

**Easy file - tól
MASTER 64-ig**

Adatfeldolgozó programcsomagok

COMMODORE 64-re

**LSI Alkalmazástechnikai
Tanácsadó Szolgálat**

EASY file-tól a MASTER 64-ig
Adatfeldolgozó programcsomagok
COMMODORE 64-re

EASY file

SUPERBASE

COMPAL

MULTIPLAN

MASTER 64

Írták: Bakallár Sándor
Kruzslák Erzsébet
Nagy József
Várdainé Kollár Judit

Lektorálta: Dr. Úry László

Szerkesztette: a MIKROVILÁG GM

Tartalomjegyzék

Bevezetés	3
1. Adatfeldolgozás személyi számítógéppel	5
1.1 Az adatfeldolgozási feladat elkészítése	7
1.2 Az adatok csoportosítása és az adattárolók	11
1.3 A programcsomagok kiválasztása a feladathoz	14
2. A programcsomagok általános ismertetése	15
3. Az egyes programcsomagok ismertetése	18
3.1 Az EASY-FILE	19
3.1.1 Az EASY-FILE eszközigénye	20
3.1.2 Az EASY-FILE részletes ismertetése	21
3.1.3 Személyi nyilvántartás EASY-FILE segítségével	62
3.2 A SUPERBASE	77
3.2.1 A SUPERBASE eszközigénye	77
3.2.2 A SUPERBASE részletes ismertetése	79
3.2.3 Személyi nyilvántartás SUPERBASE segítségével	98
3.3 A MULTIPLAN	113
3.3.1 A MULTIPLAN eszközigénye	114
3.3.2 A MULTIPLAN részletes ismertetése	114
3.3.3 Főkönyvi könyvelés MULTIPLAN segítségével	136
3.4 A COMPAL	146
3.4.1 A COMPAL eszközigénye	146
3.4.2 A COMPAL részletes ismertetése	148
3.4.3 Személyi nyilvántartás COMPAL segítségével	168

Tartalomjegyzék

3.5	A MASTER-64	188
3.5.1	A MASTER-64 eszközigénye	190
3.5.2	Tartalomjegyzék a részletes ismertetéshez	192
3.5.3	A MASTER-64 részletes ismertetése	196
3.5.3	Személyi nyilvántartás a MASTER-64 segítségével	246
4.	Adatvédelem	251
4.1	Lemezvédelem	251
4.2	Adatbiztonság	252

A mikroszámítógépek, illetve a személyi számítógépek térhódítása a számítástechnikát közelebb hozta az emberekhez. Ma már előfordul, hogy nem mítikus elem a számítógép, hanem eszköz, amellyel nagyon sok feladat oldható meg.

A nagy- és kissozámítógépek ára olyan magas volt, hogy azt még a nagy vállalatok sem voltak képesek mindig megvásárolni. A személyi számítógépek ára a nagy- és kissozámítógépekhez viszonyítva annyival kevesebb, hogy azt még magán személyek is megvásárolhatják. Ha vállalat vásárolja, akkor akár fogyóeszközként lehet elszámolni és nyilvántartani, tehát majdnem mindegyik vállalat illetve intézmény hozzájuthat.

Az, hogy nagyon sok háztartásban jelen van a számítógép azt jelenti, hogy önképzés útján nagyon sok ember elsajátítja a gépek kezeléséhez szükséges ismereteket, és így ezek az emberek a munkahelyi feladataik megoldásához, mint eszközt, könnyebben használják a számítógépeket.

Amikor elfogadjuk a mikroszámítógépet, mint eszközt, nagyon kell figyelni arra, hogy a feladatokat hogyan oldjuk meg. Nem célszerű egyedi programokat írni, hiszen rendelkezésünkre áll nagyon sok általános programcsomag, amelyet az ügyviteli feladatok gépesítéséhez erre a célra szakosodott szoftver cégek dolgoztak ki.

Ha programcsomag segítségével oldjuk meg a feladatot, akkor időt és pénzt takaríthatunk meg. Ezt azzal magyarázhatjuk, hogy a programcsomag megvásárlása egyszeri költség. Ha már rendelkezünk vele, akkor nagyon sok feladat számítógépes megoldására használható, nem szükséges külső programozói kapacitást igénybe venni.

Igyekeztünk olyan programcsomagokat kiválasztani az ismertetés céljára, amelyek beszerezhetők hazai forgalmazásban.

A könyvben segítséget szeretnénk nyújtani a programcsomagok ismertetésén és példákon keresztül egy-két vállalati ügyvitelgépesítési feladat megoldásában.

Az első fejezetben az adatfeldolgozás néhány fontos szempontját emeljük ki, amelynek alapján a konkrét feladatok megoldásához szeretnénk segítséget nyújtani.

A második fejezetben általános ismertetőt adunk a programcsomagok használatáról.

A harmadik fejezetben az egyes programcsomagokat részletesen és - a könnyebb érthetőség kedvéért - egy példa megoldásán keresztül ismertetjük.

A negyedik fejezetben foglalkozunk a lemez- és adatvédelemmel, amelynek az adatfeldolgozásban nagyon fontos szerepe van.

1. Adatfeldolgozás személyi számítógéppel.

A mikrogépek és a hozzá kidolgozott adatkezelő programcsomagok lehetőséget adnak azoknak a felhasználóknak is, akik nem rendelkeznek komoly számítógépes ismeretekkel ahhoz, hogy helyi-, kisméretű feladataikat önállóan tudják megoldani.

A programcsomagok a hozzátartozó felhasználói dokumentációval, a feladat megfelelő előkészítése után, hatékonyan használhatók a manuális ügyvitel helyettesítésére.

A könyvben öt adatfeldolgozó programcsomagot ismertetünk módszertani útmutatóval. Ez a könyv nem a programcsomagok felhasználói leírását tartalmazza, hanem a programcsomagok általános ismertetését példákon keresztül, és segítséget ad a felhasználónak saját feladatainak megoldásához.

A bemutatott programcsomagokat négy kategóriába csoportosíthatjuk

1. Rekord illetve file kezelő programcsomagok.
(pl. az EASYFILE vagy a SUPERBASE)
2. Adattábla kezelő programcsomagok.
(pl. MULTIPLAN)
3. Hierarchikus alkalmazás generátor.
(pl. COMPAL)
4. Az ügyviteli feladatok programozását segítő programcsomag.
(pl. MASTER-64)

Mindegyik kategóriájú programcsomag általános jellegű, a felhasználó saját feladataihoz igazíthatja.

A személyi számítógépek elterjedése nagy lehetőséget teremtett az ügyviteli adatfeldolgozás területén. A könnyen beszerezhető általános jellegű adatfeldolgozó programcsomagok könnyen adaptálhatóak saját feladataink megoldására, nem szükséges külső szervezési és programozási segítség.

Ez a megállapítás természetesen csak akkor igaz, ha a meglévő adatfeldolgozást akarjuk gépesíteni, és nem új szervezésben akarjuk kivitelezni az ügyviteli feladatainkat. Vagy ha nem egy kis terület feladatait szeretnénk számítógépesíteni, inkább egy átfogó teljes rendszerre vonatkozó folyamatot akarunk számítógépes rendszerre szervezni.

A számítógépes adatfeldolgozásnál nagyon fontos az előkészítési folyamat, amelynek a következő lépésekből kell állnia:

1. Feladat meghatározása.
2. Feladat részletes leírása.
3. Adatok körének meghatározása.
4. Kapcsolódó adatáramlások feltérképezése.
5. Részletes adattleírás.
6. Programcsomag kiválasztása.

Az 1. és 2. pontokról az 1.1 fejezetben, a 3., 4. és 5. pontokról az 1.2 fejezetben, és végül a 6. pontról az 1.3 fejezetben fogunk írni.

1.1 Az adatfeldolgozási feladat előkészítése.

Mielőtt bármilyen számítógépes programcsomaggal elkezdünk dolgozni, a feladatot meg kell határozni. Erre azért van szükség, mert ha nem fontoltuk meg eléggé a feladatkört, akkor nagy valószínűséggel nem jól választjuk ki a programcsomagot, vagy ha a programcsomagot jól is választottuk ki, az adatok körében módosulás lesz, ami ezt eredményezi, hogy a kitöltött adatállományainkat újra kell rögzíteni.

Miután ezt végig gondoltuk és leírtuk (olyan részleteséggel, amit a munkafolyamataink megkívánnak), meg kell határozni azon rész illetve teljes feladatot, amelyeket a mikroszámítógéppel akarunk megoldani.

A feladat definiálása után határozzuk meg azt az adatkört, amely szükséges a feladat számítógépesítéséhez. Ezt a problémát legkönnyebben úgy tudjuk megoldani, ha megvizsgáljuk azokat a táblázatokat ill. eredményeket, amelyeket elő akarunk állítani. Ha megvan minden adat, amely szükséges a táblázatainkhoz illetve számításainkban, akkor meg kell határozni azokat a bizonylatokat, amelyekből ki tudjuk venni a szükséges adatokat. Ily módon megadtuk a kiinduló adatokat is, tehát definiáltuk a teljes adatkört.

Ezután következik azon algoritmusok meghatározása, amelyek segítségével a kiinduló adatokból megkapjuk az eredményeket.

A fent leírtakat példával is illusztráljuk.

Személyi nyilvántartás mikroszámítógéppel.

A személyi nyilvántartás számítógépesítésének célja, hogy a vállalatoknál felhalmozódó adatokból könnyen, ill. a manuális munkánál könnyebben nyerjük a szükséges adatokat a kimutatásainkhoz vagy a statisztikai beszámolóinkhoz.

Feladatmeghatározás:

1. A személyi nyilvántartó adatlap vagy karton alapján számítógépes adathordozón tárolni az adatokat.
2. Leválogató, lekérdező rendszer készítése, amelyek segítségével előállíthatjuk a szükséges táblázatokat, statisztikákat és vezetői információkat.

A személyi nyilvántartó rendszernek a nyilvántartásban nemcsak azokat az adatokat kell tárolnia, amelyek az aktuális kimutatásokhoz szükségesek, hanem azokat is, amelyeket bizonyos előírások szerint hosszabb időn keresztül tárolni kell.

Ebből következik, hogy minden nyilvántartásnál, minden ügyvitelgépesítési feladatnál tisztázni kell a jogszabályokban meghatározott előírásokat. Az általános jogszabályok után a vállalati ügyviteli rendet meghatározó szabályokat is alaposan meg kell ismerni. A személyi nyilvántartásnál jogszabályok írják elő, hogy mely adatokat nem szabad megsemmisíteni, hanem meghatározott időn keresztül őrizni kell, esetleg akkor is, ha a dolgozó már kilépett.

A mikroszámítógépes nyilvántartásnál szem előtt kell tartani, hogy a háttértárolók kis kapacitásúak, ezért a feladat meghatározását úgy kell elvégezni, végiggondolni, hogy adataink elférjenek az adattárolón, és a programjaink

is beférjenek a gép memóriájába.

Például Commodore 64 mikrogépen nem oldhatjuk meg egy ötezres személyi állományú nagyvállalat személyi nyilvántartását egy feladatként. Ha Commodore 64 géppel rendelkezünk, akkor kisebb szervezeti egységekre kell felbontani a személyi nyilvántartást. Nem szabad egy állományban tárolni a nyilvántartáshoz szükséges összes adatot, de a különböző állományok egymáshoz kapcsolását biztosítani kell.

Feladataink számítógépesítésénél szem előtt kell tartani, hogy megoldott legyen a vállalati információs rendszerbe illeszkedés.

Például a személyi nyilvántartás adatait nemcsak a személyzeti osztályok használják, hanem a vállalat területén még igen sok egység (munkaügy, munkaerőgazdálkodás, bérelszámolás stb.). Tehát amikor megvalósítjuk a személyi nyilvántartás számítógépesítését, akkor biztosítani kell a már meglévő rendszerekkel, illetve egy későbbi rendszer kidolgozásához a kapcsolatot. Ez alatt elsősorban az adatok illeszthetőségét (kompatibilitását) értjük a rendszerek között, mind tartalomban, mind formában.

A feladat részletes leírásának tartalmaznia kell a feladat jelenlegi megoldásának az ismertetését, azon bizonylatok felsorolását, amelyekről az adatokat biztosítani tudjuk, valamint az eredményeket és táblázatokat, amelyeket elő kell állítani a feldolgozás során. Továbbá a leírásban szerepelnie kell az esetleges szervezési módosításoknak, a gépi megvalósítás módjának és a feltételezett konfigurációnak is.

A személyi nyilvántartás akár számítógéppel, akár anélkül, minden vállalatnál, intézetnél stb. létezik. A nyilvántartás nem minden

vállalatnál teljesen azonos, de alapjaiban megegyezik. Minden dolgozóról köteles nyilvántartást vezetni a személyzeti osztály. A nyilvántartás módja függ attól, hogy a dolgozó fizikai állományban, vagy nem fizikai állományban van. Emiatt több fajta bizonylatot is használnak egy vállalatnál belül is. A dolgozók egy körére az országosan elfogadott 'Személyi adatlap'-ot használják, a másik körére a vállalatnál kidolgozott kartonokat.

A dolgozó belépésével ezeket az adatlapokat, kartonokat kitöltik és munkahely szerinti rendezettségben (amit általában költséghely számmal határoznak meg), azonbelül ABC sorrendben letárolják a megfelelő szekrényekben. Ha a dolgozó valamilyen adata megváltozik, azt módosítani kell a kartonon, de ez sajnos nem megy, az egyes szervezeti egységek nem jelentik minden alkalommal a módosítást. A dolgozó pedig nem járja végig az összes helyet, ahol a módosításokat át kéne vezetni. Ebből következik, hogy nem naprakész a személyi nyilvántartás.

A személyi nyilvántartás számítógépes feladat meghatározása:

A dolgozó belépésekor a nyilvántartási karton kitöltése helyett számítógépen hajlékony mágneslemezre rögzítve viszik fel az adatait. A mágneslemezek tárolása, ahogyan az a kézi feldolgozásnál is történt, költséghelyenként történik. Ezzel természetesen nem oldottuk meg az adatok naprakész voltát. Ezért elő kell írni, hogy az egyes szervezeti egységek minden módosításról értesíteni kötelesek a személyzeti osztályt.

Amennyiben a belépés a vállalatnál a személyzeti osztályon történik, akkor a már lerögzített adatokból másolatokat küldenek a többi szervezeti egységhez (mágneses adathordozón), ahol a dolgozókról nyilvántartást kell vezetni. (pl. munkaügy, oktatás, bérelszámolás, és az a szervezeti egység, ahol a dolgozó alkalmazásban lesz stb.).

Ezzel elérjük azt, hogy ott rögzítik az adatokat, ahol keletkeznek, és megtakarítunk sok időt azzal, hogy elkerüljük a többszörös adatrögzítést. Ilyen módon az adatok valósága is biztosított, hiszen az elírásokból, a hibás olvasásból nem keletkezhet hiba.

Az egyes szervezeti egységek ugyancsak mágneses adathordozón (floppy-n) kötelesek a módosításokat megküldeni. Ezzel a módszerrel biztosíthatjuk, hogy a vállalatnál naprakész adatok állnak mindenkor rendelkezésre. Ezáltal elérhetjük, hogy a felsőbb szervekhez küldendő statisztikáink megfelelnek a valóságnak.

A leválogató rendszert az adott feladatokhoz kell kialakítani. Vannak olyan kötelezettségei a vállalatnak, hogy a felsőbb szervekhez (például Ipari Minisztérium, KSH) meghatározott szempontok alapján kell a vállalat dolgozóiról összesített kimutatást küldeni. Ezeket a feladatokat a programcsomagok segítségével beprogramozhatjuk, és amikor szükséges lefuttatjuk.

Vannak olyan leválogatási feladatok, amelyek nem állandóan ismétlődnek, hanem csak egyszer - egyszer szükségesek. Ezeket a feladatokat nem célszerű beprogramozni, mert a programcsomagok lehetővé teszik a programon kívüli leválogatást is.

1.2 Az adatok csoportosítása, figyelembevéve az adattárolókat.

A feladat meghatározása bizonyos mértékig maga után vonja az adatok halmazát.

Az első lépés, hogy vegyük elő a bizonylatokat, amelyeket az adott feladathoz használunk. Ezután írjuk le az adatok neveit, típusát, hosszát, amelyeket fel fogunk használni a feladatunkhoz. Itt még nem kell törődni azzal, hogy milyen sorrendben fogjuk felhasználni ezeket az adatokat. Miután leírtuk az összes adatot, vizsgáljuk meg a táblázatainkat, hogy meg tudjuk állapítani, hogy minden adat a rendelkezésünkre áll-e az eredményeink, ill. táblázataink létrehozásához. Természetesen a számított adatoknál nem az eredményeknek kell ebben a stádiumban rendelkezésre állni, hanem azoknak az adatoknak, amelyekből kiszámíthatjuk azokat.

Ha nem áll rendelkezésre minden adat, akkor meg kell vizsgálni még egyszer a bizonylatokat. Ha azokon nincs rajta a szükséges adat, akkor azt kell megnézni, hogy honnan származtathatjuk, és mint kiinduló adatot fel kell venni a listára.

Nagyon lényeges szempont, hogy indulásnál meghatározzunk minden szükséges adatot, mert a későbbiek folyamán, amikor már gépen rögzítettük az adatokat, többszörös munkával tudjuk csak kijavítani az itt elkövetett hibákat.

Ahhoz, hogy megítéljük, hogy minden adatot figyelembe vettünk-e, pontosan meg kell határozni a többi osztállyal, ill. szervezeti egységgel meglévő adatkapcsolatot is. A bizonylatokról vett kiinduló adatokat és a csatlakozó szervezeteknél figyelembe vett közös információs kapcsolatokból származó adatokat össze kell vetni az eredményként használandó adatokkal, és így feltételezhetően hiánytalanul határoztuk meg az input adatokat.

A Személyzeti nyilvántartás példájánál maradva, input bizonylatul szolgál a 'Személyi adatlap', amelyet minden felsőfokú végzettségű emberről vezetni kell. De minden dolgozóról nyilvántartást kell vezetni, ezért a vállalatok saját maguk is kialakítottak valamilyen bizonylatfajtát, amely adatait tekintve nagyon sokban hasonlít az országosan egységes 'személyi adatlap'-hoz.

Ezekből a bizonylatokból már egyértelműen meghatározható az az adathalmaz, amelyre a vállalat, vagy intézmény személyzeti osztályának szüksége van. Általában a mindennapos feldolgozásokhoz ezeknek az adatoknak csak egy részére van szükség, a másik része azért szükséges, mert a dolgozókról éveken át régi adatokat is nyilván kell tartani.

Ilyen meggondolások alapján el is lehet dönteni, hogy a nagy adathalmazt több csoportra kell osztani:

1. csoport

A mindennapi használathoz szükséges adatok.

2. csoport

Azok az adatok, amelyekből állandó változásuk miatt a régebbi adatokat is tárolni kell.

3. csoport

Azok az adatok, amelyeket bizonyos előírások miatt tárolni kell, de nem használják a feldolgozások során.

Ebből a három csoportból, és a Commodore 64-es számítógépekhez tartozó mágneslemezegység kis kapacitását figyelembe véve, három adatállományt lenne célszerű létrehozni, amelyek elférnek egy-egy lemezen:

-az első adatállományt a mindennapi feldolgozáshoz, a statisztikák készítéséhez.

-a második adatállományt azért, ha szükség van a dolgozók régebbi adatainak ismeretére, elő tudjuk venni.

-a harmadik adatállományt pedig a kötelező nem használatos adatok nyilvántartására.

Amennyiben az egész adathalmaz elfér egy lemezen, akkor is több adatállományban jó tárolni, mert a feldolgozást meggyorsítjuk ezzel.

1.3 A programcsomagok kiválasztása.

A feladat meghatározásánál definiáltuk az adattömeget, az adatcsoportokat, amelyek teljes egészében meghatározhatják a meglévő programcsomagok kiválasztását. Természetesen nem csak a nyilvántartás, hanem a leválogató rendszer is befolyásolja a kiválasztást. Ezenkívül még nem kis szerepet játszik a kiválasztásban az általunk meghatározott algoritmus is.

Ha csak egy adatállományos rendszert és csak egyszerű táblázatokat akarunk készíteni. Akkor olyan helyen, ahol nincs számítógépes szakember, ott az EASY-FILE programcsomag használata célszerű.

Ha a feladat meghatározásánál kiderült, hogy két adatállomány szükséges egyszerre és a táblázataink sem a legegyszerűbbek, akkor célszerű a SUPERBASE programcsomagot használni.

Ha számviteli, vagy főkönyvi feladatot akarunk megoldani a számítógépen, akkor a MULTIPLAN programcsomagot érdemes használni.

Ha hierarchikus struktúrájú adatállományokat akarunk létrehozni, akkor a COMPAL alkalmazás generátor programcsomagot célszerű kiválasztani. A COMPAL, mint interaktív alkalmazás tervező lehetőséget nyújt termék családfa, darabjegyzék nyilvántartására.

Mindegyik feladat megoldására alkalmas a MASTER-64 programcsomag. Ezt a programcsomagot abban az esetben célszerű használni, ha rendelkezünk szabad programozói (adatfeldolgozó) kapacitással.

2. A programcsomagok általános ismertetése

Az általunk ismertetett első négy programcsomag használata azért célszerű, mert ügyviteli adatfeldolgozásra dolgozták ki. A ötödik programcsomag használata pedig elősegíti az adatfeldolgozó programok készítését programozók számára. A képernyő-, file- és a táblázat generátor olyan segédeszközt ad a programozó kezébe, amely a paraméterezhetőségével könnyíti meg a feladat megoldását.

Ha minden feladathoz saját, ill. egyéni programot dolgozunk ki, természetesen az adott feladathoz pontos megoldást kapunk, feltételezve, hogy elég sok programozói kapacitás áll rendelkezésünkre.

Ha ügyviteli programcsomagot használunk a feladataink megoldására, akkor programozói kapacitást takarítunk meg. Saját munkatársainkkal oldhatjuk meg a feladatot, akik teljes egészében ismerik azt a területet, ahol számítógépes ügyvitelgépesítést szeretnénk végrehajtani. Ebben az esetben nem szükséges számítógépes programozói kapacitást igénybe venni, hanem elég egy tanfolyam elvégzése, amely ismerteti a Commodore 64 kezelését és a programcsomag használatát.

Ha programozói környezetben akarunk ügyvitelgépesítési feladatot megoldani, akkor célszerű a MASTER-64 használata, mert segíti az interaktív jellegű feldolgozások programozását. Ha a felhasználó megfogalmazta a feladatot, könnyen kialakítható olyan program, amely a megoldást adja.

Az általunk ismertetett programcsomagoknál nem meghatározó a hardver környezet, mivel mindegyik több lemezegységet és a Commodore 64 géphez használatos nyomtatókat is tudja kezelni. Az esetleges specialításokat az egyes programcsomagoknál külön ismertetjük. Abban mindegyik programcsomag megegyezik, hogy csak lemezegységet tud kezelni, mágnesszalagegységet nem. Maguk a programcsomagok is hajlékony lemezen találhatóak.

A programcsomagok használatához rendelkezésünkre kell hogy álljon egy, vagy több (magyar nyelvű) felhasználói kézikönyv. Természetesen egy programcsomaghoz elég egy felhasználói kézikönyv, de sajnos ma Magyarországon nincs egységes felhasználó-orientált tanácsadás és könyvkiadás az ügyviteli programcsomagokról. Ezért természetesen a kézikönyvek sem egységesek, előfordul, hogy egy és ugyanazt a könyvet többen lefordítják. Amennyiben több könyv áll rendelkezésünkre ez nem hátrány, mert többet megtudhatunk a programcsomagról, mivel a fordítások minősége különböző.

Amit a felhasználói kézikönyvekről írtunk az még inkább igaz a felhasználói programcsomagokra. A programcsomagok ilyen-olyan úton kerülnek az egyes felhasználókhoz, akik esetleg - minthogy nem számítógépes szervező szakemberek - ezért nem tudják melyik programcsomag illik legjobban a saját feladatuk megoldásához.

Bármilyen vállalatnál, intézetnél, szervezeti egységnél célszerű mindenképpen ragaszkodni az egységes programcsomag használatához.

Ezt a következőkkel lehet indokolni:

Egy intézményen belül nagyon sok területen szükségesek azonos adatok. Amennyiben rendelkezésükre áll személyi számítógép és nem azonos programcsomagot használnak, akkor kénytelenek a régi ügyviteli megoldáshoz folyamodni, és bizonylatokat kiállítani, hogy az adatok eljussanak egy másik szervezeti egységhez. Ez a szervezeti egység szintén rendelkezik személyi számítógéppel, csak másik programcsomag áll rendelkezésére. Ilyen esetekben az adatokat ismételtelen rögzíteni kell, hogy kényelmesen feldolgozhatók.

Ha egységes programcsomaggal rendelkezünk, akkor nem szükséges még egyszer bizonylatot kiállítani, rögzíteni az adatokat, hanem mágnesesadathordozón a másik szervezeti egységnek átküldhetjük a szükséges adatokat, ahol könnyen és kevesebb munkával végezhetjük el a feldolgozást.

Természetesen nem elég csak a programcsomagok azonossága. Nagy figyelmet

kell szentelni az adatok azonosságára is. Az adatok azonosságához hozzátartozik az adatok kódolása is. A kódolás tervezéséről, kivitelezéséről itt nem akarunk említést tenni, de mint alapvető szempontra felhívjuk a figyelmet az egységes kódrendszer fontosságára. Például ha egy vállalatban belül a különböző szervezeti egységek különböző módon jelölik az országokat. Ez igen komoly fennakadást okozhat a számítógépes adatfeldolgozásban, a ki-mutatásainkban hamis eredményt kapunk. Az adatok azonossága a következőkből áll:

- Az adatoknak minden szervezeti egységnél azonos típusúaknak és struktúrájúaknak kell lennie
- Az adatoknak mindenütt egyforma hosszúságot kell meghatározni
- Az adatoknak azonos sorrendben kell következniük egymás után
- Az adatoknak azonos tartalmúaknak kell lenniük azaz azonosan kell kódolni

A programcsomagok lehetőségeit nem szükséges minden szervezeti egységnél azonos módon felhasználni. Ha azonos módon használják fel a lehetőségeket - tehát, ha például szükséges egy fajta táblázat készítése a munkaügyi adatokról több szervezeti egységnél - akkor időt takarítunk meg a feldolgozásnál.

Ezek a lehetőségek csak könnyebbé teszik a feldolgozást, de nem elengedhetetlen feltételei az egységes feldolgozásnak, mint az azonos programcsomagok, és az azonos adatszerkezetek.

3. Az egyes programcsomagok ismertetése

Ebben a fejezetben ismertetjük a bevezetőben felsorolt programcsomagokat, a használatát ugyanazon feladat megoldásával illusztrálva. Az ismertetés sorrendje megegyezik azzal a felsorolással, amelyet az első fejezetben használtunk.

Ez a felsorolás természetesen nem véletlen, azért választottunk így, mert a programcsomagok kezelhetősége is ilyen sorrendben következik. (Ez természetesen nem a programcsomagok értékelését akarja jelenteni.)

A felhasználó szempontjából az a programcsomag tekinthető könnyen használhatónak, amelyet számítógépes szakemberek segítségével is lehet alkalmazni.

Még egyszer hangsúlyozni kívánjuk, hogy nem felhasználói kézikönyvként kell olvasni az egyes fejezeteket, hanem segítségnyújtásként az egyes programcsomagok kiválasztásához, a kijelölt feladatok megoldásához.

A felhasználók alatt ebben az esetben nemcsak a számítógép szempontjából laikus embereket értjük, hanem alkalmasint a programozni tudókat is, akik nem értenek az adatfeldolgozáshoz. Hiszen az adatfeldolgozás, a számítógépes rendszerszervezés egy külön szakma, amit nem minden esetben ismernek a programozó szakemberek.

Könyvünkben igyekszünk olyan módon leírni a programcsomagokat, hogy azt mind a számítógépes szakemberek, mind a nem számítógépes szakemberek fel tudják használni a megfelelő programcsomag kiválasztásához és használathoz.

3.1 Az EASY-FILE.

Az EASY-FILE Commodore 64 személyi számítógépre kidolgozott rekord szervezésű adatállománykezelő programcsomag. A programcsomag hajlékony mágneslemezes adathordozón található, és összefűzött programok sorozatából áll, amelyből az egyes programok speciális funkciókat látnak el. Az EASY-FILE lehetővé teszi, hogy ne minden esetben kelljen begépelni a rendezéshez vagy a táblázat készítéshez szükséges paramétereket, mert az adatlemezünkre elmenthetjük ezeket.

A programcsomag használatát, megtanulását nemcsak azért javasoljuk, mert könnyen kezelhető, és menü rendszere vezeti a felhasználót munkája megoldásában. Még egy nagyon fontos szerepet tulajdonítunk az EASY-FILE-nak. Mindenki számára sokkal használhatóbb lesz később bármilyen más adatfeldogozói programcsomag, ha elsajátították az EASY-FILE programcsomagon keresztül a file-ok, rekordok fogalmát. Megtanulták a file-kezelést, és a file-okból különböző szempontok szerinti leválogatásban a táblázatok (report) készítését.

Ha két meghajtós lemezegység áll rendelkezésünkre, akkor gyorsabb az adatfeldolgozás, mert nem kell a lemezeket állandóan cserélni. A 0-ás meghajtóba helyezzük a programlemezt, az 1-es meghajtóba az adatlemezt. Ha egy lemezegységünk van, természetesen akkor is meg tudjuk oldani a feladatainkat, csak valamivel hosszabb időt vesz igénybe a feldolgozás, mert a programok beolvasásakor a lemezeket cserélni kell. A programok akkor is tovább futnak, ha rossz lemezt teszünk be, mert be van építve a lemezhiba kezelése. Ilyen esetben a képernyőn megjelenik a felirat, hogy nem találja az adott file-t. Ha kicseréljük a lemezt a megfelelőre, akkor folytathatjuk a munkát.

3.1.1 Az EASY-FILE eszközigénye.

1. Commodore 64 személyi számítógép
2. Commodore 1541 lemez meghajtóegység és csatlakozó
3. Sornyomtatóegység a hozzátartozó interface-szel.
(CMB 1515, 1526, 801, 802)
4. Televízió vagy monitor, a megfelelő csatlakozóval
5. Rendszer lemez

```

18  "easyfile2"      prg
29  "easy.ovr02"    prg
63  "easy.ovr12"    prg
91  "easy.ovr22"    prg
41  "easy.ovr32"    prg
84  "easy.ovr42"    prg
87  "easy.ovr52"    prg
50  "easy.ovr62"    prg
1   "easy.ovr0"     prg
1   "easy.ovr1"     prg
1   "easy.ovr2"     prg
1   "easy.ovr3"     prg
1   "easy.ovr4"     prg
1   "easy.ovr5"     prg
1   "easy.ovr6"     prg
16  "easy.cd1"      seq
9   "easy.cd2"      seq
169 blocks free.

```

6. Adatlemezek.

Az IEEE interface-szel csatlakoztatható Commodore dual lemezegység (4022, 4027, 8023).

Ha két meghajtós egységet használunk, a programlemezt mindig a 0-ás

egységbe kell helyezni.

Az RS232C interface-en keresztül a nem Commodore nyomtatók is csatlakoztathatók.

3.1.2 Az EASY-FILE részletes leírása.

Az EASY-FILE olyan programcsomag, amely rekord szervezésű adatállományokat hoz létre, és menü szervezésével segíti a felhasználót. A rekordok fix hosszúságúak, és állandó mezőkből állnak. A mezőknek nincs külön nevük, a felvétel sorrendjében számozza a rendszer. Ha számítási műveleteknél akarunk hivatkozni a mezőkre, úgy f'n'-nel hivatkozhatunk rájuk, ahol 'n'-a mező sorszáma.

Az EASY-FILE programcsomag menü (választék-lista) vezérelt programcsomag. A menükben található meg az egyes funkciókat, amelyek a feladatunk megoldására szolgálnak. Az első menü a főmenü, amelyből kiválaszthatjuk a főfunkciókat, majd ha elvégeztük az adott menürészben a feladatot, visszatérhetünk a főmenübe. Minden főmenü funkcióhoz saját menü tartozik, amelyben kiválaszthatjuk a nekünk szükséges lehetőségeket.

Az adatállomány struktúráját képernyő tervező segítségével adhatjuk meg. A képernyő segítséget nyújt az adatbevitelnél, hogy pontosan tudjuk milyen mezőket töltünk ki, hiszen a képernyőn mezőnevet is adhatunk. Ez a mezőnév csak az adatrögzítőnek nyújt segítséget, mert ahogy már írtuk, a hivatkozásoknál sorszámokat kell majd megadni a mezőnevek helyett.

Az EASY-FILE programcsomag nemcsak a menü vezérlésével segíti a felhasználót, hanem különböző üzenetekkel is, hogy mikor kell a programlemez és mikor kell az adatlemez használni.

Az EASY-FILE főmenüjéből (választékából) teljesen egyértelműen kiderül, hogy milyen funkciókat képes ellátni a programcsomag:

1. Adatlétrehozás (Create Data File)
2. Adatmanipuláció (Data Manipulation)
3. Táblázat készítés (Report Generation)
4. Rendezés (Sort Data File)
5. Segédprogramok (Support Utilities)
6. Karbantartás (File Update)
7. Segítségnyújtás (Help)
8. Kilépés (Exit)

Fontosnak találjuk a főmenü egyes pontjainak magyarázatát, először azért, mert a főmenü menüválasztéka nem teljesen azonos azzal a terminológiával, amelyet megszoktak a számítógépes szakemberek, másodsorban azért, mert a nem számítógépes szakembereknek meg kell érteni az egyes szavak, kifejezések értelmét, hogy ebben a programcsomagban miképpen kell értelmezni ezeket.

1. Adatlétrehozás (Create Data File)

Ebben a funkcióban kell megtervezni a képernyőt, ha még nem tettük meg. Ha már rendelkezünk vele, itt lehet módosítani, kijavítani. Ezzel a funkcióval kell meghatározni, ha számításokkal akarunk létrehozni mezőket. Ha megterveztük a képernyőt, az egyben a rekord kialakítását is jelenti. Miután befejeztük a tervezést a programrész létrehoz egy üres adatállományt, s lefoglalja neki a helyet a lemezen. Az adatállomány nagyságát a rekordok számával adjuk meg.

2. Adatmanipuláció (Data Manipulation)

Ezzel a funkcióval tudjuk feltölteni az adatállományunkat adatokkal, itt

lehet elvégezni az adatok módosítását, törlését. Különböző szempontok alapján kikereshetünk rekordokat az adatállományunkból. Ha túlleptük az adatállomány nagyságát, akkor itt bővíthetjük a megfelelő nagyságúra. Ez a funkció takarja azt a fogalmat, amelyet általában karbantartásnak (adatfelvitel, adatmódosítás, rekordtörlés) szoktunk hívni.

3. Táblázat készítése (Report Generation)

Ezzel a funkcióval tudunk táblázatokat, kimutatásokat készíteni az adatállományunkból. Az adatállományt több módon rendezhetjük. Ha nincs szükségünk az egész adatállomány adattartalmára, akkor különböző szempontok alapján leválogathatjuk a számunkra szükséges adatszoportokat.

4. Rendezés (Sort Data File)

Az adatállományt nemcsak a táblázat készítésekor, de más esetben is rendezhetjük, több kulcs alapján.

5. Segédprogramok (Support Utilities)

Ebben találjuk a lemezműveletekhez szükséges funkciókat. Ez egyrészt fontos, hogy ne kelljen kilépni az EASY-FILE programból, ha olyan műveleteket akarunk elvégezni, amelyek nem egyértelműen az adatfeldolgozáshoz tartoznak (például formázás, tartalomjegyzék megtekintése stb.). De ide tartozik az archív állományok készítése is, ami nagyon fontos a számítógépes adatfeldolgozásnál.

6. Karbantartás (File Update)

A karbantartás alatt itt nem a szokásos adatállomány karbantartást kell

érteni. Ez a funkció lehetőséget ad, hogy egyszerre egy mezőt módosítsunk az adatállományban azonos adattartalommal. Nem kötelező az egész állományban módosítani az adott mezőt, mert itt is van lehetőség leválogatásra. Itt lehet módosítani az aritmetikai műveleteket, és újra számítani a kiválasztott mezők értékét.

7. Segítségnyújtás (Help)

A segítségnyújtás funkció lehetőséget ad arra, hogyha nem jut eszünkbe hirtelen az EASY-FILE használatáról valami, és nincs a közelünkben a felhasználói kézikönyv, akkor itt elolvashassuk, hogy mit kell tennünk.

8. Kilépés (Exit)

Ez a funkció lehetőséget nyújt visszatérni a Basic-be anélkül, hogy kikapcsolnánk a gépet.

Mielőtt hozzáfognánk az EASY-FILE alkalmazásához, fontos, hogy papíron tervezzük meg a feladatot. Írjuk le, hogy mi a feladat, milyen adatokat akarunk használni, milyen leválogatási feltételeink vannak stb. Annak megfelelően, ahogy azt az első fejezetben leírtuk. A feladatterv elkészítése azért is fontos, mert számolnunk kell az EASY-FILE korlátaival is.

Szükséges tudnivalók az EASY-FILE rekordtervezésének lehetőségeiről és korlátairól:

1. Egy rekord legfeljebb két képernyőt használhat.
2. A képernyőn a mező nevét a tartalmától a ':'-t választja el.
3. A mezőnévnek és a tartalomnak kijelölt helynek egy soron belül kell kezdődni és befejeződni.
4. Egy mező hossza maximálisan 38 karakter lehet.

5. Minden rekordnak legalább egy mezőt kell tartalmaznia.
6. Egy rekord maximálisan 80 mezőből állhat.
7. Egy rekord maximálisan 506 karakterből állhat.
8. Az egykarakteres mezőt §-al kell jelölni.

A rendszer indítása:

Ha minden eszközt bekapcsoltunk az előírásoknak megfelelően, akkor tegyük be az EASY-FILE programlemezt az egységbe. Ezután tölthetjük be a programot.

```
LOAD'EASYFILE2',8,1 (RETURN)
```

Ez után kb. 2 percig villog a képernyőn az EASY-FILE felirat, majd megjelenik a Főmenü.

A főmenüben lévő funkciókat a megfelelő számú billentyű leütésével választhatjuk ki. Ezután a képernyő alján villog az üzenet, hogy tegyük be a programlemezt. Ha megtettük, akkor a RETURN billentyűt kell leütni és kezdődik a kiválasztott program betöltése. A program beolvasásának befejezése után az adatlemezt kell betenni az egységbe. Ez a felirat is villog, amíg le nem ütöttük a RETURN-t. Ez a folyamat minden főmenü és almenü funkció kiválasztásánál így zajlik le, ezért ezt nem fogjuk a továbbiakban minden alkalommal megismételni. Minden menüből lehetőség van abba a menübe visszatérni ahonnan hívtuk.

A FŐMENÜ LEÍRÁSA

MAIN MENU OPTIONS

(fő működési lehetőségek)

1. CREATE DATA FILE

- adatfile létrehozás

Az adatfile rekord- és képernyőtervének létrehozása és módosítása.

Az előző pontban létrehozott adatállomány adatainak kezelése, beleértve a bevitelt is.

2. DATA MANIPULATION - adat kezelés
Az adatok kezelése; beírás, javítás, törlés, keresés, kiterjesztés.
 3. REPORT GENERATION - táblázat készítés
Táblázat készítése adatfile-ból.
 4. SORT DATA FILE - adatfile rendezése
Az adatfile kulcs(ok) szerinti rendezése
 5. SUPPORT UTILITIES - segédprogramok
Kiegészítő segédprogramok, file műveletek
 6. FILE UPDATE - file aktualizálás
Módosítás az adatfile mezőiben.
 7. HELP - segítség
Általános információk a programcsomag működtetéséhez.
 8. EXIT - kijárat
Kilépés az EASY FILE-ból.
- INPUT MENU OPTIONS - beírandó a kívánt EASY-FILE művelet száma

A képernyő közepén megjelenik a

PROGRAM LOADING

felirat, eközben betöltődik a programcsomag megfelelő része.

A főmenüben lévő funkciók részletezése.

A pontos utasításokat a fejezet második felében ismertetjük.

1. Adatfile létrehozás

Ez a funkció is tartalmaz almenüt a következő lehetőségekkel:

1. Képernyő tervezés (Create Screen Layout)
2. Aritmetikai műveletek leírása (Create Arithmetic operations)
3. Képernyő módosítás (Modify Screen Layout)
4. Kilépés (Exit)

Az első adatfeldolgozási munkánál mindenképpen a tervezéssel kell kezdenünk, hiszen még semmi nem áll rendelkezésünkre. A későbbiekben, ha új feladatot akarunk megoldani, szintén a képernyőtervezést kell választanunk. Ezzel az opcióval határozzuk meg az adatállományunk struktúrájának összes jellemzőjét. Ezek a jellemzők a következők:

- a mezők hosszúsága és típusa
- a rekord hossza
- az állomány nagysága (rekordszámban kifejezve)
- titkosítási jelszavak

A képernyőn való tervezésnél fontos tudnunk, hogy a mező nagyságának a kijelölésére használt '<' és '>' jelek beleszámítanak a mező hosszúságába. Minden képernyőnek van egy fejléce, amit központosítva helyezhetünk el az első sorban. A képernyő 22 sorból állhat és 40 oszlopból. A mezőknek kötelező ugyanabban a sorban befejeződni, ahol elkezdődött. Nagyon lényeges a jó képernyőterv kialakítása. Nemcsak arra kell figyelni, hogy az adatrögzítőknek a szemét ne bántsa a rossz adat elhelyezés, de arra is, hogy a képernyő meghatározza a rekordtervünket is. Amennyiben később ezt

változtatni akarjuk, figyelni kell a már lerögzített adatokra.

Rossz tervezéssel a munkánkat megsokszorozhatjuk a felesleges adatrögzítések ismétlésével. A képernyő tervezésének befejeztével elmentjük az adatokat az adatlemezre, ha valóban megfelelőnek találjuk.

Kiírásra kerülnek az általunk meghatározott file jellemzők, a rekord hossza byte-okban, és az összes mező darabszáma.

Ezután következik a file jelszó szintjének a meghatározása. Ha kiválasztottuk a megfelelő szintet, akkor beírhatjuk a jelszavakat is. Nagyon fontos, hogy ezeket a jelszavakat megjegyezzük, mert ha elfelejtjük, többet nem tudjuk használni az adatállományunkat. Az első alkalommal, amíg nem sajátítottuk el az összes lehetőséget, célszerű nem használni jelszavakat. Később lehetőség van ennek módosítására.

Itt határozhatjuk meg az adatállományunk nagyságát rekord darabszámmal. Ha ez véletlenül nem volt elég, később még növelhetjük. Az állományt ebben a munkafázisban hozza létre a program. A képernyőt 'név.for' nevű formátum file-ba menti el. Az adatállományt 'név.dat' kiterjesztéssel hozza létre.

Az aritmetikai (számtani) műveletek meghatározásánál fontos tudnunk, hogy a négy alapművelet áll rendelkezésünkre, ezek:

- + összeadás
- kivonás
- × szorzás
- / osztás

A mezők között összesen 24 numerikus mező megengedett. Műveletet nemcsak a mezők között, hanem a regiszterek között, illetve mezők és regiszterek között is lehet végezni. Az eredményeket elhelyezhetjük a rekordban, vagy felhasználhatjuk a táblázat készítésénél is. Amennyiben megfelelőnek találjuk a műveleteink felsorolását, ezt is elmenthetjük az adatlemezünkre.

Ezek a műveletek minden egyes rekord felvitelnél végrehajtásra kerülnek.

A képernyő módosításnál ugyanazok a szabályok érvényesek, mint a tervezésnél. Itt még egyszer felhívjuk a figyelmet arra, hogy lerögzített adatállomány képernyőjét, ami azonos a rekordtervvel, nem szabad tartalmilag módosítani. Tehát alfanumerikus mező helyébe nem írhatunk numerikust, és megfordítva. A mezők hosszúságát sem szabad módosítani, mert nem biztos, hogy követni tudjuk anélkül, hogy az adatállományunkat el ne rontanánk.

Ha befejeztük a munkát ebben a menüben, visszatérhetünk a főmenübe.

1. CREATE DATA FILE - ADATFILE LÉTREHOZÁS

A továbbiakban az EASY-FILE vezérlésének leírása következik.

A főmenü első működési lehetősége, amely további választási lehetőséget tartalmaz.

- | | |
|---------------------------------|--|
| 1. Create screen layout | - a rekordkép megtervezése |
| 2. Create arithmetic operations | - az aritmetikai műveletek leírása |
| 3. Modify screen layout | - a rekord-kép módosítása |
| 4. Exit to main menu | - kilépés a főmenübe |
| INPUT MENU OPTION_ | - 1-4-ig a szükséges művelet számának beírása |
| 1.1 CREATE SREEN LAYOUT | |
| INPUT NUMBER OF SCREENS_ | - az input képernyők száma |
| | - a rekord méretétől függően - 1 vagy 2 |
| DESIGN SCREEN n | - az n. képernyő kialakítása, megtervezhető a rekord kép, kilépés f8 |

Miután befejeztük a képernyő tervezését még van lehetőségünk javítani.

ARE YOU SURE Y/N	- bizonyos vagy ? Y=igen, N=nem
SAVING SCREEN	- kép tárolás
RECORD SIZE = nnn	- a rekordhossz kiírása
No.of fields = nn	- a mezők száma
ENTER PASSWORD LEVEL__	- a védelmi szintek kulcs- szavainak beírása az alábbiak közül
0. file only	- csak file-ra
1. file and record	- file-ra és rekordra
2. file and field	- file-ra és mezőkre
3. all three	- mindháromra
4. none	- egyikre sem

Ha kértünk védelmet, értelemszerűen kéri a jeszavak beírását.

