

MÓRICZ ATTILA



DOS

alapsmerek I.



Móricz Attila

DOS Alapismeretek I.

Lektorálta:
Dr. Kiss Jenő
Főiskolai adjunktus

**Nyitott rendszerű képzés - távoktatás -
oktatási segédlete * Tankönyv**

LSI Oktatóközpont

A könyvben szereplő ismeretek a MS-DOS 5.0 és a MS-DOS 6.00 operációs rendszerben is alkalmazhatóak.

Nyelvi lektor: Hérincs Judit tanító

ISBN 963 577 092 8 Ö
963 577 093 6 I.

Kiadó: **LSI Oktatóközpont Alapítvány**
Felelős vezető: **Dr. Kovács Magda**
Témafelelős: **Flier István**

LIGATURA KFT - VÁCI ÁFÉSZ NYOMDA
Munkaszám: 108-94

Tartalomjegyzék

Alapismeretek

A számítógép felépítése, a fizikai teste (HARDWARE)	2
A kapcsolódó ki- és bemeneti egységek	3
A számítógép bekapcsolás után (SOFTWARE)	3
A számítógép használata	5
Számítástechnikai alapfogalmak	7
A számítógép részei	11
Monitor	11
Billentyűzet	12
Lemezegység, hajlékony lemez	16
Merevlemez háttértár	18
Adatok tárolása lemezeken	20
Billentyűzetkezelés a DOS-ban	21
A számítástechnikában használatos jelölések	26

Parancsok ismertetése:

DATE - dátum	29
TIME - időállítón	30
PROMPT - képernyő jel	31
VER - Dos verziója	33
CLS - képernyő törlés	33

✗ Könyvtárak, alkönyvtárak, fájlok	34
DIR - könyvtár lista	35

✗ Műveletek könyvtárakkal	41
MKDIR - H.D. - könyvtárak létrehozása	41
CHDIR - C.D. - könyvtárváltás	41
RMDIR - R.D. - könyvtárak törlése	42
TREE - könyvtárfa rajzolás	46

Műveletek lemezekkel	48
VOL - lemez címke kiindegés	48
LABEL - lemez címke előírás	48
CHKDSK - lemez vizsgálat	50
✗ FORMAT - lemez formázás	51
SYS - új rendszerállományok formázása	59
DISKCOPY - lemez másolás	60

Műveletek fájlokkal	62
✓ COPY <i>Fájlok másolása</i>	62
ATTRIB <i>jelszót állít</i>	70
XCOPY <i>másolat másolás</i>	78
✓ DEL <i>állományok törlése</i>	78
✓ ERASE	78
✓ UNDELETE <i>törölt állományok visszahozása</i>	80
RENAME <i>állományok átnevezése</i>	82
REN	82
Távoli fájl elérése, környezeti változók	83
✓ PATH <i>húsvéti környezet kiadása</i>	84
APPEND <i>adatok állományokba húsvéti útra</i>	86
FASTOPEN <i>állományok gyors megnyitása</i>	87
SET <i>környezeti változók beállítása</i>	88
Az állományok megjelenítése különböző helyeken	88
Állományok átirányítása	89
Az állományok megjelenítése	92
TYPE <i>húsvéti állományok tartalmát</i>	92
Szűrő parancsok	93
✓ MORE <i>húsvéti állományok tartalmát</i>	93
SORT <i>szabványos</i>	93
FIND <i>húsvéti fájlban</i>	96
Az állományok megjelenítése nyomtatón	99
PRINT <i>nyomtatás</i>	100
GRAPHICS <i>grafikus képernyő megjelenítés</i>	104
Bepillantás egy batch-fájlba	105

*Alattó:**Minden ismeretnek van alapja,
ami nélkül csak kártyavárat építhetsz,
de a jövődet nem.**(Morafi)*

Olvasd el, érdemes !

Ezt a könyvet azoknak ajánlom, akik még nem foglalkoztak a számítástechnikával, és szeretnék elsajátítani a számítógép kezelésének alapfogásait. Ajánlom elsősorban diákoknak, akik még a szakkönyvek sajátos nyelvezetét nem értik, és emiatt a számítástechnikát nehezen tudnák elsajátítani. Azt azonban el kell fogadnunk, hogy a számítógép nyelve az angol, és minimális angol nyelvismeret megszerzése nélkül ne is várja el senki, hogy értelemszerűen meg fogja tudni tanulni, a számítógépet a saját céljainak megfelelően hasznosítani. A könyvben minden üzenetet, amit a számítógép küld nekünk, magyarul is közlök idézőjelek között. Ez közel áll a "nyersfordítás" -hoz, de ahhoz közel áll azért, hogy ellenőrizhető legyen, és az angol szavak megtanulásában segítséget nyújtson. A parancsok paraméterei általában értelmes angol szavak rövidítései vagy kezdőbetűi. Emiatt a paraméterek ismertetésekor, ahol csak lehet zárójelben ismertetem azt az angol szót, amiből eredeztetni lehet a paraméter betűjelét. Példákban sem szűkölködik a könyv, amelyek pontosan tükrözik a számítógépben a parancsok végrehajtásának a menetét. Ezzel is segítségére kívánok lenni azoknak, akik otthon számítógép nélkül is tanulnak. Viszont a leírt példákat, és saját gondolatainkat mindenképpen próbáljuk ki a számítógépen is, mert csak úgy lehet megtanulni a számítástechnikát, ha a számítógépen tapasztaljuk meg, mit hogyan kell használni. Jó tanulási módszer az, hogy az otthon megszerzett elméleti ismereteket, másnap számítógépen kipróbáljuk, és ha működik olyan sikerélményben lesz részünk, aminek hatására jobban rögzül bennünk a megszerzett ismeret. A könyv nincs szabályos fejezetekre osztva azért, mert minden fejezetben lenne olyan parancs, amit jobb ha csak később, több ismeret birtokában próbálunk ki. A könyv olyan sorrendben halad végig a parancsokon, ahogyan azokkal egy kezdő találkozik. A megtanult parancsot fel tudjuk használni a következő parancsok jobb bemutatására, és ha felmerül eközben egy új parancs, akkor biztos az lesz a következő. Könyvem célja, hogy megkönnyítsem a kezdők számítástechnikai ismereteinek mielőbbi megszerzését. Ezt az általam megismert sok-sok - lexikális ismereteket adó - MS-DOS témájú könyvről nem mondhatom el. Azok a könyvek viszont a megszerzett ismeretek tökéletesebb elsajátításához nyújtanak sok segítséget. Remélem könyvem eléri célját, és sokakat segít hozzá az áhított számítógépkezelési ismeretek elsajátításához.

Kelt: Győr, 1993. július

Móricz Attila sk.

Alapismeretek

A számítógép felépítése, a fizikai teste (HARDWARE)

A személyi számítógépek (PERSONAL COMPUTERS) nagy fejlődésen mentek át az utóbbi néhány évben Magyarországon. A jelenlegi PC-XT, PC-AT gépek sokban különböznek a kisebb otthoni számítógépektől (pl.: COMMODORE 64). Most csak az elterjedtebb asztali típusokat tárgyaljuk, a LAPTOP, NOTEBOOK és társait nem. A számítógép egy tetszőleges kiépítésű, dobozba zárt főegységből és a hozzá kapcsolódó ki- és bemeneti egységekből áll. Fő erénye az, hogy egyes részek meghibásodása esetén, nem az egész számítógépet kell javítani, hanem csak a hibás kártyát kell kicserélni. Minden része cserélhető más, esetleg jobb típusra is. Például egy 286-os 12 MHz -es számítógépből, az alaplap cseréjével egy 386-os 40 MHz-es gépet varázsolhatunk, ami nagyobb memória kiépítését teszi lehetővé, vagy a monitor kártya cseréje után egy ahhoz illő monitort csatlakoztathatunk.

Mi van a dobozban:

- TÁPEGYSÉG: ami a működéshez szükséges áramot adja.
- ALAPLAP: amin a memória, a processzor,...stb van. Erre csatlakozik a többi áramköri kártya is.)
- I/O KÁRTYA: (INPUT/OUTPUT) ki- és bemeneti műveleteket bonyolító kártya. LPT , COM csatlakozási helyek.
- LEMEZVEZÉRLŐ: (CONTROLLER) ami két hajlékony lemez és két merev lemez vezérlését tudja ellátni. Újabban már az I/O KÁRTYÁVAL egybeépítik.
- HAJLÉKONY LEMEZT MEGHAJTÓ EGYSÉG: (FLOPPY DRIVE) amibe 3,5" vagy 5,25" méretű lemezt lehet betenni.
- MEREVLEMEZEKET MEGHAJTÓ EGYSÉG: (WINCHESTER) aminek ma már több fajtája is van, de abban azonosak, hogy több lemez, kötegben helyezkedik el. A különbségek a lemezek, illetve az egész egység mozgathatóságában van.
- MONITORT MEGHAJTÓ KÁRTYA: ami lehet MONO, CGA, EGA, VGA, .. Színben és felbontásban különböznek egymástól.

Ennyi is elég lenne, de lehet még:

- Szalagos adatrögzítő eszköz (STREAMER) és a kártyája.
- CD lemezes adattároló egység (CD-ROM) és a kártyája.
- Telekommunikációs adatátviteli eszköz (MODEM).
- Grafikus képleolvasó (SCANNER) vezérlő kártyája.
- Zenei hangot visszaadó-digitalizáló kártya (pl.: SOUND BLASTER).
- Grafikai alkalmazást gyorsító kártya (ACCELERATION).
- Monitor jel - Video jel - TV jel átalakító kártya.

A doboz előlapján szokott lenni:

- KI/BEKAPCSOLÓ GOMB (lehet hátul is).
- RESET GOMB (számítógépben futó programot megszakító, újraindító gomb).
- TURBO GOMB (a processzor két sebessége közül választ).
- SEBESSÉGET DIGITÁLISAN KIJELZŐ EGYSÉG.
- BILLENTYŰZET KULCSOS ZÁRJA.

A kapcsolódó ki- és bemeneti egységek

Billentyűzet: Többféle létezik, 86, 101, 102 gombos lehet, a billentyűk elrendezéséről és jelentéséről majd később részletesen beszélünk. A billentyűzet a gép hátulján az alaplaphoz csatlakozik.

Egér: Olyan adatbeviteli eszköz, amiben egy golyó relatív elfordulásával arányosan mozog egy nyílkurzor a számítógép képernyőjén. Van rajta kettő vagy három gomb is, amivel kijelölhetjük az aktuális pozíciót. Az egér (MOUSE) az I/O kártya COM (COMMUNICATIONS) feliratú aljzataira csatlakozhat.

Követő golyó: Hasonló elven működik, mint az egér, de itt nem egy "dobozt" kell mozgatni amiben a golyó van, hanem a golyót kell elforgatni a "doboz"-ban, és kattintani a billentyűkkel. A gép COM aljzataira csatlakozhat.

Track Ball: Működési elve azonos az egérével, de ez egy golyóstollhoz hasonló formájú eszköz, amivel egy tollhoz hasonlóan tudunk rajzolni.

Monitor: A számítógépben lévő monitort meghajtó kártyához csatlakozik, azzal egyező típusú lehet. A számítógépbe beírt karakterek jelennek meg rajta, illetve a számítógép által küldött üzenetek, és a futó programok eredményei.

Nyomtató: Az adatokat papíron megjelenítő eszköz, amelynek szintén több típusa van. A legelterjedtebb a mátrix nyomtató, de a tintasugaras és lézer nyomtatók is egyre nagyobb teret hódítanak. A régebbi elavult típusokat nem tárgyaljuk. A nyomtatók az I/O kártya LPT (LOCAL PRINTER - helyi nyomtató) aljzataira csatlakozhatnak.

Rajzoló gép: (PLOTTER) Ami csőtollakkal rajzolja ki a grafikus ábrát, több színben. A nyomtatókkal ellentétben nem soronként írja ki az adatokat, hanem a papír és a csőtoll együttes mozgásával rajzolódik ki a kívánt ábra.

A számítógép bekapcsolás után (SOFTWARE)

A számítógépünk bekapcsolása után, egy memória IC-be égetett program indul el, ami különböző teszteléseket végez el a gép alapvető részein. (Memóriateszt, lemez meghajtók tesztje...) Utána az egyik lemezegységben levő lemez első szektorától kezdve betölti az ott található, rendszer működését biztosító fájlokat (programokat), majd átadja neki a vezérlést. Ami (esetünkben) ilyenkor betöltődik az a DOS (DISK OPERATION SYSTEM - Lemez Operációs Rendszer). Az **operációs rendszer** azon programok együttese, amelyek a számítógép működését alapvetően biztosítják, és lemezzel töltődnek be a számítógép bekapcsolásakor. Ez azért lemezes, mert az otthoni számítógépekkel (C-64) ellentétben nem állandóan bent lévő (memória IC-be égetett) programokról van szó, hanem lemezzel betöltött programokról. Aminek az a nagy előnye, hogy folyamatosan megújítható, megváltoztatható, hátránya hogy megsérülhet, vírusos lehet, ami a rendszer működését rontja, esetleg megbénítja. Most vegyük alapul azt, hogy van egy beüzemelt rendszerünk, ami kész a parancsok fogadására.

A számítógép bekapcsolás után egy **öntesztet** végez el a megadott adatok alapján. (SETUP - aminek állítása az indulás után a **[DEL]** billentyű lenyomása után végezhető el, de kezdőknek, még véletlenül sem ajánlom, hogy ezeket az adatokat megváltoztassák!)

Először egy **memóriatesztet** végez el, amit a képernyő bal felső sarkában futó számsorral mutat nekünk. Ha nem egyezik a számítógépben lévő memóriaegységek száma és a programból beállított (SETUP) érték, akkor azt kijelzi nekünk. Ha valamilyen hibája van valamelyik memória cellának, akkor azt is tudatja velünk (MEMORY ERROR).

A **billentyűzet tesztje** következik, amit a felvillanó ledek (pici zöld kijelző) jeleznek. Ha a számítógép előlapján lévő kulcsos zár be van zárva, akkor felszólít a hiba kijavítására (KEYBOARD LOCKED). Lehet azonban az is, hogy a billentyűzet zsinórja van kicsúszva a gép hátuljából.

Majd a **hajlékony lemezes meghajtó** (FLOPPY DISK DRIVE), és a **merevlemez** (WINCHESTER) **tesztje** következik, aminek hatására a meghajtó motorja felpörög, és a led kijelzője felvillan.

Ezek után **kírja** a számítógép felépítését alapvetően meghatározó adatokat (processzorok, memória méret, kapcsolódó egységek...).

