
- 8 Tárhely-igénye és futási ideje kicsi.

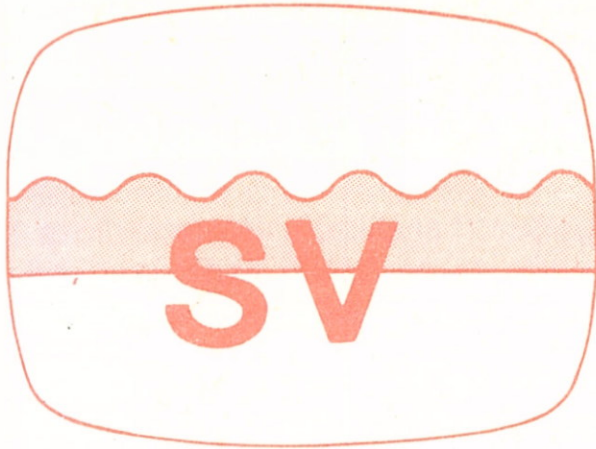
2. Jó minőségű szoftvert az alábbi szabályok betartásával lehet fejleszteni:

- **A szoftvertermékek világos tagolása.**
- **A fejlesztés valamennyi szakaszának állandó dokumentálása.**
- **Jól működő munkacsoportokba szervezett, jól képzett programozók alkalmazása.**
- **Az egyes feladatok optimális szétosztása a munkatársak között.**
- **A munkatársak ösztönzése arra, hogy törekedjenek jó minőségű szoftver kidolgozására.**
- **Már meglévő, bevált programok és programrészek alkalmazása.**
- **A szoftvertechnika korszerű elveinek, módszereinek és eszközeinek felhasználása.**
- **Valamennyi felsorolt szabály betartásának ellenőrzése a projektvezetés által.**

3. Az, hogy az egyes jellemzőknek mekkora jelentőséget tulajdonítunk, a megoldandó feladattól és a peremfeltételektől függ.

Igen.

Nem.



Mi a szoftver?

Hogyan fejlesztik a szoftvert?

Mi segítheti a szoftvertervezést és -fejlesztést?

Mit csinál a programozó?

Milyen szolgáltatásai vannak az operációs rendszernek?

Ebben a fejezetben megtanulja:

- hogyan lehet a szoftverfejlesztés folyamatát elhatárolt szakaszokra felosztani, valamint megfelelő szervezés segítségével a szoftverfejlesztést áttekinthetővé és ellenőrizhetővé tenni;
- miként lehet a szoftverfejlesztés nehéz feladatát különböző szakképesítésű szakértők projekt-munkacsoportokban (projektteamekben) való együttműködésével megoldani.

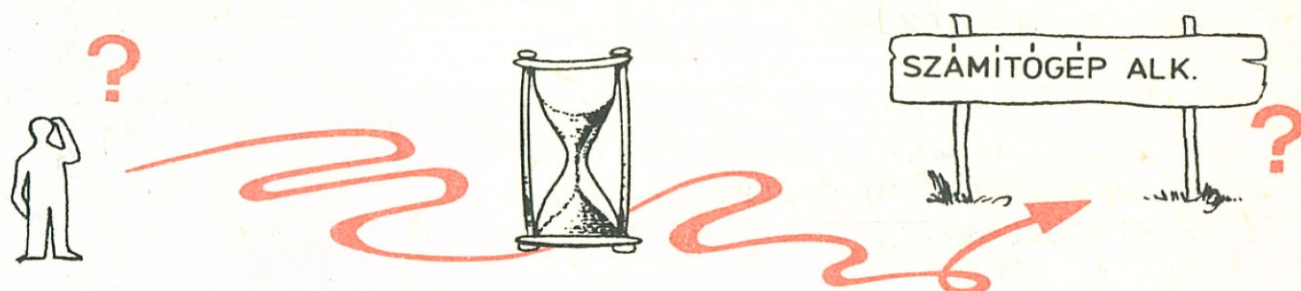
A szoftverfejlesztés szakaszai

Ki fejlessze a szoftvert?

A szoftverfejlesztés szakaszai

A szakaszokra való felosztás alapelvei

A szoftvertermék előállítása hosszadalmas folyamat. A számítógépes eljárás alkalmazásának ötletét és az eljárás bevezetését rendszerint hosszú idő választja el egymástól.



Azért, hogy ezt a hosszadalmas folyamatot áttekinthetővé tegyék és valamennyi végrehajtandó munkát össze tudják hangolni, szükséges volt az egész fejlesztést különálló szakaszokra felosztani, és pontosan definiálni, mely munkákat kell az egyes szakaszokban elvégezni, milyen részeredményekkel ér véget egy-egy szakasz, és mikor indul a másik.



A projekt csak akkor folytatódik a következő szakasszal, ha megrendelő vagy más illetékes szervezet elégedett a munkával, vagyis ha a mindenkorra közbenső célt sikerült elérni. Ebben az összefüggésben beszélünk a célhoz vezető út „mérőköveiről”.

A végrehajtandó feladatok pontos meghatározásával lehet **a munka előrehaladását ellenőrizni, és így az esetleges hibákat időben helyrehozni.**

Azt, hogy hogyan kell egy projektet szakaszokra felosztani, a szoftverfejlesztéssel foglalkozó intézmények egymástól eltérően szabályozzák. A nagyobb projekteket rendszerint jobban felosztják, az egészen kicsi, áttekinthető projekteket esetleg egyáltalán nem osztják fel. Minden esetben fontos, hogy az egyes szakaszokban elvégzendő munkákat pontosan meghatározzák.

Szokásos a szoftverfejlesztési folyamatot **öt szakaszra** felosztani. A következőkben mi is ezt a felosztást ismertetjük egy példán keresztül, amelyben röviden leírjuk, hogy az egyes szakaszokban milyen anyagokat kell elkészíteni, és melyek az esedékes munkák.

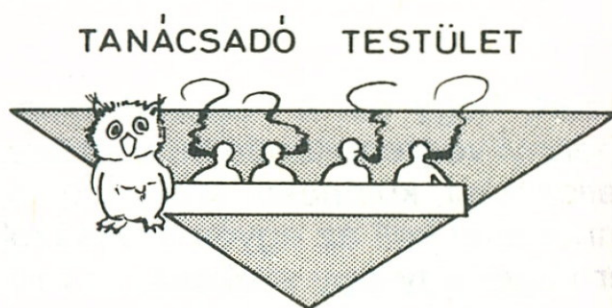
Mivel bizonyos szervezési tevékenységek minden szakaszban ismétlődnek, ezért először ezeket mutatjuk be.

A tulajdonképpeni fejlesztési tevékenységgel párhuzamosan minden szakaszban el kell látni a következő **szervezési tevékenységeket**:

- Minden szakasz kezdetén meghatározzák:
 - kik,
 - mely időpontban,
 - milyen esedékes munkákat és felelősségeket vegyenek át; azaz rögzíteni kell **a szakasz szervezését**.



- Minden szakasz folyamán folyamatosan ellenőrizni és esetleg korigálni kell a tevékenységekre vonatkozó:
 - határidőket,
 - költségeket és
 - esedékes eredményeket.



Szakmai kérdések tisztázásában egy **tanácsadó testület** segíthet.

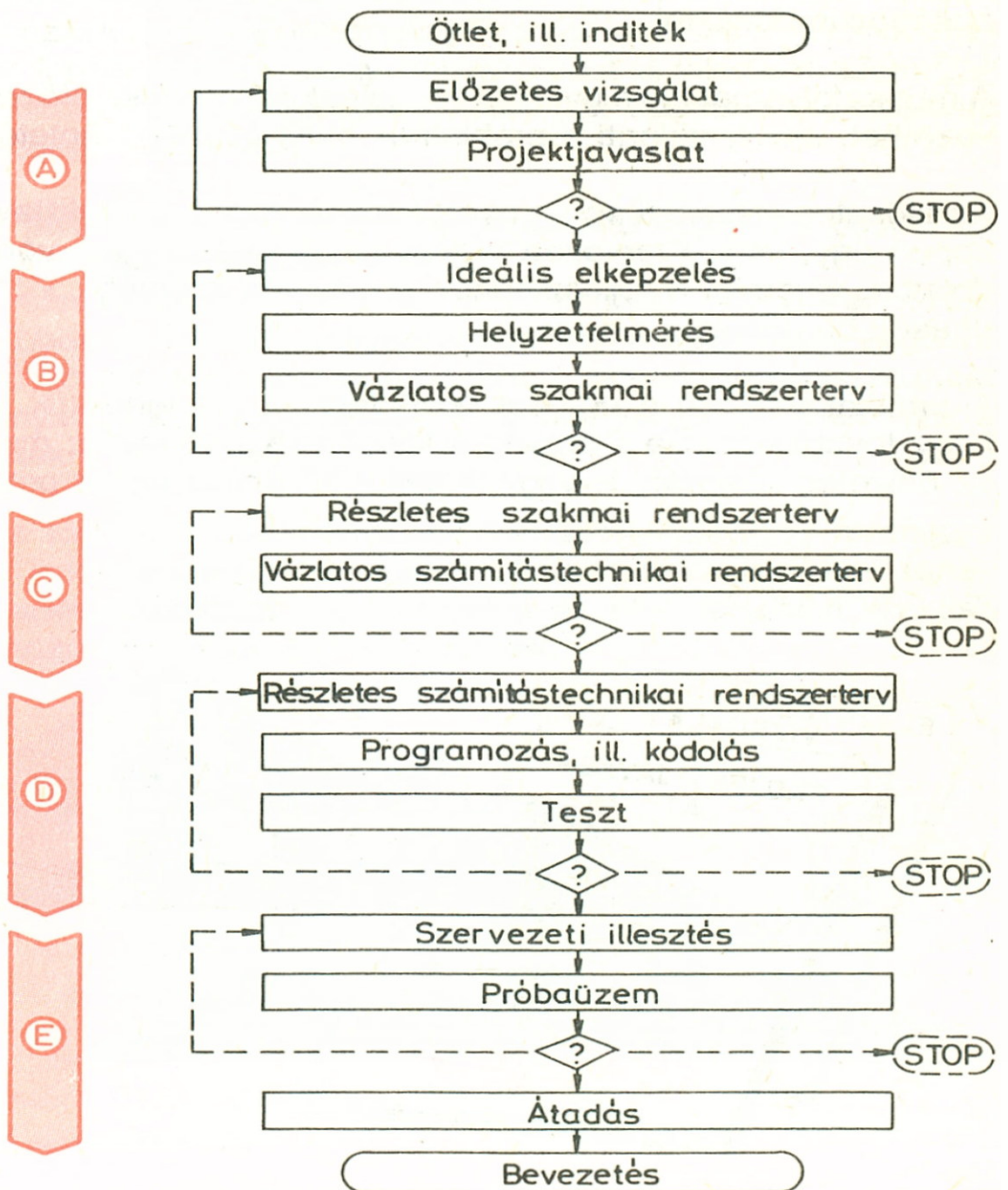
- Minden szakasz befejezésekor az elért eredményeket egy, a vállalatvezetés által kijelölt **döntéshozó testület** elé viszik, amely arról dönt, hogy:
 - a projekt következő szakasza megkezdődhet-e,
 - az éppen lezárt szakaszt más feltételekkel meg kell-e ismételni,
 - a projektet félbeszakítják és befejezik, mivel semmi kilátás sincs a kívánt eredmény elérésére.



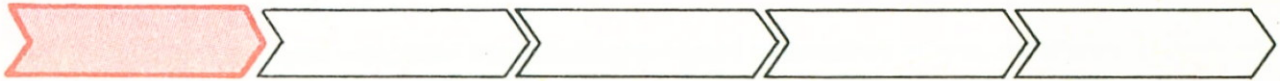
Az öt szakasz, amit ismertetünk: **1.** a javaslati szakasz, **2.** az I. tervezési szakasz, **3.** a II. tervezési szakasz, **4.** az I. megvalósítási szakasz és **5.** a II. megvalósítási szakasz.



A szakaszokat először program-folyamatábra alakjában mutatjuk be, amelyen valamennyi szakasz fontosabb munkáit és tevékenységeit címszavakban megnevezzük.



Javaslati szakasz

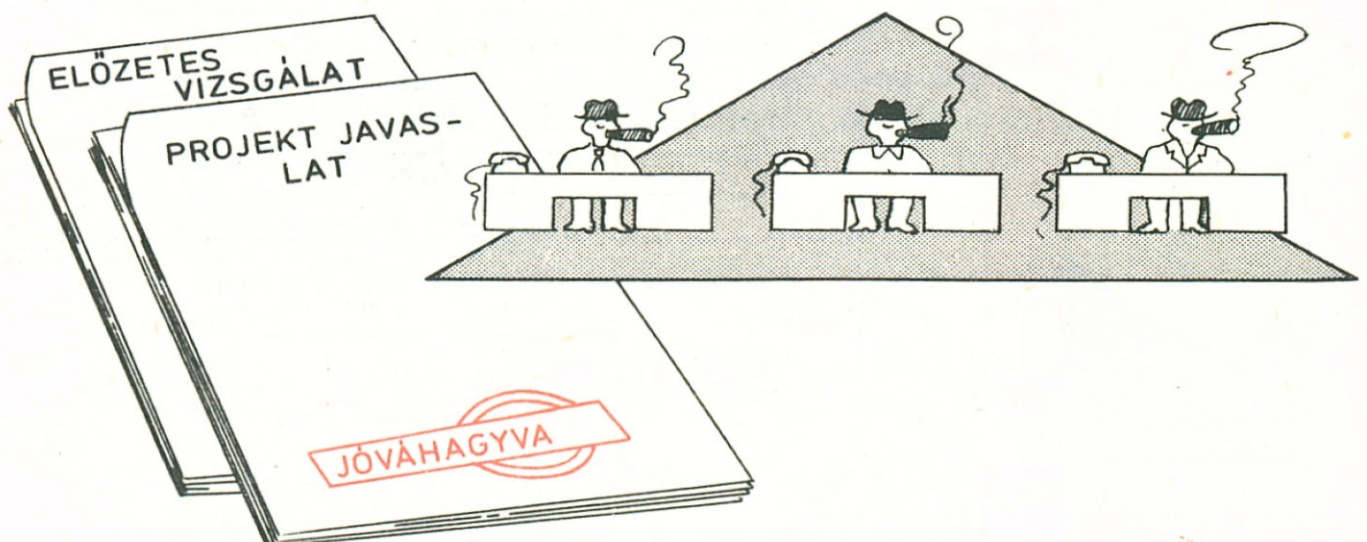


Ez a szakasz a számítógépes eljárás létrehozásának **ötletével** indul. A számítógépes eljárás kezdeményezése a vállalat, hatóság különböző helyeiről érkezik. Újítási vagy módosítási javaslat rendszerint akkor keletkezik, ha a jelenlegi eljárásban hiányosságok merülnek föl, pl. ha bizonyos helyeken nehézségek lépnek föl a feladatok ellátásában, ha az egyes egységek közötti együttműködés nem funkcionál, vagy egyszerűen, ha nem lehet tovább gazdaságosan dolgozni.

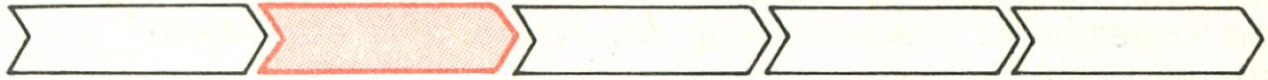
A munkafolyamat ilyen gyenge pontjainak felfedezése **előzetes vizsgálathoz** vezethet, amely kideríti a problémák okait és megszüntetésük lehetőségeit.

Az előzetes vizsgálat során nyert felismerések a **projektjavaslatba** kerülnek éppen úgy, mint a tervezett eljárás gazdaságosságára vonatkozó első gondolatok, valamint a fejlesztéshez szükséges személyi, idő- és költségáfordítások becslései.

A projektjavaslatot a döntést hozó testület elé terjesztik, amely arról dönt, hogy továbbléphet-e a projekt a következő szakaszba; meg kell-e ismételni az előzetes vizsgálatot; vagy le kell-e állítani az egész projektet.



Az I. tervezési szakasz

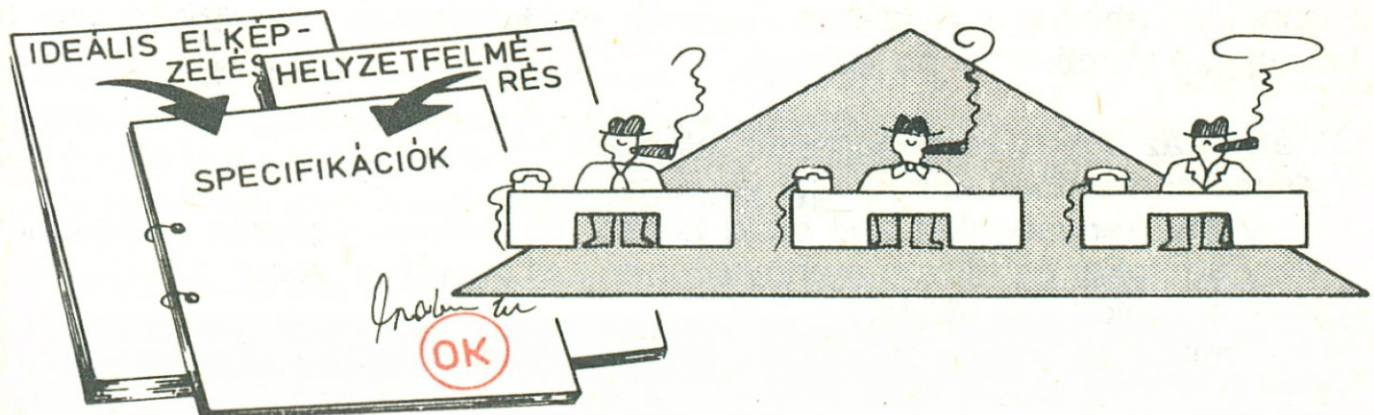


Miután a javaslati szakasz a jóváhagyott projektjavaslattal lezárult, elkezdődik az első tervezési szakasz, amelyben először csak a **szakmai problémát** vizsgálják. A szakmai probléma lehet műszaki természetű, pl. mely esetekben milyen legyen a nyomás, a fordulatszám vagy a hőmérséklet egy műszaki berendezésben. A szakmai probléma üzemgazdasági jellegű is lehet, ha pl. az a kérdés merül fel, hogy az üzemi munkafolyamatban melyik helyen milyen információk keletkeznek és milyen eredményekre van szükség. Ezen a ponton csak másodlagos fontosságú az, hogy hogyan lehet a feladatot számítástechnikailag megoldani. Sőt, *tudatosan elválasztják egymástól a szakmai és a számítástechnikai koncepciót.*

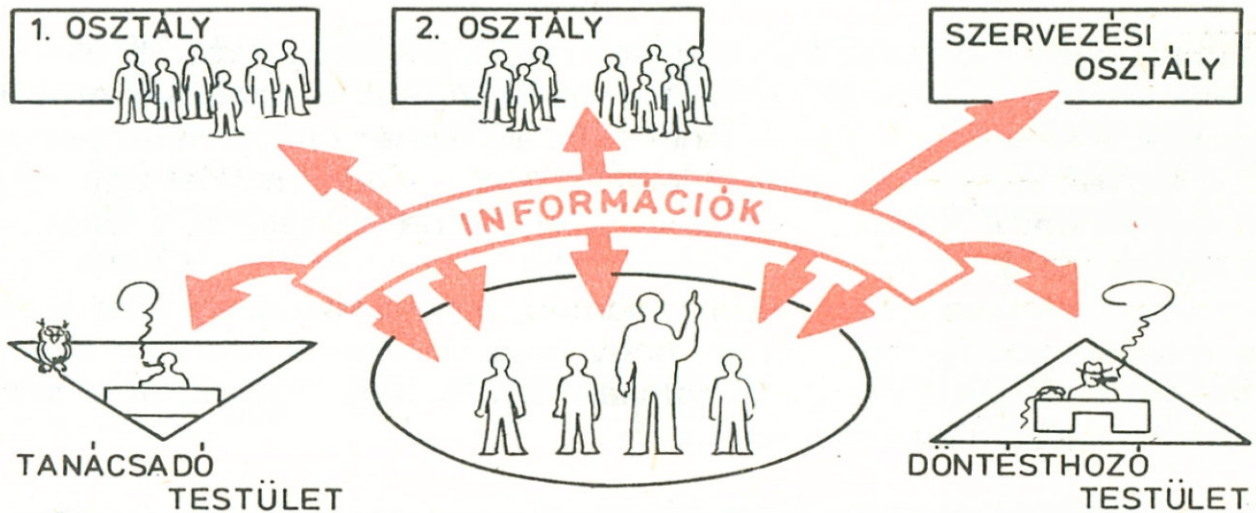
Ezután ötleteket, javaslatokat és modellmegoldásokat gyűjtenek össze, és ezeket az ún. **ideális elképzelésben** rögzítik.

Egyidejűleg pontosan megvizsgálják a régi folyamatokat: **helyzetfelmérést** hajtanak végre. Ebben pl. elemzik, milyen egyedi lépésekben megy végbe a jelenlegi munkafolyamat, mely adatok mozognak, milyen segédeszközöket (irodagépek, bizonylatok stb.) használnak, mely személyek vesznek részt a munkában, milyenek a térbeli viszonyok.

A helyzetfelmérés kiértékelt eredményeiből (helyzetelemzés) és az ideális elképzelésre vonatkozó javaslatokból alakítják ki a vázlatos **szakmai rendszertervet** — a **specifikációkat** —, amelyeket az első tervezési szakasz végén a döntést hozó illetékesek elé terjesztenek.



Miután minden új eljárás bevezetése szükségszerűen a korábbi munkafolyamatok megváltozását jelenti, az érintett szervezeti egységeket és személyeket már ebben a szakaszban meg kell hallgatni, illetőleg tájékoztatni kell. Csak így lehet az ellenállást leküzdeni, és az alkotó együttműködés azon előfeltételeit megteremteni, amelyekre a fejlesztők már a helyzetfelmérés során rá vannak utalva.



Már ebbe az első tervezési szakaszba sok fáradságot és igyekezetet kell befektetni, mivel *minél gondosabban dolgozzuk ki egy eljárás szakmai rendszertervét, annál könnyebb lesz a számítástechnikai megoldást is megtalálni.*

A II. tervezési szakasz



A II. tervezési szakaszban részletezni kell a vázlatos **szakmai rendszertervet**, és ebből létrehozni a részletes szakmai rendszertervet. Ez többek között a következő kérdésekre ad választ:

- Melyek az egyes végrehajtandó feladatok?
- Milyen sorrendben kell ezeket végrehajtani?
- Mely tevékenységeket kell majd kézzel és melyeket géppel elvégezni?
- Mely adatokat és milyen mennyiségben kell majd rögzíteni?
- Mely adatokat kell tárolni?
- Hol, mikor és milyen formában kell az eredményeket szolgáltatni?
- Milyen adatok esnek adatvédelem alá, és milyen biztonsági intézkedésekről kell gondoskodni?

Miután a részletes szakmai rendszerterv elkészült, meg lehet tervezni az eljárást számítástechnikai szempontból is. Elkészül a **vázlatos számítástechnikai rendszerterv**, amely már tartalmazza a számítástechnikai megoldás alapelveit, és a tervezett szoftver vázlatos felépítését is.

A vázlatos számítástechnikai rendszertervben már — többek között — a következőket kell meghatározni:

- az adat-folyamatábrát,
- a tervezett hardverkonfigurációt (a szükséges berendezések típusát és darabszámát, a központi egység fajtáját és operatív-tár-kapacitását stb.),
- az operációs rendszert és az üzemmódot (pl. kötegelt üzemmód, valós idejű üzemmód),
- az egész projekt tagolását egyes építőelemekre,
- a már létező szoftver alkalmazásának lehetőségeit,
- a szoftverfejlesztésnél alkalmazandó módszereket és segédeszközöket (pl. strukturált programozás, előírt programozási nyelv).

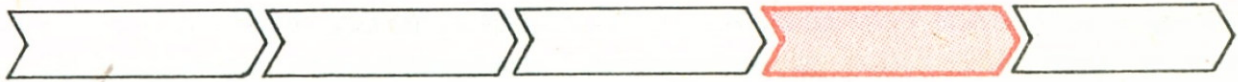
A két rendszerterv — a részletes szakmai rendszerterv és a vázlatos számítástechnikai rendszerterv — együtt adják a **szolgáltatások leírását**. Ezt az érintett egységekkel egyeztetni és teljességüket ellenőrizni kell!



Ezenkívül ebben a szakaszban kell megkezdeni a következő szakaszban meginduló programozáshoz a programozócsoportok összeállítását. Gondolni kell ezenkívül a programok teszteléséhez szükséges gépidőre és az eljárás bevezetésében részt vevő személyek kiképzésére.

A II. tervezési szakasz vége fontos mérföldkő az egész projektben. A rendszerterv e szakaszában levő hibák későbbi kijavítása nagyon sokba kerülhet! *Minden esetben kifizetődik, ha a tervezési szakaszokban jól képzett emberek vesznek részt!*

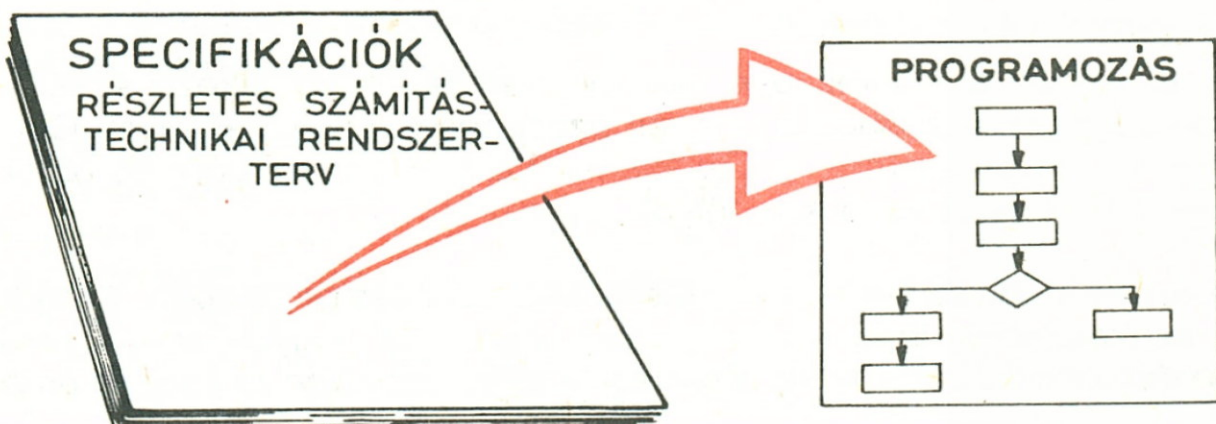
Az I. megvalósítási szakasz



Az I. megvalósítási szakaszban a szolgáltatások leírása alapján a **részletes számítástechnikai rendszertervet** dolgozzák ki. Ez a programozók munkájának alapja és **programspecifikációknak** is nevezik.

A részletes számítástechnikai rendszertervben részleteiben és igen pontosan megadnak minden olyan információt, ami már a vázlatos számítástechnikai rendszertervben is szerepelt. Ezek lényegében a következők:

- Az ajánlás pontos leírása.
- Az eljárás egyes alkotóelemeinek, moduljainak, programjainak leírása.
- Annak meghatározása, milyen lesz az eljárás vezérlése.
- Az eljárás korlátai és lehetséges bővítése.
- Az egész eljárásra érvényes általános szabványok és előírások.
- Az alkalmazandó műszaki berendezések pontos leírása (hardverkonfiguráció).
- Adatok a tárolási módokról és az alkalmazandó állományok felépítéséről, ill. tartalmáról.
- A már meglévő, az eljárásba beépítendő szoftverelemek leírása.
- A programfejlesztésben alkalmazandó szoftvertechnikák (pl. strukturált programozás, moduláris programozás) leírása.
- Az alkalmazandó programnyelvek meghatározása.
- Az egész eljárás tesztelési stratégiájának leírása.
- Az eljárás üzemeltetési módjának meghatározása.
- A felhasznált operációs rendszer megadása.
- Az eljárás felhasználói leírása.



A programozást csak akkor lehet elkezdni, ha a részletes számítástechnikai rendszerterv már elkészült.

A **programozás** tulajdonképpen nem más, mint a részletes számítástechnikai rendszerterv átültetése számítógépprogramokká. Ezt a következő lépésekben hajtják végre:

- a programstruktúra meghatározása,
- a program folyamatábrájának elkészítése,
- kódolás,
- tesztelés,
- programdokumentálás.

A programozást — mint a számítógépes eljárás fejlesztésének magját — egy későbbi fejezet tárgyalja részletesen.

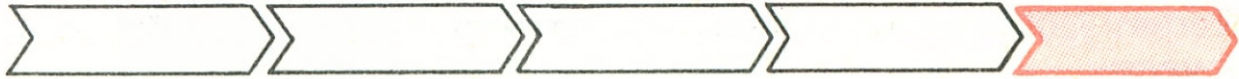
Azért, hogy az érintett szervezeti egységek a szoftvert és annak használatát megismerjék, rendszerint szükséges **tanfolyami oktatást** is tervezni, és ezt egy **oktatói tervben** összeállítani. Ez, többek között, rögzítse azt, hogy mit kell oktatni, kinek kell az oktatást végeznie, és milyen eszközök állnak ehhez rendelkezésre.



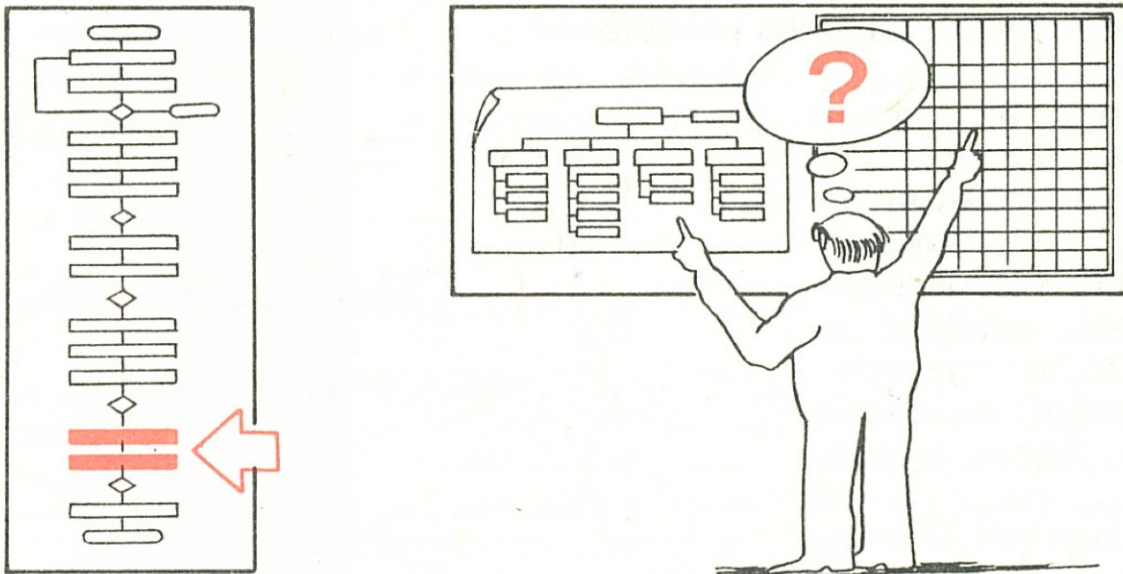
Csak ha valamennyi program tesztelése befejeződött, és kész a teljes dokumentáció, akkor járul hozzá a döntést hozó testület ahhoz, hogy a II. kivitelezési szakasz elkezdődjék. Ebben az új eljárást már a későbbi alkalmazás helyén próbálják ki.



A II. megvalósítási szakasz



Egy új eljárás bevezetéséhez rendszerint egész sor változtatásra van szükség a szervezeti felépítésben és a munkafolyamatokban egyaránt, a részletes szakmai rendszertervnek megfelelően. Ezért ezt a szakaszt a szükségessé váló **szervezeti módosításokkal, illesztésekkel** kell elkezdni, amelynek során az érintett részlegeknél a **feladatokat, felelősségeket és illetékeségeket** részben újra meg kell határozni.



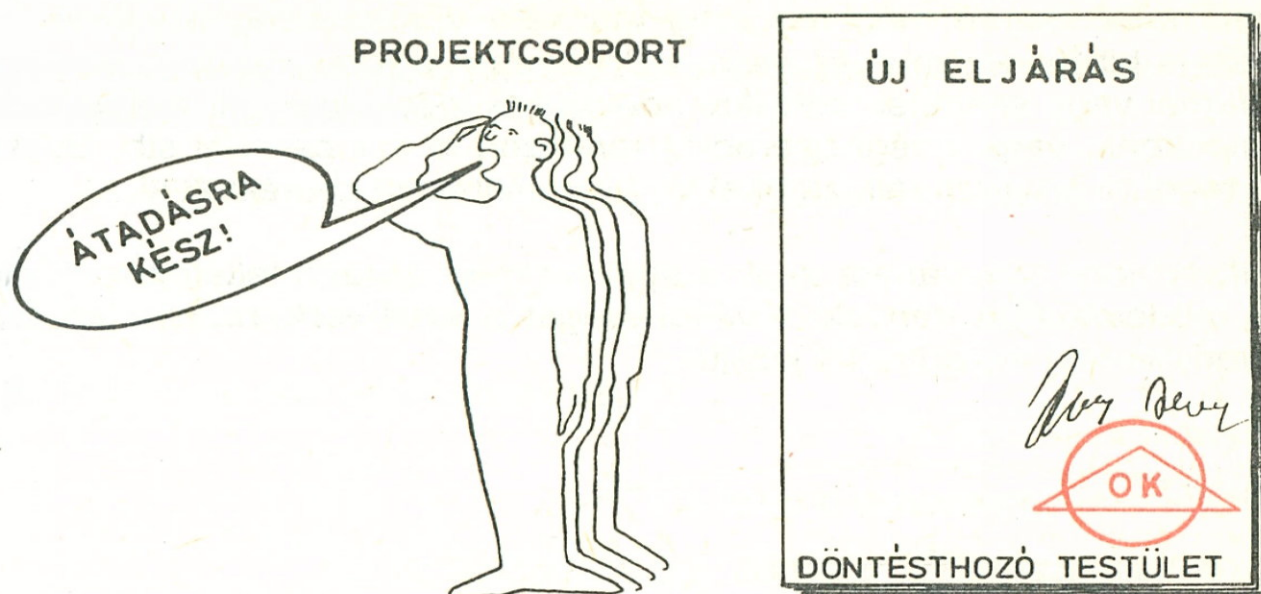
A következő lényeges lépés a **próbaüzem előkészítése**, amelynek során többek között elkészítik az új bizonylatokat, rendelkezésre bocsátják a megfelelő adatokat, és gondoskodnak a szakmai és számítástechnikai személyzet kiképzéséről.

Ha minden előkészület megtörtént, elkezdődhet a **próbaüzem**. A jövődő felhasználónak először van lehetősége az új rendszerrel dolgozni és alkalmazásának célszerűségéről meggyőződni.

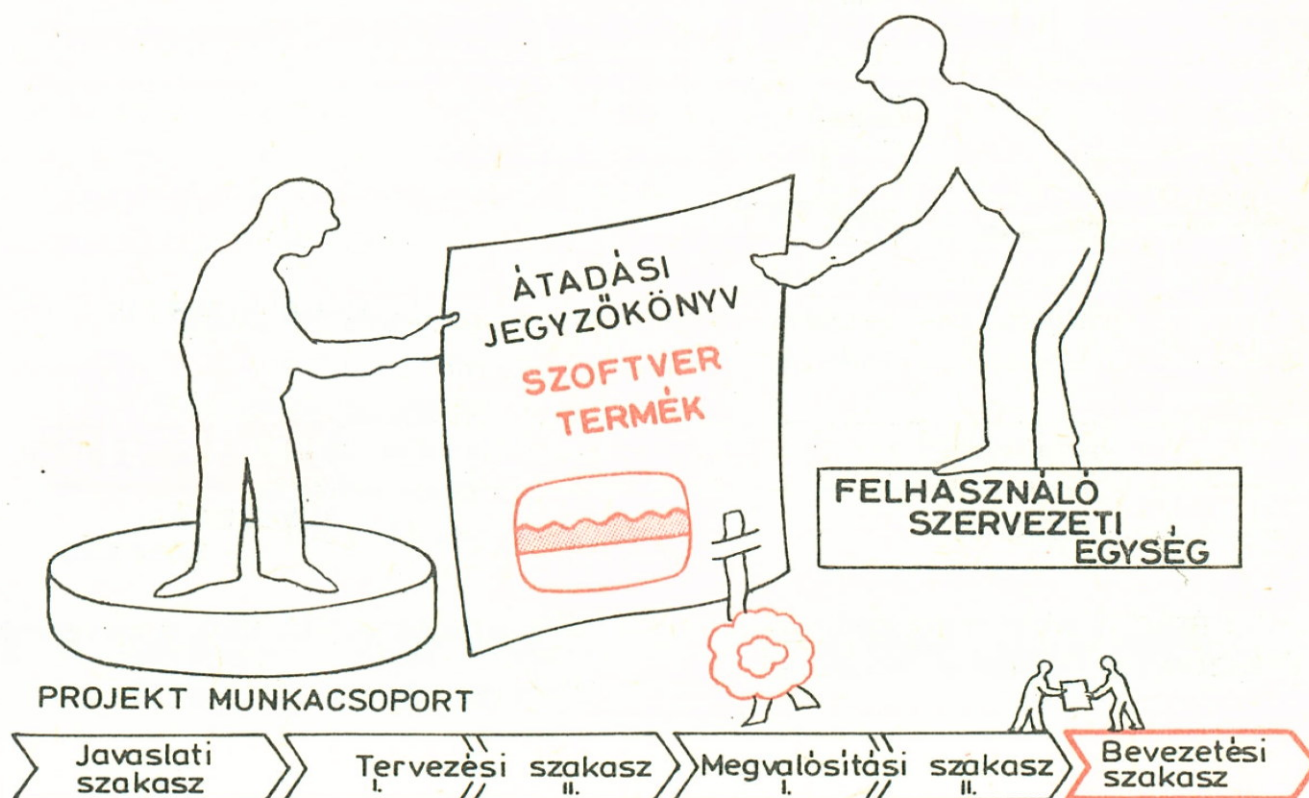
A próbaüzemet biztonsági okokból rendszerint az **eddiggi eljárással párhuzamosan folytatják le**. Ezáltal el lehet érni, hogy hiba esetén az érintett egységek gazdasági tevékenysége zavartalanul folyjon.

A próbaüzem során értesülhetünk az eljárás viselkedéséről éles üzemben, és a felhasználó ismereti szintjéről, valamint az eljárás és kezelésének biztonságáról.

A próbaüzem eredményeit **zárójelentésben** foglalják össze. A döntést hozó testület ennek alapján ellenőrizheti, hogy a próbaüzemi eredmények indokolják-e az eljárás üzemi bevezetését.

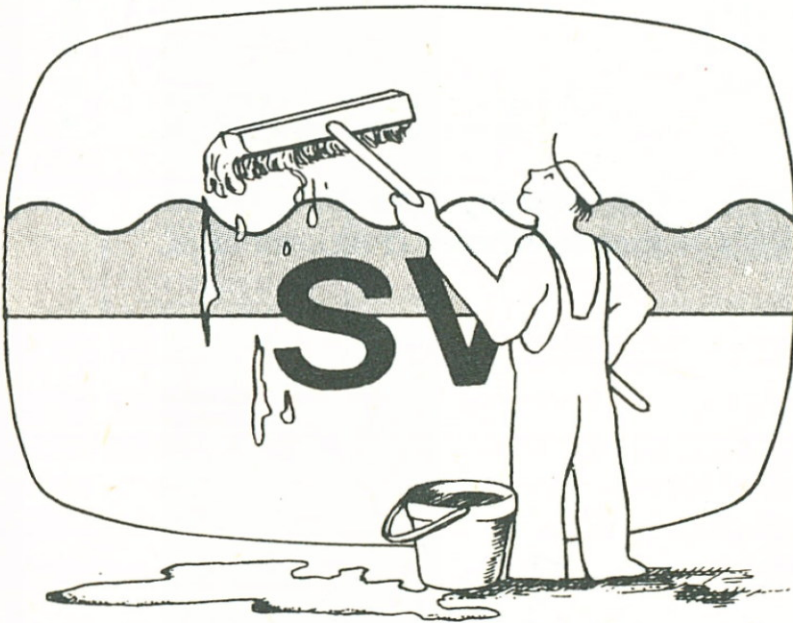


Ha a zárójelentést elfogadják, az eljárást szabályosan át lehet adni a megbízónak. A szabályos átadást **átadási jegyzőkönyv** dokumentálja.



Az eljárás átadásával és bevezetésével véget ér az öt szakaszban ismertetett fejlesztési folyamat. De ez nem jelenti azt, hogy a projekt valamennyi munkája véget ért. Most következik a projekt **bevezetési szakasza**, amelynek során a tapasztalat szerint még rengeteg munkára van szükség. Gyakran még ilyenkor is hibák derülnek ki, vagy változtatásokra lesz szükség, amelyeket új szakmai vagy számítástechnikai adottságok indokolnak, pl. megváltoznak a kamatlábak, vagy a régi operációs rendszert le kell cserélni stb. Gyakran teljes régebbi programváltozatokat új, jobb változatra cserélnek ki.

A szoftverhibák kijavítására vagy a szoftver megváltozott feltételeknek megfelelő módosítására irányuló tevékenységet a **szoftverkarbantartás** (vagy szoftverkövetés) fogalommal jelöljük.



KARBANTARTÁS



BEVEZETÉSI
SZAKASZ

Még az átadás előtt meg kell nevezni azokat a személyeket, akik a bevezetési szakaszban a követést átveszik.



1. Milyen indokok szólnak a szoftverfejlesztés folyamatának szakaszokra való felosztása mellett?

.....

.....

2. Mikor van egy szakasznak vége?

Ha kész a munka.

Ha kész a dokumentáció, vagy

Ha az illetékesek úgy döntenek, hogy a projektet a következő szakasszal folytatni lehet.

3. Mely szakaszokban kell a megadott tevékenységeket elvégezni, ill. dokumentumokat kidolgozni?

Végezze el az egymáshoz rendelést úgy, hogy írja be a megfelelő számokat!

Javaslati szakasz	= 1	I. megvalósítási szakasz	= 4
I. tervezési szakasz	= 2	II. megvalósítási szakasz	= 5
II. tervezési szakasz	= 3		

Projektjavaslat.

Vázlatos számítás-
technikai rendszerterv.

Vázlatos szakmai
rendszerterv.

Részletes számítás-
technikai rendszerterv.

Részletes szakmai
rendszerterv.

Próbaüzem.

Számítógép-
programok.

Az érintettek
kiképzése.

4. A részletes szakmai rendszerterv és a vázlatos számítástechnikai rendszerterv együtt képezik a

.....

5. A programozás minek alapján történik?

.....



Válaszok

1. A következő indokok szólnak a szoftverfejlesztés folyamatának szakaszokra való felosztása mellett:
**A munka előrehaladása ellenőrizhető,
 és a hibás fejlesztések kellő időben megakadályozhatók.**

2. Egy szakasz véget ér.

Ha az illetékesek úgy döntenek, hogy a projektet a következő szakasszal folytatni lehet.

3. Az itt még egyszer röviden felsorolt öt fázishoz az alábbi tevékenységek, ill. dokumentumok tartoznak:

Javaslati szakasz = 1

I. tervezési szakasz = 2

II. tervezési szakasz = 3

I. megvalósítási szakasz = 4

II. megvalósítási szakasz = 5

1

Projektjavaslat

3

Vázlatos számítás-
technikai rendszerterv.

2

Vázlatos szakmai
rendszerterv.

4

Részletes számítás-
technikai rendszerterv.

3

Részletes szakmai
rendszerterv.

5

Próbaüzem.

4

Számítógépprogramok.

5

Kiképzés.

4. A részletes szakmai rendszerterv és a vázlatos számítástechnikai rendszerterv együtt alkotják a **szolgáltatások** leírását.

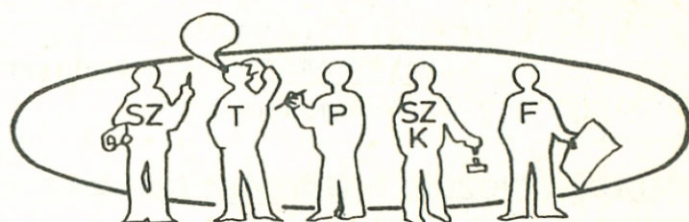
5. A programozás a **programspecifikációk** alapján történik.

Ki fejlessze a szoftvert?

Azok a feladatok, amelyek a szoftverfejlesztés során előfordulnak, igen sokrétűek, és eltekintve az egészen kis méretű projektektől, egy személy már egyáltalán nem tudja azokat áttekinteni.

Számítógépes eljárások fejlesztését ésszerű módon tulajdonképpen csak **projekt munkacsoportokban** lehet végrehajtani, amelyek tagjai különösen a szervezés és a számítástechnika területén jártasak. A projekt munkacsoportok tagjainak tehát a következőkről legyenek ismeretei és tapasztalatai:

- a mindenkori szakterületről,
- az általános szervezésről,
- az eljárások fejlesztéséről,
- a hardverről,
- a szoftverfejlesztésről,
- a programozásról.



SZ - Szervezés
 T - Tervezés
 P - Programozás
 SZK - Számítóközpont
 F - Felhasználás

A felsoroltakon kívül minden projekt munkacsoporthoz tartozzék legalább egy számítástechnikai **összekötő munkatárs**, aki annak a felhasználónak a képviselője, akinek számára a szóban forgó eljárás készül.

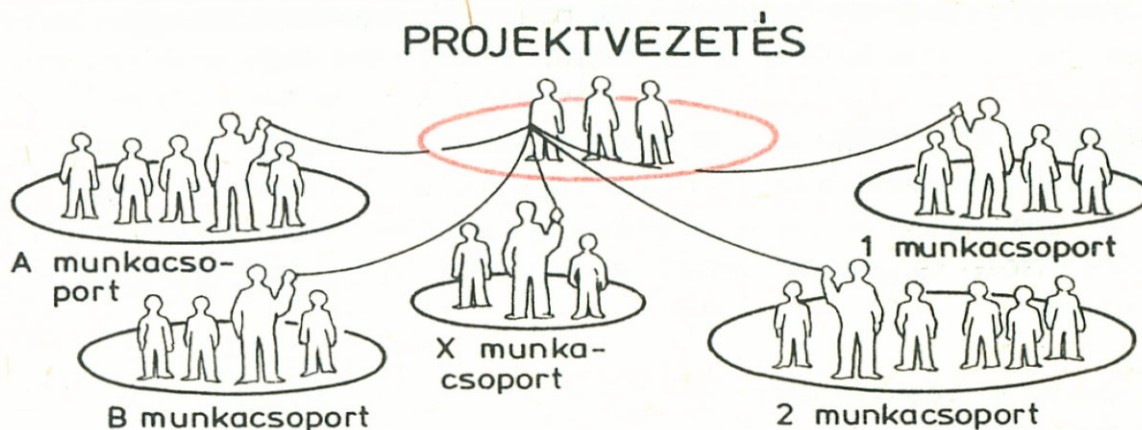
A projekt munkacsoportokhoz gyakran egy **tanácsadó bizottságot** is hozzárendelnek, amelyet elsősorban a felhasználó érintett osztályainak szakembereiből állítanak össze, akik a munkacsoportoknak esetenként szakmai tanácsokat adnak.

Ezeknek a munkacsoportoknak további fontos partnere a **számítóközpont felelős személyzete**, aki az eljáráshoz szükséges hardverteendőket elvégzi.

Az, hogy egy ilyen munkacsoport hány személyből álljon, általában előre nem mondható meg. Ez függ attól, hogy mekkora a projekt, és hogy a munkacsoport egy-egy tagjának milyen széles körű speciális ismeretei vannak.

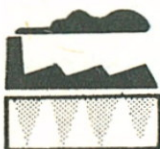
A munkacsoportok nem merevek, összetételük változik a projekt előrehaladása folyamán. Különösen azokban a munkaszakaszokban, amelyekben a programozás elkezdődik, a munkacsoportot ki kell bővíteni. Ilyenkor rendszerint több programozócsoportot alakítanak és ezek között felosztják a munkát. Elvileg azonban mindig valamennyi specialista készen áll a tanácsadásra úgy, hogy valamennyi döntésben valamennyiük ismeretét fel lehessen használni.

Ezeket a munkacsoportokat a **projektvezetés** hozza létre és koordinálja is munkájukat. A projektvezetés önmagában is több személyből állhat, és munkacsoportonként működhet.



A projektvezető a **döntést hozó illetékesekkel** szemben tartozik felelősséggel, amely többek között a megbízó képviselőit is magába foglalja. Ellenőrzi a munka előremenetelét, és eldöntik az egyes munkaszakaszok végén, hogy lehet-e folytatni a projektet, és ha igen, milyen eszközök segítségével és milyen módon.

Az említett személyek mellett a kifejlesztendő számítógépes eljárással kapcsolatban további szervezetek is foglalkoznak. Ezek pl. az üzemi tanács, az adatvédelmi megbízott és az ellenőrzési osztály.



Az **üzemi tanácsnak** időben értesülnie kell az új munkafolyamatok és új munkahelyek tervezéséről, tekintettel arra, hogy az üzemi tanács képviseli az új eljárás által érintett dolgozók érdekeit.



Az **adatvédelmi megbízottnak** kell véleményeznie, hogy a tervezett megoldás törvényesen megengedhető-e, az érintetteknek milyen kötelezettségei és milyen jogai vannak.



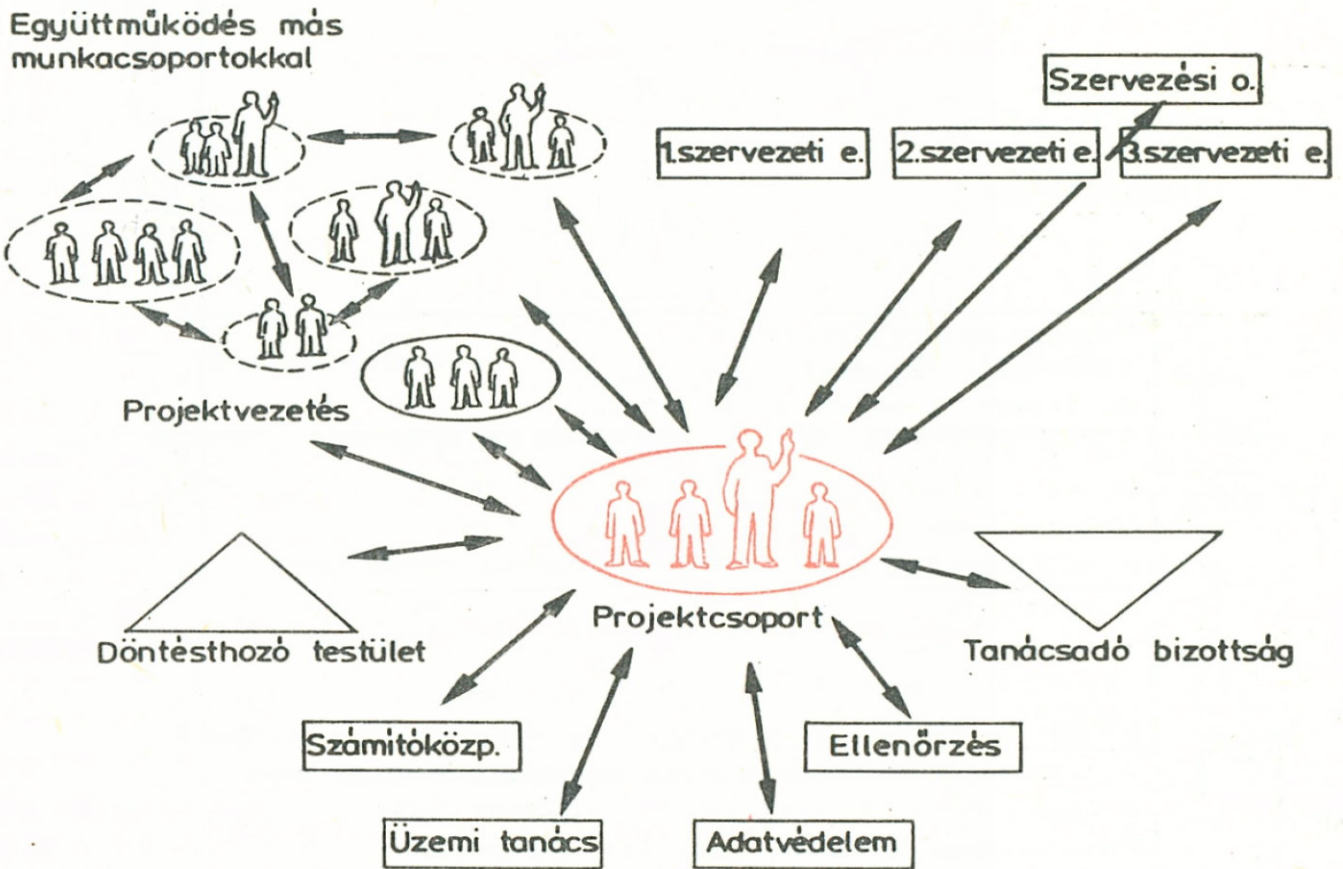
Mint semleges szervezet, az **ellenőrzési osztály** képviselői vizsgálják meg az új eljárást abból a szempontból, hogy az megfelel-e a gazdaságosság elveinek és figyelembe veszi-e a hatályos rendeleteket, valamint a biztonsági előírásokat. Tehát az ellenőrzési osztálynak időben értesülnie kell a kifejlesztendő új számítógépes eljárásról!