ENTER FILE PASSWORD__	- file védelmi jelszó beírása
ENTER RECORD PASSWORD__	- rekord jelszó beírása
ENTER FIELD PASSWORD__	- mező jelszó beírása
ALL FIELDS PROTECTED Y/N	- minden mezőt védesz ? Y=igen, N=nem

ENTER PROTECTED FIELD NUMBERS PRESS RETURN TO EXIT - írd be a védeni kívánt mezők sorszámát és a végén üss RETURN-t.

ENTER FIELD NUMBER__	- írd be a mező rekordon belüli sorszámát
ENTER FILE NAME__	- írd be az adathalmaz nevét

Ezután megjelenik a következő felszólítás:

INSERT DATA DISK AND PRESS RETURN - tedd be az adatlemezt és üss RETURN-t

SAVING SCREEN FORMAT FILE - a képformátum-file elmentése - folyik

MAXIMUM RECORDS AVAILABLE nnnn - nnnn a rekordok lehetséges száma - a lemezen

NUMBER OF RECORDS REQUIRED? - a lefoglalandó helyek száma - ahány rekordnak helyet akarunk biztosítani a lemezen - később még növelhető a 2.5.-ben

CREATING DATA FILE

- az adatfile létrehozása -folyik

1.2.CREATE ARITHMETIC OPERATIONS

ENTER FILE NAME

- file név beírása, egy már meglévő file neve, aminek mezőivel aritmetikai műveleteket akarunk végeztetni.

INSERT DATA DISK AND PRESS RETURN

- helyezd el az adatlemezt és üss RETURN-t

READING FORMAT FILE

- a format file beolvasása

ENTER ARITHMETIC OPERATIONS

- írd be az aritmetikai műveleteket , kilépés f8

ARE YOU SURE Y/N

- biztos vagy?

SAVING ARITHMETIC FILE

az aritmetikai műveletek eltárolása

A képernyőtervének módosításával vigyázni kell, mert ez értelemszerűen a rekord módosítását is jelenti.

1.3.MODIFY SCREEN LAYOUT

ENTER FILE NAME

- a módosítani kívánt képernyőhöz tartozó file nevének beírása

DESIGN SCREEN n

- bemutatja a rekordképet, ami egyszerű átírással módosítható. A továbbiakban az 1.1 szerint működik.

1.4.EXIT TO MAIN MENU

Visszalépés a főmenübe.

2. Adatmanipuláció

Ez a főmenü opció tartalmazza a legtöbb lehetőséget az adatállományok karbantartására. Az almenü a következő részekből áll:

1. Adatbevitel (Enter)
2. Javítás (Amend)
3. Törlés (Delete)
4. Keresés (Search)
5. Bővítés (Extend)
6. Kilépés (Exit)

Mindegyik funkciónál betölti a program a formátumfile-t. A filenév megadása után meg kell adni az adatállomány létrehozásánál beírt jelszavakat. Mindegyik opciónál lehetőség van rekordonként befejezni a műveletet, vagy tovább folytatni a munkát a következő rekorddal. Fontos, hogy az egyes rekord műveletek után mentsük el a rekordot az adatlemezre, mert különben elvesznek.

Az első funkció a már megtervezett képernyőt tölti fel adatokkal, majd ebből kialakítja a rekord struktúrát. Az adatbevitel mindig a már rögzített adatrekordok után írja az új rekordokat, a rögzítés sorrendjében.

Nem kötelező minden mezőt kitölteni a képernyőn. Van azonban néhány megkötés, amit kötelező betartani, a program nem is engedi ezek megszegését. Így numerikus mezőbe nem írhatunk csak számjegyeket és tizedespontot. Fontos megjegyezni, hogy a számítástechnikában nem tizedesvesszőt, hanem tizedespontot használunk. A mezőknél nem tölthetünk ki több karaktert, mint amennyit meghatároztunk a képernyőterven. Az természetesen megengedett, hogy ne töltsük ki teljes hosszában a mezőket. Ha befejeztük az adatbevitelt az adott rekordban, akkor azt el kell menteni az adatlemezre. Az elmentés az 'f8' billentyűvel történik. Ha meghatároztunk aritmetikai műveleteket, azokat a mentés parancs kiadása után, de a lemezreírás előtt fogja végrehajtani.

A következő funkció ad lehetőséget a lemezen lévő rekordok javítására illetve módosítására. Sokszor előfordul az adatfeldolgozás során, hogy módosítani kell az egyes mezőket, például valamilyen előírás megváltoztatása, vagy rossz rögzítés miatt. Ezt a módot csak akkor tudjuk használni, ha ismerjük a rekordunk sorszámát. A program kiválasztja az általunk megadott sorszámú rekordot és kiírja képernyőre. Ezután kijavíthatjuk a szükséges mezőket. Ha nem az egyes rekordok mezőit akarjuk javítani, hanem az egész adatállományon áthúzódik a javítás, például az adókulcs megváltozott, akkor célszerűbb az UPDATE opciót használni. Ha nem tudjuk a rekord sorszámát, akkor nem ezzel az funkcióval kell a javítást elvégezni.

A következő funkció a rekordtörlés. A rekordok kitörlésénél a helyet a lemezen nem szabadítja fel, ezt nekünk később kell megoldani egy segédprogrammal. A rekordtörlésnél feltétlen szükségünk van a rekord sorszámára. Ha nem ismerjük, akkor a keresési funkcióval megkeressük a rekordot és megállapítjuk sorszámát. Ezután már folyamatosan használhatjuk a törlés lehetőségeit. A kiválasztott rekordot nem feltétlenül kell törölnünk, mert minden törlésre kijelölt rekordnál megkérdezi, hogy valóban törölni akarjuk-e. Ha a törlés funkcióban vagyunk, de nem vagyunk biztosak a rekord tartalmában, akkor megnézhetjük azt.

A következő funkció a keresés, amely saját menüvel rendelkezik:

1. Az 'X' mezőben az 'Y' értékét keressük
(Search field X containing Y)
2. Bármely mezőben az 'Y' értékét keressük
(Search Y in any field)
3. Egy rekord olvasása
(Get a record)
4. Lépés előre/hátra
(Step forwards/backwards)
5. File tartalom kiíratása folyamatosan.
(Screen dump)

6. Kilépés (Exit)

Az első három pontban a megadott feltételek alapján a program kikeresi a rekordot, amelyet a megjelenítés után javítani is tudunk. Ezek az opciók akkor fontosak, ha nem tudjuk a rekord sorszámát. Ha tudjuk, hogy adott mezőnek mi az értéke, vagy tudjuk, hogy egy rekordunkban előfordul valamilyen érték, de nem tudjuk melyik mezőben, akkor használhatjuk az első két keresési lehetőséget. A feltételek, amelyek alapján kikeresethetjük a rekordot a következők lehetnek:

1. 0 (= egyenlő)
2. 1 (< kisebb)
3. 2 (> nagyobb)
4. 3 (<= kisebb vagy egyenlő)
5. 4 (>= nagyobb vagy egyenlő)

Az előre/hátra lépést akkor célszerű használni, ha valamiért elhagytuk a nekünk szükséges rekordot. A filetartalom kiíratásánál a képernyő formátumában folyamatosan írónak ki az egyes rekordok.

A következő funkció az adatállomány kibővítése. Ezt a műveletet akkor kell használni, ha a generálásnál rosszul ítéltük meg az adatállomány nagyságát. Itt hasonlóan, mint a létrehozásnál, a képernyőn kiírja, hogy hány rekord fér még a lemezre és megkérdezi, hogy hány rekorddal akarjuk kibővíteni. Maga a bővítés körülbelül annyi időt vesz igénybe, mint a létrehozás.

Ugyanezeket a keresési módokat használja a programcsomag mind a táblázat készítésnél, mind a karbantartásnál, ezért ott ezeket már nem részletezzük.

Az adatkezelés leírása

2. DATA MANIPULATION MENU

- | | |
|-----------|----------------|
| 1. ENTER | - beírás |
| 2. AMEND | - javítás |
| 3. DELETE | - törlés |
| 4. SEARCH | - keresés |
| 5. EXTEND | - kiterjesztés |
| 6. EXIT | - kilépés |

2.1. ENTER

ENTER FILE NAME_ - írd be a -feltöltendő- file nevét
Megjelenik a megtervezett képernyő.

A kitöltendő adatok helyén villog a cursor. A kitöltött rekordok elmentése f8-cal történik.

Lehetséges hiba:

END OF FILE ENCOUNTERED - a rekordok részére lefoglalt hely elfogyott

Ha szükséges még adatot beírni ,akkor a 2.5 szerint bővítsük a file-t, térjünk vissza a 2.1-be és folytatható a bevitel.

2.2. AMEND

ENTER RECORD NUMBER_ - írd be a -javítandó - rekord sorszámát

Megjelenik a képernyőn a javítandó rekord, javítás után mentés f8-cal.

SAVING RECORD - a rekord elmentése/folyik/

CONTINUE Y/N - folytatod? /a javítást/

Y=igen, N=nem

A rekordok törlésénél figyelembe kell venni, hogy a rekord helye nem szabadul fel, ezért a segédprogramok sűrítési funkciójával kell összetömöríteni a rekordokat.

2.3. DELETE

- ENTER FILE NAME__ - írd be a file nevét - amiből törölni akarunk
- READING FORMAT FILE - a formátum file beolvasása - folyik
- ENTER NUMBER OF RECORDS__ - írd be a rekordok számát - ahányat törölni kell
- ENTER START DELETE RECORD- írd be hányadik rekordnál kezdődjön a törlés

Megjelenik a törlendő rekord, és:

- ENTER C/D/S/T
- C lépj a következő rekordra
 - D töröld
 - S lapozz a második képre - ha van
 - T lépj ki a törlésből

Ezek a rövidítések a továbbiakban is érvényesek, ezen kívül:

- A - módosítható, javítható a rekord tartalma
- + - lépj előre egy rekordot
- - - lépj vissza egy rekordot

- ARE YOU SURE Y/N
- Y - elvégzi a törlést
 - N - visszaadja a kiválasztás lehetőségét

Az adatállományban való keresés lehetőséget ad arra, hogyha nem ismerjük a rekord sorszámát, akkor kiválaszthassuk.

2.4. SEARCH

1. Search field X containing Y - X-ik mezőben Y keresése
2. Search Y in any field - Y keresése az összes mezőben
3. Get a record - rekord beolvasása
4. Step forwards/backwards - lépkedés előre hátra
5. Screen dump - adatok bemutatása a képernyőn
6. Exit - kilépés

INPUT MENU OPTION_ - 1-6-ig a szükséges művelet számának beírása

2.4.1.

ENTER FIELD NUMBER_ - írd be a mező sorszámát
ahányadik a rekordban

ENTER SEARCH STRING_ - írd be a keresendő láncot
Megjelenik a keresett láncot tartalmazó rekord, és:
ENTER A/C/S/T

2.4.2.

ENTER SEARCH STRING_ - írd be a keresett láncot

Megjelenik az a rekord, amelyikben először megtalálta a láncot.

ENTER A/C/S/T

2.4.3.

ENTER RECORD NUMBER__ - írd be a beolvasandó rekord
sorszámát

Megjelenik a beolvasott rekord.

ENTER A/C/S/T

2.4.4

ENTER RECORD NUMBER__ - írd be a rekord sorszámát

Megjelenik a megadott rekord.

ENTER A/S/T/+/-

2.4.5

Bemutatja a képernyőn a rekordok tartalmát - megállítható és tovább
indítható SPACE-szel.

2.4.6

Visszatér 2.-ba.

2.5.

MAXIMUM RECORDS nnnn - a felírható rekordok száma -
ennyi fér a lemezre

NUMBER OF RECORDS REQUIRED ? - a továbbiakban szükséges
rekordhelyek száma

EXTENDING DATA FILE - az adatfile kiterjesztése
folyik

2.6.

Visszatérés a főmenübe.

3. Táblázat (report) készítés

A táblázatgeneráló funkció lehetőséget ad az adatállomány valamennyi illetve valamilyen szempont szerint leválogatott rekordjainak kilistázására.

A táblázatkészítő funkció is rendelkezik saját menüvel:

1. Billentyűzetről (Keyboard)
2. Paraméter file-ból (Report file)
3. Kilépés (Exit)

Paraméterfile-t csak akkor használhatunk, ha már egyet létrehoztunk, és elmentettünk az adatlemezünkre.

A táblázat készítés paraméterek megadásával történik. A paramétereket, ha a későbbiekben fel akarjuk használni, tárolhatjuk lemezen.

A táblázat készítés folyamata.

Először meg kell adni a leválogatási feltételeket, hasonlóan ahhoz, ahogy a keresési feltételeket megadtuk. Ezután történik a táblázat rendezettségének kialakítása, amely lehetséges a már meglévő mutató file megadásával, amint azt majd a Rendezési funkciónál leírjuk. Ezek után következik a táblázat paramétereinek megadása. A táblázat paramétereinél a következőket kell megadni:

1. Terület (area - nyomtatási sorrendet határoz meg)
2. Tartalom (contns - mező számot kell megadni)
3. Típus (type - a program írja ki)
4. Hossz (length - a program írja ki)
5. Az oszlop fejlécének megnevezése (title)
6. Az oszlop kezdő pozíciója (start column)

7. Aritmetikai műveletet kell-e végezni (arithmetic)
8. Összeg fokozat képzése szükséges-e (accumulation)
9. Magyarázó szöveg a felhasználónak (user message)

Ha befejeztük a paraméterek megadását, lehetőségünk van az elmentésére egy paraméter file-ba. Ha elmentjük, a későbbiekben felhasználhatjuk, mint azt a táblázatkészítés menüje is mutatja. Ha később szükséges, a paraméter file-t javíthatjuk is.

A táblázatunkat elkészíthetjük képernyőre. Erre akkor van szükség, ha még csak próbáljuk a táblázatunkat, vagy vezetői információt készítünk, ami elfér egy képernyőn is. Természetesen ehhez feltételezzük, hogy a vezetőnél is van Commodore64 számítógép. A táblázatunkat elkészíthetjük lemezre is, mert még nem tudjuk, hogy hány példányban szükséges a későbbiekben kinyomtatni. A legtermészetesebb a nyomtatóra készülő táblázat. Itt nagyon fontos, hogy megfelelően válasszuk ki a nyomtató típusát.

A táblázat készítés leírása

REPORT GENERATION MENU

1. KEYBOARD - a paramétereket billentyűzetről lehet megadni
2. REPORT FILE - az előzőleg eltárolt táblázat-paraméter file-ból kapja a paramétereket
3. EXIT - kilépés

3.1.

- ENTER FILE NAME_ - írd be a file nevét - amiből a táblázat készül

INSERT DATA DISK AND PRESS RETURN - tedd be az adatlemezt és üss RETURN-t

READING FORMAT FILE - a formátum file beolvasása folyik
 ENTER FILE PASSWORD - írd be a jelszót

Kétféle input lehetőség van az adatokra vonatkozóan:

FILE OPTIONS - file lehetőségek
 1. DATA FILE - adatfile
 2. POINTER FILE - mutató file, csak rendezés után
 INPUT OPTION_ - a válasz 1, ha adatfile-ból,
 - a válasz 2, ha rendezés -SORT-
 funkció után készítünk kivonatot

Háromféle kimenet (output) lehetőségünk van az adatok megjelenítésére, tárolására.

OUTPUT DEVICE OPTIONS - eredmény tároló-megjelenítő készülék
 lehetőségek

1. PRINTER - nyomtató
 2. SCREEN - képernyő
 3. DISK - lemez
 INPUT OPTION_ - a megjelenítés ill. tárolás kívánt
 módja, 1-3-ig

3.1.1.

PRINTER OPTIONS

1. VIC 1515/1525

2. MODEL 1526

INPUT OPTION_ - 1 vagy 2, attól függően, hogy milyen
 nyomtatóval rendelkezünk.

Figyelmesen válaszoljunk a kérdésekre, mert az egész eddigi munkánk kárba veszhethet.

3.1.2.

SCREEN

- nincs üzenet, nincs tennivaló, a készített táblázat a megjelenés után elveszik

3.1.3.

DISKETTE

- tárolja az elkészített táblázatot, de a táblázat paramétereit nem (paraméter tárolás később).

SEARCH PARAMETERS

- a keresés paramétereit

ENTER NUMBER OF SEARCH KEYS_

- írd be a keresés kulcsainak számát, hányféle kulcs szerint keressen

SEARCH CRITERIA

- a keresés ismérvei

ENTER FIELD NO.

- írd be a mező számát

ENTER SEARCH CRITERIA_

- 0 ha egyenlő
- 1 ha kisebb
- 2 ha nagyobb
- 3 ha kisebb egyenlő
- 4 ha nagyobb egyenlő

ENTER SEARCH STRING_

- írd be a keresett láncot - amit keresünk

ARE YOU SURE Y/N

- biztos vagy? ha N, újra paraméterezhető

ENTER NUMBER OF DECIMAL PLACES_

- írd be a tizedes jegyek számát ahány tizedes pontosságot akarunk látni a megjelenítéskor

ENTER NUMBER OF PRINT LINES/RECORD_ - írd be a sor/rekord számát

Ide a sor/rekord számot kell beírni 1 és 3 között, attól függően, hogy egy rekordot hány sorban akarunk megjeleníteni

Lemez outputnál:

ENTER SCREEN WIDTH_ - írd be a képernyő
szélességét - max.40

Nyomtató outputnál:

ENTER FORM WIDTH/NO OF COLUMNS/ - írd be a nyomtatási formátum
szélességét, max 80

DO YOU WANT DOUBLE SPACING Y/N - kívánsz-e dupla üres helyet
N = normál üres helyek,
Y = dupla üres helyek

DO YOU WANT PAGE NUMBERING Y/N - kívánod-e a lapszámozást
igen/nem

ENTER REPORT TITLE_ - írd be a táblázat nevét -
ami a táblázat neve lesz

ENTER NUMBER OF PRINT AREAS_ - írd be a nyomtatandó mezők
darabszámát
ha itt RETURN-t adunk,
megkérdezi:

ABORT Y/N - megszakítsam?
Y válasznál leállítja a
paraméterezést

ENTER AREA CONTENTS_ - írd be a területek tartalmát
- azt a számot várja, hogy a
kiírandó mező hányadik
a rekordban

A hozzátartozó típus és hossz információkat a gép írja ki.

TYPE A - ha alfanumerikus - lánc típusú adat,
 N - ha numerikus - szám típusú adat

ENTER AREA TITLE__ - írd be a mező megnevezését
 - a mező nevét, amit a táblázat-
 ban az oszlop fölött
 akarunk látni

ENTER START COLUMN NUMBER__ - írd be a kezdő pozíció
 számát, ahol az adat megjele-
 nítése kezdődjön

ARITHMETIC REQUIRED Y/N - aritmetikai művelet lesz?
 ha N, akkor elindul a táblá-
 zat készítés

Különben megkérdezi:

ENTER OPERATION__ - írd be a műveletet
 +, vagy -, vagy *, vagy /,

ENTER OPERAND__ - írd be az operandust -
 azt a mezőszámot, amelyikkel
 műveletet kell végezni

ACCUMULATION REQUIRED Y/N - összegzés szükséges?
 ha Y (igen), összegezi az
 eredményeket, és végül kiírja

WISH TO SAVE REPORT PARAMETERS Y/N

Szükséges-e a táblázat paraméterek eltárolása? igen/nem. Ha Y (igen), akkor eltárolja és később REPORT PARAMETER FILE - ként újra indítható a táblázat készítés.

Ha N (nem), akkor a táblázat elkészülte után elvesznek a paraméterek.

ENTER FILE NAME__	- írd be a (táblázat-paraméter) file nevét
SAVING REPORT PARAMETERS	- a táblázat paramétereinek elmentése folyik
GENERATE REPORT Y/N	- készüljön a táblázat? ha N, visszatér 3.-hoz

Nyomtatásnál:

CONTINUOUS FORMS Y/N	- folyamatos összefüggő formát kívánsz? Y = folyamatosan nyomtat N = lapokat nyomtat
SWITCH ON THE PRINTER	- kapcsold be a nyomtatót
SET PAPER AND PRESS RETURN	- állítsd be a papírt és üss RETURN-t

Diskettnél:

.ENTER FILE NAME__	- írd be a /készülő táblázat/ nevét
SAVING REPORT	- a táblázat elmentése -folyik

a képernyőre készül: SPACE gombbal megállítható bármennyi időre és tovább indítható.

3.2.

ENTER FILE NAME__	- írd be a /táblázat paraméter/ file nevét
-------------------	--

DO YOU WISH TO MODIFY PARAMETERS Y/N - akarod módosítani a paramétereket?

Ha N, elkezdődik a táblázat készítése.

Ha Y, visszatér az ENTER NUMBER OF PRINT AREAS kérdéshez, és innen lehet újra paraméterezni a táblázatkészítést.

4. Rendezés

Ha a rendezési funkciót választottuk ki a főmenüből, akkor nem menü jelenik meg, hanem a rendezés paramétereinek megadásához szükséges képernyő. Ez a képernyő annyiszor vált, ahány rendezési kulcsot adtunk meg az elején.

A rendezési paramétereket tárolhatjuk mutató (pointer) file-ban, amit később felhasználhatunk például a táblázat készítésénél.

A paramétereknél meg kell adni a rendezési kulcs sorszámát, a mező sorszámát, a rendezés minőségét (növekvő/csökkenő). Ha minden paramétert megadtunk, és (ha szükség volt rá) elmentettük a mutató file-t, akkor megtörténik a rendezés.

Az adatfile rendezés leírása

A főmenü negyedik működési lehetősége. Az adatfile növekvő vagy csökkenő rendezésére használjuk.

4. SORT DATA FILE

ENTER FILE NAME_ - írd be a (rendezendő) file nevét

READING FORMAT FILE

ENTER FILE PASSWORD - írd be a jelszót
(és a többit, ha van)

ENTER NUMBER OF SORT KEYS_ - írd be a rendezési kulcsok
számát (hány szempont
szerint kérjük a rendezést)

ENTER FIELD NUMBER_ - írd be a mező sorszámát,

olyan sorrendben, ahogy a
rendezést el akarjuk végezni

A gép kiírja a mező típusát és hosszát:

TYPE	- típus
LENGTH	- hossz
ORDER_	- kérdésre A vagy D a válasz A - (ascending) növekvő D - (descending) csökkenő sorrendben rendezzük a rekordokat
ARE YOU SURE Y/N	- Y elindul a rendezés - N előről indulhat a paraméterezés
SORT IN PROGRESS	- folyik a rendezés
NUMBER OF SORTED RECORDS=nnnn (a rendezett rekordok száma)	
PRESS ANY KEY TO RETURN	- nyomj meg bármilyen gombot (a főmenübe való visszatéréshez)

5. Segédprogramok

Ezek a programok azt a célt szolgálják, hogy egy sor lemez vagy nyomtatási műveletet meg tudjunk oldani anélkül, hogy kilépnénk az EASY-FILE-ből. A következő menü lista áll rendelkezésünkre:

0. Lemez formázás (Format)
1. Tartalomjegyzék készítés (Summary)
2. Archiválás (Backup)
3. Átvitel (Transfer)

4. Alfile készítés (Subfile)
5. Nyomtatás (Print)
6. Jelszó csere (Change password)
7. Törlés (Delete)
8. Sűrítés (Compress)
9. Átnevezés (Rename)
- e. Kilépés (Exit)

Ezeknél a funkcióknál nem fogjuk elmagyarázni azokat a műveleteket, amelyeket a Commodore-64 gép használatánál már úgylis elsajátított minden felhasználó. A fentebb felsorolt menüből ezek a következők: a formázás, tartalomjegyzék, átvitel, törlés, átnevezés.

A lemezműveletek elvégzésénél a programcsomag nagyon óvatosan jár el, mert mielőtt végrehajtaná az általunk kiadott parancsot, visszakérdez, hogy valóban azt akarjuk csinálni, amit kértünk.

Az adatlemez archiválása nagyon fontos az adatfeldolgozásnál, mint azt majd a negyedik fejezetben leírjuk. Mielőtt elkezdjük az archiv adatállomány készítését, a lemezt formálni kell.

Az alfile készítő funkció az adatállományból létrehozhat egy mellékállományt, amelynek segítségével részkimutatásokat készíthetünk. (Például havi kimutatást.) Ebben az esetben a teljes állomány adatbevitelével egyidejűleg elkészít egy alfile-t (például a tárgy havi rekordokból). Ezt az adott időszak letelte után törölni kell, hogy a következő feldolgozásnál megint elkészíthessük az alfile-t.

Még egy nagyon lényeges funkcióval rendelkezik a segédprogramrész. Ez a lehetőség a nyomtatás. Itt nem táblázatokat készítünk, hanem lehetőség van az adatállományról a fontosabb információk kinyomtatására.

Segédprogramok leírása

A főmenü ötödik működési lehetősége.

5.UTILITY MENU

- | | |
|--------------------|---|
| 0. FORMAT | - lemez formálás |
| 1. SUMMARY | - összefoglalás, tartalomjegyzék a lemeztől |
| 2. BACKUP | - másolat az egész lemeztől |
| 3. TRANSFER | - file áthelyezés |
| 4. SUBFILE | - részfile, alfile kijelölés |
| 5. PRINT | - megjelenítés nyomtatón vagy képernyőn |
| 6. CHANGE PASSWORD | - jelszó csere |
| 7. DELETE | - file törlés |
| 8. COMPRESSE | - file sűrités |
| 9. RENAME | - átnevezés |
| e. EXIT | - kilépés (a főmenübe) |

5.0.FORMAT

- a lemez előkészítése adatok felvitelére, sávok kijelölése, inicializálás. A régi tartalom - ha volt - elveszik.

ENTER DRIVE NUMBER__

- írd be a készülékszámot
0, ha egy, 1, ha kettő lemezeget használunk

ENTER DISK NAME__

- írd be a lemez azonosítóját -
2 karakter hosszú azonosító

INSERT SCRATCH DISK__

- helyezd el a törlendő
(formalandó) lemezt

Figyelmesen kell itt eljárni, mert a formált lemezeiről eltűnnek az adatok!

ARE YOU SURE Y/N

FORMATTING THE DISK - a lemez formatálása folyik

5.1.SUMMARY - listát készít a képernyőre
a lemezen lévő összes file
nevéről és/vagy a lemezen levő
szabad helyről

DO YOU WANT PARTIAL SUMMARY Y/N

- részleges összefoglalást kívánsz?

- Y csak a szabad blokkok számát
írja ki
- N teljes tartalomjegyzéket készít,
a szabad blokkok számát is kiírja

5.2.BACKUP

- másolatot készít egy előzőleg
formált lemezre az adatlemezeiről

ENTER DRIVE OF SOURCE - írd be a forráskészülék számát
0, ha egy lemezegegyéssel dolgozunk

ENTER DRIVE NUMBER OF DESTINATION_

- írd be a célkészülék számát
- 0, ha egy lemezegegyéget
használunk, 1, ha kettőt

INSERT SOURCE DISK - helyezd el a forráslemezt - RETURN

READING BAM OF SOUR - forráslemez információk beolvasása
kiírja a képernyő tetejére
a forráslemez nevét
és megkérdezi:

Fontos, hogy nagyon figyelmesen készítsük el az adatainkról az archív lemezt

ARE YOU SURE Y/N - biztos vagy? ha igen, Y a válasz

READING DATA FROM SOURCE - az adatok beolvasása a forrás-
lemezről

INSERT DESTINATION DISK - helyezd el a céllemezt

READING BAM OF DESTINATION - a céllemez jellemzőinek
beolvasása - kiírja a képernyő
tetejére a céllemez nevét

ARE YOU SURE Y/N - biztos vagy? ha igen, Y válasz

WRITING DATA OF DESTINATION - az adatok céllemezre írása
folyik

BACKUP OVER - túl a másoláson

5.3.TRANSFERRING FILES - adatfile,táblázatfile,táblázat-
paraméter, al-file és képernyő-
terv-file másolás, egyik lemez-
ről a másikra.
Figyelem : csak két lemezegységnél
javasolt!

ENTER SOURCE DRIVE NUMBER_ írd be a forráskészülék számát
0 vagy 1 , egy vagy kettő
készülék

ENTER FILE NAME_ - írd be az átviendő file nevét
kiterjesztéssel

file-név.for - rekord formátum file
 file-név.dat - adat file
 file-név.ptr - pointer file
 file-név.rpt - report data file
 a report paraméter és az alfile-hoz
 nem kell kiterjesztés

ENTER DESTINATION DRIVE NUMBER__

- írd be a célkészülék számát - 0 vagy 1 lásd a forrás
 egységnél

ENTER FILE NAME__

- file név ugyanazzal
 a kiterjesztéssel

TRANSFERRING FILE

- a file átvitele folyik

TRANSFER OVER

- túl az átvitelen

5.4.SUBFILE

- részfile, alfile létrehozása

ENTER SOURCE DRIVE NUMBER__

- írd be a készülék számát
 - feltöltése a főfile feltöltésével
 párhuzamosan folyik

ENTER FILE NAME__

- írd be a (fő)file nevét

INSERT DATA DISK AND PRESS RETURN

- helyezd el az adatlemezt és üss RETURN-t

READING FORMAT FILE

- a formátum file beolvasása folyik

ENTER SUBFILE NAME__

- írd be az alfile nevét

5.5.PRINT

- megjelenítés a képernyőn vagy
 a nyomtatón

Ez a lehetőség nem helyettesíti a táblázat készítést.

PRINT OPTIONS	- megjelenítési lehetőségek
1. DATA FILE	- adatfile
2. SCREEN LAYOUT FILE	- képernyőterv file
3. HEADER FILE	- file jellemzők
4. EXIT	- kimenet
INPUT OPTION_	- 1-4-ig a szükséges működés
OUTPUT DEVICE OPTIÓNS	- kimenő készülék lehetőségek
1.PRINTER	- nyomtató
2.SCREEN	- képernyő
INPUT OPTION_	- 1 vagy 2 ,attól függően, hogy nyomtatóra vagy képernyőre kérjük az outputot
PRINTER OPTION	- nyomtató lehetőségek
1.VIC 1515/1525	
2.MODEL 1526	
INPUT OPTION_	- 1, vagy 2, attól függően,hogy melyik nyomtatóval rendelkezünk
PRINTING IN PROGRESS	- a kért file megjelenítése következik
5.6.CHANGE PASSWORD	- jelszó csere
ENTER FILE NAME_	- írd be a file nevét, aminek a jelszavát cserélni akarjuk

INSERT DATA DISK AND PRESS RETURN - tedd be az adatlemezt és üss RETURN-t

READING FORMAT FILE - a formátum file beolvasása folyik

ENTER FILE PASSWORD_ - írd be a file jelszavát -
amit meg akarunk változtatni

ENTER PASSWORD LEVEL_ - az 1.1. szerint ismételjük meg
a file védelmi teendőket

SAVING FORMAT FILE - a formátum file elmentése folyik

5.7.DELETE - file-ok törlése

ENTER DRIVE NUMBER_ - írd be a készülék számát 0, vagy 1,
attól függően, hogy egy vagy
két lemezegységet használunk

ENTER FILE NAME_ - írd be a törlendő file nevét

DELETING FILE/S - a file-ok törlése folyik

DELETING OVER - túl a törlésen

5.8.COMPRESS - a file sűrítése - a törlés után
hozzá nem férhető területek
felszabadítása történik

ENTER FILE NAME_ - írd be a sűrítendő file nevét

INSERT DATA DISK AND PRESS RETURN - tedd be az adatlemezt,és üss RETURN-t

READING FORMAT FILE - a formátum file beolvasása folyik

PROCESSING - eljárás - folyik a sűrítés

Az átnevezés a tartalomjegyzékben (directory) változtatja meg a file nevét.

5.9.RENAME	- file átnevezés
ENTER DRIVE NUMBER_	- írd be a készülékszámot 0 vagy 1
ENTER NEW FILE NAME_	- írd be az új file nevet - kiterjesztéssel együtt!
ENTER PRESENT FILE NAME_	- írd be a jelenlegi file nevet - amit át akarunk nevezni
RENAMING FILE	- a file átnevezése folyik
RENAMING OVER	- túl az átnevezésen

6. Karbantartás (Update)

A karbantartási funkció értelmét itt nem fogjuk megmagyarázni, mert ezt már megtettük a főmenü funkció leírásánál.

A karbantartást elvégezhetjük az egész adatállományra vagy meghatározott szempontok alapján leválogatott rész állományra. A karbantartási funkció menüje:

1. Mezőtartalom felülírás
(Replace field contents)
2. Aritmetikai művelet módosítása
(Arithmetic update)
3. Kilépés
(Exit)

Az első művelet végrehajtásánál meg kell adni a mezőt, amelyet felül

akarunk írni, és a tartalmat, amellyel felül akarjuk írni azt a mezőt. Ezután kerül végrehajtásra a karbantartási művelet, az általunk meghatározott rekordokon. Például a bérelszámolásnál az egyéni adókulcs megváltozott. Az összes rekordban felül kell írni ezt a mezőt a keresettől függően. A keresési funkciónál leírtak alapján kiválasztjuk a rekordjainkat, beírjuk hányadik mező az adókulcs és mi a megváltozott tartalma. Ezután minden kiválasztott rekordban felülírja az általunk megadott értékkel.

A második menü funkciónál az aritmetikai műveleteinket módosíthatjuk, vagy újakat írhatunk. Az aritmetikai műveletek leírása után a kiválasztott rekordokra elvégzi a számításokat. Vigyázni kell, ha nem minden esetben akarjuk ezen műveletek végrehajtását, akkor majd vissza kell állítani az eredeti aritmetikai műveleteket.

Az UPDATE (karbantartás) leírása

Az EASY-FILE hatodik működési lehetősége. Lehetséges mezőtartalmat átírni, és/vagy az aritmetikai műveleteket megváltoztatni.

- | | |
|---------------------------|--|
| 6. UPDATE | - aktualizálás |
| ENTER FILE NAME_ | - írd be a file nevét |
| READING FORMAT FILE | - a formátum file beolvasása |
| ENTER FILE PASSWORD_ | - írd be a védelmi szintek jelszavait |
| FILE UPDATE | |
| 1. REPLACE FIELD CONTENTS | - mezőtartalom átírása |
| 2. ARITHMETIC UPDATE | - az aritmetikai műveletek aktualizálása |
| 3. EXIT | - kilépés az UPDATE funkcióból |
| INPUT OPTIONS_ | - 1-3-ig a szükséges működés |

Ez a funkció az adatállományban lehetővé teszi egy meghatározott mező felülírását a kiválasztott rekordokban.

6.1. REPLACE FIELD CONTENTS

ENTER NUMBER OF SEARCH KEYS_ a keresési kulcsok számát hányféle szempont szerint keresse. Ennyiszer megkérdezi a következőket:

- | | |
|--------------------------|---|
| ENTER FIELD NUMBER_ | - írd be a mező számát -
a rekordon belül hányadik mezőről
van szó |
| ENTER SEARCH CRITERION_ | - írd be a keresési feltételeket
- 0, ha egyenlő
- 1, ha kisebb
- 2, ha nagyobb
- 3, ha kisebb egyenlő
- 4, ha nagyobb egyenlő |
| ENTER SEARCH STRING_ | - írd be a keresett láncot |
| ARE YOU SURE Y/N | - ha N, újra kéri a keresett láncot |
| ENTER REPLACEMENT STRING | - írd be a behelyezendő láncot |
| UPDATE IN PROGRESS | - halad az aktualizálás |

6.2. ARITHMETIC UPDATE

- | | |
|------------------------------|---|
| ENTER NUMBER OF SEARCH KEYS_ | - írd be a keresési kulcsok
számát. Ha 0-át adunk, akkor
GLOBAL UPDATE, vagyis a file
összes rekordján elvégzi az
aktualizálást,
ha n-et, egy konkrét számot
adunk, akkor a működés
megegyezik a 6.1.-ben
leírtakkal. |
|------------------------------|---|

6.3.EXIT

- | | |
|--|----------------------|
| | - kilépés a főmenübe |
|--|----------------------|

A 7. Segítségnyújtás és 8. Kilépés funkcióval itt nem fogunk részletesebben foglalkozni, mert amit leírtunk az EASY-FILE áttekintésénél, az elégséges.

A segítség (HELP) leírása

CREATE DATA FILE HELP

- aritmetikai műveletek: + , - , , /
- mezőhossz max. 38 karakter
- a felső kettő és az utolsó sor foglalt a rendszer számára , üzenetek , hibajelzés stb.
- F1 a sor közére igazítása
- F3 egy üres sort helyez a cursor fölé
- F4 sor törlés
- F5 a sort balra ütközteti
- F6 a sort jobbra ütközteti
- F8 elmenti a képernyőt

SPECIÁLIS FUNKCIÓK

RETURN	a következő sor elejére áll
SHIFT RETURN	az adott sor elejére áll
DEL	egy karakter törlése
INST	egy karakterhely behelyezése, beszúrása
HOME	bal felső sarokba tér
CLR	törli a képernyőt
	mezőkezdet jele
	mezővége jel
	egy karakteres mező jele

DATA MANIPULATION HELP

F1 balra ütközteti a mezőt
F2 jobbra ütközteti a mezőt
F3 automatikus növelés - csak bevitelnél
F4 duplikálja - visszaállítja - a kezdő területet
F5 mező törlése
F6 másolja ill. visszaállítja az előző rekordot
F7 törli a rekordot
F8 tárolja a rekordot

SPECIAL KEY SUPPORT

RETURN	a következő mezőre lép
LEFT	egy karaktert balra lép
RIGHT	egy karaktert jobbra lép
SHIFT RETURN	a mező elejére lép

CONTROL KEY SUPPORT

CTRL-P	bemutatja az előző rekordot
CTRL-N	bemutatja a következő rekordot
CTRL-E	befejezi a bevitel-javítást

ÁLTALÁNOS TUDNIVALÓK

SEARCH OPTION I.

A/S/T/+/-

A Goto Amend	menj javítani
S Change Screen	cseréld a képernyőt
T Terminate	szüntesd meg, fejezd be
+ Step forward	lépj előre - egy rekordot
- Step Backward	lépj vissza - egy rekordot

SEARCH OPTION II.

A/C/S/T

C Continue Search folytatd a keresést

DELETE OPTION

C/D/S/T

C Continue Delete	folytasd a törlést
D Delete Record	töröld a rekordot

3.1.3 Személynyilvántartás EASY-FILE segítségével.

Példánkban egy vállalat főosztályának nyilvántartását mutatjuk be a nyilvántartó kartonok alapján. Az első fejezetben leírt feladat-meghatározás alapján kell kialakítani a rekord leírásokat, a képernyőtervet, az adatátlományokat, és a táblázatterveket.

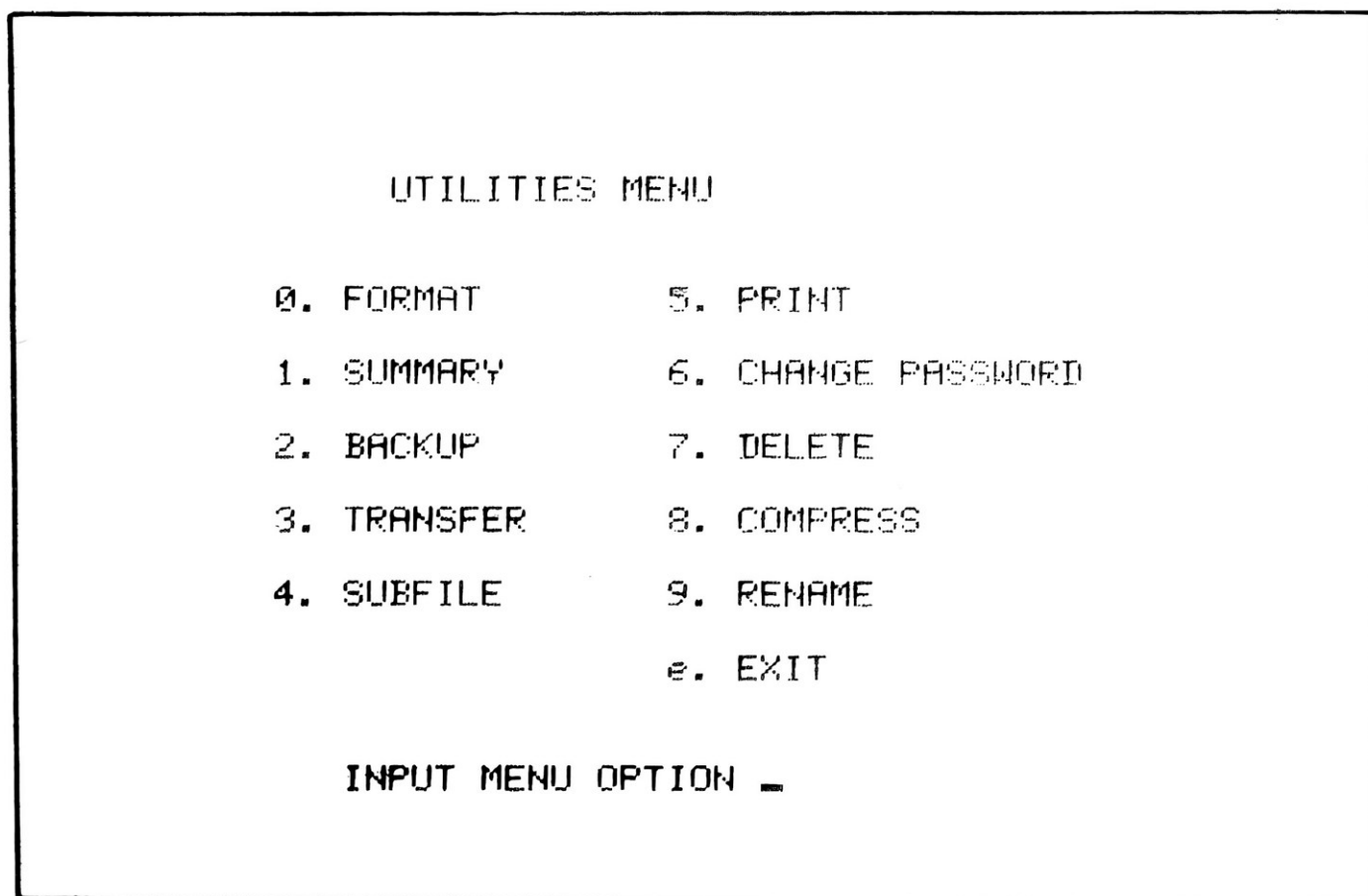
```
MAIN MENU OPTIONS

1. CREATE DATA FILE
2. DATA MANIPULATION
3. REPORT GENERATION
4. SORT DATA FILE
5. SUPPORT UTILITIES
6. FILE UPDATE
7. HELP
8. EXIT

INPUT MENU OPTION _
```

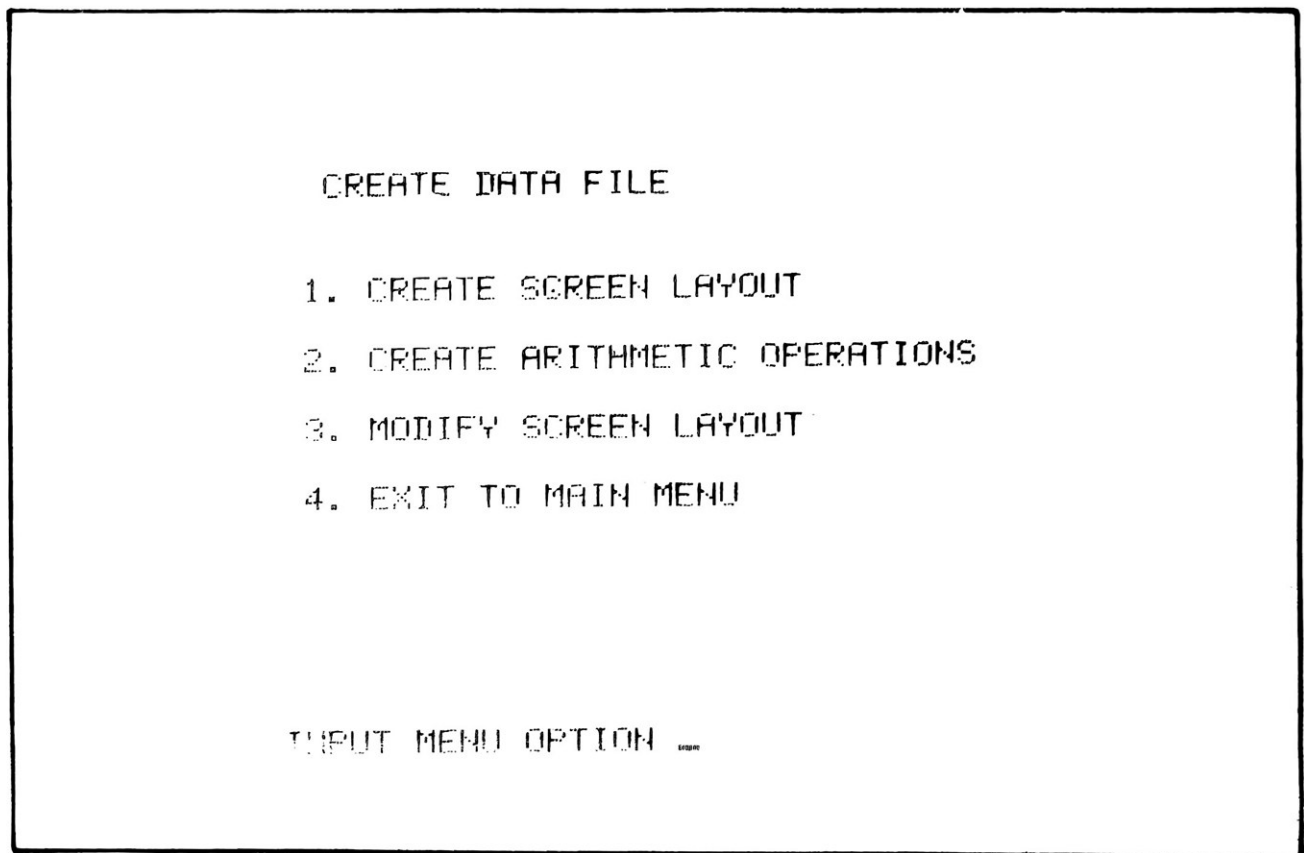
3.1.1 ábra

Mivel már elhatároztuk, hogy EASY-FILE-lal oldjuk meg a feladatot, ezért a terveknél tekintettel kell lenni a 3.1.2 fejezetben leírt korlátokra. Ezek, hogy az adatainkat két képernyőn helyezhetjük el. Egy mező 38 karakterből, egy rekord 80 mezőből és 506 karakterből állhat. Nagyon fontos, hogy a feladat megtervezésének minden fázisát papíron végezzük. Amennyiben papíron készen vagyunk a tervezéssel, elkezdhetjük betölteni az EASY-FILE programcsomagot (lásd 3.1.2 fejezetben). A betöltés után beje-