System Configuration (C) Copyright 1985-1989 American Megatrends Inc.,			
Main Processor	: 80286	Base Memory Size	: 640 KB
Numeric Processor	: None	Ext. Memory Size	: 1024 KB
Floppy Drive A:	: 1.2 MB, 5¼"	Hard Disk C: Type :	33
Floppy Drive B:	: None	Hard Disk D: Type	: None
Display Type	: VGA or EGA	Serial Port(s)	: 3F8, 2F8
ROM-BIOS Date	: 04/30/89	Parallel Port(s)	: 278
Bank 0/1 DRAM Type	: 256K DRAM	Bank 0/1 DRAM W/S	: 0 W/S
Bank 2/3 DRAM Type	: 256K DRAM	Bank 2/3 DRAM W/S	: 0 W/S
Memory Interleave	: Enabled	Processor Clock	: CLK2IN
Shadow C000H,16K	: Disabled	Bus Clock	: ATCLK
Shadow C400H,16K	: Disabled	DMA Clock	: SCLK
Shadow F000H,64K	: Disabled		

Miután a tesztelésekkel végzett, **betölti a DOS-t**. Ha floppy lemez van bent a meghajtóban, és a meghajtó ajtaja is csukva van (5¼"), akkor az **A:** nevű hajlékony lemeze (FLOPPY) fogja a DOS-t betölteni. Ha a floppy meghajtóban nincs lemez vagy nincs lecsukva az ajtaja (5¼" -osnál fontos!), akkor a **C:** nevű merevlemeze (WINCHESTER) fogja a rendszert betölteni. Azért ez a sorrend, mert ha a merevlemezen nincs rendszer vagy hibás, akkor a másik számítógépen készített rendszerlemezzel indítható ez a gép is. Ez a sorrend egyes számítógépeknél fel is cserélhető. Ha a számítógépünkben nincs merevlemez és a hajlékony lemezes meghajtóban sincs rendszerlemez, és nincs hálózatra kötve a számítógép, akkor egy hibaüzenettel le fog állni a számítógép:

Non-system disk or disk error

Replace and press any key when ready

("Nincs rendszerlemez vagy lemezhiba

Cseréld ki és nyomj meg egy billentyűt, akkor kész."

Ha van betölthető rendszer, akkor a következő állományok kerülnek betöltésre (melyek a DOS rendszer magját alkotják):

A betöltésre kerülő fájlok (FILE-PROGRAM):

Név+kiterjesztés	méret	dátum	idő
IO SYS	33430	91-04-09	5:00
MSDOS SYS	37394	91-04-09	5:00
COMMAND COM	47845	91-04-09	5:00

✓ Az IO.SYS program biztosítja az adatok alapvető ki-és bemeneti (INPUT-OUTPUT) áramlását. A SYS kiterjesztése jelzi számunkra azt, hogy ez a fájl, rendszerfájl (SYSTEM PROGRAM). Az MSDOS.SYS program a DOS rendszerhívások alprogramjait tartalmazza. A COMMAND.COM a parancsokat feldolgozó program. Ez a fájl tartalmazza a *belső parancsokat*, végrehajtja a parancsfájlokat, betölti a külső programokat, és futtatja (elindítja) azokat.

(Valójában a következő események zajlanak le: Az önteszt után a számítógép betölti a lemez első szektorában (BOOT SECTOR - betöltő szektor) található információkat, melyekből megtudja, hogy melyek az operációs rendszer betöltendő állományai. Utána - ezek ismeretében - betölti az IO.SYS programot, és átadja neki a vezérlést. Utána már ez a program végzi a további teendőket. Eddig a BIOS bonyolította le a műveleteket, ami egy olyan memóriaegység, amibe be van égetve egy program, ami állandóan bent van, és soha nem "felejt el" (ROM). Az IO.SYS ezután megkeresi a betöltő egység (!) gyökérkönyvtárában a CONFIG.SYS fájlt és végrehajtja a kapott utasításokat. Ezután az MSDOS.SYS fájl betöltése következik, majd a COMMAND.COM, ami meg is kapja a vezérlést. Ez a program megkeresi a betöltő egység (!) gyökérkönyvtárában az AUTOEXEC.BAT nevű batch-fájlt (lásd utolsó fejezet!) és végrehajtja soronként a kapott parancsokat. A parancsfájl lefutása után a számítógép várja a további parancsainkat.)

A számítógép használata

Az előző oldalakon ismertetésre került, hogy miként indul a számítógép, mi zajlik le benne addig, amíg várakozó álláspontra nem helyezkedik. Ilyenkor a mi parancsainkra vár a számítógép. Ezt a várakozását a kurzor (_■) villogása jelzi nekünk. A kurzor mutatja meg azt a helyet, ahová a kiírandó karakterek kiírásra kerülnek. A rendszer betöltése, indulása általában minden bekapcsoláskor ugyanúgy zajlik le mindig. Viszont célszerű nyomon követni, és a szokatlan kijelzésekre fokozottan odafigyelni, mert sok apró, hibának tűnő esemény ered onnan, hogy nem vesszük figyelembe a számítógépünk jelzéseit, üzeneteit.

Most vegyük sorra mi fog lezajlani a számítógép használata során. A kurzor villog a sor elején, és arra vár, hogy valamit beírjunk neki. Két dolgot írhatunk be, parancsot és programnevet. A parancs szó alatt több mindent értünk. A *belső parancs* az, amit a memóriába betöltött COMMAND.COM feldolgoz és végrehajt. A *belső parancsot* tetszőleges helyről kiadhatjuk, mindig végre lesz hajtva, míg a *külső parancs* egy programnak a neve, ami valamelyik könyvtárban található. Ha nem abban a könyvtárban állunk, ahol az adott program van, vagy más egyéb módon nem tudatjuk a számítógéppel, hogy hol a programunk, akkor nem fogja tudni végrehajtani a DOS. Ekkor a következő üzenetet küldi:

Bad command or file name - "Rossz parancs vagy fájlnev"

A *külső parancs*, az tulajdonképpen egy futtatható program, amit általában a "DOS" nevű könyvtárban tartunk. Ebből következik, hogy *külső parancsnak* nevezzük azokat a

futtatható programokat, amelyeket az operációs rendszerhez mellékel a szoftvert gyártó cég. Ezek a programok (EXE és COM kiterjesztésűek) segítik elő a mindennapi munkánkat. Vannak azonban olyan programok is, amelyek nem az operációs rendszer részei, de annak használatát esetleg jobban elősegítik, mint a kapott szoftverek. Akkor persze azokat is használhatjuk ugyanúgy, mintha DOS parancsok volnának, sőt sok esetben jobban megoldanak egy adott feladatot, mint a DOS külső vagy belső parancsai. A parancsokkal a számítógépet utasítjuk valamire. Van egy feladat, amit meg kell oldanunk, és megkeressük azt a parancsot, amelyik ezt meg tudja oldani. Kommunikálunk a számítógéppel, parancsot adunk neki, az pedig megpróbálja végrehajtani. Közben információkat ír ki nekünk, amivel tájékoztat bennünket a parancs végrehajtásának menetéről.

Ha egy olyan programmal találkozunk, amelyiknek a funkcióját nem ismerjük, akkor kérhetünk *segítséget* (HELP) a számítógéptől. A DOS 5.00 verziójától kezdve minden parancsról kérhetünk segítséget a ? paraméter megadásával. Ekkor nem hajtódik végre a parancs, hanem kiírja a használatához szükséges paramétereket, és a feladatát. Hasonló feladatot lát el a HELP parancs is:

```
D:\DOS> help diskcopy
Copies the contents of one floppy disk to another.

DISKCOPY [drive1: [drive2:]] [/1] [/V]

/1    Copies only the first side of the disk.
/V    Verifies that the information is copied correctly.

The two floppy disks must be the same type.
You may specify the same drive for drive1 and drive2.

D:\DOS> diskcopy /?      { ugyanaz a hatás, mint az előbb }
:
:
D:\DOS>_
```

A külső parancsok gyakorlatilag a DOS -hoz tartozó programok (FILEs - fájlok) együttese. Egy ilyen fájl betöltődik a számítógép memóriájába, végrehajtódik és törlődik a memóriából. Ha egy külső parancsot nem hajt végre a számítógép, akkor az vagy nincs az *aktuális könyvtárban* vagy a *keresési úton* (PATH), vagy hibásan gépeltük be. Ugyanis a program első megközelítésben csak abból a könyvtárból indítható, amelyikben van. (Gondoljunk csak bele, mennyi időbe telne, ha minden parancsunknál a gép minden könyvtárat végignézne, hogy ott van-e a keresett parancs.)

Azonban segítségünkre van a keresési útvonal (PATH), ami biztosítja, hogy ne csak azokat a fájlokat használhassuk, amelyek az aktuális könyvtárban vannak. Ha egy fájlt nem talál az aktuális könyvtárban, akkor megnézi azokat a könyvtárakat, amelyek benne vannak az *elérési útban*, a PATH környezeti változóban.

Tulajdonképpen egy mesterséges intelligenciával rendelkező géppel kommunikálunk. Mi a billentyűzeten keresztül hozzuk a tudomására a parancsainkat (lehetséges már beszéd útján is), az pedig vagy a képernyőre írja ki a válaszait, vagy elvégzi válasz nélkül a kapott parancsot, vagy hang útján válaszol. (Lásd zenei illetve játék programok.) Vannak már olyan szoftverek is, amelyek a program által kiírt üzeneteket szóban is elmondják, így emelve a programok színvonalát, illetve segítve a vakok munkáját.

A parancsokon kívül tetszőleges feladatot ellátó *programokat* is futtathatunk. Egy *programot futtatni*, annyit jelent, hogy azt betöltjük a számítógép memóriájába, és átadjuk neki a vezérlést. Ezek után minden úgy történik a számítógépben, ahogy azt a programban lévő utasítások vezérlik. Ha a programban valami nincs megszerelve rendesen, akkor bizony előadódhat olyan helyzet, hogy a program végtelen ciklusba kerül (azaz például folyamatosan akar lemezről olvasni, ha nincs is lemezegység), és nem tud kikerülni onnan. Akkor mondjuk, hogy "lefagyott" a programunk. Ha viszont a **[CTRL] + [C]** vagy a **[CTRL] + [BREAK]** billentyűk sem segítenek, akkor mondjuk, hogy a számítógépünk is "lefagyott". Ekkor csak az **[ALT] + [CTRL] + [DEL]** billentyűk együttes lenyomása, vagy a **[RESET]** gomb benyomása segít, aminek hatására a számítógépünk újra indul. Olyan helyzet áll elő, mintha most kezdenénk a munkát.

Jobb esetben a programunk sikeresen lefut, átadja a vezérlést az operációs rendszernek (egész pontosan a parancsfeldolgozóknak, a COMMAND.COM -nak), ami felszabadítja az előző program által lefoglalt memóriát (nem törli ki a memóriából), és kiírja a készenléti jelet, várva a további parancsainkat.

Vannak olyan programok is, amelyek nem "törlődnek" a memóriából, hanem állandóan bent maradnak, és folyamatosan foglalják a memória egy részét. Ezek a programok folyamatosan befolyásolják a munkánkat, vagy annak körülményeit, vagy csak egy adott billentyűkombinációra adódik át nekik a vezérlés.

Ezeket a programokat nevezzük *rezidens programoknak* (TERMINATE, but STAY RESIDENT - A programnak vége, de állandóan bent marad). Ilyen lehet például, két billentyű együttes lenyomására előjövő számológép vagy szótárprogram, ami használat után "visszabújik a memóriába", és csak a legközelebbi "hívásra" jön elő.

Számítástechnikai alapfogalmak

A számítógép memóriájában lévő program végrehajtását végzi el a *processzor*, ami a számítógép egyik legfontosabb egysége. Alapvetően meghatározza a gép sebességét, és az elvégezhető műveleteket. A számítógépeket a processzor *típuszáma* alapján szoktuk megkülönböztetni. A 8086 és 8088 -as típusú gépeket XT -nek nevezzük. (XT - EXTENDED TECHNOLOGY - kiterjesztett technológia.) A 80286, 80386, 80486 -os processzorú gépeket pedig AT -nek nevezzük. (AT - ADVANCED TECHNOLOGY - előrehaladott technológia). A számítógép sebességét is behatárolja a processzor típusa. Egy 80286 -os gép sebessége 12 - 20 MHz -ig terjed. A számítógép "sebességét", mint láthatjuk *megahertz*ben mérjük. Minden frekvencia egy impulzust ad a processzornak, mire az végrehajt egy műveletet. Ebből következik, hogy minél nagyobb a frekvenciája egy processzornak, annál több műveletet tud végrehajtani egy időegység alatt. Viszont azt is figyelembe kell vennünk, hogy nem minden a sebesség, mert egy 80386 -osnak az utasításkészlete is bővebb, és egyéb kiépítettsége miatt, gyorsabban hajtja végre az utasításokat, mint egy 80286 -os. (Ezekről részletesebben a "*Hardver alapismeretek*" című könyvemben olvashattok.) Az adatokat, a programokat, a számítógép a *memóriájában* tárolja. Minden memóriarekesznek van egy címe (sorszama), és egy *bájt*ot tárolhatunk benne. Egy bájt (BYTE) értéke 0 - 255 -ig terjedhet, és nyolc *bit*ből (BIT - BINARY DIGIT - egy kettes számrendszer- beli szám) áll. A bájt a legkisebb megcímezhető elem. A bit értéke 0 vagy 1 lehet, mert így lehet a legkönnyebben eltárolni egy adatot egy memóriaegységben (0 ≡ nincs áram, 1 ≡ van áram).

Helyi értékek: (kettő hatványai)	7	6	5	4	3	2	1	0
Egy bájt =	1	1	0	1	0	1	0	1
Az értékek:	128	64	32	16	8	4	2	1

Egy memória rekesz tartalma minimum nulla (0000 0000₂)

maximum 255 (1111 1111₂) ≡ (128+64+32+16+8+4+2+1 = 255₁₀)

Az értéke a példában:

$$11010101_2 = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 \cdot 2^7 + & 1 \cdot 2^6 + & 0 \cdot 2^5 + & 1 \cdot 2^4 + & 0 \cdot 2^3 + & 1 \cdot 2^2 + & 0 \cdot 2^1 + & 1 \cdot 2^0 \\ \hline \end{array}$$

$$11010101_2 = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 128 & 64 & 0 & 16 & 0 & 4 & 0 & 1 \\ \hline \end{array}$$

$$11010101_2 = 213_{10}$$

A memóriarekeszben tárolt bájt mind a 256 elemének jelentése van. Ha egy végrehajtandó program kódjai, akkor az adott bájt lehet egy utasítás kódja, amit a processzor hajt végre, vagy egy adat, amin a műveletet végzi a processzor. Ha ezek az adatok a képernyőmemória területre íródnak, akkor megjelennek a képernyőn is. A bájt mind a 256 elemének van egy pontokból álló képe, ami lehet betű, számjegy, írásjel vagy egyéb jel, összefoglaló néven **karakter**. Hogy melyik kódhoz melyik karakter van hozzárendelve, azt az **ASCII kódtáblázatok** tartalmazzák. Több is van belőle, több országnak is van saját kódtáblázata, ezeket nevezzük **kódlapok**nak. Viszont minden kódtáblázatban vannak olyan karakterek, amelyek kódjai azonosak.

kódok	jelentéseik
0-31	általában vezérlő kódok
32-47	írásjelek
48-57	0-9 számjegyek
58-64	további írásjelek
65-90	A-Z nagybetűk
91-96	további írásjelek
97-122	a-z kisbetűk
123-127	további írásjelek
128-255	ékezetes betűk, görög betűk, grafikus karakterek, országok sajátos nemzeti betűi, jelei, ... <i>stb</i>

A kódlapok a 0-127-ig lévő kódokban megegyeznek, csak a 128-255-ig terjedő részen különböznek.