A munka felosztásának szempontjából még egyszer tekintsük át az egyes munkaszakaszokban elvégzendő tevékenységeket! Ennek során eldöntjük, milyen szakemberek vesznek részt az egyes tevékenységekben mint végrehajtók (**v**), mint közreműködők (**k**), vagy csupán mint tanácsadók (**t**).

		Felhasználó	Szervezési szakemberek	Számítástechnikai szakemberek	Programozók	Számítóközpont személyzete
JAVASLATI SZAKASZ	Feladatmegfogalmazás	v	v	v		
	Előzetes vizsgálat végrehajtása	k	v	k		
	A projektjavaslat elkészítése		v			
I. TERVEZÉSI SZAKASZ	Az ideális elképzelés elkészítése	k	v	t		
	A helyzetfelmérés végrehajtása	k	v			
	Helyzetelemzés	k	v	t		
	A vázlatos szakmai rendszerterv elkészítése	k	v			
II. TERVEZÉSI SZAKASZ	A részletes szakmai rendszerterv elkészítése	k	v	k		
	A vázlatos számítástechnikai rendszerterv elkészítése		k	v		t
	Tesztidők tervezése			v		k
	Oktatási terv	k	v	v		t
I. MEGVALÓSÍTÁSI SZAKASZ	A programozási és tesztelési megállapodások rögzítése			v		k
	A részletes számítástechnikai rendszerterv elkészítése			v		k
	Programozás, kódolás, szoftver illesztése			t	v	
	A teszt végrehajtása			v	k	k
	A programdokumentáció elkészítése			v	v	
	Kezelési utasítás elkészítése			v	v	t
	Előírások összeállítása	t	v	v		
II. MEGVALÓSÍTÁSI SZAKASZ	A szakmai és számítástechnikai személyzet kiképzése		v	t		
	A szervezeti illesztés	k	v	v		
	Próbaüzem előkészítése		v	v		
	A programkövetés módjának meghatározása		v			
	Próbaüzem	v				v
	Az eljárás átadása		v	v		
BEVEZETÉS	Az eljárás alkalmazása	v	t	t		v

A legkülönbözőbb szervezési feladatokat végző szakembereket ebben a táblázatban egyszerűen szervezőknek jelöltük. Ezek közé értjük azokat a munkatársakat is, akik szakmailag tájékozottak a kifejlesztendő eljárás vonatkozásában. A számítástechnikusok rovatába vettük föl az olyan hardver- és szoftverszakembereket, akik magukat rendszerelemzőnek, vezető rendszertervezőnek vagy programozónak nevezik.

Abban az esetben, ha a munkacsoportok valamennyi tagja, valamint az összes részt vevő bizottság a szoftvertechnika elveinek betartásával koordináltan együttműködik, minden föltételt biztosítottunk ahhoz, hogy jó szoftvert fejlesszünk ki, és az új rendszer bevezetése súrlódásmentesen történjék.



Ki fejlessze a szoftvert?

Kérdések az 53...56. oldalakhoz



1. Milyen szakismeretekkel rendelkező munkatársakból kell össze-
tevédnie annak a munkacsoportnak, amely egy új számítógépes
eljárás tervezésével, kifejlesztésével és bevezetésével van meg-
bízva? Jelölje meg a megfelelő ismereteket!

- Programozói ismeretek.
- A hardver ismerete.
- Általános szervezési ismeretek.
- Az érintett szakterület ismerete.
- Szoftvertervezési ismeretek.
- Folyamatszervezési ismeretek.
- Az adattárolás és adatátvitel ismerete.
- A felhasználó igényeinek ismerete.

2. Ki koordinálja a projektmunkacsoportot?

.....

3. Az eljárás fejlesztésénél az alábbi szakemberek és bizottságok
működnek közre:

Projektvezetés (1)	Számítástechnikusok (4)
Döntést hozó testület (2)	Programozók (5)
Szervezési szakemberek (3)	Felhasználók (6)

Ki hajtja végre az alábbi tevékenységeket?

Írja be a megfelelő számokat!

- A projektjavaslat kidolgozását.
- Az ideális elképzelés kidolgozását.
- A vázlatos szakmai rendszerterv kidolgozását.
- A részletes szakmai rendszerterv kidolgozását.
- A vázlatos számítástechnikai rendszerterv kidolgozását.
- A részletes számítástechnikai rendszerterv kidolgozását.
- Programozást, kódolást, a már meglévő szoftver illesztését.
- Valamennyi munka ellenőrzését és koordinálását.
- Annak eldöntését, hogy a munkaszakasz lezárása után
folytatható-e a projekt és hogyan.



Válaszok

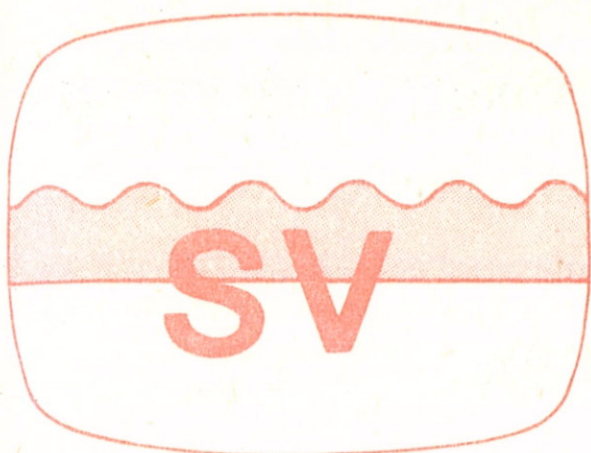
1. Valamennyi felsorolt szakmai ismeretre szükség van!

- Programozói ismeretek.
- A hardver ismerete.
- Általános szervezési ismeretek.
- Az érintett szakterület ismerete.
- Szoftvertervezési ismeretek.
- Folyamatszervezési ismeretek.
- Az adattárolás és adatátvitel ismerete.
- A felhasználói igények ismerete.

2. A projektmunkacsoportot a **projektvezetés** koordinálja.

3. A megnevezett tevékenységeket az alábbi szakemberek és bizottságok hajtják végre:

- 1 A projektjavaslat kidolgozását.
- 3, 4 Az ideális elképzelés kidolgozását.
- 3 A vázlatos szakmai rendszerterv kidolgozását.
- 3 A részletes szakmai rendszerterv kidolgozását.
- 4 A vázlatos számítástechnikai rendszerterv kidolgozását.
- 4 A részletes számítástechnikai rendszerterv kidolgozását.
- 5 Programozást, kódolást, a már meglévő szoftver illesztését.
- 1 Valamennyi munka ellenőrzését és koordinálását.
- 2 Annak eldöntését, hogy a munkaszakasz lezárása után folytatható-e a projekt és hogyan.



Mi a szoftver?

Hogyan fejlesztik a szoftvert?

Mi segítheti a szoftvertervezést és -fejlesztést?

Mit csinál a programozó?

Milyen szolgáltatásai vannak az operációs rendszernek?

Ebben a fejezetben megismerünk számos olyan módszert és eszközt, amely a szoftverfejlesztőt segíti a szoftver megtervezésében, kifejlesztésében, elkészítésében, ellenőrzésében és dokumentálásában.

A projekttervezés módszerei

Tervezési módszerek

Ellenőrzési módszerek

Automatizált segédeszközök

Programnyelvek

Mindenhol, ahol feladatokat kell megoldani, akár tudományos, gazdasági vagy műszaki területeken, olyan segédeszközöket dolgoztak ki, amelyek a feladat elemzésében, a megoldások megtervezésében és ezek megvalósításában segítséget nyújtanak.

A szoftverfejlesztéshez is készültek ilyen segédeszközök.

A szoftver tervezéséhez és fejlesztéséhez rendelkezésre álló segédeszközök spektruma igen széles. Ide tartoznak az olyan hagyományos segédeszközök, mint az adat-folyamatábra és a program-folyamatábra, ide tartoznak a korszerű szoftvertechnológia **módszerei** és **segédeszközei** is.

A modern szoftvertechnológia módszereinek egy része egészen más szakterületekről származik, ezeket a számítástechnika igényeinek megfelelően csupán átalakították.

Módszerek a szoftverfejlesztés valamennyi területén léteznek. A következőkben bemutatunk néhány módszert, amelyek a projekt szervezését, a program tervezését és az eredmények ellenőrzését segítik.

A szoftverfejlesztés munkáinak megkönnyítésére arra törekedtek, hogy egyrészt ezeket bizonyos módszerek segítségével — amennyire csak lehetséges — automatizálják, azaz a feladatok egy részét rábízják magára a számítógépre, másrészt valamennyi lehetséges tevékenység segítésére automatizált eszközöket igyekeztek alkotni.

Az ilyen **automatizált segédeszközöket**, amelyeket egyszerűen **eszközöknek** vagy a megfelelő angol kifejezés szerint **toolnak** nevezünk, számítógép-programok, amelyek a szoftverfejlesztő munkáját segítik. Az ilyen eszközökből itt csupán néhány példát tudunk bemutatni.

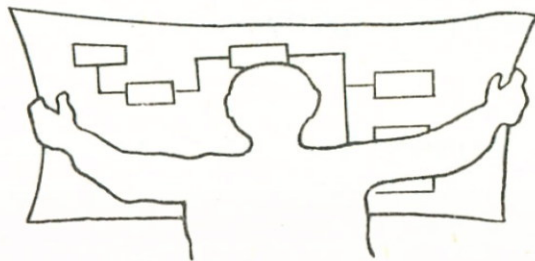


A programfejlesztő munkáját segítő eszközökhöz tartoznak a **programozási nyelvek** is, amelyek közül néhány ma használatos nyelvet röviden ismertetünk.

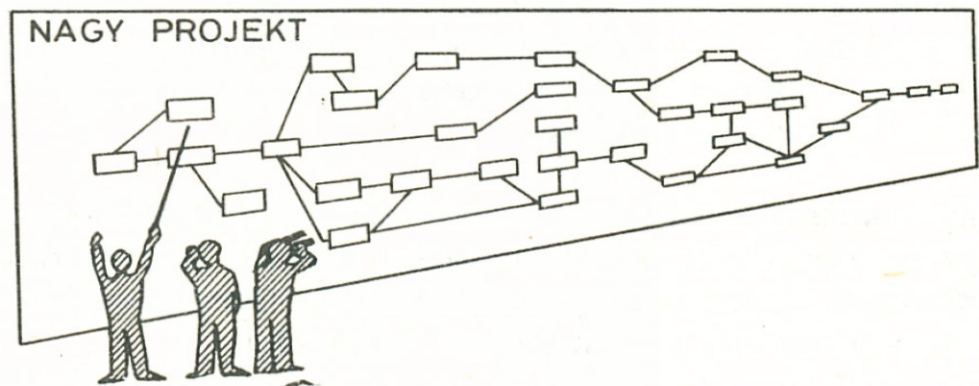
A projekttervezés módszerei

Terjedelmes projektek végrehajtása — függetlenül attól, hogy egy út megépítéséről, vagy egy számítástechnikai eljárás kifejlesztéséről van-e szó — pontos tervezést, koordinálást és ellenőrzést igényel az egyes munkaszakaszokban.

Minél nagyobb egy projekt, annál nehezebb az áttekintése, és annál nehezebb a részt vevő emberek, szervezeti egységek vagy a vállalatok koordinálása.



KIS PROJEKT



NAGY PROJEKT

Annak érdekében, hogy ezen nehézségeket megoldják, olyan módszereket fejlesztettek ki, amelyek révén lehetséges egy projekt *szakmai, időbeli és személyi összefüggéseit* szemléletesen ábrázolni.

Jól bevált módszer — amelyet időközben továbbfejlesztettek — a **hálótervezési eljárás**.

A hálótervezés manapság sok vállalatnál nélkülözhetetlen segédeszköz, és a számítástechnika területén is nélkülözhetetlenné vált.

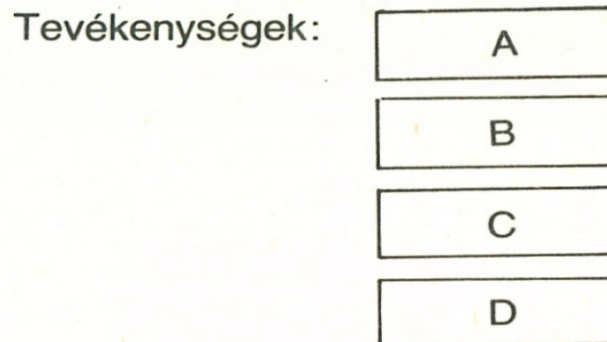
A hálótervezés

A hálós tervezési technika a **hálóterven** alapszik.

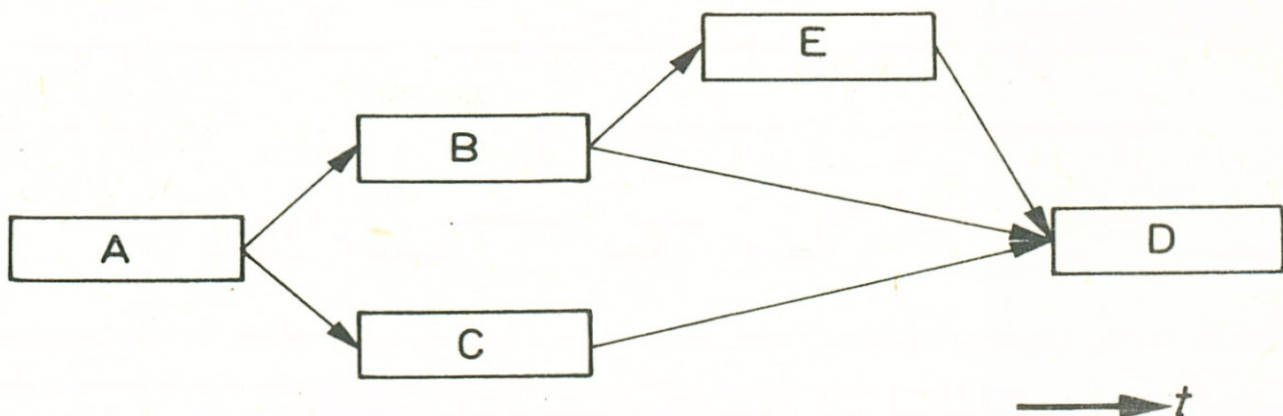
A hálóterv a projekt során végrehajtandó valamennyi tevékenység áttekinthető és jól érthető grafikus ábrázolása, amely figyelembe veszi az *időbeli, szakmai és személyi összefüggéseket* az egyes tevékenységek között.

A hálóterv elkészítésének első lépése a szóba jövő *tevékenységek felsorolása* úgy, hogy a vázlatos háló felrajzolásához az egyes tevékenységeket csoportokba lehessen összefoglalni.

Példa:



A második lépés a *tevékenységek rendezése* az összefüggéseik alapján. Példánkban a B és a C feladatok párhuzamosan futhatnak, de csak akkor, ha az A-nak vége van. A D-nek szüksége van a B, a C és az E eredményeire. Az E tevékenységet lehetne a B és a C tevékenységgel párhuzamosan folytatni, mégis időben el kell csúsztatni, mivel ugyanannak a munkacsoportnak kell csinálnia, mint amelyik a B tevékenységet folytatja.



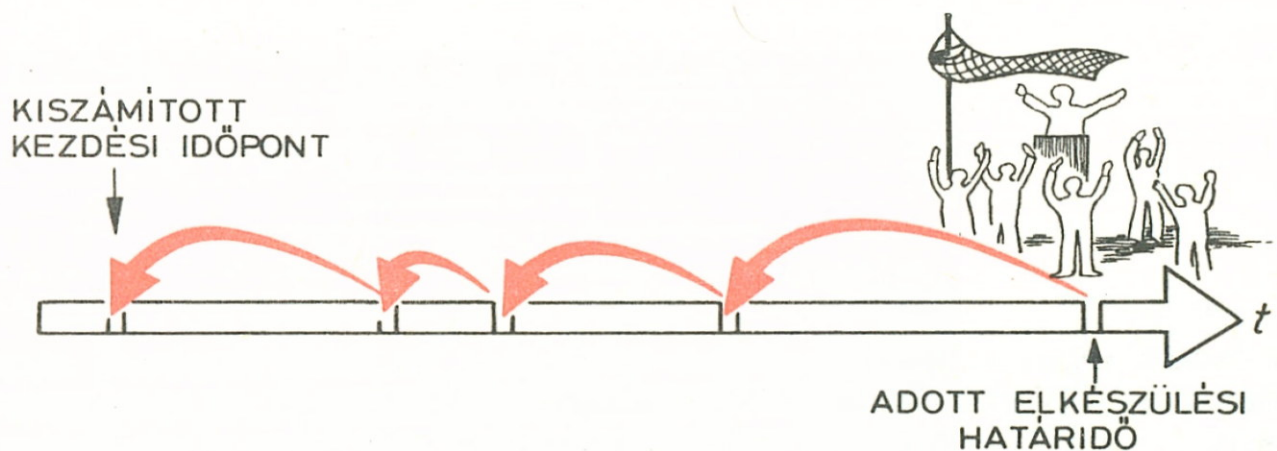
Miután elkészült a **Projektháló**, meg kell határozni a határidőket!

Rendszerint van egy *adott, rögzített határidő*, amelyre az egész projektet be kell fejezni, pl. „egy autópályát június 1-re, a fő utazási szezon megindulására el kell készíteni”, vagy „egy közvetlen helyfoglaló rendszernek október 1-re a menetrendváltás idejére üzembe kell állnia”.

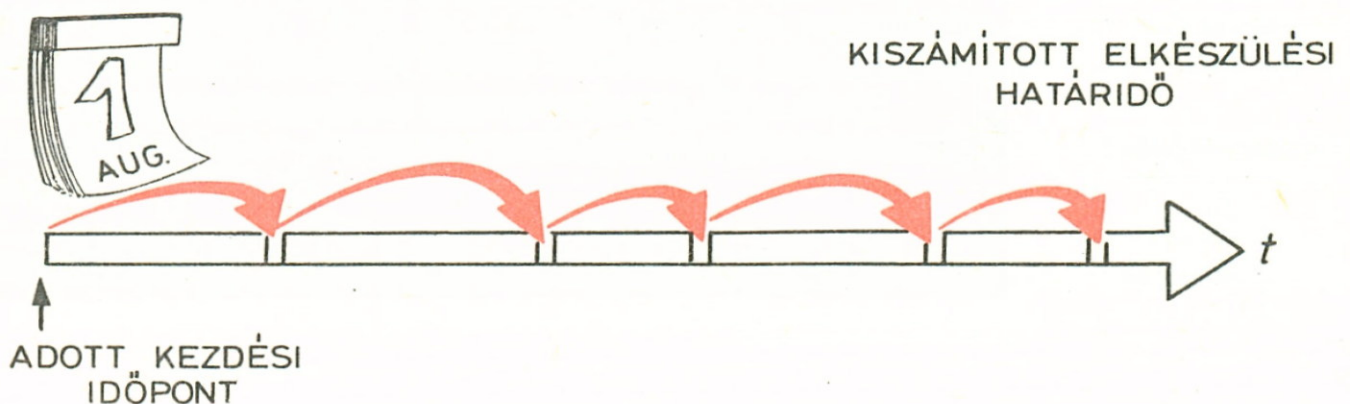
Más esetekben a *kezdési időpont rögzített*, pl. „az A projekt befejezése után, január 8-án azonnal el kell kezdeni a B projektet”.

A mindenkori tevékenységekért felelős szakemberek ezek után **becsléseket** adnak arra az időre, amelyre szerintük az egyes tevékenységekhez szükség van.

Egy előre megadott elkészülési határidő esetén ezeknek a becsléseknek az alapján a véghatáridőből kiindulva kiszámítható a kezdés időpontja.



Más esetben a megadott kezdési határidőből kiindulva kiszámítható a várható véghatáridő.

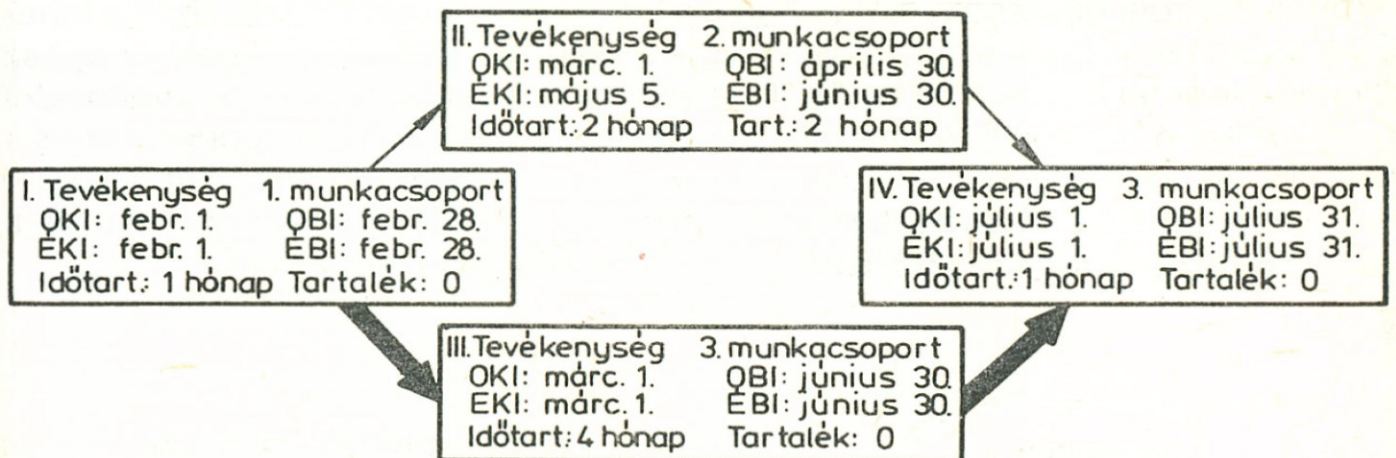


Azt, hogy ezeket a számításokat hogyan hajtják végre, az alábbi kis példán mutatjuk be:

Egy projekt I...IV-ig jelölt tevékenységekből áll. Az I. tevékenységre az azért felelős személy egyhónapos kidolgozási időt ad meg, a II. tevékenységre ez az idő két hónap, a III. tevékenység négy hónapig tart és a IV. tevékenységhez szükséges idő egy hónap. A projektet február 1-én lehet indítani. Az I. tevékenységet le kell zárni, mielőtt a II. és a III. tevékenységet el lehetne kezdeni. A IV. tevékenységnek szüksége van a II. eredményeire. Ezenkívül tudjuk, hogy a IV. tevékenységet ugyanaz a munkacsoport dolgozza ki, mint amelyik a III. tevékenységet végzi, ezért ezt csak a III. befejezése után lehet elkezdni.

Ennyi adatot ismerünk a négy tevékenységről. El tudjuk-e képzelni, hogyan lehet ily módon sok száz tevékenység összefüggését szöveggel leírni?

Bizonyára nem! Ennél sokkal áttekinthetőbb a hálóterv, amely a példabeli négy tevékenységre a következő lesz:



OKI: legkorábbi kezdési időpont.

ÉKI: legkésőbbi kezdési időpont.

OBI: legkorábbi befejezési időpont.

ÉBI: legkésőbbi befejezési időpont.

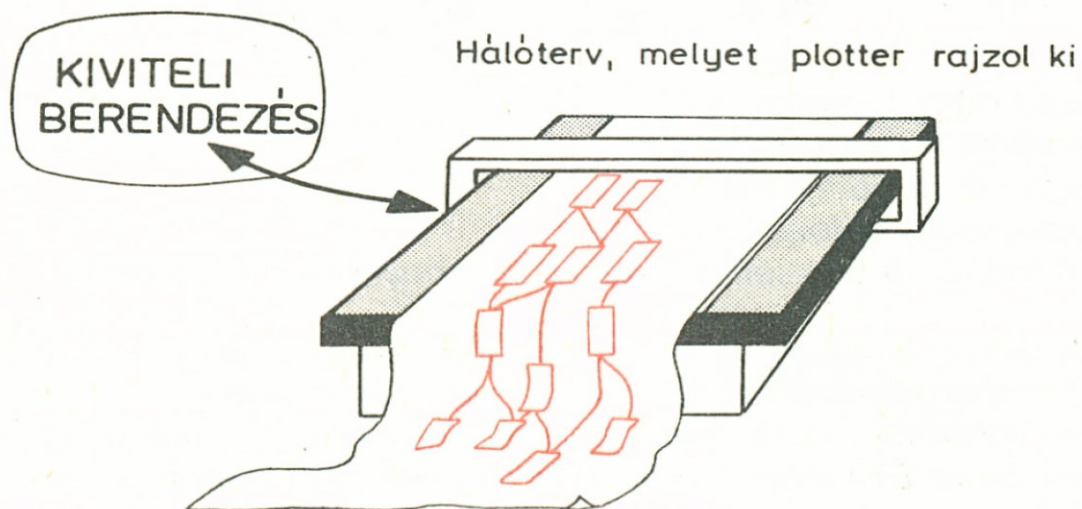
Az időtartam- és a tartalékidő-adatokat hónapokban kell érteni.

A téglalapokba, amelyek az egyes tevékenységeket jelölik, mindig beírjuk a legkorábbi és legkésőbbi kezdési időpontot, valamint a legkorábbi és legkésőbbi befejezési időpontot, ezenkívül a tevékenységhez szükséges időtartamot. Ha az OKI és az ÉKI időpontok nem egyeznek meg, akkor tartalékidő adódik, amely akkora, mint az említett két időpont közötti eltérés. A **tartalékidő** a megfelelő tevékenység végrehajtásakor szükség esetén rendelkezésre áll. Az előző példánál csupán a II. tevékenységnél adódott tartalékidő, azért, mert a II. és a III. tevékenység egymással párhuzamosan végezhető, és a II. tevékenység rövidebb ideig tart.

Azoknak, akiknek a II. tevékenységet kell végrehajtaniuk, lehetőségük van arra, hogy a két hónapig tartó munkát a tartalékidő révén valahol a négyhónapos időtartam közben tetszőlegesen végezzék el. A többi tevékenység esetén ez a lehetőség nincs meg. Ha ezeknél a munkáknál csak egy nappal is késnek, a rá következő tevékenységet nem lehet a tervezett időpontban elindítani, és az egész projekt befejezési határideje veszélybe kerül.

Valamennyi, tartalékidő nélküli tevékenység az ún. kritikus út mentén fekszik. Példánknál a kritikus utat vastag nyilakkal jelöltük. A projektvezetés a hálótervről leolvashatja a **kritikus utat**, és a kritikus út mentén fekvő tevékenységek végrehajtását különös gonddal ellenőrizheti.

A gyakorlatban a hálótérvek segítségével tevékenységek százait ábrázolják. Az időtartam- és tartalékidő-adatokat a projekt méretétől és a hálóterv részletességétől függően hónapokban, hetekben vagy napokban — kivételes esetekben órákban vagy percekben — adják meg. A számítástechnika alkalmazásával a hálótérvezési eljárások széles körben elterjedtek. A számítástechnika lehetővé teszi azt, hogy kiszámítsuk az egyes határidő-elcsúszásoknak az egész projekt menetére gyakorolt hatását, és igen rövid idő alatt több új tervváltozatot hozzunk létre. A projekt menete folyamán ezenkívül mindig az éppen aktuális hálóterv áll rendelkezésre. Egy számítógép segítségével előállított hálóterv eredményeit vagy a **sornyomtatóval** íratjuk ki, vagy egy számítógép által vezérelt rajzgép, az ún. **plotter** segítségével ábrázoljuk.





1. Mit ábrázol egy hálóterv?

Jelölje a helyes válaszokat!

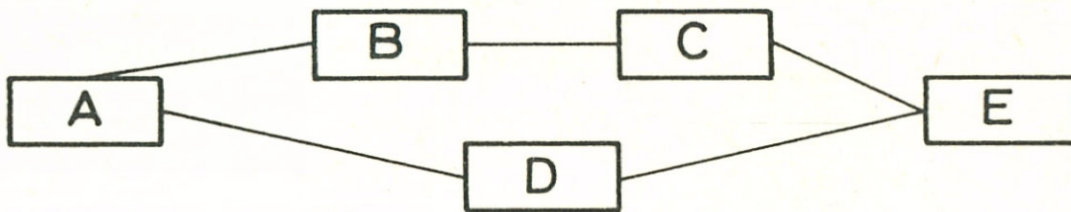
Személyi összefüggéseket.

Időbeli összefüggéseket.

Szakmai összefüggéseket.

2. Az ábra jelképes hálót mutat:

Mikor lehet az E tevékenységet végrehajtani?



3. Honnan kap a hálóterv készítője egy előre megadott véghatáridő esetén adatokat az egyes tevékenységek végrehajtásához szükséges időtartamról?

.....

4. Mi a kritikus út?

A tartalékidő nélküli tevékenységek egymás után következő sora.

Valamennyi késésben levő tevékenység.



Válaszok

1. A hálóterv a következőket ábrázolja:

- Személyi összefüggéseket.
- Időbeli összefüggéseket.
- Szakmai összefüggéseket.

2. Az E tevékenységet akkor lehet elkezdni,
ha a **C és a D tevékenységek véget értek.**

3. A hálóterv kidolgozója az egyes tevékenységek végrehajtásához
szükséges **időtartamról felelős szakemberektől** kap becsléseket.

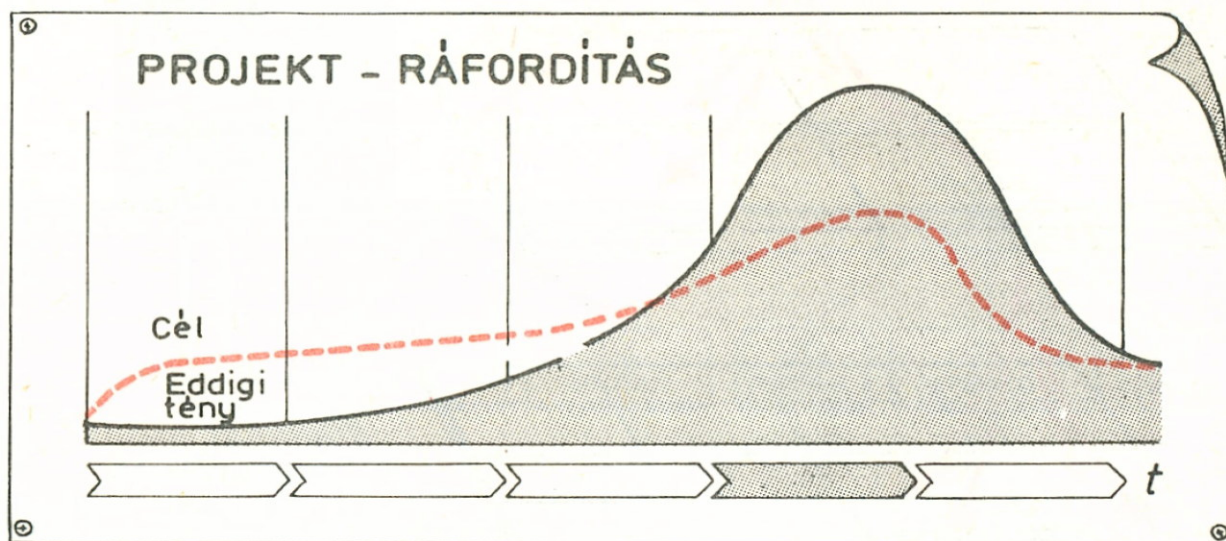
4. Egy hálótervben a kritikus út:

- A tartalékidő nélküli tevékenységek egymás után következő sora.
-

Tervezési módszerek

Már néhány éve tudatosult az a tény, hogy különösen nagy jelentőségű a szoftver tervezési szakasza. Ez azért van így, mert a feladat elemzéséhez és a számítógépes eljárás tervezésének pontos végrehajtásához viszonyítva a programok kódolásához és teszteléséhez viszonylag kevesebb idő szükséges.

Ezért nem meglepő, hogy a szoftverfejlesztés eddig megvalósított módszereinek legnagyobb része a *programtervezés* segítségét tűzte ki célul. Valamennyi ilyen módszer a program tervezésében jelent segítséget, de közülük néhány ezenfelül befolyásolja a *program elkészítését* is, egészen a kódolásig, ill. a tesztelésig.



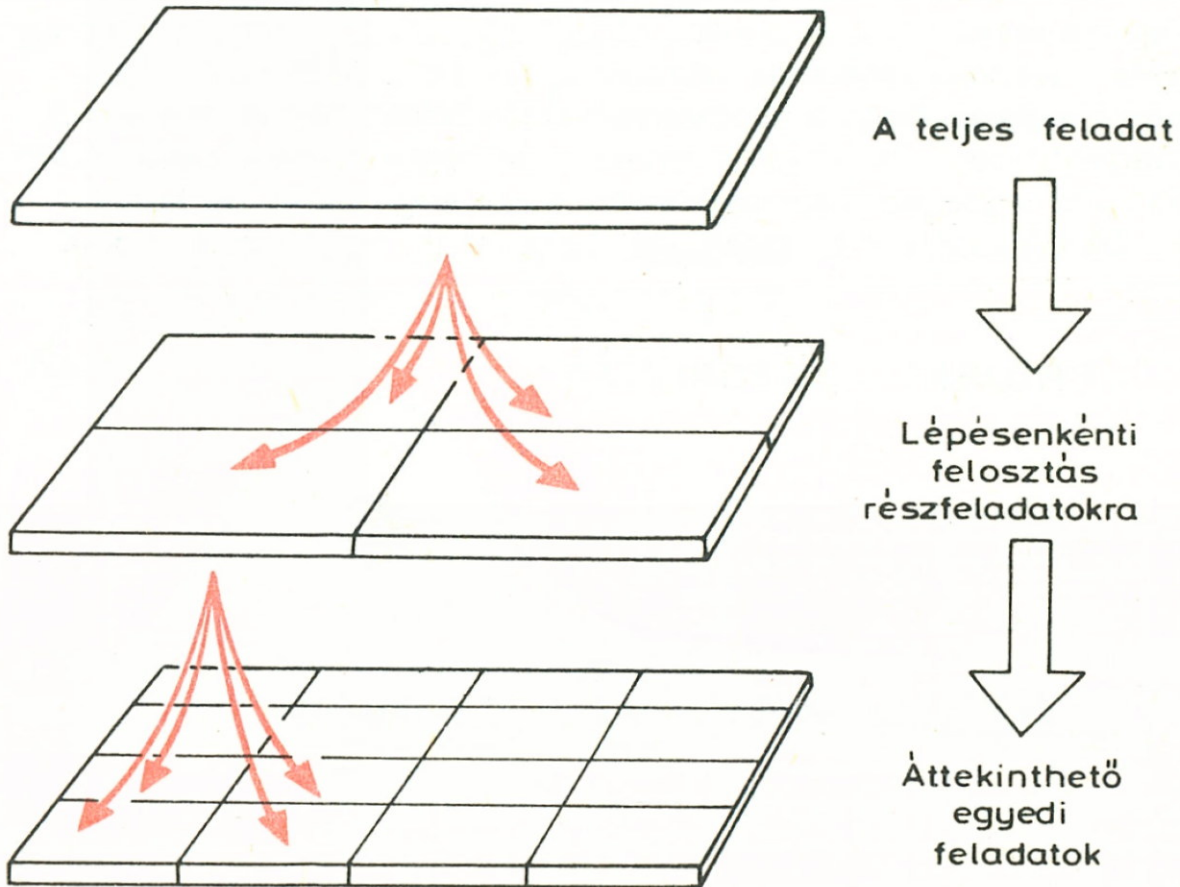
A következőkben e módszerek közül ismertetünk néhányat.

A moduláris programozás

Az elektronikus készülékeknél (pl. a tv-készülékeknél) már huzamosabb ideje használják az ún. **építőköcka-** vagy **modultechnika-**elvet. Ez többek között azért is előnyös, mert meghibásodás esetén csupán komplett modulokat kell kicserélni, vagy további kiegészítő funkciókhoz csupán kiegészítő modulokat kell behelyezni. További előnye, hogy ezt a munkát olyan szerelő is végre tudja hajtani, akinek a berendezés áramköreit egyáltalán nem kell ismernie és megértenie.

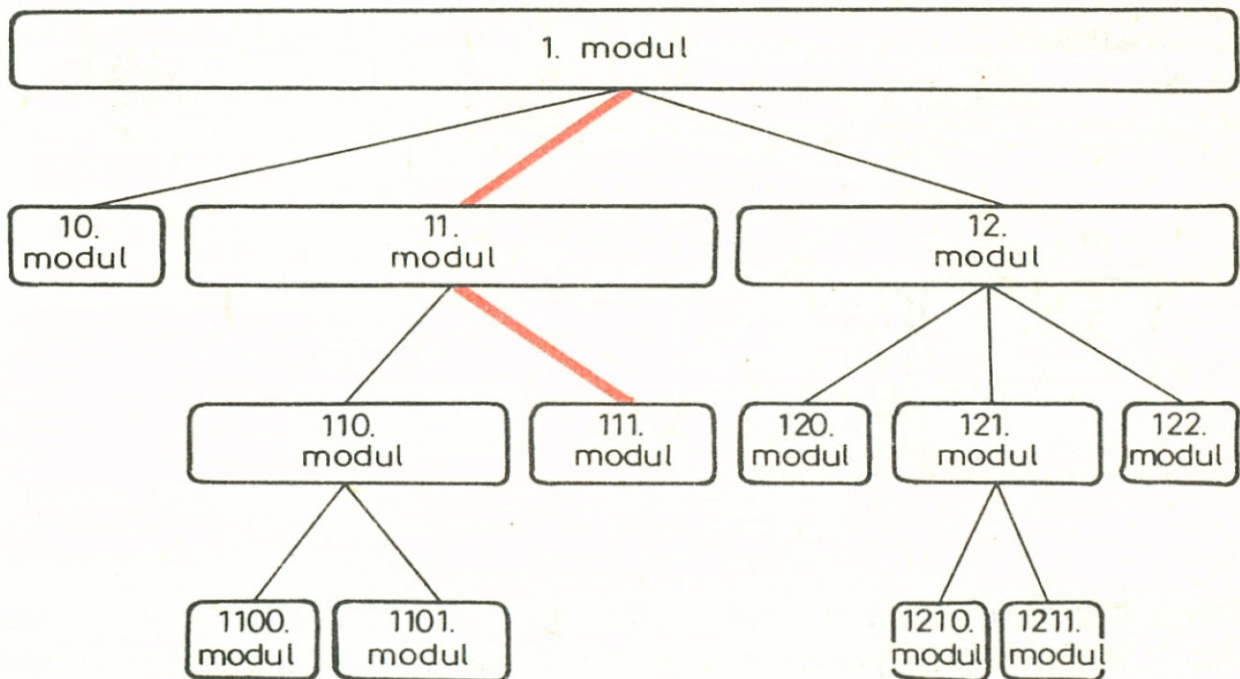
A számítástechnikában a **modultechnika** olyan módszer, amely a számítógépes eljárás fejlesztését segíti. Azért használják, hogy egy komplex feladat, pl. a személyi nyilvántartás számítógépesítése, áttekinthető részfeladatokra legyen felosztható. Ezek után az egyes részfeladatok megoldásához **program-modulokat** lehet tervezni és készíteni.

E módszerek alkalmazásakor a megoldandó feladatokat részfeladatokra osztjuk fel, majd ezeket továbbtagoljuk egészen addig, amíg értelmes, áttekinthető egyedi feladatokat kapunk.

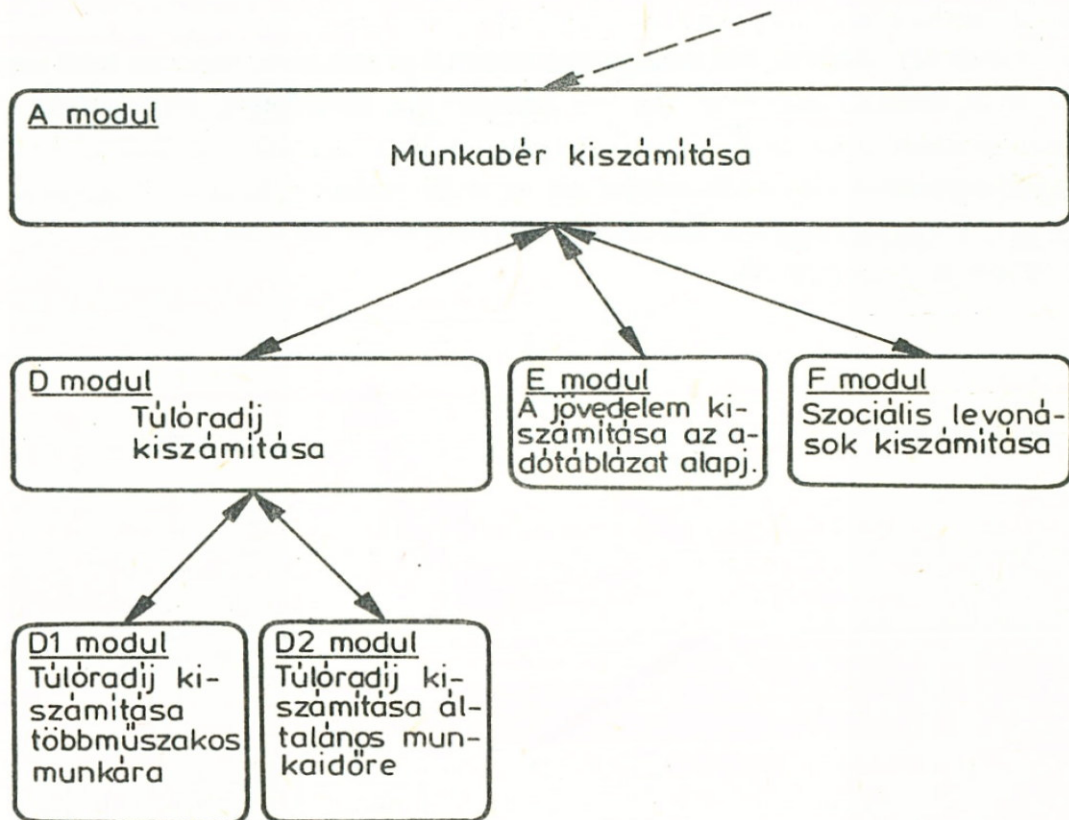


A megoldáshoz csak akkor lehet hozzáfogni, ha a feladatot ilyen módon részekre bontottuk. Valamennyi megfogalmazott problémához egy **programmodult** tervezünk úgy, hogy az áttekinthető számú programutasításból álljon. Az így kapott építőelemeket **moduloknak** nevezzük.

Egy, az építőköckaelv alapján kifejlesztett program végrehajtása mindig a legmagasabb szintű modullal kezdődik (az alábbi ábrán az 1. modul). Ez a modul szükség esetén egy alárendelt modult hív: a példában a 11. modult. Ha a feladat megkívánja, akkor ez a modul ismét egy vagy több alárendelt modult hív meg (pl. a 111. modult). Akkor, ha a legalsó modul is lefutott, vissza kell ugrani a legközelebbi magasabb szintre, és — abban az esetben, ha ennek nincsenek további alárendelt moduljai — a legmagasabb szintű modullra. Tehát mindig csak **függőlegesen** (fentről lefelé és letről felfelé) szabad ugrani, sohasem vízszintesen! Ezáltal nagyon terjedelmes rendszerek esetén is az utak világosak és áttekinthetők maradnak.



Az alábbi példa (kivonatosan) egy építőköcsoelv szerint felépített bérelszámoló-program egyszerűsített ábráját mutatja.



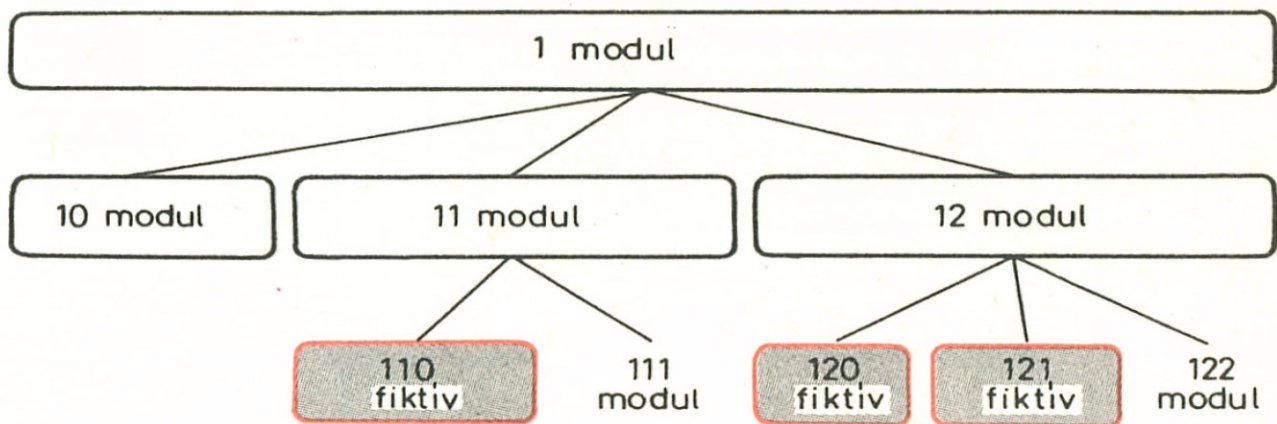
Az A modul a feladatot felosztja az alárendelt D, E, F stb. modulok között. A D modul feladatait még tovább osztja, és átadja azokat a D1, ill. a D2 moduloknak. Ha pl. a D2 modul kiszámította a túlóradjakat, átadja azokat a D modulnak (az ő megbízójának), amely azt a maga részéről továbbadja az A modulnak.

Ha a jövedelemadó, a túlórák, a szociális levonások vagy a betegbiztosítás tételei megváltoznak, akkor csak az érintett modulokat szükséges megváltoztatni. A fennmaradó program-építőelemek változatlan formában érvényben maradnak.

Ezzel a módszerrel nem kerülhet hiba azokba a programmodulokba, amelyeket a változások nem érintenek. Ezenkívül annak a programozónak, akinek a változtatást végre kell hajtania, csak egy modulba kell bedolgoznia magát.

Ha egy eljárást a moduláris technika elvén építettünk föl, akkor már korai időszakban megindulhat az egész rendszer tesztelése. Ha a felső szinteken elhelyezkedő modulok elkészültek, de a legalsó szintűek csak részben állnak rendelkezésre, akkor a még hiányzó modulokat **helyettesítő modulokkal** lehet pótolni (angolul: dummy), amelyek csak arra valók, hogy a fölérjük rendelt építőelemek számára tesztadatokat állítsanak elő. Az előzőleg rögzített tesztadatokat a fölérendelt modul átveszi úgy, mintha azok az alárendelt modul valódi futásának eredményei lennének.

Egy példa:



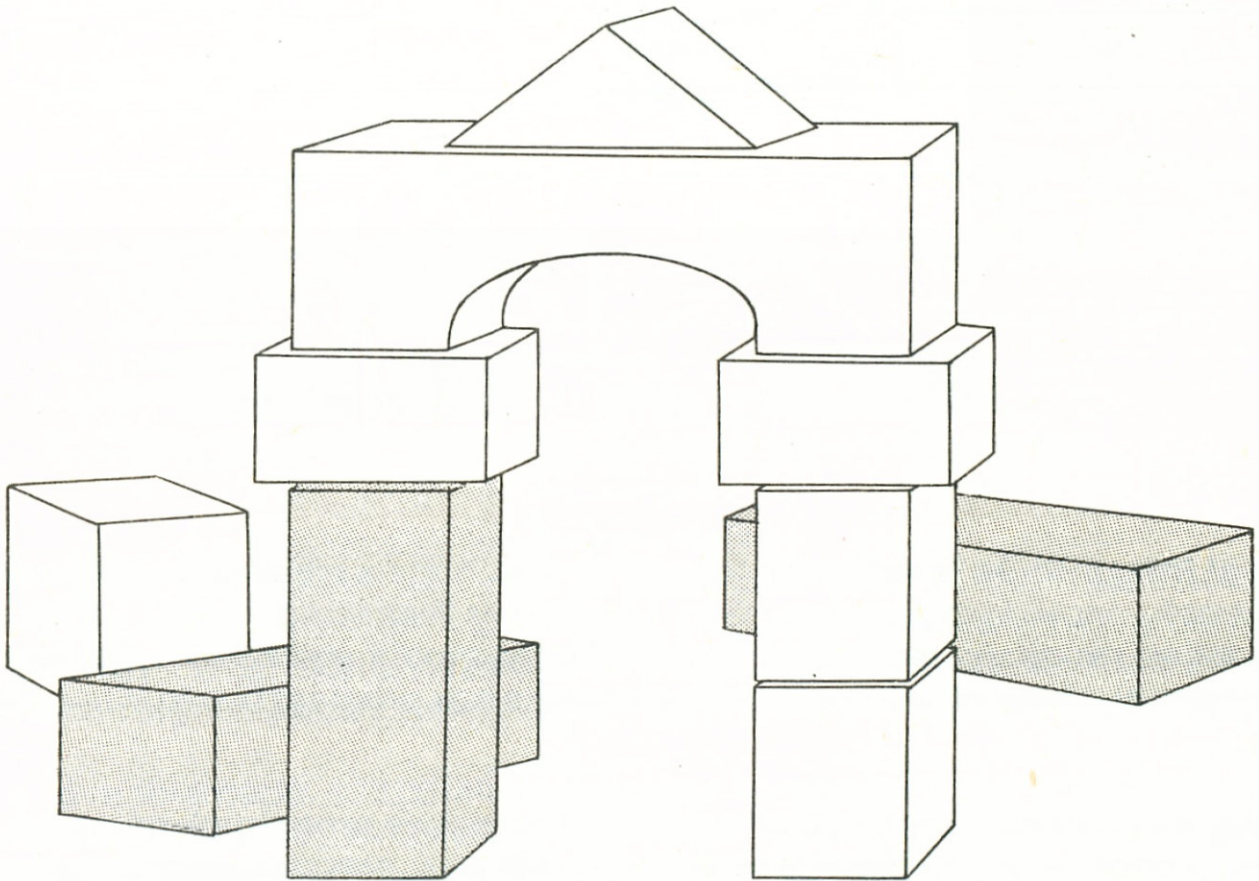
A **korai integrációs teszt** révén lehetővé válik, hogy a hibák olyan időpontban kerüljenek napvilágra, amikor még viszonylag kisszámú modul készült el. Ha a teljes rendszer tesztjét csak valamennyi modul elkészülte után lehetne végrehajtani, akkor pl. a tervezési hibák miatt a sok program használhatatlanná válna.

Mivel a moduláris technika alkalmazásával az egyes programok önmagukban zártak, ezeket a programrendszeren belül, sőt más programrendszerekben is többszörösen ki lehet használni. Az ilyen kidolgozott és ellenőrzött modulokat egy **modulkönyvtárban** lehet tárolni, amelyből a többi felhasználó, mint valami könyvtárból, a saját feladataihoz a szóba jöhető modulokat ki tudja másolni. Ez azt jelenti, hogy egy újonnan kifejlesztendő programot egy modulkönyvtárban tárolt programmodulokból lehet összeállítani. Természetesen fennmaradnak még olyan részek, amelyeket programozni kell! Különösen az olyan jelentős modulok, amelyek az alárendelt programok végrehajtási sorrendjét határozzák meg.

A moduláris technika előnyei igen kézenfekvők:

Az egyes feladatok és az azoknak megfelelő különálló programmodulok világos elhatárolása által a programok áttekinthetők, ezáltal:

- jobban ellenőrizhető a projekt előrehaladása,
- a programokat egyszerűbben lehet karbantartani és változtatni,
- az egyes programmodulokat más programokban vagy más rendszerekben is fel lehet használni,
- a rendszert korai állapotában lehet tesztelni, ezáltal a hibás fejlesztés veszélye csökken.





1. Az alábbi állítások közül melyek igazak a moduláris programozásra?

A moduláris programozás olyan módszer, amely támogatja a számítástechnikai eljárások fejlesztését.

Ez egy olyan elv, amelynek alapján egy feladatot részfeladatokra osztanak fel mindaddig, amíg áttekinthető egyedi feladatokhoz jutnak.

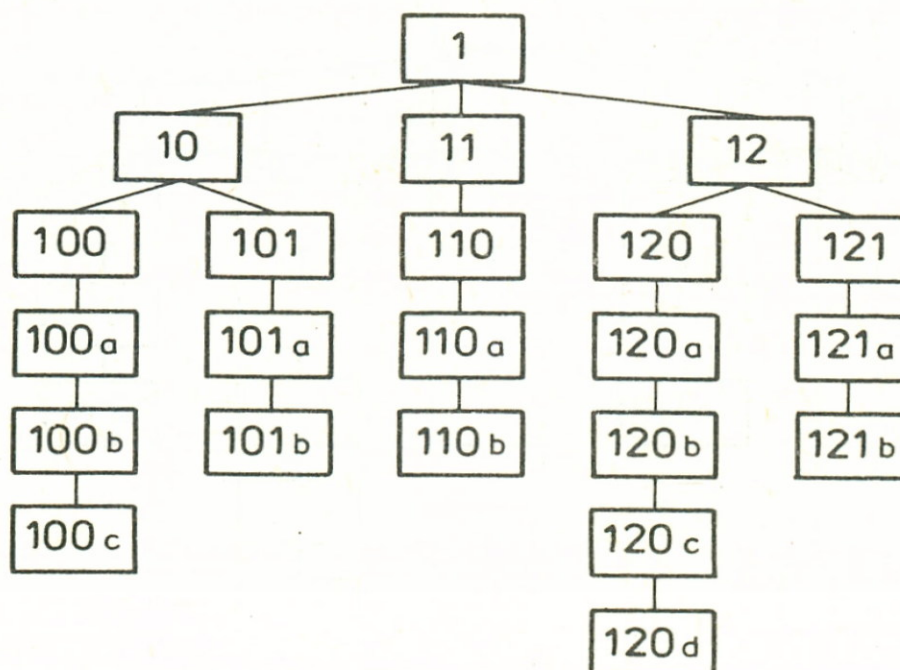
Valamennyi megfogalmazott feladathoz a későbbiekben egy programmodult fejlesztenek ki.