3.1.2 ábra

lentkezik az EASY-FILE főmenüje (lásd 3.1.1 ábrán). A munkánk elkezdéséhez először az adatlemezünket kell létrehozni. Ennek az elvégzéséhez a főmenüből a segédprogramokat kell kiválasztani, amellyel megformáljuk a lemezünket. Ezt az 5 -ös menü funkció segítségével érhetjük el. Az funk-



3.1.3 ábra

ció megjelölése után betöltődik a program, majd be kell tenni az üres lemezt (nagyon fontos!), majd leüttni a RETURN billentyűt és megjelenik a 3.1.2 ábrán látható menü. Itt a 0. funkciót választjuk ki. A program meg-

kérdezi az eszköz számát, a lemez címkéjét (label), és az azonosítási számát (identifier). Minden beírás lezárása RETURN-nel történik. Ha ezt mind beírtuk, akkor kb. 1 percig tart a formálás. A formálás befejezése után visszaadja a segédprogram menüjét. Innen az 'e' funkció segítségével

```

                                DESTING SCREEN
123456789.123456789.123456789.123456789.

                                Személyi lap1

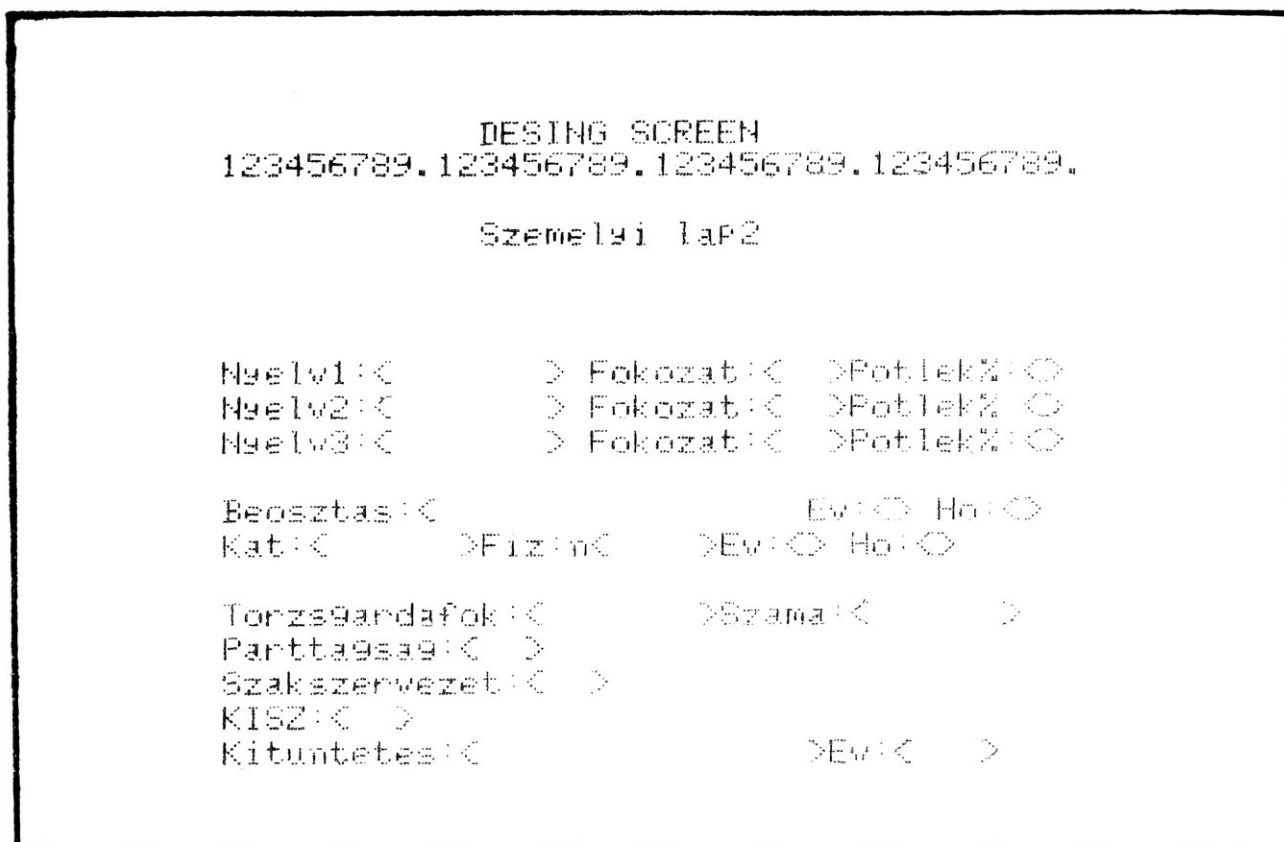
Törzsszám <          > Szm.sz: <          >
Név: <
Szül. i: <          > Családi név: <
Szül. h: <          > Gyerm. sz: <
KITÖH: <          > ALL. rész: < > Oszt. @ Fog. @
FEOP: <
Lakcím nr. sz: < > Város: <          >
Utca: <          > Sz: < > DE: < > OR: < >
Telefon <          > Bej. rész: < > Mod: <          >
Katonaság: <          > Szab: < >
Isk. végz: <          > Idő: < > Biz. sz: < > < >
Szakkep: <          > Idő: < > Biz. sz: < > < >
Tanfoly: <          > Idő: < > Biz. sz: < > < >

```

3.1.4 ábra

vel térhetünk vissza a főmenübe. Minden következő funkciót a főmenün keresztül tudunk elérni. A főmenü funkciói közül a tényleges feladat megoldásához az adatfile létrehozás funkciót kell kiválasztani, tehát az 1-es

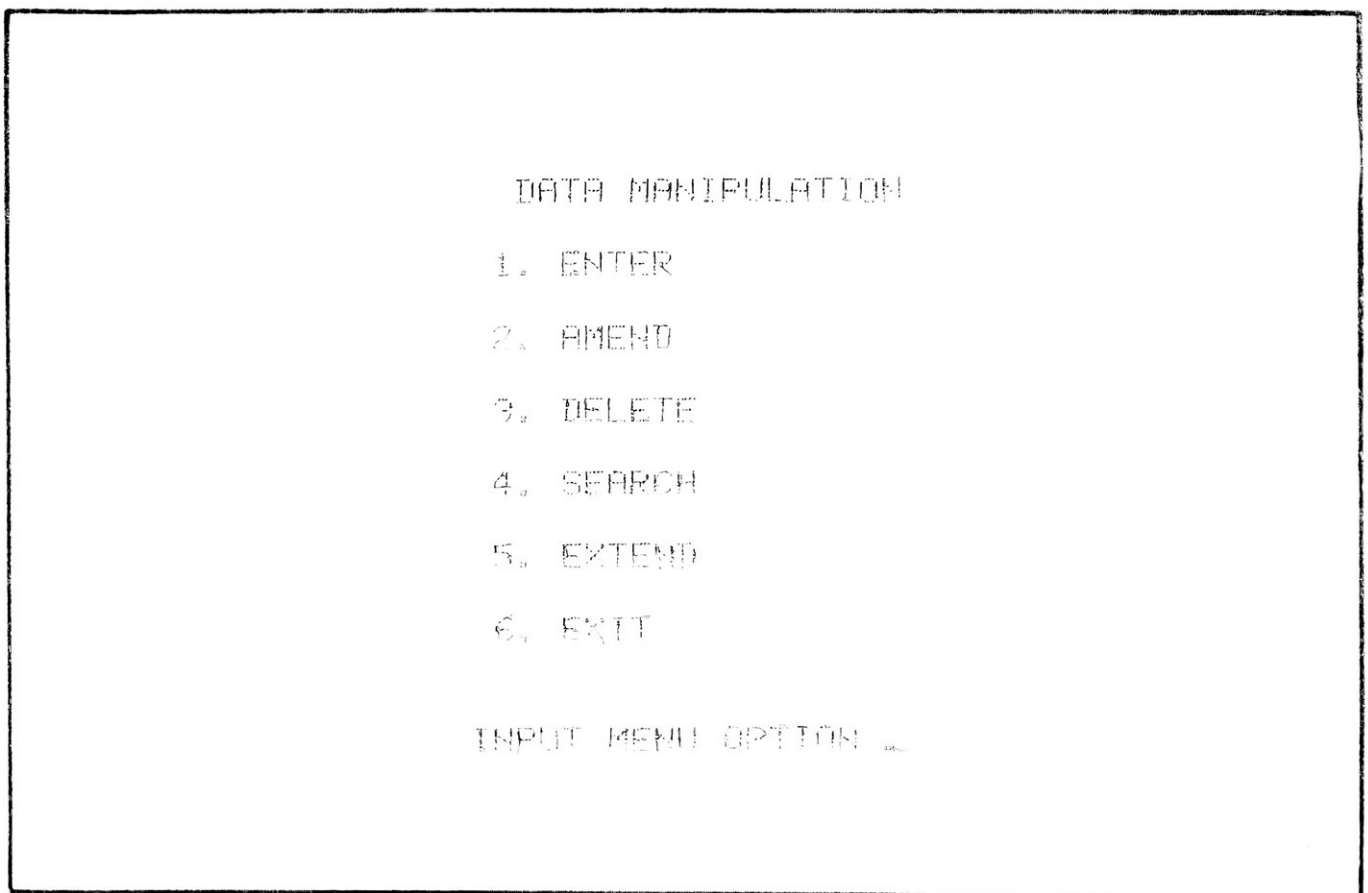
billentyűt kell leütni. A kiválasztás után be kell tenni a programlemezre. Ha betöltötte a programot, akkor megjelenik az ehhez a funkció-



3.1.5 ábra

hoz tartozó választék lista. Az általunk már meghatározott képernyő-tervezést kell kiválasztanunk. Az adatfile létrehozását, tehát az 1-es billentyűt kell leütni, (lásd 3.1.3 ábrát). Ezután betöltődik a program és megjelenik egy üres képernyő, amelyre beírhatjuk az általunk megtervezett képernyőtervet. Ennek az első és második képe a 3.1.4 és a 3.1.5 ábrákon

látható. Az első képernyőtervet a begépelés után az 'f8' funkció billentyű leütésével elmentjük a memóriába. Ezután adja az következő üres



3.1.6 ábra

képernyőt. A második kép begépelése után az elmentés jelzésére kijelzi a rekord méretét és a mezők számát. Ezután a jelszó szintjét kell meghatározni. Most a legkönnyebb esetet választjuk ki, hogy a példánkat ne bonyolítsuk el. Ezért a jelszó szintjénél a 4. számot jelöljük ki, ami azt jelenti, hogy nem kell jelszó. Ez azért sem jelenhet problémát, mert később

lehetőségünk van a módosításra.

FILE NEVE	szemely1	
RECORD NO	1	
Szemelyi lap1		
Torzsszam:	1112222	Szem.sz: 2350909
Nev:	Klampa Klara	
Szul.i:	350909	Csaladi al:n
Szul.h:	BudaPest	Gyerm.sz: 2
KTGH:	10499	All.csoP:f Orsz:is Foo:k
FEDR:	22334455	
Lakcim i.n.sz:	1033	Varos: BudaPest
Utca:	Kiss Jozsef	Sz: 12 E.l 9:12
Telefon:	BelePes i: 1986 Mod.ath.	
Katonasag:	Szah: 15	
Isk.veg:	alt.isk	Ido: 49 Biz.sz: 222/49
Szakkep:		Ido: Biz.sz:
Tanfoly:		Ido: Biz.sz:

3.1.7 ábra

Miután ezt is kijelöltük, meg kell adni a file nevét. Ha mindezt megtettük, akkor történik a képernyőtervének lemezre mentése a 'név.for' névvel megjelölve. Ezután megvizsgálja, hogy mennyi üres hely van a lemezen, és az általunk készített rekordból mennyi fér rá. Megkérdezi, hogy mi mennyit akarunk felvinni. Azért, hogy kipróbálhassuk az adatainkat, csak 3 rekordot veszünk fel. Az adatállományt 'név.dat' néven hozza létre a lemezünkön. Ez

után következik az adatbevitel. Ehhez a főmenüből a 2-es funkciót választjuk ki. Itt szintén programtöltés következik, majd megjelenik az adatszervezés menü listája (lásd 3.1.6 ábrát). Ebből a menülistából az 1-es pontot kell választani, mert ez jelenti az adatbevitelt. A választásunk után megkérdezi az adatfile-unk nevét.

Személyi2

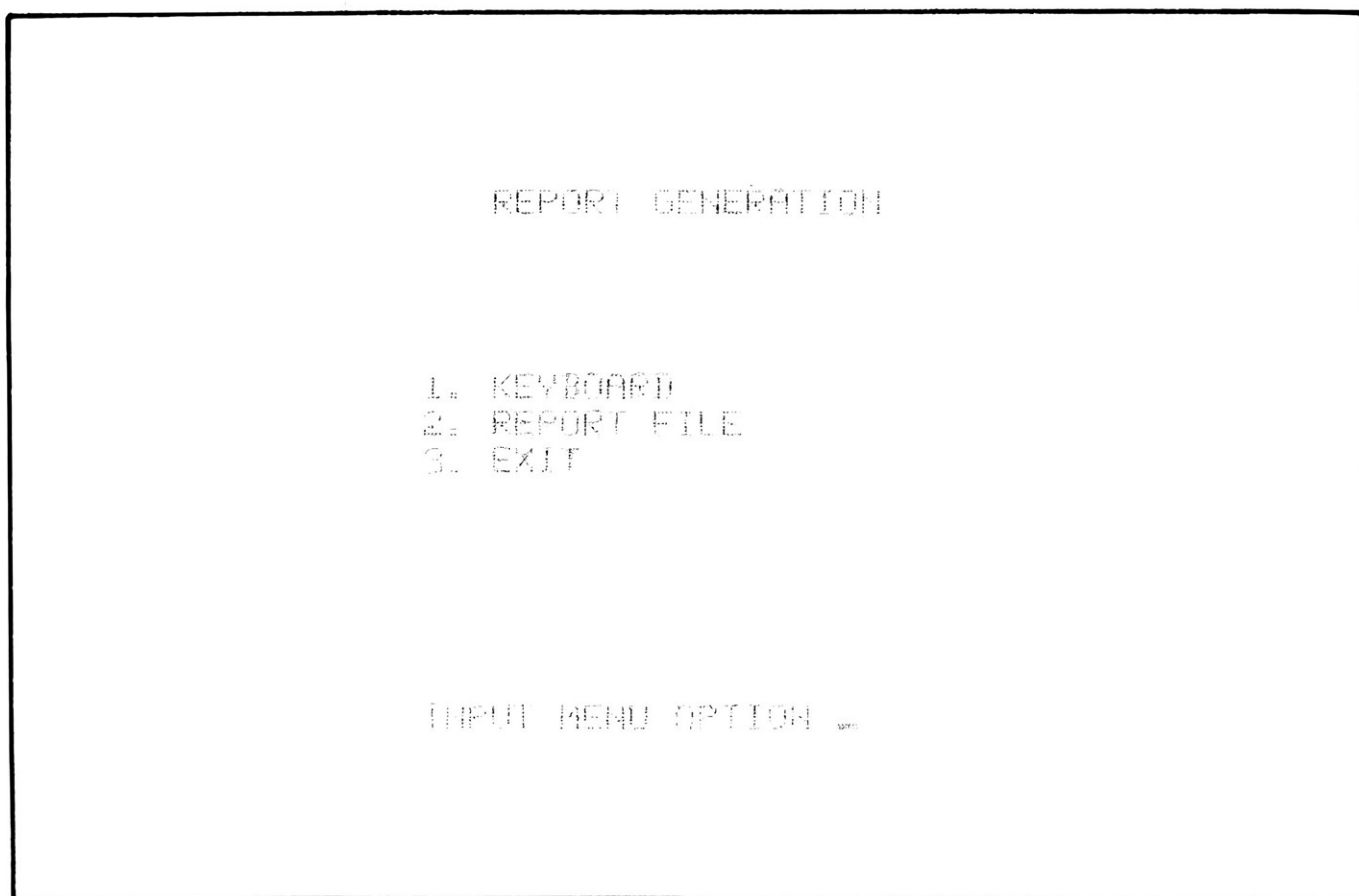
Nyelv1:	Fokozat:	Potlek%
Nyelv2:	Fokozat:	Potlek%
Nyelv3:	Fokozat:	Potlek%
Bepoztas: takerito		Ev: 80 Ha: 02
Kal: 41	Fiz: 2500	Ev: 80 Ha: 02
Tanysgand: fok:	Szám:	
Panba: 9518:		
Szakszervezet:		
KISZ:		
Kiuntetes:	Ev:	

3.1.8 ábra

Ha jó nevet adtunk, akkor betölti a formátum file-t. Ez a folyamat minden funkció kiválasztásánál megismétlődik ugyanilyen formában, ezért nem

ismételjük meg a leírását a többi esetben. Az adatbevitelnél megjelenik az általunk tervezett képernyő (lásd 3.1.7 és 3.1.8 ábrát) üres formában, és máris tölthetjük fel adatokkal az állományt. Nem kötelező minden mezőt kitölteni. Ha az első képernyő végére értünk, akkor a következő képernyőre automatikusan tér át. Amennyiben befejeztük a teljes képernyő kitöltését, akkor az 'f8' funkcióbillentyűvel kell elmenteni a rekordot az adatlemezünkre. Minden rekord elmentése után megkérdezi, hogy akarjuk-e folytatni. Mivel most 3 rekordnak foglaltunk helyet, ezért 3-at viszünk fel. Az adatrögzítés eredményét ellenőrzés céljából meg tudjuk nézni a segédprogram segítségével. A segédprogramnál az 5-ös funkciót (print), majd azon belül az 1-es (adatfile) funkciót választjuk. Eredményül a 3.1.9 és 3.1.10 ábrán látható formátumban kapjuk az adataink kinyomtatott képét. A rögzítésünk hibátlan volt, ezért most nem fogunk javítani, hanem a táblázat tervezéssel folytatjuk a munkát. Ezután a táblázatgeneráló funkciót választjuk ki. A program betöltése után a táblázatkészítés menüje jelenik meg a képernyőn (lásd 3.1.11 ábrát). Most az 1-es funkciót választjuk ki, hogy billentyűzetről gépelhessük be a paramétereket. Ezután van lehetőségünk válogatni a file-unokban. Először megkérdezi, hogy hány szempont alapján akarunk válogatni. Most 1 szempontot határoztunk meg. Ezután megjelenik a válogatást meghatározó képernyő. Itt az első helyen a mező sorszámát kell beírni, ekkor a program kiírja, hogy milyen típusú a mező. Mi a kettős mezőt választottuk, ami a 'személyi számot' jelenti. Ezután a feltételhez a 1-es számot írtuk be, ami azt jelenti, hogy kisebb a keresendő karaktersorozatnál. Ezt azért választottuk, mert a férfi dolgozókat szeretnénk leválogatni az állományból. Amelyik mezőn áll a pozíciómutató (cursor), a képernyő alján megjelenik a magyarázó szöveg, hogy mit kell csinálni. A 3.1.12 ábránkon idáig jutottunk el. A továbbiakban még be kell írni a keresendő karaktersorozatot (stringet). Ide '20000000000' szám sorozatot kell beírni, mert az ezalatt lévő személyi számok, biztos hogy mind férfiakat jelölnek. Ezután már következhet a tényleges táblázat paramétereinek pontos megadása, hogy melyik mezőt melyik oszlopba akarjuk írni. Miután meghatároztuk a táblázatunk nevét és beírtuk a nyomtatandó mezők számát, megjelenik a nyomtatási paraméter file (3.1.13 ábra). Az első paraméterrel a nyomtatandó mezőt határozzuk meg, ezután a

program kiírja a mező hosszát és típusát. Most beírhatjuk az oszlop megnevezését, majd a kezdő pozíciót.



3.1.9 ábra

Ebben a táblázatban csak két mezőt akarunk kiírni, még pedig a személyszámot és a nevet. Tehát először a mező tartalomhoz a 2-es számot írjuk, a megnevezésbe a 'SZEM.SZAM' kerül, a kezdő pozíciónak a 20-ast jelöltük. Másodszer a mezőtartalomba a 3-as számot írjuk, amelyik a nevet takarja, tehát az oszlop megnevezésbe a 'NEV' kerül. Azt, hogy hányadik

pozícióba kezdjük kiírni a nevet, úgy számolhatjuk ki , hogy az előző mező a 20-dik oszlopon kezdődik és 11

The image shows a terminal window with the following content:

```
SEARCH CRITERIA
I (E)U (M)  2
TYPE        4
CRITERIA    2

SEARCH STRING
_____

USER RESPONSE
ENTER CRITERIA
```

3.1.10 ábra

karakter hosszú. Azt is figyelembe kell venni, hogy ne közvetlen az előző mező után kezdődjön. Ezért a 40-dik pozícióba tettük a nevet. Ezzel tulajdonképpen a végére értünk a táblázat meghatározásának. Ezután elmentjük a paramétereket, mert később is akarjuk használni. Ha elmentettük, akkor már kérhetjük is a táblázat elkészítését.

A táblázatot a kevés adatunkkal kipróbálhatjuk, hogy megfelel-e a nyomtatási formája.

The image shows a terminal window with the following content:

```
PRINT PARAMETERS
AREA      2   CONTENTS  2   TYPE      A   LENGTH  11
TITLE
SZEM. SZAM
START COLA  ARITHMETIC  ACCH
USER MESSAGE
ENTER AREA TITLE
```

3.1.11 ábra

Amennyiben jónak találjuk, kibővíthetjük a file-unkat az adatszűrés bővítési (extend) funkciójával. A rekordjainkat 40-re bővítettük, ezután ugyanebben a menü funkcióban kiválasztjuk a felvitelt (enter) és kitöljük a file-t valóságos adatokkal. Ezután már elkészíthetjük a végleges táblázatot (lásd 3.1.1 melléklet).

Már tulajdonképpen befejeztük az EASY-FILE bemutatását egy példán keresztül, de azért még a segédprogramoknál lévő nyomtatási lehetőségeket bemutatjuk. Ehhez vissza kell térni a főmenühez.

```

FILE NEVE      : szemely11

                <Szemelyi lap>

Torzsszam:<    > Szem.sz:<    >
Nev:<
Szul.i:<    > Csaladi al:<
Szul.h:<    > Gyer.m.sz:<
KTGH:<    > All.csoP:<    > Oszl:@ Foo:@
FEOR:<
Lakcim ir.sz:< > Varos:<
Utca:<
Telefon:<    > Belep.es i:< > Mod:<
Katonasa9:<
Isk.ve9z:<    > Ido:< Biz.sz:<
Szakkep9z:<    > Ido:< Biz.sz:<
Tanfoly9z:<    > Ido:< Biz.sz:<

```

3.1.12 ábra

A főmenüből kiválasztjuk a segédprogramokat az 5-ös billentyűvel. Ekkor a már leírt módon kell eljárni. Ezután a 3.1.2 ábrán látható menülistából kiválasztjuk a nyomtatás (print) funkciót. Ehhez az 5-ös billentyűt kell leütöni. A következő file-okat lehet nyomtatni: Adatfile, Képernyőterv,

File-jellemzők. Megjeleníteni lehet nyomtatón és képernyőn. Miután kiválasztottuk, hogy a file-jellemzőket nyomtatjuk ki, megkérdezi milyen eszközre akarunk nyomtatni. Mi a nyomtatót választottuk. Ezután felhívja a figyelmet a nyomtató üzembehelyezésére, és indulhat a nyomtatás (lásd 3.1.14 ábra).

```

                                <Szemelyi2>

Nyelv1:<      > Fokozat:<      > Potlek?:<○
Nyelv2:<      > Fokozat:<      > Potlek?:<○
Nyelv3:<      > Fokozat:<      > Potlek?:<○

Beosztas:<      > Ev:<○ Ho:<○
Kat:<      > Fiz:<      > Ev:<○ Ho:<○

Torzs9anda fok:<      > Szama:<      >
Parttag9aa9:<      >
Szakszervezet:<      >
KISZ:<      >
Kituntetes:<      > Ev:<○
```

3.1.13 ábra

Itt befejeztük a munkát, tehát amikor visszakapjuk az almenüt, a kilépést kell választanunk. Ezzel visszatérünk a segédprogram menüjébe, majd itt az

'e' funkció segítségével visszatérhetünk a főmenübe. Innen a 8. funkció segítségével térhetünk vissza a Basic-hez.

```
FEJLEC INFORMACIÓK
FILE NEVE          : szema1911
REKORDOK BEÍRVA   : 3
REKORDOK TÖRÖLVE : 0
TELJES REKORDOK  : 3
REKORD MÉRLET    : 345
TÖRÖLT REKORD NÖ : 0
```

3.1.14 ábra

FERFI DOLGOZOK

SZEM. SZAM	NEV
13503150123	Simon Karoly
13910098909	Katona Istvan
14009124356	Szabo Istvan
14309248967	Antal Janos
14409044567	Fodor Janos
14509093456	Kolozsvari Emil
14512310098	Kovacs Laszlo
15010085576	Robert Imre
15204072376	Mihalei Tamas
15309189876	Kiss Rudolf
15407139241	Kolompár Andras
15411237890	Kiraly Andras
1550906	Kertes Robert
1561110016	Marosi Laszlo
15711210978	Nagy Sandor
15910110957	Kiss Imre
16011041109	Toth Lajos
17501200899	Konda Mihaly

3.2 A SUPERBASE

A SUPERBASE elsősorban menü-vezérelt adatkezelő rendszer, azonban rendelkezik egy igen hatékony alkalmazás generátorral is. Ez a programozni tudók érdeklődését is felkeltheti, hogy a Basic-hez hasonló, saját adatbázis-programozási nyelve is van. Nagyon sok feladat azonban anélkül is megoldható. Saját menü kialakításával még könnyebben kezelhetővé, és szinte automatikus lefutásúvá tehető sok művelet. A programozás során a programból is irányíthatók a menük. Ezek ilyen módon való használata természetesen csak a menük alapos megismerése után javasolt. A rendszer annyi lehetőséget tartalmaz, hogy a menü 1 és a menü 2 képernyőn található funkciókon belül is több helyen újabb menü található. A felhasználói adatok kezelését támogatja a rendszer többféle file-kezelő funkcióval, pl. file- és lemezmásolás, tartalomjegyzék bemutatás, formázás, stb. - anélkül, hogy ki kellene lépni a SUPERBASE-ből.

3.2.1 A SUPERBASE eszköz igénye

1. Commodore 64 személyi számítógép.
2. Commodore 1541 floppy egység, + csatlakozó.
3. Monitor, vagy televízió, + a megfelelő csatlakozó kábel, antenna.
4. Nyomtató egység a hozzátartozó interface-szel.
/CBM 1515,1526, MPS801,802,803./

5. Rendszerlemez,

3	"start.p"	seq
4	"henter"	seq
5	"hselect"	seq
6	"hfind"	seq
5	"houtput"	seq
5	"hcalc"	seq
5	"hreport"	seq
3	"hexecute"	seq
5	"hhelp"	seq
5	"hfile"	seq
7	"hformat"	seq
4	"hbatch"	seq
5	"hsort"	seq
3	"hprog"	seq
7	"hmaintain"	seq
3	"hmemo"	seq
8	"hmenu"	seq
2	"hcommands"	seq
3	"himport"	seq
3	"hexport"	seq
6	"hlabels"	seq
1	"cust.inv"	seq
2	"cust.rec"	seq
1	"TRAINING"	seq
2	"sb"	prg
10	"loader"	prg
7	"labels.p"	seq
1	"v1.0k"	seq

6. Adatlemezek.

3.2.2 A SUPERBASE részletes ismertetése.

A programcsomag elindítása a következő módon történik:

```
LOAD'SB',8,1 (RETURN)
```

A képernyő kb. 2 percig villog különböző színekkel, majd megjelenik a felszólítás:

```
Remove Program Disk
```

```
Insert Data Disk and Press RETURN
```

or

```
Press F1 to Create Data Disk
```

Cseréld ki a programlemezt!

Tedd be az adatlemezt és nyomd meg a RETURN-t!

vagy

Nyomd meg az F1-et az adatlemez létrehozásához!

Az adatlemez létrehozása után a SUPERBASE automatikusan leáll. Újra kell indítani a betöltést. Saját tapasztalatunk szerint az alapgép és a lemez ki- és újra bekapcsolása nélkül. Az újraindítás után, nem működik mindig tökéletesen. Mint láthatjuk a program első esetben elkészíti magának az adatlemezt, de ha már van, a RETURN megnyomásával tovább irányítható. A következő képernyőn megkérdezi az adatbázis nevét.

mode : Processing

```
-----  
SUPERBASE 64  
-----
```

```
-----  
Enter Database Name:  
-----
```

A megadott nevet megkeresi a lemezen. Ha nem találja , megkérdezi: 'Create it?', így lehet az első esetben, és később is új adatbázist létrehozni. Az adatbázis létrehozása , vagy az adatbázisba való bejutás után megjeleníti a képernyőn a már meglevő file-ok nevét és méretét, majd megkérdezi:

mode : Enter File Name

Database Catalog

Filename.. Size

file1..... 60

file10.... 10

Beírható az a file név , amivel dolgozni akarunk. Ha nem találja , ismét megkérdezi, készítsen-e , file-t, ha igen, a FORMAT opcióba lép, és elkezdhetjük felépíteni a képernyőformátumot, a rekordképet.

Ha sikerült bejutni az adatbázisba, és kiválasztottuk a file-t, megjelenik - menul felíráttal - a főmenü első része.

mode : menu1 SUPERBASE 64
V 1.0E (c) Precision Software,1983

File Selected = filenév

- F1 Enter új rekord bevitele a file-ba

- F2 Select Rekord kiválasztása a file-ból, vizsgálat,
ellenőrzés, vagy további feldolgozás céljából

- F3 Find Adott kritériumoknak megfelelő rekordok listába
gyűjtése későbbi felhasználáshoz

- F4 Output Kiíratás a teljes file-ból, vagy listából a
nyomtatóra, vagy a képernyőre

- F5 Calc Számítás végrehajtása a feldolgozás alatt
álló rekordban, vagy az ún. memória változókon

- F6 Report Programgenerátor, a megadott paraméterek
alapján kivonatot készít a teljes file-ból, vagy
egy kulcslista alapján

- F7 Execute A memóriába beírt, ill. behívott, felhasználó
által írt program végrehajtása.

- F8 Help Help - segítség, a funkciók nevének beírásával
hívható, kiírja a képernyőre az adott funkció
működtetéséhez szükséges alapvető tudnivalókat

A funkció billentyűk megnyomásával, vagy a funkció nevének beírásával

beléphetünk a megfelelő opcióba. A RETURN billentyű megnyomásával átléphetünk a menü 2-be, ami a következőképpen néz ki:

```
mode: menu2          SUPERBASE 64
V 1.0E  (c) Precision Software,1983
```

File Selected = filename

- F1 File File kiválasztása a file listából

- F2 Format A rekordformátum meghatározása, mezőtípus és mezőnevek definiálása

- F3 Batch A CALC eljárás végrehajtása a teljes file-on vagy listán, a módosítások automatikus visszairásával a file-ba

- F4 Sort Egész file, vagy válogatott lista rendezése további mezők szerint, a kapott lista más funkciókkal is feldolgozható

- F5 Prog Program írási-előállítási lehetőség a SUPERBASE saját nyelvén

- F6 Maintain Általános file kezelési funkciók

- F7 Menu Általunk kitalált névvel előhívható, valamely munkához szükséges információk, jegyzetek, HELP képernyők készítése

A funkciók bővebb leírása:

ENTER - beírás.

Új rekord bevitele a file-ba. Ebben a működésben megjelenik a kiválasztott file-ra jellemző képernyő. A kurzor a kitöltendő mező első pozícióján villog. A mezők lezárása RETURN-nel, a mezők közötti mozgás a kurzor fel és le gombokkal, a mezőn belüli 'DEL' billentyűvel történik. Az utolsó mező lezárása után kiírja:

Press RETURN To Store - nyomd meg a RETURN-t a tároláshoz

A rekord egyébként bármikor tárolható a SHIFT/RETURN gombok együttes megnyomásával. Kilépés tárolás nélkül 'fl q' gombok egymás utáni megnyomásával lehetséges. A kulcs- illetve kötelezőmező kitöltése nélkül a rekordot nem tárolja a program. Tárolás után:

Press Space To Another - nyomd meg a Space-t a (beírás) folytatásához.

Space válasz esetén folytatja a bevittelt egy másik üres képernyővel. A RETURN megnyomása esetén visszatér a főmenübe.

SELECT - kiválasztás.

Formája : SELECT (opció)

SELECT 'kulcs'

SELECT MATCH where [mező] is 'kritérium'

Ez a funkció saját menüvel is rendelkezik a további kiválasztás megkönnyítéséhez. Az ismerv alapján kiválasztott rekordokat mindig megjeleníti a képernyőn.

mode : Select Menu

F1 Key	Rekord keresése teljes- vagy részkulcs megadásával, a megtalált rekordot kijelzi a képernyőn
F2 Current	Az aktuális vagy az éppen feldolgozott rekord ismételt kiírása a képernyőre
F3 Next	Az egyébként kulcsmező szerint növekvő sorrendbe rendezett file, soron következő rekordja
F4 Last	A file utolsó rekordjának kiírása
F5 Previous	A file előző rekordjának kiírása
F6 First	A file első rekordjának kiírása
F7 Match	Az adott kritériumnak megfelelő rekordok egymás utáni kiírása
F8 Output	A kurrens rekord output formában való kiírása
A Add	Új rekord létesítése egy meglévő rekord tartalmának módosításával és/vagy a kulcsmező átírásával
R Replace	A kurrens rekord mezőinek módosítása a kulcsmező kivételével
D Delete	A kurrens rekord törlése, 'igazolás' szükséges

FIND (jelentése - találni)

A FIND parancs az adott kritériumoknak megfelelő rekordokat kiválogatja, és egy ún. 'listába' gyűjti össze, ez a kulcslista. Az így kapott lista-file kiíratható a későbbiekben SELECT, SORT, OUTPUT, BATCH, vagy REPORT parancsokkal, vagy a PROG funkcióban megírt programmal kezelhető.

Formája: FIND 'listanév' where
 [mező1] is 'kritérium';
 [mező2] is 'kritérium'....

Ha nem adunk meg nevet a kulcslistának, akkor a lista neve 'hlist' lesz.
 Ha a Where-t elhagyjuk, a kritériumokat az Enter-nél használatos képernyőn kell megadni valamennyi mezőhöz.

A Where paraméter alakja mindig
 mező is 'kritérium'

Több 'Where' paraméter esetén pontosvesszővel kell elválasztani a paramétereket.

A lehetséges összehasonlítások:

Pontos egyezés	=szám	=text
Folytonos összehasonlítás a mezőkben		=text
Ha nem egyenlő	#szám	#text
Folytonos összehasonlítás a mezőtől		text-
Folytonos összehasonlítás vége		*
Az adat a mintánál nagyobb	>szám	>text
Az adat a mintánál kisebb	<szám	<text
Az adatok összehasonlítása vagy kapcsolattal		
	1/2/3	a/b/c
Adatcsoportok összehasonlítása	>l&<6	>a&<e
Szövegrészek egyezése		=text*
Egyezés vizsgálat joker-rel		=Te?t?

Egyezés leírás az utasítássorban

OUTPUT - (jelentése - itt kimenő adat)

Az OUTPUT paranccsal a képernyőre vagy a nyomtatóra vihetünk ki adatokat, a teljes file-ból vagy listából. Ezzel a paranccsal a mezők tartalmát, szöveget, a Basic változók tartalmát, és/vagy számítások eredményeit írhatjuk ki, a kiviteli forma megadásával.

Formája: OUTPUT from 'lista' [mező] &[mező]
 @ x,y[mező] 'text'

A mezők az eredetileg definiált hosszal kerülnek kiírásra, ha nem használjuk az & parancsot.

Az & parancsot a kimenő adatok formátumozására a következőképpen használhatjuk:

'&' az utána álló mező végéről elhagyja a szóközöket

'&x' az utána álló szövegmező tartalmát x karakterre vágja le.

'&x,y' numerikus mezők esetén a számot x egész és y tizedes jeggyel írja ki

A '@' parancsot a kiírás pozicionálására használhatjuk. A képernyőn és a nyomtatón egyaránt működik

'@x' a követő mező kiírását az x-ik pozíción kezdi

'@x,y' a következő mezőt az x-ik oszlopon, az y-ik sorban kezdi kiírni

CALC - számítás

A CALC funkcióval különböző számításokat végezhetünk el, és az eredményt meg is jeleníthetjük. A képletekben valamennyi Basic függvény használható, a trigonometrikus függvények is. Az eredmény mezőkben, vagy memóriaváltozóknál tárolható, és ki is jelezhető.

Formája: CALC [mező] = képlet; [mező] = képlet
CALC a = képlet ; nb = képlet
CALC képlet

Több képlet elválasztására pontosvesszőt használjunk. Egyenlőségjel nélküli képlet eredményét a program az utána álló információkkal együtt kiírja, még akkor is, ha utána további pontosvesszők vannak.

Ezért a megjelenítésre szánt információkat mindig a CALC parancs végére kell tenni. Egy teljes adatrekord is bevihető ilyen módon úgy, hogy először CLEAR paranccsal töröljük az aktuális rekordot, majd a CALC formula segítségével a rekord minden mezőjét feltöltjük tartalommal. A rekord a STORE - tárold - paranccsal helyezhető el a file-ban.

REPORT - (jelentése - táblázatkészítés)

A REPORT utasítás egy programgenerátort indít el. A generált program a teljes file-ból vagy listából táblázatot (kivonatot) készít. A programgenerátor párbeszédéses üzemmódban paraméterezhető, e paraméterek alapján állítja össze a programot. A generált program a LOAD opcióval betölthető és a RUN vagy EXECUTE paranccsal elindítható.

A PROG menü-funkcióval a programlista javítható, módosítható, a SAVE paranccsal pedig tárolható. Az esetleges régebbi, ugyanilyen nevű programot felülírja minden kijelzés nélkül!

A párbeszéd hat lépésben zajlik le:

1. Idézőjelek között ('név') megadjuk annak a file-nak a nevét, amelyről táblázatot akarunk készíteni.
2. Idézőjelek között megadjuk a táblázat (kivonat) címét.

3. Definiáljuk az összegeket ($t_0 \dots t_9$) és részösszegeket. ($s_0 \dots s_9$) pl. $s_1 = s_1 + [\text{összeg}]$

4. Megadjuk azoknak a mezőknek a nevét, amelyek váltásakor ki kell írni a részösszegeket. Természetesen az ilyen módon figyelt mezők alapján előzőleg rendezni kell a file-t. Megadjuk a részösszeg kiírásánál megjelenítendő szöveget. A kiírás után a részösszeg mező kinullázódik.

5. Adjuk meg az 'ALL'-t, ha a reportot az egész file-ból, vagy a lista nevét, ha csak az azokból a meghatározott rekordokból akarunk táblázatot készíteni.

6. Adjuk meg a táblázat végére kiírandó szöveget és összeget.

EXECUTE - betöltés-végrehajtás

Az EXECUTE parancsot a LOAD paranccsal betöltött, vagy a PROG opcióban megírt program futtatására használhatjuk.

Formája: EXECUTE (sorszám)

EXECUTE 'név'

EXECUTE

Sorszám megadásával a tárban már benn lévő program futtatását a megadott sortól kezdi. Név megadása esetén betölti a kért programot és elindítja. Sorszám vagy név megadása nélkül, ha van a tárban program, azt futtatja, ha nincs program a memóriában, akkor kéri a futtatandó program nevét.

HELP - segítség

A HELP funkció feladata, hogy a SUPERBASE lényeges funkcióiról, azok működéséről rövid emlékeztető jellegű információt, áttekintést adjon.

Formája: HELP 'név'

A névvel adjuk meg azt a funkciót (működést), amelyről az információt kérjük.

A következő help képernyők állnak rendelkezésre:

ENTER	SELECT	FIND	OUTPUT
CALC	REPORT	EXECUTE	HELP
FILE	FORMAT	BATCH	SORT
PROG	MAINTAIN	MEMO	MENU
LABELS	PARANCSONK	IMPORT	EXPORT

A BACKUP help-je a MAINTAIN-nél található. A HELP szövegei az eredeti képernyőfomátumos minta szerint épülnek fel. A bal szélen a SUPERBASE kézikönyv oldalszámára hivatkozik. Középen a funkció nagybetűvel írt neve található. A jobb oldalán a funkció help képernyőjének oldalszáma áll.

mode : Help Screen
Press Return for more

17 HELP 1.page

A CTRL-P gombok együttes megnyomásával bármilyen képernyőtartalom - így a help képernyők is - kinyomtatható.

Saját help képernyő is létrehozható a MEMO opcióban. Működését lásd a MEMO-nál.

FILE - adathalmaz

A FILE parancs segítségével egy már létező file feldolgozásra kijelölhető. Ha nincs a megadott nevű file, új file létesíthető. Maximum 15 file hozható létre egy adatbázison belül.

Formája: FILE 'filenév'

A FILE parancs használatakor a program kéri a filenevet, ugyanakkor ki is írja a katalógust - az adatbázisban már létező file-okról. Ez a katalógus a CATALOG paranccsal bármikor lekérdezhető. A Maintain-ból is kérhetünk katalógust.

Új file létesítésekor a program automatikusan behívja a FORMAT opciót.

Létező file definíciókat úgy használhatunk fel, hogy átmásoljuk egy másik névre a létezőt, a FILE opcióval az újat kiválasztjuk, majd meghívjuk a FORMAT opciót, és elvégezzük rajta a módosítást. A lemezen levő file neveket a DIRECTORY paranccsal lehet kilistáztatni. Ez a parancs a Maintain menüben is rendelkezésre áll, más file-kezelő parancsokkal együtt, pl: copy, rename, validate, stb.

FORMAT - (jelentése - forma-formátum)

A FORMAT parancsban a létrehozandó file rekord-felépítését, és képernyő formátumát határozhatjuk meg. A kurzort a szokásos módon mozgathatjuk a képernyőn. A mezők nevét és szöveges információkat adhatunk meg ilyen módon. A mezőnév max. 12 karakter lehet. A mezőnév után definiálhatjuk a mező típusát az alábbi parancsokkal:

F1+k	Kulcsmező	(max. 30 karakter)
F1+t	Szövegmező	(max. 25 karakter)
F1+n	Számmező	(max. 9 egész és 4 tizedes tart)
F1+r	Eredménymező	(mint n-nél)
F1+d	Dátummező	(7 vagy 11 kar.)
F1+c	Konstansmező	(max. 30 karakter)
F1+shift+c	Naptármező	(7 vagy 11 kar.)
F1+shift	Kötelezőmező	(a mezőt kötelező kitölteni)

A mező típusának megadása után jobbra lépkedve a kurzorral a kívánt mezőhosszig, a RETURN megnyomásával lezárhatjuk a mezőt. A jobb felső sarokban folyamatosan kiíródik a mező hossza a jobbra lépkedés közben. A 'normál' mezőknél a RETURN, az ú.n. 'kötelező mezők'-nél a SHIFT/RETURN megnyomása szükséges a lezáráshoz.

A kötelező mezők kitöltése nélkül nem lehet eltárolni a rekordot.

Egy rekordkép meghatározásánál maximálisan 4 képernyőt lehet felhasználni.

A képernyőt az 'F1+stop/run' billentyűk egymásutáni lenyomásával tárolhatjuk a lemezen. További képernyőkezelő parancsok:

'F1 +'	Ad egy üres képernyőt, melyet ismét teleírhatunk mező definíciókkal.
'F1 -'	Visszaugorhatunk az előzőleg megírt képernyőre.
'F1 i'	A kurzor aktuális sorának invertálása.
'F1 s'	A teljes képernyő invertálása.
'F1 b'	A kurzorral rajzolható 'keret-karakter' megadása.
'F1 e'	A kurzor sorában levő szöveg törlése.

- 'F1 HOME' Ha a kurzor mezőjelen áll, mező törlődik, a szöveg nem a kurzort a képernyő bal felső sarkába állítja.
- 'F1 CLR' A teljes képernyőt törli.
- 'F1 p' A képernyő kinyomtatása.
- 'F1 DEL' Sor törlése.
- 'F1 INST' Sor beszúrása.
- 'F1 q' Kilép a 'FORMAT' funkcióból és nem fejezi be.

A CTRL 1/2/3 - a kurzor, a háttér és a keret színének állítása. Ezt a színbeállítást a rekordképpel együtt eltárolja a SUPERBASE, és a file későbbi felhasználásánál ezt állítja be!