A nyomtatók is ezeket a kódokat ismerik, viszont itt a 0-31 lévő vezérlőkódok értelemszerűen a nyomtatót utasítják.

A 32-255-ig lévő karakterek pedig általában az amerikai kódtábla karaktereit ismerik. Ez némely nyomtatónál nem változtatható meg, de vannak olyan nyomtatók is, amelyeknél megoldható a kódlap váltása. Egyes karakterek képeit viszont magunk is megváltoztathatjuk, a megfelelő parancsok segítségével. A kódlapok fogalma több helyen is elő fog fordulni, általában ott, ahol ékezetes karakterek használatával kapcsolatos a parancs.

A memóriarekeszben tárolt bájt, lehet egy programban utasítás vagy adat is. A processzornak ugyanis van egy utasítás készlete, ami alapján a programokat végrehajtja. Az utasítások általában az adatbájtokat dolgozzák fel.

Programokat különböző programozási nyelveken írhatunk. Mindegyik nyelv tartalmaz sok utasítást, amit megfelelő fordító programok segítségével, le tudunk fordítani **gépi kódra**. Ezek azok a kódok, amelyek a processzort utasítják a program végrehajtására. Ha az **Assembly programozási nyelv** utasításaiból összeállított programot, lefordítjuk program kódokra, akkor közvetlenül a processzort utasítjuk, ezért ezen a nyelven írt program lesz a leggyorsabb. Hátránya viszont az, hogy egy egyszerű művelet végrehajtására is (pl.: képernyőre írás) hosszú programot kell írni, mert az Assembly nyelv csak a legelemibb utasításokat tartalmazza. Ha ezen a nyelven megírjuk azokat a programokat, amelyekre minden program írásakor szükség lehet, akkor ha programot írunk, már csak ezeket az új "utasításokat" használjuk. Ezen az elven alakult ki sok programozási nyelv, például a "C", a "PASCAL", a "BASIC", ...stb. Ezek a nyelveken írt programok már lassabbak lesznek egy kicsit, mert nem csak azt a műveletet végzi el egy utasítás, amit mi szeretnénk, hanem többet is. A megírt programjainkat a számítógép kikapcsolása előtt "**el kell mentenünk**", azaz a gép memóriájából ki kell másolni, valamilyen **mágneses adatrögzítő eszközre**. Ilyen például a **hajlékony lemez, merevlemez** vagy a **szalagos adatrögzítő**. Ezek az eszközök tartósan eltárolják a programjainkat úgy, hogy a számítógép kikapcsolása után sem vesznek el az adataink, a programjaink. A következő alkalommal, mikor újra bekapcsoljuk a számítógépünket, akkor **betölthetjük** az említett egységről a programunkat, és folytathatjuk tovább a programozást. A számítógép minden program felírásakor először bejegyzí a megfelelő helyre a nevét és egyéb adatait, majd kimásolja a lemezre a memóriából a program bájtjait. A lemezen tárolt programot és minden állományt, **fájlnak** (FILE) nevezzük. Minden fájlt a **nevével** és a **kiterjesztésével** azonosítunk a lemezen. Az adatrögzítő eszközön meghatározott rendben tárolja a gép a programjainkat. A rendszerfájlok is itt helyezkednek el. A számítógép bekapcsolás után egy ilyen egységről tölti be a rendszer működéséhez szükséges programokat, más néven fájlokat. A leggyakrabban használt mágneses adatrögzítő eszközök a merevlemez (WINCHESTER) és a hajlékony lemez (FLOPPY DISK). A szalagos adatrögzítőt nagy tömegű adatok eltárolására szoktuk használni, ezért ritkábban használjuk, mint a floppy lemezt.

A fájloknak több adatát tárolja a számítógép. A legfontosabb azonosítója a fájlnak a **neve**, ami maximum nyolc karakter lehet, és az **elválasztó pont** (.) utáni **kiterjesztése**, ami három karakter hosszú lehet. A fájlnévhez tartozó kiterjesztés el is maradhat, de ez jelzi nekünk, hogy milyen típusú fájlról van szó.

Betölteni, futtatni egy programot csak akkor lehet, ha annak a kiterjesztése COM, EXE, BAT.

Nézzünk meg néhány általában használatos kiterjesztést:

COM - Assembly programozási nyelven írt, lefordított, futtatható program, a parancs szóra utal (**COMMAND**).

EXE - A másik futtatható fájl (**EXECUTABLE**), ami közvetlenül indítható a DOS-ból.

BAT - Kötegelt parancsfájl (**BATCH** - csomó,köteg), amiben sok parancs van felsorolva és ezeket a **BAT** fájl indításával sorban végrehajtja a számítógép. (Bővebben lásd az utolsó fejezetben.)

ASM - **Assembly** nyelvű forrásprogram, lefordítva **EXE** vagy **COM** típusra futtathatóvá válik.

BAS - **BASIC** programozási nyelven írt program, csak a BASIC fejlesztő környezetben futtatható, vagy DOS -ban, ha le van fordítva EXE formátumra.

PAS - **Pascal** nyelven írt program.

C, CP, CPP - **C** nyelvű forrásprogram kiterjesztései (**P** - Plussz).

DAT - Adatokat tartalmazó fájl (**DATA** - adat).

DBF - **dBASE** fájl, adatfájl.

TXT, DOC - Szöveges fájlok (**TEXT, DOCUMENTUM**).

Ezen kívül még nagyon sok fajta kiterjesztés létezik, ami az adott program típusára utal, ezért fontos, hogy ne keverjük össze a fájlnevek kiterjesztéseit. Mi is találhatunk ki újabb kiterjesztéseket, de lefuttatni csak a .COM, .EXE, .BAT fájlokat lehet. Például a megírt leveleinknek adhatunk .LEV, az érkezett leveleinknek .ERK (vagy ÉRK) kiterjesztést is.

Vannak olyan parancsok, ahol több fájlra is lehet vonatkoztatni a kiadott parancsot. Ezt nem úgy értem, hogy felsoroljuk az adott fájlok neveit, hanem a fájl nevében **helyettesítő**, úgynevezett **dzsókerkaraktereket** használunk. Ezek a '* és a '?'. A CSILLAGKARAKTER, minden utána következő karaktert helyettesít az adott helyen (pl.: fájlnevében, vagy kiterjesztésben), míg a KÉRDŐJEL csak azt az egy karaktert, aminek a helyén áll.

Például:

- minden fájl meghatározása: '*.*'
- minden 'TXT' kiterjesztésű fájl: '*.TXT'
- minden fájl ami 'A' -val kezdődik, négy karakter hosszú és 'DOC' a kiterjesztése: 'A???.DOC'

Ezekre például a DIR parancs használatánál lehet szükségünk, ahol azt adjuk meg, hogy melyik fájlokat listázza ki a számítógép. Ha nem akarjuk az összes fájlt kiírni a képernyőre, akkor használjuk a dzsóker karaktereket (* ?).

Például:

DIR *.??A - minden fájl, aminek a kiterjesztése 'A' betűre végződik. Nem használhatjuk például a **'DIR *.*A'** formát, mert a csillag után már nem vesz figyelembe semmilyen név karaktert a DOS.

A fájloknak a számítógép tárolja még a **méretét**, azaz hány bájt hosszú, a keletkezésének vagy a legutolsó módosításának a **dátumát** és **idejét**, és néhány olyan **jelzőbitet** (ATTRIBUTUM), ami a fájlokat bizonyos csoportokba sorolja. Bővebben az ATTRIB parancsnál olvashatunk.

A fájl dátuma és ideje sokszor igen fontos információ lehet a számunkra, ezért nagyon fontos, hogy a számítógépben mindig az aktuális dátum és a pontos idő legyen eltárolva. Ha például olyan számítógépünk van, amiben nincs elem, ami az időt tárolja, akkor mindig pl.: **1980.01.01** lesz a dátum az idő pedig: **00:00:00.00** - ról indul a számítógép bekapcsolásakor. Ebben az esetben, viszont mindig minden programunknak egyforma lesz a dátuma. Bővebben lásd a DATE illetve a TIME parancsoknál.

A számítógép részei

Most vizsgáljuk meg közelebbről a számítógépünket, amivel az elkövetkezőkben dolgozni fogunk.

Monitor

Többféle típus létezik, *színben* és *felbontásban* különböznek egymástól. A *méretük* általában a képcső átlója szerint 14" (1" col, inch, hüvelyk 2,54 cm), de van kisebb és nagyobb típus is. A kisebbeket egyre többet boltokban lehet látni, pénztárgépnek használt számítógép kijelzőjeként, nagyobbakat kiállításokon használnak, hogy nagyobb távolságból is jól látható legyen mindenki számára.

A monitor, a számítógépben lévő *monitort meghajtó kártyához* csatlakozik, áramot pedig típustól függően vagy a számítógép tápegységétől kap vagy saját 220 Voltos csatlakozó dugója van.

A monitor kezelőszervei

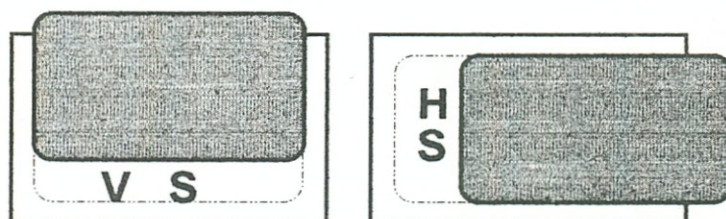
fényerő : A monitor fényének erejét (BRIGHTNESS CONTROL) állíthatjuk be vele, a helyiség megvilágításától és a szemünktől függően.

kontraszt : (CONTRAST) A fényes és tompa fényű karakterek közötti különbséget emeli ki. Fényerő emelésre ne használjuk, mert az a kontraszthatás rovására megy.

Ki/Bekapcsoló gomb : (ON/OFF ; 0 / 1) A monitorba jövő áramot kapcsolja. Ha hosszabb ideig nem dolgozunk a számítógépen, de nem akarjuk kikapcsolni, a monitort azért még lekapcsolhatjuk. Ugyanis ha sokáig világít egy adott pont, akkor ott az a pont beéghet a monitor képcsővébe, és akkor az a szöveg ott mindig meg fog jelenni.

A hátsó oldalán - egyes típusoknál elöl - még találhatunk különböző képméretállító potmétereket, de ezek ha be vannak állítva, akkor nem célszerű elállítani őket.

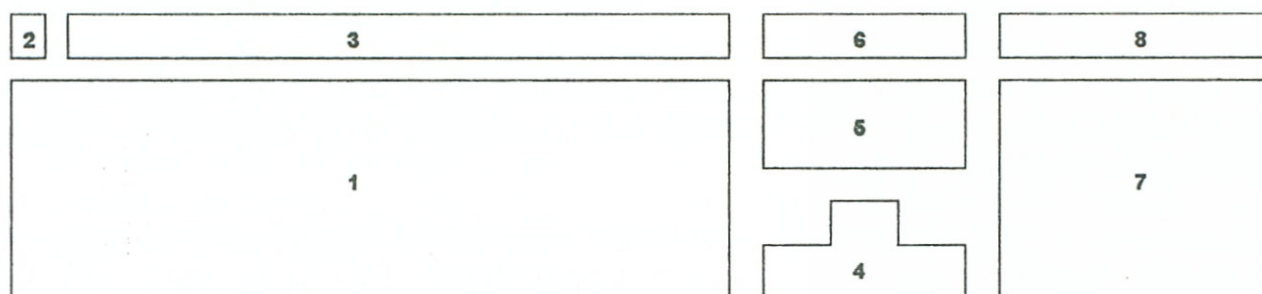
- VERTICAL SHIFT (Magassági képeltolás)
- HORIZONTAL SHIFT (Vízszintes képeltolás)



Billentyűzet

A **billentyűzet** az a része a számítógépnek, aminek segítségével az adatokat bevisszük a számítógépbe. Sematikus elrendezését mutatja a következő ábra:

101 GOMBOS BILLENTYŰZET:



A 101 gombos billentyűzet részei:

1.: Ez a része a billentyűzetnek hasonlít az írógéphez. Itt találhatóak a **számok**, **betűk**, **írásjelek** a fehér színű billentyűkön, valamint a **speciális jelentésű billentyűk** szürke színnel.



Speciális billentyűk

[ENTER]: A parancsok, adatok beírása után kell leütni, és a számítógép feldolgozza a bevitt adatokat. A funkciója megegyezik a számbillentyűzeten lévő **[ENTER]** billentyű jelentésével, a **[NUM LOCK]** állapotától függetlenül.

[BACKSPACE],[←]: Visszatörlésbillentyű, amivel a tévesen beírt karaktereket lehet letörölni, azt amelyik a villogó kurzor előtt van.

[TAB]: (Oda-vissza mutató nyilakkal is jelzik) Tabulálni, pozícionálni lehet vele, bizonyos programok használják, de a DOS nem.

[SHIFT]: Segédbillentyű, a számsornál a felülre írt jelek jelennek meg, a betűknél pedig a nagyokat írja ki. Minden billentyűvel kombinálható, és új jelentést adhat neki egy adott programban. Ha **[CAPS LOCK]** be van nyomva, akkor a nagybetűk helyett a kisbetűk jelennek meg.

[CTRL]: Segédbillentyű, néhány **[CTRL]** + **[betű]** kivételével nincs jelentése a DOS-ban, de egy programban a **[CTRL]** + **[bármely billentyű]**, új funkciót kaphat.

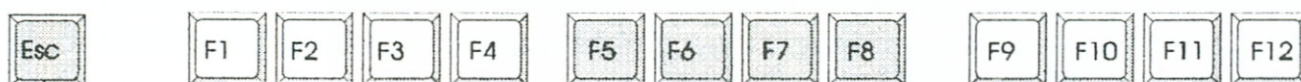
[ALT]: Segédbillentyű, néhány ASCII kód kivételével azok megjeleníthetők vele: **[ALT]** + a jobb oldali számokkal beírt kód segítségével. Egyéb jelentése nincs a DOS-ban, de más programban minden billentyűnek új jelentést adhat.