Az egyes modulok között sohasem szabad vízszintesen, hanem mindig csak függőlegesen ugrani.

A moduláris programozás segítségével tervezett programok jól áttekinthetőek, ellenőrizhetőek, karbantarthatóak, könnyen megváltoztathatóak és tesztelhetőek.

2. Mely moduloknak kell lefutniuk, mielőtt a 121b modul lefuthat?

.....



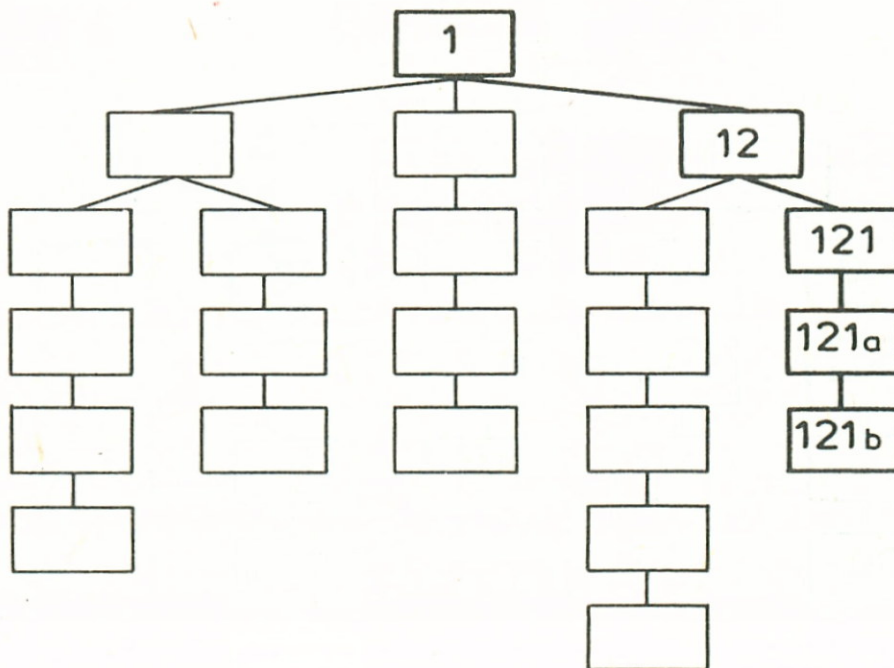


Válaszok

1. Valamennyi állítás helytálló!

- A moduláris programozás olyan módszer, amely támogatja a számítástechnikai eljárások fejlesztését.
- Ez egy olyan elv, amelynek alapján egy feladatot részfeladatokra osztanak mindaddig, amíg áttekinthető, egyedi feladatokhoz jutnak.
- Valamennyi megfogalmazott feladathoz a későbbiekben egy programmodult fejlesztenek ki.
- Az egyes modulok között sohasem szabad vízszintesen, hanem mindig csak függőlegesen ugrani.
- A moduláris programozás segítségével tervezett programok jól áttekinthetőek, ellenőrizhetőek, karbantarthatóak, könnyen megváltoztathatóak és tesztelhetőek.

2. Mielőtt a 121b modul feldolgozására sor kerülhetne, végre kell hajtani az **1**, **12**, **121** és **121a** modulokat.



Döntési táblázatok

A **döntési táblázatok** elsődlegesen a döntési helyzetek átgondolására és ábrázolására alkalmasak, és ezáltal gyakran a programtervezés egyéb módszereivel együtt alkalmazzák őket.

A döntési táblázatok segítenek abban, hogy az olyan *döntési helyzeteket*, amelyek sok különböző tényezőtől függenek, áttekinthetően ábrázoljuk. Segítségükkel a döntési helyzetet gyorsan és teljes egészében át tudjuk tekinteni.

Egy döntési táblázatba valamennyi szóba jöhető **tevékenységet** és minden **feltételt**, amelyektől tevékenységek végrehajtása függ, rögzítünk. Ezek után táblázatosan ábrázolni lehet, hogy mely feltételek fennállása esetén mely tevékenységeket kell végrehajtani.

A táblázat mindig az alábbi kialakítású:

FELTÉTELEK	SZABÁLYOK
TEVÉKENYSÉGEK	EREDMÉNYEK

A bal oldali két mezőbe valamennyi szóba jöhető feltételt és tevékenységet felveszünk.

A szabályok elnevezésű mezőben az *igen* vagy *nem* bejegyzések segítségével valamennyi megadott feltétel kombinációját lekérdezzük.

Ez elméletileg nehezen érthető meg. Szemléltessük a módszert egy példa segítségével!

Példa:

Számlák előállításakor különböző árengedményeket érvényesítenek, attól függően, hogy a vásárolt áru mennyisége egy bizonyos árumennyiség felett van-e, vagy hogy törzsvevőről van-e szó vagy sem.

ÁRENGEDMÉNYEK ALKALMAZÁSÁHOZ DÖNTÉSI TÁBLÁZAT					
FELTÉTELEK	SZABÁLYOK				
	SZ1	SZ2	SZ3	SZ4	Egyéb
Megrendelt menny. < 1000 db	i	-	i	-	-
Megrendelt menny. > 1000 db	-	i	-	i	-
Törzsvevő	n	n	i	i	-
TEVÉKENYSÉGEK	EREDMÉNYEK				
Nincs engedmény	x				x
10 % engedmény		x	x		
15 % engedmény				x	

i : igen
n : nem

- : közömbös
x : végrehajtandó tevékenység

A négy szabálynak itt a következő jelentése van:

Az 1. szabály (SZ1) azokra az ügyfelekre vonatkozik, akik 1000 darabnál kevesebbet rendelnek és nem állandó ügyfelek; ebben az esetben árengedmény nincs.

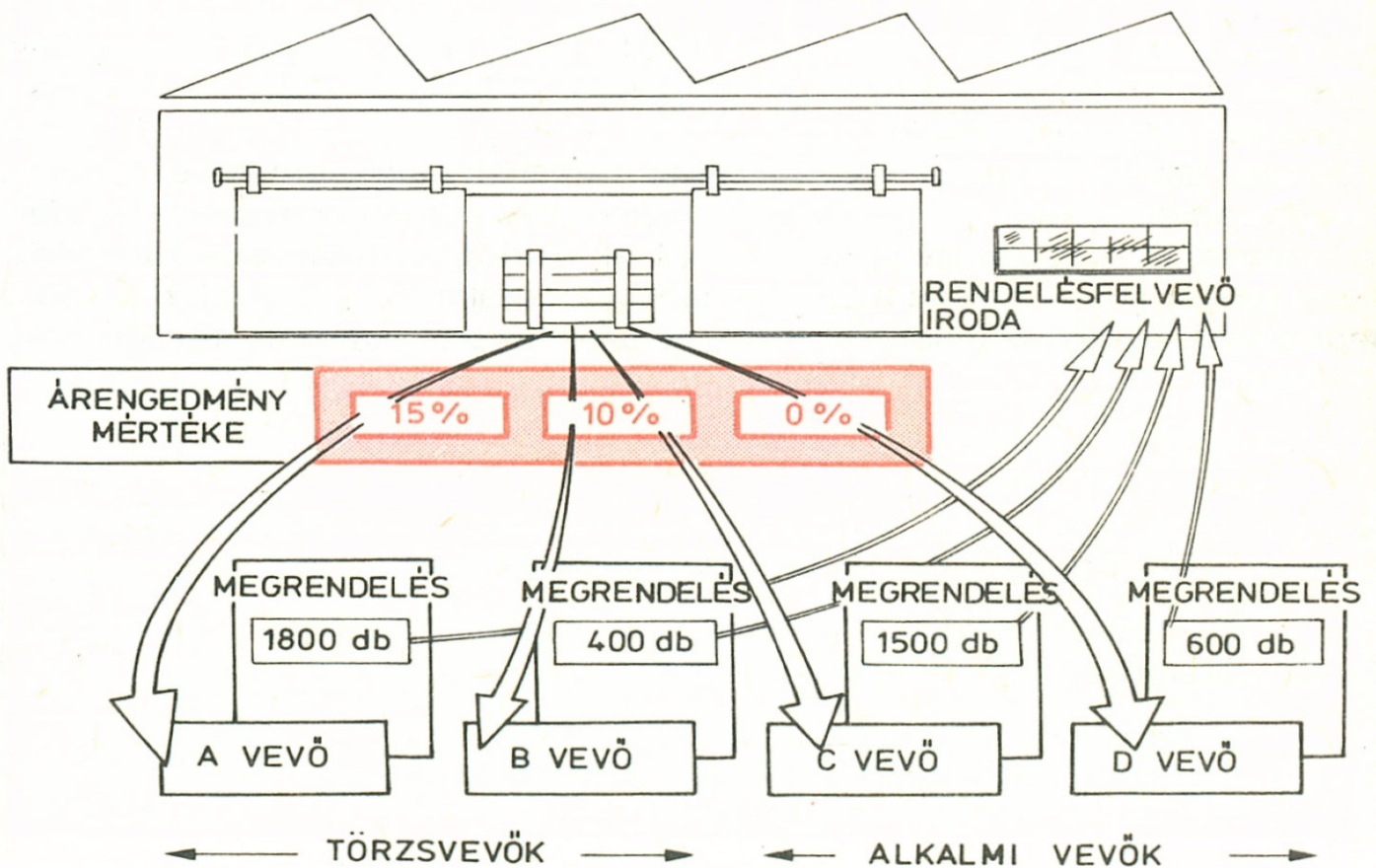
A 2. szabály (SZ2) azokra az ügyfelekre vonatkozik, akik 1000 darabnál többet rendelnek, ez esetben 10% engedmény érvényesíthető.

A 3. szabály (SZ3) olyan törzsvevőkre vonatkozik, akik 1000 darabnál kevesebbet rendelnek, ebben az esetben az engedmény 10%.

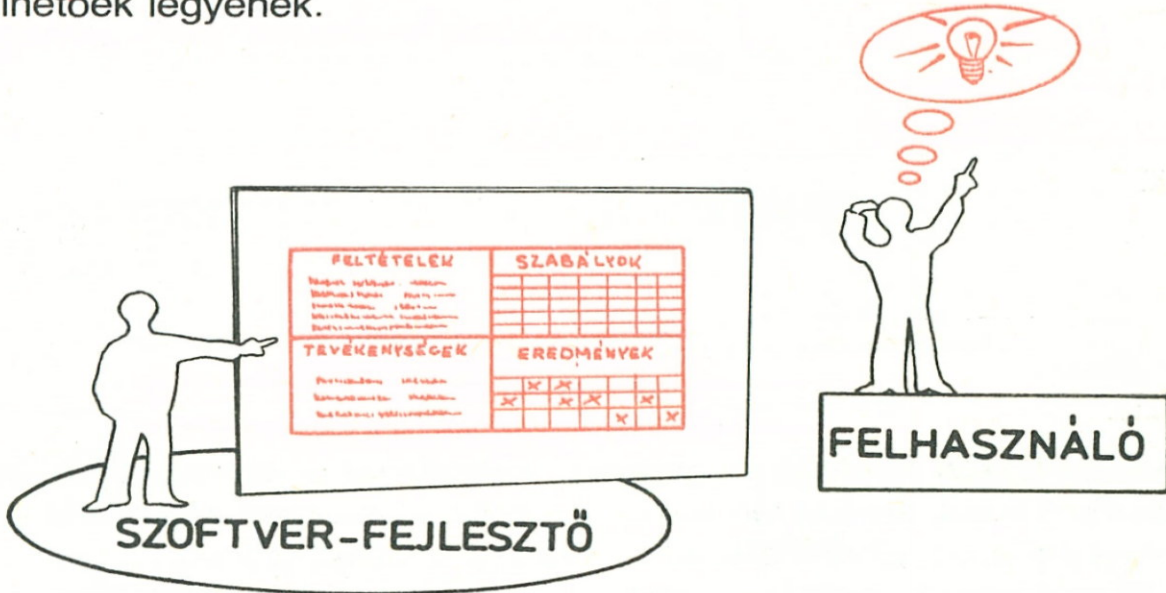
A 4. szabály (SZ4) azokra a törzsvevőkre vonatkozik, akik 1000 darabnál többet rendelnek. Ezek 15% árengedményben részesülnek.

A túloldalon álló példában azt az esetet emeltük ki, amikor egy törzsvevő 1000 darabnál többet rendel és 10% árengedményben részesül.

A szabályok között rendszerint „**egyéb**” kategóriát is felveszünk. Ebbe azok az esetek kerülnek, amiket rendszerint nem kezelünk, mert a feladat leírásához nincs rájuk szükség. Ennek ellenére mindig fel kell őket sorolni, hogy arra kényszerítsük magunkat, hogy meggondoljuk: melyek azok a feltételkombinációk, amelyek a szabályok között nem lehetségesek.



A döntési táblázatok legnagyobb előnye abban van, hogy laikusok által is érthetőek; ez megkönnyíti az együttműködést a számítástechnikusok és a mindenkor szakterület munkatársai között, valamint segít abban, hogy a félreértések elkerülhetőek legyenek.



Azonban nem alkalmas valamennyi feladat arra, hogy döntési táblázatban írassuk le azokat (pl, elágazás nélküli hosszú folyamatok, vagy sokszorosan egymásba ágyazott hurkok).

A döntési táblázatok nemcsak segítséget nyújtanak a program tervezésében, hanem amennyiben döntéstáblázat-előfordítónk (preprocesszor) is van, programozási segédletnek is használhatóak. A **döntéstáblázat-előfordítók** olyan programok, amelyek a döntési táblázat tartalmát kódolják! Ezt a kódot beprogramozáskor fel lehet használni, és ezáltal a programozási munka csökkenthető.



1. Töltse ki az alábbi döntési táblázatot úgy, hogy tartalma a következő tényeket írja le:

A bank ügyfelei egy bizonyos összeget csak akkor emelhetnek le számlájukról, ha arra a számlán fedezetük van, vagy ha valaki hitelt biztosít, és a kifizetendő összeg kisebb vagy egyenlő az engedélyezett hitelösszegnél.

.....	SZ1	SZ2	SZ3	SZ4	Egyéb
Kifizetendő összeg \leq folyószámlán lévő összeg					
Van hitel					
Kifizetendő összeg \leq hitel					
Kifizetendő összeg $>$ hitel					
.....					
Kifizethető					
nem fizethető ki					

2. A következőkben felsorolt előnyök közül melyek érvényesek a döntési táblázatra?

A döntési táblázatokat laikusok is megérthetik.

Döntési táblázatok segítségével a döntési helyzeteket teljes egészében át lehet tekinteni.

A döntésitáblázat-előfordítók segítségével a döntési táblázatból kódot lehet generálni.



Válaszok

1. A kitöltött döntési táblázat a következő:

Feltételek	Szabályok				
	SZ1	SZ2	SZ3	SZ4	Egyéb
Kifizethető összeg \leq folyószámlán levő összeg	i	n	n	n	-
Van hitel	-	i	i	n	-
Kifizethető összeg \leq hitel	-	i	-	-	-
Kifizethető összeg $>$ hitel	-	-	i	-	-
Tevékenységek	Eredmények				
Kifizethető	x	x	-	-	-
nem fizethető ki	-	-	x	x	x



*Gondolt Ön az „egyéb” rovatra?
Elvileg ezt mindig fel kell venni és be kell jelölni azért,
hogy valamennyi feltételkombinációt figyelembe vegyünk,
azokat is, amelyek a feladat leírásához nem szükségesek.*

2. Valamennyi állítás a döntési táblázatok előnyei közé sorolható.

- A döntési táblázatokat laikusok is megérthetik.
- Döntési táblázatok segítségével a döntési helyzeteket teljes egészében át lehet tekinteni.
- A döntésitáblázat-előfordítók segítségével a döntési táblázatból kódot lehet generálni.

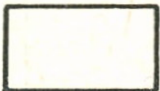

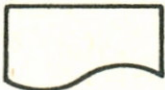




Az adat-folyamatábra

Az adat-folyamatábra olyan segédeszköz, amellyel egy számítógépes eljárás lefolyása grafikusan ábrázolható. Az adat-folyamatábra feltünteti az adatok útját, az érintett hardvert és az eljárás feldolgozási állomásait.

Az adat-folyamatábráról le lehet olvasni:

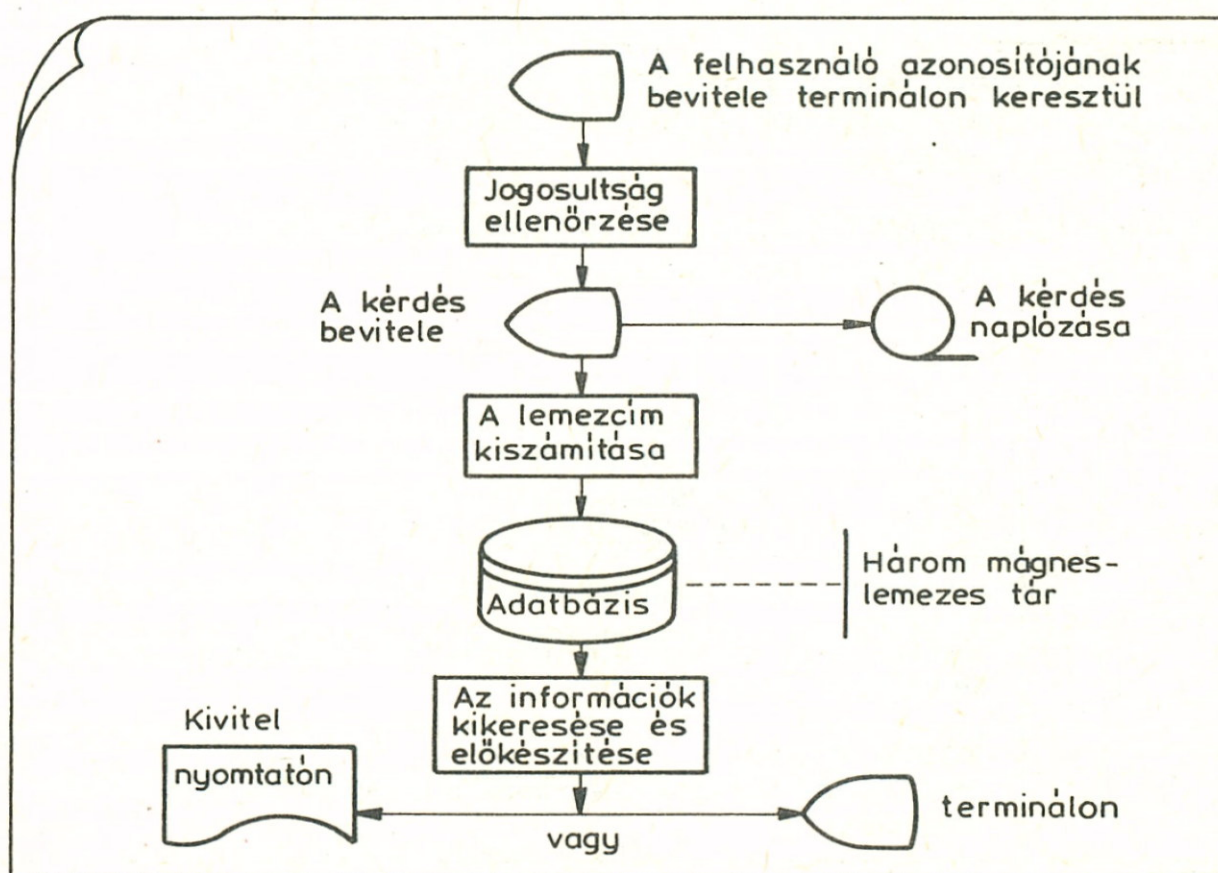
- milyen *adathordozókra* rögzítik az adatokat,
- mely *perifériákon* át történik az adatbevitel,
- milyen *utat* járnak végig az adatok a számítógépen belül,
- melyek azok a *programok*, amelyek az adatok feldolgozását hajtják végre a számítógépen,
- mely *perifériákon* át történik az adatok kivitele,
- milyen *műveleteket* kell a számítógépen kívül végrehajtani, ezek közül melyek történnek segédeszközök igénybevételével, és melyek manuálisan.

Az adat-folyamatábrákon szabványosított **jelképek** szerepelnek. Ezek közül a legfontosabbak a következők:

Jelkép	Jelentés
	Feldolgozási tevékenység általában: Ezzel a jelképpel kell minden számítógépes feldolgozást ábrázolni. A számítógépen belül rendszerint minden ilyen tevékenységéhez egy program vagy egy programmodul tartozik.
	Kijelzés optikai vagy akusztikus módon. Rendszerint a „terminál” jelölésére használják.
	Írásos anyag, mint űrlap, géppel olvasható bizonylat, de kinyomtatott eredmény is.
	Mágnestlemezes tár
	Mágnesszalagos tár
	Adatfolyam, amelynek végét mindig a folyam irányába mutató nyíllal kell ellátni.
	Adatátvitel: pl.: telex vagy távbeszélő összeköttetés.

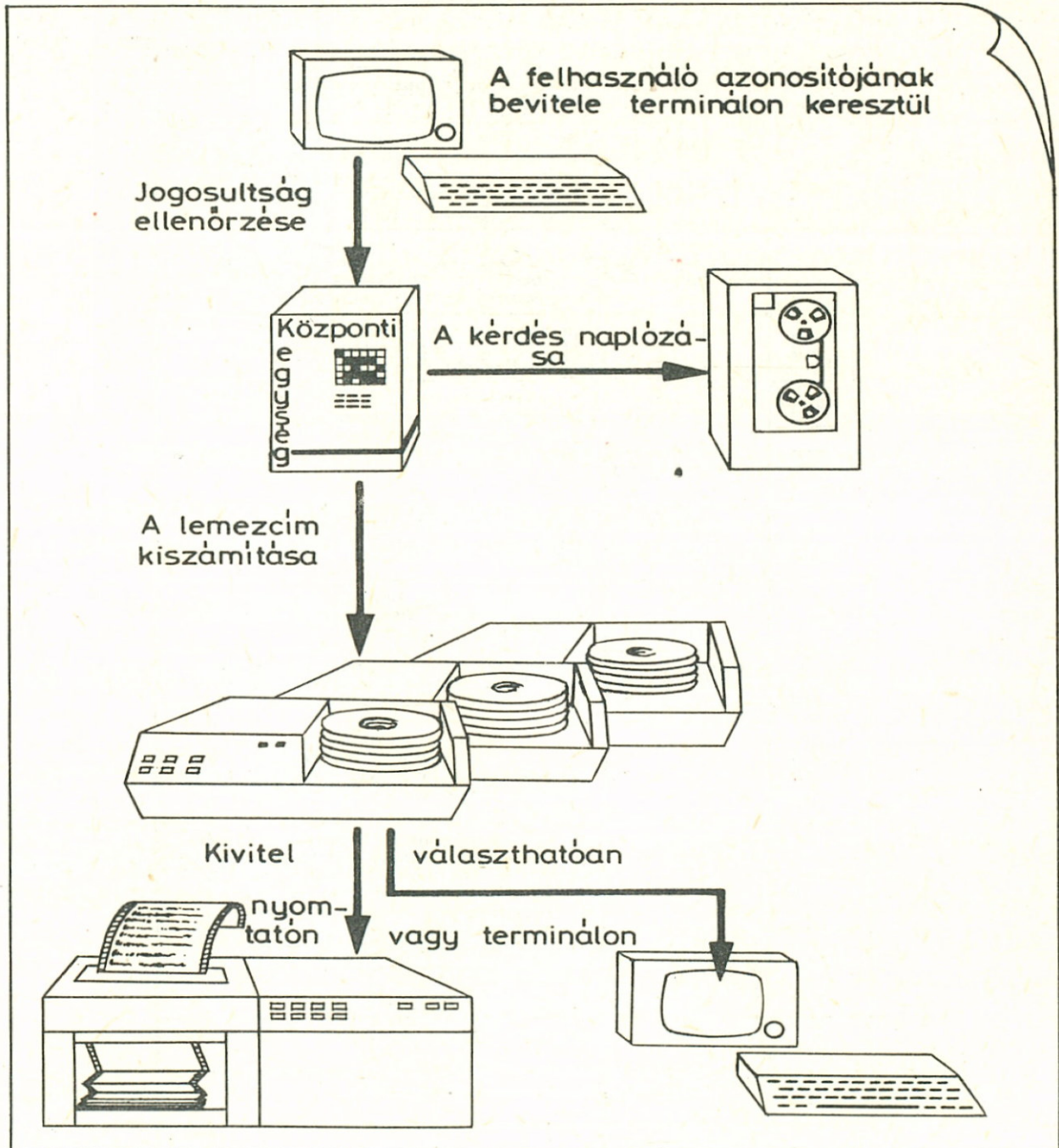
Példa:

Ábrázoljuk az adatok útját információk munkaügyi adatbázisból való visszakeresése esetén. Egy dolgozó tárolt adatait akarjuk visszakeresni. Egy munkaügyi előadó terminálon át felteszi a kérdéseket. Ezeket mágneszalagra rögzítik. Az illetékességi ellenőrzést követően — amelynél megtörténik az előadó által begépelte kulcsszó ellenőrzése — a számítógép meghatározza a mágneslemezen tárolt adatok címét. Ezt követően be kell olvasni a megfelelő információkat a mágneslemezről az operatív tárba, azokat áttekinthető módon rendezni kell, ezután kiválasztható, hogy azt nyomtatóra vagy a terminálra kell-e kivinni.
























Az adat-folyamatábrából megállapítható, hogy az itt leírt számítógépes eljárás esetén alapvetően három folyamatot kell figyelembe venni, amelyet — terjedelmétől függően — három különböző programmal vagy alprogrammal lehetne megvalósítani. Ennek alapján egy-egy programot kellene megírni az „illetékességi vizsgálat”, a „lemezcím megállapítása” és az „információk rendelkezésre bocsátása és előkészítése” feladatokra.

Annak érdekében, hogy a számítógépes eljárásokat laikusok számára is érthetővé tegyük, gyakran használunk olyan ábrázolási módot, amely nem a szabványos adat-folyamatábra jelképekből épül fel, és elsősorban az alkalmazott hardvert ábrázolja. Erre a célra szemléletes **berendezésjelképeket** használunk. A szóban forgó példára ilyen jelképekkel készített ábra a következő:



Hivatalosan az adat-folyamatábrákat a ma érvényben levő szabványos jelképekkel kell felrajzolni. Az alábbiakban bemutatunk egy összeállítást a szabványos adat-folyamatábra **jelképekből**:

Jelkép	Megnevezés	Magyarázat, ill. felhasználás
	Feldolgozás általában	Ezzel a jelképpel kell jelölni minden számítástechnikai műveletet, beleértve a lyukasztást, ellenőrzést, egyesítést stb.
	Kézi bevitel	Vezérlő-, ellenőrző-, vagy javító adatok bevitele kézzel
	Kijelzés	Optikai vagy akusztikus kijelzés. Pl.: rajzgép vagy zümmögő.
	Adathordozó általában	Ezt a jelképet kell alkalmazni ha az adathordozót nem akarjuk pontosabban meghatározni, vagy a rendszer tervezése során az adathordozó fajtája még nincs eldöntve
	Adathordozó, közvetlen eltérésű tár	A tár fajtáját (pl.: mágneslemez) a jelképen meg kell adni. Ha többfajta tár is szóbajön, a lehetséges változatokat megjegyzésben meg kell adni.
	Írásos anyag	Ide tartoznak a nyomtatott listák, a géppel olvasható bizonylatok, az űrlapok, a bizonylatok stb.
	Lyukkártya	Ha lehetséges, meg kell adni a lyukkártya fajtáját. Pl.: anyagkártya, változáskártya, összegkártya stb.
	Lyukszalag	
	Mágnesszalag	A szalag fajtáját és az állomány azonosítóját meg kell adni. Pl.: Növekményszalag F014.
	Kapcsoló	A kapcsolat több helyről, de csak egy pontra mutathat. Összetartozó kapcsolók jele azonos legyen.
	Megjegyzés	Ezt a jelképet valamennyi eddig bemutatott jelkép mellett lehet alkalmazni.
	Folyamatvonal	Előnyben kell részesíteni: a/ a felülről lefelé mutató, b/ a balról jobbra mutató irányokat. A vonal végére mindig az adatáramlás irányát jelző nyílhegyet kell rajzolni.
	Adathordozók szállítása	Ha az adathordozók szállítását különösen ki akarjuk emelni, a folyamatvonal végére a szállítás irányába mutató kettős nyilat kell rajzolni és a küldő ill. a fogadóhelyet meg kell adni.
	Adatátvitel	Adatátvitel pl.: telexkapcsolaton át
	Egyesítés	
	Válogatás	
	Rendezés	
	Egyesítés egyidejű válogatással	
	Segédfunkció végrehajtása	Gépesített, de nem a számítógép által vezérelt segédeszköz igénybevétele. Pl.: lyukkártyák vagy lyukszalag kézi előállítása
	Kézi beavatkozás	Nem gépi segédeszköz igénybevétele. Pl.: bejegyzés egy listába, szalagcsere.
	Adathordozó	Olyan adathordozó, amelyet nem a számítógép vezérel. Pl.: fiókos kartotékrendszer



1. Mit lehet az adat-folyamatábráról leolvasni?
Jelölje meg a megfelelő válaszokat!

Milyen adathordozóra történik az adatrögzítés?

Mely perifériákat használunk adatok be- és kivitelére?

Milyen az adatok útja a számítógépen belül?

Mely programokkal történik az adatok feldolgozása a számítógépben?

Hogyan megy végbe az adatok feldolgozása a számítógépen belül?

Az eljárás mely műveleteit kell manuálisan végrehajtani?

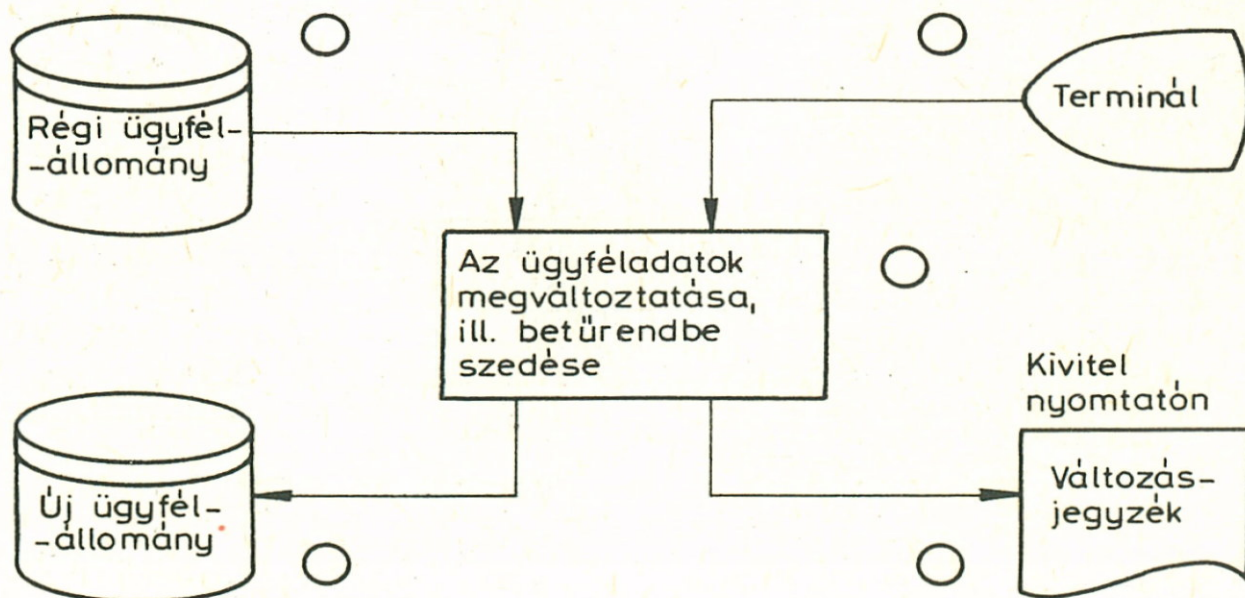
2. Tekintse meg az alábbi adat-folyamatábrát!

a) Milyen perifériák vesznek részt a feldolgozásban?

.....

.....

b) Jelölje meg, hol vannak ábrázolva a feldolgozási folyamatok!





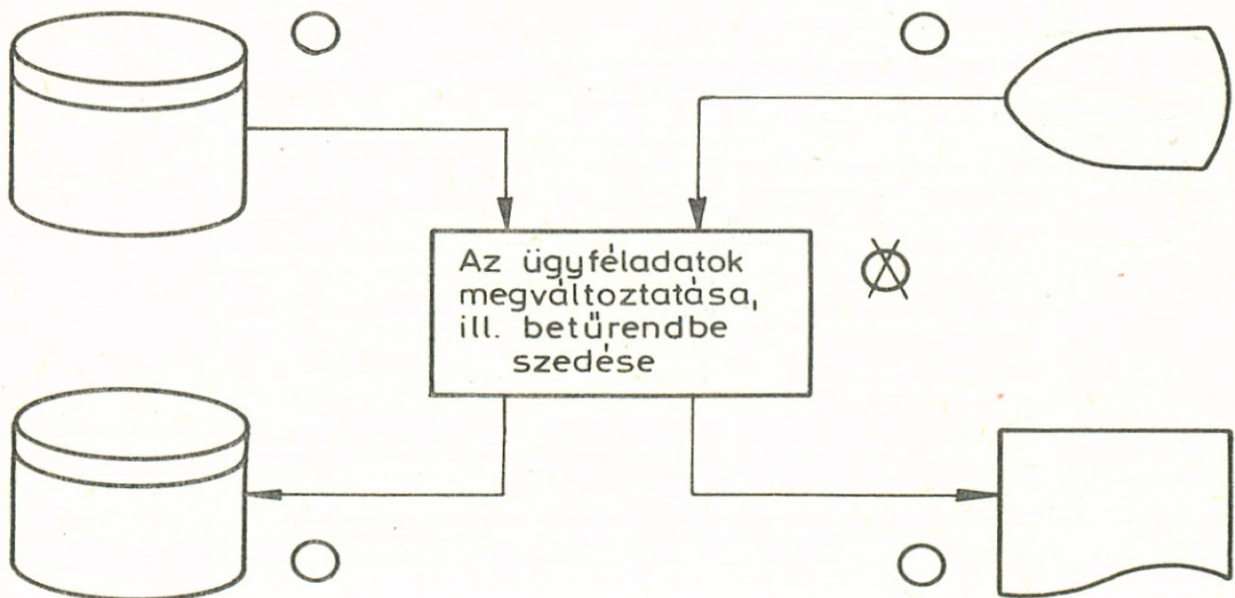
Válaszok

1. Az adat-folyamatábra megmutatja, hogy:

- Milyen adathordozóra történik az adatrögzítés.
Mely perifériákat használunk adatok be- és kivitelére.
- Milyen az adatok útja a számítógépen belül.
- Mely programokkal történik az adatok feldolgozása a számítógépben.
- Azt, hogy hogyan megy végbe az adatok feldolgozása a számítógépen belül, az adat-folyamatábráról nem lehet leolvasni, ezt a program-folyamatábra mutatja, amelyet a következő fejezetben ismertetünk.
- Az eljárás mely műveleteit kell manuálisan végrehajtani.

2. a) A feldolgozásban a **terminál**, a **mágneslemezes tároló** és a **nyomtató** vesznek részt.

b) Ebben az adat-folyamatábrában csak egy feldolgozási folyamat van.

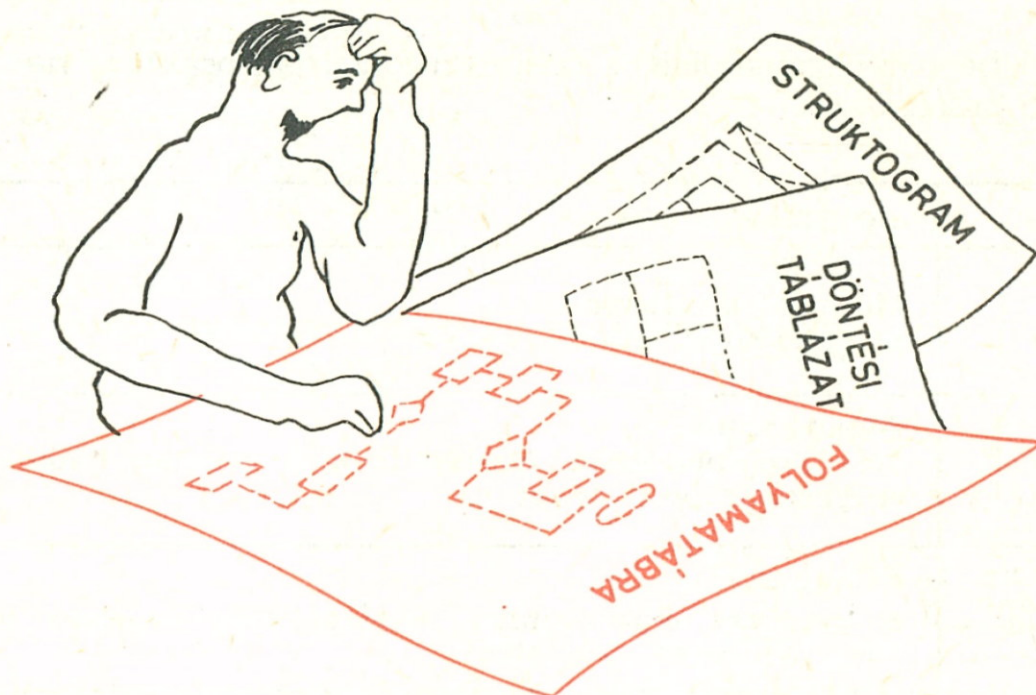


A program-folyamatábra

A program megtervezéséhez és ábrázolásához sok módszer létezik.

A program megtervezésének hagyományos módja a **program-folyamatábra** elkészítése. A program-folyamatábrákat ma már egyre nagyobb mértékben váltják le az újabb módszerek, amelyek a szoftvertechnika elveihez közelebb állnak.

A program-folyamatábrát ennek ellenére még ma is használják, mégpedig azért, mert az a laikusok számára igen szemléletes. Mint a programtervezés egyik lehetséges módszerét, mi is ismertetjük.

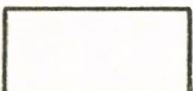
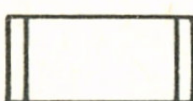

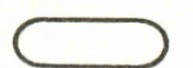

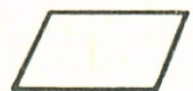


A program-folyamatábra egy segédeszköz, amely a számítógép-logikának megfelelően munkaszakaszokra bontja, és grafikusán ábrázolja a feladatot. Valamennyi olyan műveletet szemléletesen ábrázol, amelyet a programnak végre kell hajtania. Ábrázolja ezenkívül a program elágazásait.

Rendszerint nem lehetséges a feladatot olyan kis lépésekre felosztani, hogy azokat a programozási nyelv egyes utasításainak a programozó megfeleltethesse. A feladatot először csupán nagyobb részeke osztják, majd ezután lehet egyre jobban finomítani a felosztást. A program-folyamatábrát mint grafikus segédeszközt valamennyi részletességi fokon alkalmazhatjuk a feladat megoldásának ábrázolására.

A részletesen kidolgozott program-folyamatábra alapján közvetlenül lehet a programot megírni (kódolást elvégezni).

A következőkben összefoglaljuk a program-folyamatábrákban használatos legfontosabb **jelképeket**.

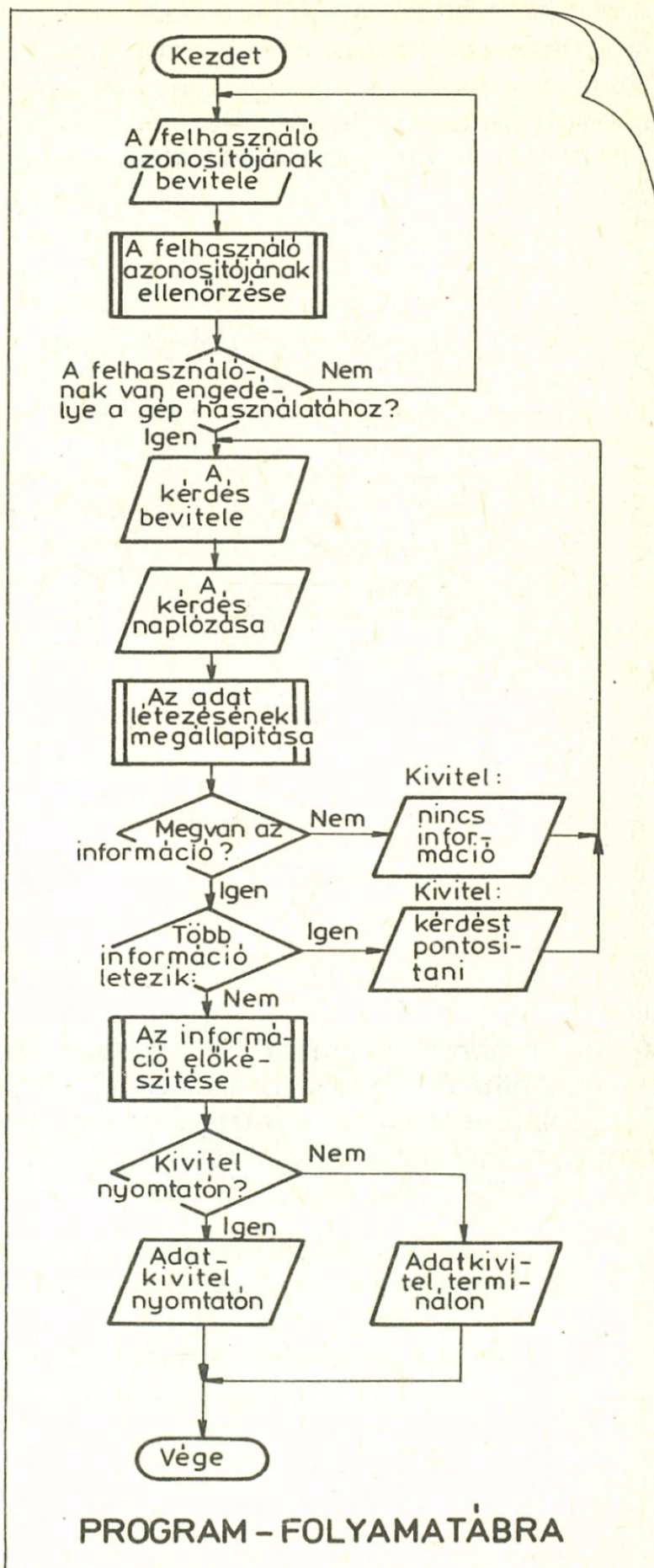
Jelkép	Jelentés
	Művelet általában: pl.: számítás, átvitel, törlés
	Alprogram: Zárt programrészlet, amelyet a hívó program különböző helyeiről hívni lehet
	Elágazás: A program végrehajtása feltétel függvényében különböző helyeken folytatódik
	Határoló: mint pl. a programfutás eleje és vége
	Folyamat vonal: A programfutás irányának kiemeléséhez nyíllal lehet ellátni
	Bevitel, kivitel: Ezzel a jelképpel ábrázoljuk az összes be- és kivitelt a perifériákról és a perifériákra

Példa:

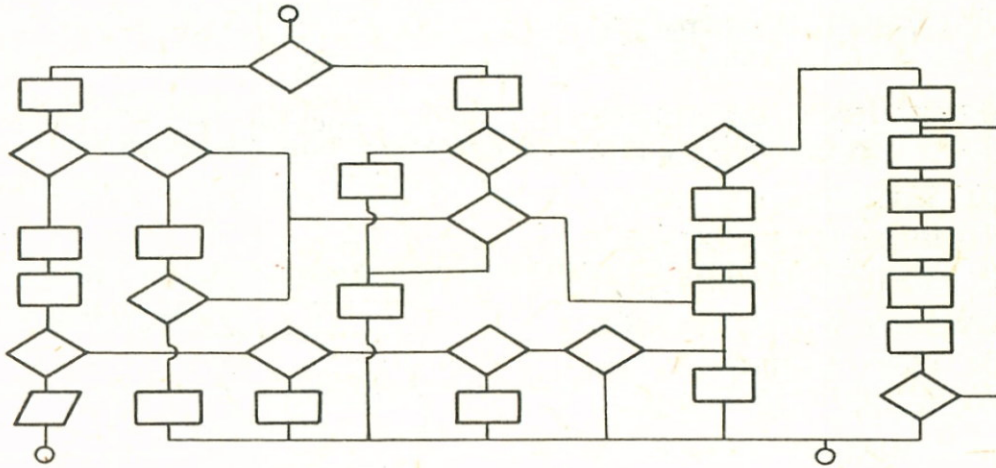
Az itt látható **program-folyamat-ábra** fő lépéseiben mutatja meg az információk munkaügyi adatbázisából való visszakeresés feladatának programját, amely feladatra az **adat-folyamatábrát** az előző fejezetben ismertettük.

A részletezés foka az egyes lépéseknél különböző. Az „a keresett adat létezésének megállapítása” feladatot pl. több száz utasítással lehetne leírni, mért az adatbázisban folyó keresés legtöbbször igen bonyolult feladat. A megfelelő, két vonallal lezárt téglalapokban ezt mint alprogramot vettük fel.

Valamennyi alprogram számára önálló program-folyamatábra készül.



A program-folyamatábra a hozzá nem értő számára is a program logikájának legérthetőbb ábrázolásmódja. Ez a megállapítás azonban csak viszonylag rövid programokra érvényes. Azonban minél terjedelmesebb és bonyolultabb a megoldandó feladat, annál áttekinthetlenebbé válik a folyamatábra, és annál nehezebb az utólagos változtatások végrehajtása.

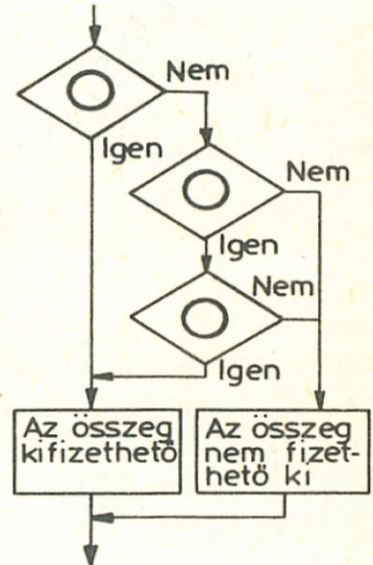


Az újabb ábrázolási módokat, amelyek segítségével a bonyolultabb feladatok is áttekinthető módon ábrázolhatók, ma már szívesebben használják. Ilyen ábrázolási eszköz pl. a **struktogram**, amelyet a továbbiakban a strukturált programozás téma keretében tárgyalunk.



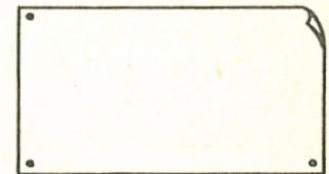
1. Ábrázolja program-folyamatábra segítségével az alábbi döntési táblázatban levőket úgy, hogy a megfelelő számokat beírja!

Feltételek	Szabályok					
	SZ1	SZ2	SZ3	SZ4	egyéb	
Kifizetendő összeg \leq folyószámlán levő ö.	i	n	n	n	-	①
Van hitel	-	i	i	n	-	②
Kifizetendő összeg \leq hitel	-	i	n	-	-	③
Kifizetendő összeg $>$ hitel	-	-	i	-	-	
Műveletek	Eredmények					
Kifizethető	x	x	-	-	-	
nem fizethető ki	-	-	x	x	-	



2. Rajzolja be az alprogram jelképét, és egészítse ki a következő mondatot:

Az alprogram jelképe



Minden alprogramnak külön

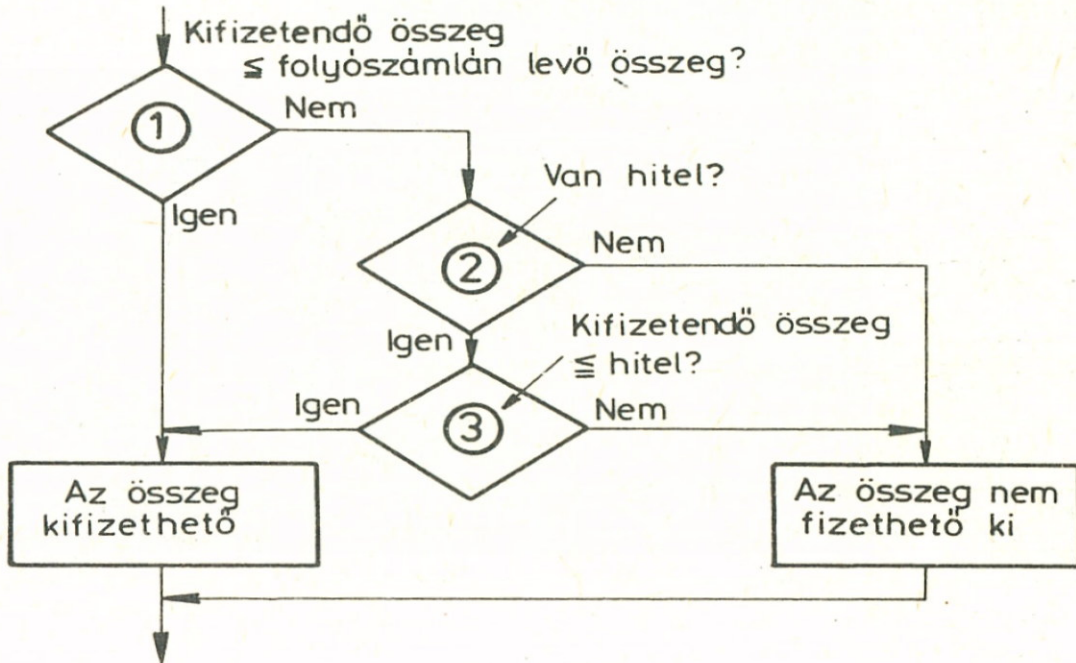
3. Mely állítások illenek a program-folyamatábrára?

- A program-folyamatábra segédeszköz a program logikájának ábrázolására.
- Egy feladat programjának megtervezése közben csak egy program-folyamatábrát rajzolnak.
- A program-folyamatábra a legrészletesebb alakjában a program kódolásának alapja lehet.
- Minél bonyolultabb a megoldandó feladat, annál áttekinthetlenebb a program-folyamatábra.

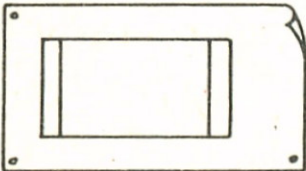


Válaszok

1.



2. Az alprogram jelképe:



Minden alprogramnak külön **program-folyamatábrája** van.

3. Az alábbi állítások illenek a program-folyamatábrára:

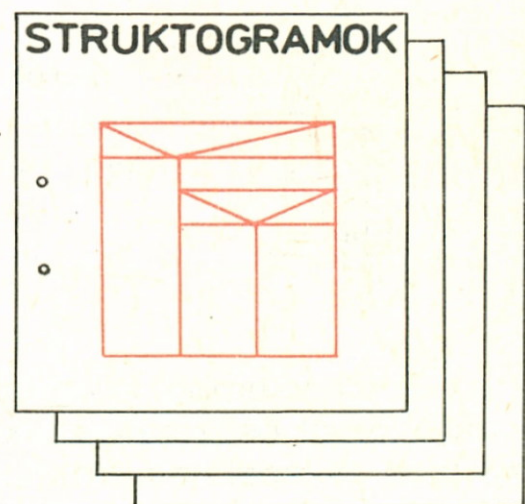
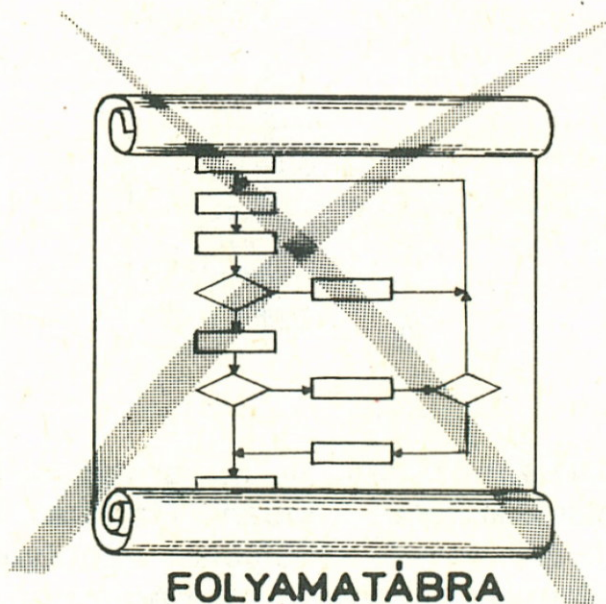
- A program-folyamatábra segédeszköz a program logikájának ábrázolására.
- Egy feladathoz több, egyre részletesebb folyamatábra tartozik.
- A program-folyamatábra legrészletesebb alakjában a program kódolásának alapja lehet.
- Minél bonyolultabb a megoldandó feladat, annál áttekinthetlenebb a program-folyamatábra.