BATCH - (jelentése - kötegetelt feldolgozás)

A BATCH paranccsal a teljes file-on, vagy egy kiválogatott lista rekordjain végezhetünk műveleteket. A műveleti utasításokat CALC formulával adjuk meg. Az eredményeket Basic változóban, mezőkben tárolhatjuk, vagy egyszerűen kiírhatjuk. Az adott műveleteket az adatállomány tárolt rekordjainak kiválogatott mezőin végezhetjük el. Ez az adatállományok karbantartását egyszerű, automatikus folyamattá teszi.

Formája: BATCH from 'lista' [mező] = képlet;
 a = a + [mező] ; képlet

A teljes file feldolgozása esetén a 'lista' helyett 'all'-t írjunk. Több képlet elválasztására használjuk a pontosvesszőt. Egyenlőségjel nélküli képlet eredményét a program az utána álló információkkal együtt kiírja akkor is, ha még további pontosvesszők is vannak. Ezért a nyomtatásra szánt információkat mindig a BATCH képlet végére kell írni.

SORT - (jelentés - rendezés)

A SORT parancs segítségével a file vagy egy lista rekordjait a kulcsmezőtől független mező tartalma szerint átrendezhetjük csökkenő vagy növekvő

sorrendbe. Fontos tudni, hogy a bevitelnél - ENTER- a rekordok a kulcsmező tartalma szerinti növekvő sorrendben kerülnek eltárolásra. A SORT - rendezés - eredménye a lemezre kerülő lista-file lesz, amelyben a rekordok sorrendje megfelel a SORT-ban meghatározott paramétereknek. A kapott listát a továbbiakban az OUTPUT, BATCH, PROG, REPORT, vagy SELECT opciókkal később is feldolgozhatjuk.

Formája: SORT from 'lista' D-on[mező1]
 [mező2] &5[mező3]...TO'újlista'

A 'D' paraméter a Descending - csökkenő - sorrendet határozza meg.

A növekvő sorrendhez 'I-on' megadás szükséges, ahol a'I' elhagyható. Ha a teljes file-t akarjuk rendezni a - from 'lista' - helyett az 'all'-t adjuk meg.

Ha nem adunk meg lista nevet (from ' '), akkor 'hlist' lesz az input listanév, ezt a file-t keresi meg a lemezen. A rendezett file neve 'hlist' lesz ha a TO kulcsszót elhagyjuk. A rendezési eljárás a megadott mezők első tíz karakterét veszi figyelembe a rendezésnél, ha csak az '&' használatával más hosszt nem adunk meg. Az '& [mező]' megadásnál az egész mezőhosszt figyelembe veszi. Az '&5[mező]' megadásnál a [mező] -ből csak az első 5 karaktert figyeli.

Fontos tudnivaló, hogy kevesebb paraméter, vagy rövidebb mezőhossz-vizsgálat gyorsabb rendezést eredményez.

PROG - program

A PROG parancs új programok előállítására és a már meglévők módosítására szolgál. Új programnál kapunk egy üres képernyőt, amelybe sorszámozott sorokban megírhatjuk a szükséges programot. Használhatunk sok Basic utasítást, mint pl. POKE, CHR , függvények, stb., de rendelkezik sok saját lehetőséggel is. Lehetséges programból is működtetni sok olyan funkciót amely egyébként menüből vezérelt. Nagyon jó példa erre a FILE 'filenév' programból való kiválasztása. A menü 1-ből vagy menü 2-ből kiadott - LOAD 'programnév' - utasítással betölthető a program a memóriába, ezután a

program módosítható. Ha nem módosítjuk, akkor futtatható a RUN , vagy az EXECUTE utasítással. A - SAVE 'programnév' - utasítással a létrehozott vagy módosított program kimenthető a lemezre. A menühöz F1/q-val vagy F1/stop-pal térhetünk vissza.

Formája : PROG (sorszám)

A képernyő törlődik, és a programból tíz sornyi megjelenik , attól a sortól kezdve amelyik sorszámot megadtuk.

A megengedett Basic parancsok:

clr	goto	rem
data	if..then	restore
dim	list	return
for..next	load	run
end	new	save
get	on	step
gosub	read	stop

Operátorok és kifejezések:

abs	int	right\$
asc	left	sin
atn	len	sqr
chr\$	log	str\$
cos	mid	tan
exp	'pi'	val
fre		

MAINTAIN - karbantartás

Ez a funkció is tartalmaz újabb menüt.

mode : Maintain Database

- F1 Status
- F2 Catalog
- F3 Import
- F4 Export
- F5 Directory
- F6 Backup
- F7 New Disk
- F8 Other

Ezek a menü lehetőségek elsősorban a SUPERBASE file-ok karbantartását szolgálják. Ezekon kívül a Basicben használatos file kezelő funkciók is megtalálhatók, pl. egy file törlése, stb.

STATUS - a rekordok meződefinícióit írja ki a képernyőre, vagy a nyomtatóra, a lista a mezők méret- és típusadatait tartalmazza.

CATALOG - az adatbázisban levő összes file nevét, valamint a rekordok számát tartalmazó listát írja ki az képernyőre

IMPORT - az IMPORT segítségével egy 'külső' formátumú file olvasható be a SUPERBASE-be.
A külső file más programmal vagy korábban a SUPERBASE-zel előállított file, vagy EXPORT funkcióval készített 'archív' file lehet

Formája: `IMPORT 'név' (,képlet)`

A képlet egy string kifejezés, amelyben a mezők és a rekordok elválasztásához használt határoló karaktert adunk meg.

Alapértelmezésben mindkettő `RETURN`, azaz a `SUPERBASE` az egyes mezők és a rekordok végén `RETURN`-t vár, a képletben ez `CHR (13)`.

Az adatrekord végét nem kell külön jelezni.

`EXPORT` - az `EXPORT` segítségével a `SUPERBASE`-beli file-ról 'külső' formátumú másolatot készíthetünk. Az így készült file más programokkal feldolgozható, vagy az eredeti 'archív' másolataként használható. Egy lemezegységes konfiguráció esetén az `EXPORT` és az `IMPORT` parancsot használjuk a `BACKUP` helyett.

Formája `EXPORT 'név' (,képlet)`
a képlet funkciója u.a. ,mint az `IMPORT`-nál

`DIRECTORY` - A lemezen levő összes file-ról a szokásos tartalomjegyzék listát adja

`NEW DISK` - Új lemez szabványos formázását teszi lehetővé

`OTHER` - DOS formátumú parancs kiadása a lemezegységre
`pl.c0:célfile=0:forrásfile` - file másolás
`r0:új név=0:régi név` - filenév átírása

AZ adatbázis ilyen módon való átmásolása problémákat okozhat. Ne kísérletezzünk ilyennel!

Nézzük meg az `EXPORT`, `IMPORT`, és `BACKUP` funkciókat!

`BACKUP` - (jelentése - másolás, másolat)

A Backup paranccsal az adatlemezünkkel azonos másolatot készíthetünk egy új hajlékony mágneslemezre.

Formája: MAINTAIN BACKUP

Ha a: 'kétmeghajtós lemezegység?' van kérdésre 'y'-t (igent) válaszolunk, a program üres lemezt kér az egyes meghajtóba. A válasz után megkezdődik a másolási eljárás.

Ha a konfigurációban csak egy meghajtó van, a program kéri, hogy a nullás egységben formáljunk egy lemezt. A program az összes felhasznált szektort beolvassa, és másolja. Egy menetben maximálisan 170 blokkot visz át.

EGYÉB PARANCSONK

A menükben megadott parancsokon kívül is vannak parancsok, ezek az u.n. DIRECT COMMAND-ok.

Példák: QUIT	kilépés a SUPERBASE-ből
DATABASE	új adatbázis keresése
FILE	új file keresése
DISPLAY	output készülék a képernyő
PRINT	output készülék a nyomtató
ACROSS	a mezők kiírási iránya - (egymásután, keresztben)
DOWN	a mezők kiírási iránya - (egymásalá, lefelé)

Általában minden F'n' (n=1-8) billentyűvel hívható funkció használható parancsként.

MEMO - emlékeztető

A MEMO parancs célja jegyzetek és saját HELP képernyők létrehozása, javítása.

Formája: MEMO 'név'

A HELP képernyő neve mindig a 'h' betűvel kezdődik.

Ha egy jegyzetet, pl. a kezelő részére készülő utasításokat kell help képernyőként kezelni, akkor a neve 'h.....' legyen.

Ha egy létező file-t javítunk, a program egymás után kijelzi a képernyőket, ekkor javíthatunk. A következő képernyő kiírása F1/STOP-pal kérhető.

Ha a javítás során bármikor leütjük az 'F1/q'-t, a kijelzett képernyő törlődik. Törlődik a lemezen a file még hátralevő része, a többi képernyő is.

A FORMAT-nál megismert szerkesztő parancsokat is használhatjuk.

3.2.3 Személyzeti adatok kezelése.

A következő részben egy rövid példában az előzőhöz hasonló feladatot mutatunk be.

A feladat tartalmazza az adatbázisba való bejutást, a file kijelölését, a rekordformátum létrehozását, a rekord kitöltését, keresést a rekordok között, a START program átírását, és egy rövid programot az adatkezelő rendszer saját nyelvén.

A 3.2.2 részben leírt módon töltjük be a SUPERBASE-t és a 3.2.1 ábrán látható képernyő formátumot láthatjuk.

Helyezzük el az adatlemezt, és üssük le a RETURN-t.

Az adatbázis nevét kell ezután megadni (lásd a 3.2.2 ábrán).

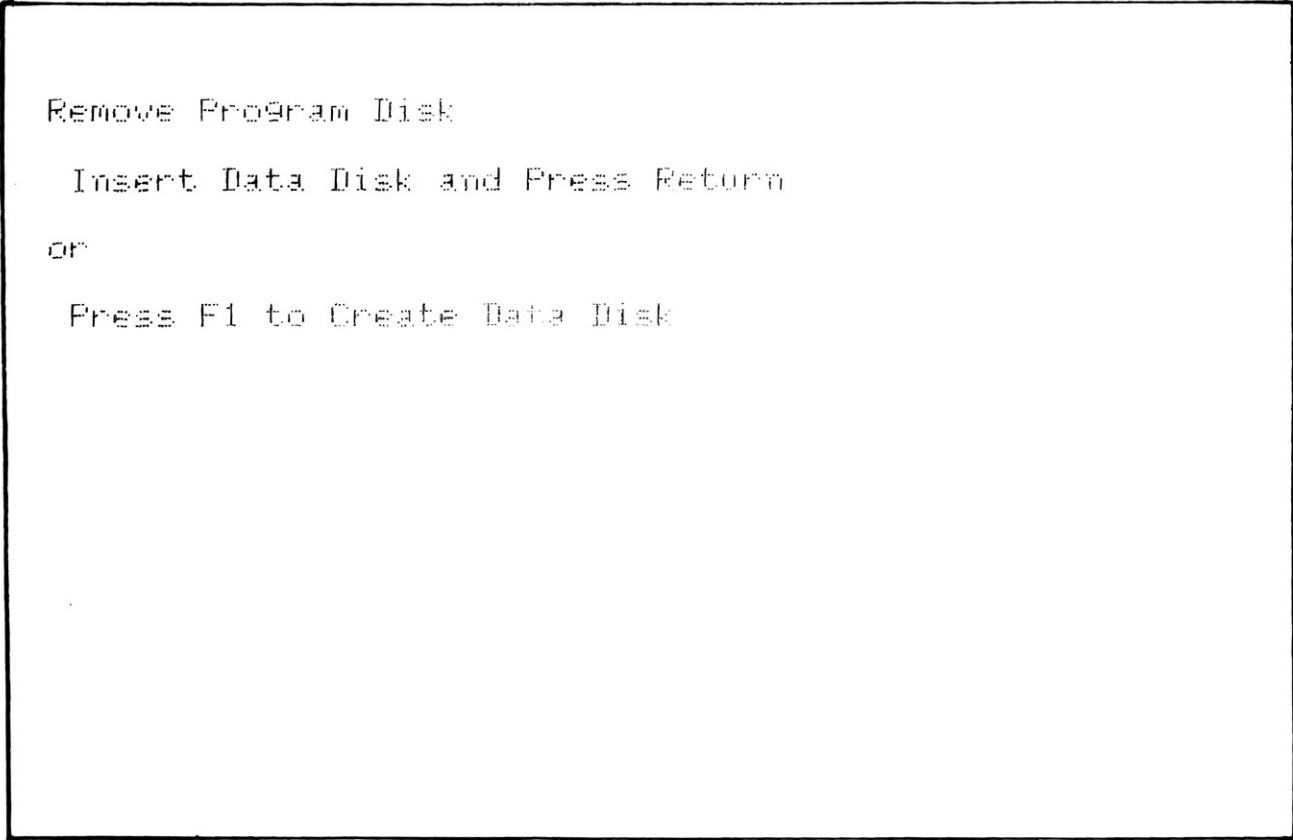
Ha beírtuk az adatbázis nevét, megkeresi és tovább kérdez:

Ird be a file nevét: 'név' (lásd 3.2.3 ábrát).

A file kiválasztása után megjelenik a " menu 1 " képernyő (lásd a 3.2.4 ábrát).

A menü 1 és menü 2 (lásd a 3.2.5 ábrát) között RETURN-nel lehet lépkedni.

```
mode : menu 1          Superbase 64  
V 1.0K    (c) Precision Software, 1983
```

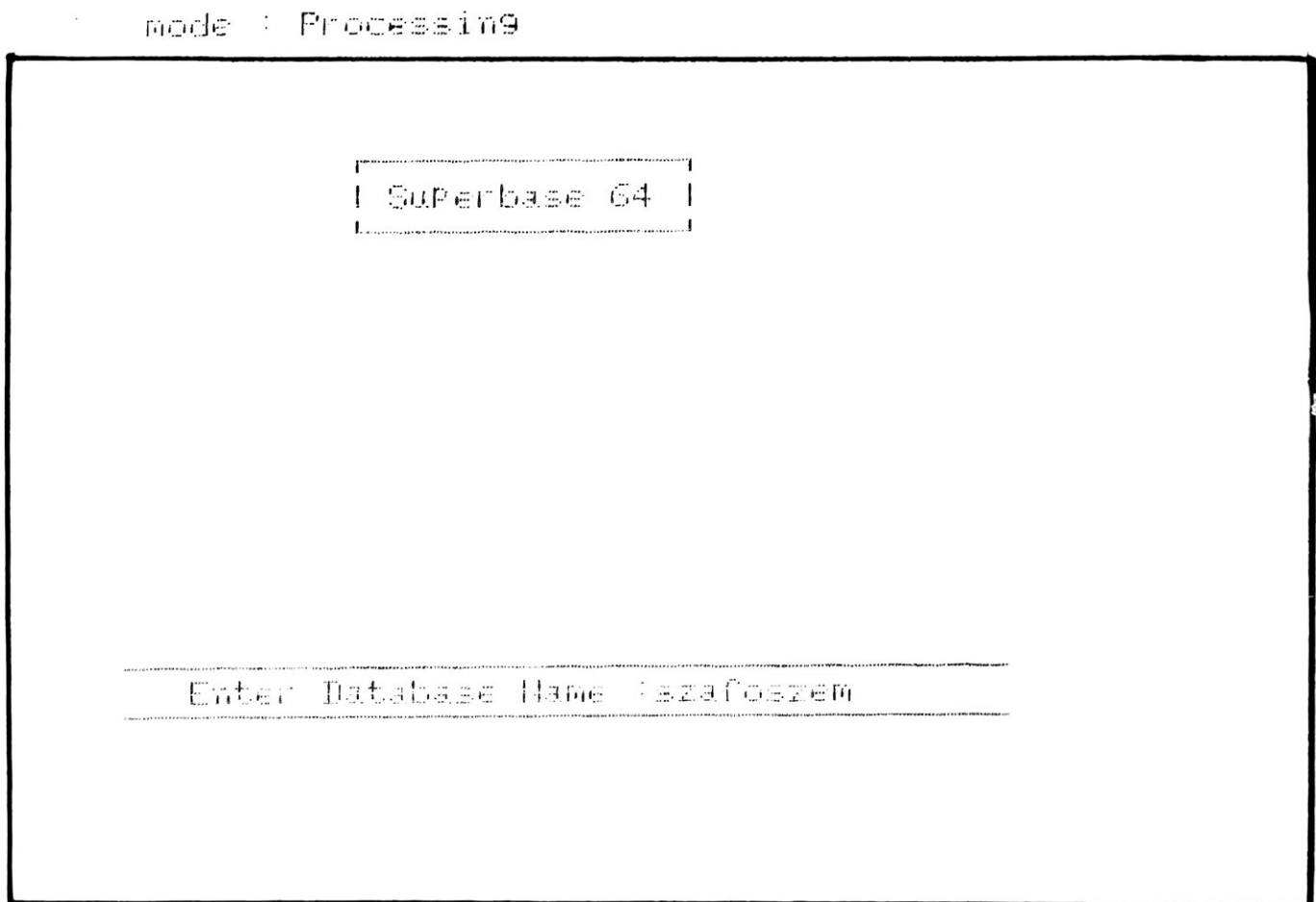


```
Remove Program Disk  
Insert Data Disk and Press Return  
or  
Press F1 to Create Data Disk
```

3.2.1 ábra

A menü 2-t kiválasztva nyomjuk meg az "F2"-t.

Az üres képernyőre most lehet begépelni a rekord képernyőtervét (lásd a 3.2.6 ábrát).



3.2.2 ábra

Ha a rekord hosszú, és nem elég egy képernyő, a következőn (lásd a 3.2.7 ábrát), összesen négy képernyőn folytatható a tervezés.

Ha a második képernyőt (lásd a 3.2.9 ábrát) is kitöltöttük, az * utolsó mező után felszólít :nyomd meg a RETURN-t a tároláshoz!

```
mode : menu 1          Superbase 64  
V 1.0K      (c) Precision Software, 1983
```

```
File Selected = szafoszem
```

```
f1 Enter
```

```
f2 Select
```

```
f3 Find
```

```
f4 Output
```

```
f5 Calc
```

```
f6 Report
```

```
f7 Execute
```

```
f8 Help
```

3.2.4 ábra

Előfordulhat, hogy egy már eltárolt adatot kell javítani.

```
mode 1 menu 2          Superbase 64  
V 1.0K      (c) Precision Software, 1983
```

```
File Selected = szafoszem
```

```
f1 File  
f2 Format  
f3 Batch  
f4 Sort  
f5 Prog  
f6 Maintain  
f7 Memo  
f8 Help
```

3.2.5 ábra

A Menü 1-ben F2-t megnyomva beléphetünk a Select funkcióba lásd a 3.2.10 ábrát).

```

mode : Format
-----
                SZAMOT91 14P 1
-----
tonzs.szam: "          ?
feors.szam: "          ?
szam.szam: "          ?

nev: "                                ?
szul.ido: "          ?
szul.nem: "          ?
csalad: "all: "          ?
lakcso: "          ?
in.sz: "          ?
utca: "          ?
szam: "          ?
telefon: "          ?

ktgh: "          ?
osztaly: "          ?
katonasag: "          ?

```

3.2.6 ábra

Ha például az F3-at nyomjuk meg, /Next/ akkor megkeresi az utolsó rekordot, és kiírja a képernyőre (lásd a 3.2.11 ábrát).

mode : Format

```

                személyi lap 2

isk.vegz:  █          @biz1sz:  █          @id1:  █          @
szakkepiz:  █          @biz2sz:  █          @id2:  █          @
tanfoly:  █          @biz2sz:  █          @id3:  █          @

nyelv1:  █          @fok1:  █          @pot11:  █          @
nyelv2:  █          @fok2:  █          @pot12:  █          @
nyelv3:  █          @fok3:  █          @pot13:  █          @

beoszt:  █          @ev1:  █          @ho1:  █          @
kat:  █          @fiz:  █+#####@ev2:  █          @ho2:  █          @
Premium:  █+#####@jutalom:  █+#####

torzs9arda-fokozat:  █          @
szama:  █          @

Parttag9sa9:  █          @
szakszervezet:  █          @
kisz-ta9sa9:  █          @

kituntetes:  █          @ev3:  █          @

```

3.2.7 ábra

3.2 fejezet

A "+" jel segítségével megnézhetjük a rekord második képernyőjét is.

mode : Entry

:# 1 t

szemelyi lap 1

torzsszam:<1118681>

feorszam: <17829961>

szem.szam:<15312300401>

nev:<Kiss Janos >

szul.ido:<1953.12.30>

szul.hely:<BudaPest >

csaladi all:<chazas >gyer.szama:<2>

lakcim

ir.sz:<1039> varos:<BudaPest >

utca:<vizior9ona >

szam:<14 > emelet:<1.> ajto:<12 >

telefon:<886-340>

ktgh:<10944> all.csof:<002 >

osztaly:<2> fooszt.:<1>

katonasag:< >

3.2.8 ábra

A program betöltése után bejelentkező képernyőt, (start.p) amely az adatbázis nevét kérdezi, át lehet írni a betöltése után.

Direct command-al: load"start" (return)

```
mode : Entry           :# 54 t
Press Return to Store
```

szemelyi lap 2

```
isk.vegz:<e99et.>bizisz:<      >id1:<78>
szakkep2:<menn. >biz2sz:<      >id2:<78>
tanfoly:<basic >biz2sz:<      >id3:<88>

nyelv1:<an9ol >fok1:<alapp >Pot11:<  >
nyelv2:<nemet >fok2:<felso >Pot12:<18>
nyelv3:<      >fok3:<      >Pot13:<  >

beoszt:<terv.menn      >ev1:<78>ho1:<88>
kat:<C-IV/3>fiz:< 5600>ev2:<85>ho2:<12>
Premium:<      >intalom:<      >

tonzs9anda-fokozat:<ezust>
                szama:<      >

    Parttag9a9:<1988>
szakszervezet:<1978>
kisz-ta9a9:<1968>

kituntetes:<2.sz kiv dol9. >ev9:<1984>
```

3.2.9 ábra

Ilyen képernyőt is lehet tervezni (lásd a 3.2.12 ábrát).

```
mode : Select Menu
```

```
f1 Key
f2 Current
f3 Next
f4 Last
f5 Previous
f6 First
f7 Match
f8 Output
A Add
R RePlace
D Delete
```

3.2.10 ábra

Egy példa az adatbázis kezelő saját programnyelvét nagyon szűken bemutatva lásd a 3.2.1 mellékletet.

```
mode Select n.l.P.f.k.c.m.p.o.a.m.d.++
```

```
szemelyi lap 1
```

```
torzsszam: <1118681>
feorszam: <17829961>
szem.szam: <26402136950>
```

```
nev: <Varadi Julianna >
szul.ido: <1964.02.13>
szul.hely: <Győr >
csaladi all: <Hajadon > evet.szama: < >
lakcim
ir.sz: <1039> varos: <Budapest >
utca: <Vizion90na >
szam: <3 > emelet: <2 > auto: <13 >
telefon: < >
```

```
ktgh: <10944> all.csop: < >
osztaly: < > fooszt.: < >
katonasa9: < >
```

3.2.11 ábra

Ez a program nem a menü segítségével választja ki a file-t.

```
mode Select n,l,P,f,k,c,m,o,a,r,d,+
```

szemelyi lap 2

```
isk.vegz:<C8 alt.>biz1sz:<          >id1:<C78>
szakkepz:<          >biz2sz:<          >id2:<    >
tanfoly:<Basic >biz2sz:<          >id3:<    >

neelv1:<          >fok1:<          >potl1:<    >
neelv2:<          >fok2:<          >potl2:<    >
neelv3:<          >fok3:<          >potl3:<    >

beoszt:<Adatno9zito    >ev1:<C83>ho1:<C09>
kat:<C31    >fiz:<    3200>ev2:<C85>ho2:<C02>
Premium:<          > jutalom:<          >

torzsaganda-fokozat:<          >
                szama:<          >

Parttagssag:<          >
szakszervezet:<C1984>
kisz-tagssag:<C1979>

kituntetes:<          > ev3:<          >
```

3.2.12 ábra

választja az első rekordot, és vizsgálja a fizetés mezőt; nagyobb-e mint 999. Ha nagyobb, megy és veszi a következőt.

```

10 nem kozPonti nyilvantartas
20 nem Programnev: start
50 bekon:90sub 400:nem a STOP billentyu engedelyezese
60 display @6,3"#####"
70 display @6,4"#####"
80 display @6,5"#####"
90 display @6,6"#####"
100 display @6,7"#####"
110 display @4,10" SZARU SZEMELYI NYILVANTARTAS"
160 display @0:nem display azanalogneset
170 ask &16@1,18"Az adathazis neve? :":a$
180 database a$,8:nem lemeze9999 P
190 file:nem az adathazis 999 file-lanekendyz
200 new:nem Programtoltes visszazelenes memobe
400 nem *** rendszer parameterek ***
410 lmar9 1:mar9 80:nem mar9ok beallitasa
420 Plen 72:tlen 60:nem oldal * sorveg hatar
430 Pdev 4:Pdef 0:nem szanf. es-koz 4 zha kod
440 lfeed 0:cont 1
450 space 0:across
460 screen 0
470 return

```

3.2.13 ábra


```
10 rem Pelda Program 5000 Ft-os fiz alatti nok nevsora
20 file "szafoszem"
25 fo=20
30 select firstrecord
40 fi=[fiz:]:sz$=[szem.szam:]
50 if fi>4999then 70
55 if sz$<"20000000000"then 70
60 gosub 150
70 select next record
80 eof 100
90 goto 40
100 end
150 rem kiiras
160 if fo>19then gosub 300
170 Print [nev:]@35[beoszt:]@57[fiz:]
180 Print [torzsszam:]@10[feorszam:]
185 Print " "
190 fo=fo+1
200 return
300 rem fejléciras
310 Print @20,1"NOK FIZETESLISTAJA 5000Ft ALATT"
320 Print @20"-----"
340 Print " "
350 Print "NEV"@35"BEOSZTAS"@62"FIZETES"
360 Print "TORZSSZ"@10"FEORSZ"
370 Print " "
380 fo=0
390 return
```

NOK FIZETELISTAJA 5000Ft ALATT

NEV TÖRZSÍV FELPISZ	BÉCSZTÁS	FIZETÉS
Gálidos Beáta 1107458 21099970	analit. elemzo	4500.00
Kiss Ilona 1118424	szs.karbantarto	4400.00
Szabo La'osne 1118684 21600170	tanonca kezelo	4050.00
Gyarmati Eva 1106001 20600160	muzezesz	2350.00
Varga Ilona 2118863 23319960	stervezo	4850.00
Jozsika Sarolta 2105368 22600160	fejlesztomernok	2650.00
Szabo Gaborne 2107649 29479990	muaszaki ussinte	4900.00
Pavel Istvanne 1107561 07960000	fejlesztomernok	4600.00
Kiss Judit 1118969 23657090	szam.tech.eload	4450.00
Kiss Gaborne 1118067	muzezesz	2950.00
Ferkas Zsuzsanna 1118435 06354660	operator	3600.00
Kiss Julianna 1118601 17829961	szamlazo	3200.00

3.3 A MULTIPLAN

A MULTIPLAN olyan adatkezelő programcsomag, amely táblázatos módon tárolja az adatokat. A hozzáférés egyszerű, táblázatonként hívható, de ha az adatok mennyisége megkívánja, több táblázat is összekapcsolható. Az így összefüggő adatmennyiséget nevezzük file-oknak. A file azonosítónévvel kerül a mágneslemezre, később a hozzáférés is ezen a néven történhet. A táblázatok összekapcsolása az adatok cseréjénél is szükséges lehet. Erre is van lehetőség, még hozzá ha egy táblázatban megváltoztatjuk az adatokat, akkor a tőle függő többi táblázatban is automatikusan megtörténik a módosítás.

Mivel a MULTIPLAN futtatása közben bármikor szükség lehet adatátvitelre, illetve az egyes utasítások végrehajtásakor a lemezről való olvasásra, ezért az alapgép - mágneslemez tároló kapcsolatának állandónak kell lennie. A számítógép egyszerre több mágneslemez tárolót is tud kezelni, tehát az adatfile-okat több lemezen is tárolhatjuk. Lényeges azonban, hogy tudjuk, hogy a keresett adathalmaz melyik lemezen, melyik meghajtóegységben található. Csak így tudunk a gép számára egyértelmű utasítást adni.

A táblázat egy sorokból-oszlopokból álló u.n. rendezésmátrix. Mérete változhat, maximum 255 sort és 63 oszlopot tartalmazhat.

A táblázat elemeit mezőnek nevezzük, minden mezőhöz tartozik egy képlet, melynek alapján meghatározható szintén a mezőhöz tartozó érték.

A MULTIPLAN programcsomag nagy előnye, hogy bármely mező tartalma függhet más mezők tartalmától. Ha egy mező tartalmát megváltoztatjuk, az összes tőle függő mező tartalma automatikusan változik.

3.3.1 A MULTIPLAN eszközigénye

A MULTIPLAN programcsomag használatához szükséges hardver környezet:

Commodore 64 számítógép

Commodore 1541 típusú mágneslemez meghajtó egység

Televízió, vagy monitor, megfelelő antenna csatlakozóval

Commodore sornyomtató/pl. 1515, 1525, 1526 stb. típusú/
és csatlakozó

Rendszer és adatlemezek

A MULTIPLAN rendszert tartalmazó mágneslemezen lévő file-ok nevei:

1	'MP'	PRG
26	'MP.DATA'	SEQ
97	'MP.SYS.'	REL
153	'MP.HELP'	REL
81	'MP.CODE'	SEQ
45	'MP.'	PRG

Mivel a számítógép több lemezegységet is képes kezelni, így a MULTIPLAN programcsomagot is úgy készítették el, hogy több mágneslemez egységet is tudjon egyidejűleg használni.

3.3.2 A MULTIPLAN részletes ismertetése

A rendszer indítása:

A gépek használati utasítás szerinti bekapcsolása után beütjük a következő parancsot:

```
LOAD 'MULTIPLAN',8,1
```


Ha megjelenik a READY kiírás és villog a kurzor, akkor a RUN paranccsal indíthatjuk a program futását. Ekkor újabb programrészek töltődnek a memóriába. A későbbiek során is folyamatos a mágneslemez meghajtóegység-alapgép kapcsolat, ezért ne kapcsoljuk ki a meghajtóegységet és a lemezt se vegyük ki belőle. Ha másik lemezre lenne szükség, a cserére a gép utasítást ad.

A MULTIPLAN betöltése után a képernyőn megjelenik egy üres táblázatnak egy része - 'egy ablak'. A képernyő felső részén az oszlop, bal oldalán a sorok számozása látható. A táblázat első elemén egy fehér téglalap látható ezt kurzornak nevezzük, és az aktív mezőt mutatja. A képernyő alsó részén angol nyelven öt szövegsor látható. Ebből a felső három az utasításkínálatot (a menüt) tartalmazza. Az első utasításon szintén látható egy téglalap, ez a menü kurzora, mely általában az aktív utasításon áll. A negyedik sorban az üzenetek láthatók, pl. hibajelzés vagy tájékoztatás. Az ötödik sor az u.n. 'állapot sor'. Itt látható a kurzor helye, az aktív mező aktuális tartalma, a még rendelkezésre álló tárkapacitás %-ban és a feldolgozás alatt álló táblázat neve, mely a rendszer indításakor TEMP. Az utasítások segítségével lehet a táblázaton módosításokat végezni, új táblázatot szerkeszteni, ill. a meglévő jó táblázatot lemezen tárolni, vagy a lemezen lévő táblázatot behívni a memóriába.

Az utasítás kínálatot a főmenü tartalmazza.

ALPHA-szövegbeírás	MOVE-mozgatás
BLANK-törlés (logikai)	NAME-név
COPY -másolás	OPT -kiegészítés
DEL -törlés (fizikai)	PRINT-nyomtatás
EDIT -érték	QUIT-kilépés
FORM -formátum	SORT-rendezés
GO -ugrás	TRANSFER-átvitel
HELP -segítség	VALUE-módosítás
INS -beszúrás	WINDOW-ablak
LOCK -védelem	XTERN-külső adatok

Utasítást úgy lehet kiválasztani a főmenüből, hogy vagy beütjük az utasítás kezdőbetűjét, vagy a menü kurzorát a SPACE (előre) ill. az INST/DEL (hátra) billentyű nyomogatásával a megfelelő utasításra állítjuk, és leütjük a RETURN-t. A kiválasztás után a program kijelzi az utasítás nevét és az esetleges almenüt, amelyből a választás szintén a fentiek szerint történik.

Ezután kérdések jelenhetnek meg, melyekre a MULTIPLAN választ is javasol. Ezeket helyben hagyni is, de módosítani is lehet.

Bármely megkezdett utasítás végrehajtását megszakíthatjuk a RUN/STOP gomb lenyomásával. Ilyenkor az utasítás aktivizálása előtti állapotot kapjuk vissza.

A fentieket egy konkrét példán mutatjuk be:

Az utasítás kínálat 18. eleme: WINDOW-ablak. Beütjük a W-t, ekkor képernyő alsó öt sora megváltozik.

```
WINDOW: SPLIT  BORDER  CLOSE  LINK
```

Mivel most csak az utasítások választásának módjával foglalkozunk, így a almenü jelentésére majd később visszatérünk.

Leütve az S gombot újabb kínálat jelentkezik:

```
WINDOW SPLIT:Horizontal,  Vertical,  Title
```

A *T* gomb leütése után már válaszra váró kérdések jelennek meg.

```
WINDOW SPLIT TITLE: # of rows:0
```

```
                # of coloumns:0
```

Mindkét sorban a zérusok ajánlatot jelentenek.

Az almenüben a kurzort a sorok között az *f1* billentyűvel mozgathatjuk. Ha a RETURN gombot nyomjuk le, akkor az utasítás végrehajtódik, s az így nyert táblát és az eredeti utasítás kínálatot kapjuk vissza.

A táblázat elemeinek betöltése, változtatása:

A MULTIPLAN indításakor az R1C1 jelű első elem helyén egy világos téglalap látható. Ez jelöli az aktív mezőt, ezt mezőmutatónak nevezzük. A fenti utasítások általában erre a mezőre vonatkoznak. Ha más mezőn akarunk utasítást végrehajtani, akkor a világos téglalap helyét kell megváltoztatni, vagy az almenüben kell más kurzor pozíciót megadni. Ez a kurzor mozgó billentyűk segítségével történik értelemszerűen. Ha a téglalap eléri az ablak szélét, akkor újabb léptetésekkor az egész ablak elmozdul a megfelelő irányba. Ha a táblázat szélét is elértük, a mozgás leáll, s ilyenkor figyelmeztető hangjelzés hallható.

Ha egyszerre egy irányban sokat akarunk lépni, akkor lehetőség van az u.n. lapozásra. Ilyen esetekben az aktív ablak egy képernyőnyit lép arrébb. Ez oly módon hajtható végre, hogy a CTRL és R gombok egyidejű lenyomása után használjuk a kurzort mozgó billentyűket, szintén értelemszerűen.

Ha az utasítás sorban a javasolt választ módosítani akarjuk, akkor a karakterek közötti közlekedést a funkció billentyűk segítségével tehetjük meg.

f1-tabulátor segítségével az almenü következő utasítására léptethető a menükurzor.

f2-leütésével a kurzor a beírt szövegen belül egy szóval balra lép. (A szóközt egy üres hely 'space' jelöli).

f3-törlés;(mint:CTRL/y, DEL)

f4-leütésével a beírt válaszon belül a kurzor egy karaktert balra lép.
f6-leütésével a kurzor egy karaktert jobbra lép
f8-a beírt szövegen egy szóval jobbra lépteti a kurzort.

A MULTIPLAN táblázat tartalmazhat sztringet, számokat, képletéket, stb.

Lehetséges értéktípusok:

1. Számok

2. Szöveg - mindig ' ' között adandó meg, az üzenet sorban az ALPHA típusjelzéssel.

3. Pozíciós adatok - megadhatjuk egy mező tartalmi függését több más mezőtől.

4. Logikai értékek - < , > , =, ill. ezek kombinációja.

5. Hibaértékek - az adott mező képletének számítása során létrejövő hiba jelzése a program által.

Nézzük meg kicsit részletesebben az értékek megadási módját, és a végezhető műveleteket:

1. Számok: lehetnek egész, tizedes és exponenciális írásmódban adottak:
1926; 1926,5; 1,926E+3

A képletek segítségével számított értékek 14 jegyre pontosak, az ábrázolható számok maximális kitevője -63 és +63 között lehet.

Lehetséges műveletek: hatványozás

* szorzás

/ osztás

% százalék számítás

+ összeadás

- kivonás

Az utóbbi két műveletijel előjelként is használható.

Műveleti sorrend: negatív értékek meghatározása

%

* , /

+ , -

logikai operátorok értelmezése

A számok közötti kapcsolatot függvények segítségével is megadhatjuk. A függvényeket mindig egy képlet részeként alkalmazzuk.

Ismerkedjünk meg azokkal a függvényekkel, amelyeket a MULTIPLAN rendszer ismer.

A függvények argumentumában az 'N' mindig numerikus értéket, az 'A' szöveg^o karakterek összességét jelenti.

ABS(N) - az argumentum abszolút értékét adja.

AND(lista) - ez a logikai függvény igaz értéket csak abban az esetben szolgáltat, ha a lista minden eleme igaz értékű, különben hamis értéket kapunk. Az argumentum csak logikai értékeket tartalmazhat.

ATAN(N) - a tangens függvény inverze, az argumentumhoz tartozó szöget adja, ívmértékben.

AVERAGE(lista) - az argumentumban szereplő numerikus értékek átlagát számolja ki.

COLUMN() - a függvényt tartalmazó oszlop számát adja. Ezt a függvényt nemcsak képletben, hanem önállóan is alkalmazhatjuk.

COS(N) - az ívmértékkel adott szöghöz tartozó cosinus értéket adja.

COUNT(Lista) - az argumentumban lévő numerikus értékek darabszámát adja. Ha

a lista tartalmaz nem numerikus értékeket, úgy azokat nem veszi figyelembe.

DOLLAR(N;S) - az N számot az S paraméternek megfelelő értékű tizedes jeggyel írja ki. Ha az S nem szerepel az argumentumban, úgy ezt a MULTIPLAN automatikusan 2-nek értékeli.

EXP(N) - a természetes alapú logaritmus inverze.

FIXED(N;számjegyek) - az N értéket szöveges információvá alakítja.

INDEX(tartomány, helyzet) - a kijelölt tartomány adott mezőjének értékét szolgáltatja. Ha a helyzet megadásakor kilépünk a tartományból, akkor a függvény értéke (nem létezik) hibaérték lesz.

INT(N) - az N-hez legközelebbi, nálánál nem nagyobb egész számot adja, u.n. egészrész függvény.

LN(N) - az argumentum természetes alapú logaritmusát számolja ki. Az N csak pozitív szám lehet, különben a függvény értéke hibaérték lesz.

LOG10(N) - az argumentum tízes alapú logaritmusát adja. Az N-re az előbbiekben leírt megkötések érvényesek.

LOOKUP(N;tartomány) - a kijelölt tartomány első sorában, vagy oszlopában keresi az N értéket. A jó működés feltétele, hogy a tartományban a vizsgálandó számértékek növekvő sorrendben legyenek.

MAX(lista) - az argumentum elemei közül kiválasztja a legnagyobbat, és az lesz a függvény értéke.

MIN(lista) - a lista elemeiből a legkisebbet teszi a függvény értékévé.

MOD(N;M) - a függvény az N/M osztás maradékát szolgáltatja.

NPV(kamat, lista) - a függvény az u.n. alapértéket szolgáltatja. Az alapérték az az összeg, amelyet az adott kamatláb mellett el' kell helyeznünk ahhoz, hogy a periódus végén az adott hozam álljon rendelkezésre.

PI() - a függvény értéke 3, 141 592 653 589 konstans.

ROUND(N, számjegyek) - a függvény az N értékét a számjegyekkel megadott tizedesjegyekre kerekíti.

SIGN(N) - u.n. előjel függvény. Értéke az N előjelétől függően:

ha az N pozitív, akkor 1

ha az N negatív, akkor -1

ha az N nulla, akkor 0

SIN(N) - az ívmértékkel adott szöghöz tartozó sinus értéket adja.

SQRT(N) - az N négyzetgyökét számítja ki. Az N csak nem negatív szám lehet, különben hibaértéket kapunk.

STDEV(lista) - a függvény az argumentum értékeinek szórását, standard eltérését szolgáltatja.

SUM(lista) - a lista elemeinek összegét számolja ki.

TAN(N) - az ívmértékben adott N tangense a függvény értéke.

2. Szöveg: maximálisan 255 karaktert tartalmazó karakterlánc. A karakterek között szerepelhetnek betűk, számok, írásjelek. (Kivételt képez az idézőjel, mivel ez a karakterlánc elejét ill. végét jelzi.)

Lehetséges művelet: & két karakterlánc összekapcsolása

pl: "140"&" km" = "140 km"

Alkalmazható függvények:

LEN(T) - a szövegváltozó karaktereinek számát adja.

MID(T,N,M) - a függvény azt a szövegrészletet szolgáltatja, amely a T változó N-edik karakterénél kezdődik, és M karakter hosszú.

FIXED(N,számjegyek) - az N értéket szöveges információvá alakítja.

3. Pozíciós adatok: a táblázat mezőinek helyzetét határozzák meg. A sor és oszlop számából áll, ezek metszete adja a kiválasztott mezőt. A sor jele R, az oszlopé C. Mivel a jel is hozzátartozik az adathoz, így a sorrend tetszőleges. A program a kijelzéskor mindig a sor, oszlop sorrendet használja.

Háromféle pozíciós adat létezik: abszolút

relatív

név

Abszolút pozíciós adatok meghatározhatnak:

egy sort	(képzése: R10	jelentése: 10. sor)
egy oszlopot,	C20	20. oszlop
sor tartományt	R10:20	10-20 sorok
oszlop tartományt	C20:30	20-30 oszlopok

Ha a meghatározó képletben R is, C is szerepel, akkor ezek metszete adja a kérdéses négyzög tartományt a határokkal együtt.

egy mezőt	R5C10	5. sor 10. eleme
téglalap tartományt	R5:10C10:12	

Relatív pozíciós adatok:

Egy alapmezőből kiindulva határozzák meg a célmező koordinátáit. Alapmező az a mező, ahol a kurzor az utasítás kiadásakor áll. A relatív pozíciós adat az alapmezőtől való távolságot adja meg. Az adatnak a sor, ill. oszlop jelén és számán kívül előjelet is kell tartalmaznia. A '+' jel felfelé ill. jobbra, a '-' jel balra ill. lefelé irányuló elmozdítást jelent.

R -az alapmezővel azonos sor
 C -az alapmezővel azonos oszlop
 R[+n] -az alapmező felett n sorral
 R[-n] -az alapmező alatt n sorral
 C[+n] -az alapmezőtől jobbra n oszloppal
 C[-n] -az alapmezőtől balra n oszloppal
 RIC -az alapmező oszlopában egy sorral lejjebb

Név-azonosító szó, mely vonatkozhat egy mezőre, ill. résztáblázatra.

Képzése: betűvel kell kezdődnie

tartalmazhat betűt, számot, pontot

maximális hossza 31 karakter lehet

Használata ugyanaz, mintha abszolút pozíciós adatot adtunk volna meg.

A pozíciós adatok közötti lehetséges műveletek:

metszet operátor

tartomány operátor

kapcsoló operátor

Metszet operátor: két pozíciós adatot úgy kapcsol össze, hogy az új adat az

eredetiek közös mezőire mutat.

Alkalmazása: R C2

'NÉV' R5C8

Ha egy mező megadásakor a pozíciós adatok fajtáit keverjük, azt csak a metszet operátor segítségével tehetjük, azaz köztük betűközt kell hagyni. Ha a metszet operátor alkalmazásakor nem talál közös adatot, akkor a program 'NULL' hibajelzéssel leáll.

Tartomány operátor: két pozíciós adat úgy határoz meg egy tartományt, hogy az a legkisebb olyan téglalapot adja, mely tartalmazza a két adatot. Általában a két adat a téglalap jobb alsó ill. a bal felső sarkát jelöli. A különböző pozíciós adatok tetszőlegesen keverhetők ennek az operátornak az alkalmazása során.

Kapcsoló operátor: két, nem összefüggő pozíciós adat összekapcsolására alkalmas. Operandusaként szerepelhet tartomány, ill. metszet is, bármelyik pozíciós adat megadásával.

Alkalmazása: R5C2;R7C2

4. Logikai értékek: két tetszőleges szám összehasonlítása során igaz ill. hamis értéke adódhat.

Az összehasonlítási műveletek:

< kisebb

> nagyobb

= egyenlő

<= kisebb vagy egyenlő

> = nagyobb vagy egyenlő

<> nem egyenlő

Alkalmazhatók még az: AND - és

OR - vagy

NO - nem logikai függvények

Szöveg típusú értékek összehasonlítása ezeknek az operátoroknak a segítségével lehetetlen.

5. Hibaértékek: Ezeket a program szolgáltatja olyan esetekben, mikor valamilyen függvényt vagy operátort, ill. pozíciós adatot nem megfelelően alkalmaztunk. A hibák továbbterjednek, azaz egy mező hibás volta magával hozza az összes tőle függő mező hibás voltát. Ilyen esetekben mindig vissza kell fejteni a hibákat a kiindulási alapmezőre.

Hibaértékek:

#DIV/0! - osztás nullával.

#NAME? - nem definiált név használata.

#NV! - olyan értéket használtunk, amely még nincs meghatározva.

#NUM! - a metszet operátor nem talált közös tartományt.

#POS! - a pozíciós adat nem létező tartományra mutat.

#TEXT! - felcseréltük a számok ill. szöveg alkalmazását.

Most nézzük kicsit részletesebben, hogy milyen műveletek végezhetők a táblázatokon, milyen utasításokat tartalmaz a főmenü.