(A segédbillentyűket úgy kell használni, hogy lenyomjuk a megfelelő segédbillentyűt, és lenyomva tartjuk folyamatosan, majd lenyomjuk egyszer, röviden és határozottan, a másik betű- vagy más egyéb - billentyűt, és utána engedjük fel csak a segédbillentyűt.)

[CAPS LOCK]: Nagybetűzár, vagyis ennek lenyomása esetén a betűbillentyűknél a nagybetű fog megjelenni. Ebben az esetben kigyullad egy led a 7-es területen a "Caps Lock" felirat alatt, ami a bekapcsolt állapotot jelzi.

2. ESCAPE billentyű: **[ESC]** (menekülés)

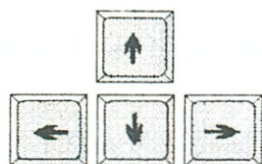
A DOS-ban nincs sok szerepe, a parancs sorban egy '^' jel jelenik meg, és nem hajtja végre a beírt parancsot.



3. Funkcióbillentyűk: **[F1]** - **[F12]**

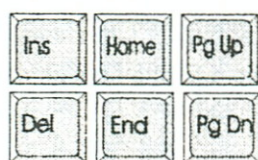
A DOS -ban csak az első ötnek van jelentése, de **[SHIFT]**, **[ALT]**, **[CTRL]** segédbillentyűkkel együtt, más programokban nagyon sok funkciót rendelhetünk hozzájuk.

4. Kurzormozgató-billentyűk

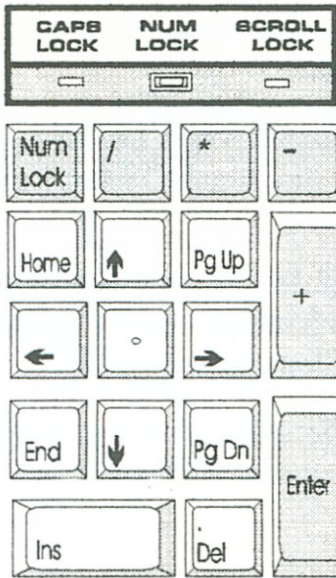


A megfelelő irányba mozgatják a kurzort. A DOS-ban nincs jelentésük, de más programokban a **[SHIFT]**, **[CTRL]** és **[ALT]** billentyűkkel új jelentést kapnak.

5. Kurzorvezérlő-billentyűk



[INSERT]: A parancssorban és más programokban is beszúrás módba kapcsol. Az utána leütött karakter a tőle jobbra álló szöveget nem írja felül, hanem eltolja. Általában a programokban ez az alapértelmezés. Funkciója megegyezik a számbillentyűzeten lévő **[0/INS]** billentyű jelentésével, ha a **[NUM LOCK]** nincs benyomva.



[DELETE]: A kurzor alatti karaktert törli le, és a tőle jobbra lévő karaktereket egyvel előrébb (balra) lépteti. Funkciója megegyezik a számbillentyűzeten lévő **[./DEL]** billentyű jelentésével, ha a **[NUM LOCK]** nincs benyomva.

[ALT] + [CTRL] + [DEL] vagy **[ALT] + [CTRL] + [DELETE]:** A rendszert tudjuk vele újra indítani, ami hasonló a számítógépünk ki- és bekapcsolásához, csak a gépet így jobban kíméljük. (Ilyenkor az *önteszt* nem hajtódik végre.)

[HOME]: A DOS-ban nincs jelentése, de máshol a kurzort a sor elejére mozgatja. Funkciója megegyezik a számbillentyűzeten lévő **[7/HOME]** billentyű jelentésével, ha a **[NUM LOCK]** nincs benyomva.

[END]: A DOS-ban nincs jelentése, de máshol a kurzort, a sor végére mozgatja. Funkciója megegyezik a számbillentyűzeten lévő **[1/END]** billentyű jelentésével, ha a **[NUM LOCK]** nincs benyomva.

[PAGE UP]: A DOS-ban nincs jelentése, de a szövegszerkesztőkben egyet lapoz felfelé, a szöveg eleje felé. Funkciója megegyezik a számbillentyűzeten lévő **[9/PGUP]** billentyű jelentésével, ha a **[NUM LOCK]** nincs benyomva.

6. Speciális feladatot ellátó billentyűk



[PRINT SCREEN] vagy **[PTR SC]:** Szöveges képernyő tartalmát nyomtatja ki, a hozzá csatlakozó nyomtatóra. Ha nincs nyomtatónk, akkor egy sípoló hanggal válaszol.

[SHIFT] + [PRINT SCREEN]: Ha a GRAPHICS.COM parancs el volt indítva, akkor a grafikus képernyő tartalmát is kinyomtatja.

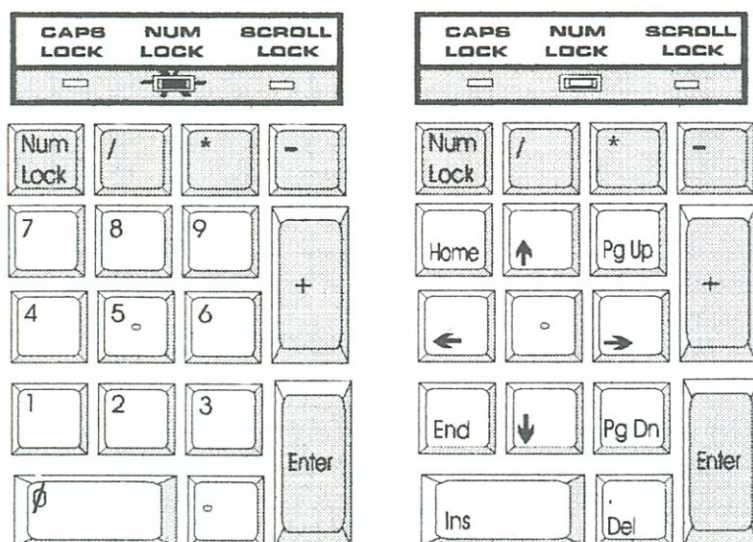
[SCROLL LOCK]: A DOS-ban nincs szerepe, leütésekor a 8-as területen lévő "Scroll Lock" alatti led kigyullad, ami a bekapcsolt állapotot jelzi.

[PAUSE]: A számítógép futását függeszti fel, amíg le nem ütünk egy másik billentyűt. Olyankor hasznos, amikor gyorsan sok adat érkezik a képernyőre, és mielőtt elolvasnánk le is fut onnan.

[CTRL] + [BREAK]: A "break" felirat a **[PAUSE]** billentyűn olvasható. Szándékosan nem ctrl+pause -t írtam, mert a régebbi számítógépeken is a **[CTRL] + [BREAK]** volt az a billentyű-kombináció, amelyik megszakította program futását. Ezt teszi még a **[CTRL] + [C]** is.

[PAGE DOWN]: A DOS-ban nincs jelentése, de a szövegszerkesztőkben egyet lapoz lefelé, a szöveg vége felé. Funkciója megegyezik a számbillentyűzeten lévő **[3/PGDN]** billentyű jelentésével, ha a **[NUM LOCK]** nincs benyomva.

7. Számbillentyűzet



I./ Számszár bekapcsolva (NUM LOCK ON):

A 8-es területen a "Num Lock" felirat alatt lévő led világít. A fehér billentyűk leütésekor számok jelennek meg, illetve a tizedespont.

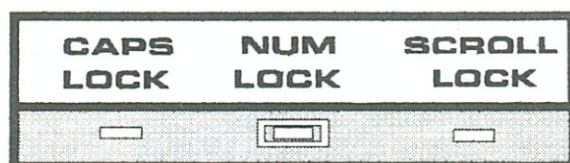
II./ Számszár kikapcsolva (NUM LOCK OFF):

A 8-es területen a "Num Lock" felirat alatt lévő led nem világít. A fehér billentyűk használata az 5-ös területen lévő kurzor vezérlő billentyűkkel egyezik meg.

III./ A számszár állapotától függetlenül:

A sötét [/] [*] [+] [-] [ENTER] billentyűk jelentése nem változik. Az [ENTER] jelentése azonos az 1-es területen lévőével. A [/] [*] [+] [-] billentyűknek a DOS-ban nincs speciális jelentése, a feltüntetett karakter jelenik meg a leütésére. Azért vannak mégis külön megismételve ezek a billentyűk, mert ha nagy tömegben kell számokat bevinni, adatrögzítő program futásakor, akkor sokkal könnyebb ezt használni, mint a felső sorban lévő számokat. A kurzor vezérlő billentyűfeliratok pedig azért vannak rajta, mert régebben a 86 gombos billentyűzeteken is így volt elrendezve. Aki viszont ehhez szokott hozzá, annak se kelljen szokásait feladni.

8. Led kijelzők



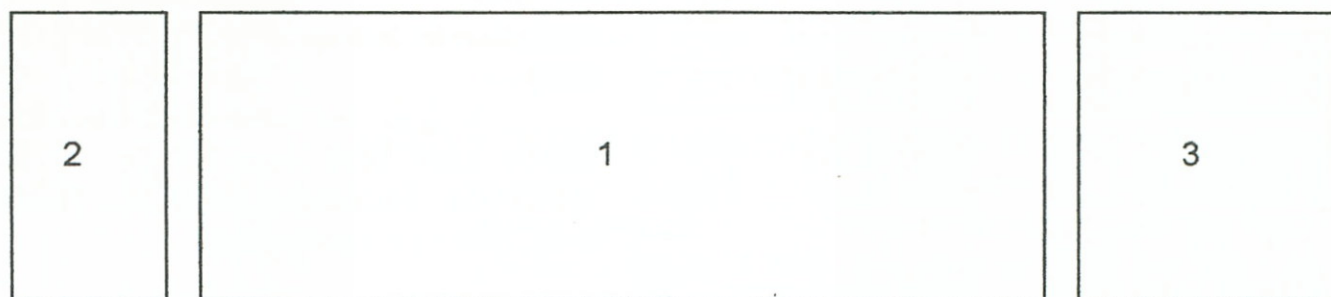
CAPS LOCK: Nagybetűzár, csak a 26 db angol ábécé betűire van hatással.

NUM LOCK: Számbillentyűzetzár, csak a fehér számokra és tizedes pontra (.) van hatással.

SCROLL LOCK: A DOS-ban nincs szerepe, nincs hatása semmire sem.

Mindegyik billentyű újbóli megnyomásra átvált a másik állapotába, és a ledet is váltja. A rendszer indulásakor a billentyűzet tesztje során felvillannak a ledék egyszerre, ha nincs semmi baj. Ha a kulcsos zár be van zárva, akkor a billentyűzetről semmit nem lehet bevinni a számítógépbe.

86 GOMBOS BILLENTYŰZET



Mivel már egyre kevésbé előforduló billentyűzetről van szó, csak a különbségeket ismertetem.

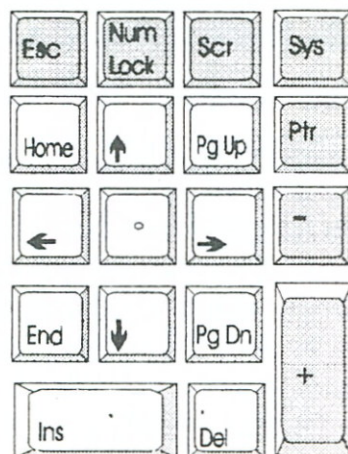
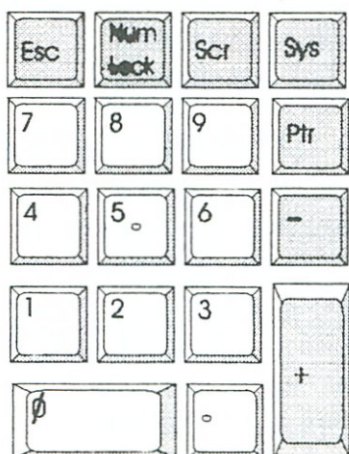
1.: Ez a része a billentyűzetnek hasonlít a 101 -eséhez, azzal a kis eltéréssel, hogy néhány billentyű máshová került. A **[CTRL]**, **[ALT]** segédbillentyűkből csak egy-egy darab van a bal oldalon, míg a 101 gomboson mindkét oldalon van, igaz erre csak kényelmi szempontból van szükség. A jobb oldali **[SHIFT]** billentyű alatt van a **[CAPS LOCK]** (nagybetűzár).

2.: A **funkcióbillentyűk [F1] - [F10]** -ig tartanak csak, és két oszlopban helyezkednek el. Ez az oka annak, hogy sok programban csak 10 funkcióbillentyűhöz rendelnek feladatokat, és 11. és 12. -hez nem.

3.: Ezen a területen vannak összezsúfolva azok a billentyűk, amelyek a 101 -esen a 4. 5. 6. 7. területen vannak elosztva. A terület használatát alapvetően a **[NUM LOCK]** billentyű állapota határozza meg.

Benyomott állapotban (a led világít): a **[0] - [9]** -ig billentyűk és a **[.]** számokat jelenítenek meg. Előre haladni (jobbra) a karakterek beírásával lehet, balra pedig csak törölni lehet a **[BACKSPACE]** (**[←]**) billentyűvel.

Ha a **[NUM LOCK]** -ot megnyomjuk még egyszer, akkor a led nem fog világítani. Ekkor a számokat csak az 1-es területen lévő billentyűkkel lehet megjeleníteni. A kurzort pedig a nyílbillentyűkkel lehet mozgatni, és az **[INS]** és **[DEL]** billentyűk is használhatóak. A többi (szürke) billentyűnek a funkciója független a **[NUM LOCK]** állapotától.



[ESC]: Funkciója megegyezik a 101-esen lévővel.

[NUM]: NUM LOCK - Számbillentyűzár.

[SCR]: SCROLL LOCK nincs használva.

[SYS]: SYS REQ nincs használva.

[PRT]: PRINT SCREEN Képernyő nyomtatását végzi.

A kurzorvezérlő-billentyűk és a számbillentyűk ugyanazt a funkciót töltik be. A műveleti jelek közül csak a **[+]** és a **[-]** billentyűnek jutott hely, a **[*]** **[/]** és **[ENTER]** billentyűk csak az 1 -es területen találhatók meg.