Strukturált programozás

A **strukturált programozás** a szoftver tervezésének és a programozásnak manapság sűrűn alkalmazott módszere. A strukturált programozás a moduláris építőköckaelven alapszik.

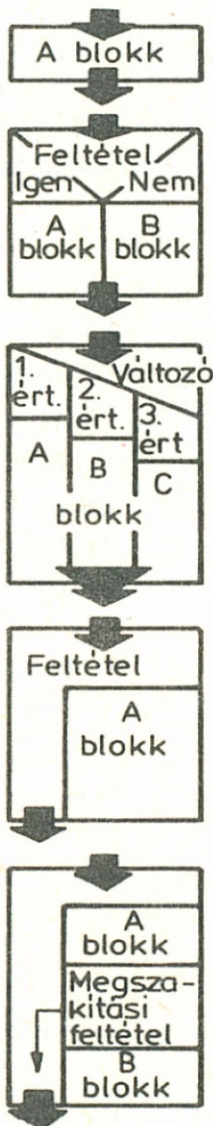
Az a szoftver, amelyet a strukturált programozás módszerével állítottak elő, mindig „felülről lefelé” készül. Az adott feladat megoldását először egészen nagyvonalakban körvonalazva köznapi nyelven fogalmazzák meg, majd több lépésben finomítják mindaddig, ameddig a részletes megoldás előáll, esetleg a kiválasztott programozási nyelv utasításainak felhasználásával. Ez a lépésenkénti eljárás a vázlatos megfogalmazásból a részletes megoldásig „**top-down**” (*felülről lefelé*) **fejlesztés** néven ismeretes.

A strukturált programozás további fontos alapszabálya, hogy az mindössze néhány **programelem** alkalmazására korlátozódik. Ilyenek pl.: az egyszerű elágazás, a többszörös elágazás, a programhurok. (Ezeket a következőkben ismertetjük.) Megállapítható ui., hogy bonyolult programokat is fel lehet építeni az ilyen típusú elemek alkalmazásával. Ez az eljárás ugyan korlátozza a programfejlesztő szabadságát, mivel nincs lehetősége többé többszörösen hurkolt és egymásba ágyazott programok tervezéséhez. Ezeknek a formai előírásoknak a betartásával azonban *áttekinthető*, és a kívülállók számára *jobban olvasható* programok írhatók.



A szoftverfejlesztőt strukturált programozásra kényszerítő ábrázolási segéd-eszköz a **struktogram**. A struktogram egy vagy több **struktúrablokkból** áll. A struktúrablokkok megfelelnek az említett típuselemeknek. Az alábbiakban a struktúrablokkok öt legfontosabb fajtáját mutatjuk be, és összehasonlításként szembe állítjuk velük a program-folyamatábra megfelelő jelképeit.

Folyamatábra



Feldolgozási utasítás

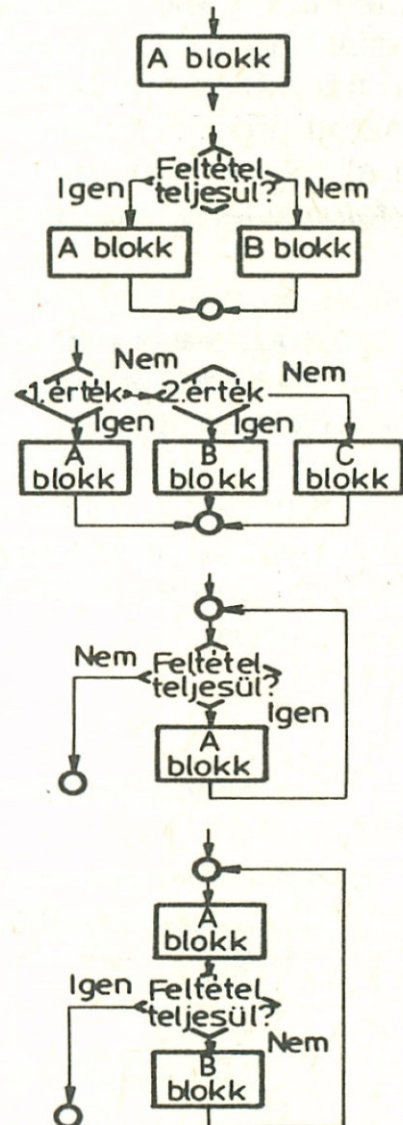
Egyszerű elágazás

Többszörös elágazás

Programhurok elővizsgálattal

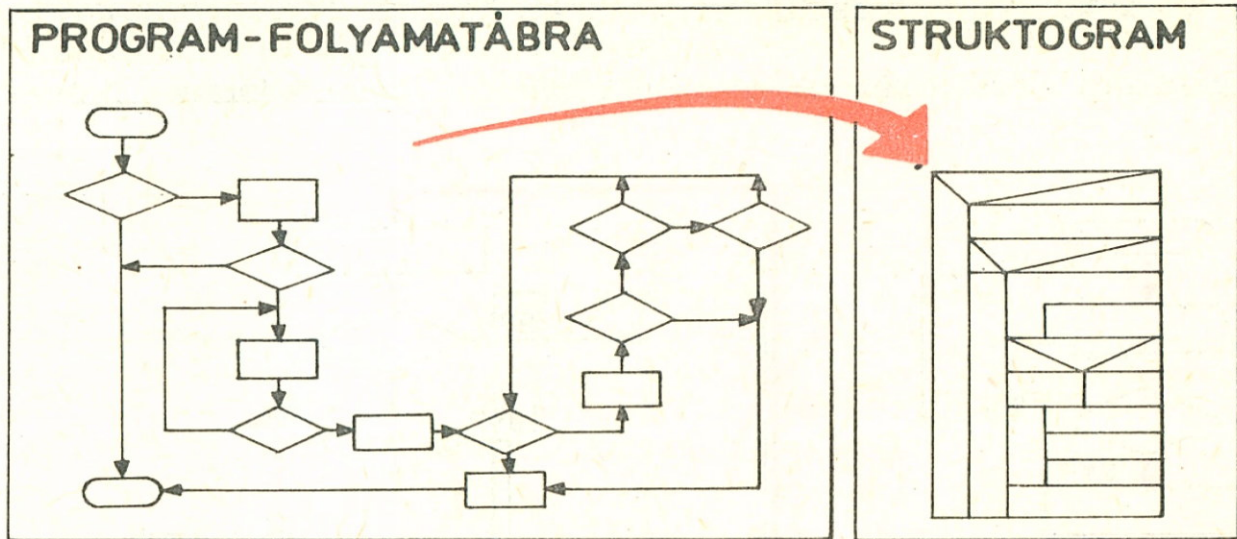
Hurok, hurkon belüli megszakítási feltétellel

Program-folyamatábra

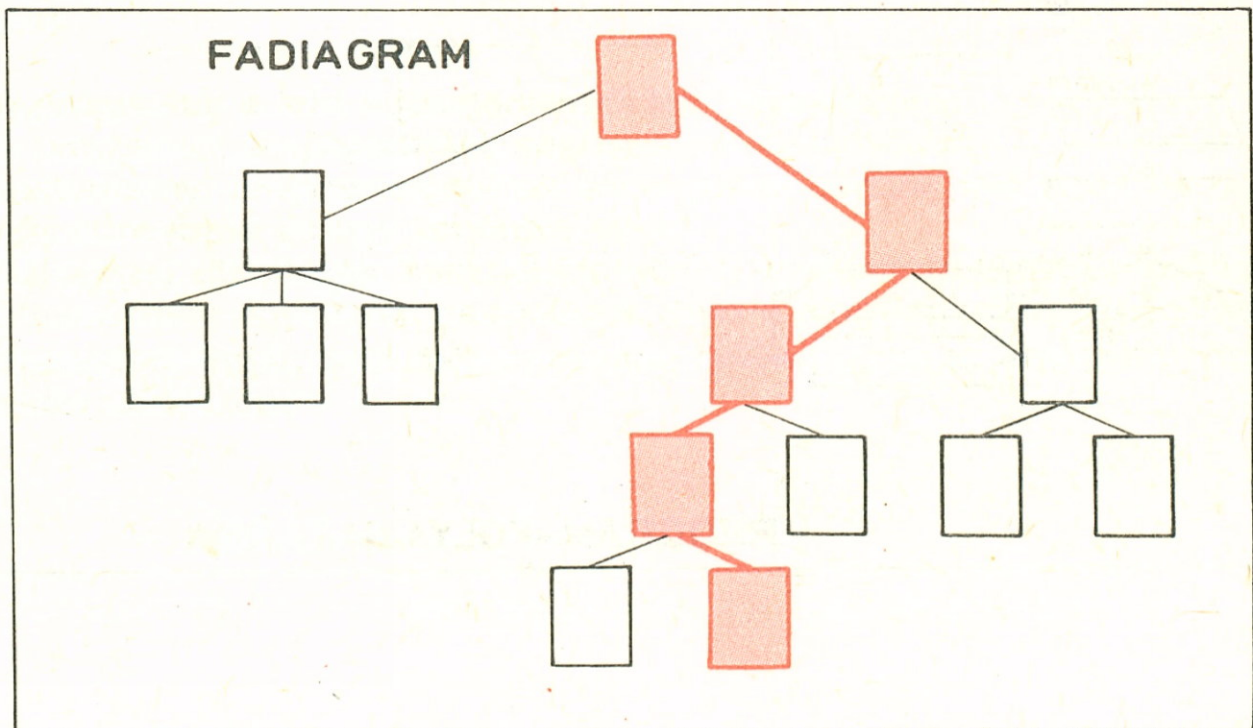


Ebből az öt alapelemből egymás utáni alkalmazással, ill., integrációval több struktúrablokkból összetett struktúrablokkokat lehet képezni. Ezen az úton esetleg igen terjedelmes struktogramok állnak elő, amelyeket azonban az áttekinthetőség érdekében legtöbbször valamilyen szabványos méretre (pl. A4) bontanak fel.

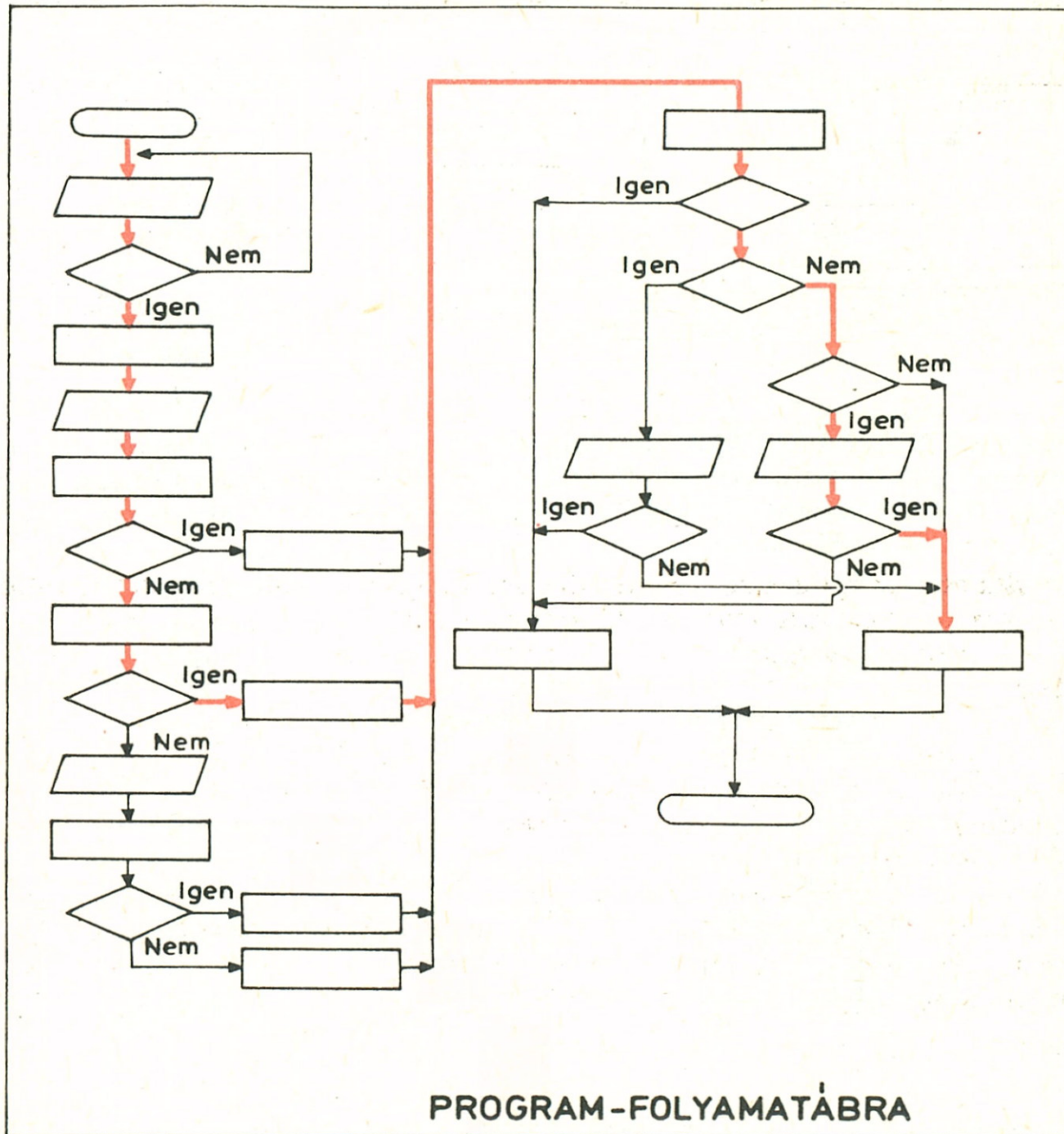
Példa:



Az elkészült programterv több struktogramból áll, amelyek kapcsolatát alá- és fölérendelésekkel szabályozzuk. Ezeket a kapcsolatokat egy **fadiagram** segítségével ábrázoljuk. Ahhoz, hogy a program folyamatát az elejétől a legalsó szinten elhelyezkedő struktogramig követhessük, csak kevés struktogramon kell végigmennünk.

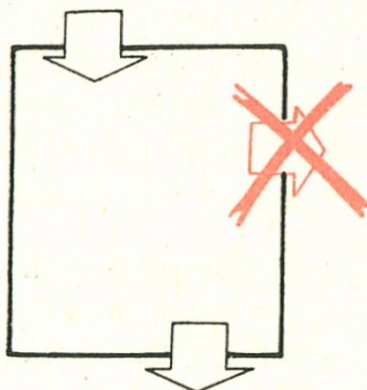


A hagyományos, nem strukturált program-folyamatábránál, amely rendszerint bonyolult háló, a megoldás útját ezzel szemben áttekinthetetlen utakon át az egész folyamatábrán keresztül végig kell követnünk.



Minden struktogramhoz a későbbiekben önálló programmodult lehet hozzárendelni. Mivel valamennyi struktogramnak csak egy bemenete és egy kimenete van, a struktogramok segítségével kifejlesztett programmoduloknak is csak **egy bemenetük és egy kimenetük** van.

Alárendelt elemeket csak a fölérrendelt modulon át szabad megközelíteni!



A strukturált programozás nem kötődik meghatározott programnyelvhez. Az alkalmazott programnyelvtől függetlenül a strukturált programozás alkalmazása a *programozó számára néhány következménnyel* jár. A legfontosabbak:

- A programot a leírt típuselemekből kell összeállítani.
- A programot struktogramok segítségével kell ábrázolni.
- Valamennyi programmodulnak önmagában zártnak kell lennie, és csupán egy bemenete és egy kimenete lehet.
- Az ugrási utasítások használata erősen korlátozott. Ugrási utasításokat, amennyiben egyáltalán használunk, csak a programmodulon belül szabad használni. Egyik modulból a másikba történő ugrás az előre adott fastruktúrán kívül tilos!

A strukturált programozás előnyei:

- A felülről lefelé fejlesztés által kisszámú típuselemre korlátozódik a programok ábrázolása. A struktogramok alkalmazásával és az ugrási utasítások kizárásával a *programok áttekinthetőbbé válnak, könnyebben olvashatók és könnyebben javíthatók*, mint a hagyományos úton előállított programok. Egy, a strukturált programozás módszerével készített program kívülálló számára is könnyebben megérthető, mivel a program fejlesztőjének rendszerint bonyolult gondolkodásmódját nem kell rekonstruálni.
- Egy *struktúrablokk módosítása vagy kicserélése* viszonylag könnyen végrehajtható, mivel csak az alárendelt blokkhoz való kapcsolatot, valamint a fölérendelt blokk kapcsolatát kell figyelembe venni.
- A programok tesztelése egyszerűbb, mivel az egyes programmodulokat a teljes programtól elkülönítve lehet kezelni. Ezenkívül az egész rendszert a még hiányzó építőelemek helyett alkalmazott fiktív paraméterek (dummy) segítségével korábban lehet tesztelni.
- Ha az ábrázoláshoz struktogramot alkalmazunk, akkor a kevés számú típus programelem alkalmazásával nagy lépést teszünk az érthetőbb dokumentálás irányába.



1. A strukturált programozás alkalmazása esetén a kitűzött feladat megoldását a nagyvonalakban vázolttól a részletes megoldás felé haladva lépésenként hajtjuk végre.

Ezt az eljárást nevezzük

2. Hogy nevezzük azt az ábrázolási segédeszközt, amely a programozót strukturált programozásra kényszeríti?

.....

3. A strukturált programozásra jellemzőek az alábbiak:

Ugrási utasítások nincsenek megengedve.

Struktúrablokkokon belül az ugrási utasítások megengedettek.

A struktúrablokkok közötti ugrások az előre megadott fastruktúra keretein belül megengedettek.

4. A struktogramoknak mindig csak:

Egy bemenete lehet.

Több bemenete lehet.

Egy kimenete lehet.

Több kimenete lehet.

5. Mik a strukturált programozás előnyei?

A programok áttekinthetőbbek, jobban olvashatók és könnyebben változtathatók.

A programok tesztelése egyszerűbb.

Hozzásegít az érthetőbb dokumentációhoz.

A programozó szabadságát semmilyen szempontból nem korlátozza.



Válaszok

1. A strukturált programozás alkalmazása esetén a kitűzött feladat megoldását a nagyvonalútól a részletes megoldás irányába fejlesztik. Ezt az eljárást **felülről lefelé** (top-down) **fejlesztésnek** nevezik.
2. Azt a programábrázolási segédeszközt, amely a programozót strukturált programozásra kényszeríti, **struktogramnak** nevezzük.
3. A strukturált programozás esetén:
 - A korszerű, magas szintű programnyelvek lehetővé teszik, hogy teljesen ugrási utasítás nélküli programokat írjunk.*
 - Struktúrablokkokon belül az ugrási utasítások megengedettek.
 - A struktúrablokkok közötti ugrások az előre megadott fastruktúra keretein belül megengedettek.
4. A struktogramoknak:
 - Egy bemenete lehet.
 -
 - Egy kimenete lehet.
 -
5. A strukturált programozással a következő előnyökhöz jutunk:
 - A programok áttekinthetőbbek, jobban olvashatók és könnyebben változtathatók.
 - A programok tesztelése egyszerűbb.
 - Hozzásegít az érthetőbb dokumentációhoz.
 -

Petri-hálók

A problémaelemzés és az eljárásstervezés további segédeszköze a **Petri-háló**. Petri-hálók segítségével az eljárásokat modellként lehet ábrázolni és azokat egyszerűen szimulálni. Mindenekelőtt akkor alkalmazhatók, ha érvényes az, hogy az egyes események közötti függőségeket akarjuk ábrázolni.

A Petri-hálót csupán az alábbi **három alapjelképből** állítják össze:

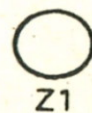
Az eseményeket *vastag vonalakkal* (esetleg négyszögekkel) ábrázoljuk.

Az események bekövetkezését kiváltó állapotokat, ill. feltételeket *körökkel* jelképezzük.

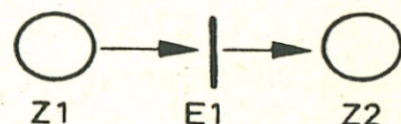
Az egymás után következő köröket és vastag vonalakat *nyilakkal* kötjük össze.



E1



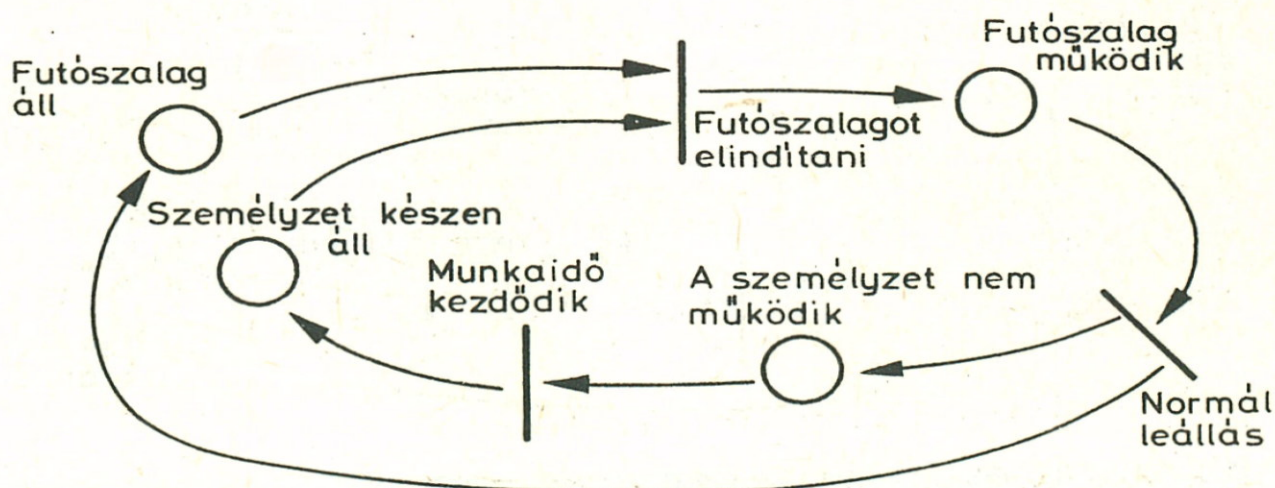
Z1



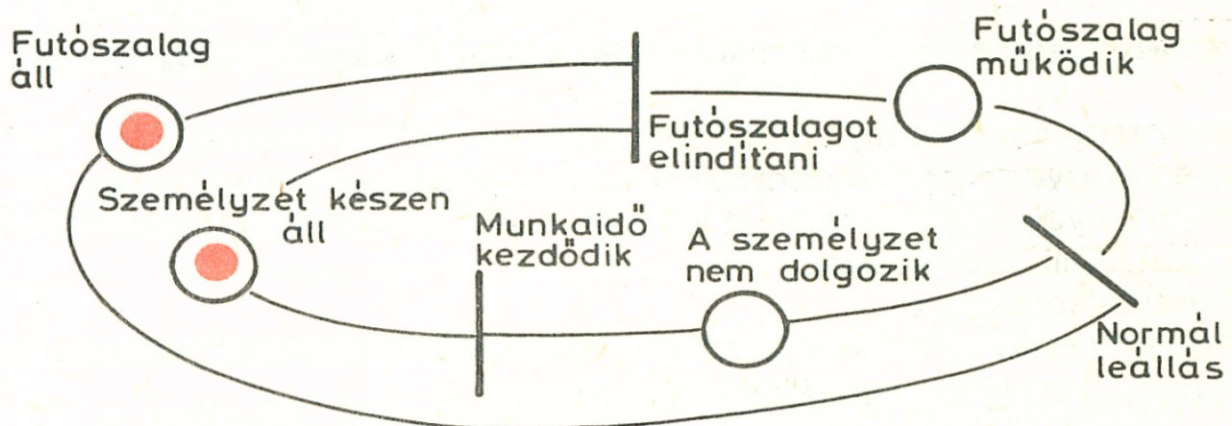
Z1 E1 Z2

Példa:

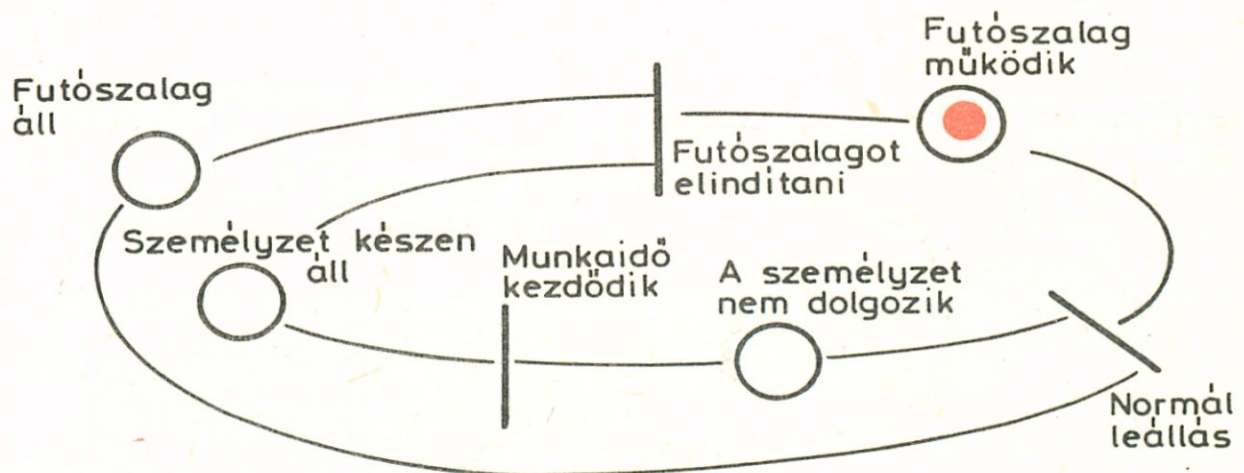
Egy *futószalag üzemét* (közelítőleg) Petri-hálóval ábrázoljuk. A körök azokat az állapotokat mutatják, amelyeket a futószalag felvehet, a vastag vonalak pedig azokat az eseményeket jelentik, amelyek állapotváltozást válthatnak ki.



Ahhoz, hogy a Petri-háló által ábrázolt eljárást „lejátszhassuk”, az aktuális állapotnak megfelelő körökbe jelzésként pontokat rajzolunk. A gyakorlatban a körökre rendszerint valamilyen jelzót vagy kis érmét (esetleg mágneses lapocskát) helyeznek rá. Ezek alapján egy munkanap kezdetét jelentő *kezdeti állapotot* az alábbi módon ábrázolunk:



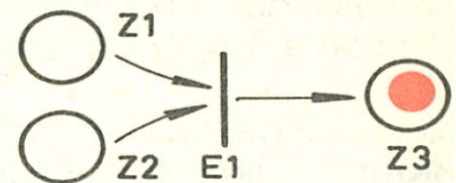
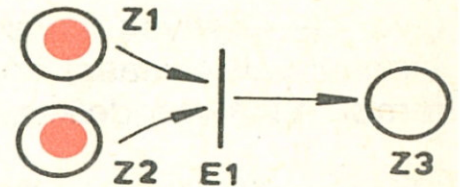
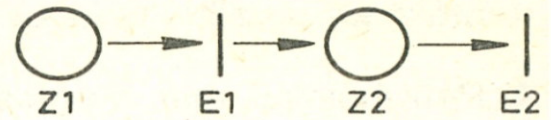
Az ábrázolt folyamat további szimulációjának érdekében a használt jelzéseket (körökbe írt pontokat) körről körre, vagyis állapotról állapotra a nyilak mentén továbbléptetjük. A fenti hálón kezdetben két kör van megjelölve: a futószalag áll, és a személyzet munkára kész. Ezek azok a **feltételek**, amelyeknek teljesülniük kell, mielőtt a következő esemény: „a futószalag indítása” bekövetkezhet. Ha ez az esemény bekövetkezik, akkor a Petri-háló a következő lesz:



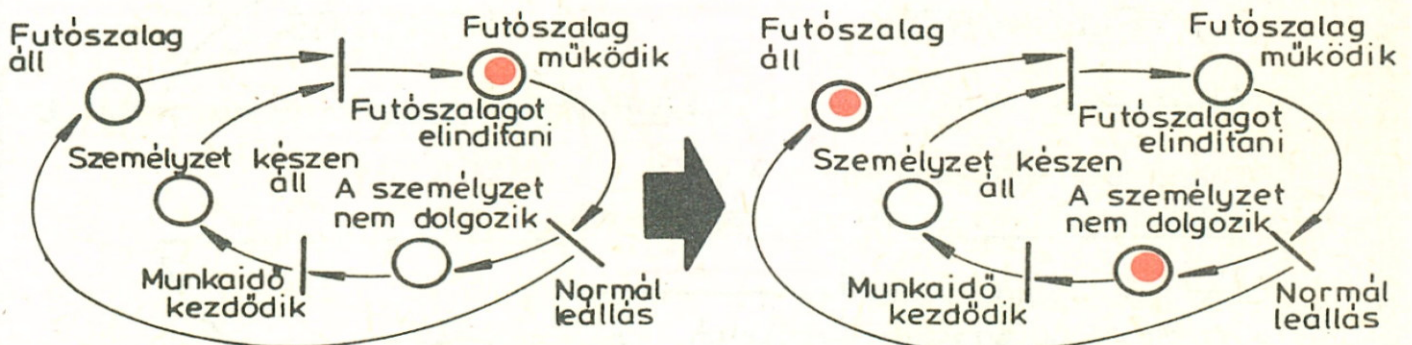
A „futószalag áll” és a „személyzet munkára kész” állapotok itt már nincsenek megjelölve. Most viszont a „futószalag működik” esemény van megjelölve, mivel a szalag megy. Ahhoz, hogy ezt az állapotot ábrázoljuk, csak egy jelzésre van szükség. A második jelzés ekkor nem szükséges. Ha ezzel szemben pl. egy feltétel teljesüléséhez három állapotnak kellene bekövetkeznie, három jelzésre lenne szükség.

Ebből a példából **három szabály** vált világossá:

1. Egy kört sosem szabad körrel és egy vastag vonalat sem szabad vastag vonallal összekötni!
Az állapotok és az események tehát egymást mindig felváltva követik.
2. Egy esemény csak akkor következhet be, ha *valamennyi* bemeneti feltétel igaz — tehát valamennyi előtte levő kör meg van jelölve.
3. Ha egy esemény bekövetkezik, akkor a bemeneti körök jelölését meg kell szüntetni: azok ezután mindig a következő körre vagy körökre kerülnek át.
Ez azt jelenti, hogy a kimeneti állapot bekövetkezett.



Tekintve, hogy jelenleg a „futószalag működik” állapot van megjelölve, most ez az állapot érvényes. A „normál leállítás” állapot akkor következhet be, ha igaz, hogy munkaszünet kezdődik, vagy ha vége van a munkanapnak. A „futószalag üzemben” állapotról minden esetre eltávolítjuk a jelzést, és egy-egy jelet helyezünk el a „futószalag áll” és a „személyzet nem dolgozik” következő állapotokra.



A Petri-háló fenti jelölési állapota a munkanap kezdetét is ábrázolja, röviddel a munkaidő megkezdése előtt.

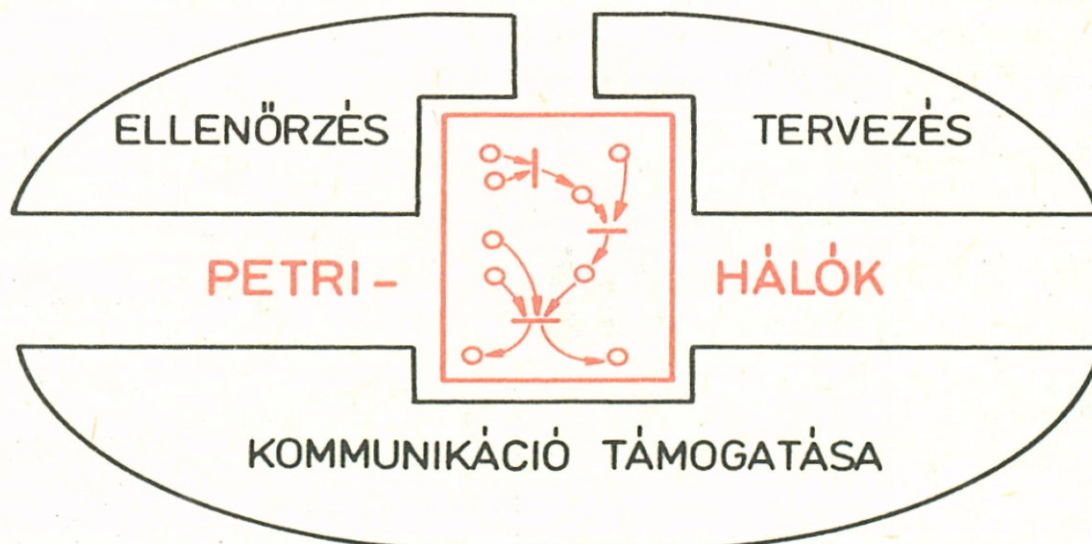
A Petri-háló alkalmazási lehetőségeit, és ezek közül egy különösen fontosat: egy rendszer szimulációját Petri-hálóval, az előbbieken csak korlátozott mértékben mutathattuk be.

A Petri-hálókat már **meglevő folyamatok ellenőrzésére** is felhasználják, és pedig annak megvizsgálására, hogy a rendszer valamennyi állapota bekövetkezik-e, és vannak-e olyan helyek, amelyek ellentmondásosak, vagy olyanok, amelyek nincsenek kellőképpen meghatározva. Egy folyamat valamennyi változtatását íróasztal mellett vagy ha megfelelő számítógépes programok állnak rendelkezésre, akkor terminál mellett szimulálhatjuk.

Másrésről a Petri-hálót fel lehet használni **új eljárások tervezésének támogatására**. Az állapotok és események állandó változtatása arra kényszeríti a fejlesztőt, hogy pontosan meggondolja, mely feltételeket kell kielégíteni, mielőtt a megkívánt esemény bekövetkezhet.

A felsoroltakon kívül a Petri-háló **a megbízó és a rendszerfejlesztők közötti kommunikációt** is megkönnyítheti, azért, mert a Petri-háló mindössze néhány szimbólum alkalmazásával ábrázolja a folyamatot, és így az nem szakemberek számára is érthető.

A Petri-háló olvasása egyszerű! A háló felhasználása azonban némi gyakorlatot igényel. E rövid terjedelemben ezt a módszert csak bemutatni tudjuk; a témában való elmélyüléshez a vonatkozó szakirodalom olvasását ajánljuk.





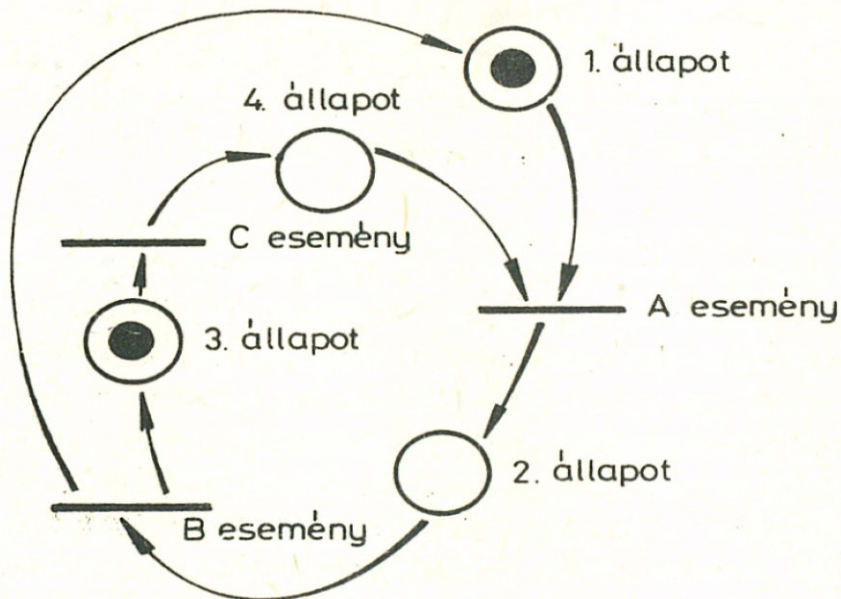
1. Nevezze meg azokat az alapszimbólumokat, amelyekből a Petri-hálók felépülnek! Az eseményeket

..... vagy ábrázoljuk.

Az állapotokat, ill. feltételeket jelképezik.

Ezek a jelképek egymást váltva szerepelnek és
..... vannak összekötve.

2. A jelölt aktuális állapot után melyik esemény következhet be?



- A esemény.
- B esemény.
- C esemény.

3. Mely állítások igazak a Petri-hálókra?

A Petri-háló a problémaelemzés és az eljárás tervezés segédeszközei.

A Petri-hálóban két kört mindig nyíllal kell összekötni.

A Petri-háló új eljárások tervezése és meglévő eljárások ellenőrzése esetén használhatók segédeszközként.



Válaszok

1. Az eseményeket *vastag* vonalakkal vagy *négyszögekkel* ábrázoljuk.

Az állapotokat, ill. a feltételeket *körök* jelképezik.

Ezek a jelképek egymást váltva szerepelnek és *nyilakkal* vannak összekötve.

2. *Csak a C esemény következhet be, mert ennek bemeneti állapota van jelölve. Az A eseménynek csak egyik bemeneti állapota jelölt, ezért nem következhet be.*

C esemény

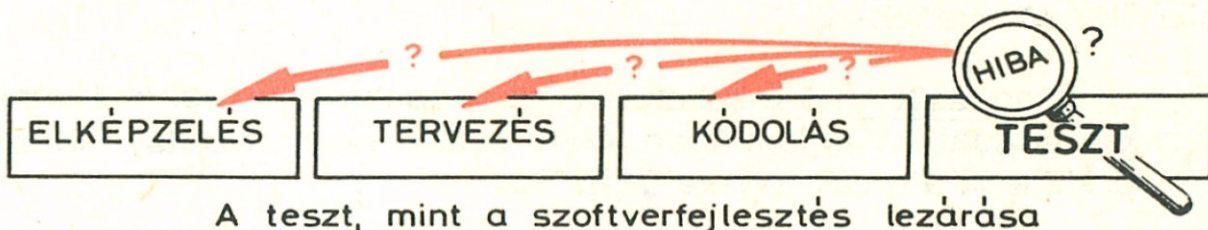
3. A Petri-hálókra a következő állítások igazak:

A Petri-hálók a problémaelemzés és az eljárástervezés segédeszközei.

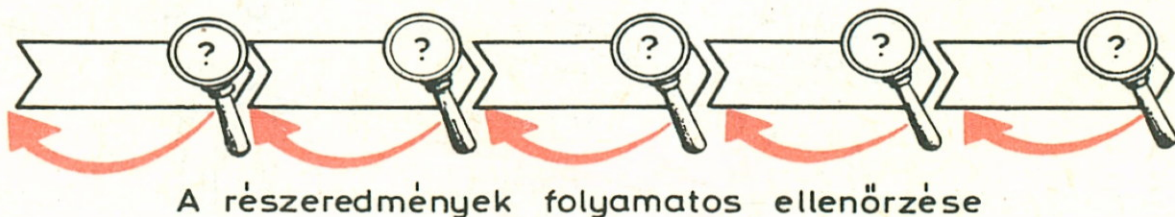
A Petri-hálók új eljárások tervezése és meglévő eljárások ellenőrzése esetén használhatók segédeszközként.

Ellenőrzési módszerek

Abban az időben, amikor még „szoftvertechnika” helyett „programozásról” beszéltek, a programfejlesztés során a következőkben leírásra kerülő eljárást használták. A programot megtervezték, kódolták, majd ezután tesztelték. A folyamat végén végrehajtott **programteszt** azt az alapvető hátrányt jelentette, hogy bizonyos körülmények között csak a fejlesztés végén derült ki, hogy a program — vagy sokkal inkább az egész rendszer — nem felel meg az elvárásoknak. Az, hogy a hiba a tervezés vagy a kódolás során történt-e vagy hogy összetett hibáról van-e szó, nagy fáradság árán volt csak kideríthető.



Jelentős ráfordítást igényelt egy olyan hiba elhárítása, amelyet a rendszer tervezése során követek el, azonban csak az átadási folyamat során derült ki. Ezért ma arra törekednek, hogy *minden munkaszakaszban* megvizsgálják: a mindenkor végrehajtott munka nem rejt-e magában hibát? Ezáltal messze-mentően meg lehet azt akadályozni, hogy egy hibát további munkaszakaszokra átvigyünk. Egy részletes számítástechnikai rendszertervet, amelyben pl. még logikai hibák találhatók, időpocsékolás lenne programozni. Az egyes munkaszakaszok folyamatos ellenőrzése által előállított résztermékek (szakmai rendszerterv, számítástechnikai rendszerterv, kódolás) hibái esetén azokat azonnal meg lehet szüntetni és el lehet kerülni, hogy a további szakaszokra káros hatással legyenek.



Általában érvényes a következő:

Minél előbb sikerül egy hibát felfedezni, annál kisebb ráfordítással lehet azt elhárítani.

A folyamatos ellenőrzésre két sajátos **ellenőrzési módszert** alakítottak ki, ezek a következők:

A **walk-through (sétálj végig) módszert** és az **inspekciós technikát.**

A walk-through módszer

A **walk-through** módszer, amelynél az írásos anyagokat és a munkák eredményét egy munkatársakból álló csoport úgy ellenőrzi, hogy még egyszer átvizsgálja őket: helyesek-e?

A walk-through-ban részt vevő kollégáknak olyan szakembereknek kell lenniük, akik az adott feladat megoldásában nem vettek részt, ezáltal előítéletmentesen végezhetik a hibák felkutatását.

Mielőtt a kollégák egy walk-through-megbeszéléshez összeülnek a fejlesztőcsoporttal, gondosan át kell tanulmányozniuk az anyagokat. A beszélgetés során kérdéseket tesznek fel a fejlesztőkhöz, ez a hibák felfedéséhez vezethet. A csoport feladata csak az, hogy a hibákat felderítse; a megoldásokat nem kell kidolgozniuk.

A walk-through módszert az összes írásos eredmények ellenőrzésére fel lehet használni. Ezzel a módszerrel a szakmai rendszertervet, a számítástechnikai rendszertervet, a kódolt programot vagy a dokumentációt egyaránt ellenőrizni lehet.

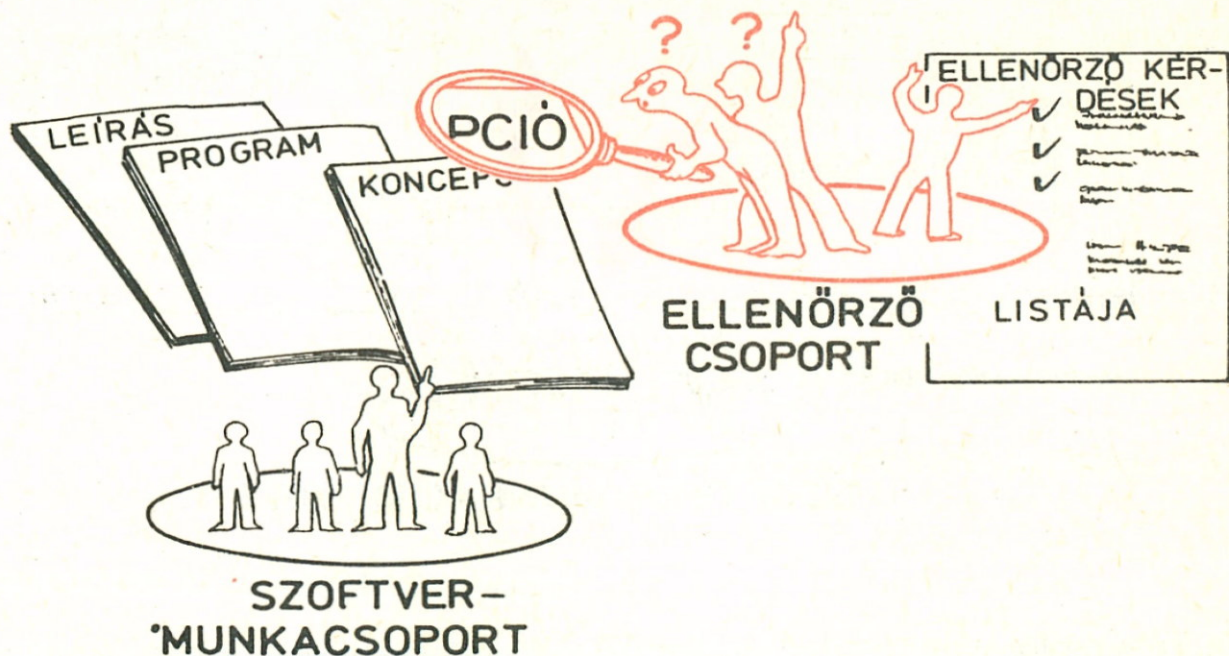


Az inspekciós technika

Az **inspekciós technika** — hasonlóan a walk-through módszerhez — olyan módszer, amelynek segítségével a kódolt programot vagy rendszertervet ellenőrizni lehet, hogy nem tartalmaz-e hibákat. Mindenesetre az inspekciónak lényegesen szigorúbb ellenőrzést jelent, mint a walk-through, amely kollégialis formában végrehajtott beszélgetés keretében zajlik le.

Az inspekciónak ezzel szemben egy speciális szakembercsoport végzi, akik azt a megbízást kapják, hogy az anyagokban levő hibákat felderítsék.

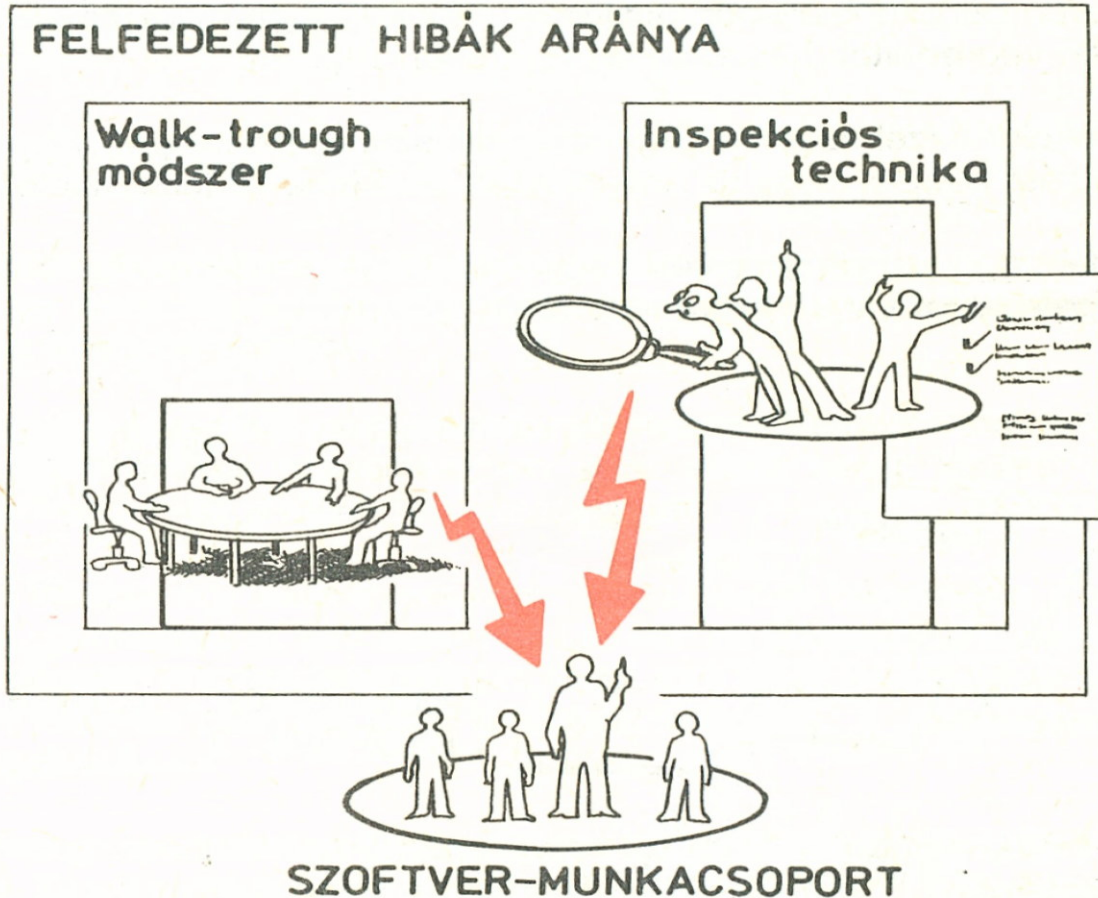
Aszerint, hogy rendszertervet vagy már kódolt programot ellenőriznek, **tervinspekcióról** vagy **kódinspekcióról** beszélünk.



Mindkét esetben az ellenőrök adott listák alapján céltudatosan keresik a hibákat. Az ellenőrzőlista pontjai szerint ellenőrizhetik pl., hogy: „valamennyi programhurok megfelelően van-e befejezve?”, vagy „miként gondoskodtak arról az esetről, amikor a számítógép kiesik?”.

Egy ilyen ellenőrzés befejezése után az ellenőrök jelentést készítenek, amely egyidejűleg az elvégzett munka értékelésének egy módja is.

Az inspekciós technika — ugyanúgy, mint a walk-through módszer — a szoftverfejlesztés valamennyi szakaszában alkalmazható. Az inspekciós technika segítségével a tapasztalatok szerint több hibát lehet felfedezni, mint a walk-through módszerrel.



Ha az inspekciót valamennyi munkaszakasz után következetesen végrehajtjuk, számolhatunk azzal, hogy a hibák nagy részét még a számítógépes futási teszt előtt ki tudjuk javítani.

Az itt leírt ellenőrzési módszerek mellett továbbiak is léteznek, amelyek lényegesen csökkentik a futásra kész programok teszteléséhez szükséges gépidőt.



1. A szoftverfejlesztés melyik szakaszaiban kell az eredményeket ellenőrizni?

Javaslati szakasz.

Tervezési szakasz.

Megvalósítási szakasz.

2. Az inspekciós és a walk-through módszer közül Ön melyiket alkalmazná, ha egy elkészült repülésirányító programot kellene ellenőriznie?

3. Az ellenőrzőcsoportoknak mi a feladatuk?

A hibák felderítése.

A hibák elhárítása.

A megoldások kidolgozása.

4. Egészítse ki az alábbi mondatot:

A projekt valamennyi szakaszában az anyagok és az eredmények állandó ellenőrzése az elkészített programok teszteléséhez szükséges gépidőt:

Csökkenti.

Növeli.



Válaszok.

1. Az eredmények ellenőrzését valamennyi munkaszakaszban végre *kell* hajtani!
 - Javaslati szakasz.
 - Tervezési szakasz.
 - Megvalósítási szakasz.

2. Egy repülésirányító program ellenőrzéséhez az **inspekciós módszert** kellene alkalmazni, mert ezen a területen a hibamentességnek a legnagyobb jelentősége van.

3. Az ellenőrzőcsoportok egyedüli feladata a
 - hibák felderítése.
 -
 -

4. A mondat helyesen így hangzik:

A projekt valamennyi szakszában az anyagok és eredmények állandó ellenőrzése az elkészített programok teszteléséhez szükséges gépidőt:

 - Csökkenti.

Automatizált segédeszközök

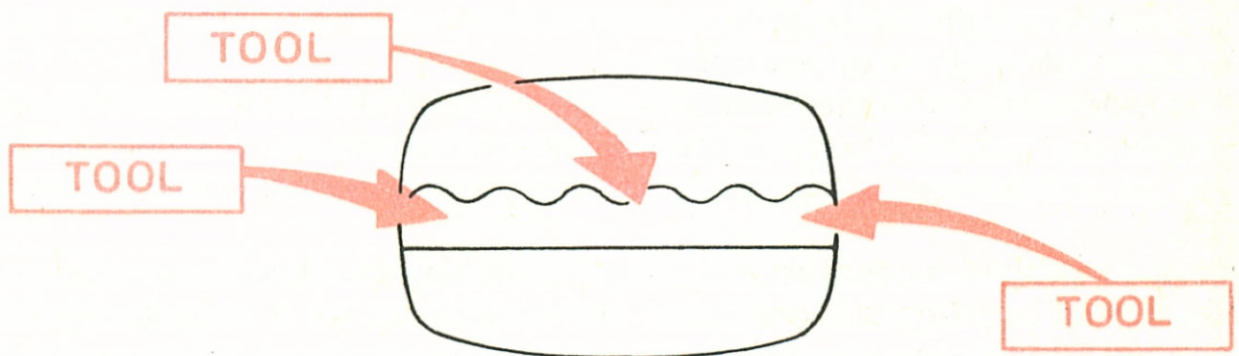
A kézműves munkákhoz szerszámokra, pl. ásóra, kalapácsra, vagy gépekre van szükség, amelyek a munka elvégzését ésszerűsítik és könnyebbé teszik.

Kézenfekvő, hogy a szoftverfejlesztéshez is szerszámokat alkossunk, amelyek a szoftver készítőit a fejlesztői munkában támogatják.

A **szoftverfejlesztés szerszámai**: a számítógépes programok, vagyis olyan automatizált segédeszközök, amelyeket a számítógép saját maga kínál. A szerszám vagy eszköz az angol **tool** szó — amelyet a számítógépes szakemberek jobban ismernek — fordítása.