ALPHA - utasítás segítségével az aktív mező szöveggel tölthető fel. Ha már van az aktív mezőben értékes karakter, akkor javasolt válasznak ezt írja ki a gép, de ez felülírható. A mezőben lévő szöveget az állapotsorban is kijelzi a program, idézőjelek közé ékelve.

Ha kurzorbillentyűvel fejezzük be az aktív mező kitöltését, akkor megváltozik a begépelte karakterlánc, felkerül a táblázat aktív mezőjébe, a táblázat mutatója (kurzor) a megadott iránynak megfelelően a következő mezőre lép. Ez lesz az új aktív mező. Az utasítás végrehajtása tovább folytatódik. Így egymás után tetszőlegesen sok mező kitölthető. Az kurzorbillentyű használatára megjelenik az ALPHA/VALUE kiírás. Az első beütött karakter típusától függően a gép maga választja ki, hogy milyen mezőt képeztünk, szöveg ill. értékmezőt.

Az utasítás befejezéséhez a RETURN gombot kell leütni.

BLANK - utasítás a megadott mező tartalmát törli. Ha nem az aktív mezőt akarjuk törölni, amely a javasolt válasz, akkor meg kell adni a kívánt mező pozícióját, vagy azonosító nevét. Az utasítás végrehajtása a nevet nem módosítja, a mezőnek csak a tartalmát törli, magát a mezőt nem, ezért nevezzük logikai törlésnek.

COPY - utasítás egy mezőről formátum és tartalom másolatot készít, az eredeti mező változatlanul hagyásával.

Az utasítás kiválasztása után a következő almenüt kapjuk:

COPY: RIGHT DOWN FROM

COPY: RIGHT - egy mezőt vagy egy oszlopot jobbra másol

COPY: DOWN - egy mezőt, vagy sort lefelé másol

Mindkét utasításnál meg kell adni a másolatok számát és kezdőpozícióját. Ha nem egy mezőt, hanem egész sort adunk meg kezdőpozíciónak, akkor több mező tartalma egyszerre változik meg.

COPY:FROM - a legáltalánosabb mód a másolásra. Itt meg kell adni a forrás mezőt és a célmezőt, (bármely pozícionálási mód segítségével) így bármely mező tartalma átvihető bárhová. Ezzel az utasítással másolhatunk

Ilyen esetekben, ha célmezőként csak egy mezőt adunk meg, úgy ezt a rendszer a tartomány bal felső sarkának tekinti.

DEL - ez a törlés utasítás, amely oszlopok, sorok, ill. egyes mezők a táblázatból való kitörlésére szolgálnak. Ilyen esetekben a többi mező pozíciós adata megváltozik, mely módosítást a MULTIPLAN automatikusan elvégez (fizikai törlés).

Ha az aktív táblázatnak más táblázatokkal van kapcsolata, úgy ezekkel a hivatkozásokkal a megváltozott pozíciók szerint egyeztetni kell.

EDIT - utasítás segítségével a mezőket értékkel vagy képlettel tölthetjük fel. Ezt az utasítást nemcsak a szokásos módon érhetjük el, hanem egy szám, illetve valamilyen matematikai műveleti jel leütésével is. Az utasítás befejezésére, illetve az iránybillentyűk használatára igazak az ALPHA utasításnál leírtak.

FORM - utasítás segítségével az egyes mezők megjelenési formáját állíthatjuk be. Mivel ez is egy bonyolultabb utasítás, ezért részletesebben megvizsgáljuk az almenüt, és egyes pontjait.

FORM: CELLS, DEFAULT, OPTIONS, WIDTH

FORM:CELLS - a mezők tartalmának kijelzését határozhatjuk meg. A mező adatait ütköztethetjük jobbra, balra, de beállíthatjuk úgy is, hogy a mező közepét foglalják el. Ezzel az utasítással állítjuk be az adat formátumát is, azaz azt, hogy százalékos, vagy pénzügyi formátumot kérünk-e, vagy egyszerű számkijelzést.

A MULTIPLAN indításakor az összes mező szabványos formátumú, s ha új sorokat, oszlopokat iktatunk be, azok is ilyenek lesznek.

Az utasítás hívásakor javasolt válaszként az aktív mező adatai, formátum kódjai és beállítási paramétereit láthatók. Az aktív mező formátuma az egész tartományra kiterjeszhető.

Kijelzési kód változtatása:

D - szabványos paraméterek

C - központosítás

G - szöveg balra, számok jobbra ütköztetése

L - balra igazítás

R - jobbra igazítás

- - a kötőjel a korábban beállított paramétereket hagyja érvényben

Formátum kód beállítása:

Def - szabványos formátum

Cont - a hosszú karakterlánc kijelzésére alkalmas, egymás melletti üres mezőket képes használni

Exp - számok normál alakban való kijelzése

Fix - fixpontos ábrázolás, megadott tizedes jegyre kerekítés

Gen - a számot a mező teljes szélességének megfelelő tizedes jeggyel írja ki

Int - egész szám kijelzésére

\$ - a szám adatok után kiírja a '\$' jelet

* - diagramok készítésére alkalmas formátum

% - a mező adatát százalék értéként kezeli

- - a kötőjel itt is meghagyja a korábban beállított értéket

FORMAT: DEFAULT - CELLS, WIDTH

Midkét utasítás a szabványos formátum beállítását mondja, az első a mezőkre, a második az oszlopok szélességére vonatkozik.

FORMAT: OPTIONS - Az utasítás lehetőséget ad az u.n. 'ezrespont' használatára, amikor is a MULTIPLAN a számokat hármass csoportokba rendezve ábrázolja. Az utasítás lehetőséget ad arra is, hogy a képlettel adott mezőtartalom kiszámítását ellenőrizzük. Ez azt jelenti, hogy az érték

helyett a képlet jelenik meg.

FORMAT: WIDTH - egy vagy több mező szélességét állítja be.

Ha a szöveg hosszabb, mint a megadott szélesség, akkor a felesleges jobboldali karaktereket a program automatikusan levágja. Ha a mező tartalma szám, és az nem fér el a megadott hosszon, akkor a mezőben egy '!' felkiáltójel jelenik meg.

GO - ugrás utasítással a táblázat egy adott mezejére ill. tartományára vihető a kurzor. A kívánt pozíció megadható azonosító névvel, ill. sor-oszlop kombinációval. Új ablak aktiválására is alkalmas az ugrás utasítás. Ekkor aktív mezővé automatikusan az új ablak bal felső eleme válik.

HELP - segítség utasítás részletes leírást tartalmaz az egyes utasítások alkalmazásáról, ismerteti a javasolt válaszok módosításának lehetőségeit, a billentyűzet használatát, a funkciós billentyűket.

Az utasítás a program futtatás elején, de menet közben is bármikor hívható. Természetesen, mint maga a program is, a tájékoztató szöveg is angol nyelvű. Az utasítás végrehajtásakor a megfelelő szöveg részeket a mágneslemezzről olvassa a program.

Ha a főmenüből hívjuk az utasítást, úgy általános információkat ad a MULTIPLAN programról. Ha valamelyik utasítás végrehajtása közben nyomjuk le a '?' billentyűt, akkor az ehhez az utasításhoz tartozó információk jelennek meg a képernyőn. Kivételt az ALPHA utasítás képez, ilyenkor ugyanis a '?' jelet szövegkarakterként értékeli az aktív mező részére.

Ha az üzenet sorban hibajelzés olvasható, akkor erről is részletesebb tájékoztatást kaphatunk ennek az utasításnak az alkalmazásával.

INS - beszúrás utasítással meglévő sorok illetve oszlopok közé újabbak iktathatók be. A MULTIPLAN a beszúrás végrehajtásakor a pozíciós adatokat automatikusan módosítja. Az utasítás végrehajtásakor a táblázat struktúrája megváltozhat, erre figyelni kell.

Ha az utolsó sorban illetve oszlopban van értékes adat, és beszúrást alkalmazunk, akkor az adatok eltűnnének a táblázatból. Ilyen hibák ellen véd a program, az utasítást nem hajtja végre.

LOCK - utasítás a megadott mezőt, illetve tartományt a véletlenszerű törlés ellen védi. Két változata van:

1. a megadott mezőt védi minden hozzáféréstől.
2. valamennyi szöveget, vagy képletet tartalmazó mezőt védi.

A védett mezőkben az ALPHA; BLANK; COPY; EDIT; VALUE; XTERN utasítások hatástalanok.

MOVE - utasítás segítségével a mezők tartalmát mozgathatjuk a táblázaton belül. A mozgatás oly módon történik, hogy a mozgatandó sor, illetve oszlop a célsor, illetve oszlop előtti sorszámú helyre kerül. A két sor (oszlop) közötti sorok (oszlopok) automatikusan elmozdulnak úgy, hogy feltöltik az így üresen maradt sorokat (oszlopokat), és a célsor (oszlop) előtt üres helyet biztosítanak az ide helyezendő sornak (oszlopnak).

A pozíciós adatok megfelelően megváltoznak. A mozgás bármely irányba történhet. Figyelni kell arra, hogyha a tartomány határait alkotó sort, illetve oszlopot mozgatjuk, ne veszítsünk el értékes adatokat.

NAME - utasítással egy mezőhöz , illetve tartományhoz azonosító név rendelhető. A későbbiek során ezzel a névvel hivatkozhatunk a kívánt mezőre.

A név 31 karakterből állhat, az első karakter csak betű lehet. Betűn kívül tartalmazhat a karakterlánc számokat, és írásjelek közül a pontot és az aláhúzást. A MULTIPLAN a meg nem engedett karaktereket figyelmen kívül

hagyja.

A név módosítható, törölhető.

OPT - utasítás segítségével az aktív mező tartalmát módosíthatjuk, akár szöveg, akár képlet tartalommal. Az utasítás aktivizálásakor javasolt válaszként az eredeti mezőtartalmat kapjuk, a kurzor a válasz jobbszélén áll. Mozgatása a már ismertetett módon a funkció billentyűk segítségével történik.

Az érték bevitel a megszokott módon történik, a szöveg karaktereket idézőjelek között kell megadni.

Ha az utasítást több mezőn is alkalmazni akarjuk, akkor a lezárást a kurzorbillentyűk segítségével végezzük. A RETURN leütése az utasítás befejezését jelenti.

Képletek megadásánál a rendszer az utasítás végrehajtása előtt ellenőrzi annak szintaktikus helyességét. Ha hibát talál, akkor megjelöli a hibás részt, nem végzi el a módosítást és az utasítás aktív marad.

PRINT - az aktív táblázat kinyomtatására szolgáló utasítás. Négy féle szolgáltatással rendelkezik.

PRINT: PRINTER, FILE, MARGINS, OPTIONS

PRINT: OPTION - Nyomtatás opciók meghatározása.

A nyomtatás megkezdése előtt választhatunk a következő funkciók közül:

- résztáblázat nyomtatása
- értékek helyett a meghatározó képletek nyomtatása
- sor és oszlopszámok nyomtatása
- vezérlőképek kivitele

PRINT: MARGINS - Mi választhatjuk meg a nyomtatási lap formátumát. Azaz beállíthatjuk a bal oldali és felső margót; a nyomtatási szélességet; a lap hosszúságát.

PRINT: PRINTER - A fentiek beállítása után kerülhet sor a tényleges nyomtatásra. Ha a rendszerbe kapcsolt nyomtatóra akarjuk kivinni a táblázatot, akkor az almenüből a PRINTER utasítást kell kiválasztani. A nyomtatás a megadott formai követelmények alapján történik. Módosulás akkor jön létre, ha több oszlopot akarunk kiírni egy sorba, mint az fizikailag elfér. Ilyen esetekben a MULTIPLAN automatikusan úgy tördeli a nyomtatást, hogy a végrehajtás után a lapok szétvágásával teljes táblázat legyen összeállítható. A táblázat jobb oldalán illetve alsó részén levő üres oszlopok nem kerülnek kinyomtatásra.

PRINT: FILE - A nyomtatást végezhetjük nemcsak nyomtatóra, hanem lemezfile-t is készíthetünk. Ez később behívható a gép memóriájába. Ez módosítható táblázat rögzítési lehetőség.

QUIT - utasítással lépünk ki a MULTIPLAN program futásából. Ilyen esetben a rendszer nem tárolja az aktív táblázatot, a monitoron a COMMODORE 64 eredeti képernyője látható.

Mivel az utasítás véletlen hívása sok munkánkat teheti tönkre, ezért a gép a végrehajtás előtt visszakérdezi, hogy a kilépési szándékunk valóságos-e, és csak igenlő válasz esetén törli a memóriát.

A MULTIPLAN program futásába visszatérési lehetőség nincs.

SORT - utasítással az egy oszlopban, illetve annak egy részében lévő adatokat lehet növelni, csökkenteni, illetve sorba rendezni. Javasolt válaszként mindig a teljes aktív oszlop növekvő rendezését kapjuk, de ez felülírható.

A program a mezők karaktereinek függvényében négy különböző csoportot létesít:

- 1, számok
- 2, szöveg
- 3, logikai, illetve hibaértékek
- 4, üres mezők

A szöveges mezőket a MULTIPLAN az alábbiak szerint rendezi:

! " \$ % & ' () * + , - . / 0-9 : ; < = ' > ? @ A-Z ' a-z

Ha azonos típusba tartozó adatok között van egyező, akkor ezek sorrendjét az utasítás nem változtatja.

A MULTIPLAN előnyei közé tartozik az is, hogy a rendezést több kulcs szerint is végezhetjük. Először a leggyengébb kulcs szerint, s utána a növekvő erősség irányába rendez.

TRANSFER - utasítás hat féle átviteli funkciót takar.

TRANSFER: LOAD, SAVE, CLEAR, DELETE, OPTIONS, RENAME

TRANSFER: LOAD - Betöltés a mágneslemezeiről. A memóriába tölti a névvel megadott file-t. A betöltés következményeként az eddigi aktív táblázat törlődik a memóriából, és a betöltött táblázat lesz az aktív. A MULTIPLAN előnyös tulajdonsága az is, hogy más rendszerrel dolgozó, a Commodore 64 géppel kompatibilis formátumban előállított file-ből is képes adatokat kiolvasni. Ilyen esetekben a töltés előtti táblázat aktív marad, az adatok ebbe a táblázatba kerülnek.

TRANSFER: SAVE - Tárolás lemezen. Az aktív táblázatot lemezen tárolja file formájában. A MULTIPLAN ennél az utasításnál is megvéd a véletlenszerű törléstől, mert ha a lemezen már szerepel egy file a megadott névvel, akkor kérdés formájában megbizonyosodik, hogy tudatos-e a felülírás.

A file-ek normál vagy szimbólikus formában (lásd TRANSFER OPTION) tárolhatók.

TRANSFER: CLEAR - aktív táblázat törlése a képernyőről és a memóriából. A program itt is véd a véletlenszerű törléstől. Az utasítás hatása hasonló a MULTIPLAN újraindításával. A különbség az, hogy néhány beállított opció, pl. a formátum, nyomtatás, átvitel érvényben marad.

TRANSFER: DELETE - File törlés, mely a lemezeiről törli a feleslegessé vált

adatfilet.

TRANSFER: OPTION - különlegesen beállított opciók az adatátvitelnél is lehetségesek.

Adatformátum meghatározása:

normál-bináris MULTIPLAN formátum - Ez biztosítja a legrövidebb átviteli időt.

szimbólikus formátum - teszi lehetővé más programokkal az adatszert.

idegen formátum - más programok által írt file-okból is képes adatot olvasni. Ebben a formátumban az adattárolás lemezen nem lehetséges.

Itt adhatjuk meg azt is, hogy melyik mágneslemez meghajtót kívánjuk használni. A rendszer az indításakor a 8-as lemezegységet működteti. Így ebbe kell a rendszer-lemezt helyezni.

Ha a 'sheet' almenü sorban a 8-tól különböző, például 9 számot adunk meg, akkor a TRANSFER utasítás a lemezre írás illetve lemezzől a lemezzől olvasás funkcióknál ezt a lemezegységet fogja használni. A MULTIPLAN rendszerhez tartozó információkat továbbra is a 8-as egységben lévő lemezzől olvassa.

TRANSFER: RENAME - az aktív táblázatot a megadott névvel tárolja, és egyidejűleg a forrás táblázatokban is módosítja a nevet.

VALUE - utasítással az aktív mezőket lehet módosítani. Használatára, továbbléptetésére a ALPHA utasításnál leírtak érvényesek.

WINDOW - utasítás alkalmazásakor a táblázatot oszthatjuk több részre, u.n. ablakot nyithatunk benne. Ez az utasítás is több szolgáltatást tartalmaz:

WINDOW: SPLIT, BORDER, CLOSE, LINK

WINDOW: SPLIT - ablak készítés. A táblázatnak a képernyőn látható részét nevezzük az aktív ablaknak.

Lehetőség van az aktív ablak maximálisan nyolc részre osztására, vízszintes, vagy függőleges irányban.

Az ablakok közötti közlekedés a CTRL-W billentyűk használatával lehetséges. A 2 gomb egyidejű megnyomásával a következő ablak válik aktívvá.

WINDOW: BORDER - utasítás választásával megváltozik az aktív ablak keretezettsége. Pontosabban, ha eddig az ablak be volt keretezve, akkor ez megszűnik, és ha nem volt, akkor keretezetté válik.

A keret készítésénél tudnunk kell, hogy a szélső oszlopot illetve sort ezzel felhasználjuk, azaz a táblázat két sorral, és két oszloppal keskenyebb lesz.

WINDOW: CLOSE - utasítás a megadott ablakot törli. A megmaradt ablakok mérete úgy változik, hogy a törölt ablak sorait és oszlopait is lefedjék, de ezek tartalmát nem változtatja. Az ablakok átszámozását automatikusan elvégzi.

WINDOW: LINK - utasítás két ablakot kapcsol össze. A MULTIPLAN csak olyan ablakokat képes összekapcsolni, amelyek egy korábbi ablak megfelelő felosztásával jöttek létre.

XTERN - az utasítás segítségével léthesíthetünk kapcsolatot az aktív és az úgynevezett forrás táblázatok között. Az aktív táblázat mezőit úgy is feltölthetjük, hogy tartalmukat más táblázatok mezőinek kiolvasásával kapják. Ha a forrástáblázat adatait módosítjuk, úgy ez közvetlen hatással van az aktív táblázat értékeire is. Ez a változás természetesen csak az aktív táblázat betöltésekor jön létre, s ha a változást meg akarjuk őrizni, akkor az így létrejött táblázatot ismételtlen mágneslemezre kell menteni.

XTER: COPY, LIST, USE

XTER: COPY - egy nem aktív külső (external) táblázatból adatokat másol az aktív táblázatba. Az aktív táblázat védelme érdekében csak akkor lehet a

forrástáblázatból a célterületre adatokat másolni, ha a célterület minden mezője üres.

A külső másolat utasítás csak a mezők értékes tartalmát másolja, tehát képleteket nem.

Ha a kapcsolatot létrehoztuk, akkor minden TRANSFER: LOAD utasításnál automatikusan bemásolódnak a kijelölt értékek a célmezőbe. A létrehozott kapcsolat stabil, és a forrásértékek változásait a céltáblázatban is figyelembe veszi.

XTERN: LIST - utasítás az aktív táblázatot segítő, illetve az aktív táblázattól függő táblázatok listáját adja meg.

XTERN: USE - utasítás egy adott segítő, vagy forrás táblázathoz tartozó kapcsolatokat egy másik táblázathoz tartozó kapcsolatokkal helyettesíti.

3.3.3. Főkönyvi könyvelés a MULTIPLAN segítségével

Először azt nézzük meg, miért hasznos ha a könyvelést a MULTIPLAN alkalmazásával készítjük el.

A táblázatos formátum nagyon jó, olyan mintha könyvelési bizonylat lenne. Az oszlopok nevét mi adhatjuk meg.

Az egyes mezők kitöltésénél képletek is alkalmazhatók, amelyek segítségével nemcsak az elkönyvelés, hanem pl. az összegzés is könnyen elvégezhető.

A mezőkre azonosítókkal is lehet hivatkozni, ami nemcsak emlékeztetőnek, hanem a csoportosításnál is segítség.

A MULTIPLAN programcsomag lehetővé teszi a különböző táblázatok közötti kapcsolatot, és ez a könyvelési rendszer egymásba kapcsolódó számlaosztályainál elengedhetetlen.

Tekintsünk most egy konkrét könyvelési példát.

A Magyar Acél (fiktív) Vállalat 1985. decemberi mérlegének egy részletét, a 2. számlaosztályt készítjük el.

A 2. számlaosztály a készenléti állományt foglalja magába, amelynek része a készletállomány, a vásárolt készletek, a saját termelésű készletek, és a jóléti eszközök.

Legyen a vállalatnak nyolc gyáregysége, az ország különböző pontjain. Minden egységben külön végzik a könyvelést, a központba már csak összesített adatok kerülnek. Ideális esetben a gyáregységek is rendelkeznek Commodore 64 számítógéppel, mert így az adatcserét már MULTIPLAN rendszerű táblázatokkal lehet megoldani. Ebben az esetben célszerű az összes táblázatot egy mágneslemezre gyűjteni (ha elférnek), így könnyebb a feldolgozás.

A mi feladatunk a gyáregységek részadatainak vállalati szintű összesítése.

Az alapgép és csatlakozó egységeinek bekapcsolása után betöltjük a MULTIPLAN programcsomagot a memóriába.

Az alábbi képernyő jelenik meg (lásd 3.3.1 ábrát).

Megkezdődhet a táblázat feltöltése.

A kurzor az R1C1 mezőn áll. Célszerű oszloponként haladni a feltöltéssel.

A formátum kódok beállítását is célszerű oszloponként végezni. A táblázat kitöltése előtt a szabványos formátum kódokat kell rögzíteni.

Az F, D, W betűk egymás utáni leütése a szabványos oszlop beállítás lehetőségét adja, válasszuk ezt most 22-re. Az első oszlop hosszú szövegkarakterei is elférnek egy ilyen oszlopban.

Az utasítás menüje:

```
FORMAT DEFAULT width in chars:22
```

Az F, D, C betűsorozattal pedig a mezők szabványos formátumát állítjuk be. Legyenek ezek: a szöveg balra, a számok jobbra igazítása, valamint a számok legyenek a maximálisan lehetséges tizedesjeggyel kiírtak.

```
#1      1      2      3      4
1      ████████
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
COMMAND ALPHA Blank Copy Edit Form
Go Help Ins Lock Move Name Opt Print
Quit Sort Transfer Value Window Xterm
Select option or type command letter
R101      1000 TEMP
```

3.3.1 ábra

Hogy az első oszlop első huszonhét elemére ne a szabványos formátum legyen érvényes, a FORMAT cells utasításban kell eltérő utasítást adni. *A szöveg itt központosított legyen.

```
#1          1
1          [REDACTED]
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
ALPHA: MAGYAR ACEL
Enter text (no double quotes)

R1C1          100%,TEMP
```

3.3.2 ábra

A többi oszlopelem formátumát így állítjuk be: /Az almenü minden lehetséges opciót mutat, és a kiválasztott fehér téglalapban (inverz alakban) látható.

Mi az egyszerűség kedvéért csak a kiválasztott betűjelét írjuk ki. Ez érvényes az ezután következő utasításleírásokra is. /

```
FORMAT cells: R28:70C1 align:L
```

```
code:D
```

(A kódok részletes leírása a 3.3.2. részben szerepel.)

A formátum kódok beállítása után az ALPHA utasítással megkezdjük a megnevezések bevitelét. A MULTIPLAN előnyei közé tartozik, hogy nem kell minden mező kitöltése után lezárni az utasítás végrehajtását, majd újra aktiválni, hanem a bevétel az iránybillentyűk segítségével folyamatosan végezhető.

Ha tehát beírtuk a Magyar Acél nevet az R1C1 kurzor állásnál (lásd 3.3.2 ábrát), és leütjük a lefelé billentyűt, a szöveg bekerül a táblázat adott helyére, a kurzor az R2C1 mezőbe kerül, és az utasítás sorban az ALPHA/VALUE kiírás várja a folytatást (lásd a 3.3.3 ábrát).

Miután kitöltöttük az első oszlopot, RETURN-nal lezárjuk a bevételt.

A második oszlop a rövidítéseket tartalmazza. Ezek a betűcsoportok lesznek az adott sorok azonosítónevei is.

A formátum megadását itt is célszerűségi szempontok alapján választottuk ki. A szabványostól csak az oszlopok szélességében tértünk el, elegendő 8 karakter az azonosítók számára.

A módosítást a:

```
FORMAT WIDTH in chars or default: 8  
          columns:2   trough:2
```

Mivel a szabványos oszlopszélességet csak az első oszlopban hagyjuk meg, így ezt a módosítást minden kitöltendő oszlopnál el kell végezni, természetesen mindenhol a legalkalmasabb szélességet kijelölve.

Az azonosítók kitalálásánál igyekeztünk olyan betűcsoportokat adni, melyek emlékeztetnek a megnevezésre.

A bevitel itt is az ALPHA utasítással történik.

```
#1          1
  1 MAGYAR ACEL
  2          ██████████
  3
  4
  5
  6
  7
  8
  9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
= ALPHA/VALUE █
Enter text or value

R1C1                                98% TEMP
```

3.3.3 ábra

Az utasítás lezárása után a kurzor gyors felvitele a GO segítségével oldható meg. Ha a G leütése után a következő oszlop első elemének abszolút pozíciós adatait adjuk meg, akkor a RETURN leütése után a kurzor mindjárt

odakerül.

A következő oszlopok már a konkrét értékeket tartalmazzák. Ezt formátumuk megadásában is jelezzük. A számok előtt a \$ jel kiírását, valamint az ezrespont használatát szeretnénk elérni. Ennek módja:

```
FORMAT cells:R3:70c3:11 align: R code: $
```

```
FORMAT OPTIONS commas:yes
```

```
          formulas:no
```

Ezzel azt is megadtuk, hogy a kiszámított, illetve a más táblázatból kivett értékeket mutassa a táblázat, ne a meghatározó összefüggéseket.

Ha az utóbbi választást megváltoztatjuk, az oszlopok szélessége automatikusan a megadott duplájára nő.

Nagyon fontos, hogy a képletek bevitelére az EDIT utasítást használjuk. Ha az ALPHA utasítással próbálkoznánk, lévén betű karakterek vannak a képletekben, akkor a MULTIPLAN szövegnek értékelné a beviendőket, és sem számítást, sem kiértékelést nem végezne.

Ezek az oszlopelemek, mint az az első mellékletből is látszik, négy féle módon kaptak értéket.

A legegyszerűbb értékmegadás, mikor magát a számot írjuk be, és az a táblázatba a megadott formátumban kerül be.

Az értékeket megadhatjuk a relatív pozíciós adatok kombinációjával is, pl.

```
R[-1]C+R[-2]C
```

Ez azt jelenti, hogy ide az oszlop előző két elemének összege kerüljön.

A legutolsó oszlop minden eleme a SUM(név) függvény segítségével képződött.

A legbonyolultabb értékmegadási forma, ha külső táblázatból kell adatot bevinni az aktív táblázatba.

Ilyenkor a XTERN COPY utasítás szükséges. A megjelenő almenüt a következő képpen kell kitölteni:

```
EXTERNAL COPY from sheet:9,fok2.85a
      name:osszeg   to:R3C3
      linked: yes
```

Ez azt jelenti, hogy a 9,fok2.85a táblázatból vesszük ki az összeg értéket, az R3C3 mező részére.

Összekapcsoljuk a két táblázatot a linked:yes paranccsal, ami azt jelenti, hogy a forrástáblázat változása magával hozza a céltáblázat automatikus változását.

Az első mellékletből jól látszik, hogy a 3-10 oszlopok megadása nagyrészen megegyezik. Ezért célszerű csak a harmadik oszlopot kitölteni, majd utána a COPY utasítás segítségével a tartalmát átmásolni a többi oszlopba. Így a számmal megadott értékek is átmásolódnak, holott ezek különböznek egymástól.

Ha a kurzor-t az adott mezők egyikére állítjuk, és a VALUE utasítást aktiváljuk, akkor könnyen felülírhatjuk az ott szereplő számot. Az egész táblázatban kevés átírandó mennyiség van, a kurzor-nak a GO utasítással való mozgatásával lerövidíthetjük a javítási időt.

A másolás menete tehát:

```
COPY RIGHT: number of cells:7
      starting at:R3:61C3
```

Azaz az R3C3 mezőtől jobbra hétszer másoljuk le ugyanazt az oszlopot.

Vigyázat! A másolás előtt már be kell állítani az oszlopok formátumát, különben hibajelzést kaphatunk, vagy nem történik meg a másolás.

A második oszlopban reálított rövidítéseket rendeljük hozzá az ugyanabban a sorban levő, értékeket tartalmazó oszlopokhoz. Erre azért van szükség, mert a 11. oszlopban az azonosítók szerint végezzük az összegzéseket.

A név megadása a 2. oszlop felhasználásával gyorsan végrehajtható.

A kurzor-t vigyük az R3C2 mezőre és adjuk ki a NAME utasítást.

```
NAME:define name:osszeg to refer to:R3C3:10
```

Tehát, ha az R3C2 mezőn állunk, akkor a MULTIPLAN javasolt névként az aktív mező értékes tartalmát javasolja azonosító névnek.

Így egyszerűen csak a kurzor mozgatásával, és a tartományok megadásával elvégezzük az egymáshoz rendeléseket.

A táblázat ilyen módon történő kitöltése után ellenőrizzük a felvitelt. Ha hibátlanak ítéljük meg, akkor érdemes levédni a mezőket a véletlen törléstől. Ehhez a LOCK utasítást aktiváljuk.

Először:

```
LOCK FORMULAS: yes
```

 utasítással minden képletet, illetve szöveget tartalmazó mezőt levédünk. Mivel ez az utasítás a számokkal feltöltött mezők védettségi állapotát nem befolyásolja, így erről külön kell gondoskodni.

Ennek formája:

```
LOCK cells: R10C3 Locked
```

Ha valamelyik védett mezőnél hibát találunk, úgy szintén ennek az utasításnak a segítségével megváltoztatjuk a védettségi állapotot, és csak utána végezhetjük el a javítást.

A levédést utána ismételten végre kell hajtani.

A munka befejezéseként még lemezre visszük a felvitt táblázatot. Ennek módja a TRANSFER utasítás használata:

```
TRANSFER SAVE filename:klt85i
```

Így klt85i néven lemezfile-t készítettünk, mely minden bevitt utasításunkat tartalmazza.

Szükség esetén nyomtatóra is kivihetjük a táblázatunkat. Elsőként a nyomtatási opciókat határozzuk meg.

```
PRINT OPTIONS: area:R1:61C1:11  
                formulas:yes   r-c numbers:yes
```

utasítás kiadásával készült.

A 'formulas:no' változtatás hatására nem a képletek, hanem az általuk meghatározott értékek kerültek nyomtatásra.

A nyomtatás előtt meg kell adni a margókat, és a laphosszúságot is. Mivel minden nyomtatóhoz más választás célszerű, így ezt nem részletezzük.

Ha a munkát folytatni akarjuk, csak más táblázat létrehozásával, akkor a TRANSFER CLEAR utasítás segítségével kitöröljük a memóriából az eddigi táblázatot, és új beviteléhez kezdhetünk, az eddigiekben ismertetett módon.

Fontos tudnivaló még, hogy a mágneslemez meghajtó egységben általában a MULTIPLAN rendszerlemezt tartjuk, mert egy-egy utasítás kiadásakor az almenük betöltéséhez beolvasásra van szükség.

Ne legyen azonos a rendszerlemez a munkalemezünkkel.

A rendszerlemezt írásvédő szalaggal védjük le, nehogy megsértsük a programot.

Ha a munkalemezre lenne szükség, akkor

```
Enter Y to retry access to klt85i(file név)
```

üzenettel jelzi a gép. Kivételt a TRANSFER SAVE utasítás képez, ilyenkor felszólítás nélkül a munkalemezt tegyük a meghajtó egységbe, az utasítás kiadása előtt.

3.4 A COMPAL

A COMPAL rendszert akkor célszerű használni, ha az adatállomány hierarchikus rendszerű, vagy ilyenné alakítható. Az adatok karton formátumban tároltak, a kartonok u.n. kartoték dobozban gyűjthetők. Ezeknek a dobozoknak egymáshoz való viszonya reprezentálja a hierarchikus felépítést. A magasabb szintű dobozok kartonjaihoz vannak hozzárendelve az alacsonyabb szintű dobozok, kartotékok.

A COMPAL programcsomag olyan adatfeldolgozó rendszer, mely felhasználásához minimális programozói tevékenység, ismeret szükséges. Széles lehetőségeket biztosít viszont a jól programozó felhasználónak is, hiszen különböző szubrutinok beépítésével a programcsomag szolgáltatásainak lényeges bővítéséről lehet gondoskodni.

Az adatok bevitelekor többfajta ellenőrzési módra van lehetőség. Az adatok egymás közti műveletei új adatok létrehozását szolgálják. Az adatok között megfelelő információk alapján kereshetünk, válogathatunk, kigyűjtést végezhetünk. Ezekre később részletesen visszatérünk.

3.4.1 A COMPAL eszközigénye

Commodore 64 személyi számítógép

Commodore 1541, vagy 8250 típusú mágneslemez meghajtó egység.

Televízió, vagy monitor, megfelelő antennacsatlakozóval

Commodore sornyomtató (pl. 1515, 1525, 1526 stb. típusú) és csatlakozó

COMPAL rendszerlemez

6	"szemny.bkg.01"	seq
6	"szemny.dsp.01"	seq
20	"rparin.par.00"	seq
21	"p"	prg
14	"rparin.frt.00"	seq
77	"s21"	prg
55	"s1"	prg
2	"lista"	prg
4	"s6"	prg
7	"s7"	prg
24	"s8"	prg
98	"s23"	prg
82	"s3"	prg
35	"s4"	prg
90	"s22"	prg
17	"s12.all"	prg
20	"szemny.dsd.01"	rel
7	"pall"	prg
13	"szemny.asr.02"	seq
12	"szemny.asr.01"	seq
9	"szemny.par.00"	seq
5	"szemny.frt.00"	seq
1	"s11"	prg
17	"s12"	prg
2	"s5"	prg
21	blocks free.	

COMPAL tranzakciólemez

83	"t21"	prg
77	"t71"	prg
10	"tranze"	prg
57	"s81"	prg
4	"s82"	prg
109	"t22"	prg
119	"t23"	prg
65	"p/2"	seq
128	blocks free.	

3.4.2 A COMPAL részletes ismertetése

A COMPAL alkalmazás generátor ügyviteli feladatok számítógépes megoldására szolgál. Ebből következően a manuális ügyviteli kartoték struktúrát igyekszik követni.

A COMPAL-ban nem beszélhetünk a hagyományos értelemben vett adatállományról, hanem kartoték struktúráról.

Az elemi adatokat, mezőket kartonba szervezi. Az azonos típusú kartonok halmaza létrehozza a kartoték dobozt. Egy feladat meghatározásában a több kartoték doboz előfordulhat. Ezek a kartoték dobozok hierarchikus struktúrát alkotnak. Erre a hierarchikus struktúrára a fa adatstruktúra a jellemző, mindig felülről lefelé halad a legalacsonyabb szintig. Egy hierarchia szinten több kartoték doboz is lehet.

A kartoték dobozokban elhelyezkedő adatok alkotják az adatállományt. Lehetőség van az adatok egy részéből adatbázist létrehozni, mely segítségével más COMPAL alkalmazás is hozzáférhet ezekhez az adatokhoz.

Az elején néhány alapvető fogalmat kell megismerni, melyek nélkülözhetet-

lenek a rendszer alkalmazásában.

Alkalmazások

Az alkalmazás a COMPAL rendszerben megírt felhasználói program, amely tartalmazhat nyilvántartásokat, tranzakciókat. Ezt a programot az alkalmazás-generátor értelmezi és alakítja át futtatható formára.

Az alkalmazások az adatállomány elemein műveleteket végezhetnek. A COMPAL rendszer erre is széles körű lehetőséget biztosít.

1. Az adatmezőn belül, kartonmezők, kartonok, dobozok közötti mozgás lehetőségei.

2. Karton adatain végezhető

- összeadás
- kivonás
- szorzás
- osztás
- összegzés
- felülírás
- mezők karaktereinek ütköztetése

3. Az általánosítható belső műveleteken kívül még Basic nyelven megírt tranzakciós szubrutinok más műveletekre is képessé teszik az alkalmazást.

A COMPAL rendszert általában adatállomány létrehozása, annak kezelése céljából alkalmazzuk. Az adatállomány hierarchikus felépítésű, melynek struktúráját az első adatfelvitelkor kell meghatározni. Későbbiekben az adatok módosítása, törlése stb. ennek a struktúravezérlésnek a felügyelete alatt történik.

Az adatállományok és alkalmazások között egy-egy értékű kapcsolat van. Éppen ezért az alkalmazás egyidejűleg csak egy adatállományban dolgozhat, illetve más alkalmazás által létrehozott adatállományhoz nincs hozzáférési lehetősége.

Ezeknek a szigorúknak tűnő megkötéseknek a feloldására van mód az adat-

bázis létrehozásával.

Adatbázis

Az adatbázis alapvetően különbözik az adatállománytól. Elsősorban, nem hierarchikus felépítésű, inkább hálós rendszerrel jellemezhető.

Az adatbázis is természetesen adatokat tartalmaz, még hozzá olyanokat, amelyek bármely adatállományban előfordulhatnak. Az adatbázis tehát olyan adathalmaz, melyhez bármely alkalmazás hozzáférhet, módosíthat rajta, illetve új adatokat jegyezhet bele. Célszerű az adatbázisban csak olyan adatokat elhelyezni, melyek ténylegesen több adatállomány létrehozását, feldolgozását segítik.

Bázisállomány

Az adatbázis bázisállományokból áll. Minden bázisállományt egy-egy kulcs segítségével lehet elérni. A kulcs, melyet a létrehozó alkalmazás definiál, meghatározza a bázisállomány rekordtípusát, mezőfelosztását.

Egy bázisállomány tetszőlegesen sok, azonos típusú adatot tartalmazhat, de csak egy kulcsot. Valamely kulcsot az alkalmazás során definiáljuk, a többi alkalmazás csak hivatkozik rá. Elvileg bármely alkalmazás létrehozhat új kulcsot, új bázisállományt az adatbázisban, de ez nem feltétlenül szükséges.

Mivel az alkalmazások is hierarchikus alapra épülnek, ezért adódik az a megkötés, hogy egy alkalmazás egy hierarchia szintje csak egy bázisállományt jelölhet ki kulcsaktualizálásra. Természetesen hivatkozás során több bázisállományt is elérhet, és itt létrehozhat új mezőket is.

A kulcsok segítségével nagyon gyors gyűjtés, keresés jön létre.

Az alkalmazások kétféle módon használják a kulcsokat:

1. újat jegyeznek az adatbázisba, és ezzel új bázisállományt hoznak létre.
2. már meglévő bázisállományban dolgoznak, az eredeti kulcsra való hivatkozással.

A bázisállományban megtalálhatók mindazon kartonok mutatói, amelyek

hivatkoznak az adott bázisállományra, illetve adatot írnak bele. Ezen mutatók tehát, az adott alkalmazásra nézve megmondják, hogy mely kartonok használják az állományt. Így a COMPAL rendszer gyorsan, kevés lemezmeghajtó művelettel válogat.

A fentiekből adódik még egy megkötés. Egy alkalmazás egy adatbázishoz viszonyítva csak egy adatállományt hozhat létre, különben keverednek az alkalmazáshoz tartozó adatkartonok.

Adatok

Az adatok szám illetve betű típusú karaktereket tartalmazhatnak. A legkisebb, logikailag értelmes karakterláncot nevezzük mezőnek. A mezők hosszát a fizikai képernyősor korlátozza, a mezőt megelőző, a mezőhöz szorosan kapcsolódó megnevezés határozza meg. A mezők tartalma és keletkezése alapján több csoportba sorolhatók.

Tartalom szerint lehetnek:

- tetszőleges karakterlánc
- táblaelem sorszám
- belső művelettel létrejött mező
- bázisállomány kulcsa
- bázisállomány adata

Keletkezése szerint lehet a mező tartalma:

- bevitt
- tárolt
- kijelzett

ezek tetszőleges kombinációja

Táblaelem

Az előbbieken azt mondtuk, a mező tartalmazhat táblaelemsorszámot. A tábla

egy maximálisan 255 elemből álló adathalmaz. Akkor célszerű táblákat készíteni, ha előre létrehozott fix menüből akarunk többször is aktuális adatot keresni.

Ha egy mező egy táblaelem sorszámot tartalmaz, akkor a mező elérésekor a tábla automatikusan listázható a képernyőre.

Karton

A mezők fizikailag az adatkartonokon helyezkednek el. A karton maximálisan 12 sort tartalmazhat, s egy-egy sorban több mező is lehet. Az alkalmazás tervezésnél meg kell adni a kartonok képernyő megjelenítési formáját.

Az adatkartonok is követik a hierarchikus építkezést, ily módon egy alkalmazás maximum 99 fajta kartont definiálhat.

Kartotékdoboz

Az azonos hierarchia szintű, azonos fajtájú kartonok kerülnek egy kartotékdobozba.

/Az eddig használt kifejezések, közöttük ez is, alkalmazása azért célszerű, mert utalnak az irodai ügyvitel eddig kialakult szóhasználatára./

A doboz méreteit az alkalmazás tervezésnél meg kell adni, melynél tetszőleges bővítési lehetőséget az ad, hogy ha a felső limitként 9999-et adunk meg, azt a rendszer végtelennek értékeli, tehát akárhány kartont elhelyezhetünk az ilyen dobozban. A hierarchikus felépítés igényeket támaszt a felhasználó felé, de támogatást is nyújt. A struktúra akkor működik jól, ha az eredetileg definiált dobozok mindegyikét feltöltjük, illetve legalább a minimális számú kartont kitöltjük. Enélkül megszűnne a folyamatos lánc a kartonok-kartotékok között. Éppen ezért a COMPAL rendszer az adatok beírásánál figyel erre, és nem engedi a felhasználót újabb doboz aktivizálására, míg az adott dobozt minimálisan fel nem tölti.

A rendszer indítása:

Mielőtt bármelyik részegységet bekapcsolnánk a COMPAL hardver bővítő cartridge-t kell csatlakoztatni az alapegységhez.

A cartridge csak a felirattal felfelé üzemképes, fordított csatlakoztatás hatására a cartridge károsodik.

/A cartridge nem befolyásolja a Commodore 64 működését, csak a COMPAL program futtatása aktivizálja. Épp ezért ne húzzuk ki a hardver bővítőt, ha nincs szükség a bemenetre más rendszerek futtatásakor. /

Ezután a használati utasításban megadott sorrendben üzembe helyezzük az alapgépet és kiegészítő egységeit. A

```
LOAD 'PALL', 8
```

utasítással a rendszerlemezről beolvassuk a programot. A READY kijelzésre kiadhatjuk a RUN parancsot. Ekkor a képernyőn rövid ismertető olvasható, mely mindaddig látható, míg az I gombot meg nem nyomjuk. Ezután ismételt beolvasás történik, majd a képernyőn a főmenü jelenik meg.

1. Segítő információk
2. Alkalmazás futtatása
3. Alkalmazás tervezése
4. Alkalmazás generálása
5. Tranzakcióprogram indítása
6. Tranzakció modul tervezése
7. Tranzakció modul tesztelése/belövése
8. Tranzakcióprogram generálása
9. Lemez karbantartó műveletek

Foglalkozzunk az egyes lehetőségekkel kicsit részletesebben. Tudnunk kell azonban, hogy az egyes menüpontok működése a RUN/STOP és RESTORE gombok egyidejű megnyomásával megszakítható. Ilyen esetekben a főmenühez jutunk, vagy egy teljesen üres képernyőt kapunk a READY felirattal. Ekkor a POKE 51200-128,0 utasítás begépelésével vissza jutunk a főmenühez, azaz

visszaállítottuk a rendszer indítása utáni állapotot.

A menü egyes pontjait nem a felsorolás; hanem a munkák elvégzésének sorrendjében ismertetjük.

1. Segítő információk

Az első menüpont táblázatok formájában a rendszer rövid ismertetését adja, szemléletesen bemutatva a hierarchikus felépítését.

A táblázatok fizikailag több képernyőn férnek el, a lapozás a RETURN (előre) és a cursor-le (hátra) gombokkal lehetséges.

3. Alkalmazás tervezése

Ez a menüpont új alkalmazás tervezéséhez ad interaktív segítséget. Ez a tervezés legutolsó fázisa. Kezdetben csak a papír és a toll nyújt segítséget.

Az első munkafázis a feladat pontos meghatározása. Milyen adatokat kell használni, és ezeknek milyen a hierarchia szerkezete. Az adatbázissal hogyan kell kapcsolatot létesíteni, és milyen kulcs aktualizálású bázisállományt kell készíteni.

Legcélszerűbb az u.n. családfa rajzolás, mely a legszemléletesebben mutatja az adatstruktúrát. A nagyvonalú rendszer szervezés után a legkisebb elemétől indulva építjük fel a struktúrát.

Első feladat az adatkartonok formai, logikai megtervezése. Azaz figyelni kell a karton megjelenítésére a képernyőn, a kartonok egymás közti kapcsolatának megteremtésére, valamint, hogy a kartonokon elhelyezett adatmezők tartalmazzák az összes szükséges információt.

Következő lépés a kartonok dobozba rendezése. Itt nemcsak a kartonok jó sorrendjére, megfelelő dobozolására kell ügyelni, hanem a dobozok megfelelő szintre való helyezésére is. Az azonos hierarchia szintű dobozokat az alkalmazás a definiálás sorrendjében értékeli.

A COMPAL rendszer maximálisan 9 különböző szintet képes kezelni, és ezeken

belül összesen 99 doboz helyezhető el.