Lemezegység (FLOPPY DRIVE), hajlékony lemez (FLOPPY DISK)

A **lemezegység** az a része a számítógépnek, ami az adatokat felírja és leolvassa a **hajlékony lemezről**. A lemezt a meghajtó egységbe helyezve (5¼" -esnél a kallantyú elfordítása után), a lemezt szorosan megfogja. Ha adatot olvas a lemezről, akkor megpörgeti a lemezt, és a meghajtón lévő led kijelző világít. A meghajtóegységben két író-olvasó fej található, ami a lemez két oldalához simul. Ahhoz, hogy a lemez mindkét oldalára írjunk adatokat, nem kell kivenni és megfordítani a lemezt, hanem azzal a fejjel ír a lemezre az egység, amelyik oldalára dolgozik a lemeznek. Különösen vigyázni kell a lemezeink tisztaságára, mert a lemezről a fejre átkerülő szennyeződés, más lemezeinket is tönkretelheti.

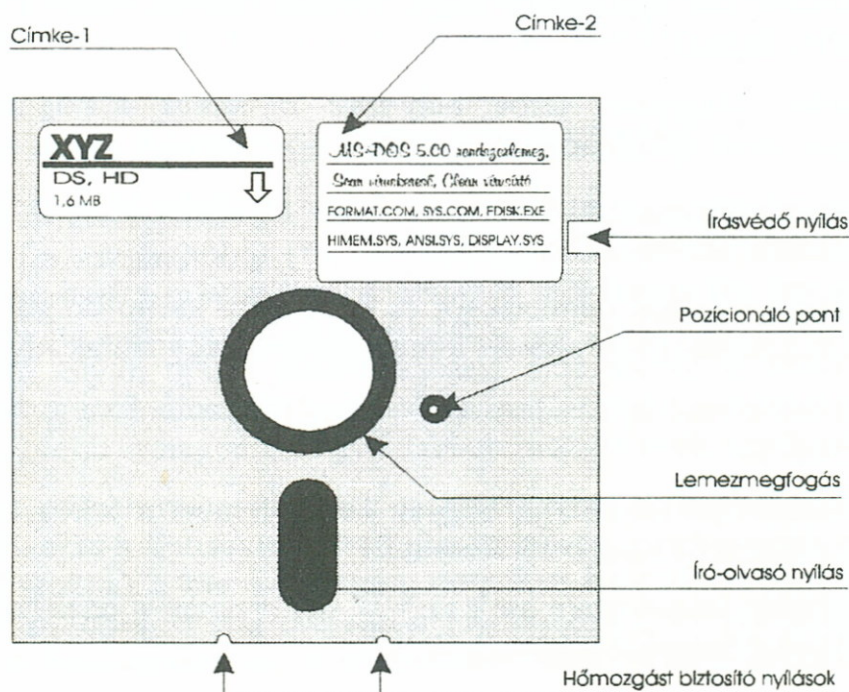
A lemezegységet külön **vezérlőkártya** (CONTROLLER) vezérli. Ennek több típusa létezik, attól függően, hogy még milyen feladatot lát el. Lehet rajta még **ki- és bemeneti csatlakozó hely** (COM, LPT vagy GAME port), akkor "MULTI I/O" kártyának nevezzük. Vannak olyan vezérlők is -ez a gyakoribb-, amelyeken két hajlékony- és két merevlemez vezérlő van. Újabban már csak olyan típusok vannak forgalomban, amelyek az előbb említett összes funkciót ellátják.

Lemezegység is több típus létezik, attól függően, hogy milyen adatsűrűséggel írhatunk a lemezre, illetve a hozzá használható lemez méretétől függően. A használható lemezek típusai:

5¼": 5,25 col (inch vagy hüvelyk) Kapacitása: 360 kilobájt DS DD -s típusú lemezre, 1.2 megabájt DS HD -s típusú lemezre. (Részletesebben lásd a **FORMAT** parancsnál.)

3½": 3,50 col Kapacitása: Általában 720 kilobájt vagy 1.44 megabájt, de már létezik 2.88 megabájtos és 20 megabájtos (FLOPTICAL) adatsűrűségű lemez is.

Általános szabály az, hogy a nagyobb kapacitású lemezegység olvassa az azonos méretű, de kisebb lemez tartalmát, de fordítva nem igaz. Sajnos ez a kompatibilitás nem mindig áll fenn a "Made in Taiwan" felíratú meghajtóknál, de sok esetben még az azonos típusú lemezeket sem olvassa, ha az másik meghajtón lett felírva.



A lemez részei:

CÍMKE_1: A lemez típuscímkéje, a gyártó cég ragasztja fel. Van rajta néhány információ a lemez típusával kapcsolatosan.

CÍMKE_2: Mi ragaszthatunk fel öntapadós papírcímkét a lemez borítójára, amire felírhatunk, a lemezen tárolt programokra vonatkozó információkat.

ÍRÁSVÉDŐ NYÍLÁS: Kis ragasztószalaggal le lehet ragasztani, így nem lehet a lemezre írni. Olyankor használjuk, ha fontos programokat tárolunk a lemezen. Leragasztásával meg tudjuk védeni a programjainkat a véletlen törléstől és a vírusoktól.

LEMEZ MEGFOGÁS: A lemez közepén van egy kör alakú nyílás, amibe egy kúpos elem illeszkedik a lemezegységben. Ennek segítségével fogja meg a lemezt a meghajtóegység, így tudja megpörgetni.

POZÍCIONÁLÓ PONT: Mivel a lemez kör alakú, így nincs eleje és vége. Ezért jelöltek ki a lemezen egy pontot (lyukat), ami a lemezen való pozícionálást segíti elő.

ÍRÓ-OLVASÓ NYÍLÁS: A lemeznek azon része, amit nem fed a borító, így itt lehet a lemezre írni, illetve onnan olvasni. Éppen ezért ez a lemez legsérülékenyebb része. Ha a lemez itt bepiszkolódik, megsérül, akkor az adatainkat már nagy valószínűséggel nem tudjuk soha visszaolvasni.

HÓMOZGÁST BIZTOSÍTÓ NYÍLÁSOK: A lemez borítójának a hő hatására bekövetkező természetes mozgásait biztosítja ez a két kis nyílás, így nem fog deformálódni a lemezünk hő hatására sem.

A lemez papírborító tasakjának a hátulján található néhány nemzetközi információs képjelet, ami a lemez használatával kapcsolatos:

LEMEZ, HŐMÉRŐ ÁBRA: A lemez csak 10-60°C hőmérsékletet tud károsodás nélkül elviselni.

LEMEZ, TOLL ÁBRA: A lemezre írni csak puha hegyű ironnal szabad, mert a borítón átütő tollhegy karcolhatja a lemezt.

LEMEZ, PATKÓMÁGNES ÁBRA: Mágneses térbe helyezve a lemez átmágneseződhet, és elveszíthetjük a rajta tárolt információkat.

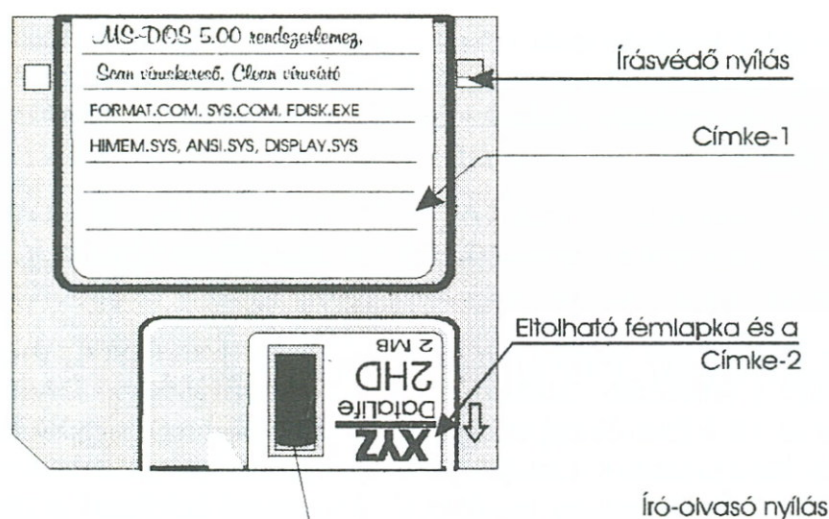
LEMEZ MEGHAJLÍTOTT ÁBRÁJA: A lemez azért hajlékony, hogy ne törjön el, és nem azért, hogy hajlítgassuk. Maradandó károsodást szenvedhet.

LEMEZ GÉMKAPOCCSAL ÁBRA: A lemezre gémkapcsot és bármilyen kiálló dolgot rátűzni tilos, mert akadályozhatja a lemez behelyezését a meghajtóba, vagy annak kivételét onnan.

LEMEZ, KÉZ ÁBRA: A lemez író-olvasó nyílásához nyúló kezlet ábrázoló ábra arra figyelmeztet, hogy oda nyúlni tilos.

LEMEZ TOKBA HELYEZÉSÉNEK ÁBRÁJA: Bemutatja hogyan kell a lemezt a tokba tenni, hogy az valóban megvédje a lemezt a kisebb sérülésektől.

A fent említett intelmek, javarészen csak az 5¼" -os lemezre illenek, mert a 3½" -os lemezen már ezeket a hibalehetőségeket kiküszöbölték.



Az 5 1/4" -os és a 3 1/2" -os lemezek összehasonlítása:

- A 3 1/2" -os lemez felületi mérete jóval kisebb, és kétszer olyan vastag, mint az 5 1/4" os lemez, miáltal az sokkal merevebb és nehezebben hajlítható meg.
- Az 5 1/4" -os lemeznek a meghajtóban történő megfogása egy kúpos elemre szorítás révén jön létre. Ez nem olyan stabil, mint a 3 1/2" -osnál, ahol a lemez közepén van egy kör alakú fémlemez, amiben egy kör, és egy téglalap alakú nyílás van. A megfogó szerkezet ezekbe a nyílásokba nyúlva biztosan meg tudja fogni a lemezt, és a pozícionálásra sincs gondja.
- A 3 1/2" -os lemeznek az író-olvasó nyílása egy eltolható fémlappal le van fedve, így az nem tud károsodni.
- Az 5 1/4" -os írásvédő nyílását le kell ragasztani, ha nem akarjuk az adatainkat letörölni, míg a 3 1/2" -os lemeznél egy kis lapkát kell áttolni egy másik helyzetbe. Lényegesen egyszerűbb az utóbbi és korszerűbb, lévén hogy a lemezre ragasztgatás nem egy tiszta munka.

Összegezve a leírtakat, annak ellenére, hogy az 5 1/4"-os lemez van jobban elterjedve, a 3 1/2"-os lemez sokkal jobb, tartósabb, és ami a legfontosabb védettebb, mint az 5 1/4" -os. A 3 1/2" -os kapacitása is valamivel nagyobb, annak ellenére, hogy mérete jóval kisebb.

Merevlemez háttértár (WINCHESTER)

Hasonló elven működik, mint a hajlékonylemez meghajtó, de itt a lemezeket nem lehet kivenni a meghajtóból (*), és nem egy lemez van benne, hanem több (típusonként különböző darabszám lehetséges). A lemezek megfogása fix, így sokkal gyorsabban fel lehet pörgetni a kívánt sebességre, és az adatokat így sokkal gyorsabban lehet beolvasni róla. (* Már kaphatók olyan típusú merevlemezek, amelyekből ki lehet venni a lemezkötegeket. Létezik már olyan hordozható merevlemez is, amelyik a nyomtatócsatlakozóra köthető, így az adatok nagyobb tömegben, és gyorsabban mozgathatók egyik számítógépből a másikba.)

Vannak olyan merevlemez is, amelyeknél az egész egység kiemelhető, és másra cserélhető. Ezeket a típusokat azért fejlesztették ki, mert vannak olyan programok, amelyek sok helyet foglalnak a merevlemezen, de ritkán használjuk őket. Ebben az esetben, amikor szükség van rá, csak ki kell cserélni a merevlemezt, és már használhatjuk is a programot. Nem foglalja feleslegesen azt a lemezünk, amelyiken a gyakran használt programjainkat tároljuk.

Itt viszont a merevlemez előnyei elvesznek, mert a cserélhető lemezes típusnál az adatok elérési ideje hosszabb lesz. A cserélhető merevlemez típusnál viszont a csatlakozási hely fog idő előtt kikopni, és érintkezési hibák miatt nem tudjuk kihasználni a merevlemezünknek a teljes élettartamát.

Az előbb említett okok miatt, csak a hagyományos merevlemezzel foglalkozunk. Ezek légmentesen le vannak zárva, és a lemezeket nem lehet kivenni belőle, így azok nem is sérülnek meg.

A merevlemez háttértár kapacitása sokkal nagyobb a floppy lemezénél, mert több lemez van egy egységben, és ezekre a lemezekre sokkal nagyobb adatsűrűséggel lehet írni. A legkisebb merevlemez kapacitása is 10 Mb volt régebben, ma már 120 megabájtól kisebbet nem is gazdaságos vásárolni, sőt hamarosan a kereskedelmi forgalomból is kikerülnek. A több lemezhez, értelemszerűen több író-olvasó fej is tartozik, lemezenként kettő, viszont az egyik legszélső lemeznél csak egy fej van konstruktív okokból. Tehát egy 5 fejes Winchester -ben 3 lemez van, aminek a kapacitása a típusától függ.

Adatok tárolása lemezeken

Ahhoz, hogy a lemezen adatokat tárolhassunk, fel kell azt osztani egységes méretű területekre. Ezek közül ki kell jelölnünk megkülönböztetett területeket, ahol fontos adatokat fogunk eltárolni. Ahhoz, hogy a lemezen tárolt adatokat megtalálhassuk, fel kell jegyeznünk az adatait a lemez egy területére. A rendszer betöltésekor is szükség van a lemezen egy olyan területre, ahol mindig, minden körülmények között megtalálja a számítógép a betöltendő rendszerfájlokat. Szükséges még egy olyan terület is, ahol a lemezen lévő területek foglaltságát tartjuk nyilván, nehogy olyan területre írjunk, ahol már van egy program. A lemezen lévő területek írását, olvasását a rendszer működését biztosító fájlok végzik. A kijelölt területeknek tehát olyan helyen kell lenniük, amelyet bármilyen kapacitású lemezen el tudunk érni.

A lemezen lévő adatokat - bájtokat - *szektorok*ban (SECTOR) tároljuk. Egy szektorba általában 512 bájt kerül. A következő területi egység a *klaszter* (CLUSTER), ami az egymás feletti szektorok összességét jelenti. Ez egy hajlékony lemez esetében kettő szektort jelent, viszont egy nagyobb merevlemez esetén 8 vagy 16 szektor is lehet. Így egy klaszter mérete hajlékony lemezen 1024 bájt, míg merevlemezen 4096 bájt, vagy annak többszöröse. (Minél nagyobb kapacitású a merevlemez, annál nagyobb a klaszter mérete. Novell hálózati merevlemeznel, az installáláskor mi választhatjuk ki a klaszter méretét a lehetséges értékek közül. Lásd "*Novell hálózati alapismeretek II.*" című könyvemben.)