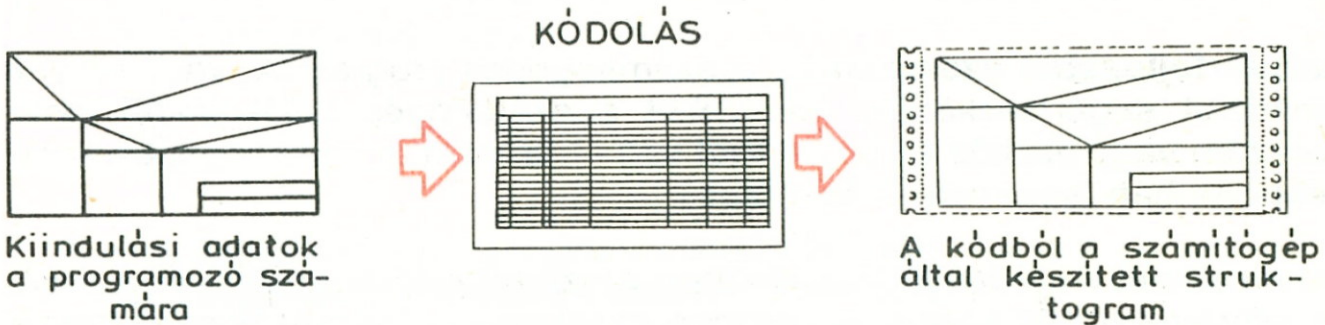
Az automatizált segédeszközök részben a számítógépek szállítóitól a rendszerszoftverrel együtt kaphatók, részben azonban az alkalmazók saját maguk fejlesztik ki őket.

A szoftverfejlesztés valamennyi területén vannak ma már segédeszközök: a projekttervezés, a programtervezés, a programozás, a programok tesztelése és dokumentálása területén.



Itt az eszközök széles választékából csak egy néhányat ismertetünk, azért, hogy megmutassuk, hogyan segíthetnek a szoftverkészítőknek.

Vannak pl. eszközök a program-folyamatábrák megrajzolásához, amelyek lehetővé teszik, hogy egy programozási nyelven megírt programhoz utólag program-folyamatábrát vagy struktogramot nyomtassanak. Az ilyen eszközöket pl. programok dokumentálásához alkalmazzák. Ezekkel a programok logikája is ellenőrizhető, ui. ha a számítógép a kódolt utasítások alapján ugyanazokat a struktogramokat rajzolja ki, amelyeknek alapján a kódolást végrehajtották, akkor feltételezhető, hogy a logikát helyesen ültették át az utasítások nyelvére.



Egész sor további segédeszköz létezik még, amely segítséget nyújt a dokumentáció elkészítéséhez. Segítséget nyújtanak a szövegek megfogalmazásában és nyomdakész állapotba hozásában azáltal, hogy pl. valamennyi szövegoldalt egységes kerettel, fejléccel állítanak elő, elkészítik az oldalszámozást, végrehajtják a sorkiegyenlítést, tagolják és megszámozzák a feliratokat. Más eszközök ellenőrzési listákat generálnak, amelyek biztosítják, hogy a dokumentációból semmi se maradjon ki, és a dokumentáció a megfelelő időben készüljön el. A programfejlesztőnek eközben válaszolnia kell a segédprogramok által feltett kérdésekre.

Vannak továbbá olyan eszközök, amelyek a projektvezetés munkáját támogatják oly módon, hogy a projekt előrehaladásának pontos adatait rögzítik, így pl. a fejlesztés állapotát, a szükséges gépidőket, a hálótervek betartását stb.

Más eszközök a tesztadatok előállítását segítik. A felhasználónak pl. a legkisebb és legnagyobb értékeket kell megadnia, és a segédprogram önállóan állít elő tetszőleges mennyiségű közbenső tesztadatot. Ezeket az ellenőrzendő program tesztfutása alkalmával feldolgozza. Az ilyen eszközöknek különösen akkor van jelentőségük, ha nagy mennyiségű tesztadatra van szükség.

A fejlődés abba az irányba mutat, hogy mindig tökéletesebb eszközöket fejlesztenek ki, amelyek a szoftverfejlesztőket a programok előállításában még jobban segítik és nagymértékben tehermentesítik őket a rutinmunkáktól.



1. Mik a szoftverfejlesztés segédeszközei?
Jelölje meg a helyes válaszokat!

Terminálok.

Programok.

Folyamatábrák.

2. A szoftverfejlesztés mely szakaszaiban lehet
segédeszközöket igénybe venni?

A projekt tervezésekor.

A program tervezésekor.

A programozáskor.

A programok tesztelésekor.

A dokumentáláskor.

3. Mi a különbség a szoftverfejlesztés módszerei és segédeszközei
között?

.....

.....



Válaszok

1. A szoftverfejlesztés segédeszközei:

Programok.

2. A szoftverfejlesztés valamennyi szakaszában lehet segédeszközöket igénybe venni.

A projekt tervezésekor.

A program tervezésekor.

A programozáskor.

A programok tesztelésekor.

A dokumentáláskor.

3. A módszerek

**a programok elkészítésénél használatos eljárások;
a segédeszközök ezzel szemben programok,
vagyis automatizált segédeszközök.**

Programnyelvek

Programnyelvek nélkül a programozás nagyon fáradságos dolog lenne, mert a számítógép elvileg csak a „nincs áram” (0) és a „van áram” (1) információkat érti meg. Ezen információk egymásutániségából különböző információkat képezhetünk, és ezekhez mindenféle jelentést rendelhetünk hozzá. Pl. a nullák és egyesek itt látható sorozata 0001 1010 0011 1110 egy összeadási utasítást jelenthet.

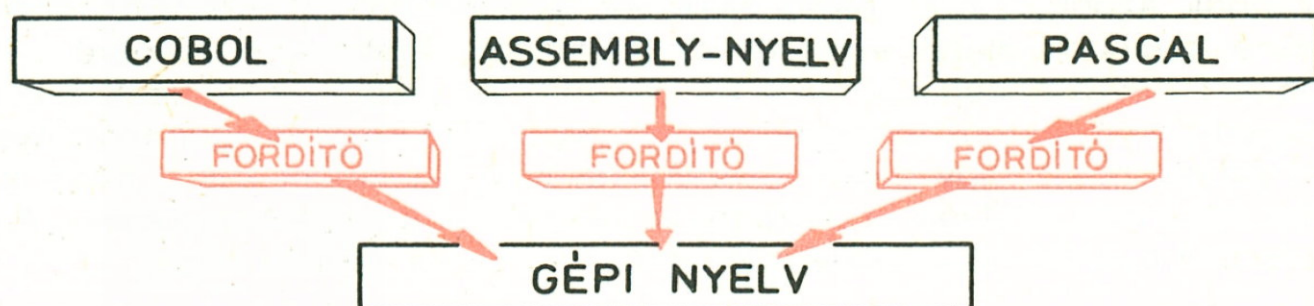
Ha egy program sok száz utasítását egyesek és nullák egymás után írásával kellene elkészíteni, ez nagyon fáradságos és gazdaságtalan lenne. A számítógép viszont kizárólag olyan utasításokat ért meg, amelyek egyesekből és nullákból állnak. Az utasítás egyik része az utasítás fajtáját, pl. „add össze”, a másik rész pedig az összeadandó operandusok operatív tárbeli címét jelenti.

A programozók munkájának megkönnyítésére a számítógép sajátosságait is figyelembe véve kifejlesztették a **programnyelveket**, amelyek hidat teremtenek a számok világa és az emberi nyelv között.

Az említett — egyesekből és nullákból álló — összeadási utasítás egy programnyelvben a következő lenne:

ADD ÉRTÉK TO RÉSZÖSSZEG

Ez az utasítás az ember számára már szinte teljesen magától értetődő: „add hozzá az ÉRTÉK-et a RÉSZÖSSZEG-hez”. Ahhoz azonban, hogy a gép megértse, hogy miről van szó, az utasítást először le kell fordítani a számítógép saját nyelvére, azaz le kell bontani egyesekre és nullákra. Ezt a munkát végzi el a számítógép a fordítás során. A számítógép tehát lefordítja az utasításokat a programnyelvről a számára érthető gépi nyelvre. Ezt a feladatot a következő fejezetben részletesen taglaljuk.



Most először a programnyelvek különböző fajtáival és ezek jellemzőivel foglalkozunk.

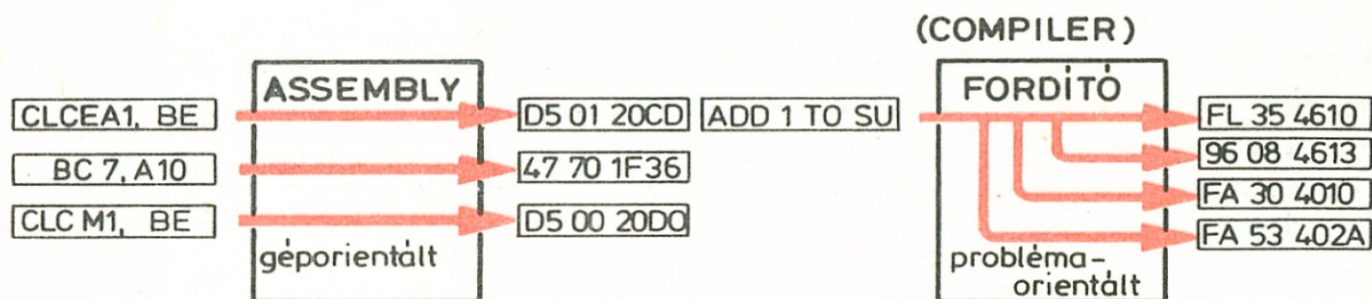
A programnyelveket két osztályba sorolhatjuk:

géporientált és **problémaorientált** nyelvekre.

A **géporientált** programnyelvek egy bizonyos számítógép gépi nyelvéhez igazodnak. Az ezeken a nyelveken írott programokat más típusú számítógépen rendszerint nem lehet futtatni.

A **problémaorientált** nyelveket ezzel szemben minden géptípuson lehet használni, amelyre az adott nyelvhez fordítóprogram létezik.

További különbség: egy géporientált programnyelv minden utasításának a fordítás után rendszerint egy gépi utasítás felel meg. Ezzel szemben a problémaorientált nyelvek minden utasítása több gépi utasítást hoz létre.



Géporientált programnyelvek

A géporientált programnyelvek az adott számítógép valamennyi lehetőségének kihasználását lehetővé teszik, mert a programnyelv minden utasításának meglehetősen pontossággal megfelel az adott számítógép egy gépi utasítása — csupán az utasítások az ember számára jobban olvashatóak. A géporientált nyelveket legtöbbször **assembly nyelveknek** nevezik. Nem helyes általában az assembly nyelvről beszélni, hanem mindig valamelyik géptípus assembly nyelvéről.

A fejlődés ma az assembly nyelvektől a problémaorientált nyelvek felé halad. Vannak azonban még olyan területek, amelyeken az assembly nyelveket nehézkes használatuk és körülményes megtanulhatóságuk ellenére is használják: nevezetesen ott, ahol az operatív tár jobb kihasználása a cél (az assembly programok rendszerint rövidebbek, mint a problémaorientált nyelven írt programok), vagy olyan esetben, ha takarékoskodni kell a futási idővel (az assembly nyelven írt programok rendszerint gyorsabban futnak). A géporientált nyelvek használatára jó példa a rendszerszoftver írása.

Problémaorientált programnyelvek

Amint azt az elnevezés is mutatja, a problémaorientált nyelvek a problémák, feladatok egy meghatározott csoportjához illeszkednek. Így vannak olyan nyelvek, amelyek a műszaki-tudományos területre különösen alkalmasak, mások pedig a gazdasági feladatok megoldására. Vannak szűkebb, lehatárolt témákra, pl. a szerszámgép-vezérlésre kidolgozott nyelvek. Emellett univerzális nyelveket is kidolgoztak, amelyekkel különböző szakterületek feladatait egyaránt meg lehet oldani.

A problémaorientált nyelvekkel a feladatokat nem a gépi utasítások szintjén oldják meg, hanem egy magasabb, az emberi gondolkodáshoz közelített szinten. Ezért nevezik a problémaorientált nyelveket **magas szintű nyelveknek**.

A gyakran használt **problémaorientált nyelvek** a következők:

ALGOL (ALGOrithmic Language), ez volt az első, egzakt módon definiált programnyelv. Különösen matematikai feladatok megoldására alkalmas, mert a program írása közben az aritmetikai kifejezések jól megszokott írásmódját lehet használni.

FORTRAN (FORmula **TRAN**slation), amely szintén egy formalizált nyelv, műszaki—tudományos feladatok megoldásához.

COBOL (Common **B**usiness **O**riented Language), amely különösen a gazdasági területen adódó feladatok megoldására alkalmas. A nyelvet eredetileg az adminisztratív területek feladatainak programozására fejlesztették ki. Nagy adatmennyiségek kezelésére alkalmas elemekkel látták el.

PL/1 (Programming Language **1**), olyan általánosan alkalmazható nyelv, amelyet gazdasági és műszaki—tudományos feladatokra egyaránt fel lehet használni. Többek között az ALGOL, a FORTRAN és COBOL nyelvekből vett át elemeket. Az általános alkalmazhatóság miatt a PL/1 utasításkészlete terjedelmesebb, mint a speciális alkalmazási területek nyelveié.

SPL (System Programming Language), ez olyan nyelv, amelyet — mint a neve is mondja: rendszerszoftver írásához fejlesztettek ki.

BASIC (Beginners All-purpose Symbolic Instruction Code), amely könnyen megtanulható, műszaki-tudományos feladatokra alkalmas nyelv. Ezt a nyelvet különösen a párbeszédéses üzemmódban folyó programozáshoz alkották. Minden, terminálon bevitt utasítást formailag ellenőriznek, az esetleges hibákat a rendszer kijelzi a felhasználónak, aki ezeket azonnal kijavíthatja. Ezenkívül minden egyes bevitt utasítást azonnal le lehet fordítani és végre lehet hajtani.

PASCAL egy korszerű, könnyen megtanulható nyelv, különösen a műszaki-tudományos területre. Rendszerszoftver fejlesztésére is alkalmas. A PASCAL különösen támogatja a strukturált programozás módszerét, és ezért alkalmas a programozás megismertetésére. Gyakran alkalmazzák egyetemeken és iskolákban „oktatónyelvként”.

ADA, amely a PASCAL nyelven alapszik. Fejlesztése során elsősorban a rendszerszoftver-készítésre gondoltak. Az ADA nyelvet túlnyomóan mikroprocesszorral vezérelt rendszereknél használják.

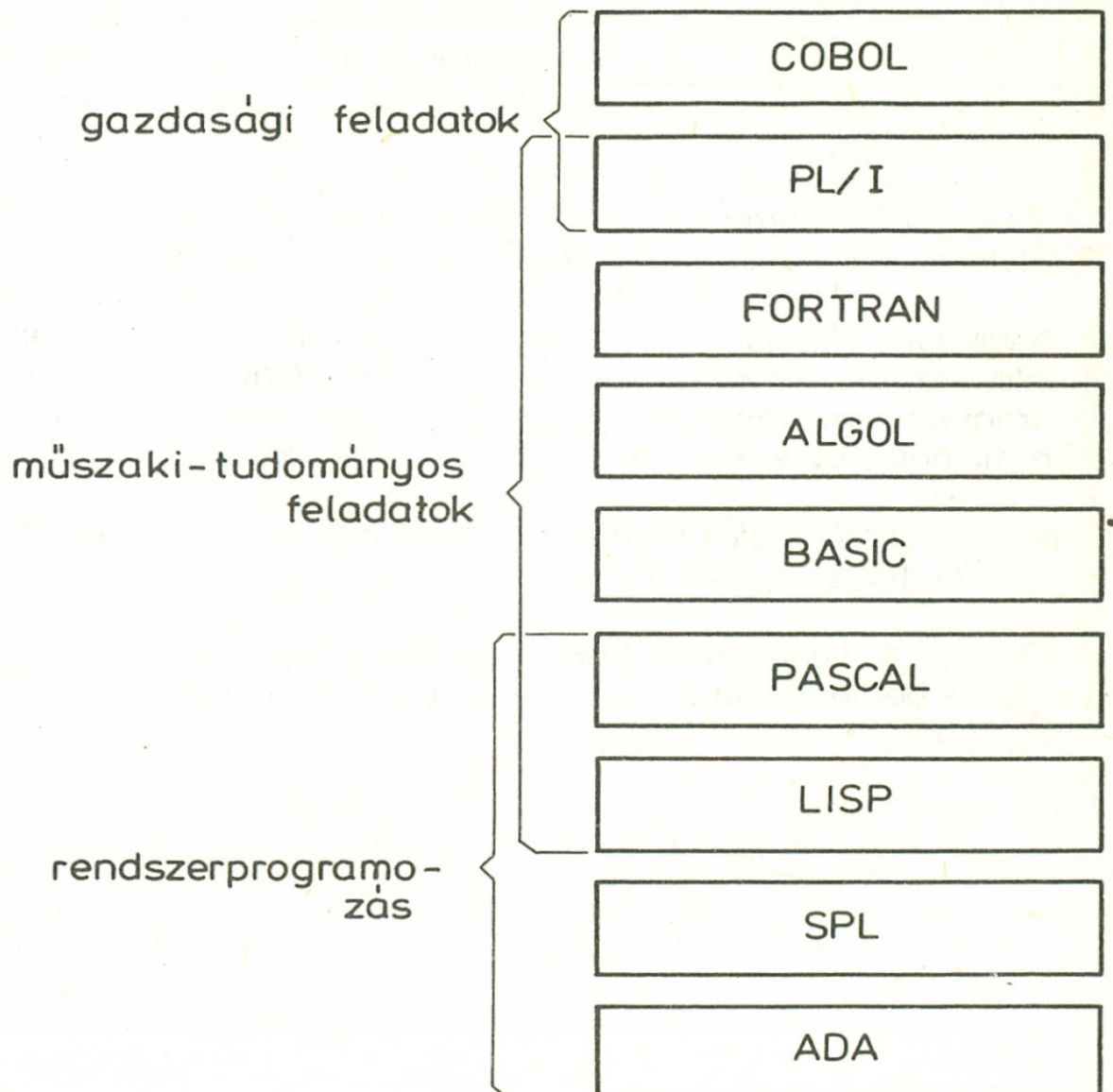
LISP (LIST Processing) elsősorban a műszaki-tudományos terület nagyon összetett feladatainál és a rendszerszoftver-fejlesztésben alkalmazható. A LISP különösen támogatja a párbeszédéses programfejlesztés kreatív folyamatát.

A következő ábra áttekintést ad a fontosabb programnyelvekről, és utal sajátos alkalmazási területükre is.

GÉP-ORIENTÁLT NYELV

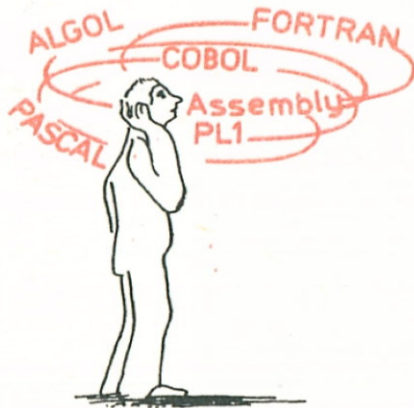
ASSEMBLY

PROBLÉMA-ORIENTÁLT NYELVEK



Azt a kérdést, hogy egy meghatározott feladat legjobb megoldásához melyik programnyelvet kell használni, nem lehet egyszerűen megválaszolni, és e kérdésben sok esetben nem is lehet egyértelműen állást foglalni. A döntést sok tényező befolyásolhatja:

- A feladat fajtája eleve kizár néhány nyelvet. Matematikai feladat megoldásához pl. nem lehet gazdasági orientáltságú COBOL nyelvet használni.



- A rendelkezésre álló programozók ismeretei szintén döntőek. Néhány magas szintű nyelv könnyen megtanulható. Ahhoz, hogy géporientált nyelven tudjunk programozni, megalapozott ismeretek kellene a rendelkezésre álló számítógépről és az operációs rendszerről.

- Ha a szoftvertermékeknek más gépeken is futnia kell (annak más gépekre átvihetőnek, vagyis portabilisnak kell lennie), a géporientált nyelvek kiesnek.
- Nagy jelentőségű a költségek kérdése is. Itt mérlegelni kell egy könnyen kifejleszthető és karbantartható rendszer előnyeit — amit a magas szintű programnyelv használata jelent —, a géporientált nyelv alkalmazása által remélhető, futási időben és tárkihasználásban várható előnyökkel szemben.

A lista, amely a megfelelő programnyelv kiválasztását befolyásoló tényezőket sorolja fel, tetszőlegesen bővíthető.

A szoftver tervezésére és fejlesztésére kidolgozott módszerek és segéd-eszközök bemutatását a ma használatos programnyelvek rövid áttekintésével befejeztük.

A következő fejezetben leírjuk, milyen állomásokon megy végig a program a tervezéstől a hibátlan futásig.



1. Hogyan lehetséges az, hogy egy számítógép, amely csak a „van áram” és „nincs áram” között tud különbséget tenni, megértse egy programnyelv utasításait?

.....

2. Miben különböznek a géporientált programnyelvek a problémaorientált programnyelvektől?
Jelölje meg a helyes válaszokat!

A géporientált programnyelvek egy meghatározott számítógép gépi nyelvéhez igazodnak, a problémaorientált nyelveket ezzel szemben minden számítógéptípuson általánosan lehet használni.

A géporientált nyelv egy utasításából a fordítás során rendszerint csak egy gépi utasítás keletkezik, a problémaorientált nyelv egy utasításából általában több gépi utasítás lesz.

A géporientált programnyelvek utasításai az ember számára jobban érthetők, mint a problémaorientált nyelvek utasításai.

3. Hogyan nevezik még a géporientált programnyelveket?

.....

4. Le tud-e futni egy program minden további nélkül más típusú számítógépen is, amelynek van megfelelő fordítóprogramja?

Géporientált nyelvénél: Igen. Nem.

Problémaorientált nyelvénél: Igen. Nem.

5. Nevezzen meg programnyelveket, amelyek különösen alkalmasak:

Műszaki-tudományos területen

Gazdasági területen

Rendszerszoftver-fejlesztésre

Párbeszédés programozásra



Válaszok

1. A programnyelv utasításait a számítógép lefordítja a gépi nyelvre.

2. A következő állítások helyesek:



A géporientált programnyelvek egy meghatározott számítógép gépi nyelvéhez igazodnak, a problémaorientált nyelveket ezzel szemben minden számítógéptípuson általánosan lehet használni.



Egy géporientált nyelv egy utasításából a fordítás során rendszerint csak egy gépi utasítás keletkezik, a problémaorientált nyelv egy utasításából általában több gépi utasítás lesz.



3. A géporientált programnyelveket assembly nyelveknek is nevezik.

4. Egy program általában csak akkor futthat le más típusú számítógépen, ha a programot magas szintű programnyelven írták.



Nem. Géporientált nyelv esetén.



Igen.

Problémaorientált nyelv esetén.

5.

Műszaki-tudományos területen:

ALGOL, FORTRAN, PL/1, BASIC, PASCAL, LISP.

Gazdasági területen:

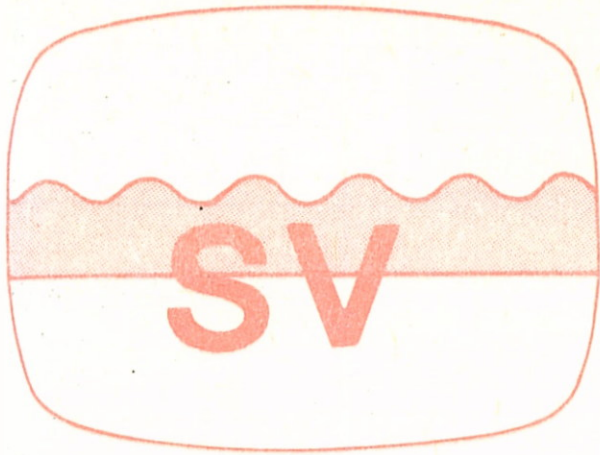
PL/1, COBOL.

Rendszerszoftver-fejlesztésre:

SPL, PASCAL, ADA, LISP.

Párbeszédés programozásra:

BASIC, LISP.



Mi a szoftver?

Hogyan
fejlesztenek
szoftvert?

Mi segítheti
a szoftver-
tervezést
és -fejlesztést?

Mit csinál
a programozó?

Milyen
szolgáltatásai
vannak
az operációs
rendszernek?

Ebben a fejezetben
— tájékozódhat arról,
hogyan kell egy programot
specifikálni ahhoz,
hogyan a programozó hozzáfoghatson
annak fejlesztéséhez;
— megismerkedik a program-
fejlesztés azon lépéseivel,
amelyek a program
logikájának megtervezésétől
a program elkészüléséig vezetnek.

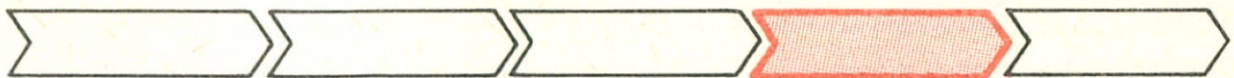
Programok specifikálása

A programfejlesztés lépései

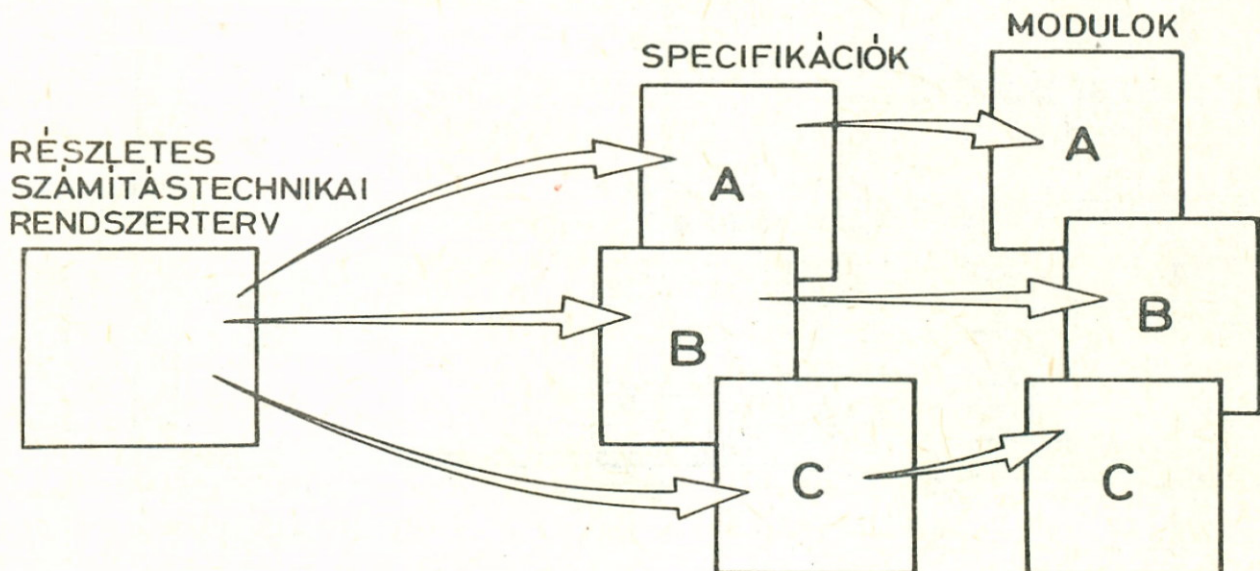
Programok specifikálása

Ha a szoftverfejlesztést a már ismertetett szakaszok alapján vizsgáljuk, akkor a programozást a **megvalósítási szakaszba** sorolhatjuk.

A megvalósítási szakaszban először a részletes számítástechnikai rendszertervet dolgozzák ki, ebből vehetők át az eljárás egyes programjainak **specifikációi**.

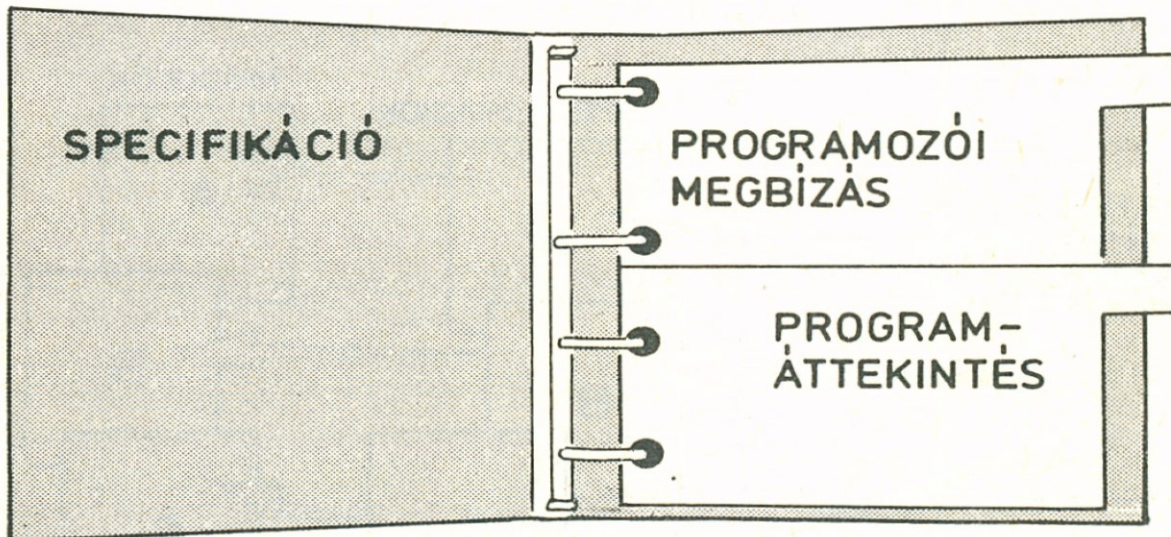


A specifikáció a programozó számára a kiinduló adatokat tartalmazza programjának elkészítéséhez.



A programspecifikáció lényegében a következőket tartalmazza:

- A **feladat leírását**, amely közli az elkészítendő program szolgáltatásait. Ezenkívül itt találjuk meg a megkívánt elkészítési határidőt, a várható költségeket, a személyi és gépi feltételeket, a már rendelkezésre álló és felhasználható egyéb szoftvertermékek felsorolását, a programnyelvet, a módszereket és a segédeszközöket.
- A **programáttekintést**, amely megmutatja a programozónak, hogy az általa készítendő szoftvermodul hol helyezkedik el a teljes rendszerben. Itt definiálják azoknak a moduloknak a csatlakozófelületeit, amelyekkel az elkészítendő modul kapcsolatba fog kerülni, azaz leírják, hogy milyen adatok milyen formában cserélődnek az egyes modulok között. A programáttekintés továbbá megadja a program belső felépítését; pl. azt, hogy az milyen feldolgozási szakaszokból tevődik össze. Lehet, hogy nagy vonalakban elkészített folyamatábrát is mellékelnek hozzá.



■ **Minden adat és adatállomány pontos leírását.**

Valamennyi olyan adatot és állományt leírnak:

- amelyek a bevitelnél szükségesek (mellékelik a felhasznált adatfelvételi bizonylatok mintáját),
- amelyeket tárolni kell, vagy a tárból le kell hívni,
- amelyeket a feldolgozás eredményeként ki kell vinni.

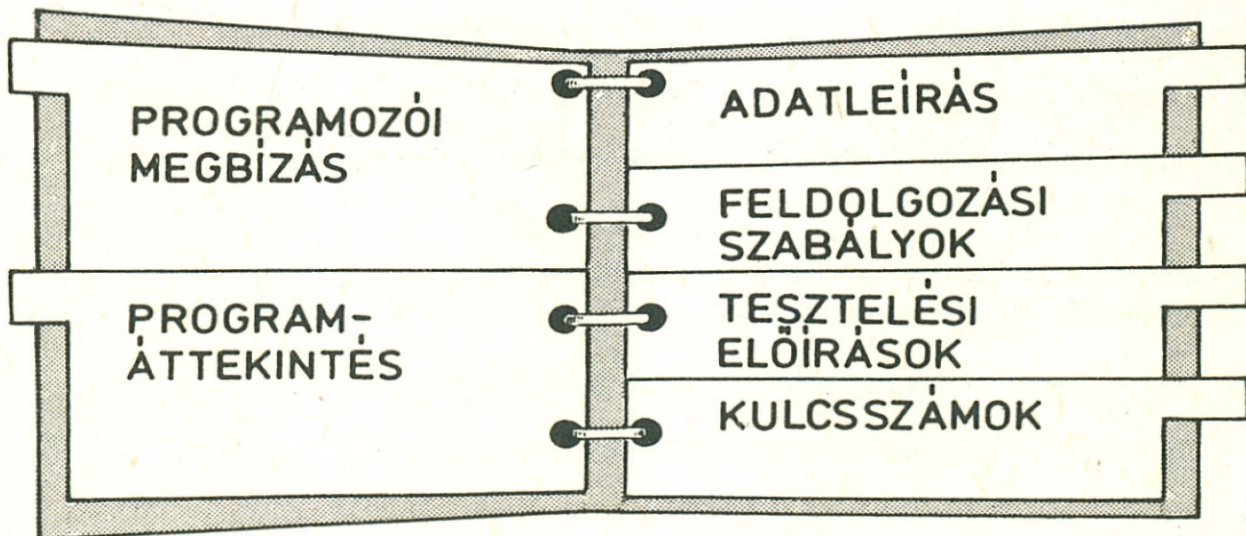
Általában megadják az eredmények nyomtatási képét, s ha lehetséges, előre nyomtatott űrlapokat is (pl. bérkifizetési bizonylat) mellékelnek.

■ Különösen figyelembe kell venni a **feldolgozási szabályokat**, amelyek szerint az adatokat fel kell dolgozni. Ezek pl. az adatok helyességének, hihetőségének, megengedhetőségének ellenőrzésére terjednek ki, vagy útmutatást adnak arra, hogy mit kell tenni hibás adatok esetén.

■ **Tesztelési útmutatásokat**, amelyek megmondják, milyen stratégiával kell a programot tesztelni és milyen tesztadatokat kell ehhez használni.

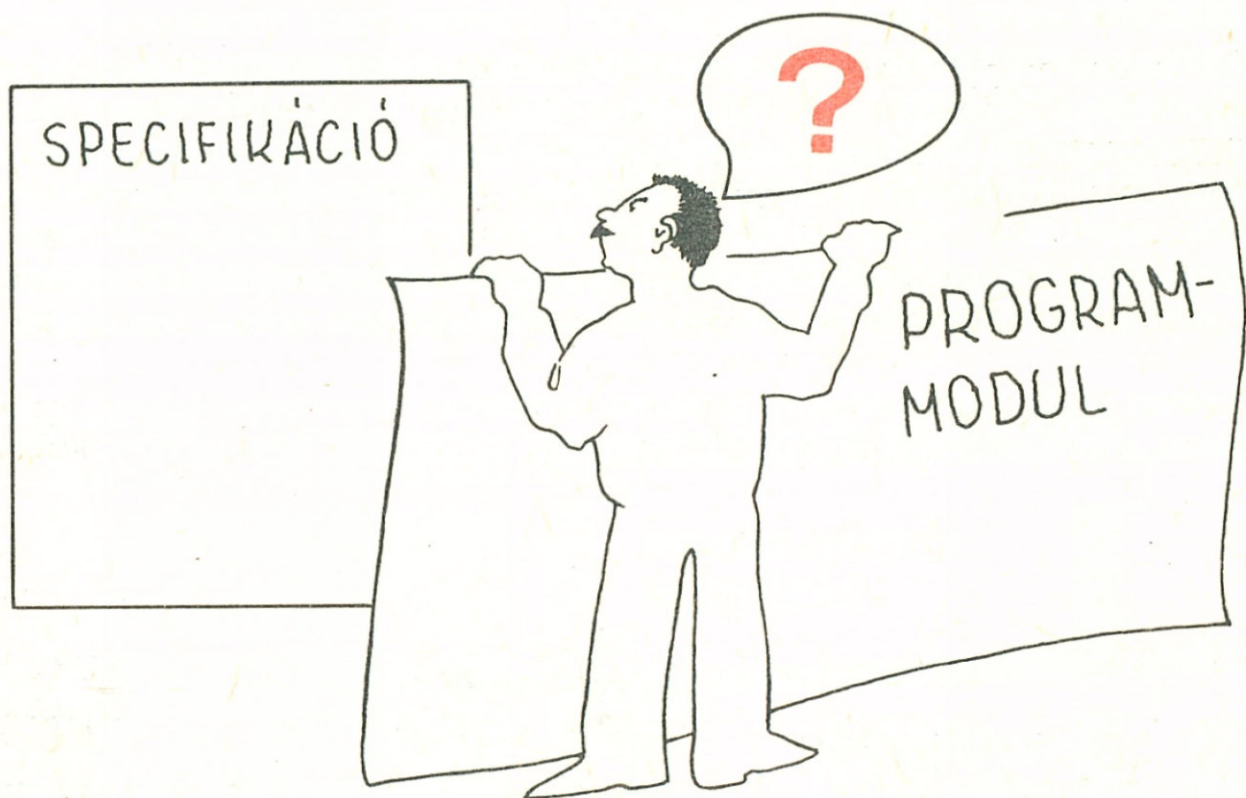
Ezekon kívül a specifikáció gyakran tartalmaz szervezési jellegű információkat a felhasználandó kulcsszámokról; bővebben erre itt nem térünk ki.

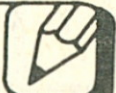
SZ →



Mindezen előírásokra szüksége van a programozónak a programmodul elkészítésének megkezdéséhez.

A következőkben azt mutatjuk be, hogy milyen lépésekből áll a programfejlesztés.





1. Az ötszakaszos koncepciót figyelembe véve mikor indul a tulajdonképpeni programfejlesztés?

Javaslati szakasz.

Első tervezési szakasz.

Második tervezési szakasz.

Első megvalósítási szakasz.

Második megvalósítási szakasz.

2. Melyik dokumentációból veszi a programozó a kiindulási adatokat programozáshoz?

3. Jelölje meg, mi az, amit a programozó biztosan nem ismer, amikor egy programmodul fejlesztését elkezdi!

Az elkészítendő modul helyét a teljes rendszerben.

Az elkészítendő programtól várt szolgáltatásokat.

A rendelkezésre álló erőforrásokat (pénz, személyek, gépek).

Az elkészítendő programmal kapcsolatban álló más modulok csatlakozófelületeit.

Információkat az alkalmazandó programozási nyelvekről, eszközökről, módszerekről.

Az elkészítendő program vázlatos felépítését.

A program részletes utasításszintű felépítési logikáját.



Válaszok

1. A programok kifejlesztése az

első megvalósítási szakaszban kezdődik.

2. A programozó a programozáshoz szükséges kiindulási adatokat a **specifikációból** veszi.

3. Az utolsó pont kivételével valamennyit ismeri a programozó, mielőtt a fejlesztést elkezdi.

A program részletes utasításszintű felépítési logikáját.

A program részletes felépítésének logikáját természetesen nem adják meg előre. Ennek kidolgozása a programozó egyik feladata.

A programfejlesztés lépései

A programfejlesztési lépések áttekintése

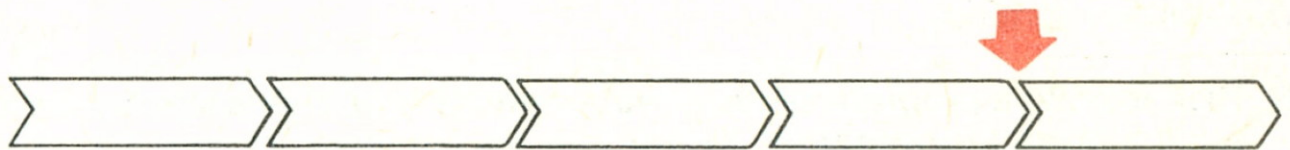
A programrendszer minden moduljának megvan tehát a specifikációja. A programozó akkor kezdheti el a modul implementálását (**megvalósítását**), ha rendelkezésére áll az általa kidolgozandó program specifikációja.

A programrendszer kidolgozása mindig azonos séma szerint, a következő lépésekben történik.

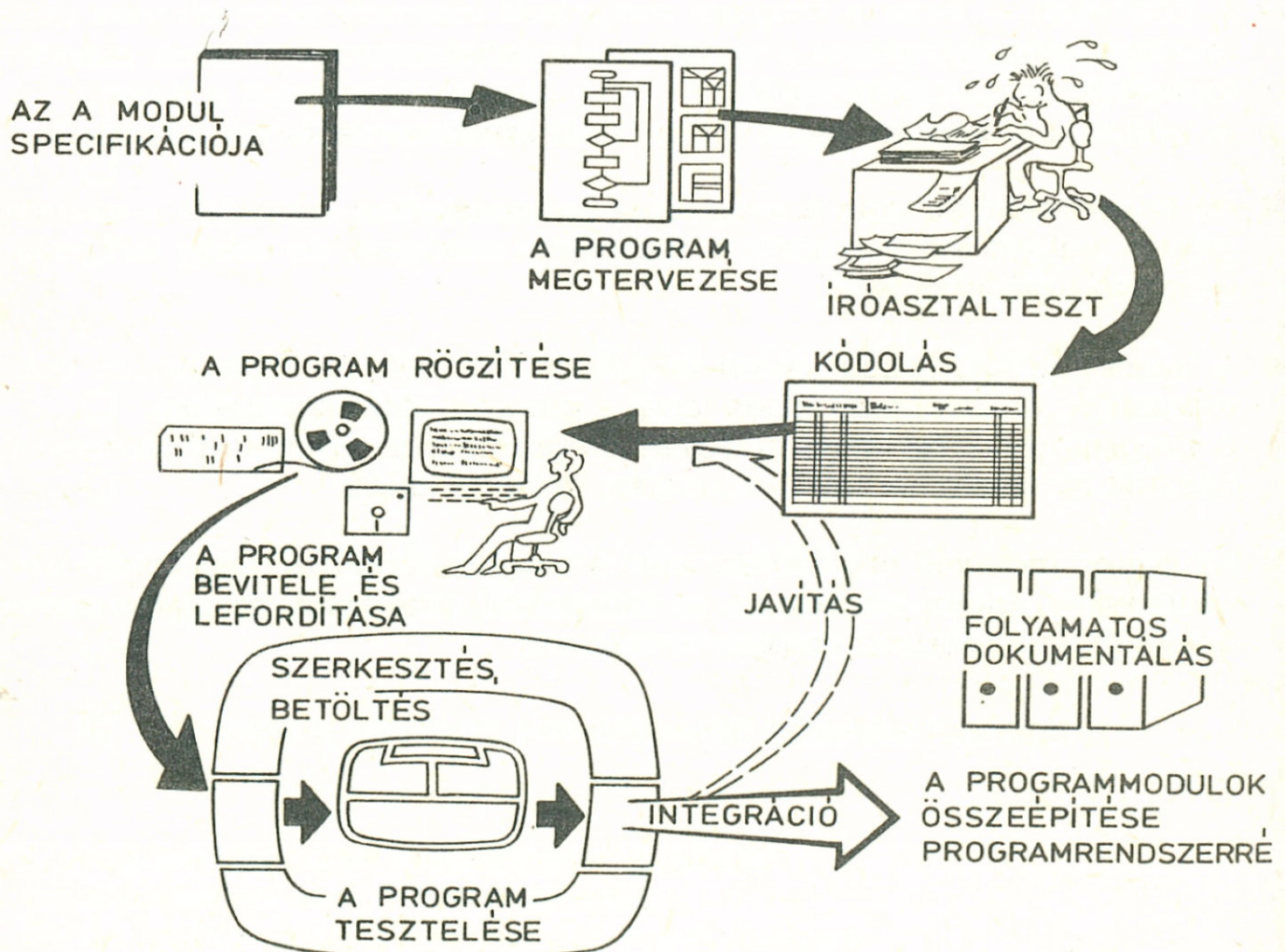
- Első teendő a programrendszer **valamennyi modulja részletes programlogikájának kidolgozása**. Ennek formája struktogram vagy folyamatábra.
- Ha a legrészletesebb szintű programlogika is elkészült, a következő lépés a **kidolgozott programlogika ellenőrzése**.
- A **kódolást** — azaz a program megírását valamely programnyelv utasításainak alapulvételével — csak akkor lehet elkezdni, ha a programlogikát gondosan ellenőriztük.
A kódolással azután véget ér az, amit *implementálásnak* nevezünk.
- Az implementálás után a programot tesztadatokkal lefuttatják a számítógépen, vagyis megkezdődik a **tesztelési szakasz**, és addig tart, míg a program hibátlanul fut.
- A tesztelt program csak akkor tekinthető késznek, ha szabályosan dokumentálták. A következő lépés tehát a program **dokumentálása**.
A programrendszer minden egyes programját az ismertetett lépésekben készítik el: implementálás, tesztelés, dokumentálás.
- Az egyes modulok elkészítése után a programrendszer még koránt sincs kifejlesztve. Ekkor következik az **integráció szakasza**, az egyes elemek összeépítése működő programrendszeré.

- Az integrációt új **teszt szakasz** követi (rendszer teszt vagy eljárás teszt), amelyben meg kell győződni az egyes moduloknak a teljes rendszerben való helyes együttműködéséről.
- Az integráció csak akkor tekinthető befejezettnek, ha a programrendszer **dokumentációját** kiegészítették az integráció és a rendszer teszt eredményeinek anyagaival.

Ha az integráció véget ért (az ötszakaszos sémában ez az első megvalósítási szakasz végén van), kezdődhet a **próbaüzem**.



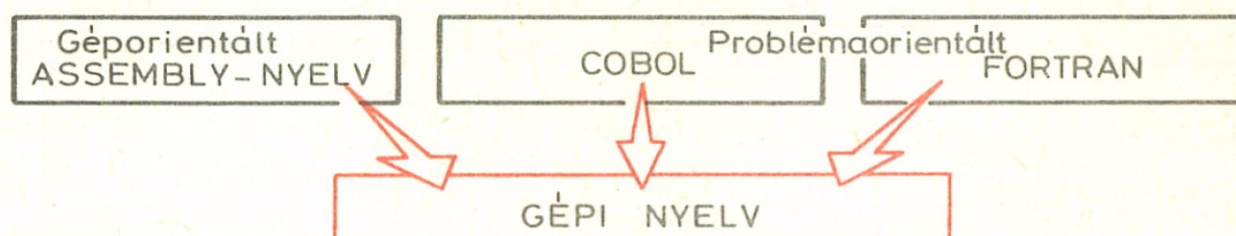
A következőkben a programfejlesztés egyes lépéseit részletesebben megvizsgáljuk.



A programlogika kidolgozása

Minden programozási feladat lényege a számítógép célszerű alkalmazása egy adott feladat megoldására. Az elgondolt feladatmegoldást ehhez — mivel az emberi logika nyilvánvalóan jóval bonyolultabb feladatok áttekintésére képes — alkalmassá kell tenni a korlátozott számítógép-logikára, amelyet a gép műszaki felépítése határoz meg. Egy számítógépet ui. csak a **gépi utasítások adott készletével** lehet működtetni. A számítógép által megértett gépi utasítások száma gépenként más és más, legtöbbször több, mint száz gépi utasításról van szó.

A programozó azonban nem gépi utasításokkal írja programját, ez túl bonyolult lenne. Sokkal inkább géporientált vagy problémaorientált programozási nyelvet választ magának, a feladattól függően.



A programozó első feladata tehát az, hogy olyan feladatmegoldást dolgozzon ki, amelyet a választott programnyelven meg lehet valósítani.

Ezt a tevékenységet nevezik a **programlogika kidolgozásának**. Rendszerint nem sikerül elsőre végleges programlogikát kidolgozni a feladat megoldására. Szokásos először a feladat nagyobb részekben való lebontását megtervezni, majd ezt lépésről lépésre finomítani mindaddig, amíg ez olyan szintet ér el, ahol lehetőség van a feladatmegoldást a választott programnyelvre átültetni.

A programnyelvek utasításait első közelítésben két csoportra lehet osztani: *műveleti utasításokra* és *döntési utasításokra*.

Műveleti utasítás minden számítási, tárolási, átviteli stb. utasítás, vagyis valamennyi olyan utasítás, amely akciót eredményez. Ezek elvileg elegendőek egy program megírásához. Világítsa meg ezt egy példa.

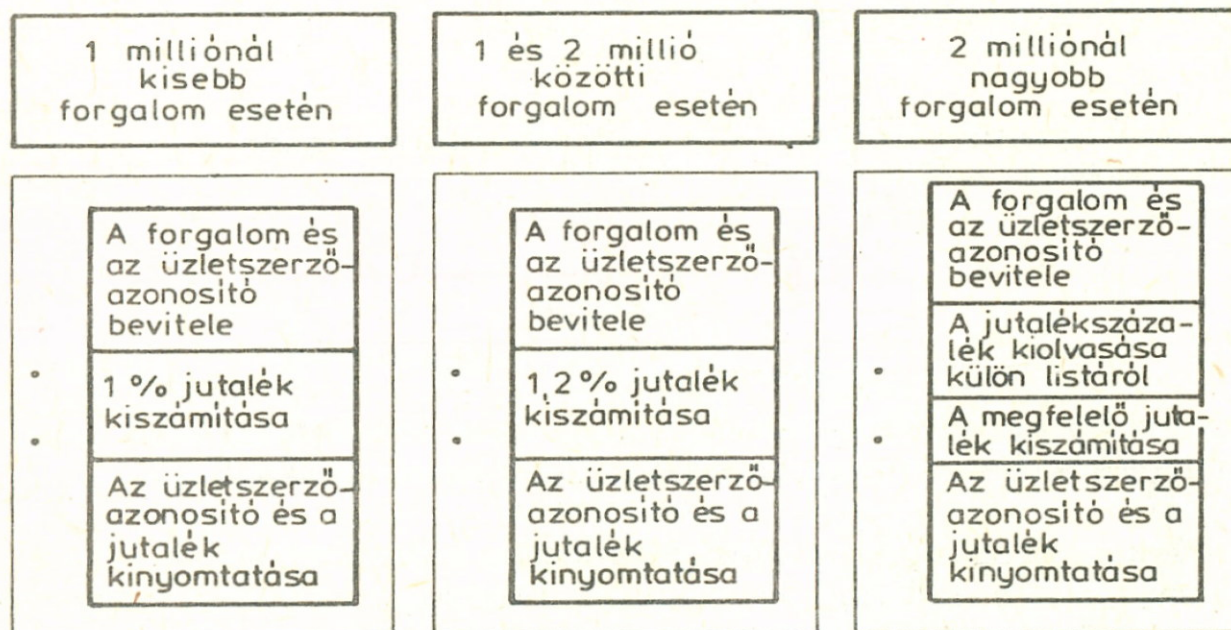
Példa:

Egy vállalat — azonosítószámokkal megadott — üzletszerzőinek a jutalékát kell egyenként kiszámítani és kinyomtatni.

A jutalék 1 millióig 1%, 1 és 2 millió között 1,2%. A 2 millió fölött teljesítő üzletszerzőkkel egyénileg kell megállapodni, ezeket külön listára írjuk.

A három szóba jövő esetre ezek szerint külön programokat írunk, amelyek közül az üzletkötő által elért forgalomnak megfelelő futtatjuk le.

Az ennek megfelelő struktogramok a következők:



Mivel mindhárom programban azonos tevékenységek fordulnak elő, kézenfekvő a három programot egyesíteni. Az ebből előálló egyetlen programnak természetesen teljes egészében teljesíteni kell az előző három szolgáltatásait. Azokat a tevékenységeket, amelyek mindhárom korábbi programban azonos módon előfordultak, csak egyszer kell leírni. A programot csak akkor kell **elágaztatni**, ha esetről esetre különböző tevékenységeket kell végrehajtani. Erre a célra szolgálnak a feltételes elágaztató utasítások, amelyek lehetővé teszik, hogy bizonyos tevékenységek végrehajtását meghatározott feltételek teljesülésétől tegyük függővé.

A következő ábra a „jutalékpélda” megoldási menetét mutatja. Ez műveletek és döntések sorozatából áll úgy, hogy egy döntés eredménye alapján mindig egy előre meghatározott tevékenység megy végbe. **Struktogramban** ezt a következőképpen ábrázoljuk:

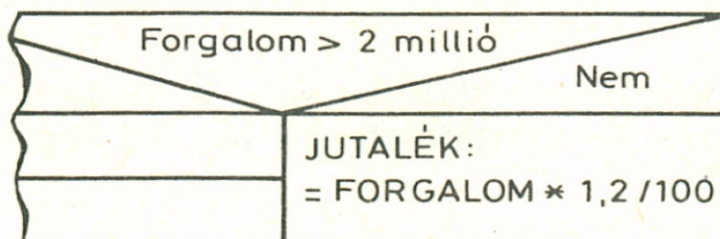
A forgalom és az üzletszerző-azonosító bevétele		
Igen	Forgalom > 1 millió	Nem
Igen	Forgalom > 2 millió	Nem
Jutalékszázalék kiolvasása az üzletszerző-azonosítóhoz egy külön listáról	1,2 % jutalék kiszámítása	1 % jutalék kiszámítása
A megfelelő jutalék kiszámítása		
Az üzletszerző-azonosító és a jutalék kinyomtatása		

A gyakorlatban a programokat csaknem mindig műveletek és döntések sorozataként írják meg.

Az első struktogram a feladat megoldásának logikai szerkezetét csak nagyvonalakban mutatja, ezt kell a fejlesztés további szakaszaiban finomítani. Az, hogy ezt hány szinten finomítjuk, a megoldandó feladat terjedelmétől és attól a programnyelvtől függ, amelyben a programot meg kell írni.