Az azonos szinten történő elágazás nem limitált.

Meg kell tervezni a munkaindító kartont, mely minden újraindításnál kitöltésre kerül. Ez a karton a hierarchia legfelső szintjén álló kartoték elé kerül, és elsőként vár a feldolgozásra. Formátuma kötött, minden mezőnek az adott hosszal, illetve sorrendben kell következniük. Megnevezésük változtatható.

Minden kartonnak, kartotéknak, bázisállománynak, bejegyzésnek számos kötelező információt kell tartalmaznia. Ezek formátuma, megkötései az alkalmazás tervező képernyő segítségével követhetők.

Aki már gyakorlott a magasszintű programozásban, annak már új alkalmazás létrehozására nem feltétlenül kell használnia az interaktív alkalmazás tervezőt. A saját készítésű Basic program csak DATA sorokat tartalmaz. A program írásakor tisztában kell lenni az összes szintaktikai megkötéssel, mert ezeket az igényeket így is ki kell elégíteni. Az így megírt alkalmazások nagy előnye a gyorsabb működés.

Az alkalmazások megírásakor gondoskodni kell megfelelő dokumentáció írásáról is, hogy minden felhasználó egyértelmű kezelői utasításokat kapjon.

Az alkalmazások tervezésénél lényeges szempont a bevitt adatok ellenőrzése, az alkalmazások futtatásakor az adatkartonok helyes kitöltésének a vizsgálata. Ezek a kritériumok tartalmazhatnak formai és tartalmi megkötést, valamint figyelhetik a mezők megfelelő összekapcsolását, összehasonlítását.

Az alkalmazás-tervezés elindításához tehát a hármas főmenüpontot aktiváltuk. Az ehhez szükséges információk a rendszerlemezről töltődnek a memóriába. Az alkalmazás-tervezés közben is gyakoriak a mágneslemezen végrehajtandó műveletek, ez a Commodore 64 számítógépen viszonylag lassú. De ne legyünk türelmetlenek, a gép akkor is dolgozik, ha ennek külső jele nincs! Óvakodjunk ilyen esetekben a billentyűk nyomogatásától, mert nehezen helyrehozható hibákat csinálhatunk.

A munkát akkor kezdhethetjük el, amikor az első táblázat megjelenik a

képernyőn, és a mágneslemez meghajtóegységbe behelyeztük az alkalmazás lemezünket. A kezelői üzemmód megadásánál jelenik meg az alkalmazás tervezéshez tartozó almenü.

- 1.első felvitel
- 2.módosítás
- 3.forrás feldolgozás
- 4.forrás kibővítés
- 5.munka vége

Alkalmazás-tervezésnél az első felvitel üzemmódot kell választani. Ilyen esetben üres, kitöltendő tervező képernyőket kapunk. A tervező táblázatok alatt segítő információkat, vagy kötelezően betartandó utasításokat kapunk.

A felvitel utolsó kartonjánál ne a RETURN, hanem az F6 funkciós billentyűt nyomjuk le, így zárjuk le a felvitelt.

Módosítás üzemmódban az előzőekben megadott kartonokon hajthatunk végre változtatásokat.

A 3. almenüpont választása teszi alkalmassá a kész alkalmazást a generálásra.

A forrás kibővítése a már létrehozott alkalmazásban új kartotékok bevitelét, a struktúra kibővítését teszi lehetővé.

A munka befejezésekor az 5-s gombot nyomjuk meg, ne a radikális RUN/STOP RESTORE kiszállást alkalmazzuk, mert ilyenkor még a rendszer végez műveleteket.

4. Alkalmazás generálás

Az alkalmazás generálás üzemmód a megtervezett, rendszerbe vitt alkalmazás forrásból készít működő alkalmazást. A soros felépítésű forrásállományból

paraméter- és másodlagos forrásállományok jönnek létre. A generálás négy menetben történik, ezek jól követhetők a képernyőn.

a, Előkészítés: ennek módja attól függően változik, hogy az alkalmazás forrást az interaktív alkalmazástervezővel, vagy a magasszintű nyelv alkalmazásával készítették. Míg az első esetben az előkészítést is maga a rendszer végzi, a második esetben a program mentése, futtatása a mi feladatunk.

b, Másodlagos alkalmazás forrás létrehozása, valamint a definiált mezők típus szerinti csoportosítása.

c, Az adatmezők és a kartonok értékelése.

d, A műveletek és az ellenőrzések kiértékelése, vezérlőparaméterek létrehozása.

A generálás során mind a szintaktikus, mind a definiálási hibák előjönnek, a logikai hibák viszont nem. A hibák javítására ismét a harmadik menüpontot kell aktivizálni, és a módosítás üzemmód segítségével hibátlan alkalmazást készíteni. Ezután ismét generálás előkészítést, majd generálást kell végezni.

A rendszerlemezen található 'LISTA' nevű programmal az alkalmazás elsődleges és másodlagos forrása listázható mind a képernyőre, mind a printerre. Ez is segítséget adhat a javításhoz.

2. Alkalmazás futtatása

A már létrehozott alkalmazás betöltése a megfelelő lemezről, majd ennek a futtatása tehető ennek a pontnak a segítségével.

Egy-egy alkalmazás többfajta adatot is kezelhet. Ezek száma minél több, annál nagyobb helyet foglalnak el a memóriában. Éppen ezért a fölösleges helycsökkenés elkerülésére célszerű előre meghatározni, hogy a futtatandó alkalmazás az adatait honnan veheti. Ennek segítésére szolgál a képernyőn

megjelenő almenü.

1.MIN - A legkisebb memóriaterületet elfoglaló változat, mely csak az adatállományokat kezelő törzsmodul.

2.MID - Ez a változat az adatállományon kívül az adatbázist is képes elérni, de új kulcsokat létrehozni, illetve a régiekbe új kartont bevinni nem.

3.EXP - A legnagyobb memóriaterületet elfoglaló, és a legtöbb szolgáltatást nyújtó törzsmodul. Ez a változat üzemelteti azokat az alkalmazásokat, amelyeknek szüksége van a teljes adatállományra, és az adatbázisban is minden, a már eddigiekben ismertetett, módosítást, változtatást el tud végezni. A gyors, kulcs szerinti válogatás is ennél a változatnál érhető el.

Miután kiválasztottuk a megfelelő törzsmodult, meg kell adni az alkalmazás 6 karakteres azonosító nevét, és egyidejűleg a lemezmeghajtó egységbe kell helyezni a megfelelő alkalmazás lemezt.

Rövid idő múlva látható az általunk tervezett munkaindító karton. Ennek kitöltése és az adatállomány lemez készenlétbe helyezése után kezdődhet az érdemi munka, azaz a kartonok kitöltése.

Az alkalmazás futtatása előtt célszerű az adat- illetve bázisállomány méreteit meghatározni. A következő számítás segít ebben:

1. Adatállomány hossza:

Kiszámítjuk, hogy az egyes kartontípus mezőinek mekkora a hosszösszege, és ehhez 8-t adunk. Az így kapott kartonméreteket megszorozzuk a kartonok várható előfordulásának számával. Összegezzük a dobozméreteket.

2. Bázisállomány hossza:

A törzsbejegyzések hosszát megszorozzuk a várható előfordulásokkal. A kartonmutatók számát megszorozzuk 2.2-vel. Ezek összegzése után még

biztonsági ráhagyásként megnöveljük a kapott értéket 20 százalékkal.

Az alkalmazás tervezésnél leírtakhoz hasonlóan az első táblázat kitöltése közben látható az ide tartozó almenü. Ezek alkalmazása is hasonló az ott elmondottakkal.

- 1.első felvétel
- 2.módosítás
- 3.adat feldolgozás
- 4.adat kibővítés
- 5.munka vége

Ha első adatfelvitel történik, akkor a kartonok kitöltésével kezdjük. Ha az adott doboz minimális számú feltöltése megtörtént, akkor a képernyőn megjelenő újabb karton kitöltése nem kötelező. A kész karton mezőinek ellenőrzését követi a kijelölt műveletek végrehajtása. A műveleti tényezőket operandusoknak nevezzük, ezek között elsődleges a céloperandus. A műveletek célja lehet:

- mezőtartalom alakítás, áthelyezés
- aritmetikai művelet
- bázisállomány kulcsainak illetve mezőinek aktualizálása

Ha az összes kartont kitöltöttük, és az ellenőrző műveletek is végrehajtottak, megtörténik a kartonok dobozokba rendeződése, elkészült az adatállomány.

Hogy elkerüljük az adatállomány megsérülését, az adatállományt le kell zárni. Ezt lehet részlegesen, vagy véglegesen elvégezni.

A részleges lezárás az F6 gombbal történik, és lehetővé teszi az adatállomány ismételt megnyitását, esetleg más üzemmódban. Ebben az esetben a munkaindító kartont kapjuk vissza, az eredeti, indító adatokkal.

A munka befejezéséhez végleges lezárásra van szükség. Ehhez ismételten meg kell nyomni az F6 gombot. Ennek hatására az adatállomány a memóriából az

adatállomány lemezre töltődik, megtartva a hierarchia szinteket.

Ha nagy adatállománnyal dolgozunk, úgy célszerű a feltöltést a munka bizonyos fázisaiban abbahagyni, és a rész adatállományt lemezre menteni, mert így az esetleges memória törlés okozta károkat csökkenthetjük.

A munka folytatható, ha a munkaindító kartont újra kitöltjük, ekkor az előbbi alkalmazással készíthetünk új adatállományt, vagy az előző adatállománnyal is dolgozhatunk tovább.

Ha új alkalmazással, vagy tranzakcióval szeretnénk a munkát folytatni, úgy a RUN/STOP RESTORE gombok egyidejű használatával, vagy az 5-s menüpont aktiválásával, megkapjuk a főmenüt, és annak bármely pontját aktivizálhatjuk.

Az alkalmazások futtatásánál nagyon jól használhatók a speciális gombok, melyek funkcióival ismerkedjünk meg részletesebben:

RUN/STOP RESTORE - a működő rendszer bármely állapotából visszatér az alapállapotba, a képernyőn megjelenik a főmenü. Ha üres képernyőt kapnánk a villogó READY felirattal, akkor a POKE 51200-128,0 utasítással jutunk vissza a főmenühöz.

RETURN - funkció az aktuális karton elhagyását teszi lehetővé. Ha kitöltött kartonon állunk, akkor az aktuális doboz következő kitöltött, illetve üres kartonját adja. Ha üres kartonon, vagy a kartoték utolsó kartonján állunk, akkor a RETURN használata megegyezik az f1 funkcióéval.

f1 - ez a funkció a családfa ábrázolás szerint előrelépve a következő kartotéket adja.

Ha a doboz a minimális feltöltöttségi követelményeknek nem tesz eleget, akkor a doboz soron következő kartonját adja.

Ha az adott doboz minimálisan feltöltött, akkor a hierarchia szintben lefelé haladva adja a következő feltöltetlen dobozt. Amennyiben ilyen nincs, illetve rendelkezik a minimális számú kartonnal, akkor az azonos szinten keres kartotékot.

Ha ilyet sem talál akkor a struktúrában felfelé lép és ott keres, az adott kartont magába foglaló, feltöltendő kartotékot.

f2 - funkciós gomb használatával az adott dobozban lapozhatunk előre. Ha a doboz utolsó kartonján állunk, akkor a parancs hatástalan.

f3 - gomb megnyomásával az aktuális dobozt tartalmazó kartonra lehet lépni.

f4 - funkció az aktuális dobozban lapoz visszafelé, az első kartonnál a működtetése hatástalan.

f5 - funkciós billentyű mezőtartalom szerinti keresést tesz lehetővé. Üzemeltetésekor kapunk egy üres kartont, melyen a keresési szempontot kell megadni. A RETURN leütésével az első, megfelelő tartalommal rendelkező karton jelenik meg.

Nagy méretű dobozban a keresés igen nehézkes, mert a kartonokat egyenként olvassa be a rendszer a lemezzről, megvizsgálja annak tartalmát, és csak utána olvassa a következő kartont.

f6 - funkciós gomb alkalmazás, illetve adatállomány lezárására szolgál. Egyszeri megnyomása csak részleges lezárást jelent, ilyenkor a munka esetleg más üzemmódban folytatható.

Mielőtt a főmenü más pontját aktiválnánk, végleges lezárást kell alkalmazni. Ennek elmulasztása tönkretelheti egész addigi munkánkat. Éppen ezért célszerű nagyobb alkalmazás, vagy adatállomány létrehozása közben is elvégezni a végleges lezárást, ezzel csökkenthetjük az áram kimaradás okozta károkat.

A végleges lezárás hatására kerülnek a mágneslemezre a memóriában tárolt struktúralánc mutatói.

f7 - alkalmazásával még betöltetlen dobozba újabb karton szűrhető be, az aktuális karton mögé.

A beszúrt karton kitöltése kötelező.

Ha a beszúrást nem a legalacsonyabb hierarchia szinten végeztük, akkor a csatlakozó szinteken a minimális feltöltést el kell végezni. A beszúrás csak ezután fejeződik be.

A beszúrást részleges, vagy végleges lezárással megszakítani nem szabad.

f8 - ha az adott kartoték több kartont tartalmaz, mint a kötelező, akkor ezek közül kitörölhető az aktuális karton. Az F8 gomb megnyomása után üzenet jelenik meg a képernyőn, és csak az F8 gomb ismételt leütésére hajtódik végre az utasítás.

A törölt kartonhoz tartozó összes kartoték szintén törlődik. A törölt kartonok tartalma vissza nem hozható.

C= és 1 - gombok egyidejű megnyomása teszi lehetővé a keresést olyan kartotékban, amely kulcsmezőt tartalmaz.

C= és 2 - funkcióval visszatérhetünk a keresés előtti állapotra, úgy, hogy közben a kereső karton aktív marad.

C= és 3 - úgy tér vissza a keresés előtti állapotra, hogy a kereső karton megszűnik.

C= és 4 - utasítás hatására az aktív doboz legelső kartonjára lép.

C= és 5 - az aktív doboz legutolsó kartonjára lépés.

C= és 9 - bármely üzemállapotban megmutatja, hogy az adott fastruktúra melyik elemén állunk.

C= és f1 - eredményeként megjelennek a kartonon a kijelölt bázismezők.

A karton mezői karakterekből állnak. A karakterek közötti mozgást a kurzor látható elmozdulása teszi nyomon követhetővé.

CRSR jobbra - jobbra lép mindaddig, míg a mező szélén fel nem ütközik.

CRSR balra - balra lép mindaddig, míg a mező szélén fel nem ütközik.

CLR HOME - a kurzor-tól jobbra levő karakterek tartalma törlődik, a kurzor a következő mező első karakterére kerül. Ha a karton utolsó mezőjén adjuk ki az utasítást, akkor a kurzor kilép a mezőből, de a kartonon marad.

CRSR fel/le - gombok az adatmezők közötti mozgást teszi lehetővé, értelemszerűen.

DEL - a kurzortól balra levő karaktert törli.

INST - üres karakterek beszúrását teszi lehetővé, a mező megfelelő része jobbra tolódik.

> illetve < - Ha a kurzor táblaelemtípusú mezőn áll, akkor ezeknek a gomboknak a segítségével lehet előre illetve hátra lapozni a tábla tartalmán.

Tranzakció

A főmenü következő négy pontja a tranzakcióhoz nyújt segítséget. A tranzakció olyan kiegészítő BASIC rutin, mely bonyolultabb műveleteket, ellenőrzést végezhet.

Bármely karton tartalmazhat tranzakció hívó-utasítást, azaz az alkalmazás tervezésénél minden kartonról megadtuk, hogy tartozik-e hozzá BASIC rutin. Lehetőség van arra is, hogy a megírt felhasználói programot csak egyes üzemmódokban használj. Például: a kartonok, mezők tartalmi és formai ellenőrzését csak felvétel illetve módosítás közben célszerű végrehajtani.

Az adatok feldolgozásakor erre már nincs szükség, csak feleslegesen megnövelné a gép munkaidejét.

A feldolgozások során is lehet olyan funkció, pl. a mezők tartalma közti műveletek végrehajtása, amely csak abban az üzemmódban végzendő el.

Tehát a tranzakció tervezésénél ismernünk kell az adatállomány struktúráját, felépítését, kezdő és befejező kartontípusait. Csak így adhatjuk meg egyértelműen a szubrutinok végrehajtásának pontos helyét.

Az ellenőrző szubrutinok megírásakor a hibajelzéseket is nekünk kell megtervezni, megadni.

Nyomatási utasítások nincsenek a programba iktatva, így ezekről is tranzakciós szubrutinoknak kell gondoskodni.

6. Tranzakció-modul tervezése

A 6. menüpont a Commodore 64 alapállapotát hozza vissza, ugyanúgy, mint amit a rendszer bekapcsolásakor kapunk. A Basic tranzakciós rutin mindkét módon megírható. Ez is azt bizonyítja, hogy tetszőleges bonyolultságú, hosszúságú programot írhatunk, mellyel összetett problémák megoldása, ellenőrzése válik lehetővé.

Néhány alapvető megkötést viszont itt is be kell tartani.

A szubrutinok írásánál a programsorok számozása 1500-nál kell kezdődjön, ha ez a munkaindító kartonhoz tartozik. Ha más kartontípushoz írunk szubrutint, akkor a kezdőcímet a következő módon kell meghatározni:

Meg kell nézni, hogy az adott karton hányadik a dobozban, ehhez hozzáadunk egyet, és az így kapott számot 1000-rel megszorozzuk. Így pl. a 10. kartonhoz tartozó kezdőcímnél $(10+1) \times 1000 = 11000$ értéket kell adni.

Már az alkalmazás-tervezés leírásnál ismertettük a MIN, MID, EXP alkalmazás vezérlő modulokat. Az ezek közötti választás itt is tárkapacitás és feldolgozási idő megtakarítást biztosít.

A tranzakciótervezés során megadjuk, hogy az alkalmazás, melyben ez a program szerepel, milyen adatfeldolgozást igényel. Ennek tudatában, már az alkalmazás írásához egy olyan munkalemezt kell előállítanunk, a tranzakciós

rendszerlemez segítségével, mely a vezérlő modulok három változata közül csak a szükségeset tartalmazza. Ezzel a munkalemez felhasználható üres blokkjainak a számát is megnöveljük.

Fontos megkötés a szubrutinok írásakor a rendszerváltozók helyes használata. A változók alapvetően két csoportba tartoznak:

1. információkat tartalmaznak a rendszerről és az adatállományról. Ezek a felhasználó számára megváltoztathatatlanok.
2. rendszer vezérlésére szolgáló változók, melyeknek a kezdeti értékét kötelező megadni, és a változók ezt a működés közben megtartják.

Ismerkedjünk meg néhány fontosabb rendszerváltozóval, a fenti csoportosítást figyelembe véve:

1. a, KP% - kétdimenziós tömb, mely a mezőparaméterről ad felvilágosítást. Az első eleme a mező sorszáma, a második 1 és 9 között változhat.

Általában a 2 jelenti a táblaelemek báziscímét

7 az adatmező hosszát

9 a táblaelemek számát

Az adatok a tömbben a leírás sorrendjében folytonosan helyezkednek el.

b, KS - az aktuális mező sorszámát tartalmazza. A munkaindító karton esetén KS=0, a többi kartonnál definiálásuk sorrendjében növekszik.

c, KS% - kétdimenziós tömb, melynek első eleme KS, a második az alábbiak közül választható:

- 1 - mezőparaméterek bázisa
- 2 - kartonon előforduló mezők száma
- 3 - a karton hierarchia szintje
- 4 - minimális előfordulási szám
- 5 - maximális előfordulási szám
- 6 - aktuális előfordulási szám

d, RS% - négy elemű, egy dimenziós tömb.Elemei:

az előző karton sorszáma
 a következő karton sorszáma
 előző karton állapota
 következő karton állapota

2. a, FR% - segítségével közlekedni tudunk az egyes kartonok között, hierarchia szinttől függetlenül.Lehetséges értékei:

FR% < 0 nincs hatása
 FR% = 0 munkaindító kartonra lép
 FR% > 0 a megadott számú kartonra lép

b, Tranzakciós rutinba olyan utasításokat is elhelyezhetünk, hogy a speciális funkciós gombokat maga a rendszer kezelje. Ehhez az szükséges, hogy a C= gomb leütését egy változó, a H% figyelje.

H% = 255 általában
 H% = 223 a gomb lenyomásakor

c, Hasonló funkciót lát el az SF% változó. Adatfelvitel üzemmódban a kartonok közötti lapozást szintén funkciós billentyűk segítségével oldjuk meg. A szubrutinban ezt úgy valósíthatjuk meg, hogy az SF% változóban elhelyezzük a billentyűk ASC kódszámát, és így automatikusan megtörténik a lapozás.

Ha pl. C= F1-et akarunk használni,akkor a rutinban mind a H%, mind az SF% értékét be kell állítani.

E néhány ismerttetett változón kívül még vannak más, kötött rendszerváltozók is, de alkalmazhatunk általunk definiált változókat is. Ezek azonosító neve csak Z betűvel kezdődhet. Ez a megkötés gátolja meg, hogy valamelyik rendszerváltozó értékét elrontsuk.

A tranzakció modult, tervezése után, be kell vinni a gép memóriájába. Mint már említettük, ez történhet a gép alaphelyzetében, vagy a 6. menüpont alkalmazásával. Ezután a megszokott 'LOAD' utasítással mágneslemezre visszük.

A megírt modul teszteléséhez, amit a 7. menüpont aktiválásával végezhetünk el, szükség van a megfelelő MIN, MID, EXP programváltozatra. Célszerű a tesztelés közben minél több nyomtatási utasítást kiadni, hogy könnyen követhető legyen a kartonok közötti közlekedés.

8. Tranzakcióprogram generálása

Ha a tesztelés szerint jó a szubrutin, akkor generálás segítségével egy gyorsabb, futtatásra alkalmasabb változatot kell készíteni.

A generáláshoz a fentiekben leírt munkalemezt használjuk. A generálás 25 - 40 percet is igénybe vehet, a memória felhasználástól függően.

Az eredményül kapott program futtatható az 5. Tranzakcióprogram indítása menüpont aktivizálásával.

9. Lemezkarbantartó műveletek

Ezzel a szolgáltatással a rendszer kikapcsolása nélkül is végezhetünk lemezkarbantartást, azaz végezhetünk file-okkal műveleteket, és felvilágosítást kaphatunk a lemez tartalomjegyzékéről.

Az utasítások formai alakja: D/M/DOS

D - a meghajtó egység száma

M - karbantartó üzemmód választása

- tartalomjegyzék

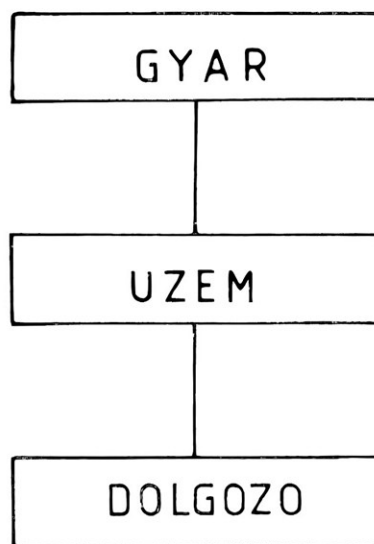
D - DOS parancsok

DOS - C 1541 mágneslemez meghajtóegységet vezérlő parancsok, eredeti Commodore 64-hez tartozó formátumban

3.4.3. Személyi nyilvántartás a COMPAL segítségével

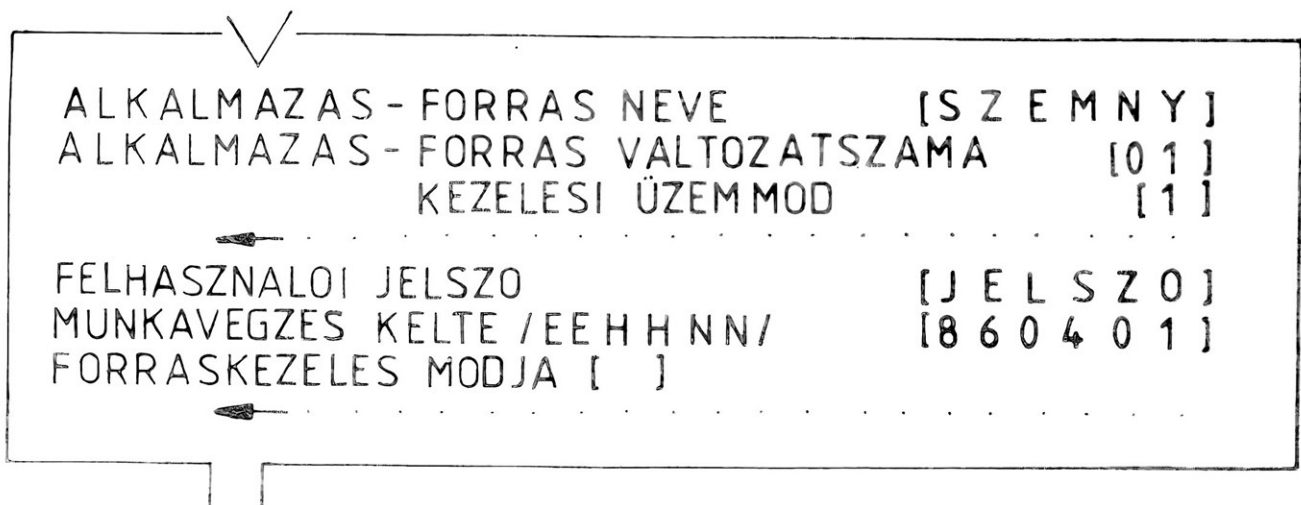
Az eddigiekben ismertettek alapján egy konkrét feladat megoldásán keresztül mutatjuk a COMPAL rendszer működését.

Mivel a COMPAL rendszer hierarchikus felépítésű, így pl. személyi nyilvántartás létrehozására is alkalmas (lásd 3.4.1 ábrát).



3.4.1 ábra

Tekintsük a legegyszerűbb gyár felépítését, három szintű hierarchiával. Legmagasabb szint a gyár, illetve gyárak, ezeken belül vannak az üzemek, és a legalsó szinten az üzemek dolgozói. Nagyon egyszerű családfával ábrázolható ez a struktúra.



3.4.2 ábra

Az alkalmazás tervezésekor a három szintnek megfelelő kartotékokat kell létrehozni. Az elágazások mennyiségéről az adatállomány felvitelekor kell

gondoskodni. Ez azért is jó így, mert pl. ha a gyáron belül két üzem egyesül, a struktúra megváltozik. Mégis szükség új alkalmazás létrehozására, csak új adatállomány bevitelre.

Mivel egyszerű a feladat, így az interaktív alkalmazás tervező használata sem növeli meg jelentősen a tervezési időt, ezért ezt használjuk (lásd 3.4.2 ábrát).

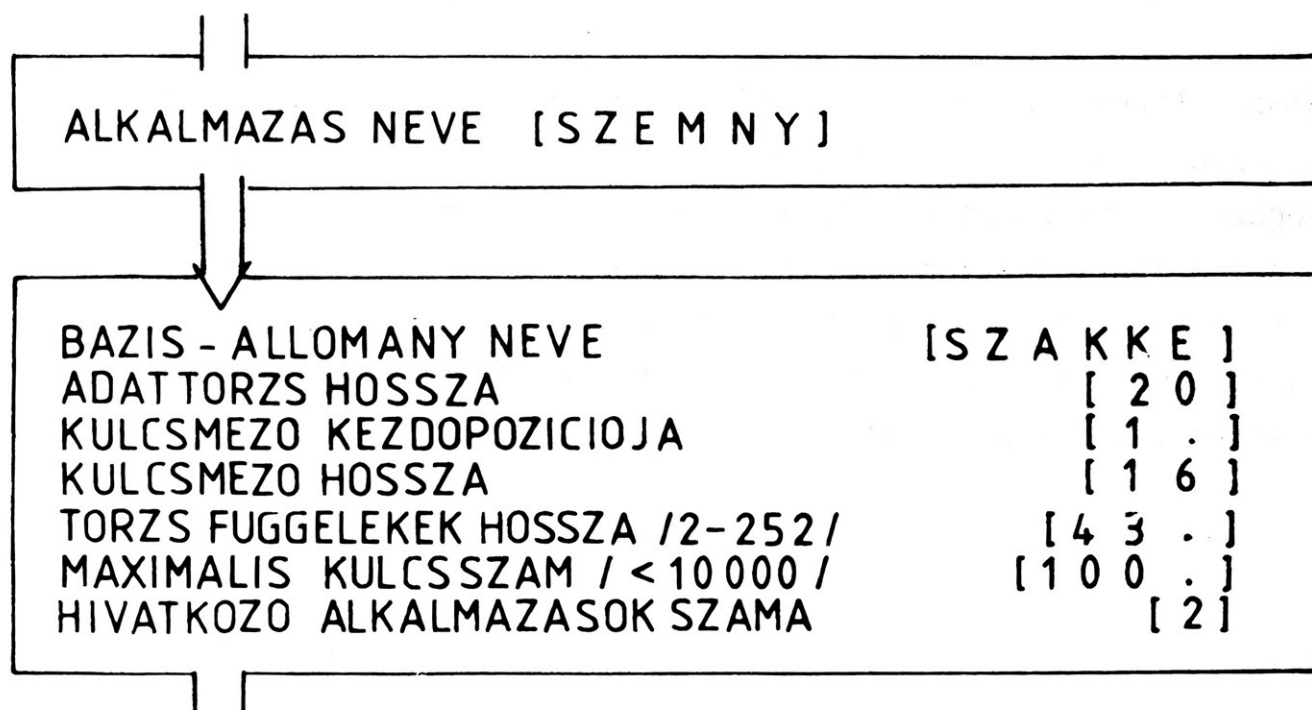
A rendszer indítása után a főmenüből a hármás menüpontot választjuk. A betöltés után megjelenik az első tervező képernyő.

A négyzetes zárójelek kitöltetlenek, a kurzor az első zárójel első karakterén áll.

'Alkalmazásforrás név'-ként tetszőleges hat karakteres nevet adhatunk, célszerű olyat választani, amelyik utal az alkalmazás adatállományának minőségére. Ha ugyanazt az alkalmazást több módosulatszámban is létrehozunk, akkor azt a 'változatszám'-mal jelezhetjük.

Ezek kitöltése után a 'kezelési üzemmód'-ot kell kiválasztani. A munka kezdetekor ez az '1.első felvitel'. Segítségként a kurzor-nak ebben a helyzetében a táblázat alatt megjelenik az almenü.

A 'felhasználói jelszó' mezőbe kötelező a 'jelszó' szót beírni, a 'munkavégzés kelte' mezőbe pedig célszerű az aktuális dátumot, elhagyva az évszázadot. Mivel az első felvitel üzemmódot választottuk, így az utolsó zárójelet kitöltetlenül hagyjuk, a RETURN gombbal zárhatjuk le a jól kitöltött kartont.



3.4.3 ábra

A lezárás előtt még lehetőség van az elütések javítására. Ha a lezárt

kartonon hagyunk kitöltetlen mezőt, vagy tartományon kívüli karaktereket adtunk meg, /pl.nem létező dátumot/, akkor a képernyő alján fekete keretben hibajelzés jelenik meg, és a karton kitöltését meg kell ismételni.

Ezek után megkapjuk a következő képernyő első sorát, majd ennek kitöltése után az alatta levő táblázatot is (lásd 3.4.3 ábrát)

Az adott alkalmazáshoz egy bázisállományt hozunk létre, melynek a SZAKKE nevet adtuk. Ezzel utalunk arra, hogy a dolgozók szakképzettségét gyűjtjük, és tároljuk ebben az állományban. Erre azért van szükség, hogyha pl. a dolgozók között szakmunkásokat keresünk, a kulcs szerint ez gyorsan végrehajtható.

Az 'adattörzs hossza' a bázisállományban a `bázisrekord` hosszát maximalizálja, ezt most 20-nak választottuk.

'Törzsfüggelék hossza'-ként 43-t írunk be. Ez a rendszer optimuma, azt jelenti, hogy husz bejegyzés fér a rekordba.

A bázisállomány maximális kulcsszámának megadása végleges, menet közben nem változtatható.

A legtöbb 'hivatkozó alkalmazás' kilenc lehet, a létrehozó alkalmazáson kívül. A táblázat kitöltése után ismét a RETURN gombbal juthatunk a következő tervező képernyőhöz.

Lehetőség van több bázisállomány létrehozására is az adott alkalmazásban, ezért az előző fejléccel ismét üres, bázisállományt létrehozó táblázatot kapunk. Miután a nyilvántartáshoz nincs szükség más kulcshivatkozásra, ezért a táblázatot üresen hagyva a RETURN gombbal lépünk a következő táblázatra.

Ezután a munkaindító kartont kell megtervezni, melynél nagyon sok kötöttséget kell szem előtt tartanunk. Ha bármelyiknek nem teszünk eleget, az alkalmazás felvitele hibás lesz, nem kapunk működő alkalmazást, legtöbbször már a generálás előkészítést sem tudjuk végrehajtani. Ennek leggyakoribb megjelenési formája, hogy a parancs kiadása után visszkapjuk a főmenüt. A hibakeresés első lépéseként a munkaindító kartont célszerű

megvizsgálni.

KARTOTEK NEVE [.]	
KOTELEZO MIN. ELOFORDULASA	[1 . . .]
LEHETSEGES MAX. ELOFORDULASA	[9 9 9 9]
FELHASZNALOI BASIC RUTIN I/N	[Y]
KARTOTEK TIPUSA N=ADAT I=SYS	[Y]
KEPERNYO KERÉTEZES I/N	[Y]
KEPERNYO MEGJELENITENDO I/N	[Y]
KARTOTEK BEMENETI FILE - JA	[.]
KARTOTEK BAZIS ALLOMANYA	[.]

3.4.4 ábra

Ha készítettünk egy jó munkaindító kartont, akkor azt célszerű többször elmenteni, és új alkalmazás tervezésnél ezzel indítani. Ekkor üzemmódnak nem első felvitelt, hanem forrás bővítést kell megadni.

A munkaindító kartoték nevének helyét végig ki kell pontozni, és a RETURN

gombbal lezárni.

A kartoték paramétereit célszerű az alábbi módon meghatározni, (lásd 3.4.4 ábrát)

Ennek a képernyőnek is ugyanaz a fejléce, hiszen még mindig a munkaindító karton tervezésénél tartunk. Ezután adhatjuk meg a karton konkrét formáját, képernyőn való megjelenítését (lásd 3.4.5).

V

KARTOTEK NEVE [.]

101 - 0 1

2 SZEMELYZETI NYILVANTARTAS : [.]

3 NYILVANTARTASI EV:

4 KEZELESI UZEMMOD :

5 ← :

6 KEZELOI JELSZO : [.]

7 MUNKAVEGZES KELTE (EE HHNN) : [.]

8 FELDOLGOZASI VALASZTEK : [.]

9 ← :

A

B

C

KARTON-MEZO MEGNEVEZESE [A D A T A Z O N]

SORSZAMA [0 1] TARTALMA [1] TIPUSA [1]

TART - - BEBILLENTYUZOTT

TIP - - BEVITT /TAROLT/ KIJELEZETT

TABLAELEM [_____]

3.4 fejezet

A legnagyobb kötöttség is itt adódik. A mezők megnevezése tetszőlegesen változhat, összefüggésben a feladat tartalmával, de a mezők hosszának és sorrendjének kötelezően így kell követniük egymást. Fontos megkülönböztetni a mezőnevet a mező belső azonosítójától. Míg a név a mi számunkra ad információt, és a karton így jelenik meg a képernyőn, a mező belső azonosítója a gép számára ad eligazítást. Ezért kötelező a munkaindító karton belső azonosítóinak mindig megegyezni a leírtakkal.

A fenti táblázatban, azaz a tervező képernyőn az alsó négy sor csak a karton megjelenítésének megtervezése után látható. A mező belső azonosítója 'karton-mező megnevezés'-ként szerepel, és 8 karakteres lehet.

Sorrendben a következő azonosítók szerepelnek:

ADATAZON

ADATVÁLT

ADATÜZEM

ÜZEMKIJE

JELSZÓ

MUNKAKLT

FELDÜZEM

FELDKIJE

A 'sorszám' megadása azért szükséges, hogy a mezőnevet és belső azonosítóját egymáshoz rendeljük.

A 'tartalom'-nál egy időlegesen látható almenüt kapunk:

1. bebillentyűzött

2. bebillentyűzött táblaelem

A típusnál

1. bevitt/tárolt/kijelzett

2. csak bevitt/kijelzett

lehetőségek látszanak a képernyőn. A már korábban ismertetett módon, ezek a kijelzések a <, > billentyűk alkalmazásával továbblapozhatók. Így újabb tartalmi, illetve típus változatok láthatóak.

Ha egy mezőhöz nem akarunk táblaelemet kapcsolni, /pl. ADATAZON, ADATVALT

akkor a legalsó sorba, a táblaelem zárójelébe '#' jelet kell beírni, majd RETURN-nal lezárni a kartonmező azonosítását.

```

KARTOTEK NEVE [ . . . . . ]
ELLENORZES MUVELET MEZONEV [AD Á T A Z Ö N ]
MEZOKITOLTESE KÖTELEZO [2] ← -IGÉN . . . . .
ELLENORZESI /MUVELET/ UTASITAS [15 ]
  ← LEGYEN TARTOMANYBAN . . . . .
ELSO OPERANDUS [A A A A A . . . . . ]
ELSO OPENANDUS TIPUSA [9 . ]
  ← KARAKTER KONSTANS . . . . .
2. OPERANDUS [Z Z Z Z Z . . . . . ]
2. OPENANDUS TIPUSA [9 . ]
  ← KARAKTER KONSTANS . . . . .

```

3.4.6 ábra

Az ADATUZEM mezőnek táblaelemeket is adunk. Az alkalmazás futtatásánál a

tervezett táblázat alatt ezek a táblaelemek fognak megjelenni, almenü formájában.

- 1.első felvitel
- 2.adatmódosítás
- 3.adatok feldolgozása
- 4.adatállomány kibővítése
- 5.munka befejezése

Ehhez a mezőhöz szorosan kapcsolódik az UZEMKIJE azonosítójú mező. A kapcsolat ugyanaz, mint a FELDUZEM és FELDKIJE mezők között, azaz a kijelző mezők tárolják a bebillentyűzött táblaelemek tartalmát. Az alapmezők ezeknek csak az azonosító kódját tartalmazzák.

Mint az alábbi táblázat mutatja az ellenőrző műveletek kijelölése következik (lásd 3.4.6 ábrát).

KARTOTEK NEVE [GYAR _____]	
KOTELEZO MIN.ELFORDULASA	[1 _____]
LEHETSEGES MAX.ELOFORDULASA	[1 0 _____]
FELHASZNALOI BASIC RUTIN <I/N>	[N]
KARTOTEK TIPUSA <N=ADAT I=SYS>	[N]
KEPERNYO KERETEZES <I/N>	[Y]
KEPERNYO MEGJELENITENDO <I/N>	[Y]
KARTOTEK BEMENETI FILE-JA	[_____]
KARTOTEK BAZIS-ALLOMANYA	[_____]

3.4.7 ábra

A munkaindító kartonnál minden mezőt ellenőrizni kell, az adatkartonnál elegendő minimálisan egy mezőre ellenőrzési utasítást kiadni.

A 'mező kitöltésére', a lehetséges 'ellenőrzési műveletekre', valamint az

'operandusok típusára' szintén időlegesen látható választékot kapunk. Ezekre az előttük álló számmal hivatkozhatunk.

```

KARTOTEK NEVE [GYAR _____]
101-01
2 GYAR NEVE : [ ]
3
4
5
6
7
8
9
A
B
C
KARTON-MEZO MEGNEVEZESE [GYAR NEVE]
SORSZAMA[01] TARTALMA [1] TIPUSA [1]
TART -- BEBILLENTYUZOTT
TIP -- BEVITT / TAROLT / KIJELEZETT
TABLAELEM[# ]
```

3.4.8 ábra

Ha minden ellenőrzési utasítást kiadtunk, akkor az újabb ellenőrzési műveletet váró képernyőt üresen hagyva, és a RETURN gombbal továbblapozva, ismét kartoték nevet kérő képernyőt kapunk. Itt már a felépített hierarchia szinteknek megfelelően dolgozhatunk. Az alacsonyabb szintek jelölésére a

név elé helyezett pontok száma utal. Azonos szintű kartotékok neve előtt azonos számú pont van.

A szögletes zárójelek kitöltésének, és a továbblépesnek a módja megegyezik a munkaindító kartonnál leírtakkal. A tervező képernyők megjelenésének sorrendje is ugyanaz (lásd 3.4.7, 3.4.8, 3.4.9, 3.4.10 ábrákat).

A dolgozók kartonján (lásd 3.4.11 ábrát) a családi állapotnál és az iskolai végzettségnél adtunk meg táblaelemeket.

Családi állapota lehet: nős
 nőtlen
 férjezett
 hajadon
 elvált

Iskolai végzettsége lehet:nyolc általános
 szakközépiskola
 gimnázium
 technikum
 főiskola
 egyetem

Ezekhez a táblaelem kódszámokat tartalmazó mezőkhöz tartoznak a CS. Áll.: és ISK.VÉGZ.: nevű, CSALKIJE, illetve ISKIJE azonosítójú mezők.

Ezek tartalmára, és típusára egyaránt 4-t kell írni, ami azt jelenti, hogy művelettel képzett tartalmú, képezett/kijelzett típusú mezők.

KARTOTEK NEVE [UZEM _____]	
KOTELEZO MIN.ELOFORDULASA	[1 _____]
LEHETSEGES MAX. ELOFORDULASA	[1 0 _____]
FELHASZNALOI BASIC RUTIN <I/N>	[N]
KARTOTEK TIPUSA <N=ADAT I=SYS>	[N]
KEPERNYO KERETEZES <I/N>	[Y]
KEPERNYO MEGJELENITENDO <I/N>	[Y]
KARTOTEK BEMENETI FILE-JA	[_____]
KARTOTEK BAZIS-ALLOMANYA	[_____]

3.4.9 ábra

Ha nem első felvitel üzemmódban vagyunk, akkor ezeknek az utasításoknak a hatására a kipontozott sorokban a kiválasztott lehetőség betűvel kiírva látható.

Ha az összes szintet elkészítettük, akkor az F6 funkciós billentyű megnyomásával visszkapjuk az első képernyőt.

KARTOTEK NEVE [. . DOLGOZO ADATAI	_____]
KOTELEZO MIN. ELOFORDULASA	[1 _____]
LEHETSEGES MAX. ELOFORDULASA	[1 0 0 0]
FELHASZNALOI BASIC RUTIN < I / N >]]
KARTOTEK TIPUSA < N=ADAT I=SYS >]]
KEPERNYO KERETEZES < I / N >]]
KEPERNYO MEGJELENITENDO < I / N >]]
KARTOTEK BEMENETI FILE-JA	[_____]
KARTOTEK BAZIS-ALLOMANYA	[SZ AK KE]

3.4.10 ábra

Ekkor az F6 ismételt leütésével a memóriából a mágneslemezre töltődnek a felépített struktúra lánctutatói. Enélkül sem a generálás, sem annak előkészítése nem fog működni.

Ezután kezelési üzemmódnak a 3.-t választva generálás előkészítést kérünk a rendszertől. Ennek folyamata jól követhető a képernyőn.

Ha a képernyőn ismét az első táblázat látható, és hibaüzenetet nem kaptunk, akkor a munka befejezésének jelzésével visszajuthatunk a főmenühez.

```

KARTOTEK NEVE [ . . . DOLGOZO ADATAI _____ ]
1 07-01
2 DOLGOZO NEVE:[ _____ ]
3 SZEMELYI SZAM: [ . . . . . ]
4 SZAKKEPZETTSEGE:[ _____ ]
5 LAKCIME: [ _____ ]
6 CSALADI ALL.: [ . ]
7 ISKOLAI VEGZETTSEGE [ . ]
8 CS. ALL: . . . . .
9 ISK. VEGZ: . . . . .
A
B
C
KARTON-MEZO MEGNEVEZESE [ C S A L A D I ]
SORSZAMA [ 0 5 ] TARTALMA [ 1 ] TIPUSA [ 1 ]
TART - - BEBILLENTYUZOTT TABLAELEM
TIP - - BEVITT / TAROLT / KIJELET
TABLAELEM [ NOS ]

```

3.4.11 ábra

A rendszerlemezt tesszük a meghajtó egységbe, majd a 4. pont segítségével generálást kérünk. A szükséges betöltés után lemezt cserélünk, erre

figyelmeztetést is kapunk a képernyőre, majd begépeljük az alkalmazás nevét.

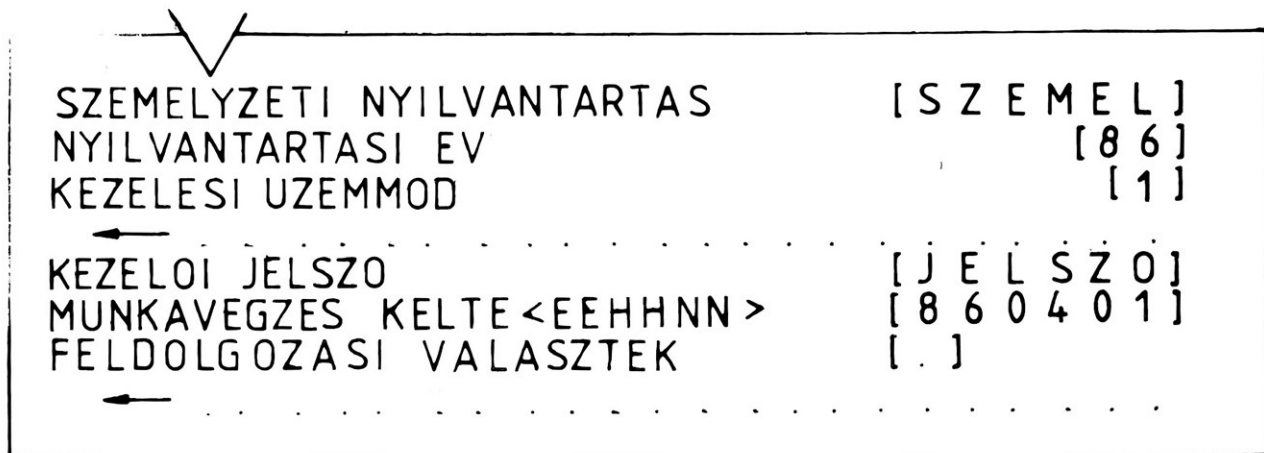
A négy menetű generálás folyamata szintén követhető a képernyőn.