Azon a területen, ahol a területek foglaltságát tartjuk nyilván, csak a klaszter foglaltságát jegyezzük fel. (Olvasd el a **CHKDSK** parancsnál található kis eszme-futtatást is!) Ha minden szektor foglaltságát külön tárolnánk a lemezen, akkor gazdaságtalanul nagy területet foglalna el ez az adminisztrációs terület. Ezt a területet nevezzük *fájl foglaltsági táblázat*nak (FAT - FILE ALLOCATION TABLE).

A szektorok, a lemezen *sávok*ban (TRACK) helyezkednek el. Ezek a sávok körvonalak mentén helyezkednek, a számuk pedig a lemez fizikai tárolóképességétől függ. Az egymás felett elhelyezkedő sávok összefoglaló neve a *cilinder*.

A számítógép bekapcsolása után a **BIOS** nevű ROM -ban (READ ONLY MEMORY - csak olvasható memória) lévő program fut le, majd betölti a kitüntetett lemezegység első

szektorában található adatok alapján a DOS rendszer állományait, és átadja neki a vezérlést. Az első szektort a lemezen, ahonnan a rendszert betölti a számítógép, **betöltő szektornak** (BOOT SECTOR) nevezzük. Van még egy könyvtár is, ahová bejegyzésre kerülnek a kimásolt fájlok, és a további alkönyvtárak nevei. Ezt a könyvtárat pedig **gyökérvönyvtárnak** (ROOT DIRECTORY, jele: '\ ') nevezzük.

A gyökérvönyvtárban további könyvtárakat hozhatunk létre. Erre azért van szükség, hogy a különböző jellegű programjainkat, külön csoportokban (könyvtárakban) tarthassuk. Ezekbe a könyvtárakba is további alkönyvtárakat nyithatunk, mire egy **könyvtárfa** alakul ki. Egyszerre csak egy könyvtárban dolgozunk, ez az **aktuális könyvtár**.

Mindig az az **aktuális** (meghajtó, könyvtár, fájlnev), amelyikről szó van, amelyik környezetben dolgozunk. Az 'aktuális' szó sok helyen szerepel, ezért érdemes memorizálni a jelentését, a későbbi keveredés elkerülése végett. A fájlokat alapértelmezésben az **aktuális meghajtó aktuális könyvtárában** keresi a számítógép. Azt pedig, hogy melyik meghajtón melyik könyvtárban állunk, azt a **készenléti jel** (PROMPT) mutatja meg általában (Ha nem, akkor írjuk be: **CD [ENTER]**). Ha a készenléti jelet még nem állítottuk be a rendszer indulása óta, akkor az csak az aktuális meghajtó betűjelét tartalmazza, és egy '>' nagyobb (GREATER THAN) jelet, aminek csak elhatároló szerepe van. Elválasztja a készenléti jelet a parancssortól. Ha a készenléti jelet beállítjuk a következő paranccsal, akkor az mutatni fogja mindig az aktuális könyvtárat is:

```
C>prompt $p$g
C:\>_
```

A **meghajtók betűjelei** általában kötöttek, egy számítógép esetén állandónak tekinthetjük. Az 'A:' és a 'B:' meghajtók mindig a hajlékony lemezes egységre vonatkoznak. Ha csak egy floppy meghajtónk van, akkor a 'B:' lemezegységre vonatkozó parancsokat is ugyanazon a lemezegységen hajtja végre, csak előtte küld egy üzenetet. Az üzenet arra vonatkozik, hogy tegyünk a 'B:' meghajtóba egy lemezt, és nyomjunk meg egy billentyűt. Ekkor abba az egy meghajtóba kell a megfelelő lemezt betenni, amink van. A számítógép csak szimulálja a 'B:' meghajtót.

Ha van egy vagy két merevlemezünk is, akkor azok a 'C:' illetve a 'D:' meghajtó betűjelet kapják. Ha nincs, akkor az első szabad meghajtó betűjel a 'C:' lesz. Ebben az esetben ha installálunk egy memórialemezt (RAM DRIVE, lásd "*DOS alapismeretek II.*" című könyvemben), akkor annak a jele az első szabad meghajtó betűjel lesz.

Billentyűzetkezelés a DOS-ban

A DOS készenléti jel mellett a használható billentyűk száma igen korlátozott. Nincs minden lehetséges billentyű-kombinációhoz valamilyen feladat rendelve, és nekünk sincs lehetőségünk ezen változtatni egyszerű módon. Természetesen vannak módszerek, amelyeknek a segítségével átdefiniálhatjuk a billentyűzetet, megváltoztathatjuk a képernyőn megjelenő karakterek színeit ...*stb*

Itt van például az **[F11] - [F12]** funkcióbillentyűsor. Aki már használt számítógépet az tudja, hogy ezekhez a billentyűkhöz hozzá lehet rendelni más számítógépeken

(COMMODORE...) olyan parancsokat, kiírásokat, amelyeket gyakran használunk. Ekkor csak az adott funkcióbillentyűt kell megnyomni, és a parancsot a számítógép kiírja helyettünk, és esetleg végre is hajtja. A DOS-ban is van erre lehetőség, de csak segédprogramok használatával, és elég nehézkes módon.

Aki esetleg megszokta egy más típusú számítógépen a teljes képernyős szerkesztést (FULL SCREEN EDITOR), vagyis hogy tetszőlegesen lehet a képernyőn mozogni a kurzorral, annak kicsit szokatlan lesz a DOS-ban az, hogy itt a készenléti jel mellett kell mindent beírni, és nem vihetjük a kurzort néhány sorral feljebb, ahol esetleg egy előzőleg beírt parancsot átszerkesztve, azt újra kiadhatnánk. Sőt, a DOS -ban alaphelyzetben - segédsoftverek nélkül - csak az előző parancs szerkeszthető, az is csak olyan "balkáni" módon.

Akkor most lássuk mit is használhatunk a DOS-ban, segédprogramok nélkül:

A billentyűzet középső részén lévő fehér billentyűk használatakor (betűk, számok, írásjelek), annak képe jelenik meg a monitoron is. A számok és írásjelek esetén az alulra írt képe jelenik meg.

Ha a **[SHIFT]** billentyűvel együtt nyomjuk le valamelyiket, akkor a betűknél a nagybetű formája jelenik meg, az írásjeleknél, számoknál, pedig a két jel közül a felső.

Ha az **[ALT]** billentyűvel együtt nyomjuk a fehér billentyűket, nem történik semmi sem.

Ha a **[CTRL]** billentyűvel nyomjuk együtt a betűket, akkor annak a nagybetűs képe jelenik meg egy hatványjellel bevezetve (például: ^A). Ennek viszont van jelentése, de ezzel jobb ha nem foglalkozunk még egyelőre. Ezek vezérlőkódokat jelenítenek meg, funkciójuk azonos néhány szürke billentyűvel. (például: ^M ≡ **[ENTER]** azaz **[ENTER]** helyett **[CTRL] + [M]** -et is nyomhatunk.

Speciális feladatot ellátó billentyűk:

[ALT] + [CTRL] + [DEL] :Tetszőlegesen válogathatunk az **[ALT]**-ok **[CTRL]**-ok és **[DEL]**-ek között, ha ezt a három billentyűt egyszerre lenyomjuk, akkor a számítógépünk újra tölti a rendszert, és mindent "elfelejt" amit idáig a memóriájában tárolt. Akkor használjuk, ha a számítógépünk "lefagyott", azaz a rajta futó program rossz működés következtében nem reagál már semmilyen billentyűre sem. Természetesen a lemezre kiírt adataink nem vesznek el, ha az állományt lezárta a programunk.

[CTRL] + [C] : A számítógépen futó programot szakítja meg, hatása elvileg azonos a **[CTRL] + [BREAK]** billentyűével, de az utóbbinak sokszor nagyobb a hatásköre.

[CTRL] + [M] : Hatása megegyezik az **[ENTER]** billentyű hatásával.

[CTRL] + [P] : A képernyőn megjelenő minden karakter a nyomtatón is meg fog jelenni. Bővebben az "Állományok átirányítása" című fejezetben olvashattok róla, addig viszont annyit, hogy ne használj ezt a billentyű-kombinációt, ha nincs nyomtató a számítógéphez csatlakoztatva. A billentyűk újbóli megnyomására ez a kettős kiírás megszűnik.

[CTRL] + [S] : A számítógépen futó program futását függeszthetjük fel addig, amíg egy tetszőleges billentyűt le nem nyomunk. Hatása megegyezik a **[PAUSE]** billentyűével.

A szóközbillentyű mindkét oldalán található egy-egy **[ALT]**, **[CTRL]**, és **[SHIFT]** billentyű. Ezek az úgynevezett *segédbillentyűk*. Abban térnek el a többi billentyűtől, hogy ezek önmagukban lenyomva nem jelenítenek meg semmit a monitoron. Akár hosszú ideig is lenyomva lehet tartani ezeket anélkül, hogy valami hatása lenne. Tehát például a **[CTRL] + [S]** billentyűket úgy kell használni, hogy lenyomjuk a **[CTRL]**-t az egyik kezünkkel, és lenyomva tartjuk, majd az **[S]**-et nyomjuk le, határozottan és rövid ideig, majd a **[CTRL]** billentyűt felengedjük.

Ha egy tetszőleges karaktert ábrázoló billentyűt lenyomva tartunk sokáig (néhány másodpercig), akkor megismétli a billentyű képét a monitoron, addig amíg a billentyűt lenyomva tartjuk.

Alapértelmezésben a DOS -ban a kisbetűket jeleníti meg egy betűbillentyű leütésére, de ha több nagybetűt akarunk leírni, akkor bizony kényelmetlen eljárás, ha az egyik kezünkkel folyamatosan lenyomva tartjuk a **[SHIFT]**-et. Erre szolgál a **[CAPS LOCK]**. Magyarul *nagybetűzárnak* lehet nevezni, mivel ennek lenyomása után minden betű lenyomásakor, annak a nagybetűs képe jelenik meg. Csak a betűkre vonatkozik, ellentétben az írógépekkel, ahol minden karakternek a felső képe jelenik meg. Ha például a számsor feletti karaktereket akarjuk megjeleníteni, akkor továbbra is használnunk kell a **[SHIFT]**-et.

A DOS -ban nem sok szerepe van a **[TAB]** (TABULATOR) billentyűnek, aminek hatására alapértelmezésben nyolc hellyel lép odébb a kurzor. Visszalépni a **[SHIFT] + [TAB]**-bal nem lehet a rajta lévő feliratozással ellentétben, de szövegszerkesztő programok már így használják.

A DOS -ban visszalépni és törölni az előzőleg beírt karaktert a **[BACK SPACE] [←]** billentyűvel lehet. Ha a **[TAB]**-ot nyomtuk meg előtte, akkor azt törli, azaz nyolc helyet ugrik vissza a kurzor a sorban.

A legfontosabb billentyű mindegyik közül az **[ENTER]**. Ha beírunk egy sort, mindaddig amíg az **[ENTER]**-t meg nem nyomjuk, az csak a képernyőn van jelen. Viszont ha lenyomtuk az **[ENTER]**-t, akkor fogja a számítógép értelmezni a beírt karaktereket, és ha végrehajtható, akkor végrehajtja, ha nem akkor hibaüzenetet ad.

A legfelső sorban található az **[ESC]** (ESCAPE - menekülés). A DOS -ban nincs sok szerepe, egy visszaper (\ - BACKSLASH) jelet ír ki, és a kurzort egy sorral lejjebb viszi. A beírtakat figyelmen kívül hagyja, és egy **[ENTER]**-rel léphetünk új sorba. A szövegszerkesztőkben és más programokban kilép az aktuális funkcióból, vagy a programból.

Mellette találhatóak az **[F1] - [F12]**-ig *funkcióbillentyűk*. Általában valamilyen feladatot (funkciót) rendelünk hozzá, és megnyomása esetén nem egy karakter jelenik meg, hanem egy feladatsor hajtódik végre. Az **[F1]** billentyű megnyomására a programokban segítséget (HELP) kapunk a program működéséről. Általában csak az első tíz billentyűhöz rendelnek feladatokat, mert a 86 gombos billentyűzeteken csak 10 funkcióbillentyű van.

A DOS -ban csak az **[F1] - [F5]** billentyűknek van szerepe a *parancssor szerkesztésében*. Az **[F1]** billentyű megnyomására az előzőleg kiadott parancs karaktereit másolja ki a parancssorba. Ha az előző parancsnak csak egy részét akarjuk kiadni, akkor csak addig nyomkodjuk az **[F1]**-et, ameddig kiírja a szükséges részét a parancsnak, és **[ENTER]**-t nyomunk.

Az **[F3]**-as billentyű megnyomására az előző parancsot teljes egészében kiírja a parancssorba. Természetesen ez tovább szerkeszthető. Ha változtatás nélkül adjuk ki a parancsot, akkor az **[F3]** és **[ENTER]** billentyűket kell egymás után megnyomnunk.

Ha látjuk a szerkesztendő sort is a képernyőn, akkor lehet a segítségünkre az **[F2]**-es billentyű, ami adott karakterig másol. Megnyomjuk az **[F2]**-t, elengedjük, majd megnyomjuk azt a billentyűt ameddig másolja ki az előtte álló összes karaktert.

Hasonlóan működik az **[F4]**-es billentyű is, csak ez adott karakterig kihagyja a kimásolandó karaktereket. Megnyomjuk az **[F4]**-et, majd utána azt a karaktert, ameddig törölje a kiírandó karaktereket. Ha most a jobbra nyilat megnyomjuk, akkor az a karakter jelenik meg, amelyikig az előbb a törlést végeztük.

Az **[F5]**-ös billentyű hatására új sorba kerül a kurzor, és az előző sorban megváltoztatott szöveget tekinti kiindulási sornak. Ha itt **[ENTER]**-rel kilépünk, akkor az **[F3]**-ra az átszerkesztett sort fogjuk megkapni, és nem az előzőleg kiadott parancsot.

A 101 gombos billentyűzeteken elkülönítve található a négy kurzormozgató-billentyű. A szövegszerkesztőkben a képernyőn, a nyíllal jelzett irányban lehet mozogni, vagy mozgatni a képernyőt (SCROLL - görgetés, mozgatás).

A DOS -ban csak a **[→]** és **[←]** billentyűknek van szerepe. A jobbra nyíllal lehet - az **[F1]**-es billentyűhöz hasonlóan - a parancssorba másolni az előző parancs karaktereit egyesével. A balra nyíl pedig visszafelé lépked a sorban, de nem törli ki a karaktert a memóriájából, mert ha a jobbra nyíl billentyűt nyomjuk meg, akkor újra előjön a következő kiírandó karakter. Hasonló a **[BACK SPACE]** (vagy máshol az **[ENTER]** feletti **[←]**) billentyű hatása is, a kurzort balra mozgatja, de csak a képernyőről tűnik el a kiírt karakter.