A fenti példa struktogramját is lehet még finomítani. Az ezutáni szinten pl. olyan megfogalmazásokat lehetne választani, amelyek már a felhasználandó programnyelvre hivatkoznak. A következő részletben az „1,2% jutalék kiszámítása” tevékenységet a PASCAL nyelv írásmódjával fejeztük ki. A jutalék a következő összefüggésből határozható meg:

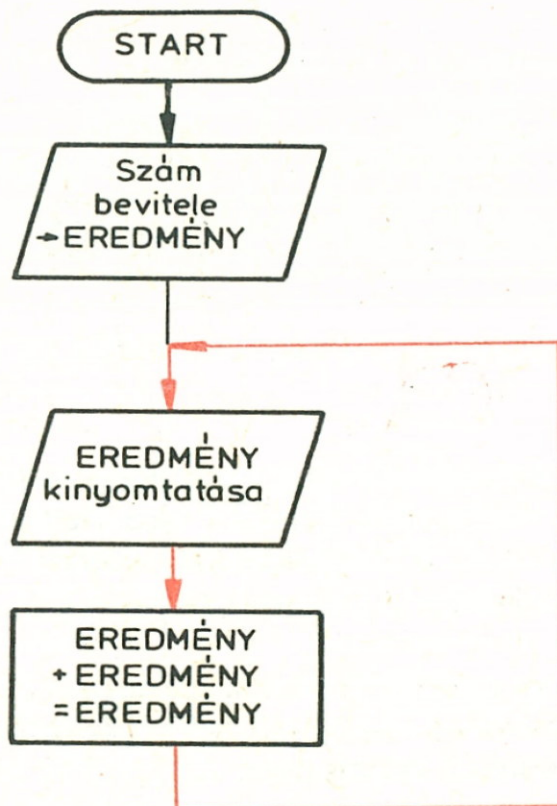
Jutalék = forgalom × százalékkulcs: 100.



A **program logikai menetének** kidolgozását szemléltesse egy másik egyszerű példa.

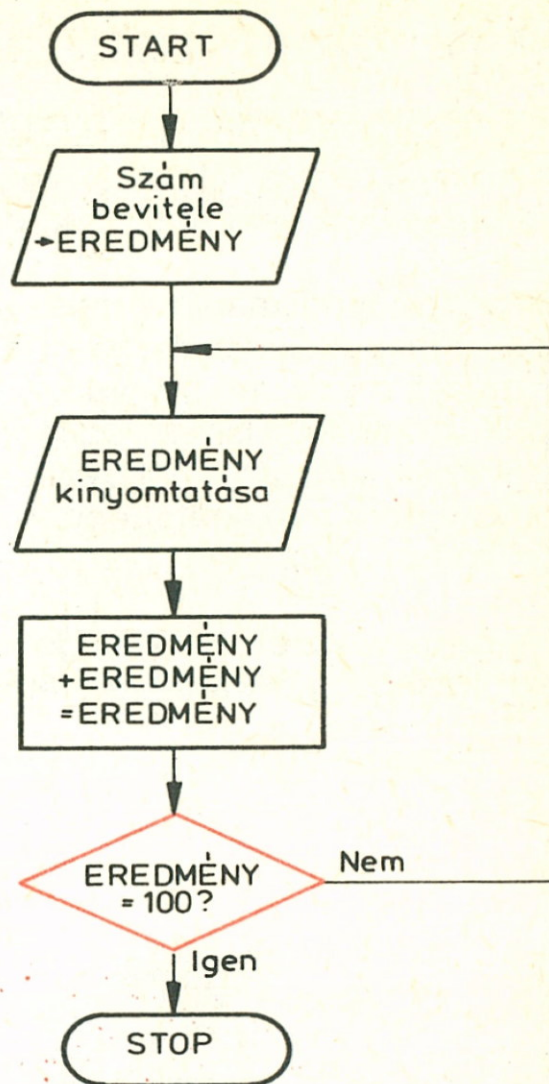
Egy terminálon át bevitt, 0-nál nagyobb számot ki kell nyomtatni, majd megkétszerezni, és ismét kinyomtatni mindaddig, amíg az eredmény kisebb, mint 100.

A megoldás menete — program-folyamatábraként megadva — valahogy így kezdődhetne:



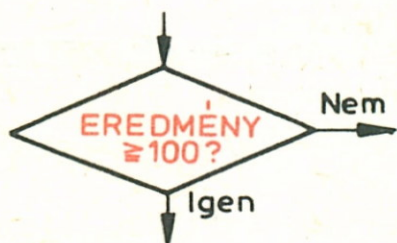
Egy számot beviszünk és azt elhelyezzük az operatív tár egy területére, amelynek neve EREDMÉNY, az EREDMÉNY tartalmát kinyomtatjuk (ennek ellenére az ottmarad tárolva). Ezután az EREDMÉNY tartalmához önmagát hozzáadjuk, majd a nyomtatás következik.

Ennél a feladatmegoldásnál **végtelen ciklus** jön létre, a feldolgozás tehát nem ér véget. Be kell építeni egy **végevizsgálatot**, amelyen már az első összeadás után keresztül kell haladni.



Az új folyamatábra első pillantásra igen logikusnak tűnik. A program működne is — azonban csak olyan számokra, amelyeknél az összeadás eredménye pontosan 100-at ad.

Az összes többi számnál a 100-zal való összehasonlítás a „nem” ágra futna, és ismét benmaradnánk a hurokban, pl.: ha a kiinduló szám 3, akkor az alábbi számsor adódna: 3, 6, 12, 24, 48, 96, 192 stb. A vizsgálatnak tehát ilyennek kell lennie:



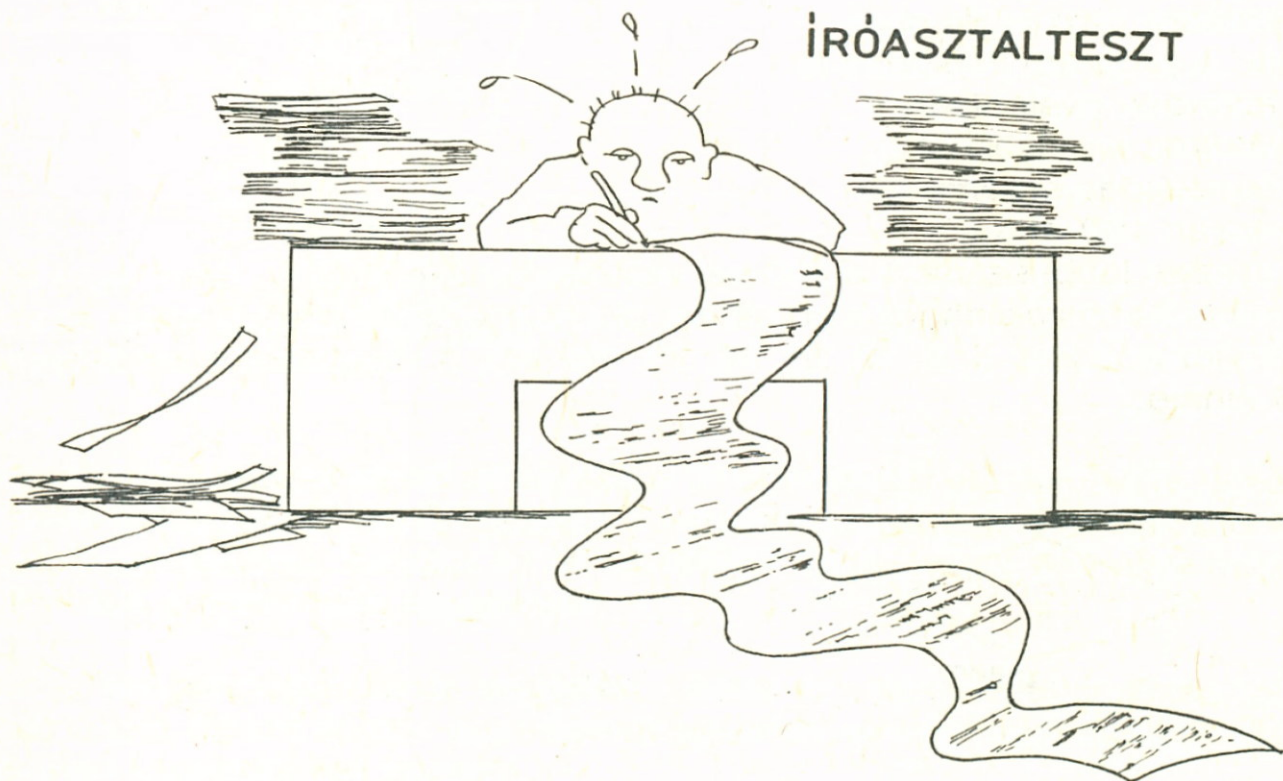
Ha az eredmény egyenlő vagy nagyobb (\geq), mint 100, érjen véget a program végrehajtása. A számítógép számára a programozás során a feltételeket ehhez hasonlóan kell megfogalmaznunk, a számítógép ui. nem képes magától gondolkozni.

A program logikájának ellenőrzése

Ha a program logikája részletes struktogram vagy folyamatábra formájában elkészült, azt a kódolás, vagyis a programnak a kiválasztott programnyelv utasításaival való megírása előtt még egyszer ellenőrizni kell.

A programlogika ellenőrzésére több módszer is van. A walk-through módszer és az inspekciós technikát már ismertettük. Mindenesetre célszerű, ha a programozó maga még egyszer átgondolja a program logikáját, és gondolatban próbaadatokkal ismét végigfuttatja azt, miközben az érintett adatok változásait egy táblázatban végigköveti. Tekintve, hogy ezt a tevékenységet az íróasztalnál és nem a számítógéppel végzik, ezt az ellenőrzést a programozók maguk is **íróasztaltesztnek** (vagy száraztesztnek) nevezik.

Megéri ezt az ellenőrzést gondosan elvégezni, mert az itt ki nem derített hibákat később gyakran csak nagy ráfordítással lehet elhárítani.

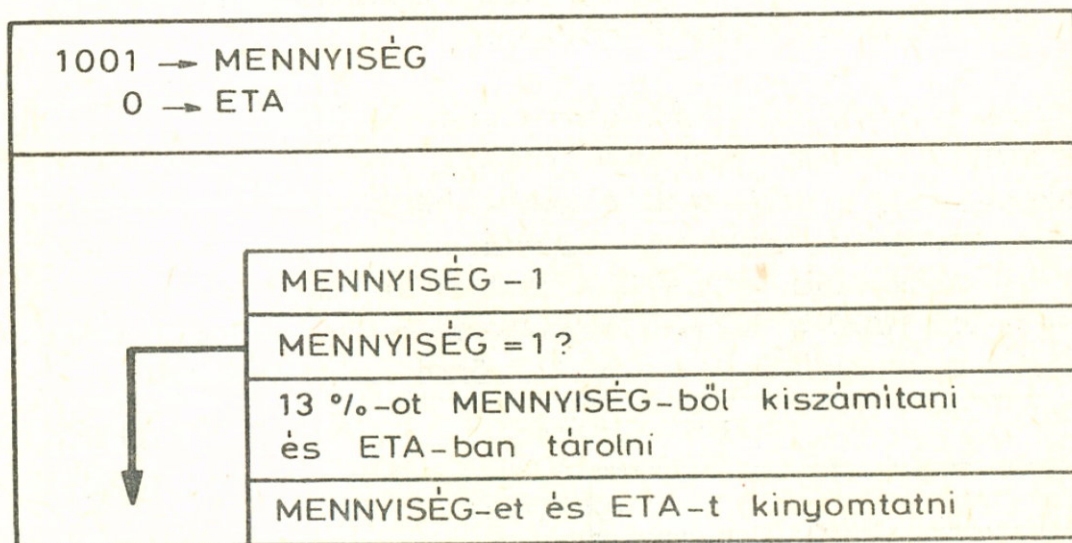




1. Miért nem lehet a számítógéppel egy feladat megoldását köznapin nyelven megfogalmazva közölni?
Jelöljön meg minden állítást,
amelynek e kérdés megválaszolásában jelentősége van!

- A számítógép csak korlátozott számú utasítást ért meg.
- A programozó a megoldáshoz csak olyan utasításokat használhat, amelyeket a számítógép megért.
- A programozónak úgy kell megfogalmaznia a feladat megoldását, hogy a részletezett folyamatot később egy programnyelven le lehessen írni.

2. Egy programozó egy feladat megoldására ezt a programlogikát dolgozta ki (ETA: értéktöbbletadó):



- a) Fogalmazza meg a feladatot, amelynek ez a program a megoldása!

.....

- b) Hajtson végre íróasztaltesztet, és ellenőrizze,
van-e benne logikai hiba!



Válaszok

1. A kérdés megválaszolásánál valamennyi állításnak van jelentősége.

- A számítógép csak korlátozott számú utasítást ért meg.
- A programozó a megoldáshoz csak olyan utasításokat használhat, amelyeket a számítógép megért.
- A programozónak úgy kell megfogalmaznia a feladat megoldását, hogy a részletezett folyamatot később egy programnyelven le lehessen írni.

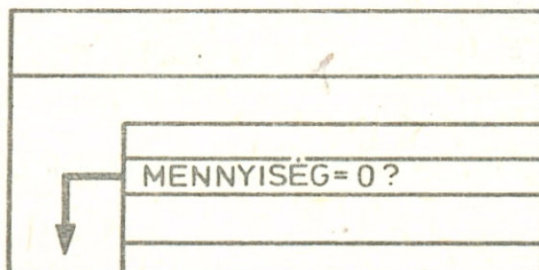
2. a) A feladat: **értéktöbbletadó-táblázatot kell kinyomtatni a 2-től 1000-ig terjedő értékekre.**

b) íróasztalteszt: ha a feladatot értelemszerűen a fentiek szerint fogalmazta meg, akkor nincs a programban logikai hiba.

Ha azonban a feladat így szólna:

„Értéktöbbletadó-táblázatot kell kinyomtatni az **1-től 1000-ig** terjedő értékekre”,

akkor a vizsgálatnak a következőnek kellene lennie:



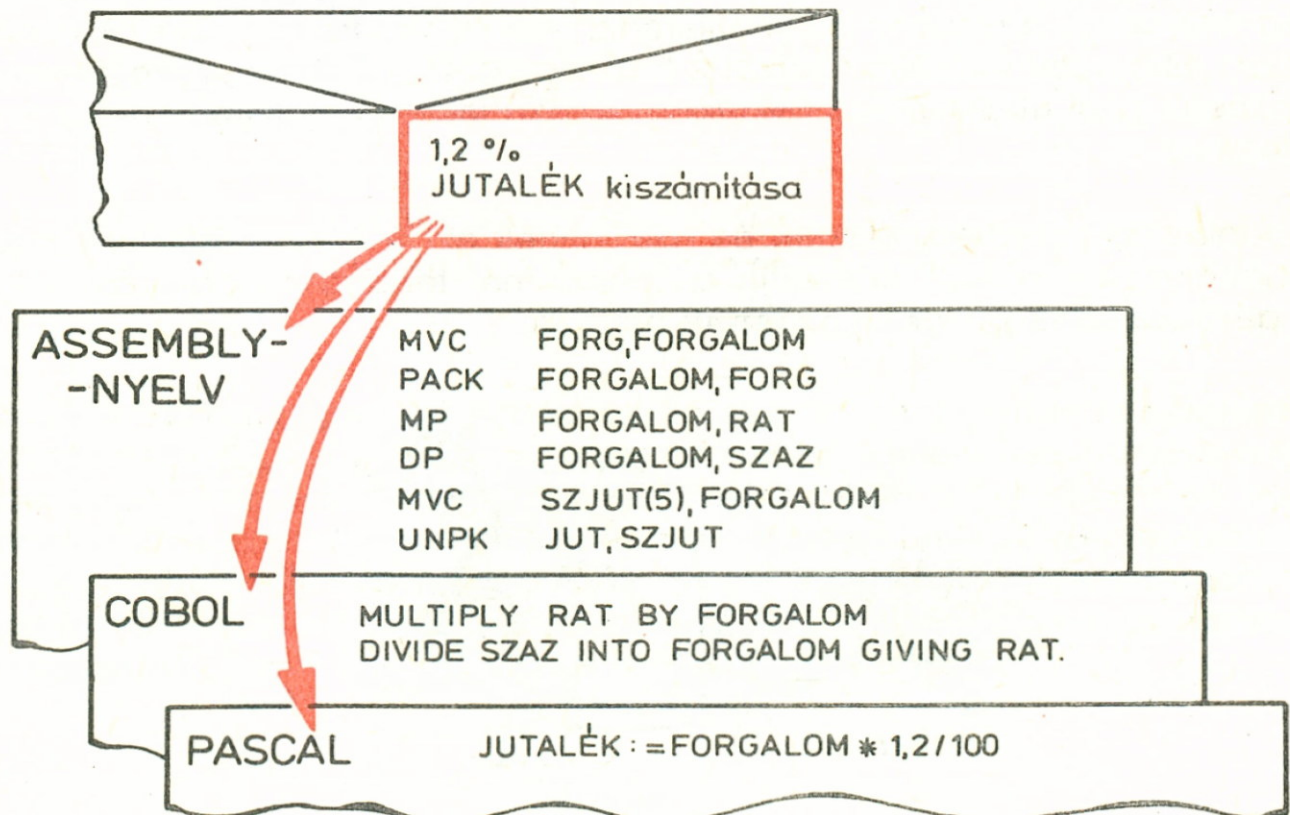
A program kódolása

Miután a program logikai menetét ellenőriztük, hozzáfoghatunk ahhoz, hogy a program logikai menetének minden lépését **egy programnyelv utasításai-val** kifejezzük: vagyis kódoljuk a programot. A kódolás tehát az utasítások többé-kevésbé köznyelven kifejezett sorozatának átalakítása egy programnyelv utasításainak sorozatára.

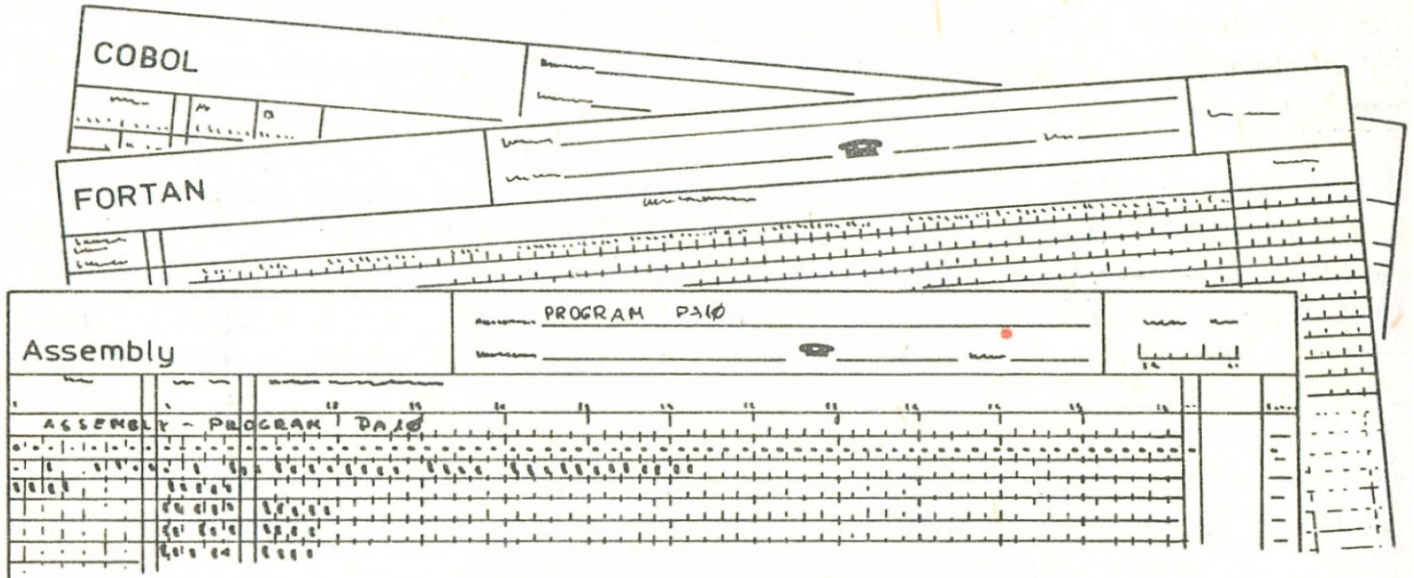
A kódolást legtöbbször a számítógéppel párbeszédéses üzemmódban, terminálon keresztül hajtják végre. A programozó eközben utasítást utasítás után billentyűz be a gépbe. Programállományt hoz létre, amely tárol minden utasítást, és a programot bármikor javíthatóvá, változtathatóvá és kiegészíthetővé teszi. Ezeket a programállományokat rendszerint mágneslemezes háttértárakon helyezik el.

Minden programnyelvnek megvannak a maga formális szabályai, amelyeket az utasítások bebillentyűzésekor figyelembe kell venni. Pl. minden COBOL utasítást ponttal kell lezárni.

Következik egy példa, amelyen bemutatjuk, hogyan történik egy struktogram egy utasításának kifejezése különböző programnyelveken.

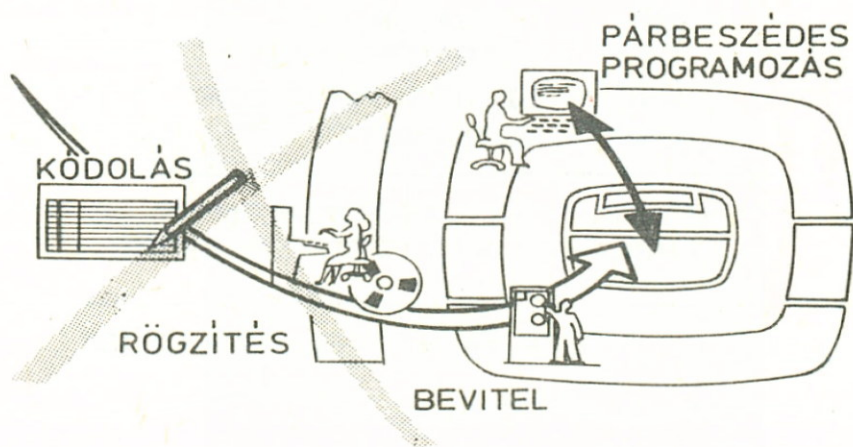


Akinek nincs módja közvetlenül a terminálnál ülve kódolni, az **kódlapokra** írja programját. Kódlapok a gyakoribb programnyelvekhez kaphatók, beosztásuk segít az illető programozási nyelv formális szabályainak betartásában.



Attól függően, hogy a programot milyen adathordozótól kívánjuk bevinni, a kódlap bejegyzéseit adatrögzítő berendezéssel a megfelelő adathordozóra (mágnesszalag, hajlékony mágneslemez, lyukkártya) fel kell vinni. A kódolás közben elkövetett formális hibákat a fordítóprogram — amely a programnyelv utasításait gépi utasításokra fordítja — ismeri fel, és egy listán közli a programozóval.

A kódolásnak, adatrögzítésnek és -bevételnek ezt a kissé nehézkes módját azonban egyre inkább kiszorítja a terminálnál történő kódolás és bevétel. Ezt **párbeszédés programozásnak** nevezik.



A program tesztelése

A program kódolásával a munka még nem ért véget. A **program tesztelése**, ami tehát ezután következik, gyakran több időt igényel, mint amennyire a fejlesztési munkában eddig összesen szükség volt. Ez különösen akkor fordul elő, ha a kódolás előtt elmulasztották a tervek és közbenső eredmények gondos ellenőrzését.

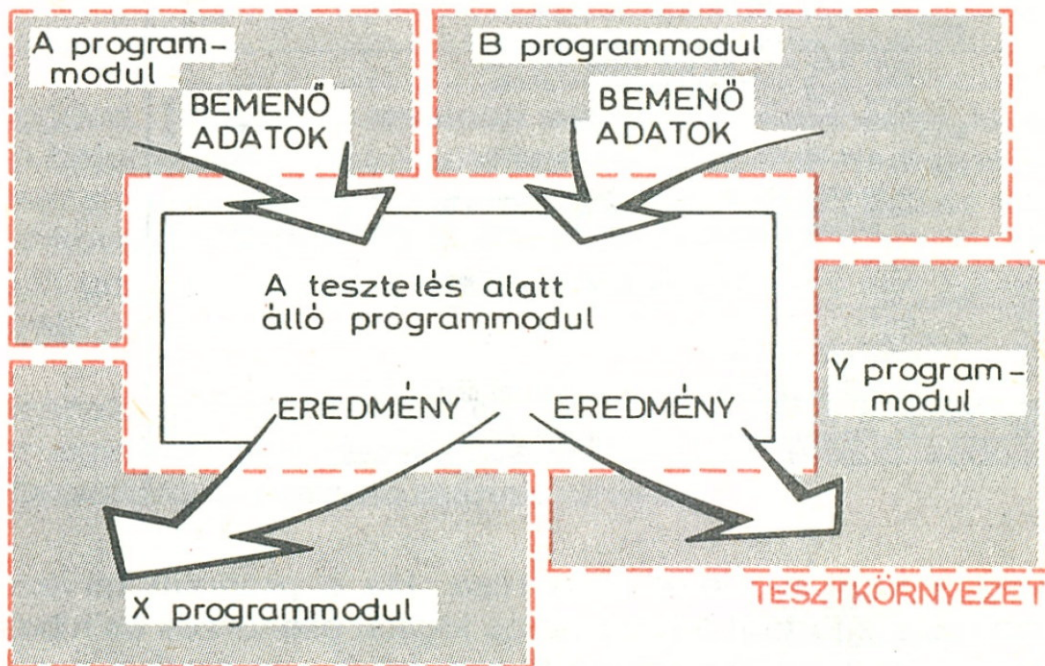
A programteszt már a fordítással kezdődik, azaz a fordítási futás közben, amikor a számítógép a kódolt programot a gépi nyelv utasításaira alakítja át. A fordítóprogram felismeri a programnyelv „nyelvtani” szabályainak megsértését, a formális vagy szintaktikai hibákat, és a **fordítási listán** rögzíti azokat. Az ilyen hibák viszonylag könnyen javíthatók, és a programot újra lehet fordítani.

128	BLOCK B.	
129	IF G60-STATUS NOT = 0. GO TO B0-Z.	
130	READ B350-ALT, AT END CLOSE B350-ALT,	
131	MOVE 2 TO G60-STATUS. GO TO B0-Z.	
132	MOVE 4 TO G60-STATUS.	
133	MOVE KUNDENR IN B350-ALT-SALZ TO G60-KUNDE.	
134	B0-Z.	
135	IF G61-STATUS NOT = 0. GO TO B1-Z.	
136	READ B50-ALT, AT END CLOSE B50-ALT,	
137	MOVE 2 TO G61-STATUS. GO TO B1-Z.	
138	MOVE 4 TO G61-STATUS.	
139	MOVE KUNDENR IN B50-ALT-SALZ TO G61-KUNDE.	
140	B1-Z.	
141	IF G62-STATUS NOT = 0. GO TO B-EX.	
142	B21.	
143	READ B10Z. AT END CLOSE B10Z.	
144	MOVE 2 TO G62-STATUS. GO TO B-EX.	
145	IF RECHNUNGSMOENT IN B10Z NOT EQUAL TO VERGLEICHSMOENT	
146	GO TO B21.	
147	MOVE 4 TO G62-STATUS.	
148	MOVE KUNDENR IN B10Z-SALZ TO G62-KUNDE.	
149	B-EX.	
150	EXIT.	
151	BLOCK G.	
152	MOVE G60 TO GGN.	
153	G1.	
	IF G61 LESS THAN GGN. MOVE G61 TO GGN.	
154	G2.	
155	IF G62 LESS THAN GGN. MOVE G62 TO GGN.	

Nehezebb dolog a **logikai hibák** elhárítása. Ha a program tervezésének előző fázisaiban meg is történt a folyamatos logikai ellenőrzés és hibajavítás, maradhattak még a programban olyan logikai hibák, amelyeket csak tesztadatokkal való számítógépes futásokkal lehet felismerni. A programfutásokat különböző tesztadatokkal hajtják végre, és az eredményeket összehasonlítják a várt eredményekkel. Ha a két eredmény nem egyezik meg, a programozó átvizsgálja az „ezért felelős” részeket, és kijavítja a hibákat.

Ritkán lehet egy programot minden lehetséges adattal kipróbálni, többnyire korlátozott számú **tesztadattal** kell megelégedni. A tesztben először olyan adatokat használunk, amelyek a **későbbi üzemszerű alkalmazás során előfordulnak**, majd **szélsőértékekkel** (legkisebb megengedett érték, legnagyobb megengedett érték, „nulla” érték, negatív számok stb.) tesztelünk, és végül még **hibás értékekkel** is (túl nagy értékek, betűk, számok helyett vagy fordítva, bitkombinációk, amelyek sem számnak, sem betűnek nem felelnek meg), hogy kipróbáljuk, hogyan viselkedik a program meg nem engedett adatokra.

Azért, hogy a programot a gyakorlathoz közel álló környezetben próbálhassuk ki, a tesztelendő program „környezetét” szimulálni kell. Ki kell alakítani a többi programhoz vagy programrészhez a csatlakozófelületeket, azaz a vizsgált program bemenő adatait biztosítani kell, és megfelelő helynek kell rendelkezésre állnia az eredmények felvételére. Ezt a szimulált környezetet **teszt-környezetnek** nevezzük.



Ha a tesztek végre lezárulnak, a programozónak hozzá kell látnia, hogy valamennyi anyagot, amelyre programjának dokumentálásához szüksége van, rendezze, kiegészítse, és a hiányzókat pótolja.

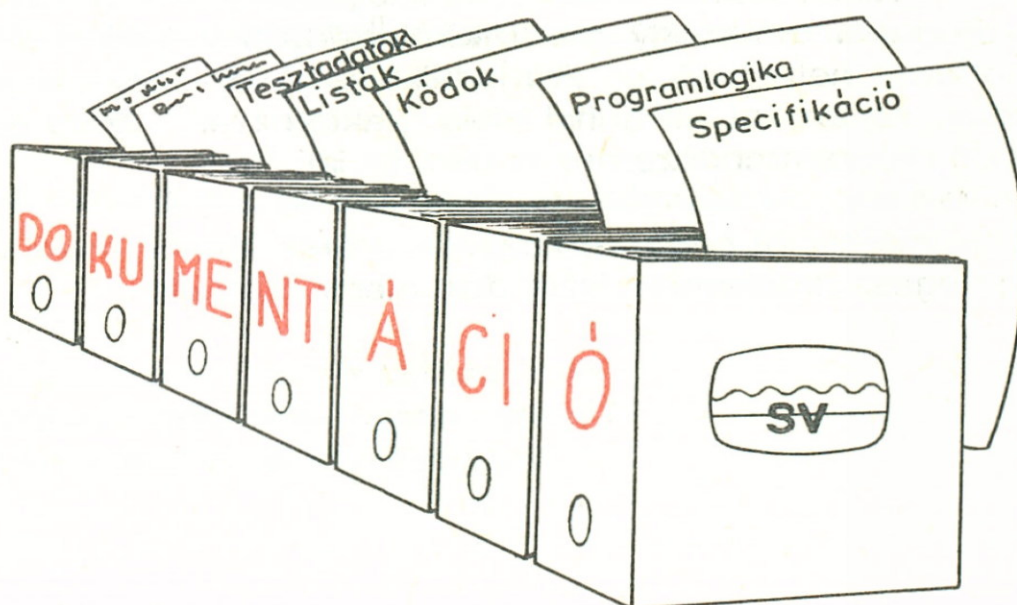
A program dokumentálása

Ha a programtesztelés szakasza lezárult, a programozó összeállíthatja mindazon anyagokat, amelyek az elkészült programot **dokumentálják**.

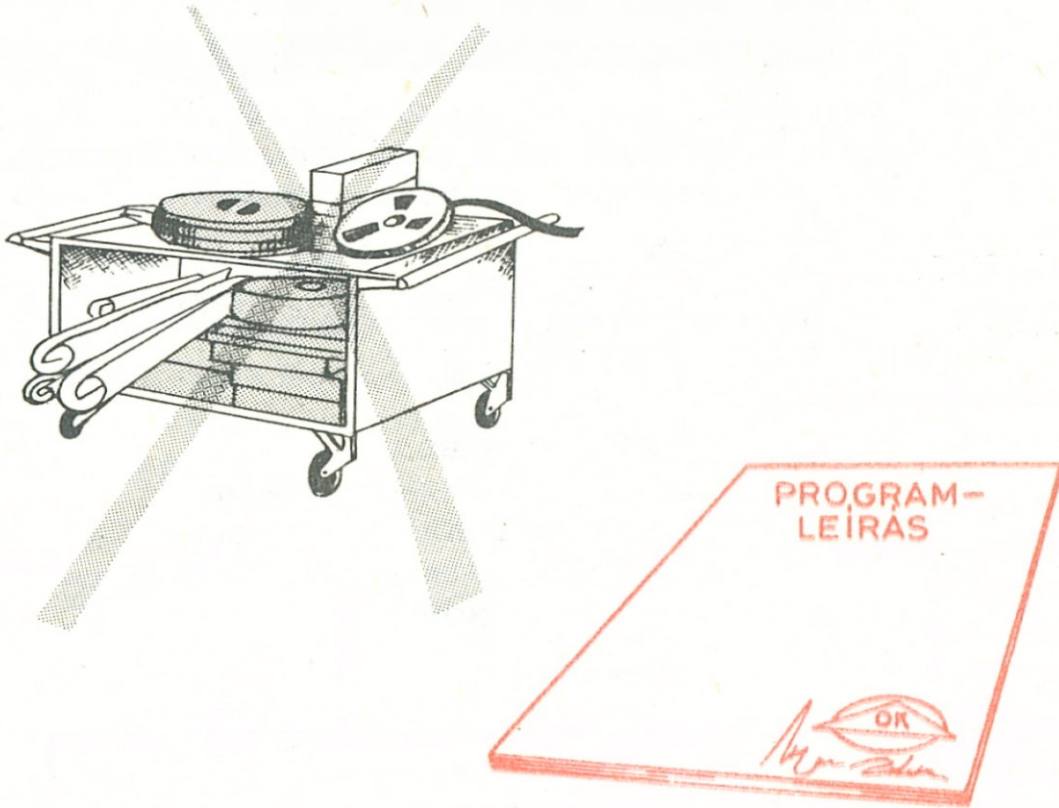
Annak megállapítására, hogy egy rendszer valamennyi programmodulja azonos módon dokumentált, olyan dokumentációs irányelveket vagy ellenőrzőlistákat bocsáthatunk ki, amelyekhez a programozók dokumentálás közben igazodhatnak. A leírások megfogalmazásában gyakran segítenek különleges dokumentációs segédeszközök.

A **programmodul teljes leírásához** a specifikáció mellett — ami már kiinduló adata volt a programnak — hozzátartozik:

- a program logikai menetének ábrázolása, ha szükséges, magyarázó megjegyzésekkel ellátva, pl. struktogramok, program-folyamatábrák, döntési táblázatok;
- kódlapok abban az esetben, ha nem terminálnál kódoltak;
- számítógépes jegyzőkönyvek (protokollok), pl. fordítási és szerkesztési listák;
- tesztjelentések, tesztadatok, tesztprotokollok.



A programleírást **tömören, pontosan és könnyen érthetően** kell megfogalmazni úgy, hogy olyasvalaki is, aki maga nem vett részt a fejlesztési munkában, gyorsan bedolgozhassa magát a programba.



A program csak akkor tekinthető késznek, ha teljes körű dokumentációval van ellátva, és azt a döntést hozó testület elfogadta.

Ami a programmodullal ezután történik, az a teljes projekt szervezésétől függ. Ésszerű a programmodult egy **modulkönyvtárban** tárolni, amelyet rendszerint háttértáron helyeznek el. A modulkönyvtárban tárolnak minden elkészült modult, és azokat ott annyi ideig „raktározzák”, amíg egy később összeépítendő programrendszerhez le nem hívják őket.

Az írásos dokumentációt célszerű központi helyen tárolni, mivel ez később része lesz az egész programrendszer dokumentációjának.



1. Mit jelent a kódolás?

Jelölje meg a helyes választ!

- A kódolás: implementálás.
- A kódolás a köznyelven megfogalmazott utasítások leképezése
egy programnyelv utasításaira.
- A kódolás egy programnyelven megfogalmazott
utasítások lefordítása a gépi nyelv utasításaira.

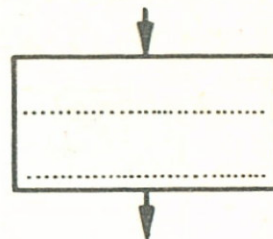
2. Mit gondol Ön, milyen lenne ez az utasítás
egy program-folyamatábrában?

PASCAL:

ÖSSZEG: = SZÁM + MENNYISÉG

COBOL:

ADD SZÁM MENNYISÉG GIVING ÖSSZEG



3. A futási teszt során mikor ismerhetők fel a formai hibák
és mikor a logikai hibák?

Formai hibák:

Logikai hibák:

4. A későbbi alkalmazásban előforduló adatokon kívül
milyen fajta adatokat kell még a programtesztbe bevonni?

..... és

5. Hogy hívják a tesztelendő program szimulált környezetét?

.....

6. A specifikáción kívül mi tartozik még
a programmodul dokumentációjához?
Írjon ide mindent, ami eszébe jut!

.....
.....



Válaszok

1. A kódolás következő definíciója helyes:

A kódolás a köznyelven megfogalmazott utasítások leképzése egy programnyelv utasításaira.

2. A megfelelő utasítás egy program-folyamatábrában ilyen vagy ehhez hasonló lenne:



3. A formális hibák a **fordítási futás** során derülnek ki.

A logikai hibákat a **tesztadatokkal történő programfutás** során lehet kimutatni.

4. A tesztet nyilvánvalóan **hamis adatokkal** és **szélsőértékkel** is végre kell hajtani.

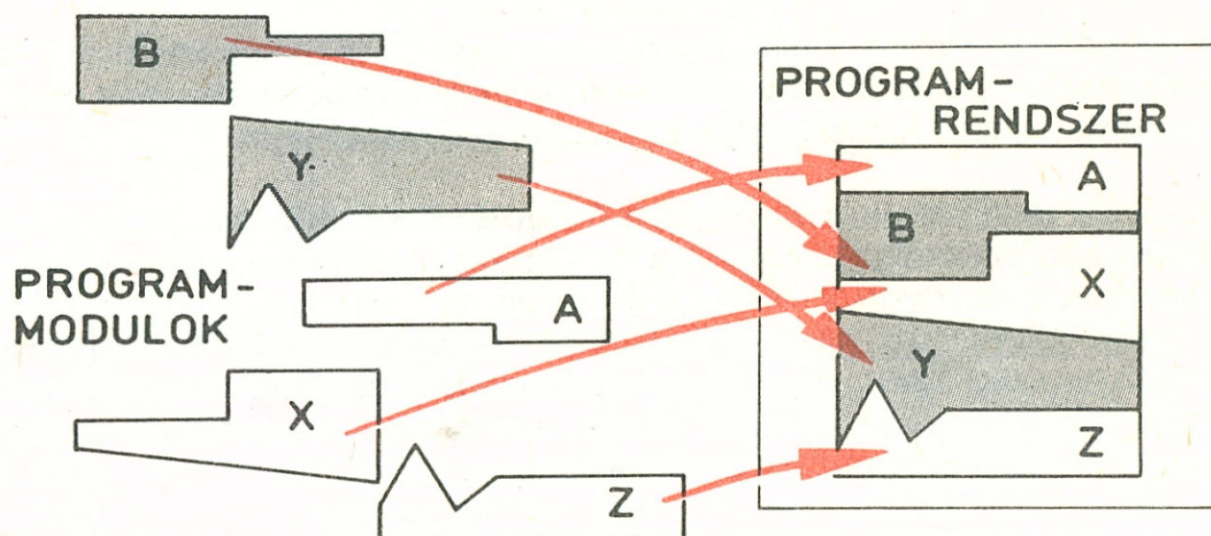
5. A tesztelendő program szimulált környezetét **tesztkörnyezetnek** nevezik.

6. A program dokumentációjához a specifikáción kívül — ami a program kiinduló információja — hozzátartozik a **program logikai menetének ábrázolása** (folyamatábrák, struktogramok), kódlapok, a **számítógépes listák** (fordítási és szerkesztési listák), a **tesztjelentések, tesztadatok, tesztprotokollok**.

A programrendszer integrálása

Az **integrálás** az egyes programmodulok összeépítése olyan **programrendszerre**, amellyel azután a teljes feladat megoldható.

Ha az egyes **modulok** csatlakozófelületeit pontosan határozták meg, ha az implementáció helyes, az összeépítés nem jelenthet többé nagy nehézséget. Nem jelent mást, mint egy szerkesztőfutást, amelynek során a lefordított programmodulokat a szerkesztőprogram összekapcsolja.



A programrendszer tesztelése

Az egész programrendszer tesztelése is az integráláshoz tartozik. Ekkor ellenőrzik, hogy az eljárás minden részletében előírászerűen teljesíti-e a megfogalmazott feladatot.

Az integrálást és a programrendszer tesztelését akkor is el lehet kezdeni, ha nem készült még el valamennyi programmodul. A hiányzó modulokat a tesztkörnyezet helyettesíti, és így a már elkészült modulok együttműködése tesztelhető. A hátramaradt modulokat azok elkészülése után lehet beilleszteni.

A teljes programrendszer ellenőrzését **rendszeresztnek** vagy **eljárástartesztnek** nevezik. A rendszereszt gyakran nagyon hosszadalmas, mert a sok, különböző programozó által kidolgozott modul együttműködésében és a hardverben még olyan hibák sora kerülhet napvilágra, amelyeket korábban nem lehetett észlelni. A tesztszakasz annál hamarabb zárható le, minél gondosabban határozták meg, hajtották végre és ellenőrizték korábban az egyes feladatokat.

A programrendszer dokumentálása

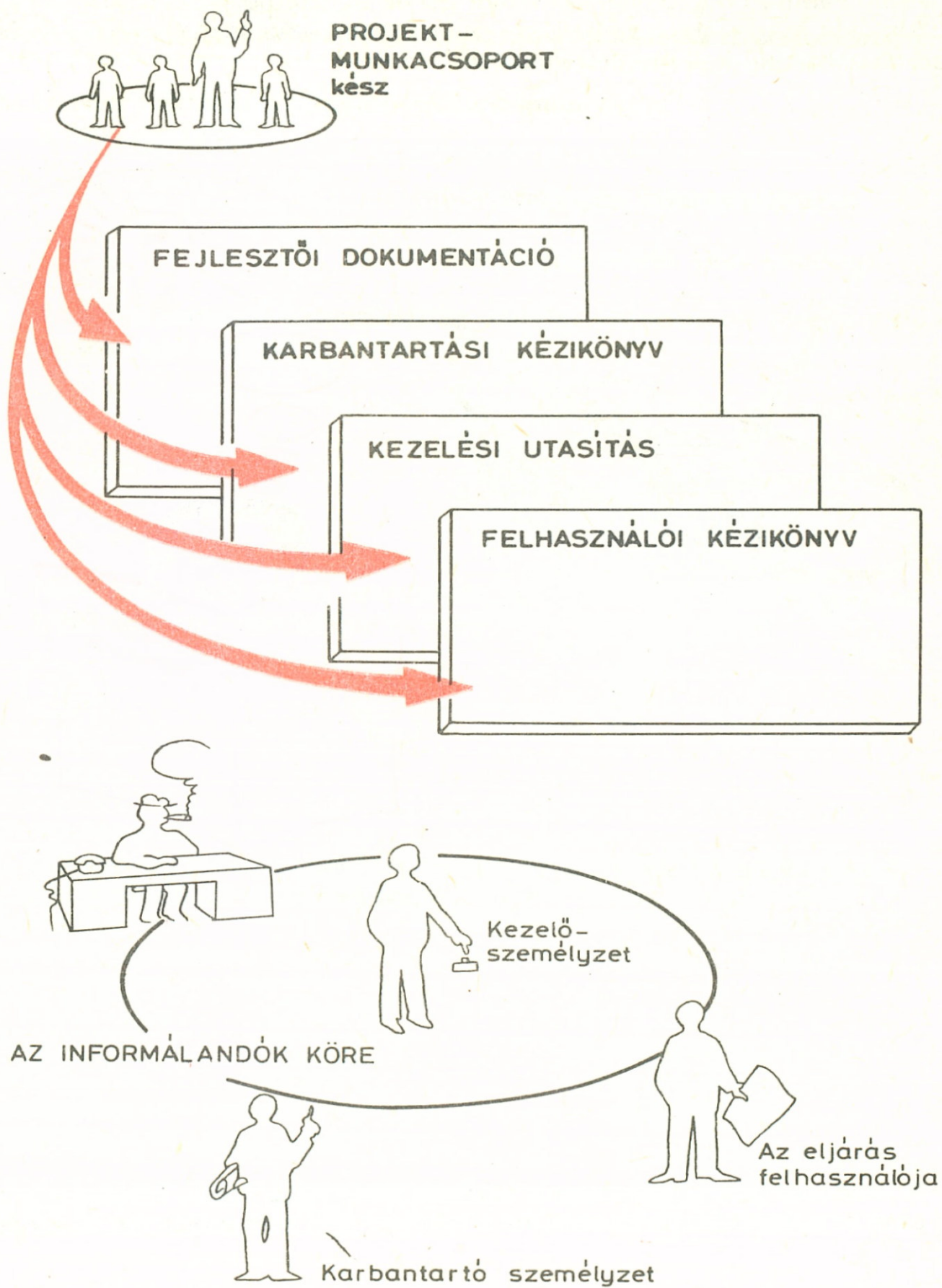
Mint valamennyi eddig leírt programfejlesztési lépésről, az integrációról és a rendszertesztről is **dokumentációt** kell készíteni. Ezeket az anyagokat az eddig elkészült dokumentációhoz — amit **fejlesztői dokumentációnak** is neveznek — utólag szerkesztik hozzá. Az **integrációs és tesztdokumentáció** mindenekelőtt az egyes modulok összeszerkesztési listája, tesztjelentések, jegyzetek a tesztadatokról, futási listák (tesztprotokollok) és az ezekhez fűzött magyarázatok.

A fejlesztői dokumentáció mellett további anyagokat is el kell készíteni: ezek a programrendszer felépítéséről és használatáról adnak kiválasztott információkat, meghatározott felhasználói csoportok igényei szerint.

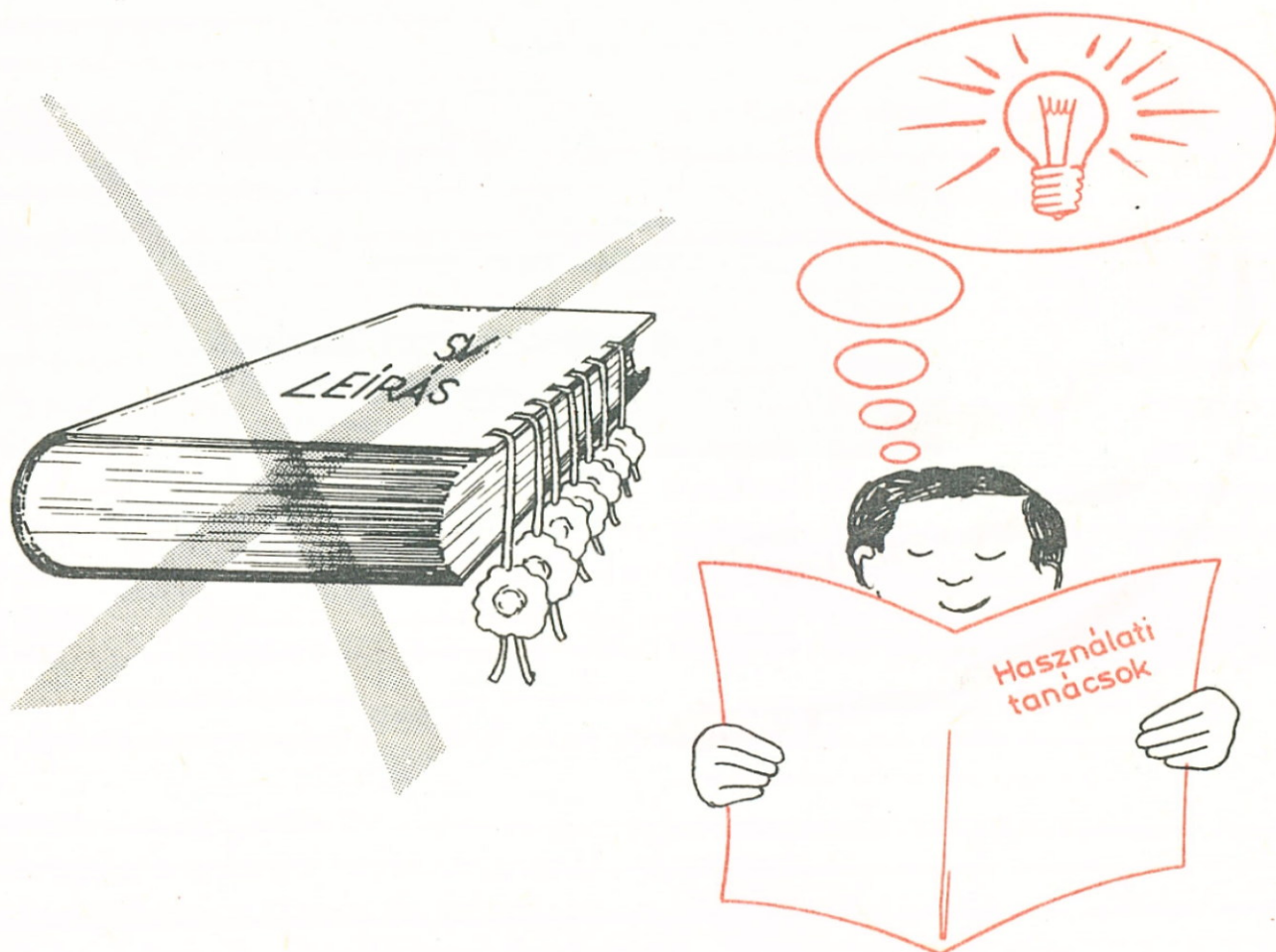
Ezek egyike a **szoftver-karbantartási (követési) kézikönyv**, amely a jövőbeni karbantartó személyzet számára tartalmazza a szükséges információkat az eljárásról és a szoftverről. A szoftver-karbantartók feladata, hogy az alkalmazás során jelentkező hibákat kijavítsák és a rendszert a megváltozott körülményekhez illesszék. A szoftver-karbantartási kézikönyvben benne van a fejlesztői dokumentáció jelentős része, ezenkívül útmutatás és előírások a szükségessé váló programmódosításokhoz.

További dokumentáció a **kezelési utasítás** (üzemeltetési dokumentáció), amely a gépkezelőkhöz szól. A gépkezelőknek a rendszer futtatásához szükséges információkon kívül ismeretekkel kell bírniuk a rendszer által igényelt hardverről (operatív tár kapacitásigénye, perifériák). Tudniuk kell, hogy milyen parancsokat kell a számítógépnek adni, milyen határidőre kell a rendszernek lefutnia, és mit kell csinálni hibák esetén.

Ezenkívül mindenkinek, akik a jövőben a rendszerrel dolgozni fognak, szüksége lesz **felhasználói kézikönyvre**, amelyet meg kell írni. Ez a dokumentáció leírja, hogyan működik a rendszer, és hogyan változik meg az általa érintett személyek munkája. A felhasználói kézikönyv írja le pl. azt, hogy milyen bemenő adatokat adhat meg a felhasználó, hogyan ellenőrzi ezeket a rendszer, milyen határidőket kell figyelembe venni, és mit kell tenni vészhelyzetekben.



Mivel a felhasználói kézikönyv többnyire nem számítástechnikusokhoz szól, ezt — ellentétben a szoftver-karbantartási és az üzemeltetési dokumentációval, amelyet magasabb színvonalon kell megírni — azt jól érthetően és könnyen olvashatóan kell megfogalmazni.



A rendszer jóságát gyakran a leírások jóságához mérik, hiszen: semmit sem ér a legjobb eljárás sem, ha a felhasználók nem tudják azt használni!



1. Hogyan történik a programmodulok integrálása a rendszerhez?

A számítógép felhasználásával.

A számítógép nélkül, csak papíron.

2. A rendszerteszt elkezdéséhez mindig valamennyi programnak készen kell lennie?

Igen.

Nem.

Indoklás:

.....

3. Nevezzen meg olyan anyagokat, amelyekkel a programrendszer fejlesztői dokumentációja az integráció folyamán bővül!

.....

.....

.....

4. Különleges felhasználói csoportok számára milyen anyagok készülnek a fejlesztői dokumentáción kívül?

.....

.....

.....