Ha hibajelzést kapnánk a generálás után, akkor vissza kell térni az alkalmazás-tervezés módosítás üzemmódjához. A szükséges javítások elvégzése után lezárást, majd generálás előkészítést, és végül magát a generálást is ismételten el kell végezni.

Ha jó a generálás, akkor READY kiírást és villogó kurzort láthatunk a képernyőn. A 4. üzemmódból csak a RUNSTOP és RESTORE gombok egyidejű leütésével tudunk kiszállni, és így üres képernyőt kapunk. A főmenüt a POKE 51200-128,0 utasítással kapjuk vissza.

Az alkalmazás - futtatás utasítást adva a számítógépnek, betöltjük a rendszerlemezről az ehhez szükséges programrészletet.

.. következő képernyőt kapjuk, ha alkalmazás-névnek a SZEMEL hat karakteres azonosítót adtuk (lásd 3.4.12 ábrát):



The screenshot shows a terminal window with a title bar containing a checkmark icon. The window displays the following text:

```
SZEMELYZETI NYILVANTARTAS      [ S Z E M E L ]
NYILVANTARTASI EV             [ 8 6 ]
KEZELESI UZEMMOD              [ 1 ]
-----
← KEZELOI JELSZO                [ J E L S Z O ]
MUNKAVEGZES KELTE <EEHHNN >   [ 8 6 0 4 0 1 ]
FELDOLGOZASI VALASZTEK        [ . ]
-----
←
```

3.4.12 ábra

Ezt kitöltve kapjuk a 3.4.13 ábrát. A gyár és üzem nevének megadása után is RETURN-t kell ütni, hogy az alsóbb táblázatok is láthatóak legyenek.

```

GYAR NEVE [ MAGYAR ACEL _____ ]
UZEM [ ONTODE _____ ]
DOLGOZO NEVE : [ KOVACS JOZSEF _____ ]
SZEMELYI SZAM: [ 156 0127 4526 ]
SZAKKEPZETTSEGE [ ONTO _____ ]
LAKCIME : [ 1115 HENGERSOR U 148 ]
CSALADI ALL. : [ 1 ]
ISKOLAI VEGZETTSEGE [ 1 ]
CS. ALL. . . . .
ISK. VEGZ : . . . . .

```

3.4.13 ábra

Akár első felvitel, akár adatállomány kibővítés üzemmódot adtunk meg, tetszőleges, a maximálisnál nem nagyobb számú kartont hozhatunk létre egy-egy dobozban.

A minimálisan megadott számú kartont mindenképpen ki kell tölteni, addig

semmiképpen nem léphetünk át másik kartotékra.

Ha jól futó alkalmazást, illetve adatállományt hoztunk létre, akkor a munkalemezen a következő file-oknak kell szerepelniük:

SZEMNY.DSP.01 SZEMEL.DSP.86 SZAKKE.PR

SZEMNY.DSD.01 SZEMEL.DSD.86 SZAKKE.SE

SZEMNY.ASR.02 SZEMEL.BKG.86 SZAKKE.EX

SZEMNY.BKG.01

SZEMNY.ASR.01

SZEMNY.PAR.00

SZEMNY.FRT.00

3.5 A MASTER-64

A MASTER-64 egy a COMMODORE-64 számítógépen futó operációs rendszer és programozási rendszer, amely széles skálán nyújt segítséget a felhasználók számára. Előnyeit azok tudják legfőképpen értékelni, akik a C-64-en adatfeldolgozási feladatokat kívánnak megoldani. Természetesen más jellegű alkalmazások esetén is értékes software eszköz. Igazából túlmutat a C-64 lehetőségein, sokrétű tudásával nagyobb számítógépek software-ével is felveszi a versenyt. Ebből következően viszonylag nagy a memória foglalkalása, kb. 20 kByte a programok számára felhasználható tárterület. Így nagyobb programok csak egymást átlapolva tudnak futni.

A felhasználók olyan csoportjának javasolható az alkalmazása - ellentétben az EASY-FILE és a hasonló típusú előzőekben ismertetett adatfeldolgozási programcsomagokkal - akik már tudnak vagy fognak tudni BASIC-ben programozni.

A kézikönyv ezen része segítséget nyújt azoknak, akik a MASTER-64-et meg szeretnék ismerni (ehhez a mintapéldák sok segítséget adnak), de hasznos lehet mindazok számára is, akik már ismerik és használják. A BASIC programozás elsajátítására nem alkalmas, feltételezi a BASIC nyelv ismeretét.

A MASTER-64 a leghatékonyabb támogatást a file-kezelés terén adja a teljes vagy részleges egykulcsos index-szekvenciális (ISAM) adatelérés biztosításával. File kezelése tulajdonképpen meghaladja az ISAM lehetőségeit. Ezzel lehetővé teszi a felhasználó számára a gyors adatelérést, a kulcs szerinti rendezettséget, a rekordok könnyen programozható módosítását, törlését. Természetesen emellett alkalmazhatók a BASIC-ben megszokott szekvenciális és relatív file-szervezések is. Több mint 15 utasítást használhatunk az index-szekvenciális file-ok kezelése során.

A MASTER négy fő részre tagozódik:

- file-kezelés
- képernyőkezelés
- nyomtatás
- kiterjesztett BASIC

amelyeket külön-külön is tanulmányozhatunk és alkalmazhatunk.

MASTER - file-kezelés

Lehetővé teszi az ISAM állományokon rekordok beírását, kulcs vagy részkulcs alapján történő elérés után módosítását, törlését. Felírási sorrendben is elérhetjük a rekordokat ami ritkábban használatos, rendkívül gyors eljárás. Rendszerváltozók könnyítik meg a rekordkezelő utasítások használatát.

MASTER - képernyőkezelés

A képernyőgenerátor egyszerűsíti a felhasználó számára a képernyő programozását. Az adatkijelzés és bevétel a képernyőről közvetlenül, vagy előzetesen definiált zónákban történik oly módon, hogy megadjuk a képernyőpozíció X,Y (sor és oszlop) koordinátáját. Képernyő input és output esetén is megadható mezőhossz, típus (numerikus, alfás, alfanumerikus) és mező formátum. Input esetén ezek a mező attributumok egyben adatellenőrzési célt is szolgálnak.

MASTER - nyomtatás

Ugyanolyan lehetőségekkel rendelkezik, mint a képernyőgenerátor. Ahhoz hasonlóan itt is tárolhatók a maszkok a floppy-n, amivel fejlesztési és futási idő takarítható meg.

MASTER - BASIC bővítés

- Adattömörítési és zónázási lehetőségek állnak rendelkezésre a mágneslemezes tároló maximális kihasználása érdekében. Az adatsűrítési funkció elsősorban a file-kezelés során lényeges.
- Többszörös pontosságú műveletek (22 jegy pontosságig +, -, ,/).
- Rugalmas 2 byte-os dátumkezelés, érvényesség ellenőrzés.
- Számított GOTO és GOSUB, IF ... THEN ... ELSE utasítások.
- Mágneslemez leegyszerűsített közvetlen elérése.
- Karakterorozaton belüli karakterkeresés.
- BASIC 4.0 bővítés és a DOS utasításai.
- EDEX bővítés (15 új utasítás, TRACE nyomkövetési lehetőség).

3.5.1 A MASTER-64 eszközigénye

1. Commodore 64 személyi számítógép.
2. Egy vagy több Commodore 1541 lemez meghajtóegység és csatlakozó.
3. Sornyomtató a hozzátartozó interface-szel
(CBM 1515, 1526, 801, 802)
4. Televízió vagy monitor a megfelelő csatlakozóval.
5. Hajlékony mágneslemezek.

Az RS232 interface-en keresztül más típusú nyomtatók is csatlakoztathatók konfigurációnkba.

A MASTER-64 rendszerlemez tartalma:

3	loaderd	pgm	---	a fejlesztő rendszer betöltője
70	master64I13d-a	pgm	---	a fejlesztő rendszer program file
54	master64I13d-b	pgm	---	a fejlesztő rendszer program file
7	uti64I13d-a	pgm	---	a fejlesztő rendszer utility funkciói
7	uti64I13d-b	pgm	---	a fejlesztő rendszer utility funkciói
3	loaderu	pgm	---	a futtató rendszer betöltője

70	master64I13u-a	pgm	---	a futtató rendszer program file
54	master64I13u-b	pgm	---	a futtató rendszer program file
7	uti64I13u-a	pgm	---	a futtató rendszer utility funkciói
7	uti64I13u-b	pgm	---	a futtató rendszer utility funkciói
1	boot	pgm	---	az automatikusan betöltődő program
30	reserve file	pgm	---	az indexelt file-t létrehozó segédprogram
1	reserve 1	pgm	---	az indexelt file előállítás, mintaprogram
24	reserve 2	pgm	---	a programozott előállítás pr. file
24	regekey	pgm	---	index file regeneráló segédprogram
45	copy file	pgm	---	indexelt file másoló segédprogram
10	backup 1541	pgm	---	teljes lemez másolása VC 1541-en
8	demoprg	pgm	---	bemutató program ISAM file kezeléshez
5	demoscreen	pgm	---	bemutató program a képernyőgenerátorhoz
37	page64	pgm	---	bemutató progr. nagyfelbontású képhez
21	proscreen 64	pgm	---	maszk generátor
8	help	pgm	---	maszk generátor help-je
44	test.master	pgm	---	MASTER-64 teszt és demonstráció

A futtató rendszerben nem fejleszthetünk programokat. Ez egyrészt gyorsabb futást eredményez, másrészt a programok védelmét segíti elő.

3.5.2 MASTER 64 utasítások

KÉPERNYŐ-GENERÁTOR

Áttekintés

Képernyő-generátor

Jelölésrendszer

- tline - vízszintes szakasz rajzolása
- tcol - függőleges szakasz rajzolása
- clear - sor-részlet törlése
- out - kiíratás képernyőre
- scroll - ablak görgetése lefelé, vagy felfelé
- rev - ablak inverzió

Képernyő-mezőkezelés

- decz - zóna definiálás
- reqz - zóna adatkérés
- inz - zóna input változóba
- outz - zóna output változóból
- clearz - zóna törlése
- revz - zóna inverz megjelenítés

Képernyőlap kezelés

- sset - képernyőlap kiválasztás
- scopy - képernyőlap bekérés memóriából
- sexch - képernyőlap kivitel memóriába
- sclear - képernyőlap törlése
- sreset - képernyőlap elengedés
- ssave - képernyőlap kimentése
- sload - képernyőlap visszatöltés

NYOMTATÓ-GENERÁTOR

Áttekintés

pcreate - lap definiálása
popen - lap megnyitása
pclose - lap lezárása
pdecz - mező definiálása
pout - karaktersorozat kiíratása
pclear - sor egy részének törlése
pclearz - mező törlése
perase - teljes lap törlése
pprint - lap nyomtatása

FILE-KEZELÉS

Áttekintés

File-definiálás

Optimalizálás

Indexelt file-kezelő utasítások

iopen - megnyitás
iwrite - új rekord írása
iadd - indexelés nélküli írás
iread - teljes vagy részkulcs szerinti olvasás
iexist - rekord egzisztencia vizsgálat
idelete - rekord törlése
iupdate - rekord módosítás
istart - pozícionálás szekvenciális olvasás előtt
inext - következő rekord olvasása index sorrendben
ivalidate - az iadd-al betoldott rekordok indexelése
irestore - a file olvasása beírási sorrendben
idata - az iadd-al betoldott rekordok olvasása
iclose - lezárás
iupgrade - floppy-ra írja az indextáblát
onerror - floppy fizikai hiba lekezelése

ireset - ISAM inicializálás
Rendszer státuszváltozók

TÖMÖRÍTÉS-ZÓNÁZÁS

Áttekintés

Tömörítési típusok

Alfanumerikus

Bináris

Pozitív egészek

Lebegőpontos számok

Tömörítés nélküli típus

Tömörítési utasítások

creatst - üres sztring előállítása

putst - rekord tömörítése

getst - rekord zónázás (ritkítás)

Kulcs tömörítése

BASIC-BŐVÍTÉSEK

Többszörös pontosságú műveletek

madd - összeadás

msub - kivonás

mmul - szorzás

mdiv - osztás

Számított ugró utasítások

goto -

gosub -

Kisbetű-nagybetű átalakítás

uplow -

Leegyszerűsített közvetlen floppy elérés

putbuf - sztringet ír a pufferba

putblock - a sztringet floppy-ra írja

inblock - sztringet ír a pufferba a floppy-ról

takbuf - változóba írja a sztringet

Listavédelem

nolist - floppy-ra mentés, tovább nem listázható

Dátumtömörítés, ellenőrzés

Áttekintés

datepack - tömörítés

dateunpack- zónázás (ritkítés)

Karakterkeresés

hunt - karakter keresése sztringben

TOVÁBBI BŐVÍTÉSEK

EDEX bővítés (programfejlesztési eszközök)

Áttekintés

auto - automatikus sorszámozás

delete - programsorok törlése

renu - program újrasorszámozás

dump - változók értékeit listázza

find - karaktersorozat keresése a programban

error - hiba esetén jelzi a hiba helyét

hardcopy - az aktuális képernyőt nyomtatja ki

pusing - formattált output

beep - beépített síp

if ... then ... else ...

trace - nyomkövetés utasításonként

screen - képernyőszínek definiálása

MASTER-MELLÉKLETEK

Tömörítési táblázatok

ISAM file definiáló képernyők

Személyzeti nyilvántartás MASTER-64-ben

3.5.3 A MASTER-64 részletes ismertetése

A MASTER-64 hajlékony mágneslemezes adathordozón helyezkedik el. A MASTER-64 betöltésekor helyezük rendszerlemezünket a VC-1541 meghajtóegységbe.

```
LOAD"loaderd",8,1
SEARCHING FOR loaderd
LOADING
```

Ez után kb. 1.5 percig villog a képernyőn a MASTER-64 töltő képernyője, majd a következő felirat jelenik meg:

```
*** master 64 v 1.1 ***
(c) micro application
ready

dL"bo*
searching for 0:bo*
loading
ready
rU
```

Ezzel gyakorlatilag betöltődött a rendszer a BOOT nevű programmal együtt. Ha a BOOT programunkban elhelyeztük valamely programunk indítását, akkor az automatikusan betöltődik és elindul.

```
P1. 10 rem --- boot program ---
    20 dload"prnev"
    30 rem -----
```

A MASTER-64 és a BASIC 4.0

A MASTER-64-ben beépített utasításokként megtalálhatjuk a BASIC 4.0 utasításait. Minden MASTER-64 utasítás token (1 byte) formában tárolódik. Ez egyben azt is jelenti, hogy rövid formában is megadhatók az utasítások, ha a második vagy harmadik betűt SHIFT-el együtt adjuk be.

Óvatosan kell azonban alkalmazni ezt a könnyítést, mert az utasítások száma megnőtt. Azonosság esetén a MASTER-64 utasítások előnyben részesülnek a többivel szemben, azaz másként tokenizál a C-64, mint a MASTER-64. Ez azt jelenti, hogy az sC nem a SCRATCH, hanem a SCROLL utasításként értékelődik ki. A SCRATCH-ot az scrA-val rövidíthetjük.

Nem MASTER-64 alatt írt program futtatása esetén hibákat fogunk tapasztalni. Ilyenkor a MASTER-64 alatt RETURN-nel menjünk végig a hibás sorokon.

Képernyő-generátor

Áttekintés

A képernyőgenerátor lényegesen lerövidíti a képernyő és a billentyűzet adatforgalmának programozását. Mivel gépi nyelven írt rutinokon keresztül valósulnak meg az utasítások, ezért sokkal gyorsabb is a programunk, mint ha BASIC rutinokat használnánk ugyanerre a célra.

A MASTER-64 az alábbi funkciókat támogatja:

- Függőleges és vízszintes vonalakat rajzoló, képernyőrészleteket görgető utasítások
- Képernyőzónák definiálása ellenőrzött adatkiviteli és beviteli lehetőséggel
- Teljes képernyőoldalt kezelő utasítások az aktív képernyő, a memória képernyő és diszken lévő képernyőterületek között.

Jelölésrendszer

A későbbiek során a szintaktikai leírásokban használt változónevek ismétlődnek. Ez azonban nem jelenti azt, hogy nem használhatunk más változóneveket a programunkban. Kivételt képeznek ez alól a rendszerváltozók, pl. a ZO,NR,stb...

A következő változók rendszeresen előfordulnak:

- s - a sor száma
- o - az oszlop száma
- lv - vízszintes vonal szélessége
- lf - függőleges vonal magassága

A MASTER-64-ben a képernyő koordinátarendszerének origója a bal felső sarokban van, koordinátája (1,1).

A 25 sor és 40 oszloppal rendelkező képernyőnk minden pontja a sor és az oszlop megadásával azonosítható.

Ha ugyanazt az utasítást egy programsorban más paraméterekkel többször használjuk akkor rövidíthetjük a balra mutató nyíl billentyű segítségével:

pl. tline 10,2,1: - 10,4,1

Az aktuális képernyő nem más, mint ami a képernyőn egy adott pillanatban látható. Ezen kereteket, szegélyeket állíthatunk elő, a teljes képernyőt kivihetjük diszkre, betölthetjük a képernyőre, vagy a memóriába. A képernyőmezőkben adatokat jeleníthetünk meg, vagy tölthetünk ki.

Képernyő-kijelzési utasítások

T L I N E

Vízszintes vonalat rajzol a képernyőn, amelyet kezdő koordinátái és hossza ad meg.

Szintaktika: `tline lv,s,o`

ahol:

`lv` - a szélesség

`s` - a sor száma

`o` - az oszlop száma

Például:

`tline 40,1,1 : tline 40,10,1`

Az utasítás két párhuzamos vonalat húz a képernyőre.

T C O L

Függőleges vonalat rajzol a képernyőn, amelyet kezdő koordinátái és hossza ad meg.

Szintaktika: `tcol lf,s,o`

ahol:

`lf` - a magasság

`s` - a sor száma

`o` - a oszlop száma

Például:

`tcol 10,1,1 : tline 10,1,10`

Az utasítás két párhuzamos vonalat húz a képernyőre.

`10 tline 40,1,1 : tline 40,10,1`

`20 tcol 10,1,1 : tcol 10,1,10`

Az utasítások keretet rajzolnak a képernyőre. A MASTER-64 automatikusan elhelyezi a metszéspontokra a lezáró karaktereket.

C L E A R

Meghatározott pozíciótól kezdve törli a képernyő egy vízszintes szakaszát.

Szintaktika: `clear lv,s,o`

ahol:

`lv` - a szélesség

`s` - a sor száma

`o` - a oszlop száma

Például:

```
clear 40,1,1
```

Az utasítás törli a képernyő legfelső sorát.

O U T

Egy sztring változót ír ki a képernyőre a megadott pozíciótól kezdődően.

Szintaktika: `out a$,s,o`

ahol:

`a$` - tetszőleges sztring kifejezés

`s` - a bal felső pont sorszáma

`o` - a bal felső pont oszlopszáma

Például:

```
10 a$="Kiss Barnabás"
```

```
20 out a$,3,20
```

Az utasítás a 3. sor 20. oszlopába kiírja a "Kiss Barnabás" szöveget.

S C R O L L

Képernyő mezőt görget felfelé vagy lefelé, amelyet bal felső pontja, hossza és mélysége határoz meg.

Szintaktika: scroll s,o,lf,lv,tip

ahol:

- s - a bal felső pont sor száma
- o - a bal felső pont oszlop száma
- lf - a magasság (sorok száma)
- lv - a szélesség (oszlopok száma)
- tip- az irány típusa
 - u - felfelé
 - d - lefelé

R E V

Képernyő ablakot fordít inverzbe.

Szintaktika: rev s,o,lf,lv

ahol:

- s - a bal felső pont sor száma
- o - a bal felső pont oszlop száma
- lf - a magasság (sorok száma)
- lv - a szélesség (oszlopok száma)

Például:

rev 1,1,10,20

Az utasítás a bal felső sarokban inverzbe vált egy 20 oszlop széles 10 sor magas ablakot.

Képernyő-mezőkezelés

Áttekintés

A MASTER-64 mező vagy zónakezelő utasításai a BASIC INPUT és GET utasításait teszik fölöslegessé. Ezek egységes, könnyen ellenőrizhető, megbízható és hatékony input-output kezelést biztosítanak számunkra.

Először is kezdjük az ismerkedést az alapelvekkel. A képernyőn egy adatmezőt, más néven zónát paraméterei megadásával definiálhatunk (DECZ). Ilyenek a képernyőmező koordinátája, hossza, megengedett karakterek a zónán belül, a zónából való kilépés lehetőségei a RETURN-on kívül, stb.

A definiálás után adatbekérést kezdeményezhetünk a zónából (REQZ). A zónán belül használhatjuk a kurzor billentyűket és a törlő-beszűrő INST/DEL billentyűt. Az adatot az INZ utasítás viszi át a megnevezett sztring változónkba. Ugyanezzel a zónadefinícióval kiírást is megvalósíthatunk az OUTZ paranccsal. Adatok karbantartásánál nagyon kényelmes a már meglévő adattartalmat a definiált zónába kiíratni az OUTZ-vel, majd a REQZ-INZ utasításpárral módosítani azt. Egy képernyőn maximum 128 különböző zónát definiálhatunk.

```
BASIC      MASTER-64
           decz 1 --- zóna def.
input a$   reqz 1 --- adatbekérés
           inz  1 --- adatátvitel az a$ változóba
```

D E C Z

Zóna definiálás kiíratáshoz és adatbevitelhez.

Szintaktika: decz n,s,o,lz,(,tip)(,f\$)

ahol:

```
n - a zóna sorszáma (változó is lehet)
s - a sor száma
o - az oszlop száma
```


- lz - a zóna hossza
- tip- a zóna típusa
- tip=n - numerikus adatok
 - tip=m - csak nagy betűk
 - tip=r - kilépés a zónából a RETURN-on kívül más
 - numerikus zóna esetén bármely nem numerikus billentyűvel
 - alfabetikus zóna esetén (SHIFT/RETURN), (RVS/OFF), (RUN/STOP), (CRSR), (TAB)
 - a fenti billentyűk ASCII kódja a ZO nevű rendszerváltozón keresztül kérdezhető le ,így programozhatjuk a kilépést is
 - tip=p - formátum hozzárendelés az f\$ sztringgel
 Az f\$-ban tárolt ellenőrző karakterek a következők:
 - 9 - numerikus
 - 8 - numerikus vezető nullákkal
 - 1 - előjel pozíció (csak -)
 - 2 - előjel pozíció (+ és -)
 Továbbá tetszőleges alfanumerikus karaktereket használhatunk pl. f\$="Ft19,988,88".

Például:

```
decz 2,8,10, 9,n,p,"199998.88"
```

Az utasítás a 2-es sorszámú zónát definiálja, amely a 8. sor 10. oszlopán kezdődik, és 9 karakter hosszú. Csak numerikus értékeket fogadunk el a pozitív előjel elnyomásával. Az utolsó 3 pozíció nullái megmaradnak. Egyszerre több típust is megadhatunk egy adott zónánál, természetesen csak az egymást nem kizárókat.

Megjegyzés:

Ha adatbevitelnél formátum túlcsoportulásra kerül sor, akkor az egész zóna törlődik, és a kurzor visszatér a zóna elejére. A formátum ellenőrzése a

kilépő billentyű lenyomásával aktivizálódik. A mező teljes hosszában inverzben jelenik meg a képernyőn.

R E Q Z

Adatbekérés a definiált zónából.

Szintaktika: reqz n

ahol:

n - a zóna sorszáma

Például:

reqz 2

Az utasítás a 2-es zónából kezdeményezi az adatbekérést.

I N Z

Egy sztring változóba viszi át a reqz-vel bekért adatot.

Szintaktika: inz n,a\$

ahol:

n - a zóna sorszáma

a\$ - tetszőleges sztring változó

Például:

reqz 2,bb\$

Az utasítás a 2-es zónából bekért adatot átviszi a bb\$ nevű változóba.

A változó hossza aktuális adattartalmától függetlenül azonos a zóna hosszával.

O U T Z

Egy sztring tartalmát írja ki a zónába.

Szintaktika: outz n,a\$

ahol:

n - a zóna sorszáma

$a\$$ - a tetszőleges sztring változó ,vagy kifejezés

Például:

```
bc$="1234" : outz 2,bc$
```

Az outz használata egyszerűbb az out-nál, azonkívül hasznosabb is, mivel formátumot is megadhatunk. A tizedespont utáni jegyeket a rendszer levágja, túlcsoordulás esetén csillagokkal tölti ki a mezőt.

C L E A R Z

Töröl egy zónát.

Szintaktika: `clearz n1 (,n2 to n3) (...)`

ahol:

n_i - a zóna sorszámok

Például:

```
clearz 2,5 to 13
```

Az utasítás törli a 2-es, és az 5-östől a 13-asig terjedő sorszámú zónákat Új adatok kérése előtt célszerű kiadni.

R E V Z

A zóna képernyő-területét inverzbe váltja.

Szintaktika: `revz n1 (,n2 to n3) (...)`

ahol:

n_i - a zóna sorszámok

Például:

```
revz 10,15 to 18
```

Az utasítás inverzbe váltja a 10-es, és a 15-östől a 18-asig terjedő zónákat. Felhívja a felhasználó figyelmét a zóna területére. Lehetőségünk van a zóna villogtatására, lásd REV.

Képernyőlap-kezelés

Áttekintés

A MASTER-64 segítségével definiált képernyőlapot egész egységként kezelhetjük. Kivihetjük a floppyra, bármikor lecserélhetjük az aktuális képernyőnköt egy másikra, pl. a saját programunkhoz HELP funkciót megvalósító lapok töltésekor.

S S E T

Aktívvá tesz egy képernyőlapot.

Szintaktika: sset
 sset sn

ahol:

sn - a képernyőlap sorszáma

Például:

```
sset 12
sclear
tline 40,1,1 : tline 40,10,1
tcol 10,1,1 : tcol 10,1,40
sset
```

Az utasításokkal a 12-es lapon keretet rajzoltunk fel, miközben a képernyőnkön nem történik semmi.

S C O P Y

A memóriából kihaz a képernyőre egy képernyőlapot.

Szintaktika: scopy

Például:

```
sset 12
scopy
```

Az utasításokkal a 12-es lapon lévő keretet jelenítjük meg a képernyőnkön.

S E X C H

Az aktuális képernyőlapot átmenetileg felcseréli a memóriában lévő lappal. Rendkívül hasznos saját HELP lapok megjelenítéséhez.

Szintaktika: sexch

Például:

```

sset
out "Sajat programom",10,1
sset 12
sexch

```

Az utasításokkal a 12-es lapon lévő keretre cseréltük fel az aktuális lapunkat.

S C L E A R

Töröl egy képernyőlapot a képernyőn vagy a memóriában. (Meghagyja a zóna-definíciókat).

Szintaktika: sclear

Például:

```

sset 12
sclear

```

Az utasításokkal a 12-es lapon mindent letöröltünk.

S R E S E T

Megszünteti a képernyőlap memóriafoglalását.

Szintaktika: sreset sn

ahol:

sn - a képernyőlap sorszáma

Például:

```
sclear   : print fre(0)
sreset 12: print fre(0)
```

Az utasításokkal a 12-es lapot letöröltük, amitől még nem tűnt el, majd megszüntettük. A felszabadult memória 1000 byte kell, hogy legyen.

S S A V E

Floppyra ír egy képernyőlapot.

Szintaktika: `ssave u,"d: név" (,n1)(,n2 to n3)...`

ahol:

```
u - a lemezegység száma (8)
d - a meghajtó száma (0,1)
ni - a képernyőlapon lévő zónák sorszáma
```

Például:

```
ssave 8,"0:egykes-kép",1,3,5 to 18,21
```

Az utasítás floppyra teszi a teljes képernyőt a megnevezett zónákkal együtt.

S L O A D

Betölt egy képernyőlapot a mágneslemezebről.

Szintaktika: `sload u,"d: név"`

ahol:

```
u - a lemezegység száma (8)
d - a meghajtó száma (0,1)
```

Például:

```
sload 8,"0:egykes-kep"
```

Az utasítás betölti a floppyról a kivitt képernyőt a zónákkal együtt.

NYOMTATÓ-GENERÁTOR

Áttekintés

A nyomtató-generátor felhasználása ritkán indokolt, általában elég a BASIC PRINT és a MASTER-64 PUSING utasítása, ezért csak felsoroljuk az utasításokat. Szintaktikájuk hasonlít a képernyő-generátor utasításainak szintaktikájához.

A lap definiálása	- pcreate 8,"0:lapnev",sorok,fejsor,labsor,balmargo
A lap megnyitása	- popen 8,"0:lapnev"
A lap lezárása	- pclose
Mező definiálás	- pdecz n,s,o,lz,(tip)(,f\$)
Karaktorsorozat kiírása	- pout a\$,s,o
Mező kiírása	- poutz n,a\$
Sor részének törlése	- pclear lv,s,o
Mező törlése	- pclearz n1 (n2 to n3) (...)
A teljes lap törlése	- perase
A lap nyomtatása	- pprint

FILE-KEZELÉS

Áttekintés

Az ISAM file-kezelés a MASTER-64 legfontosabb eszköze. Sokkal több lehetőséget ad, mint a hagyományos ISAM (Indexed Sequential Access Method). Gyors működése, megbízhatósága ideális programozási eszközzé teszi a MASTER-64-et.

File definiálása

A file definiálását a RESERVE FILE nevű programmal hajthatjuk végre. Meghatározhatjuk a rekordhosszt, és a kulcs pozícióját, hosszát. A MASTER-64 létrehozza a floppyn az indexelt file kezeléséhez szükséges file-okat. Nem a teljes méretre foglalja le a file-okat, a rekordok felírásakor állandóan bővíti a file-t.

Az állományok felhasználása során az index file-ok a rekord felírásakor, módosításkor, törléskor automatikusan karbantartódnak. Például új rekord írásakor a file kulcs sorrendben rendeződik.

Hat különböző módon érhetjük el rekordjainkat:

- teljes kulcson keresztül
- részkulcson keresztül
- kiválasztó maszkon keresztül
- következő vagy előző kulcs sorrendben
- következő vagy előző kulcs sorrendben kiegészítő maszkkal
- felírási sorrendben az indexeléstől függetlenül

A floppyn lévő indextáblázat bármely oknál fogva megsérülhet, pl. már felvittünk egy rekordot a file-ba, de az indextábla nem íródott vissza a memóriából áramkimaradás miatt. Ilyenkor használjuk a REGEKEY nevű index regeneráló segédprogramot.

A RESERVE FILE program használatát legegyszerűbben egy konkrét példán

keresztül mutathatjuk be:

A fejezet végén lévő mintapéldában a személyi állomány indexelt file-ja 124 byte hosszú, a személyi szám kulcs az első pozíción 11 byte hosszon helyezkedik el. A mellékletben megtalálhatjuk ezen file definiáló-képernyőjét, amelyet a RESERVE FILE futtatása során nyerünk.

A RESERVE FILE kérdései:

File név:

Max. 14 hosszú tetszőleges név adható meg.

Rekordok tároló-eszköz száma:

VC-1541 esetében 0-t adjunk meg.

Indexek tároló-eszköz száma:

VC-1541 esetében 0-t adjunk meg, elvileg más egységen is elhelyezhetjük az index file-okat.

Level (index-szint):

Az index-szint határozza meg az indextábla nagyságát a memóriában, az elérési sebességre is van hatása. Ha 2-es szintet adunk meg, akkor az index-tábla két részre oszlik. Egyik része memória rezidens, a másik a floppyn helyezkedik el. Az 1-es szint esetében a teljes index-tábla a memóriában lesz, így gyorsabb az indexkezelés nagyobb memória foglalás mellett. A példában 1-es szintet választottunk.

A rekordok maximális száma:

Előre meg kell becsülnünk a rekordok várható számát, példánkban 1000-et adunk meg. Ez az adat hatással van az index tábla mértére.

Rekordhossz:

Itt adjuk meg a rekordunk hosszát, amely nem lehet nagyobb 254-nél. Ebben benne foglaltatik a kulcs hossza is. Jelen esetben a rekordunk hossza 124.

Kulcshossz:

A kulcs hossza 2-től 30 byte-ig terjedhet. A példában 11-et választunk.

Kulcs pozíció:

Legtöbb esetben, mint a példában is a kulcsot a rekord elejére tesszük. Tehát 1-es a válasz .

A kulcs első byte-ja 250-nél kisebb ASCII kódú kell hogy legyen. Az adatok megadása után még módosíthatjuk őket, ha a következő kérdésre (OK ?) "n"-nel válaszolunk. Ha minden adatunk megfelelő, akkor a RESERVE FILE megkérdezi, hogy létre akarjuk-e hozni a file-okat. Erre válaszoljunk "i"-t, ha minden rendben van. Német vagy angol nyelven beszélő programok esetében a válasz "j" vagy "y".

A program a következő négy adatot maga számolja ki:

- indextábla mérete a memóriában (jelen esetben 1295 byte)
- az adat blokkok száma a floppyn, amelyet index nélkül foglalnak le (jelen esetben 501)
- index blokkok száma a mágneslemezen (jelen esetben 86)
- az index és adatblokkok összege; ennek akkor van nagy jelentősége ha mindkettőt ugyanazon a mágneslemezen tároljuk (jelen esetben 592, még éppen elfér a floppyn)

Optimalizálás

Rekordméret optimalizálás:

A rekordok nem nyúlhatnak át szektorhatárokon. Ezt a tényt figyelembe kell vennünk a rekordméret meghatározásakor. Tegyük fel, hogy 32 byte-os rekordjaink vannak. Mivel a szektorméret 254 byte, így $\text{INT}(254/32) = 7$ darab rekord fér el egy szektorban, a fennmaradó 30 byte kihasználatlanul marad. Ebben az esetben 11.8 százalék a veszteségünk. Ha 31 byte-os rekordméretet választunk, akkor a veszteség 2.6 százalék lesz. Minél nagyobb a rekord mérete, annál inkább figyelembe kell venni ezt a feltételt. Mintapéldánkban nem optimalizáltuk a rekordhosszt.

Index tábla optimalizálás

Az index tábla méretét is hasonlóképpen optimalizálhatjuk, csak a kulcshosszhoz kell 3 byte-ot hozzáadnunk.

DOS kompatibilitás a floppyn

A MASTER file-ok kompatibilisak a DOS-al, ezért indexelt file-okon kívül szekvenciális és relatív file-okat is elhelyezhetünk ugyanazon a floppyn. A SCRATCH utasítás indexelt file-okat is töröl. A COLLECT és a VALIDATE utasításokat ne használjuk! A COPY utasítás nem másol indexelt file-okat, ehhez egy segédprogram a COPY FILE áll rendelkezésre.

Indexelt-file kezelő utasítások

I O P E N

Indexelt file-t nyit meg.

Szintaktika: iopen lf,u,"d:filenév"

ahol:

lf - logikai file szám (0-9)
 u - lemezegység száma (ált. 8)
 d - a meghajtó száma (0 v.1, ált. 0)

Például:

```
10 rem --- file megnyitás ---
20 iopen 2,ui,"0:szemadat"
30 if ok=128 then:out"Sikertelen megnyitás",1,1
..
```

Megnyitjuk a 2-es logikai számú SZEMADAT nevű indexelt file-t. Ha sikertelen volt a megnyitás, akkor hibaüzenetet írunk ki.

I W R I T E

Új rekord beírása, egyidejűleg az indextábla is változik az új kulcsnak megfelelően.

Szintaktika: iwrite lf,rc\$

ahol:

lf - logikai file szám (0-9)
 rc\$- a teljes rekord a kulccsal együtt

Például:

```
100 rem --- file megnyitás ---
200 iopen 2,ui,"0:szemadat"
300 if ok=128 then:out"Sikertelen megnyitás",1,1
...
700 rc$=ra$+rb$+rc$
```

```

800 iwrite 2,rc$
810 if ok=1 then:out"Már létezik",24,1: ...
1030 if ok=64 then out"A file megtelt!",24,1:goto ...
....

```

Megnyitjuk a 2-es logikai számú SZEMADAT nevű indexelt file-t. Beírunk egy új rekordot. Ha sikertelen volt az írás,akkor hibaüzenetet küldünk.

I A D D

Új rekord beírása az indextábla változása nélkül. Gyors adatbevitelt tesz lehetővé. Az indextáblát az IVALIDATE utasítással aktualizálhatjuk.

Szintaktika: iadd lf,rc\$

ahol:

lf - logikai file szám (0-9)
rc\$- a teljes rekord a kulccsal együtt

Például:

```

100 rem --- file megnyitás ---
200 iopen 2,ui,"0:szemadat"
300 if ok=128 then:out"Sikertelen megnyitás",1,1
...
700 rc$=ra$+rb$+rc$
800 iadd 2,rc$
810 if ok=64 then out"A file megtelt!",24,1: goto ...
...

```

Megnyitjuk a 2-es logikai számú SZEMADAT nevű indexelt file-t. Beírunk egy új rekordot. Ha sikertelen volt az írás akkor hibaüzenetet küldünk.

I R E A D

Rekord olvasása teljes vagy részkulcs felhasználásával.

Szintaktika: iread lf,ku\$,rc\$

ahol:

lf - logikai file szám (0-9)
 ku\$- a teljes vagy részleges kulcs
 rc\$- a teljes rekord a kulccsal együtt

Például:

```
100 rem --- file megnyitás ---
200 iopen 2,ui,"0:szemadat"
300 if ok=128 then:out"Sikertelen megnyitás",1,1
...
700 ku$="KISS JANOS"
800 iread 2,ku$,rc$
810 if ok=1 or ok=255 then:out"A rekord nem létezik",24,1
...
```

Megnyitjuk a 2-es logikai számú SZEMADAT nevű indexelt file-t. Olvasunk egy rekordot. Ha sikertelen volt az olvasás akkor hibaüzenetet küldünk.

I E X I S T

Ellenőrzi egy rekord létezését.

Szintaktika: iexist lf,ku\$

ahol:

lf - logikai file szám (0-9)
 ku\$- teljes vagy részleges kulcs

Például:

```
100 rem --- file megnyitás ---
200 iopen 2,ui,"0:szemadat"
300 if ok=128 then:out"Sikertelen megnyitás",1,1
...
700 kc$="KISS"
800 iexist 2,kc$
810 if ok=1 or ok=255 then:out"A rekord nem létezik",24,1
...
```

Megnyitjuk a 2-es logikai számú SZEMADAT nevű indexelt file-t. Ellenőrizzük a rekord létezését. Ha nem létezik, akkor üzenetet küldünk.

I D E L E T E

Töröl egy rekordot.

Szintaktika: idelete lf,ku\$

ahol:

lf - logikai file szám (0-9)

ku\$- teljes kulcs

Például:

```
100 rem --- file megnyitás ---
```

```
200 iopen 2,8,"0:szemadat"
```

```
300 if ok=128 then:out"Sikertelen megnyitás",1,1
```

```
...
```

```
700 kc$="KISS JANOS      "
```

```
800 iexist 2,kc$
```

```
810 if ok=1 or ok=255 then:out"A rekord nem létezik",24,1
```

```
...
```

Megnyitjuk a 2-es logikai számú SZEMADAT nevű indexelt file-t. Töröljük a rekordot. Ha nem létezik, akkor üzenetet küldünk.

I U P D A T E

Egy már meglévő rekordot módosít.

Szintaktika: iupdate lf,rc\$

ahol:

lf - logikai file szám (0-9)

rc\$- a teljes rekord

Például:

```
100 rem --- file megnyitás ---
```

```
200 iopen 2,ui,"0:szemadat"
```

```
300 if ok=128 then:out"Sikertelen megnyitás",1,1
```

```
...
```

```
700 rem --- modositas ----  
800 iupdate 2,rc$  
810 if ok=1 or ok=255 then:out"A rekord nem létezik",24,1  
...
```

Megnyitjuk a 2-es logikai számú SZEMADAT nevű indexelt file-t. Módosítjuk a rekordot. Ha nem létezik, akkor hibaüzenetet küldünk.

Olvasás növekvő vagy csökkenő sorrendben

Az ISTART és az INEXT utasítások segítségével növekvő illetve csökkenő kulcssorrendben olvashatjuk a file-t. Az ISTART-al megadjuk az olvasás kezdőpozícióját, majd az INEXT-tel lapozhatunk előre felé vagy visszafelé a file-ban (+ vagy -). Ha nincs szükségünk az összes rekordra, akkor adjunk meg kiválasztási maszkot. Ekkor csak a maszknak megfelelő rekordokat veszi elő az INEXT. A maszk lehet a kulcsban vagy azon kívül.

Kiválasztási maszk

A kiválasztási maszkkal olyan rekordokat olvashatunk, amelyek eleget tesznek a maszokban meghatározott feltételeknek. Ezzel felesleges rekordolvasások és körülményes vizsgálatok kerülhetők el. A maszk az ISTART, INEXT, IDATA utasításoknál használható.

Elve a következő. Először definiálunk egy " " karakterekből álló sztringet, amely azonos hosszú lehet akár a rekordunkkal. Ezután a megfelelő helyeken a " "-at a megfelelő adatokkal helyettesítjük. Például `mk$="ABCDE XY"` csak azokat a rekordokat veszi figyelembe, amelyekben az első helytől kezdődően az "ABCD", a 12. pozíciójától kezdődően az "XY" karakterek fordulnak elő. Ha nem akarunk maszkot használni, akkor üres sztringként definiáljuk.

- `mk$=""` : rem --- nincs kiválasztás
- `mk$=" QW"` : rem --- a 4. pozíciótól "QW"-t tartalmazó rekordok ---

I S T A R T

Növekvő vagy csökkenő sorrendű olvasás kezdetének beállítása.

Szintaktika: istart (-) lf,ku\$,mk\$,rc\$

ahol:

(-) - a mínusz csökkenő sorrendre utal (opció)

lf - logikai file szám (0-9)

ku\$ - teljes vagy részkulcs

mk\$ - kiválasztási maszk

rc\$ - a teljes rekord

Például:

```
100 rem --- file megnyitás ---
200 iopen 2,ui,"0:szemadat"
300 if ok=128 then:out"Sikertelen megnyitás",1,1
...
700 rem --- kezdet beall ----
710 ku$="KISS" : mk$=""
800 istart 2,ku$,mk$,rc$
810 if ok=255 then:out"Nincs ilyen",24,1
...
```

Megnyitjuk a file-t. Részkulccsal, kiválasztási maszk nélkül ráállunk az első megfelelő rekordra. Sikertelen olvasás esetén hibaüzenetet küldünk.

I N E X T

Növekvő vagy csökkenő sorrendű olvasás kulcs sorrendben az ISTART után.

Szintaktika: inext (-) lf,mk\$,rc\$

ahol:

(-) - a mínusz csökkenő , hiánya növekvő sorrendre utal

lf - logikai file szám (0-9)

mk\$ - kiválasztási maszk

rc\$ - a teljes rekord

Például:

```

100 rem ---- file megnyitás ---
200 iopen 2,ui,"0:szemadat"
300 if ok=128 then:out"Sikertelen megnyitás",1,1
...
700 rem --- kezdet beall ----
710 ku$="KISS" : mk$=""
800 istart 2,ku$,mk$,rc$
810 if ok=255 then:out"Nincs ilyen",24,1
...
900 inext 2,mk$,rc$
910 if ok=255 then out"Az állomány vége",24,1
...
950 goto 900
...

```

Megnyitjuk a file-t. Részkulccsal és kiválasztási maszk nélkül ráállunk az első megfelelő rekordra. Sikertelen olvasás esetén hibaüzenetet küldünk. Ciklusban olvassuk a rekordokat.

I V A L I D A T E

Módosítja az index táblázatot az IADD-al felvitt rekordokat beilleszti.

Szintaktika: ivalidate lf (,n)

ahol:

```

lf - logikai file szám (0-9)
n  - a feldolgozott rekordok száma

```

Például:

```

100 rem --- file megnyitás ---
200 iopen 2,ui,"0:szemadat"
300 if ok=128 then:out"Sikertelen megnyitás",1,1
...
700 rem --- iadd után index toldás ---
710 ivalidate 2

```

```

810 if ok=1 then:out"Ok. az index",24,1
...

```

Megnyitjuk a file-t. Az előzetesen IADD-al felvitt rekordokat bevisszük az index táblába.

I R E S T O R E

I D A T A

Az indexelt file nem kulcs sorrendű, hanem beírási sorrendű olvasása.

Szintaktika: irestore lf
 idata lf,mk\$,rc\$

ahol:

lf - logikai file szám (0-9)
mk\$ - kiválasztási maszk
rc\$ - a teljes rekord

Például:

```

100 rem --- file megnyitás ---
200 iopen 2,ui,"0:szemadat"
300 if ok=128 then:out"Sikertelen megnyitás",1,1
...
700 rem --- beviteli sorrendű olv ---
710 irestore 2
800 idata  2,"",rc$
810 if ok=255 then:out"A file vege",24,1
...
900 goto 800
...