A parancssorban lévő karakterekből törölni is lehet, és bele is lehet írni újabb karaktereket. Ettől válik szerkeszthetővé a parancssor.

Kicsit kezdetleges módszerekkel, de van lehetőség az előzőleg beírtak átformálására, csak fejben kell tartanunk a beírt parancs karaktereinek sorrendjét. Hogy miért azt azonnal ismertetem:

Az előző parancsot az **[F3]** billentyűvel másoljuk ki a parancssorba, és lépünk vissza néhányat. Itt most ha megnyomjuk az **[INS]** billentyűt, akkor beírhatunk új karaktereket (például még egy paramétert). Ha ezután a jobbra nyilat nyomjuk meg többször, akkor a még ki nem írt karakterek bújnak elő. Ha nem nyomnánk meg beszúrás előtt az **[INS]** billentyűt, akkor a beírt karaktereinkkel a memóriában lévő még ki nem írt karaktereket íránk felül. Ha viszont valahol megnyomjuk a **[DEL]** billentyűt, akkor azt a karaktert fogja letörölni ("lenyelni"), amelyik a jobbra nyíl hatására előjött volna.

Ezért írtam, hogy fejben kell tartani a bent lévő karaktereket, mert amit a **[DEL]**-lel letörlünk, azt nem látjuk. Félig-meddig vakon kell szerkeszteni a sort. Ezt csak kellő gyakorlat megszerzése után lehet eredményesen használni. Anélkül még úgy is gyorsabban érünk célba, ha újra beírjuk az egész sort.

Viszont addigra megismerkedhetünk a **DOSKEY** adta lehetőségekkel, ami korszerű parancssorszerkesztő rezidens program. Ha nem a DOS -ban dolgozunk a gépidő jelentős részében, akkor a **DOSKEY** feleslegesen foglalná a drága memóriánkat. (Bár ez utóbbi kijelentés a **QEMM.EXE** memórimenedzser segédprogram használata esetén, a 6.00 -ás DOS -ban pedig a **MEMMAKER** parancs használatakor nem állja meg a helyét. Viszont az

NC (Norton Commander) használata esetén szintén felesleges, sőt a billentyűzetkezelésük is különbözik.)

Viszont ekkor is kell valamennyit a DOS -t is használnunk, és akkor hasznos lehet, ha nem kell minden parancsot újra meg újra beírunk, hanem az előző átszerkesztéséből gyorsan kiadhatjuk a kívánt parancsot. Akkor előnyös a használata, ha valaki nem ismeri a billentyűzetet eléggé - keresi még a karaktereket -, és ekkor az előzőleg beírtakat előhívhatja néhány szerkesztőbillentyű használatával.

Akkor is hasznos lehet, ha hosszú (sok paramétert tartalmazó) parancsot adunk ki, de véletlenül elgépeztük, és ki kell javítanunk. Egy-két elírt karakter miatt ha újra az egészet be kellene írunk, az sokkal több időnkbe telne, mint előhívni és kijavítani.

A vakon szerkesztést úgy lehet például kijátszani, hogy az **[F3]** billentyűvel kimásoljuk az előzőleg beírt és szerkeszteni kívánt sort, majd az **[F5]**-ös billentyűvel új sorba lépünk. Ekkor látjuk pontosan és közel a szerkesztendő sorhoz, hogy mit kell szerkeszteni.

Most tekintsük át konkrétan a parancssor szerkesztését a DOS -ban néhány példán keresztül:

<pre>C:\>fomat a: /v:cimke /8 Bad command or file name C:\>format a: /v:cimke /8 : C:\>format a: /s /8 : : : C:\> C:\>_</pre>	<p>A parancs helyesen: FORMAT</p> <p>[F1] [F1] [INS] [r] [F3] [ENTER]</p> <p>[F1] -et 11-szer, [s] [DEL] -t ötször és [F1] [F1] [ENTER]. A "v"-t felülírtuk az "s"-sel, a ":cimke" -t pedig letöröltük.</p> <p>Ha csak [ENTER] -t nyomunk, akkor az a parancs, így a következő sorban nincs mit szerkeszteni.</p>
---	---

Az első esetben használhattuk volna a következő billentyűsört is:

[F2] [m] [INS] [r] [F3] [ENTER]

Itt nem olyan jelentős az **[F2]** használata, mint a következő példában, ahol sokszor kellene az **[F1]** -et megnyomni:

[F2] [v] [F4] [/] [F3]

Az **[F4]** -es billentyű használatát ez a példa mutatta be a legjobban. A fenti példában ajánlott 19 darab billentyű leütése helyett a sor kijavítását öt billentyű leütéssel megoldottuk.

A következőkben ismertetésre kerülő billentyűknek a DOS -ban nincs jelentősége, így azokat csak röviden tárgyaljuk, feladatuk pedig a billentyűzet felépítésénél ismertetve lett.

A kurzormozgató-billentyűk felett található az előbb ismertetett **[INS]** és **[DEL]** mellett a **[HOME]**, **[END]**, **[PAGE UP]** és **[PAGE DOWN]**. Ezeknek a billentyűknek csak a szövegszerkesztőkben és egyéb programokban van szerepe, a DOS -ban semmi. Ezek felett a

felső sorban van a **[PRINT SCREEN]** billentyű, aminek a megnyomása esetén két dolog történhet:

- Normális esetben kinyomtatja az aktuális képernyő tartalmát.
- Ha viszont nincs nyomtató a számítógépünkhöz csatlakoztatva, akkor ad egy sípoló hangot, jelezve, hogy nincs hová nyomtatni.

A **[PAUSE]** egy másik billentyű lenyomásáig felfüggeszti a gépünk működését. Hosszú listák kiírása esetén használjuk, hogy a közbenső részeket is el tudjuk olvasni. Ennél a módszernél van jobb megoldás is (lásd **MORE** parancs).

A **[SCROLL LOCK]** nevével ellentétben nem zárja le a képernyő görgetését. Nem használjuk semmire sem.

A **[NUM LOCK]** állapotától függően, a számbillentyűzeten vagy a számok jelennek meg és a műveleti jelek, vagy a kurzormozgató funkciók élnek. Az **[ENTER]** a számbillentyűzétár állapotától függetlenül megegyezik a másik **[ENTER]** billentyű hatásával.

A számítástechnikában használatos jelölések

Egy külső parancs általános definíciója

UNDELETE	Törölt állományok visszahozatala
-----------------	----------------------------------

UNDELETE [[*meghajtó:*] [*elérési_út*] *fájlnev*] [/LIST | /ALL] [/DOS | /DT]

Használata:

D: \>	C: \DOS\	UNDELETE	A: \PASCAL*.PAS	/DOS
A	B	C	D	E

A: Készenléti jel (PROMPT)

B: Külső parancs helye

C: Parancs neve

D: Paraméterek

E: Kapcsolók

- A/ A **készenléti jelet**, a PROMPT belső paranccsal lehet állítani, mutatja azt a helyet, ahová a parancsot beírjuk.
- B/ Az **elérési út** megadására a külső parancsoknál lehet szükség, annak a helyét mutatja, ha nem az aktuális könyvtárban van, és nincs megadva a helye a **PATH**-ban (lásd ott!).
- C/ A **parancs** vagy a **program neve**, és a kiterjesztése COM vagy EXE, és betöltés után átveszi a vezérlést.
- D/ **Paraméter**(ek), amely(ek)re a parancs vonatkozik.
- E/ **Kapcsolók**, amelyek a parancs végrehajtását befolyásolják, területét behatárolják, vagy komfortosabbá teszik.

Nézzük meg részletesebben a parancs definícióját:

[[*meghajtó*] [*elérési_út*] *fájlnev*]

Általában ami a '[' szögletes zárójelek között van, azt nem kell megadni. Ha több szögletes zárójel is egymásba van ágyazva, akkor értelemszerűen kell használni, mint a matematikában. Ebben a példában egyik paramétert sem kötelező megadni. Ha lebontjuk a külső zárójeleket, akkor marad az, hogy:

[*meghajtó*] [*elérési_út*] *fájlnev*

A *fájlnev* külön is megadható, az előtte állókat akkor sem kötelező megadni.

A *meghajtó*: és az *elérési_út* egyenlő rangú, vagyis megválasztható, hogy melyiket adjuk meg (Természetesen az adott helyzet dönti el melyiket adjuk meg).

Megadhatjuk úgy is, hogy:

meghajtó: *fájlnev* vagy *elérési_út* \ *fájlnev*

A meg nem adott paraméternél mindig az *aktuális* értendő. Vagyis ha az aktuális meghajtó az A: akkor a példánk úgy módosulna, hogy:

A:\>c:\dos\undelete \PASCAL*.PAS /DOS

Ha pedig az aktuális könyvtár az A: meghajtón a PASCAL, akkor a D: meghajtóról úgy adjuk ki a parancsot, hogy:

```
{D:\>cd a:
A:\PASCAL }
```

D:\>c:\dos\undelete A:*.PAS /DOS

Természetesen mindkét eset, a részletes példában bemutatott megadással egyenértékű.

[/LIST | /ALL]

Ezeket szintén el lehet hagyni. Általában az ilyen esetekben a választható kapcsolók közül az egyik az alapértelmezésbeli, és azért nem kötelező megadni, mert akkor azt használja. A két kapcsoló között látható jel a 'vagy' jelképezi, hogy a két lehetőség közül választhatunk, ha meg akarjuk adni, de egyszerre mindkét kapcsoló nem adható meg. A definícióban szerepel még a [/DOS/DT], amit szintén hasonlóan használhatunk.

A parancsot, a paramétereket és a kapcsolókat szóközzel választjuk el egymástól. Ebből következik, hogy egyik sem tartalmazhat szóköz karaktert, mert onnan már új paraméternek venné a számítógép.

A parancsok és a paraméterek kis- és nagybetűkkel egyaránt megadhatók, mert egyfajta konvertálja a számítógép a betűket, és úgy próbálja meg értelmezni.

Mielőtt rátérnénk a DOS parancsok tárgyalására, nézzük meg a könyvben alkalmazott jelölések jelentéseit:

A parancs nagy betűvel: **DATE**

A paraméter, amit nem kötelező megadni, szögletes zárójelben, és döntött betűkkel:

[**dátum**]

A paraméter amit kötelező megadni, és döntött betűkkel: **dátum**

A fontosabb kulcsszavak, parancsok és általában a beírandók, kiemelten sötét betűkkel:

DATE 1993/09/01

Ha árnyékos kerettel van körülvéve, akkor a képernyőn a valóságban is ugyanúgy fog kiíródni, mint ahogyan a könyvben szerepel. (A szöveg a képernyőről lett "lelopva".)

A példákban az a rész van kiemelve, amit nekünk kell beírni, a számítógép üzenetei pedig normál betűkkel vannak nyomtatva.

Például:

```
C:\DOS>del *.arj /p
```

```
C:\DOS>_
```

A példában a "**del *.arj /p**" -t írtuk mi be, a "**C:\DOS>**" -t a számítógép írta ki. A "_" jel a kurzort szimbolizálja.

A bemutatott példákat ki lehet próbálni, ugyanazt az eredményt fogod kapni, mint ami a könyvben olvasható. A készenléti jel (**PROMPT**) által mutatott könyvtárnév lehet más is, nem kötelező ugyan abban a könyvtárban állni. Ha viszont nem adtunk meg keresési útvonalat (**PATH**), akkor a külső parancsok csak a DOS parancsok könyvtárából adhatók ki (lásd **PATH** parancsnál).

Most rátérünk a DOS használatára, a parancsok kiadására:

DATE

Dátumállítás

DATE [dátum]

Beállíthatjuk vele a számítógépünkben tárolt dátumot.

Belső parancs.

dátum: Az aktuális dátum.

formája: Általában -USA formátumban- (alapértelmezés),

[*hh-nn-ée*] vagy [*hh-nn-éééé*]

ahol a **hh** a hónap (1-12), **nn** a nap (1-31), **ée** az év (80-99) a XX. században (pl.: 1993-ban '93') vagy **éééé** az év (2000-2099) a XXI. században.

Akkor ha CONFIG.SYS fájlban meg van adva a

COUNTRY=036,852,C:\DOS\COUNTRY.SYS (lásd ott!)

sor, akkor magyar formátumban kell megadni a dátumot:

[*ée-hh-nn*] vagy [*éééé-hh-nn*]

Elválasztó karakterek lehetnek: (.) (-) (/). A dátum megadása nem kötelező, akkor kiírja a gép a rendszerben tárolt dátumot, amit **[ENTER]** -rel elfogadhatunk, vagy az előbb ismertetett módon módosíthatunk. Az újabb gépekben már lítium elem tárolja hosszú ideig (3-5 év) a beállított dátumot és időt, ami csak néha szorul pontosításra. Ha a rendszer indulásakor AUTOEXEC.BAT fájlt nem használunk, akkor a gép automatikusan bekéri a dátumot és az időt, ellenkező esetben csak akkor kéri, ha a fent említett fájlban benne van a **DATE** ill. a **TIME** (lásd ott) parancs.

Például:

```
C:\>DATE
Current date is Thu 1993-02-18
Enter new date (yy-mm-dd): 1993/04/17
C:\>DATE 2018.01.31

C:\>_
```

TIME

Időállítás

TIME [*idő*]

A számítógépünkben állíthatjuk be vele pontos időt.
Belső parancs.

idő : Az aktuális idő.

formái:Általában -USA formátumban- (alapértelmezés),

[*óra*:] vagy

[*óra*:*perc*] vagy

[*óra*:*perc*:*másodperc*] vagy

[*óra*:*perc*:*másodperc*.*századmp*] és [*a*|*p*]

ahol az *óra* (0-23), a *perc* (0-59), a *másodperc* (0-59), a *századmásodperc* (0-99) értéket képviselhet. Az '*a*' vagy '*p*' megadásának 12 órás kijelzés esetén van értelme, ahol az '*a*' a délelőtt ('*a*' - ANTE MERIDIEM) illetve a '*p*' a délután ('*p*' - POSTMERIDIEM) jele.

Elválasztó karakter a kettőspont (:), illetve amerikai formátumban a századmásodperc előtt a pont (.), magyar formátumban a vessző (,).

Az idő megadása nem kötelező, akkor kiírja a gép a rendszerben tárolt időt, amit **[ENTER]** -rel elfogadhatunk, vagy az előbb ismertetett módon módosíthatunk.

Az újabb gépekben már lítium elem tárolja hosszú ideig (3-5 év) a beállított dátumot és időt, ami csak néha szorul pontosításra.