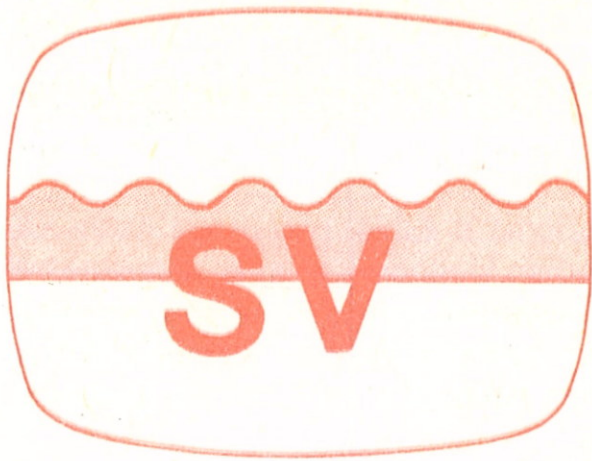
Válaszok

1. A programmodulok rendszerbe integrálása:
 a számítógép felhasználásával történik.

2. Nem.
Indoklás: A rendszerteszt elvégzése előtt **nem kell valamennyi programot elkészíteni, mert a még hiányzó modulokat helyettesíteni lehet.**

3. Az integrálás során a fejlesztői dokumentáció kiegészül:
**a modulok szerkesztési listáival,
a tesztelési jelentésekkel,
a tesztprotokollokkal,
a protokollokat magyarázó megjegyzésekkel.**

4. A fejlesztői dokumentáción kívül elkészül:
**a szoftver-karbantartási kézikönyv,
az üzemeltetési dokumentáció és
a felhasználói kézikönyv.**



Mi a szoftver?

Hogyan fejlesztik a szoftvert?

Mi segítheti a szoftver-tervezést és -fejlesztést?

Mit csinál a programozó?

Milyen szolgáltatásai vannak az operációs rendszernek?

A könyv ezen utolsó fejezetében megismeri az operációs rendszer jelentőségét.

Megtanulja azt, hogy

- mik az operációs rendszer egyes komponenseinek funkciói, és milyen kényelmet nyújtanak ezek a számítógép felhasználóinak;
- hogyan vezérli az operációs rendszer a felhasználói programok futását;
- milyen módon kell a gépet üzemeltetni, és mit kell az operációs rendszernek tudnia ahhoz, hogy a kívánt üzemmódok megvalósíthatóak legyenek.

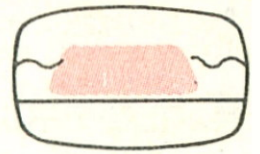
Az operációs rendszer
szolgáltatásainak
áttekintése

Az operációs rendszer
komponensei

Az operációs rendszer
szolgáltatásai
programvégrehajtás
közben

Üzem módok

Az operációs rendszer szolgáltatásainak áttekintése



Az előző fejezetben leírtuk a programfejlesztés egyes szakaszait. Eközben feltételeztük, hogy a programot párbeszédéses üzemmódban kódolják, és hogy a kódolt programot ezután lefuttatják.

A dolog ennyire nem természetes, mivel a felhasználó kódolt programja nem elegendő ahhoz, hogy a hardvert működtesse.

Ahhoz, hogy a felhasználói programok* futását egyáltalán lehetővé tegyük, még egy további programrendszerre van szükség, amelynek fontosabb programjai állandóan a számítógépben vannak tárolva. Ezt a programrendszert nevezik a számítógép **operációs rendszerének** (Operating System: OS).

■ Az operációs rendszer *vezérli a felhasználói programok futását.*

Eközben az operációs rendszer arról is gondoskodik, hogy a számítógép valamennyi egysége, amelyet a felhasználói program használ, megfelelően együttműködjék.

■ Az operációs rendszer *koordinálja a hardvert.* A számítógép operációs rendszere teszi lehetővé azt is, hogy gyakran több felhasználó is tud egy és ugyanazon gépen dolgozni.

■ Az operációs rendszer *koordinálja a különböző felhasználók hozzáférését a számítógéphez.*

Az operációs rendszernek köszönhető az is, hogy a felhasználó pl. egészen egyszerű módon tud egy terminálon át a számítógéppel kapcsolatba lépni, és erről a terminálról a gép valamennyi berendezését és szolgáltatását igénybe tudja venni.

■ Az operációs rendszer *megkönnyíti a munkát a számítógéppel.*

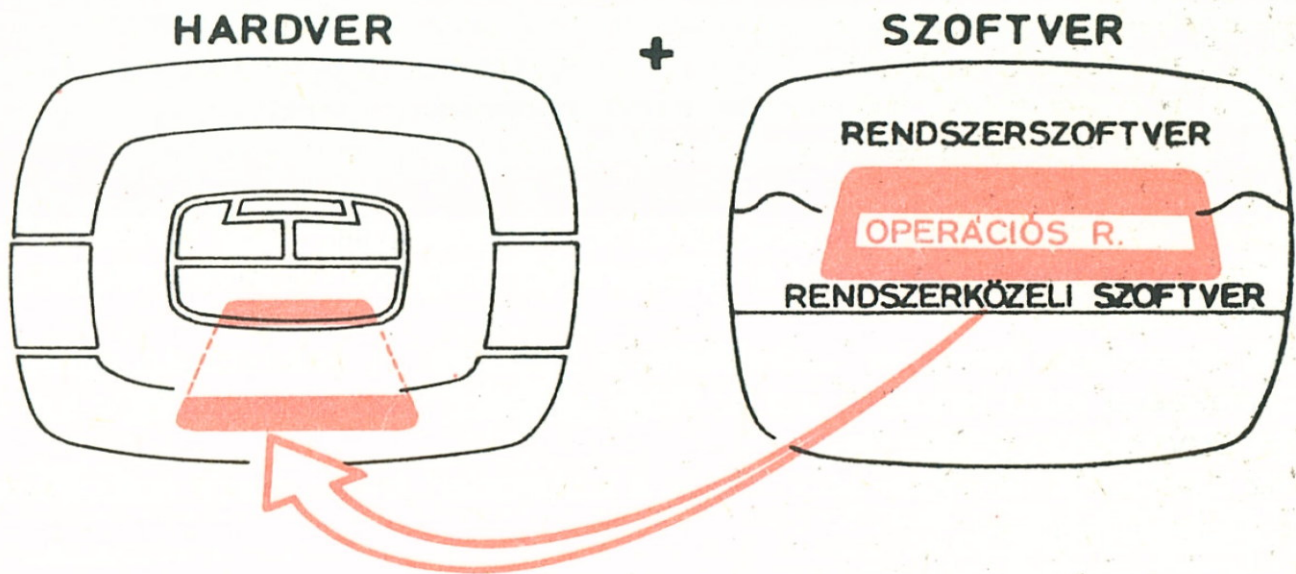
■ **Az operációs rendszer tehát koordinálja a hardvert, vezérli egy vagy több program futását, és gondoskodik arról, hogy a számítógép használata — amennyire csak lehetséges — egyszerű legyen.**

Az előzőekben általában általánosan a „program” kifejezést használtuk, függetlenül attól, hogy felhasználói programról vagy az operációs rendszer programjáról volt-e szó. Ezután meg kell tennünk ezt a megkülönböztetést. (Lektor.)

Az operációs rendszer összetevői:

- **Rendszerszoftver**, amely felügyeli és vezérli a hardvert és valamennyi igényelt program futását.
- **Rendszerközeli szoftver**, amely megkönnyíti a felhasználónak a számítógéppel való bánást, és tulajdonképpen csak kényelmi célja van.

A rendelkezésre álló rendszerszoftverból és a rendszerközeli szoftverből minden egyes installációhoz külön operációs rendszer generálható, amely illeszkedik a számítógéphardver kiépítéséhez és a tervezett alkalmazáshoz.





1. Jelölje meg valamennyi, az operációs rendszerre (OS) igaz állítást!

Az OS rendszerszoftverből
és rendszerközeli szoftverből tevődik össze.

Az OS vezérli a felhasználói programok futását.

Az OS koordinálja a hardvert.

Az OS koordinálja a különböző felhasználók
géphez férését.

Az OS megkönnyíti a munkát a számítógépen.

Az OS különleges felhasználói feladatokat is megold.

Minden számítógéphez saját OS generálható.

2. Elképzelhető-e számítógép operációs rendszer nélkül?

Igen. Nem.



Válaszok

1. Az operációs rendszerre vonatkozó alábbi állítások helyesek:

- Az OS rendszerszoftverből és rendszerközeli szoftverből tevődik össze.
- Az OS vezérli a felhasználói programok futását.
- Az OS koordinálja a hardvert.
- Az OS koordinálja a különböző felhasználók géphez férését.
- Az OS megkönnyíti a munkát a számítógépen.
-
- Minden számítógéphez saját OS generálható.

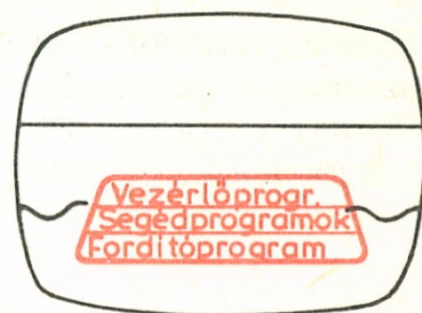
2. Igen.

A fix programozású számítógépeknek nincs szükségük operációs rendszerre. A programozható gépekhez ezzel szemben feltétlenül kell operációs rendszer, amely vezérli és ellenőrzi a felhasználói programok futását.

Az operációs rendszer komponensei

Minden egyes operációs rendszer — függetlenül attól, hogy melyik gép-konfigurációra generálták — három komponensből áll:

- a **vezérlőprogramból**, amely a számítógép részeinek együttműködését és a programok futását szervezi;
- a géptermi **segédprogramokból** (utility program), amelyek a számítógép használata közbeni segítségnyújtásokat adják, amennyiben átveszik a visszatérő rutinmunkákat és
- a **fordítóprogramokból**, amelyek a kódolt programokat lefordítják, azaz a gépi utasítások sorozatára alakítják át, hogy azokat a hardver „megértse”.



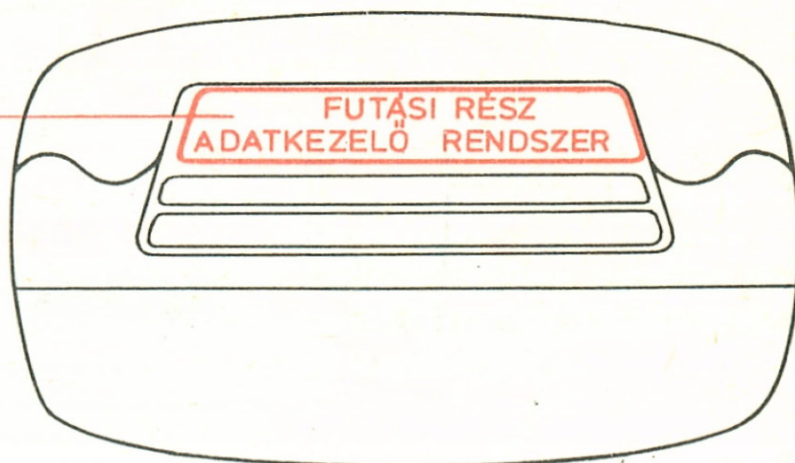
A vezérlőprogram

A **vezérlőprogram** feladata, hogy a futtatandó programok vezérlését átvegye, azaz koordinálja és ellenőrizzé a programok futását a számítógépben, és gondoskodik a számítógép egységeinek szabályos együttműködéséről.

A vezérlőprogram legfontosabb részei*:

- a **futási rész**,
- az **adatkezelő rendszer**.

VEZÉRLŐ-PROGRAM



* A megadott felosztás eltér a szokásostól. Az egyes részeket többnyire az erőforrás-kezelési vagy a folyamatmodell alapján szokás meghatározni és elnevezni (pl. tárkezelés, processzorkezelés, be-, ill. kivitelvezés stb.). (Lektor.)

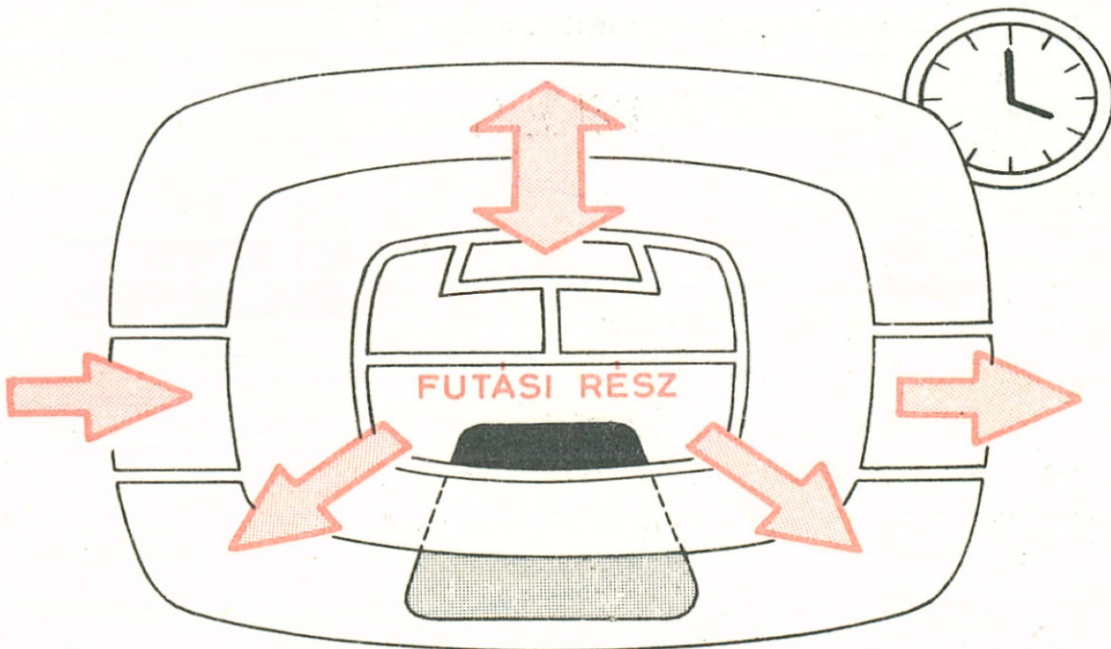
A futási rész

A **futási rész** az operációs rendszernek az a része, amely a programok végrehajtásának indítását és befejezését vezérli és ellenőrzi. Feladataihoz tartozik pl., hogy a program számára az operatív tárból a szükséges helyet biztosítsa, és a megfelelő be- és kiviteli berendezéseket a program rendelkezésére bocsássa.

A futási rész kezdeményezi továbbá az adatok be- és kivitelt, felismeri a fellépő hibákat, és esetleg automatikusan elhárítja azokat.

A program végrehajtását figyelemmel kell kísérni. Ha egyidejűleg több felhasználó dolgozik a számítógépen, akkor a futási rész végzi a processzor-hozzárendelést, miközben előre megadott fontossági sorrendet (prioritásokat) is figyelembe vesz.

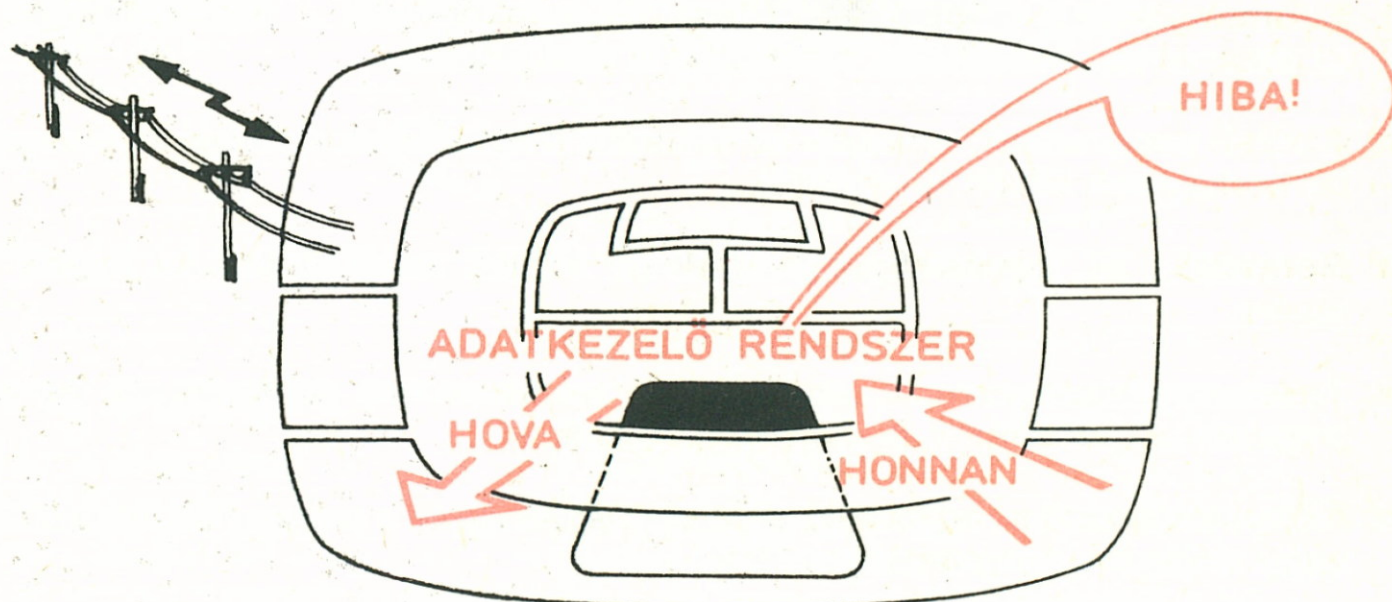
Ezeken túlmenően a vezérlőprogram futási része megkönnyíti a gépkezelő számára a számítógép működtetését azáltal, hogy koordinálja a gépkezelői tevékenységeket. Pl. a számítógép valamennyi olyan üzenetét, amelyek a mágnesszalag-egységekre vonatkoznak, egy külön kezelőpultra lehet kiadatni.



Az adatkezelő rendszer

Az adatkezelő rendszernek az a feladata, hogy vezérelje és ellenőrizze az adatoknak háttértárakon történő tárolását. A felhasználót tehermentesíti minden olyan problémától, amely állományok létesítésével, megváltoztatásával és törlésével jár a háttértárakon, ill. amelyeket az adatok visszakeresése okoz. Egy állomány létesítésekor pl. a felhasználónak nem kell azzal törődnie, hogy adatai melyik lemez melyik hengerének melyik sávján helyezkednek el. Elegendő egy rövid parancs (pl. „írás” vagy „olvasás”) és az adatkezelő programjai átveszik a szabad hely megkeresésének vagy az adatok visszanyerésének bonyolult feladatát.

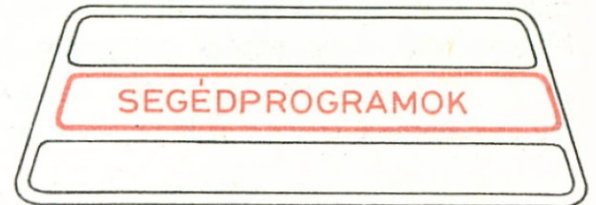
További példa az adatkezelő szolgáltatásaira: Egy állomány megnyitásakor a felhasználó megadhatja, hogy az abban tárolt információk a számítógép valamennyi felhasználója számára hozzáférhetők legyenek-e, vagy a hozzáférés jogosultságát egy kulcsszó megadásával kelljen igazolni. Az utóbbi esetben az adatkezelő minden egyes felhasználó jogosultságát ellenőrzi, aki ehhez az állományhoz fordul.



A futási rész és az adatkezelő rendszeren kívül a vezérlőprogram még további programmodulokat is tartalmaz, amelyek pl. speciálisan az **adatátvitellel** vagy a hardver-, szoftver-, ill. gépkezelői hibákkal foglalkoznak.

Segédprogramok

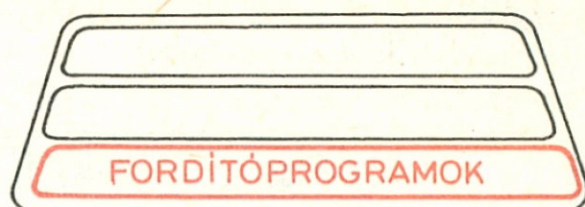
A rendszerközeli szoftverhez sorolt segédprogramok a számítógép üzemeltetésének értékes segédeszközei. Ezek a programok futtatható állapotban a programkönyvtárban vannak, és szükség esetén onnan lehívhatók.



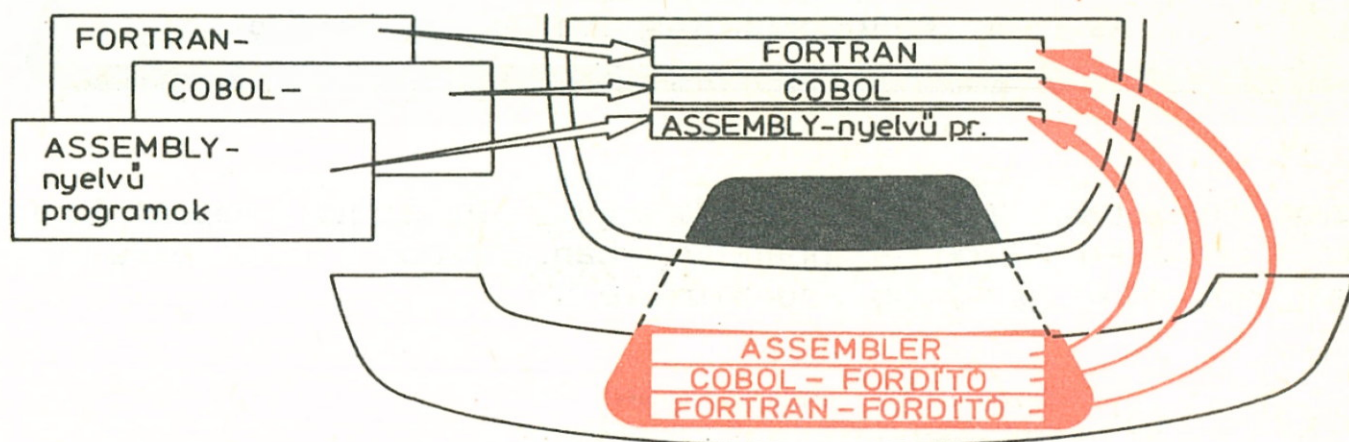
A legfontosabb segédprogramok:

- **Segédprogramok**, amelyek különböző programrészeket végrehajtható egységgé állítanak össze.
- **Töltőprogramok**, amelyek a végrehajtandó programot behozzák az operatív tárba.
- **Állománykezelő programok**, amelyek megkönnyítik a felhasználónak az állományok felépítését, módosítását és törlését.
- **Átviteli programok**, amelyek az adatokat a háttértárak és a be- és kiviteli berendezések között továbbítják.
- **Rendező-, ill. egyesítőprogramok** az adatok rendezésére és egyesítésére.
- **Teszt-segédprogramok**, amelyek pl. tesztadatokat állítanak elő és a program végrehajtását lépésenként követik.
- **Könyvtárkezelő programok**, amelyek végrehajtják a programok felvételét vagy törlését a programkönyvtárból.
- **Segédprogramok**, amelyek pl. statisztikákat készítenek a berendezések foglaltságáról stb.

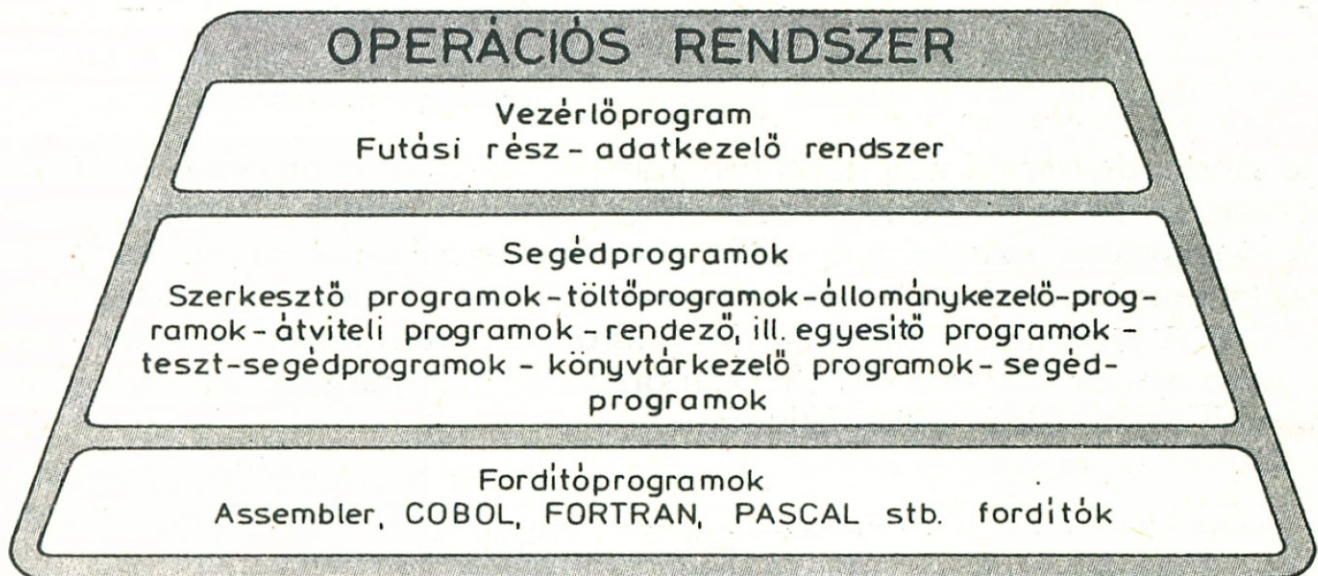
Fordítóprogramok



A **fordítóprogramok** egy adott programnyelven megírt programok lefordítását végzik el egy bizonyos számítógép gépi nyelvére. Problémaorientált programnyelvek esetén a fordítót compilernek (magas szintű fordítónak), a géporientált programnyelv esetén assemblernek (alacsony szintű fordítónak) nevezik. A program fordítása során a fordítóprogram ellenőrzi, hogy a bevitt program eleget tesz-e a használt programnyelv formai szabályainak (szintaxisának) és **fordítási listát** állít elő.



A következő ábra még egyszer összefoglalja az operációs rendszer legfontosabb komponenseit.



Az operációs rendszert a gyártó rendszerint együtt szállítja a géppel. Az operációs rendszert részben az operatív tárban, részben a berendezés háttér-tárán tárolják, és az mindig rendelkezésre áll.



1. Az operációs rendszer mely része gondoskodik a számítógép egyes részeinek együttműködéséről és a programok végrehajtásának vezérléséről?

.....
Az operációs rendszernek mely programjai végzik el a gyakran ismétlődő rutinfeladatokat?

.....
Az operációs rendszer mely programja alakítja át a kódolt programot a gépi utasítások sorozatává?

2. A futási rész (F) és az adatkezelő rendszer (A) a szervezőprogram legfontosabb részei.

	F	A
Melyik modul vezérli és ellenőrzi a programok végrehajtásának indítását és befejezését?	<input type="radio"/>	<input type="radio"/>
Melyik modul kezeli az adatokat a háttértárakon?	<input type="radio"/>	<input type="radio"/>
Melyik modul gondoskodik a processzor hozzárendeléséről?	<input type="radio"/>	<input type="radio"/>
Melyik modul végzi az adatok visszakeresését?	<input type="radio"/>	<input type="radio"/>

3. A következő programok közül melyek az operációs rendszer segédprogramjai?

Szerkesztőprogramok. <input type="radio"/>	Rendező-, ill. egyesítőprogramok. <input type="radio"/>
Töltőprogramok. <input type="radio"/>	Teszt-segédprogramok. <input type="radio"/>
Átviteli programok. <input type="radio"/>	Állománykezelő programok. <input type="radio"/>
	Könyvtárkezelő programok. <input type="radio"/>

4. Nevezzen meg néhány fordítóprogramot!

.....
.....



Válaszok

1. A számítógép egyes részeinek együttműködéséről és a programok végrehajtásának vezérléséről a **vezérlőprogram** gondoskodik.

Az ismétlődő rutinfeladatokat a segédprogramok veszik át.

A kódolt programok átalakítását gépi utasítások sorozatára a **fordítóprogramok** végzik.

2. A helyes egymáshozrendelés a következő:

F A

- | | | |
|----------------------------------|----------------------------------|---|
| <input checked="" type="radio"/> | <input type="radio"/> | A programok végrehajtásának indítását és befejezését a futási rész vezérli. |
| <input type="radio"/> | <input checked="" type="radio"/> | Az adatokat a háttértárakon az adatkezelő rendszer kezeli. |
| <input checked="" type="radio"/> | <input type="radio"/> | A processzor hozzárendeléséről a futási rész gondoskodik. |
| <input type="radio"/> | <input checked="" type="radio"/> | Az adatokat az adatkezelő rendszer keresi vissza. |

3. Valamennyi felsorolt program a segédprogramok csoportjába tartozik.

- | | |
|---|---|
| <input checked="" type="checkbox"/> Szerkesztőprogramok. | <input checked="" type="checkbox"/> Rendező-, ill. egyesítőprogramok. |
| <input checked="" type="checkbox"/> Töltőprogramok. | <input checked="" type="checkbox"/> Teszt-segédprogramok. |
| <input checked="" type="checkbox"/> Átviteli programok. | <input checked="" type="checkbox"/> Állománykezelő programok. |
| <input checked="" type="checkbox"/> Könyvtárkezelő programok. | |

4. **Assembler, COBOL-fordító, FORTRAN-fordító, P1/1-fordító stb.**

Megnevezhetnénk bármely más programnyelv fordítóprogramját is.

Az operációs rendszer szolgáltatásai programvégrehajtás közben

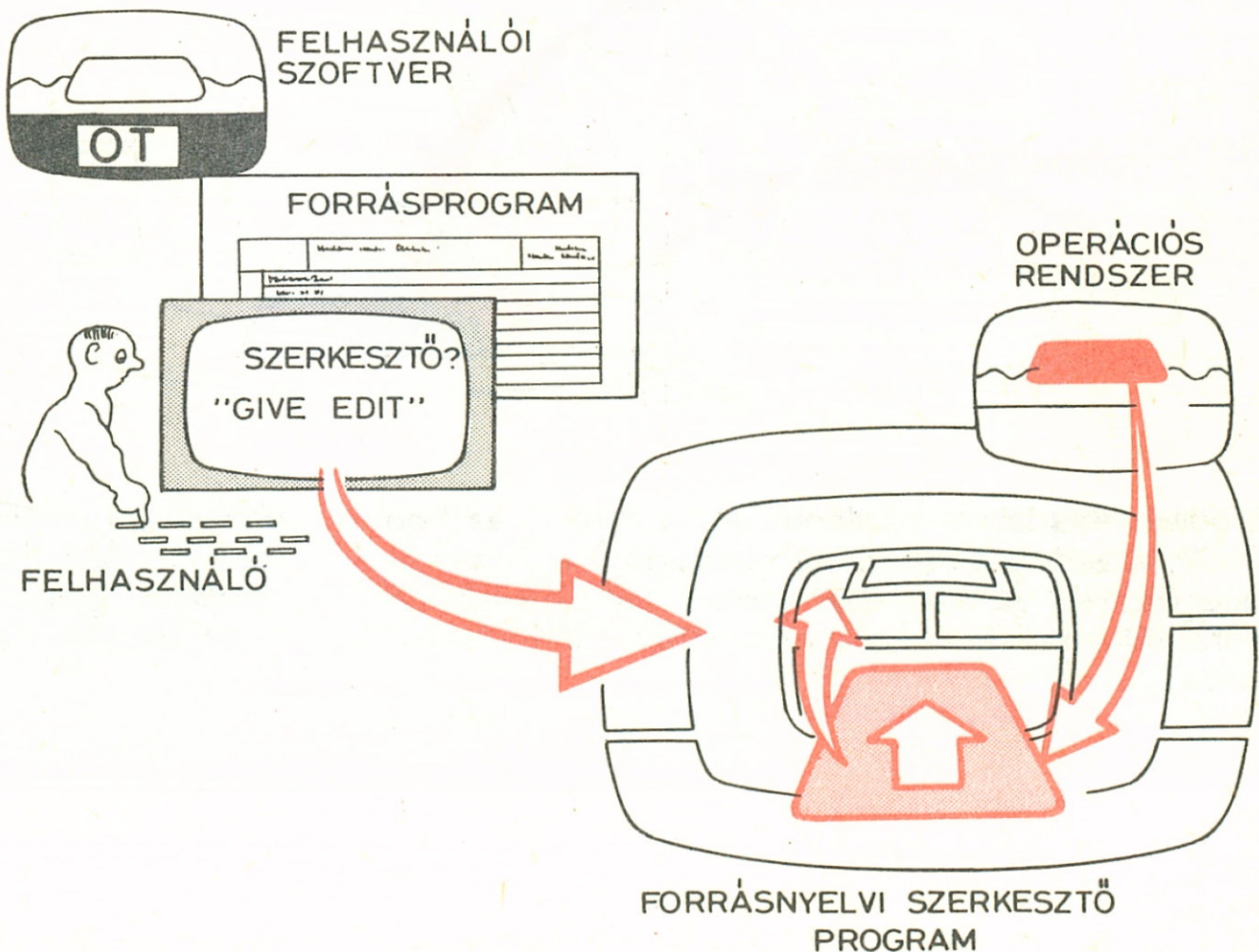
Miután megismertük az operációs rendszer komponenseit, vizsgáljuk meg, hogy milyen kényelmet nyújt a számítógép felhasználójának az operációs rendszer a felhasználói program bevitelétől annak végrehajtásáig!

Állománykezelő

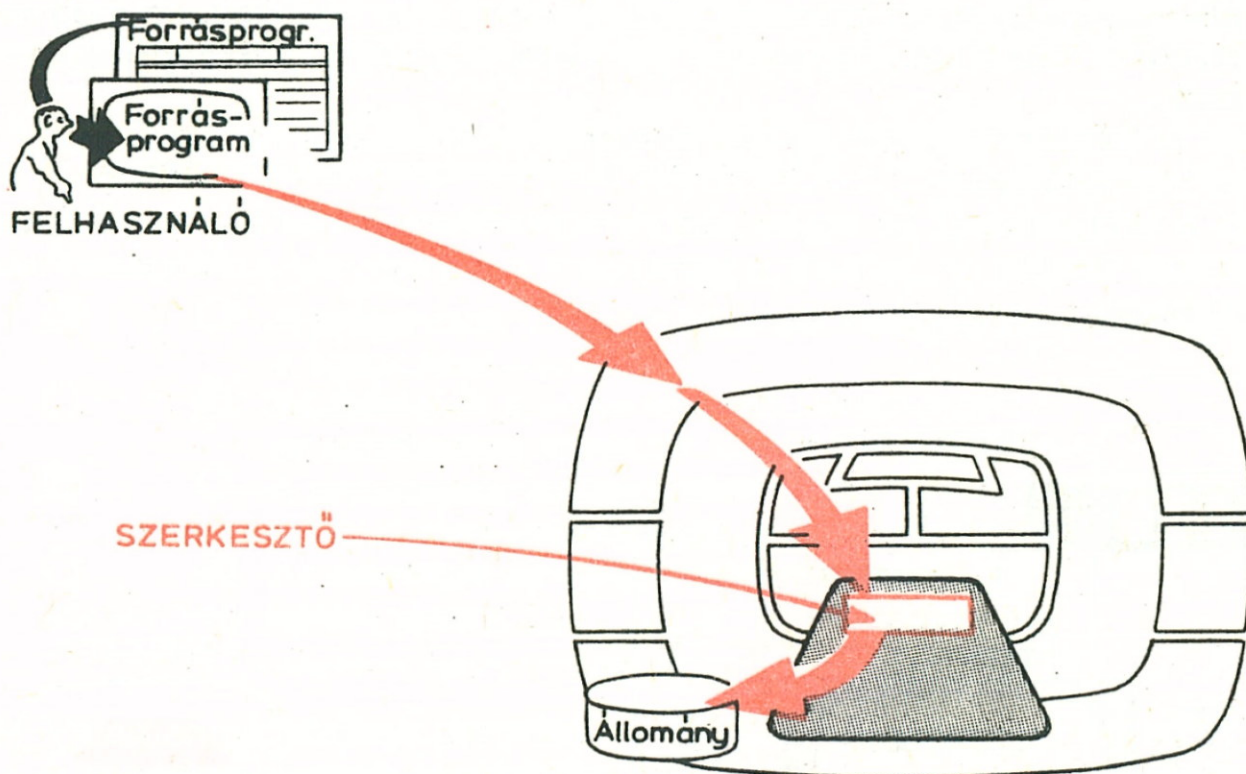
Kísérjük végig egy olyan program útját, amelyet már kódoltak, de még nincs gépi nyelvre lefordítva. Egyébként az ilyen programot **forrásprogramnak** nevezzük.

Először a forrásprogramot be kell olvasni, az operációs rendszer ekkor lép először működésbe.

Mielőtt a programozó az egyes programutasításokat beviszi — a fejlettebb rendszerek esetén —, először a terminálról a forrásnyelvi szövegszerkesztőt (editor) hívja.



A **szövegszerkesztő** az operációs rendszer egyik programja, pontosabban fogalmazva segédprogramja, amely segít a programozónak egy állomány felépítésében. Ezért ezt **állománylétesítő programnak** is nevezzük. Meghatározott neve van (pl. EDIT), amelyen a programozó hívhatja. Ez a megnevezés operációs rendszerenként más lehet. A szerkesztő — hívását követően — várja a programutasításokat. Az állománykezelő átveszi a beírt utasításokat, és egy állományban tárolja őket. Az állomány egyértelmű nevet kap, amelyet a programozó választ a programja számára. Ezen a néven az állományt mindenkor meg lehet találni (pl. szükség lehet erre, ha változtatásokat kell a programunkban végrehajtani).

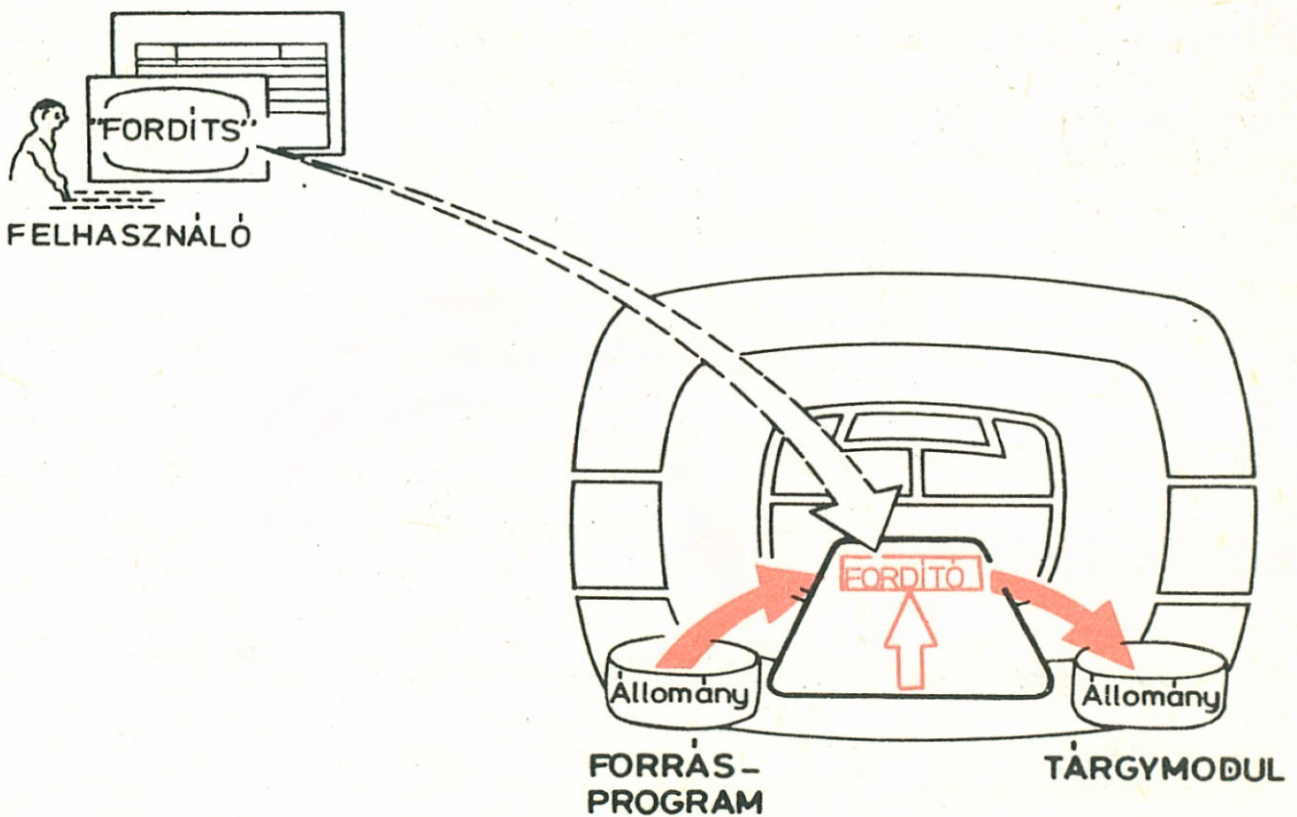


A program feladata — általános értelemben — az, hogy állományokat kezeljen: pl. már létező állományokat olvasson mágneslemeziről, ezeket módosítsa, ill. kiegészítse és ismétellen mágneslemezre írja stb. Segítségével nemcsak programállományt, hanem mindenfajta adatállományt is lehet létesíteni és módosítani.

Fordító

Miután a programot bevittük és egy állományban tároltuk, azt le kell **fordítani**.

A forrásprogramnak a gép nyelvére való lefordításához a tárolt **fordító-programot** kell hívni, amely ugyancsak az operációs rendszer része. A felhasználó parancsot ad az operációs rendszernek, és — az alkalmazott programozási nyelvtől függően — assembler vagy compiler típusú fordítóprogram kezd futni, majd az adatokat (jelen esetben a lefordítandó forrásprogramot) feldolgozza, azaz a forrásprogramot gépi nyelvre fordítja le.

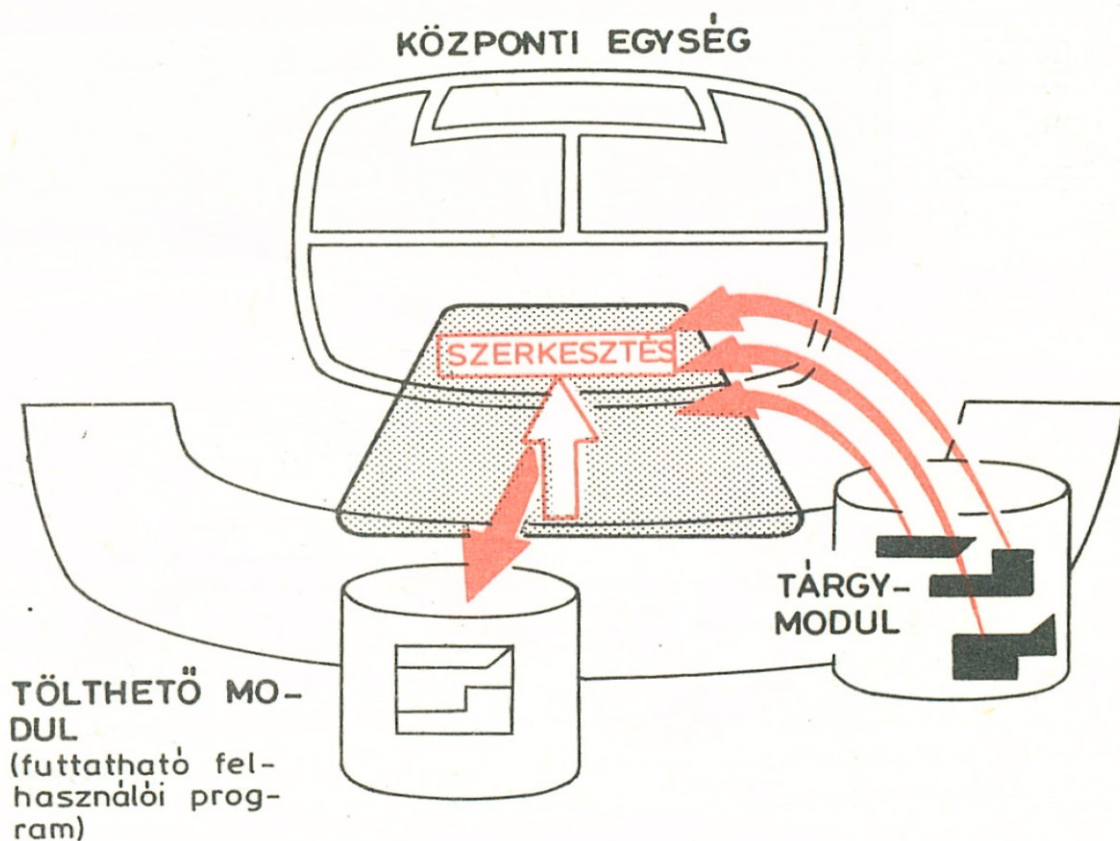


Ennek a fordításnak az eredménye rendszerint nem futásra kész program, mert a végrehajtandó program általában több modulból áll, és ezeket egyenként le kell fordítani. A lefordított programokat vagy programrészeket **tárgymoduloknak**, ill. **programmoduloknak** nevezik, és egy **modulkönyvtárban** tárolják, ahonnan szükség esetén hívhatók. A nagy és bonyolult programokat különböző programozók részenként készítik el, lefordítják, és a modulkönyvtárban tárolják.

Szerkesztő

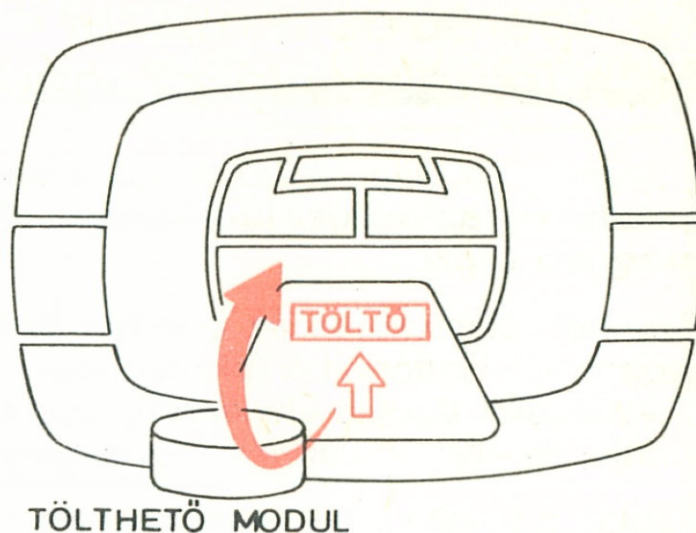
A tárgymodulok tehát azonos nyelven, a gépi nyelven rendelkezésre állanak. Ezeknek a — szélső esetben — különböző programozási nyelven megírt és más-más fordítóprogrammal lefordított részeknek egy futásra kész teljes programmá való összeállítására az operációs rendszerből újabb programot kell hívni: a **szerkesztőprogramot**.

A szerkesztő tehát olyan program, amely több, egymástól függetlenül írt programrésznek egy programmá való összeszerkesztését végzi.



Töltőprogram

A **töltőprogram** ugyancsak az operációs rendszer segédprogramja. Feladata az, hogy szabad helyet keres az operatív tárba és a megadott tárterületre beolvassa a modult. Ezenkívül utasítást ad a számítógépnek, hogy számítsa ki azokat a belső címeket, amelyeken a program utasításai és a hozzájuk tartozó adat-területek majd elhelyezkednek.



Csak a felhasználói modulok betöltése után kezdődhet a tulajdonképpeni **élesfutás** a felhasználó adataival.

A következő ábra összehasonlítja az éppen **futó programokat** és az általuk **feldolgozandó adatokat**. A nyíl a felhasználói program útját mutatja, az egyes feldolgozási lépéseken át.

Itt a felhasználói program kezelése először olyan, mint ha „közönséges adat” lenne. Csak ha mint betölthető modul jelenik meg, akkor indítható el mint felhasználó program.

Futó program	Feldolgozott adatok
Állománykezelő	A felhasználói program be- gépelt utasításai
Fordítóprogram	Forrásprogram
Szerkesztő program	Tárgymodul
Töltő program	Tölthető modul
Elindított felhasználói program (tölthető modul)	Felhasználói adatok (pl.: egy elszámolási időszak bérelszámolási adatai)

Az operációs rendszer és a felhasználói program együttműködése

Valamennyi szükséges tevékenység súrlódásmentes lebonyolítását a **vezérlő-program** végzi:

Szabad helyet keres az operatív tárban, és ide betölti az állománykezelő programot. Háttértáron (pl. mágneslemezen) helyet keres a programállomány létesítésére. Ezután elegendő processzoridőt biztosít az állománykezelőnek, indítja, és ellenőrzi annak végrehajtását.

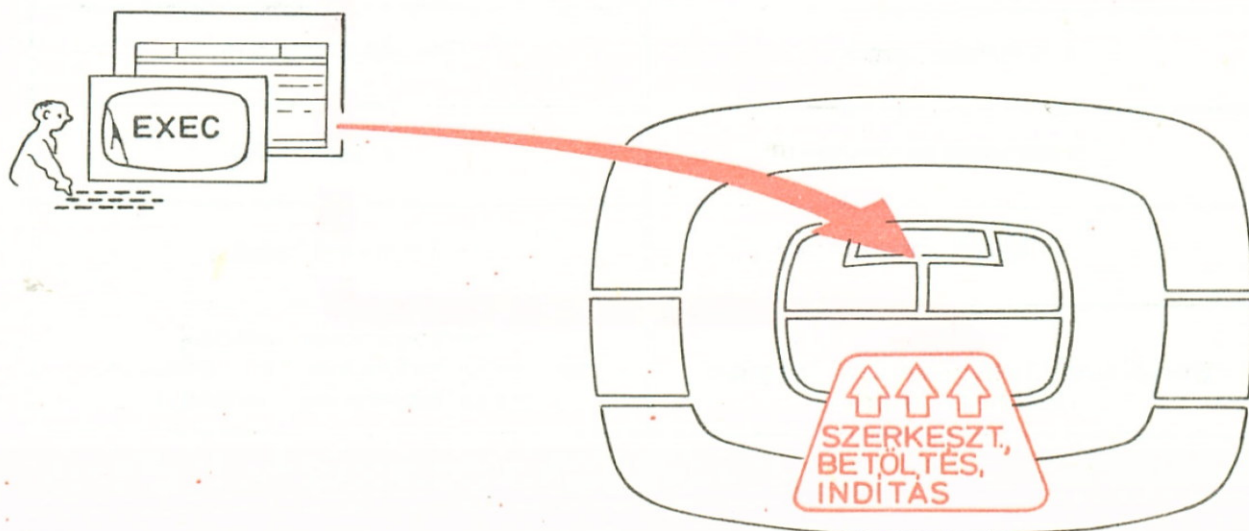
Ezután a fordító kerül be az operatív tár egy szabad területére, és ha rendelkezésre áll a forrásprogram is, akkor a vezérlőprogram irányítja és ellenőrzi a fordító futását.

Ugyanez történik a szerkesztő- és a töltőprogrammal is egészen addig, míg a felhasználói program futásra kész modul alakjában előáll.

Amikor a programmodul elindul, a vezérlőprogram vezérli és ellenőrzi a felhasználói program végrehajtását, valamint rendelkezésre bocsátja a feldolgozandó felhasználói adatokat.

Ha a felhasználói program tesztadatokkal fut, **tesztfutásról** van szó; ha viszont a letesztelt program már „valódi” adatokat dolgoz fel, akkor **élesfutásról** beszélünk.

A több kényelmet nyújtó operációs rendszerek a program szerkesztését, töltését és indítását valamennyi ehhez tartozó tevékenységgel együtt egy utasításra végrehajtják. Az utasítást a felhasználó a terminálról adja be.



Az operációs rendszer szolgáltatásai
programvégrehajtás közben

Kérdések
a 173...178. oldalakhoz



1. A vezérlőprogram mellett az operációs rendszer mely programjai lépnek sorban működésbe, amíg a felhasználói programot felhasználói adatokkal el lehet indítani?
 1.
 2.
 3.
 4.

2. Mely program segít a programállomány felépítésében?
.....

3. A tárgymodulok már eleve össze vannak szerkesztve?
Igen. Nem.

4. Melyik fordítóprogram lép működésbe, ha a forrásprogramot részben assembly nyelven, részben COBOL nyelven írták meg?
.....

5. Hogyan nevezik a már összeszerkesztett programot?
Tárgymodul Tölthető modul Szerkesztőmodul

6. Az operációs rendszer melyik segédprogramjának feladata:
Az operatív tár megadott területére beolvasni a betöltendő modult, a belső címeket kiszámítani, ahová a program utasításai és a hozzájuk tartozó adatok kerülnek?
.....