```

Megnyitjuk a file-t. Ciklusban olvasunk beírási sorrendben a file végéig.

I U P G R A D E

Mágneslemezre írja a memóriában lévő indextáblát .

Szintaktika: iupgrade lf

ahol:

lf - logikai file szám (0-9)

Például:

```
100 rem --- file megnyitás ---
200 iopen 2,ui,"0:szemadat"
300 if ok=128 then:out"Sikertelen megnyitás",1,1
...
700 rem --- indextabla floppyra ---
710 iupgrade 2
...
```

Megnyitjuk a file-t. Biztonsági célból floppyra írjuk az indextáblát .

O N E R R O R

Mágneslemez hiba esetén programban kezelhetjük le a hibát.

Szintaktika: onerror sn

ahol:

sn - a program egy sorszama

Például:

```
100 onerror 900
...
900 rem --- fizikai hiba kezelese -
910 out"Mágneslemez hiba",24,1 : print st
920 stop : rem --- continue-val a hibas sorra
930 goto li: rem --- ugras a hiba soraba
...
```

Valamelyik sorban olvasás közben fizikai hiba lép fel. A hibaüzenet kiírása után a 920-as sorban megáll a program. A CONTINUE utasítással

visszaléphetünk arra a sorra, amelyben a hiba fellépett.

I R E S E T

Inicializálja a MASTER-64-et.

Szintaktika: ireset

Például:

```

100 rem --- file megnyitás ---
200 iopen 2,ui,"0:szemadat"
300 if ok=128 then:out"Sikertelen megnyitás",1,1
...
700 rem --- inicializálás ----
710 ireset
...

```

Megszünteti a memória indextáblázatokat, elengedi a maszkok, képernyőoldalak memóriafoglalását. Ha nem zárjuk le előtte a file-okat, akkor megbomlanak az indexstruktúrák.

Rendszer státuszváltozók

Az alábbi hat változó a MASTER-64 számára van fenntartva, ne használjuk őket más célokra.

LI - mágneslemez hiba esetén azt a sort adja vissza, amelyben a rendszer a hibát észlelte.

NR - a rekordok száma az indexelt file-ban; az IWRITE és az IADD eggyel növeli.

NI - a rekordok száma az indextáblában; az IWRITE és az IVALIDATE növeli, az IDELETE csökkenti.

LR - az aktuális indexelt file rekordjának hossza.

OK - státusz változó

```
ok = 0   -- eredményes végrehajtás
ok = 1   -- ISAM hiba
ok = 64  -- a file megtelt
ok = 128 -- nincs ilyen file
ok = 255 -- file vége.
```

TÖMÖRÍTÉS-ZÓNÁZÁS

Áttekintés

Az adattömörítési utasítások segítségével nagymértékben "magnövelhetjük" a mágneslemezkapacitásunkat. Mint az eddigiekből kiderült, a MASTER-64 számára egy fix hosszúságú sztring jelent egy rekordot, amelynek egy része a kulcs.

Tömörítési szempontból négyféle adattípus különböztethető meg:

- c - alfanumerikus
- s - bináris számok
- b - pozitív egész számok
- p - tömörítés nélküli típus

Minden tömörített mezőnél szükséges a tömörített hossz pontos kiszámítása, mivel a file definiálásához ismernünk kell a tömörített rekord teljes hosszát. A tömörítést a PUTST utasítás, a zónázást (ritkítást) a GETST utasítás hatja végre. A CREATST utasítás segítségével meghatározott karaktereket tartalmazó fix hosszúságú sztringek állíthatók elő.

A tömörített file létrehozásának sorrendje:

1 - file definíció

a rekord és mezőinek definiálása

minden mező hosszának definiálása
 minden tömörített mező hosszának kiszámítása
 a tömörített rekord teljes hosszának kiszámítása

2 - a file felhasználása

```
iopen      -- megnyitás
putst rc$  -- rekord tömörítés
rekord olvasás
rekord írás
getst rc$  -- rekord zónázás
iclose     -- lezárás
```

Például:

Vegyük a következő rekordot:

a\$ - 15 karakteres alfás sztring
 b\$ - 13 jegyű numerikus sztring
 c - pozitív egész szám
 d - lebegőpontos szám.

Ezek után az rc\$ rekord sztring a következő:

```
rc$=a$b$+str$(c)+str$(d),
```

amely $14+13+3+11 = 44$ karakteres rekordhosszt eredményez. Ha a rekord minden mezőjét tömöríteni akarjuk, akkor a következőkben ismertetett módon ki kell számítani a mezők tömörített hosszát.

Tömörítési típusok

ALFANUMERIKUS tömörítés - C típus

Ezzel a típussal az alfanumerikus karaktereket (32 és 96 közötti ASCII kód) 6 bitre sűrítjük. A nagybetűk kisbetűként tárolódnak !!!

A hossz számítása:

n - a tömörítetlen sztring hossza

lc - a tömörített sztring hossza

$$lc = n - \text{int}(n/4)$$

Példa:

A fenti példában a\$ hossza 15 karakter, tehát $n=15$.

$$lc = 15 - \text{int}(15/4) = 12$$

A c típusú tömörítések táblázatát a mellékletben találhatjuk.

BINÁRIS tömörítés - S típus

A sztringekben tárolt számoknál használhatjuk, amelyeket pl. a többszörös pontosságú művelettel számítottunk.

A hossz számítása:

n - a tömörítendő szám (nem a hossza)

Az s típusú tömörítések táblázatát a mellékletben találhatjuk.

POZITÍV EGÉSZ számok - B típus

Ez a típus a 65535-nél kisebb egész számok tömörítésére szolgál.

$n=(0, \dots, 256)$ $lc = 1$ byte

$n=(257, \dots, 65536)$ $lc = 2$ byte

LEBEGŐPONTOS számok - F típus

Értékétől függetlenül minden lebegőpontos szám 5 byte-on tárolható.

TÖMÖRÍTETLEN – P típus

Az előzőekből már tudjuk, hogy egy rekordba tömörített sztringnek állandó hosszúnak kell lennie. A p típusú tömörítéssel egy sztringet egy másikba helyezhetünk a megfelelő üres karakterek betoldásával. Az alfanumerikus adatokat használhatjuk c vagy p típussal, választásunk attól függ, hogy a sebesség vagy a helykihasználás a fontosabb. A két típust egy rekordban is elhelyezhetjük. A mintarekordunk 44 byte-os, hossza az előbbiek alapján $12+7+2+5=26$ karakterre csökken, ami kb. 40 százalékos helymegtakarítást tesz lehetővé. A példa alapján először a 26 byte-os üres sztringet kell létrehozni a CREATST utasítással. A PUTST utasítás az adatokat tömöríti, a GETST pedig zónázza.

Tömörítési utasítások

C R E A T S T

Üres, megadott karaktereket tartalmazó sztringet állít elő.

Szintaktika: creatst a\$,ln,(chr\$(a))

ahol:

a\$ - az előállított sztring
 ln - a sztring hossza
 a - a feltöltő karakterek ASCII kódja

Például:

```
...
100 creatst a$,80
200 creatst b$,10,"-"
...
```

Az első utasítás egy 80 hosszú space-kből álló sztringet állít elő, míg a második egy 10 db "-" karakterből álló sztringet.

P U T S T

Egy sztringet, rekord-sztringet tömörít.

Szintaktika: `putst rc$; tip,b$,poz,ln (; ...)`

ahol:

`rc$` - az előállított sztring
`tip` - típus (p,c,s,b,f)
`b$` - a tömörítendő változó
`poz` - a sztringen belüli pozíció
`ln` - a tömörítés utáni hossz

Például:

```
...
100 creatst rc$,21
110 putst rc$;c,a$,1,12;s,b$,13,7;b,c,20,2
```

Az első utasítás a 26 hosszú space-ből álló `rc$` sztringet állítja elő.
A második tömöríti a rekordunk mezőit az `rc$`-ba.

G E T S T

Egy sztringet, rekord-sztringet zónáz (ritkít).

Szintaktika: `getst rc$; tip,b$,poz,ln (; ...)`

ahol:

`rc$` - a zónázandó sztring
`tip` - típus (p,c,s,b,f)
`b$` - a zónázott változó
`poz` - a sztringen belüli pozíció
`ln` - a tömörített hossz

Például:

```
...
100 getst rc$;c,a$,1,12;s,b$,13,7;b,c,20,2
...
```

Az utasítás visszairitkítja a PUTST-el tömörített sztringet.

Kulcs tömörítése

Feltétel, hogy a kulcs első byte-jának ASCII kódja 250-nél kisebb legyen. Így az alábbiakra kell ügyelni:

- a c típus használata tilos
- az f típus használata tilos
- a b típus csak 32768-nál kisebb számoknál használható
- az s típus csak pozitív számoknál használható
- a p típus használata minden esetben megengedett.

Célszerű a kulcsmezőnél p típust használni, ami egyszerűsíti és gyorsítja is munkánkat.

BASIC-BŐVÍTÉSEK

Többszörös pontosságú műveletek

M A D D

Többszörös pontosságú összeadás.

Szintaktika: `madd n1$,n2$,n3$`

ahol:

`n1$` - az összeadandó számsztring

`n2$` - az összeadandó számsztring

`n3$` - az eredménysztring

Például:

```
n1$="1111111111111111"
```

```
n2$="2222222222222222"
```

```
madd n1$,n2$, n3$
```

```
print n3$
```

```
3333333333333333.00000000
```

A 14 jegyű egész számok összegét 14 jegy pontossággal állítjuk elő.

M M U L

Többszörös pontosságú szorzás.

Szintaktika: `mmul n1$,n2$,n3$`

ahol:

`n1$` - a szorzandó számsztring

`n2$` - a szorzandó számsztring

`n3$` - az eredménysztring

Például:

```

n1$="11"
n2$="22"
madd n1$,n2$, n3$
print n3$

```

```
242.000000000000
```

Egész számok szorzatát állítjuk elő.

M D I V

Többszörös pontosságú osztás.

Szintaktika: `mdiv n1$,n2$,n3$`

ahol:

```

n1$ - az osztandó számsztring
n2$ - az osztó számsztring
n3$ - az eredménysztring

```

Például:

```

n1$="100"
n2$="200"
mdiv n1$,n2$, n3$
print n3$

```

```
.33333333333330
```

Számított GOTO és GOSUB

G O T O

Egy változóban megadott sorra ugrik.

Szintaktika: `goto sn`

ahol:

sn - a sor számát tartalmazó változó

Például:

```
100 input "120 vagy 130";a
110 goto a
120 ? "120-as sor":stop
130 ? "130-as sor":stop
```

A megadott értéktől függően a 120-as, vagy a 130-as sorra ugrik a program.

G O S U B

Egy változóban megadott sorszámú szubrutint hajt végre.

Szintaktika: gosub sn

ahol:

sn - a sor számát tartalmazó változó

Például:

```
100 input "120 vagy 200";a
110 gosub a
...
120 ? "120-as szubrutin"
...
150 return
...
200 ? "200-as szubrutin"
...
300 return
```

A megadott értéktől függően a 120-as, vagy a 200-as szubrutinra ugrik a program.

Kisbetű-nagybetű átalakítás

U P L O W

Nagybetű-kisbetű átalakítást végez oda-vissza.

Szintaktika: uplow a\$
 ahol:
 a\$ - tetszőleges sztring

Például:

```
a$="KISS JANCS"
uplow a$
? a$

kiss janos
```

Különösképpen a c típusú zónázás után van jelentősége, mivel a tömörítés mindent kisbetűre konvertál.

Leegyszerűsített közvetlen diszk elérés

Adatfeldolgozási feladatoknál nem használatos, ezért csak szintaktikáját ismertetjük.

Szintaktika: putbuf a\$,poz
 putblock track,sect (,d) (,u)
 takbuf a\$,poz,ln
 inblock track,sect (,d) (,u)

Programvédelem

N O L I S T

Program mentése, a program a későbbiek során nem listázható.

Szintaktika: nolist "prnev"

Dátumtömörítés, ellenőrzés

Áttekintés

A MASTER-64 támogatja a dátumkezelést is. A dr\$ nevű rendszerváltozóban kell elhelyezni a kezdő dátumot. Az ellenőrzendő dátumot a da\$ nevű rendszer-változóba kell letenni. A DATEPACK utasítás ellenőrzi a da\$-ban levő dátum érvényességét. A DATEUNPACK utasítással zónázható a dátum. Tömörített formában a da nevű rendszerváltozóba kerül a tömörített dátum. Az ellenőrzendő dátum és a kezdő dátum különbsége nem haladhatja meg a 10 évet!

D A T E P A C K

Ellenőrzi és tömöríti a dátumot.

Szintaktika: datepack

Például:

```
dr$="01/01/83"
da$="14/10/83"
datepack
?ok
```

0

Érvénytelen dátum esetén elmarad a tömörítés, az ok értéke 10 lesz.

D A T E U N P A C K

Zónázza a dátumot.

Szintaktika: dateunpack

Például:

```
dr$="01/01/83"
```

```
da$="17/11/83"
datepack
?ok
```

```
0
da$=""
dateunpack
?da$
```

```
17/11/83
```

Érvénytelen dátum esetén elmarad a tömörítés, az OK értéke 10 lesz.

Karakterkeresés egy sztringben

H U N T

Karaktert keres egy sztringben. A keresés eredménye a ZO rendszerváltozóba kerül.

Szintaktika: hunt (-) chr\$(a),a\$,p

ahol:

(-)- a mínusszal a chr\$(a)-val nem egyenlő első karaktert keresi

a mínusz nélkül az azzal egyenlő első karaktert keresi

a - a keresett karakter ASCII kódja

a\$ - a végignézendő sztring

p - a keresés kezdő pozíciója

Például:

```
a$="kiss janos"
hunt "j",a$,1
?zo
```

Ha a keresett karakter nincs a sztringben, akkor a ZO-ba 0 kerül.

E D E X bővítés (programfejlesztési eszközök)

Áttekintés

Az EDEX a BASIC 4.0 bővítése. Az alábbiakban részletezett utasítások jelentősen megkönnyítik a programok írását, tesztelését:

- programszerkesztés (auto,delete,renu)
- tesztelés (dump, error,find)
- output utasítások (hardcopy,plot,reset)
- if...then...else
- beep
- trace
- egyéb

A U T O

Automatikus sorszámozás.

Szintaktika: auto n

ahol:

n - a sorszám növekmény (1-255)

Például:

```
auto 10
10 rem --- program ----
20 (return)
```

Tízessel automatikusan írja ki a program sorszámokat. A (RETURN) billentyűvel léphetünk ki belőle.

D E L E T E

Sor(ok) törlése.

Szintaktika: delete a : rem -- egy sort töröl
 delete a- : rem -- a-tól a program végéig töröl
 delete -a : rem -- a program elejétől az a-ig töröl
 delete a-b : rem -- a-tól b-ig töröl
 delete a,b : rem -- az a és a b programsorokat törli

Például:

```
delete 200
delete 340-460
delete 12300-
```

R E N U

Újraszámozza a programot.

Szintaktika: renu (a(,b(,c)))

Például:

```
renu             :rem --- 100-tól 10-esével
renu 5           :rem --- 100-tól 5-ösével
renu 10,2       :rem --- 10-tól 2-esével
renu 9,100,4:rem --- 9.sortól kezdve,100-tól 4-esével
```

D U M P

A program változói és azok értékeinek listája.

Szintaktika: dump

Például:

```
a=5:b$="abc":c=6
dump
a=5
```

```
b$="abc"  
c=6
```

F I N D

Karaktersist keres a programban.

Szintaktika: find/szoveg/(a-b)

Például:

```
10 print "alma"  
20 print "almas"  
30 print "korte"  
find/alm/  
10 print "alma"  
20 print "almas"
```

E R R O R

A program futása közben hiba esetén kijelezhető a hiba.

Szintaktika: error

H A R D C O P Y

Nyomtatóra írja ki a képernyő tartalmát.

Szintaktika: hardcopy

P U S I N G

Formátumos írás képernyőre ill. nyomtatóra

Szintaktika: pusing f\$,a\$,b\$,...,a,b,...
pusing x,f\$,a\$,b\$,...,a,b,...

ahol:

`f$` - a formátum sztring
`a` - alfanumerikus
`9` - numerikus vezető 0-kkal
`z` - numerikus
`$` - az első karakteren \$jelenik meg
`.` -,tizedespont

`a$,b$,a,b` - a változók
`x` - az output logikai száma

Például:

```
f$="$9999.99"
pusing #2,f$,2132
```

B E E P

A beépített sípot szólaltatja meg.

Szintaktika: `beep`

Például:

```
out"Sipolo hangjelzés",1,1:beep
```

I F ... T H E N ... E L S E

Az `if ... then` utasítás bővítése, amely áttekinthetőbbé teszi a programozást.

Szintaktika: `10 IF feltétel THEN utasítások ELSE utasítások`
`10 IF feltétel THEN utasítások`
`20 ELSE utasítások`

Például:

```
10 if f1 then i1
20   if f2 then i2
30     if f3 then i3
```

```
40         else h3
50         else h2
20     else h1
```

T R A C E
O F F

A trace a program utasításonkénti végrehajtását teszi lehetővé. Az off kikapcsolja.

Szintaktika: trace
 off

Használata:

Az aktuális utasítás megjelenik a képernyő felső sorában. Tovább lépés a következő utasításra a baloldali SHIFT billentyűvel lehetséges. A jobboldali SHIFT billentyűvel gyorsabb futás kezdeményezhető.

S C R E E N

A képernyő színeit adhatjuk meg vele.

Szintaktika: screen a,b (,c)

ahol:

a - háttérszín
b - keret színe
c - karakterszín

MELLÉKLET

Tömörítési táblázatok

C-típusú alfanumerikus tömörítési táblázat

N-eredeti hossz

ln-tömörítés utáni hossz

N	ln	N	ln	N	ln	N	ln
1	1	21	16	41	31	61	46
2	2	22	17	42	32	62	47
3	3	23	18	43	33	63	48
4	3	24	18	44	33	64	48
5	4	25	19	45	34	65	49
6	5	26	20	46	35	66	50
7	6	27	21	47	36	67	51
8	6	28	21	48	36	68	51
9	7	29	22	49	37	69	52
10	8	30	23	50	38	70	53
11	9	31	24	51	39	71	54
12	9	32	24	52	39	72	54
13	10	33	25	53	40	73	55
14	11	34	26	54	41	74	56
15	12	35	27	55	42	75	57
16	12	36	27	56	42	76	57
17	13	37	28	57	43	77	58
18	14	38	29	58	44	78	59
19	15	39	30	59	45	79	60
20	15	40	30	60	45	80	60

S-típusú bináris tömörítési táblázat

N-eredeti hossz

ln-tömörítés utáni hossz

Nmax	lc
127	1
32767	2
8388607	3
2.14748 E9	4
5.49756 E11	5
1.40737 E14	6

ISAM file definiálás 1000 rekordra

File név : SZEM.ISM

Adat drive szám : 0

Index drive szám : 0

Level(index-szint): 1

MAX. rekordszám : 1000

Rekordhossz : 124

Kulcshossz : 11

Kulcs pozíció : 1

Memória index tábla : 1295

Adat blokkok száma: 501

Indexblokkok száma: 85 + 6

Blokkok össz száma: 592

ok (i/n): i lefoglalás(i/n) n

ISAM file definiálás 100 rekordra

File név : SZEM.ISAM

Adat drive szám : 0

Index drive szám : 0

Level(index-szint): 1

MAX. rekordszám : 100

Rekordhossz : 124

Kulcshossz : 11

Kulcs pozíció : 1

Memória index tábla : 245

Adat blokkok száma: 51

Indexblokkok száma: 10 + 1

Blokkok össz száma: 61

ok (i/n); i lefoglalás(i/n) i

Személyzeti nyilvántartás példaprogram

```

30 rem *****
40 rem *** személyi nyilvántartás ****
50 rem *** ****
60 rem ***** master-64-ben *****
70 rem ***** *****
80 rem *****
82 gosub 2000 : open 2,3,1
100 rem *****
102 rem *** fomenu *****
104 rem *****
105 sclear : decz 100,20,13,1,n
106 out "Személyzeti rendszer fomenu",3,6
107 out "(1) - Kilepes ",7,3
108 out "(2) - Felvitel",9,3
110 out "(3) - Modositas",11,3
112 out "(4) - Torles ",13,3
114 out "(5) - Listazas ",15,3
116 out "Funkcio ? ",20,3
117 tline 40,1,1: tline 40,23,1: tcol 23,1,1: tcol 23,1,40
118 neaz 100 : inz 100,v$
120 if v$="1" then iclose 1: close 2: stop
122 if v$="2" then gosub 200:goto 100
124 if v$="3" then gosub 300:goto 100
126 if v$="4" then gosub 400:goto 100
128 if v$="5" then gosub 450:goto 100
130 goto 118
140 rem
150 rem
160 rem
200 rem *****
202 rem *** felvitel *****
204 rem *****
210 gosub 500 : gosub 1000 : gosub 1500
212 iwrite 1,nc$
222 if ok(>)1 and ok(>)255 then goto 230
224 out "Mar van ilyen szemsz az allomanyban !",24,1
226 get xx$: if xx$(<)chr$(13) then goto 226
228 clear 40,23,1 :return
230 return
300 rem *****
302 rem *** modositas *****
304 rem *****
330 sclear
332 out "Kerem a személyi számot :".2,1
334 decz 102,2,30,11,n :neaz 102: inz102,ss$
340 inead 1,ss$,nc$
342 if ok(>)1 and ok(>)255 then goto 350
344 out "Nincs ilyen rekord az allomanyban !",24,1

```

```

346 get xx$: if xx$(<)chr$(13) then goto 346
348 clear 40,23,1 :return
350 gosub 1700 : gosub 500
351 outz 1,ts$ : outz 2,fe$ : outz 3,sz$ : outz 4,nv$
355 outz 6,sh$ : outz 7,cs$ : outz 8,gy$ : outz 9,ins$ :
    outz 10,va$
357 outz 11,ut$ : outz 12,sa$ : outz 13,em$ : outz 14,aj$ :
    outz 15,tf$
360 outz 16,kt$ : outz 17,al$ : outz 18,os$ : outz 19,fo$ :
    outz 20,ka$
365 outz 21,af$
370 gosub 1000: gosub 1500
380 iupdate 1.rc$
382 if ok(>)1 and ok(>)255 then goto 390
384 out "Mar van ilyen szemsz az allomonyban !" .24.1
386 get xx$: if xx$(<)chr$(13) then goto 386
388 clear 40,23,1
390 return
400 rem *****
402 rem *** torles *****
404 rem *****
410 sclear
412 out "Kerem a személyi számot :".2.1
414 decz 102,2,30,11,n :neaz 102: inz102.ss$
420 idelete 1,ss$
422 if ok(>)1 and ok(>)255 then goto 430
424 out "Nincs ilyen rekord az allomonyban !" .24.1
426 get xx$: if xx$(<)chr$(13) then goto 426
428 clear 40,23,1 :return
430 return
450 rem *****
452 rem *** listazas *****
454 rem *****
455 sclear
456 out " Kapscsolja be a nyomtatot !" .1.1
457 get xx$: if xx$(<)chr$(13) then goto 457
458 print#2," Szemelyzeti rendszer lista" :

    print#2: print#2
459 print#2,"Szem.Szam.      Nev      "+"
    "Alapfiz Torzsszam ":
460 print#2," Vagos": print#2: print#2
461 istart 1,"","".rc$
462 gosub 1700
464 print#2,sz$:"      ":nv$:" ":
466 pusine#2,"999999.99".af$:
468 print#2,"      ":ts$:"      ":va$
469 rem *** olvasasi ciklus *****
470 inext 1,"",rc$
472 if ok=255      then goto 494
474 gosub 1700
484 print#2,sz$:"      ":nv$:" ":

```

```

486 pusing#2,"999999.99",af$;
488 print#2,"      ":ts$;"      ";va$
492 goto 470
494 get xx$: if xx$(<)chr$(13) then goto 494
496 return
500 rem *****
502 rem *** fejlecek, zonadefiniciok *****
504 rem *****
505 sclear
506 out "Szemelyi lap ",2,10 : rev 2,10,1,12
510 out "Torzsszam :", 4,1 : decz 1, 4,12, 7,n
520 out "Feorszam :", 5,1 : decz 2, 5,12,8,n
530 out "Szem.szam :", 6,1 : decz 3, 6,12,11,n
540 out "Nev :",8,1 : decz 4,8,7,25,m
550 out "Szul. ido :", 9,1: decz 5, 9,13,10
560 out "Szul. hely :",10,1: decz 6,10,13,15,m
570 out "Csaladi all:",11,1: decz 7,11,13,10,m
580 out "Gyerm.szama :",11,25 : decz 8,11,38,2,n
590 out "Irszm.:",14,1 : decz 9,14,8,4,n
600 out "Varos :",14,18 : decz 10,14,25,15,m
610 out "Utca :",15,1 : decz 11,15,8,20,m
620 out "Szam :",16,1 : decz 12,16,8,3,n
630 out "Emelet:",16,14 : decz 13,16,21,2,n
640 out "Ajto :",16,24 : decz 14,16,31,3,n
650 out "Telef.:",17,1 : decz 15,17,8,6,n
660 out "Ktgh :",19,1 : decz 16,19,8,5,n
670 out "All.csop:",19,20 : decz 17,19,29,5,n
680 out "Dosztaly :",20,1 : decz 18,20,10,1,n
690 out "Fooszt. :",20,20 : decz 19,20,29,1,n
700 out "Katonasag :",21,1: decz 20,21,12,20,m
710 out "Alappiz :",22,1: decz 21,22,12, 5,n,p,"98888"
750 return
760 rem
770 rem
780 rem
790 rem
1000 rem *****
1002 rem ** adatbekeres *****
1004 rem ** kepennyorol *****
1006 rem *****
1008 revz 1 to 21
1010 reqz 1 : inz 1,ts$
1020 reqz 2 : inz 2,fe$
1030 reqz 3 : inz 3,sz$
1040 reqz 4 : inz 4,nv$
1050 si$="19"+mid$(sz$,2,2)+"."+mid$(sz$,4,2)+ "."+
mid$(sz$,6,2) : outz 5,si$
1060 reqz 6 : inz 6,sh$
1070 reqz 7 : inz 7,cs$
1080 reqz 8 : inz 8,gy$
1090 reqz 9 : inz 9,in$
1100 reqz 10 : inz 10.va$

```

```

1110 reqz 11 : inz 11,ut$
1120 reqz 12 : inz 12,sa$
1130 reqz 13 : inz 13,em$
1140 reqz 14 : inz 14,aj$
1150 reqz 15 : inz 15,tf$
1160 reqz 16 : inz 16,kt$
1170 reqz 17 : inz 17,al$
1180 reqz 18 : inz 18,os$
1190 reqz 19 : inz 19,fo$
1200 reqz 20 : inz 20,ka$
1210 reqz 21 : inz 21,af$
1300 return
1500 rem *****
1502 rem *** rekordsurites *****
1504 rem *****
1505 creatst r1$,23: creatst r2$,21: creatst r3$,29
1506 creatst r4$, 9: creatst r5$,23: creatst r6$,19 :
    creatst rc$,124
1508 gy%=val(gy%): in%=val(in%): sa%=val(sa%): em%=val(em%):
    aj%=val(aj%)
1509 kt%=val(kt%): al%=val(al%)
1510 putst r1$: p,sz$,1,11; c,ts$,12,6; c,fe$,18,6
1520 putst r2$: c,sh$,1,12; c,cs$,13,8; b,gy%,21,1
1530 putst r3$: b,in%,1,2; c,va$,3,12; c,ut$,15,15
1540 putst r4$: b,sa%,1,2; b,em%,3, 1; b,aj%, 4,1; c,tf$,5,5
1550 putst r5$:b,kt%,1,2;b,al%,3,2;c,os$,5,1;c,fo$,6,1;
    c,ka$,7,15; s,af$,22,2
1555 putst r6$:c,nv$,1,19
1560 rc$=r1$+r2$+r3$+r4$+r5$+r6$
1600 return
1610 rem
1620 rem
1700 rem *****
1702 rem *** rekord nitkitas *****
1704 rem *****
1705 r1$=mid$(rc$,1,23) : r2$=mid$(rc$,24,21)
1706 r3$=mid$(rc$,45,29) : r4$=mid$(rc$,74,9)
1707 r5$=mid$(rc$,83,23) : r6$=mid$(rc$,106,19)
1710 getst r1$: p,sz$,1,11; c,ts$,12,6; c,fe$,18,6
1720 getst r2$: c,sh$,1,12; c,cs$,13,8; b,gy%,21,1
1730 getst r3$: b,in%,1,2; c,va$,3,12; c,ut$,15,15
1740 getst r4$: b,sa%,1,2; b,em%,3, 1; b,aj%, 4,1; c,tf$,5,5
1750 getst r5$:b,kt%,1,2;b,al%,3,2;c,os$,5,1;c,fo$,6,1;
    c,ka$,7,15; s,af$,22,2
1755 getst r6$:c,nv$,1,19
1790 gy$=str$(gy%): in$=str$(in%): sa$=str$(sa%):
    em$=str$(em%): aj$=str$(aj%)
1800 kt$=str$(kt%): al$=str$(al%)
1802 gy$=mid$(gy$,2): in$=mid$(in$,2): sa$=mid$(sa$,2)
1804 em$=mid$(em$,2): aj$=mid$(aj$,2)
1806 kt$=mid$(kt$,2): al$=mid$(al$,2)
1810 uplow nv$ : uplow sh$ : uplow va$

```

```
1820 uplow ut$ : uplow ka$ : uplow cs$
1990 return
1992 rem
1994 rem
2000 rem *****
2002 rem ** isam file nyitas *****
2004 rem *****
2010 iopen 1,8,"0:szem.ism"
2020 if ok=128 then goto 2050
2030 sclear
2040 out "Rekordok szama="+str$(nr),2,1
2042 get xx$: if xx$(<)chr$(13) then 2042
2045 return
2050 sclear: out"Kerem feltenni a jo lemezt !",2,1
2060 get xx$: if xx$(<)chr$(13) then goto2060
2070 goto 2010
2080 rem
2090 rem
```

ready.

4. Adatvédelem

Az adat és az információ értéke az ügyviteli rendszereknél akár manuális, akár számítógépes rendszer, igen fontos. Ezért kiemelt szerepe van az adatvédelemnek. Az 1985. január 1-től érvénybe lépett MSZ 7788 szabványsorozat 8. számú szabványa szabványa szerint az adatvédelem meghatározása a következő:

'Az adatfeldolgozó rendszerben a hardver, a szoftver és az adatok akaratlan vagy rosszhiszemű megváltoztatásával, megsemmisítésével illetve nyilvánosságra hozásával szembeni védelem.'

Könyvünkben nem akarunk az adatvédelem, a szabványban említett, minden területével foglalkozni. A könyv célja, hogy segítséget nyújtsunk a Commodore 64 személyi számítógép segítségével megoldandó ügyviteli adatfeldolgozó rendszerhez a programcsomagok kiválasztásában, valamint néhány rendszerszervezési kérdésben. Bármilyen adatfeldolgozási rendszer kidolgozásához hozzátartozik az adatvédelem megszervezése. Sajnos azt lehet tapasztalni, megfigyelni, hogy a személyi számítógépek elterjedésével mindinkább háttérbe szorul az adatvédelem. Ezért tartottuk fontosnak külön fejezetet szentelni az adatvédelemnek. Az adatvédelmet csak két szempontból vizsgáljuk: az adathordozó védelme és az adatbiztonság.

A leírtak alapján elfogadhatjuk, hogy a számítógépes adatfeldolgozó rendszereknél nagy figyelmet kell szentelni mind az adatok biztonságának, mind a mágnesesadathordozók védelmének.

4.1 Lemezvédelem

A hajlékony mágneslemezeket nem szabad kitenni sem fizikai, sem mágneses,

sem hőhatásnak.

A fizikai károsodás ellen úgy tudunk védekezni, ha a lemezeinket mindig szilárd dobozban, és a dobozokat szekrényben tartjuk. Ebben az esetben nem tesszük ki olyan véletlen eseményeknek (pl. lemez félbehajtása), amelyek végleges fizikai károsodást okozhatnak a lemezeinken.

A hajlékony mágneslemezek anyaga olyan, hogy még napsugár hatására is maradandóan károsodhat, ezért vigyázni kell, hogy se ablak közelébe, se meleg munkahelyen, magas hőfok közelébe még rövid ideig se tartsuk.

Ez ellen a hatás ellen is tudunk úgy védekezni, ha dobozban tartjuk a lemezeket, és a dobozokat szekrényben. (Természetesen nem a nagy hőfokú berendezések mellett.)

A legnagyobb károsodást a mágneses adathordozókon a mágneses ill. elektromos mező jelenti. Ez azért van így, mert ezeket a hatásokat az ember érzékszervei a legkevésbé érzékelik, tehát nem veszik észre a károsodást idejében.

Természetesen ebben az esetben is a doboz és szekrény megoldás a legjobb. De erre még munka közben is nagy figyelmet kell szentelni, hogy pl. ne tegyük a lemezünket a meghajtóegységre.

Ha olyan munkakörülmények között dolgozunk, hogy állandó mágneses hatásnak vannak kitéve az adathordozók, akkor célszerűnek látszik abból a helységtől kiköltöztetni a Commodore 64 számítógépet, teljes konfigurációval.

4.2 Adatbiztonság

Az adatbiztonságot egyrészt az adatrögzítésnél az adatok ellenőrzésével tudjuk megoldani, másrészt az ellenőrzött és állományba szervezett adatok archiválásával. Itt az adatok ellenőrzésével nem foglalkozunk.

A rögzített adataink védelmét archíválásokkal biztosíthatjuk. Az archíválás, vagy más szóval backup azt jelenti, hogy másolatokat készítünk az adatlemezeinkről.

Az adatok védelme több szempont miatt szükséges.

Az első szempont, hogy az adatlemezünk elromolhat, valamilyen fizikai meghibásodás lévén.

A második szempont, hogy az adatokat rosszul rögzítettük és ezért romlott el az adatállományunk.

Az archíválásokat több generációban érdemes elvégezni.

Az első generáció az adatállományok felállítása után készül, az első lemezre. (A generáció teljes adatállomány másolását jelenti.)

A második generációt a következő karbantartás után kell egy újabb adatlemezre másolni.

A harmadik generációt a soron következő adatállomány karbantartás után kell megint egy újabb lemezre elkészíteni.

A karbantartások végrehajtása után az első archívlemezt kell elővenni, és így tartunk mindig három generációs archívot. Erre azért van szükség, hogy minden esetben a lehető legkevesebb adatrögzítési munkával tudjuk helyreállítani az elromlott adatállományt.

Ha gépmeghibásodás miatt történik az állomány elromlása, akkor a karbantartást megelőző archív állományból kell a helyreállítást elvégezni, és csak az utoljára történt adatrögzítéseket kell megismételni, hogy elérjük azt az állapotot, ami a gép meghibásadásakor készült.

Ha a feldolgozás során történt a hiba - tehát nem gépi, hanem emberi hibából származik - ezt természetesen csak leválogatáskor, vagy táblázat készítésekor vehetjük észre, amikor már a legújabb archív generáció elkészült.

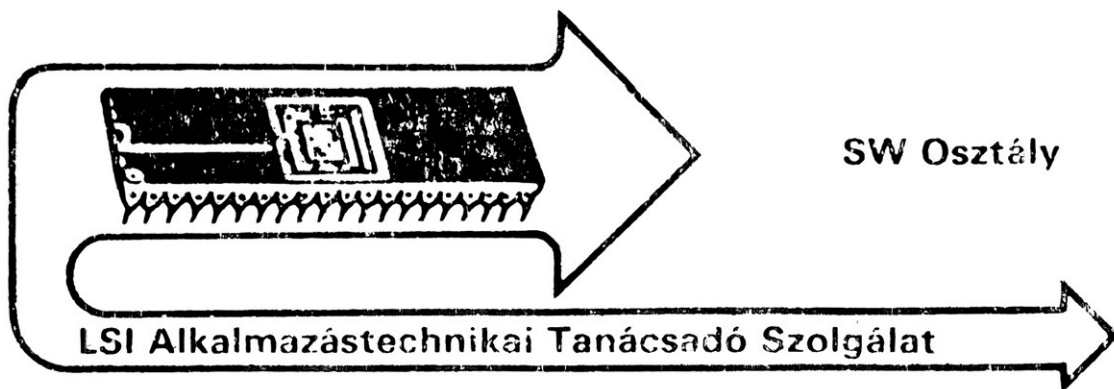
Ebben az esetben az utolsó előtti archív állományból kell helyreállítani az adatállományt, és a hibák elemzése után újra el kell végezni az utolsó feldolgozást.

Igen ritkán, de sajnos előfordul, hogy az utolsó feldolgozás hibás volt, és elromlott az előző archív generációs állomány. Ilyenkor kell fordulni a legelső archív állományhoz. Ez természetesen nagyon sok ismételt adatrögzítői, adatfeldolgozói munkát eredményez, de nem szükséges teljesen előlről kezdeni minden munkát.

Az archiválásokról rendszeresen archív naplót kell vezetni, és ezt biztonságos helyen kell tárolni.

Az archív naplónak tartalmaznia kell:

- a napi dátumot
- a generáció számát
- az adat-feldolgozó nevét



SW Osztály



573-374

570-433384

IBM PC/XT/AT MS-DOS ÉS BASIC DEMONSTRÁCIÓS LEMEZ

Célja: Az operációs rendszer használata és BASIC programozás segítése.

Részei: MS-DOS Help - a parancsok részletes ismertetése mellett valamennyi MS-DOS funkció elérése a végrehajtás alatti parancs magyarázatával, a hibalehetőségek kizárásával.

MS-DOS Tutor - oktatóprogram az operációs rendszer tanulásához. Teljesen automatikus, kézikönyv-mentes önálló tanulást biztosít.

BASIC Data I. - adatfeldolgozás BASIC-ben - mintaprogram.

BASIC Data II.- bonyolultabb adatfeldolgozási feladatok megoldása - sort mintaprogram.

BASIC Num - numerikus feladatmegoldás BASIC-ben - mintaprogram.

BASIC Graphics - üzleti grafika BASIC-ben - mintaprogram.

Ára: 12.500,- Ft

IBM PC/XT/AT ADATBÁZIS (dBASE II) DEMONSTRÁCIÓS LEMEZ

Célja: A relációs adatbáziskezelés alapjainak elsajátítása.

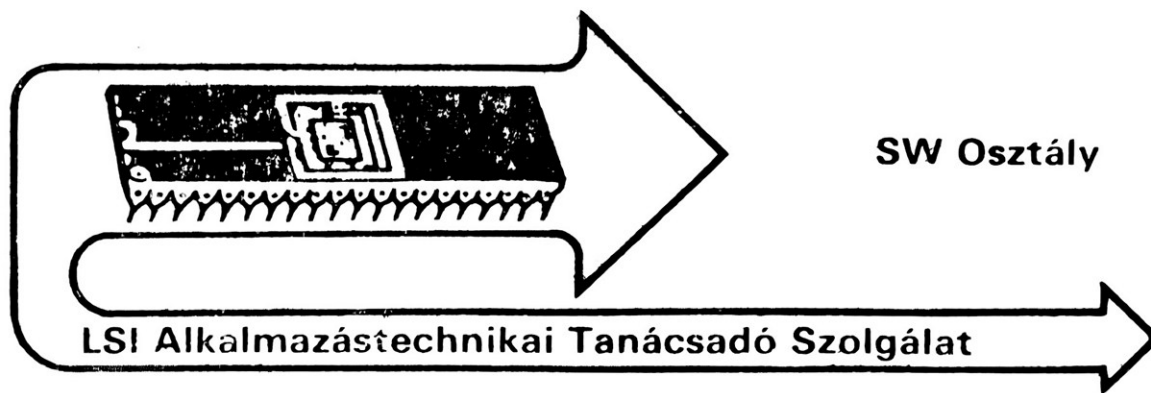
Részei: dBASE II. Help - magyar nyelvű help facility (segédmenü lehetőség) dBASE II-höz.

dBASE II.Tutor - oktatóprogram a dBASE lehetőségeinek önálló tanuláshoz automatikus, kézikönyv-mentes módon.

dBASE II.Primer - egyszerű dBASE adatfeldolgozás - címlista/stricker író mintaprogram.

dBASE II.Prof - bonyolultabb dBASE alkalmazás bemutatása egy készletnyilvántartási/gazdálkodási rendszeren.

Ára: 18.500,- Ft



573-374
570-433/384

Commodore 64 - IBM PC adatátvitel

Az általunk kifejlesztett interface és adatátviteli programok segítségével Commodore 64 számítógépét intelligens adatrögzítő gépnek használhatja, amelynek floppy lemezéről az adatok átvihetők az IBM PC/XT/AT (vagy PROPER-16, Commodore PC 10/PC 20 stb.) számítógépre. Így az adatgyűjtés az IBM géptől elkülönítve történhet, egy vagy több Commodore 64-en, az IBM PC gépidejének lekötése nélkül. Az adatcsere kétirányú lehet; tehát az IBM gépről is küldhetünk át adatfile-okat a Commodore 64-re.

Az átviteli programok gyorsak (assembly nyelven íródtak), ellenőrzik és jelzik az átvitel során esetleg fellépő hibákat.

Az adatátviteli hardver és szoftver ára együtt: 28.500,- Ft

C-64 COMPILERS DEMO-LEMEZ

Célja: A C-64 számítógépen futtatható legelterjedtebb fordítók lehetőségeinek bemutatása.

Részei: 1.BAS, 2.BAS, 3.BAS, 4.BAS

Négy különböző BASIC nyelvű rendező program)

1. BUBORÉK;
2. SHELL-METZNER;
3. HEAP-SORT;
4. QUICK-SORT algoritmus

1.AUSTRO, 2.AUSTRO, 3.AUSTRO, 4.AUSTRO

A BASIC programok fordításai az "AUSTRO-COMP" fordítóval fordítva;

1.BLITZ, 2.BLITZ, 3.BLITZ, 4.AUSTRO

A BASIC programok fordításai a "BLITZ" fordítóval fordítva;

1.PETSPEED, 2.PETSPEED, 3.PETSPEED, 4.PETSPEED

A BASIC programok fordításai a "PETSPEED" fordítóval fordítva.

Ára: 3.500,- Ft

Kiadó: LSI ATSZ

Felelős kiadó: Dr. Kovács Magda

Témafelelős: Sziklai Klára

Készült: MÁTRAINVEST GT Nyomda

Felelős vezető: Dr. Csák Máté

Engedélyszám: 47176

ISBN: 963 592 5417

Ív: **16,5** ív

Példányszám: 3000



SW Osztály



573-374

570-433/384

IBM PC/XT TURBO PASCAL DEMONSTRÁCIÓS LEMEZ

Célja: A Turbo Pascal rendszer lehetőségeinek elsajátítása.

Részei: Turbo Help - magyar nyelvű hibaüzenetek a Turbo Pascal fejlesztőrendszerhez.

Turbo Tutor - oktatóprogram a fejlesztőrendszer részletes, a Pascal nyelv változatos, a Turbo Pascal parancsok szintaxisának részletes ismertetésére. Teljesen automatikus kézikönyv-mentes önálló tanulási lehetőséget biztosít.

Turbo Quick - Az ismert leghatékonyabb rendezési algoritmus - quicksort - Turbo Pascalban. Tanulságos mintaprogram.

Turbo Graph - Grafikus mintaprogram Turbo Pascalban.

Ára: 12.500,- Ft

IBM PC/XT WORDSTAR DEMONSTRÁCIÓS LEMEZ

Célja: A Wordstar, a legelterjedtebb szövegszerkesztő és segédprogramjai, a Spellstar és a Mailmerge elsajátíttatása.

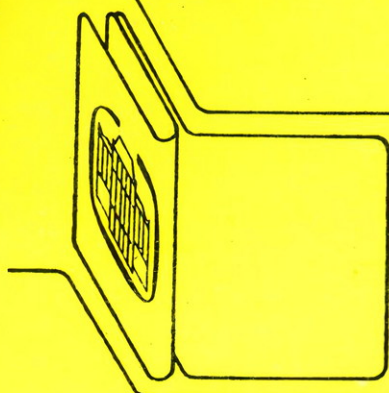
Részei: Wordstar Tutor - a Wordstar/Spellstar/Mailmerge lehetőségeinek önálló tanulásához automatikus, kézikönyv-mentes módon. Hozzásegít a programrendszer igen széleskörű és nehezen megtanulható parancsrendszerének begyakorlásához.

Wordstar Sample - a rendszer sok lehetőségét magyarázó szövegfile.

Mailmerge Sample - mintakörlevél, adatfile-lal együtt.

Ára: 12.500,- Ft

ára: 185.-Ft



Magiszter

AKADÉMIAI KÖNYVESBOLT

Budapest V. Városház u. 1. 1052

Tel.: 382-402, 382-440

MNB PÜ. jelzőszám: 215-01283

Örömmel értesítjük a tisztelt Olvasót, hogy nemrég megnyitolt boltunk az Akadémiai Kiadó teljes könyv- és folyóiratválasztékán, valamint a legjobban ellátott könyvesboltok szokásos kínálatán túl újszerű szolgáltatásokat is nyújt:

- *különbéféle típusú professzionális és házi személyi számítógépek bemutatása,*
- *játékprogramok, ügyviteli, gazdálkodási és tudományos felhasználói programcsomagok árusítása,*
- *megrendelések felvétele teljes (hardver + szoftver) rendszerekre,*
- *szaktanácsadás,*
- *C-64, PRIMO, TV Computer, VARYTER XT, PROPER 16W és VT 16 gépekkel felszerelt kis bemutatótermünkben szűk körű szakmai találkozók, bemutatók szervezését vállaljuk.*

Tisztelettel

Kucik Pál

üzletvezető