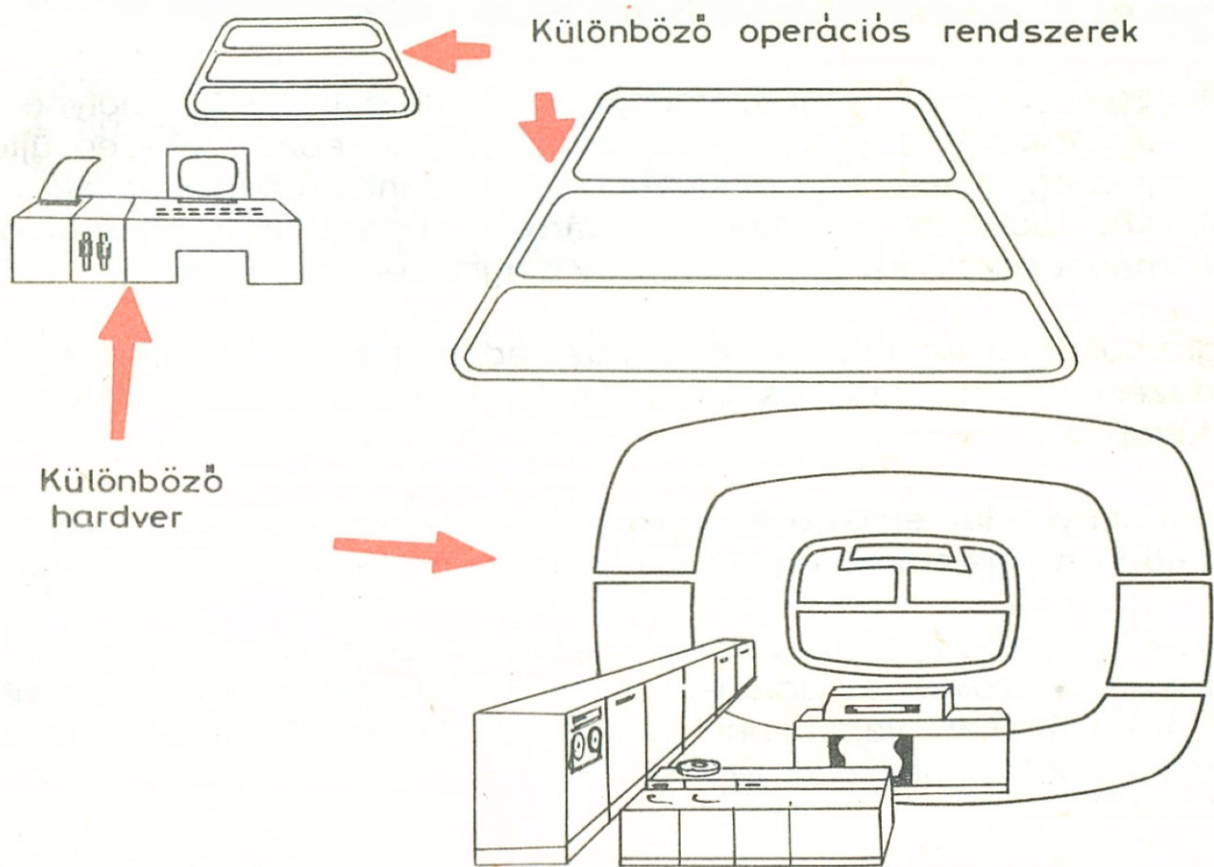
7. Milyen „adatokat” dolgoz fel a fordítóprogram?
.....



Válaszok

1. A vezérlőprogram mellett az operációs rendszerhez tartozó alábbi programoknak kell sorban lefutniuk ahhoz, hogy a felhasználói program felhasználói adatokkal elindítható legyen:
 1. **Állománykezelő.**
 2. **Fordító.**
 3. **Szerkesztő.**
 4. **Töltő.**
2. A programállomány felépítésében az *állománykezelő* segít.
3. Nem.
Az összeszerkesztett tárgymodulokat betölthető moduloknak nevezzük.
4. Az assembly, ill. COBOL nyelven megírt forrásprogramrészek lefordításához **assembler** és **COBOL-compiler (fordítóprogram)** kell.
5. Egy összeszerkesztett programot
 Tölthető modulnak nevezünk.
6. A felsorolt feladatokat a **töltőprogram** végzi el.
7. A fordítóprogram bemenő adatai a **forrásprogram utasításai**.

Üzem módok



Az operációs rendszereket több szempont szerint osztályozhatjuk, egyrészt a **hardver** alapján — pl. egy kishszámítógép, amelyhez egyetlen perifériaként csak egy hajlékony mágneslemezes tár tartozik, más operációs rendszert igényel, mint egy olyan nagyberendezés, amely több mágnesszalagos és mágneslemezes egységgel dolgozik —, másrészt az **üzemmód** alapján, amelyben a számítógépnek működni kell. A szükséges üzemmódot ugyanakkor alapvetően a számítógép által megoldandó feladatok határozzák meg.

Az alkalmazott **üzemmód** megszabja, hogy a felhasználók milyen módon használhatják a számítógépet.

A legfontosabb üzemmódok:

- a **kötegelt üzemmód** és
- a **valós idejű üzemmód**.

Kötegelt üzemmód

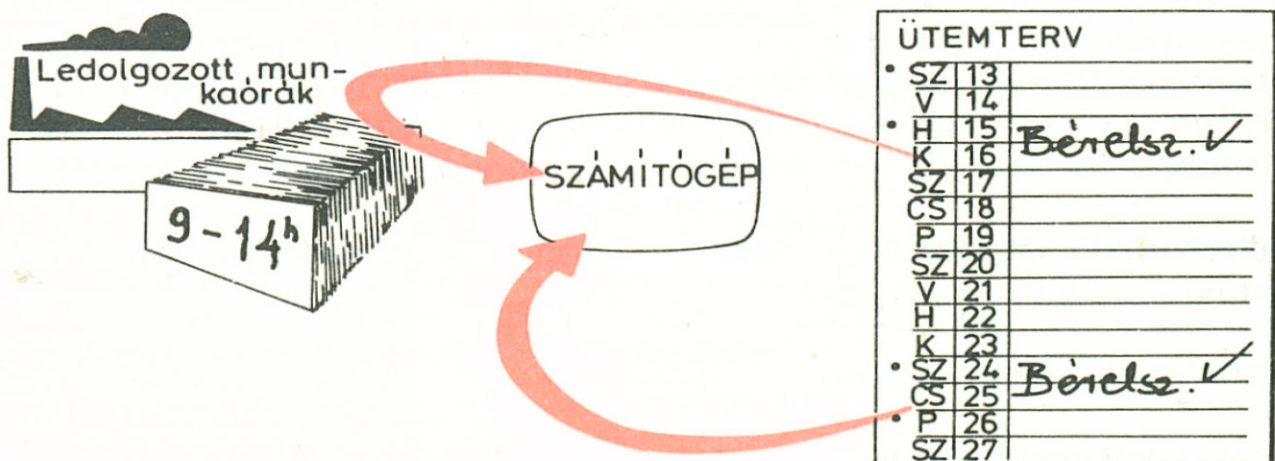
Korábban a programot és a feldolgozandó adatokat lyukkártyára lyukasztották és ezt a „köteget” leadták a számítóközpontba; innen származik a **kötegelt feldolgozás** elnevezés (angolul: **batch processing**).

Ma a kötegelt üzemen olyan számítógép-üzemmódot értünk, amelynél valamennyi ugyanazzal a programmal feldolgozandó adatot összegyűjtenek, majd azokat egyszerre, egy meghatározott időpontban feldolgozzák. Ma az adatrögzítés többnyire már nem lyukkártyára, hanem más adathordozóra, mint pl. mágnesszalagra vagy hajlékony mágneslemezre történik.

A programokat és az adatokat tartalmazó adathordozókat megfelelő beviteli berendezésekbe helyezik, és azokat meghatározott időben beolvassák a számítógépbe.

Ha valamennyi adat feldolgozása elkészült és az eredmények a programnak megfelelően megjelentek, egy másik „program- és adatköteg” következik.

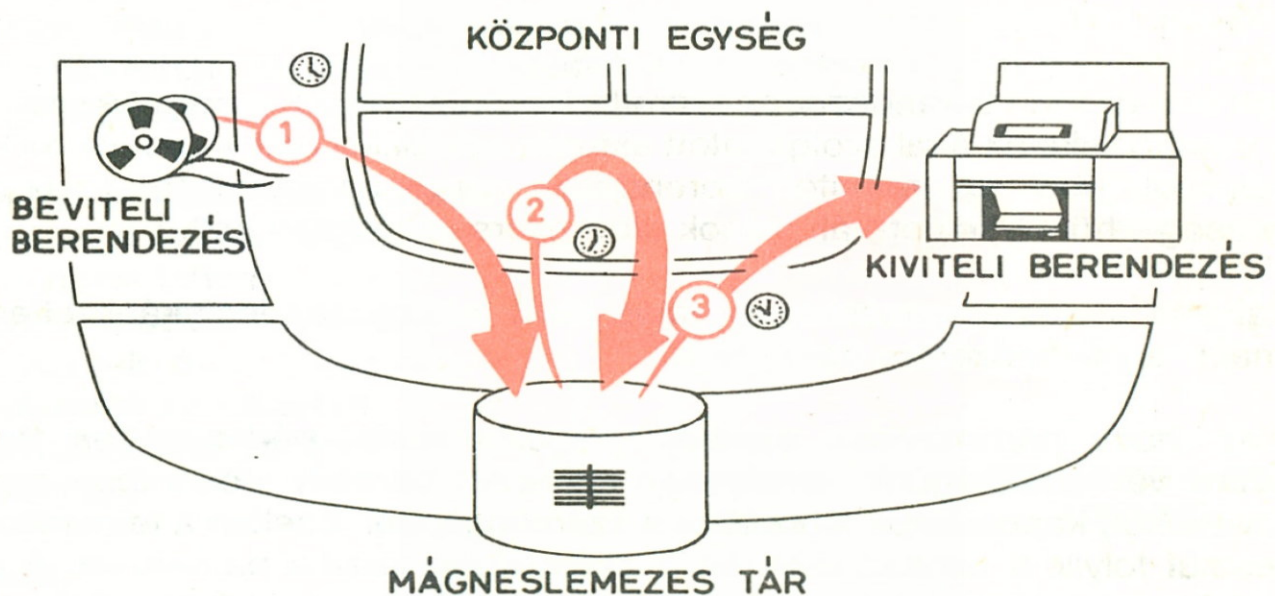
A kötegelt üzemmód tipikus példája egy olyan vállalat bérszámfejtése, ahol összegyűjtik a szükséges adatokat (pl. a ledolgozott órákat, a bértarifákat, az adószázalékokat) valamennyi dolgozóra, és a hónap végén egyetlen programfutással feldolgozzák azokat.



A kötegelt üzemmód kiterjesztett változata a **spooling-üzem**, amelynél a felhasználók tetszőleges időpontokban vihetnek be adatokat és parancsokat a számítógépbe. Ezeket előbb háttérre írják és akkor dolgozzák fel, amikor szabad kapacitás adódik. Ezt az üzemmódot azért alakították ki, hogy az időfeltételek ne függjenek többé az érintett felhasználóktól.

A spooling-üzemnél a parancsokat és az adatokat átmenetileg olyan tárokon tárolják, amelyek a központi egységgel on-line (közvetlen) kapcsolatban állnak (pl. mágneslemez). Ha a számítógépnek szabad kapacitása van, az operációs rendszer a felhasználói programot és az átmenetileg tárolt adatokat beolvassa a központi egységbe és gondoskodik feldolgozásukról.

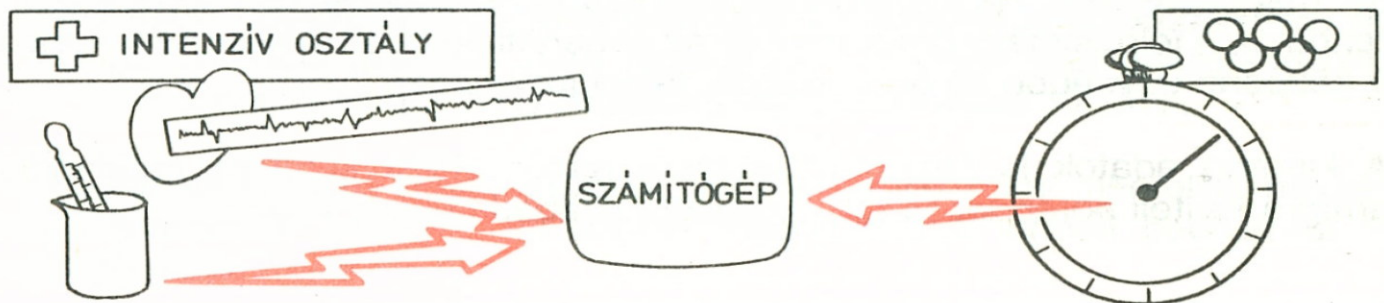
A kimenő adatok is átmeneti tárolásra háttértárakra kerülnek mindaddig, amíg a kiviteli berendezés (pl. nyomtató) szabaddá nem válik.



1. Bevitel tetszőleges időpontban;
a program és az adatok közbenső tárolása mágneslemezen.
2. Feldolgozás szabad központi egység esetén;
eredmények átmeneti tárolása mágneslemezen.
3. Az eredmények kivele a kiviteli berendezések
(pl. nyomtató) felszabadulásakor.

Valós idejű üzem

A valós idejű feldolgozás esetében a kötegelt feldolgozással szemben, az adatokat nem gyűjtik össze, hanem keletkezésükkor azonnal feldolgozzák azokat. A **valós idejű üzemmódot azonos idejű üzemmódnak (real-time üzemenek)** is nevezik.



Példa erre az ipari berendezések számítógépes irányítása, amelyet a számítógép mérőérzékelők által szolgáltatott aktuális adatokkal hajt végre. Ez történik mindenhol, ahol az irányított berendezés minden változására (nyomás-, sebesség-, hőmérséklet-változások stb.) gyorsan reagálni kell.

A vállalati adatfeldolgozásban a valós idejű feldolgozás leginkább elterjedt formája: a **párbeszédés üzemmód**.

Ahhoz, hogy párbeszédés módban dolgozhassunk, **párbeszédés terminálokra** van szükségünk, amelyeken keresztül bármely időpontban on-line (közvetlenül) kapcsolatba léphetünk a számítógéppel. Ezeken a terminálokön keresztül folyik a felhasználók és a számítógép közötti párbeszéd. A párbeszéd a megoldandó feladattól függően egymást követő kérdések és válaszok formájában történhet.

A párbeszédés üzemmód adott értelmezése szerint párbeszéd zajlik le akkor is, ha a terminálról csak egy parancs bevitele történik, amely elindítja a gépben tárolt programok egyikét és a számítógép a program lefutása után erről üzenetet küld vissza a terminálra.

Természetesen a valós idejű üzemet is megengedő operációs rendszernek nagyobb teljesítményűnek kell lennie, mint egy olyannak, amely csak kötegelt üzemmódban működik.

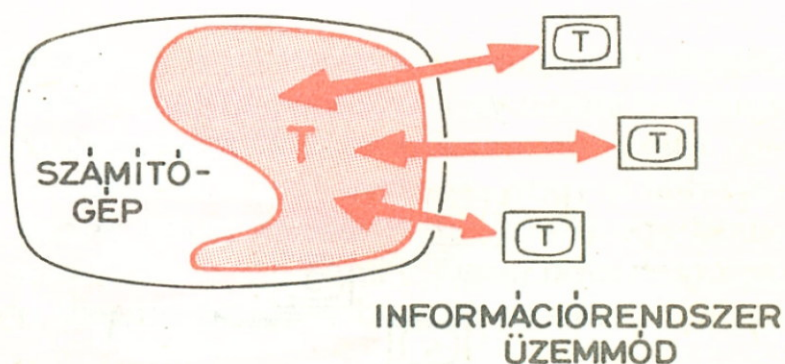
Párbeszédés üzem két módon valósulhat meg:

- mint egy **információrendszer üzemmód**, amelynél valamennyi felhasználó ugyanazt a programot és a hozzá tartozó adatokat használja, és
- mint **időosztásos üzemmód**, amelynél az egyes felhasználók saját programjaikkal és adataikkal úgy dolgozhatnak, mintha a számítógép mindegyik felhasználónak egyedül állna rendelkezésére.

Információrendszer üzemmód

Az információrendszer üzemmódban az operációs rendszer a felhasználót egy, rendszerint terjedelmes programmal szolgálja ki. A felhasználók tehát valamennyien ugyanazzal a programmal és adatbázissal dolgoznak. A felhasználók kérdéseket tehetnek fel, és a párbeszéd közben felvilágosításokat kapnak; miközben a tárolt adatokat is megváltoztathatják. Egy másik felhasználó tehát, aki valamivel később fordul ezekhez az adatokhoz, már az előző felhasználó által megváltoztatott adatokat találja ott.

A program megváltoztatására azonban a felhasználóknak soha nincs lehetőségük! A párbeszédés helyfoglaló rendszerek és a tájékoztató-rendszerek rendszerint információrendszerként működnek.



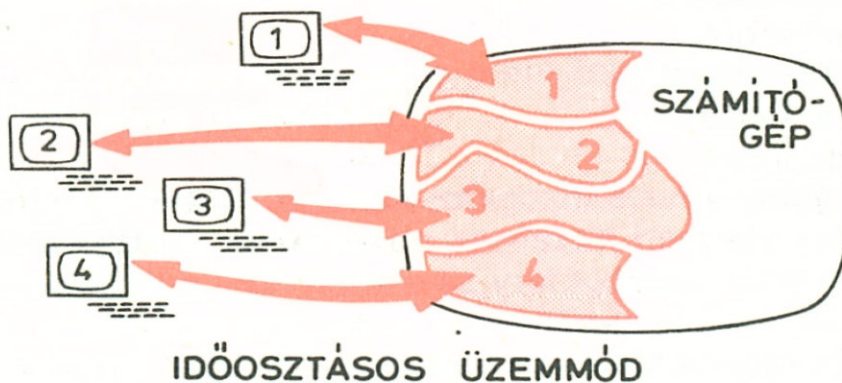
Példa:

Egy utazási irodának több városban van kirendeltsége, amelyekben egy-egy képernyős terminál ugyanahhoz a központi számítógéphez kapcsolódik. Ebben a rendszerben valamennyi kirendeltség egymástól függetlenül kérdéseket intézhet és válaszokat kaphat (pl. „hány szabad hely van még egy légi-járaton” stb.). A számítógépben tárolt felhasználói programnak, amely a repülőjegyekkel gazdálkodik, helyfoglalási igényeket lehet küldeni, a visszaigazolások vagy törlések pedig azonnal megjelennek a képernyőn. Mivel valamennyi terminál egy és ugyanazon programhoz és azonos adatokhoz fér hozzá, elkerülhető, hogy ugyanazt a helyet kétszer kiadják.

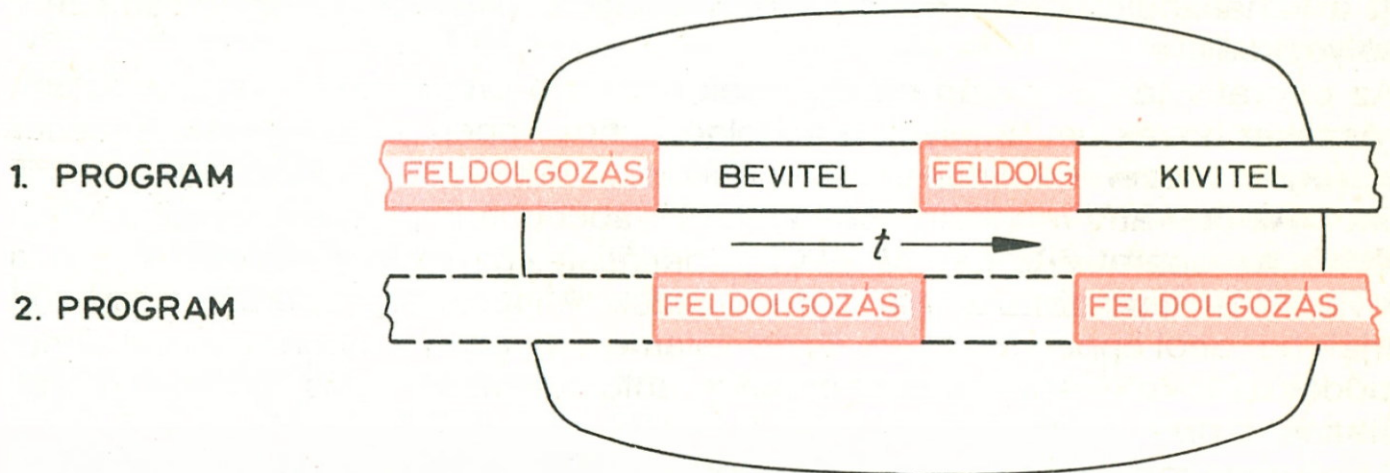
Időosztásos üzemmód

Ebben az üzemmódban sok felhasználó különböző helyekről egyidejűleg tudja ugyanazt a számítógépet használni. Az operációs rendszer valamennyi felhasználónak rendelkezésére bocsátja a teljes rendszerszoftvert és rendszerközeli szoftvert. Itt valamennyi felhasználó kizárólagosan használja saját programjait és saját adatait. Tehát programjait változtathatja, törölheti, vagy új programokat vihet be, ezeket fordíthatja, tesztelheti, ha szükséges javíthatja, tárolhatja és bármikor lefuttathatja. A felhasználó programjaival és állományaival úgy dolgozhat, mintha az egész számítógép csak az övé lenne.

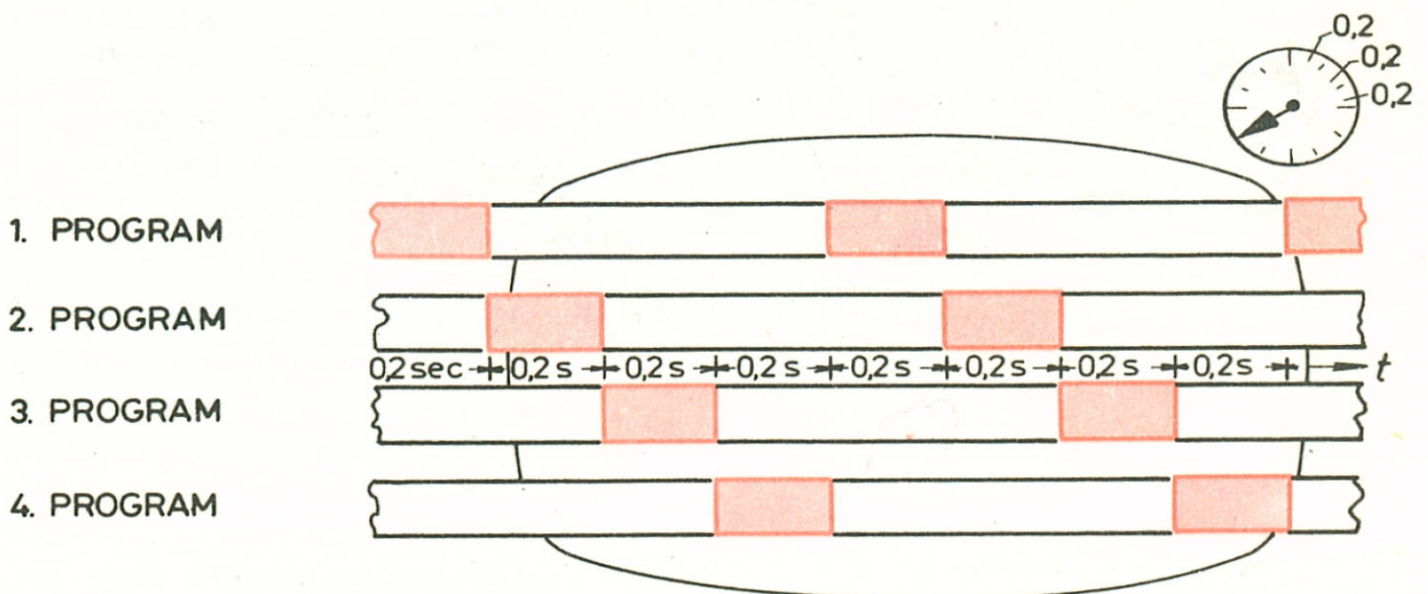
Tekintve, hogy ugyanazon a számítógépen egyszerre sok felhasználó dolgozik, és mivel az időosztásos üzemmódban — ellentétben az információrendszer üzemmóddal — különbözőek a felhasználók is, akik saját felhasználói programjaikkal dolgoznak, a számítógépnek **multiprogramozást** (angolul: multiprogramming) kell végeznie, azaz több programnak kell egyidejűleg futnia. Azonban a számítógép is csak egymás után tudja az egymást követő feladatokat elvégezni, ezt viszont nagyon gyorsan teszi, ezért mégis úgy tűnik, hogy több program egyidejűleg tud futni (de ez csak látszólagos és nem valóságos!).



Ismeretes, hogy minden program adatbevitelből, feldolgozásból és adatkivitelből áll. A központi egység érdemben csak a feldolgozás alatt működik, és mindig meg kell várnia, amíg egy bevétel és egy kivitel befejeződik. Az adat be- és kivitele perifériákon keresztül történik, amelyek lényegesen lassúbbak, mint a központi egység. Ezáltal a központi egység számára jelentős várakozási idők adódnak. Ezen idő alatt a központi egység hozzáfoghat más programok feldolgozásához.



Ahhoz, hogy a sok felhasználót látszólag egyidőben ki lehessen szolgálni, az egyes felhasználói programokat az operációs rendszer úgy osztja be, hogy pl. minden 0,2 másodpercben megszakítja a futó programot és másik felhasználó programját indítja. Ezek a váltások olyan gyorsak, hogy a felhasználók ebből semmit sem vesznek észre. Mivel minden felhasználó egy **bizonyos részidőt, osztott időt** kap, amelyben a központi egység neki dolgozik, **időszakos** (angolul: **time sharing**) üzemmódról is beszélhetünk.

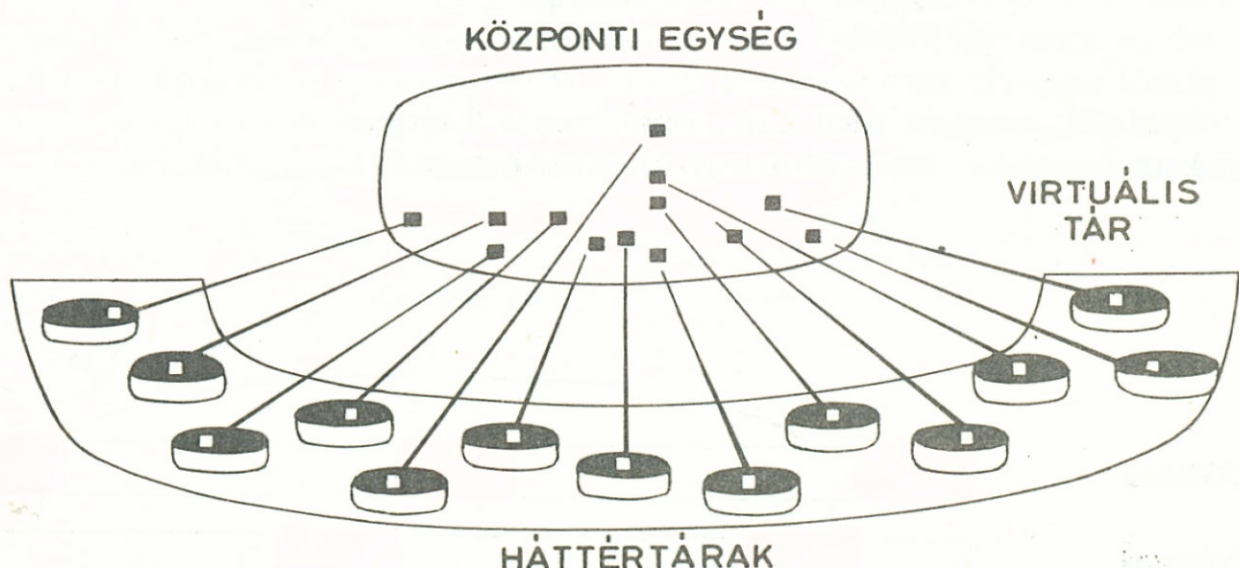


Ha sok felhasználó akarja saját programját egyszerre futtatni, akkor egyidejűleg sok programnak is rendelkezésre kell állnia. Ha tehát egyszerre száz felhasználó dolgozik a rendszerrel, egyszerre száz program kellene, hogy legyen az operatív tárban, amihez igen nagy operatív tárra lenne szükség. Azért, hogy ezt elkerüljék, kifejlesztették a **virtuális (látszólagos) tárolás technikáját**.

Itt a felhasználók programjait gyors háttértáron (leginkább mágneslemezen) helyezkedik el.

Az operatív tárba ezután mindig csak azokat a programokat vagy programrészeket hozzák be, amelyek a feldolgozáshoz éppen szükségesek. Ezeknek a programrészeknek a mérete szabványos, pl. 4096 byte, és **lapoknak** nevezik őket. Az operatív rendszer csak akkor továbbítja egy program lapjait a külső tárból az operatív tárba, ha a végrehajtásuk közvetlenül esedékes. Ezt a továbbítást **lapozásnak** (paging) nevezzük. A lapokat az operatív tárba oda írják be, ahol éppen hely van. Ezért valamennyi, ezen a lapon szereplő címét módosítani kell — a virtuális címeket automatikusan valós tárcímekké kell átszámítani.

Az alkalmazott központi egységek gyors műveletvégző sebességét az jellemzi, hogy megéri egy felhasználó program- és adatlapjait egy rövid, csupán 0,2 másodperces feldolgozási időre is egy mágneslemezről bevinni, és ezen rövid feldolgozási idő után azokat ismét visszaírni*.



* A leírt programcserén (swapping) alapuló tárkezelési eljárás helyett általában más eljárásokat alkalmaznak, amelyeknél egyidejűleg több felhasználó éppen szükséges lapjait tárolják az operatív tárban. (Lektor.)

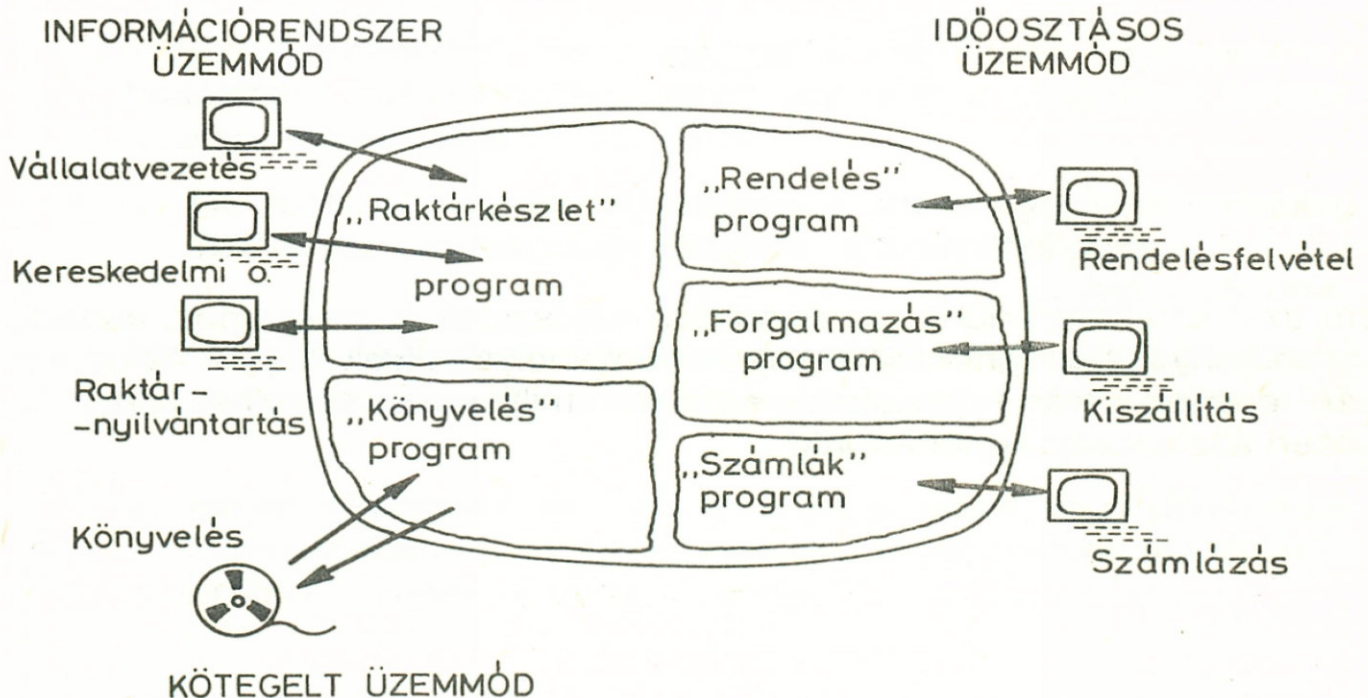
Mindegyik elmondott üzemmódhoz speciális operációs rendszer tartozik. Egy és ugyanazon számítógép pl. egy operációs rendszerrel kötegelt feloldozást tud végrehajtani, vagy egy másik, bonyolultabb operációs rendszerrel virtuális tárkezelésű időosztásos üzemben tud működni*.

Példa:

Egy kereskedelmi vállalat olyan operációs rendszerrel működteti számítógépét, amellyel kötegelt és valós idejű üzemmódot egyaránt meg lehet valósítani.

A vállalatvezetés, a kereskedelmi osztály és a raktárgazdálkodás a raktári készleteket információrendszer üzemmódban kérdezi le, ill. aktualizálja. A rendelés-nyilvántartási, kiszállítási és számlázási osztály egyidejűleg időosztásos üzemmódban dolgozik, saját külön programjaival és állományaival.

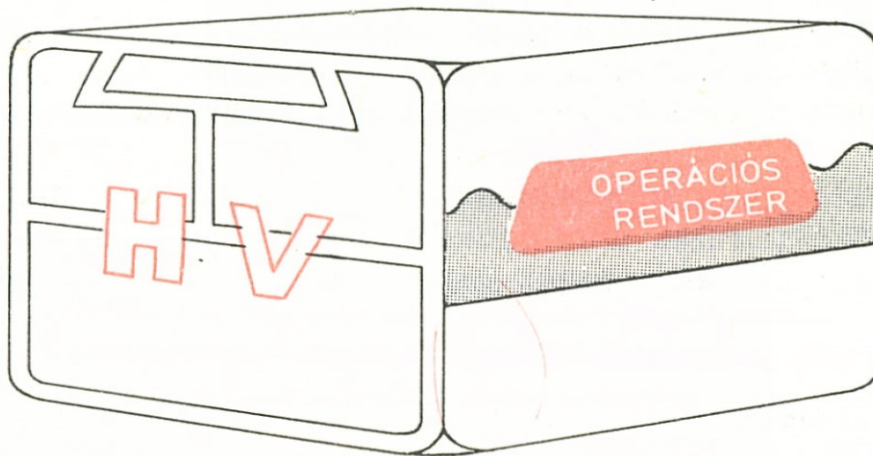
Végül a könyvelés napi könyveléseit mágnesszalag-kazettán rögzíti, és beküldi a számítóközpontba, ahol azokat a többi (párbeszédés formában futó) munkával párhuzamosan kötegelt üzemmódban dolgozzák fel.



* Az ún. többdimenziós operációs rendszerek egyidejűleg több üzemmódot (pl. kötegelt és valós idejű) is támogatnák (mint a soron következő példában is). (Lektor.)

Ezzel körülbelül ismertettük, hogy mit tud egy operációs rendszer. Az is bizonyára világossá vált az Olvasó számára, hogy egy számítógép, bármilyen jó is, csak annyit ér, mint amennyit a rendelkezésre álló operációs rendszer.

Felhasználói programot csak akkor lehet kifejleszteni és élesen futtatni, ha mind hardverünk, mind pedig operációs rendszerünk van.



Nem szabad elfelejtenünk: az operációs rendszerek is programok, és pedig nagyon bonyolult programok, amelyeket ugyanolyan elvek alapján és ugyanolyan lépések során fejlesztenek ki, mint amelyeneket ebben a könyvben teljesen általánosan ismertettünk!



1. A következő állítások közül melyek érvényesek a kötegelt üzemmódra?

Valamennyi ugyanazzal a programmal feldolgozandó adatot összegyűjtenek, és azokat egy menetben feldolgozzák.

Csak ha az egyik programnak vége van, akkor kerül sorra a következő program és adatköteg.

Valamennyi adatot rögtön a keletkezésekor feldolgoznak.

A spooling üzem a kötegelt üzemmód speciális formája, amelynél az adatokat és parancsokat tetszőleges időpontban fel lehet vinni egy háttértárra, azonban csak a központi egység szabadná válásakor lehet azokat feldolgozni.

2. A valós idejű üzemmód egyik formája a párbeszédés üzemmód. Nevezze meg a párbeszédés üzemmód két lehetséges fajtáját:

..... és

3. Milyen üzemmódról beszélünk, ha a helyfoglaló rendszerekhez hasonlóan, sok terminálról lehet ugyanazt a programot futtatni, ill. ugyanazokat az adatokat elérni?

4. a) Ha több felhasználó ugyanazon a számítógépen egyidejűleg különböző programokkal dolgozik, a számítógépnek kell működnie.

b) Hogyan nevezik a programok végrehajtásának rendezett időbeni váltakozását a központi egységben angol szakkifejezéssel?

c) Milyen tárkezelési technika teszi azt lehetővé, hogy időosztásos üzemmódban ne kelljen az operatív tárban minden felhasználói programot teljes egészében tárolni?



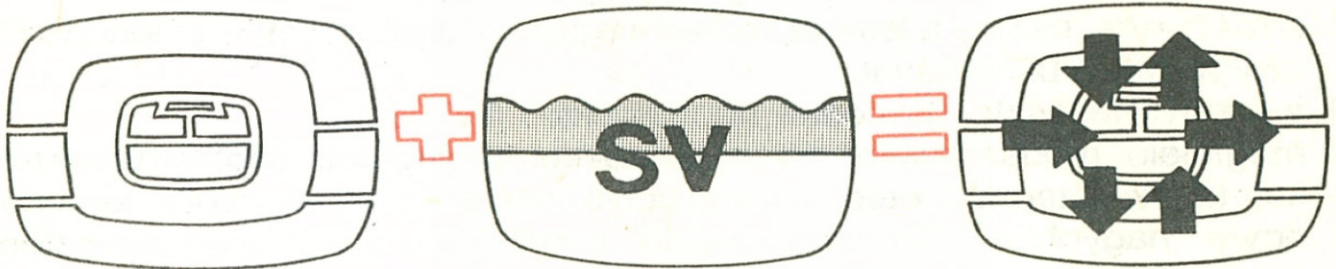
Válaszok

1. A kötegelt üzemmódra a következő állítások érvényesek:

- Valamennyi ugyanazzal a programmal feldolgozandó adatot összegyűjtenek, és azokat egy menetben feldolgozzák.
- Csak ha az egyik programnak vége van, akkor kerül sorra a következő program és adatköteg.
-
- A spooling üzem a kötegelt üzemmód speciális formája, amelynél az adatokat és parancsokat tetszőleges időpontban fel lehet vinni egy háttértárra, azonban csak a központi egység szabaddá válásakor lehet azokat feldolgozni.

2. A párbeszédés üzemmód két lehetséges fajtája az **információrendszer üzemmód** és az **időosztásos üzemmód**.
3. Az **információrendszer üzemmód** esetén terminálról lehet ugyan-
ezt a programot futtatni, ill. ugyanazokat az adatokat elérni.
4. a) Ha több felhasználó ugyanazon a számítógépen egyidejűleg különböző programokkal dolgozik, a számítógépnek **multiprogramozottan** kell működnie.
- b) A programok végrehajtásának rendezett időbeni váltakozását a központi egységben **időosztásnak** nevezzük.
- c) A **virtuális tárkezelési technika** teszi azt lehetővé, hogy időosztásos üzemmódban ne kelljen az operatív tárban minden felhasználói programot teljes egészében tárolni.

E kötet áttanulmányozása után világossá vált, hogy a gyakran idézett „számítógép-intelligencia” abban a szoftverben rejlik, amellyel a hardvert programozzák. Csak a szoftveren múlik, hogy a műszaki berendezések célszerűen működnek-e, és egy adott szakterület feladatait meg tudják-e oldani.



A szoftver előállítása sokrétű folyamat, amelynek szakmai, számítástechnikai és szervezési tényezői vannak. Ennek a folyamatnak rendezetten, koordináltan és bármikor ellenőrizhetően kell lefutnia. A résztvevők mindegyikének mindig tudnia kell, hogy a fejlesztés melyik részén dolgozik és hogy miként illeszkedik munkája az egészbe. A szoftver előállításakor — ezt a mérnöki jellegű eljárásmodot szoftvertechnikának nevezik. Beszélhetünk „szoftver-eszközökről” is, ha azokat a módszereket és segédeszközöket tekintjük, amelyek az ésszerű és áttekinthető szoftverfejlesztéshez nélkülözhetetlenek.

Ez az eljárásmod és az egyre tökéletesedő eszközök alkalmazása a következő célok elérésére irányul: A számítógépprogramok önmagában zárt funkcionális modulok sorából álljanak, amelyeknek csatlakozófelülete annyira univerzálisan legyen kiképezve, hogy különböző modulok tetszőlegesen kombinálhatók és cserélhetők legyenek. Eközben kívánatos, hogy ilyen modulok függetlenek legyenek attól a hardvertől, amelyen lefuttatjuk azokat úgy, hogy egy számítógép felhasználója a feladatát messzemenően egy modulkönyvtárban tárolt modulok automatikus összeállításával tudja megoldani.

A programnyelvek sokaságát redukálni kellene egyetlen, a természetes nyelvhez lehetőleg közel álló nyelvre, amelynek segítségével egy felhasználó terminálról beviheti parancsait. Ez azonban még messze van.

Amit eddig elértünk:

- a szoftverfejlesztés a feladat egzakt módon meghatározott és ellenőrizhető szakaszokra bontásával,
- a szoftverfejlesztés a modul- (építőköcka-) elv alapján,
- megfelelő módszerekkel és eszközökkel támogatott szoftverfejlesztés,
- kutatások a természetes nyelven folyó ember—gép kapcsolat területén,
- egyre nagyobb teljesítőképességű operációs rendszerek fejlesztése, amelyek a számítógép használatát még egyszerűbbé teszik.

A hardverterület fejlődése megelőzte a szoftvert. Szakadék keletkezett, amelyet általában szoftverkrízisnek neveznek.

A szoftverfejlesztés sürgető feladata nemcsak ennek a szakadéknak az áthidalása, hanem olyan átfogó koncepció kidolgozása is, amelyben a hardver jövője továbbfejlesztése is helyet kap.

A hardvertől független, tetszőlegesen kombinálható programmodulok talán ilyen megoldást jelenthetnek. Azonban a szervezésnek is mindig újra alkalmazkodnia kell az új adottságokhoz, és olyan megoldások után kell kutatnia, amellyel a holnap számítógépe — amelyet olyan komplex, magas fokon integrált rendszerek jellemeznek, amelyhez újszerű kommunikációtechnikákkal még a magánháztartások is hozzá tudnak férni — értelmesen és hasznosan alkalmazható.

A szoftver problémáit nem szabad elszigetelten vizsgálni, hanem mindig abban az összefüggésben, amelyben felmerültek — egy olyan rendszer egyik részeként, amelynek elemei:

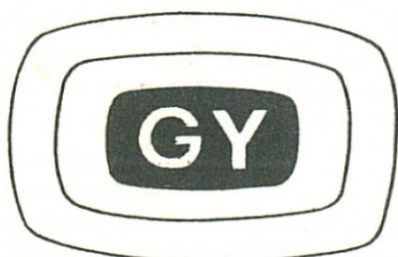
- a hardver,
- a szoftver és
- a szervezés (orgver).



E könyvsorozat megírása során ezen összefüggést tartottuk szem előtt. Ha a számítástechnika területén alapvető ismeretekre akarunk szert tenni, elengedhetetlen, hogy a szoftverrel határos területeken is tájékozódjunk.

Ehhez ajánljuk Önnek a sorozat többi kötetét:

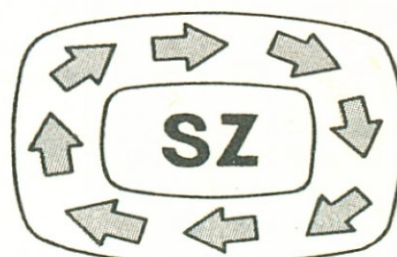
Kulcs a számítógéphez
Gyakorlat



Kulcs a számítógéphez
Hardver



Kulcs a számítógéphez
Szervezés



Tárgymutató

A, Á

ADA 122
 adat pontos leírása 131
 adatállomány pontos leírása 131
 adat-folyamatábra 83
 adatkezelő rendszer 167
 — rész 165, 167
 adatvédelmi megbízott 54
 ALGOL 121
 alkalmazói szoftver 28
 állománykezelő programok 168, 173
 állománylétesítő program 174
 assembly nyelv 120
 átadási jegyzőkönyv 49
 átviteli programok 168
 automatizált segédeszközök 61
 azonos idejű üzemmód 184

B

BASIC 122
 batch processing 182
 becslések 64
 berendezésjelképek 85

C, CS

COBOL 121
 cserélhető programok 15

D

dokumentálás 135, 149
 döntések 139
 döntéshozó testület 40
 döntési táblázatok 77
 döntési táblázat — előfordító 80
 dummy 73

E, É

élesfutás 177, 178
 eljárás 22
 eljárásteszt 136, 153
 ellenőrzési osztály 54
 építőkocka-elv 69

F

fadiagram 97
 feladat leírása 130
 fejlesztési folyamat szakaszokra tagolása 23
 — módszerek 24
 fejlesztői dokumentáció 154
 feldolgozási szabályok 131

felhasználói kézikönyv 154
 feltétel 77
 feltételes elágaztató utasítások 138
 felülről lefelé fejlesztés 95
 fordítási lista 147, 169
 fordító 175
 fordítóprogramok 165, 169, 175
 forrásprogram 173
 FORTRAN 121
 futási rész 165, 166

G

géporientált nyelvek 120
 grafikus módszerek 24

H

hálóterv 63
 hálótervezési eljárás 62
 hardver 14
 helyettesítő modulok 73
 helyzetfelmérés 43

I

ideális elképzelés 43
 időosztásos üzemmód 185, 186
 időszakos üzemmód 187
 implementálás 135
 információrendszer üzemmód 185
 inspekciós technika 111
 integráció szakasza 135
 integrációs dokumentáció 154
 integrálás 153
 íróasztalteszt 142

J

javaslati szakasz 42
 jelképek, adat-folyamatábra 83
 —, program-folyamatábra 90
 jó minőségű szoftver 33

K

kezelési utasítás 154
 kidolgozott programlogika ellenőrzése 135
 kódinspekció 111
 kódlapok 146
 kódolás 135, 145
 könyvtárkezelő programok 168
 korai iterációs teszt 73
 kötegetelt feldolgozás 182
 köteg-üzemmód 181, 182

központi egység 14
kritikus út 66

L

lapok 188
lapozás 188
LISP 122
logikai hibák 147

M

magas szintű nyelvek 121
módszerek 61
modulkönyvtár 73, 150, 175
modultechnika 69
modultechnika-elv 69
modulok 70
multiprogramozás 186
műveleti utasítás 137

O, Ő

oktatási terv 47
operációs rendszer 30, 161
operatív tár 18
OS 161
összekötő munkatárs 53
osztott idő üzemmód 187

P

párbeszédprogramozás 146
— terminál 184
— üzemmód 184
PASCAL 122
periféria 14
Petri-hálók 103
Petri-hálók alapjelképei 103
PL/1 121
próbaüzem 48, 136
— előkészítése 48
problémaorientált nyelvek 120
program elágaztatása 138
— specifikálása 129
— tesztelése 147
programáttekintés 130
programelem 95
program-folyamatábra 89
programlogika kidolgozása 137
programmodul teljes leírása 149
programmodulok 69, 175
programnyelvek 119
programok 16
programozás 47
programrendszer 153
programspecifikációk 46
programteszt 109
projekt bevezetési szakasza 50

projektben végzett munka 23
projektháló 64
projektjavaslat 42
projektmunkacsoport 53
projektvezetés 54

R

real-time üzem 184
rendező-, ill. egyesítőprogramok 168
rendszerközei szoftver 29, 162
rendszer-szoftver 27, 162
rendszer-teszt 136, 153
részletes programlogika kidolgozása 135
— számítástechnikai rendszerterv 46
rögzített program 15

S

segédprogramok 165, 168
software engineering 23
specifikációk 43
SPL 122
spooling-üzem 183
struktogram 92, 96
struktúrablokk 96
strukturált programozás 95

Sz

szakmai probléma 43
— rendszerterv 43, 44
számítógép utasításkészlete 17
számítógépes eljárás 22
számítógépprogram 14, 22
számítóközpont felelős személyzete 53
szár-teszt 142
szerkesztő 176
szerkesztőprogram 176
szervezési tevékenységek 40
szoftver 15, 16
szoftverfejlesztés 21
— szerszámai 115
szoftverkarbantartás 50
szoftverkarbantartási kézikönyv 154
szoftvertechnika 23
szolgáltatások leírása 45
szövegszerkesztő 174

T

tanácsadó bizottság 53
— testület 40
tárgymodulok 175
tartalékidő 65
tervezési módszerek 24
tervinspekció 111
tesztdokumentáció 154

tesztelési szakasz 135
— útmutatások 131
tesztfutás 178
tesztkörnyezet 148
teszt-segédprogramok 168
tevékenységek 77
time sharing üzemmód 187
töltőprogramok 168, 177
tool 115
„top-down” fejlesztés 95

U, Ő

utasítások 17
üzemi tanács 54
üzemmód 181

V

valós idejű üzemmód 181, 184
vázlatos számítástechnikai rendszerterv 45
végevizsgálat 140
végtelen ciklus 140
vezérlőprogram 165
virtuális tárolástechnika 188

W

walk-through módszer 110

Z

zárójelentés 49

I. megvalósítási szakasz 46
I. tervezési szakasz 43
II. megvalósítási szakasz 48
II. tervezési szakasz 44

Tartalom

Előszó	5
A kötet célja	7
Tanulási módszer	8
Mi a szoftver?	11
Hardver és szoftver	13
Szoftvertechnika	21
A szoftver fajtái	27
A szoftver minősége	33
Hogyan fejlesztik a szoftvert?	37
A szoftverfejlesztés szakaszai	39
A szakaszokra való felosztás alapelvei	39
Javaslati szakasz	42
Az I. tervezési szakasz	43
A II. tervezési szakasz	44
Az I. megvalósítási szakasz	46
A II. megvalósítási szakasz	48
Kifejlessze a szoftvert?	53
Mi segítheti a szoftvertervezést és -fejlesztést?	59
A projekttervezés módszerei	62
A hálótervezés	63
Tervezési módszerek	69
A moduláris programozás	69
Döntési táblázatok	77
Az adat-folyamatábra	83
A program-folyamatábra	89
Strukturált programozás	95
Petri-hálók	103
Ellenőrzési módszerek	109
A walk-through módszer	110
Az inspekciós technika	111
Automatizált segédeszközök	115
Programnyelvek	119
Géporientált programnyelvek	120
Problémaorientált programnyelvek	121

Mit csinál a programozó?	127
Programok specifikálása	129
A programfejlesztés lépései	135
A programfejlesztési lépések áttekintése	135
A program kidolgozása	137
A program logikájának ellenőrzése	142
A program kódolása	145
A program tesztelése	147
A program dokumentálása	149
A program integrálása	153
A programrendszer tesztelése	153
A programrendszer dokumentálása	154
Milyen szolgáltatásai vannak az operációs rendszernek?	159
Az operációs rendszer szolgáltatásainak áttekintése	161
Az operációs rendszer komponensei	165
A vezérlőprogram	165
— A futási rész	166
— Az adatkezelő rendszer	167
Segédprogramok	168
Fordítóprogramok	169
Az operációs rendszer szolgáltatásai programvégrehajtás közben	173
Állománykezelő	173
Fordító	175
Szerkesztő	176
Töltőprogram	177
Az operációs rendszer és a felhasználói program együttműködése	178
Üzemmodok	181
Kötegetelt üzemmód	182
Valós idejű üzemmód	184
— Információrendszer üzemmód	185
— Időosztásos üzemmód	186
Tárgymutató	187

JEGYZET

Kiadja a Műszaki Könyvkiadó
Felelős kiadó: Szűcs Péter igazgató
Felelős szerkesztő: Czere Károlyné

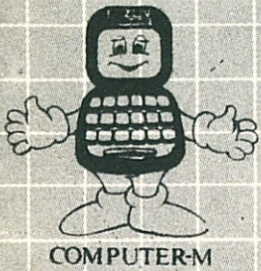
88.3970-66-13-1 Alföldi Nyomda, Debrecen
Felelős vezető: Benkő István vezérigazgató

A szedés a Műszaki Könyvkiadóban készült
Műszaki vezető: Kőrizs Károly
Műszaki szerkesztő: Molnár József
A borítót és a kötést tervezte: Kováts Tibor
A könyv ábráit rajzolta: dr. László Jenőné, Simonetz Józsefné
A könyv formátuma: B/5
Ábrák száma: 398
Papír minősége: 80 g ofszet
Betűcsalád és -méret: Helvetica, 9/10 p.
Azonossági szám: 80023
MŰ: 4091—h—8890
Készült az MSZ 5601 és 5602 szerint



SZÜV

Központi Hatástechnikai Hivatal
 Számítástechnikai és
 Ügyvitelszervező Vállalat
 1145 Budapest XIV. Sugló u. 9-15
 Telefon: 692-000 Telex: 226216



tanácsadás,
 oktatás,
 hardver-, szoftverbemutatók,
 programkészítés,
 helyi gépbérlet,
 szoftvertermékek,
 papíralapú és mágneses adathordozók,
 számítástechnikai segédeszközök,
 szakirodalom,
 hardvereszközök beszerzése,
 értékesítése,
 lízing



„Bizra a SZÜV-ne!”



COMPUTER—M Ügyfélszolgálati Irodák
 az ország minden pontján!



SZÜV

Központi Statisztikai Hivatal
Számítástechnikai és
Ügyvitelszervező Vállalat
1145 Budapest XII. Sugló u. 9-15
Telefon: 642-000 Tékax: 226216



Nagy és mikroszámítógépre
alap-, és alkalmazói szoftverek:

- fejlesztése
- adaptálása
- oktatása
- forgalmazása
- szoftverkövetés

„Bizsa a SZÜV-re!”



Fejlesztő leányvállalat
Szoftverházak
Országos fejlesztő apparátus