Ha a rendszer indulásakor AUTOEXEC.BAT fájlt nem használunk, akkor a gép automatikusan bekéri a dátumot és az időt, ellenkező esetben csak akkor kéri,ha a fent említett fájlban benne van a DATE (lásd ott) ill. a TIME parancs.

Például:

```
C:\>TIME
Current time is 9:51:07,42
Enter new time: 10:00 a      (De. 10 ó 00 p.)

C:\>TIME 10:15:25,00      (De. 10 ó 15p 25mp 00szdmp).

C:\>_
```

PROMPT

Készüléti jel

PROMPT [szöveg]

A készületi jel (PROMPT) megjelenési formáját állítjuk be vele.

Belső parancs.

szöveg: Lehet tetszőleges *szöveg* vagy/és az alábbi *formátumjelek* variációjából összeálló karaktersor.

formátumjelek:

- \$Q - = (EQUAL) Egyenlőség jel
- \$S - \$ Dollár jel
- \$G - > (GREATER-THAN SIGN) Nagyobb jel
- \$L - < (LESS-THAN SIGN) Kisebb jel
- \$B - | (PIPE) Cső-jel
- \$_ - Soremelés (**[ENTER]** billentyű)
- \$H - Egy karakter visszatörlése
- \$T - (TIME) Az idő kiírása
- \$D - (DATE) Az dátum kiírása
- \$P - (PATH) Az aktuális meghajtó és könyvtár kiírása
- \$N - (NORMAL) Csak az aktuális meghajtó kiírása
- \$V - (VERSION) A DOS verziójának kiírása
- \$E - (ESCAPE) Az ESC kód (a 27-es ASCII kódú karakter)

Az alapértelmezett készületi jel: **C:>_** ami a **PROMPT \$n\$g** vagy a **PROMPT** paranccsal hívható elő.

A \$t, \$d, \$p, \$n, \$v formátumjelek az éppen aktuális értéküket jelenítik meg. A formátumjelek betűi kis- és nagybetűvel egyaránt használhatók. A kurzor ez után áll mindig, és előrébb lépni nem lehet. A begépelte parancssornak nem része. Célszerű valamilyen határoló karaktert használni a végén, ami lezárja a prompt-ot, és egyértelműen mutatja a parancssor kezdetét.

Ha a CONFIG.SYS fájl tartalmazza a

DEVICE=C:\DOS\ANSI.SYS (lásd ott!)

sort, akkor az ANSI képernyő- és billentyűzetkezelő szekvenciák is használhatók. Ezen parancsok kiadását is elősegíti a **PROMPT** parancs.

Példák:

<pre>>prompt \$p\$g D:\>_</pre>	<p><i>Általában használatos prompt:</i> (Beírandó parancs) (Eredmény)</p>
<pre>>prompt \$n\$g vagy prompt D></pre>	<p><i>Alapértelmezett prompt.</i></p>
<pre>>prompt \$d\$b Thu 1993-02-18 </pre>	<p><i>A dátum kerül a készületi jelbe, a határolójel pedig a " ".</i></p>

VER

DOS verziója

VER

A betöltött DOS verziószámát jeleníti meg.

Belső parancs.

Sok program csak abban a DOS környezetben fut, amiben írták. Paramétert nem kell megadni. Ha más verziójú a DOS ami bent van, akkor:

Incorrect dos version ("Nem megfelelő DOS verzió")

üzenetet kapjuk, ha a programot futtatni akarjuk.

Használata:

```
C:\>ver  
MS-DOS Version 5.00  
C:\>_
```

CLS

Képernyőtörlés

CLS

A képernyő törlésére szolgál.

Belső parancs.

A képernyő letörlése, gyakorlatilag annak szóközzel való feltöltését jelenti. A készletléti jelet a bal felső sarokba viszi. Paramétert nem kell megadni.

Használata:

```
C:\>cls
```

```
C:\>_
```

Könyvtárak, alkönyvtárak, fájlok

A most ismertetésre kerülő DOS parancs, a **DIR** előtt, tisztázzunk néhány alapfogalmat. A **DIR** parancs a **DIRECTORY** (Címtár, könyvtár) szó rövidítése. Használjuk még a **SUBDIRECTORY** (alkönyvtár) szót is egy könyvtárban lévő könyvtárak megnevezésére.

A **gyökérvényvtár** (\) az a könyvtár, amelyik egyik könyvtárnak sem az alkönyvtára. Azaz bármely könyvtárból indulunk is el visszafelé, mindig a gyökérvényvtárba jutunk. Ez az első és legfőbb könyvtárunk. Éppen ezért, hogy ez mindig áttekinthető legyen, ebben a könyvtárban csak a rendszerfájlok legyenek és azok, amelyeknek még itt kell lenniük. Minden felhasználó nyisson magának saját könyvtárat, ugyanis nincs rosszabb, mint amikor nem lehet áttekinteni egy könyvtár felépítését. Gondoljunk csak bele, milyen káosz lenne, ha minden fájl egy, a gyökérvényvtárba (\) tennénk. Több ezer fájl között, sohasem találnánk meg a keresett programot. Éppen ezért hozták létre annak a lehetőségét, hogy könyvtárakat hozhassunk létre. El is jutottunk a **könyvtárfához**. Ezen az ágas-bogas fán minden elágazásnál van egy könyvtár. Ezekben a könyvtárakban általában logikailag összetartozó fájlok vannak összegyűjtve. Erre utal a könyvtár neve is. A **könyvtár neve** is nyolc karakteres névből, és három karakteres **kiterjesztés**ből áll - mint a fájlneveké -, amelyben néhány kivételtől eltekintve, tetszőleges karakterek használhatók. Azok a karakterek amelyeket nem használhatunk:

* ? \ | . , ; : + = " [] < >

A könyvtáraknak általában nem szoktunk kiterjesztést adni, de lehet, sőt bizonyos esetekben célszerű is. Például: DOS.330 DOS.500 DOS.600, ahol a DOS könyvtárak a DOS parancsokat őrzik, de mindegyik egy másik verzióét. Mivel a DOS parancsokat általában a "DOS" nevű könyvtárban tartjuk, ezért a kiterjesztésükkel különböztetjük meg őket.

A **DIR** parancs segít abban, hogy a kiválasztott könyvtár fájljait kilistázzuk, az azokról tárolt információkat megjelenítsük. Lehet egy könyvtár fájljait listázni, de lehet akár az összes könyvtár tartalmát egy paranccsal kiíratni, akár a képernyőre, nyomtatóra, de akár fájlba is.

DIR

Könyvtárlista

DIR [*meghajtó*] [*elérési_út*] [*fájlnev*] [/P] [/W] [/S] [/B] [/L] [/O [[:] *rendező kód*]]
 [/A [[:] *jelzőbit*]]

A kívánt könyvtár(ak) állományainak és alkönyvtárainak listáját adja meg a megadott paramétereknek megfelelően.

Belső parancs.

Ha nem adunk meg egyetlen paramétert sem, akkor az aktuális könyvtárból listáz. Azonban elsődleges paraméterként mindig a DIRCMD környezeti változó értéket tekinti (lásd ott!). Kiírja a *meghajtó betűjelét*, a *lemez címkéjét* (ha van), *sorozatszámát*, és a listázandó könyvtár pontos *elérési útját*. Ezek után listázza az *összes alkönyvtárat* és *fájlt* amire a paraméterek vonatkoznak. A *fájlok neve* és *kiterjesztése* mellett megjelenik a *mérete* bajtokban, létrehozásának (vagy utolsó módosításának) *dátuma* és *ideje*, az országspecifikus információknak megfelelően.

(lásd CONFIG.SYS - COUNTRY - COUNTRY.SYS - DATE - TIME)

Az *alkönyvtárak neve* (és *kiterjesztése* -ha van-) mellett a méret helyett '<DIR>' felirat látható, majd a létrehozás *dátuma* és *ideje*. A listázás után kiírja a parancs a listázott *állományok számát*, *összes helyfoglalásukat* (nem azonos a fájlok méreteinek összegével) bajtokban, és alatta a rendelkezésre álló *szabad* (FREE) *terület méretét*.

A kilistázott állományok és a tényleges számuk nem mindig egyezik meg, mert a rejtett és a rendszerfájlok nem listázódnak (ha nem utasítjuk erre külön), viszont helyet foglalnak a lemezen. A fájlok összes helyfoglalása nem azonos a méreteinek összegével, mert egy 10 bajtos fájl is elfoglal egy egész klasztert a merevlemezen.

meghajtó: Amelyik meghajtón a listázandó könyvtár van. Ha elmarad akkor az aktuális meghajtót veszi. A meghajtó betűjele után a kettőspont (:) kötelező!

elérési_út: A listázandó könyvtár elérési útja az aktuális könyvtártól (pl.: '..\..\DOS') vagy a gyökérkönyvtártól indulva (pl.: '\IBM_PC\DOS'). Ha elmarad, akkor az aktuális könyvtárat veszi.

fájlnev: A listázandó fájl, vagy fájlok köre. Használhatunk helyettesítő karaktereket is (*) (?). Ha elmarad, akkor az alapértelmezésben minden fájl listáz, kivéve a rejtett (HIDDEN) és a rendszerfájlokat (SYSTEM).

A '* karakter helyettesít minden utána következő hiányzó karaktert. A '?' pedig azt az egy karaktert helyettesíti, amelyiknek a helyén áll. (például: minden fájl: *.* vagy csak *egy pont karakter*. Ez szintén azonos a *.* formával.)

/p: Oldalakra (PAGE) tagolást végzi, egy oldalra általában 23 sort ír ki, majd üzenetet küld, hogy nyomjunk meg egy billentyűt:

press any key to continue . . .

A billentyű leütése után válaszol, majd folytatja a kiírást:

(continuing C:\DOS)

Általában 25 soros a monitor képernyője, ezért ír 23 sort, plusz két sor üzenetet (lásd előbb!) egy oldalra. Azonban EGA, VGA, ... monitor esetén lehetőség van 33 - 43 soros megjelenítésre is. A DIR parancs automatikusan alkalmazkodik a képernyő méretéhez, és mindig teleírja azt.

- ***/w:** A kiírás rövid formában, szavakban (WORD) jelenik meg, méret, dátum és idő nélkül, csak a fájlnevek, illetve szögletes zárójelben az alkönyvtárnevek. Egy sorban öt név szerepel. Általában akkor használjuk, ha a fájlokhoz kötődő információkra nem vagyunk kíváncsiak, csak a nevek között keresgélünk.
- /b:** Rövid kivonatos (BARE) formában jelenik meg a lista, csak a nevek, minden információ és végösszesen nélkül. Ha ezt használjuk, akkor a '/w' kapcsoló hatástalan. Egy sorba csak egy állománynév kerül. Ha a '/s' paramétert is használjuk, akkor a fájlnevek előtt az **abszolút elérési útja** is szerepel.
- ***/s:** A listázott könyvtár alkönyvtárainak (SUBDIRECTORY) tartalmát is kiírja. A listázáskor kiírja a lemezfejléccadatok után a listázandó könyvtár, illetve alkönyvtárainak teljes nevét (MEGHAJTÓ:\ELÉRÉSI_ÚT\), és a tartalmát a kívánt módon. A listák végén összesíti a kiírt fájlokat (darabszám, elfoglalt terület bájtokban), majd a kiírás végén főösszesenben is kiírja az összes listázott fájl darabszámát, az általuk elfoglalt terület méretét bájtokban, és a rendelkezésre álló szabad terület méretét.
- ***/l:** Minden név kisbetűvel írva (LOWER CASE) listázódik a többi paraméter által meghatározott módon..
- ***/o: rendezőkód** A rendezőkód dönti el milyen sorrendben írja ki a könyvtárakat és a fájlneveket. Ha elmarad, akkor a beolvasás sorrendjében írja ki az állományokat. Ha csak a rendezőkódot hagyjuk el, akkor először a könyvtárnevek, majd a fájlnevek kerülnek kiírásra ábécé sorrendben. A megadott rendezőkódok között nem lehet szóköz. Ha több szempont szerint akarjuk rendezni, akkor a legelől álló kód szerint rendezzi, és azonos adatok esetén figyelni csak a következő kódot.

rendezőkódok:

/N - Név (NAME) szerint, ábécé sorrendben.

/E - Kiterjesztés (EXTENSION) szerint, ábécé sorrendben.

/S - Méret (SIZE) szerint, legkisebbtől a legnagyobbig.

/D - Dátum (DATE) szerint, a legrégebbtől a legújabbig.

/G - Csoport (GROUP) szerint, ahol a könyvtárak és a fájlok külön csoportba kerülnek. Könyvtáraké az első csoport.

(-) A negatív előjel használatakor ellenkező sorrendben íródnak ki az állományok.
(Például: 'DIR /O:-n' név szerint sorban Z-től A-ig.)

/a: jelzőbitek: Csak a megadott jelzőbitekkel rendelkező nevek listázódnak ki. Ha a '/a' kapcsolót használjuk jelzőbit megadása nélkül, akkor minden fájl listázódik (még a rejtett és rendszerfájlok is). A kettőspont megadása nem kötelező, csak célszerű. A megadott jelzőbitek között nem lehet szóköz. *lásd 88 oldal*

>dir /a:d

jelzőbitek:

A **jelzőbitek** (ATTRIBUTUM), a fájlok és könyvtárak tulajdonságait jelzik. Több csoportba sorolhatjuk az állományokat (például: fájlok és könyvtárak, látható és rejtett fájlok, írható-olvasható és csak olvasható fájlok...stb). Ezen jelzőbiteket használja néhány parancs is a fájlok másolásakor. Az **ATTRIB** paranccsal állíthatjuk őket. (lásd ott!)

D - Könyvtár (DIRECTORY)	Csak könyvtárnak lehet, fájlnek nem.
H - Rejtett (HIDDEN)	Alapértelmezésbeli DIR parancs nem listázza.
S - Rendszer (SYSTEM)	Alapértelmezésbeli DIR parancs nem listázza.
R - Csak olvasható (READ ONLY)	Nem írható át, nem törölhető (de feloldható!).
A - Elmentendő fájl (ARCHIVE) .	

Minden fájl megkapja ezt a bitet (A) létrehozásakor vagy módosításakor. Ha **BACKUP** -pal elmentjük (biztonsági másolatot készítünk a fájlokról), akkor törlődik ez a bit, ami jelzi, hogy archiválva lett a fájl.

(-) A negatív előjel

Használatkor az ellenkezője értendő a leírtaknak.

(Például: ' **DIR /a:A-S** ' minden archiválandó, nem rendszerfájlt kiír.)

Egyéb tartozékok:

DIRCMD - környezeti változó (**DIR**ectory **CoM**mand).

Ennek értékét veszi alapul először a **DIR** parancs. Ha megadunk ezzel ellenkező értelmű parancsot is, akkor azt fogja használni a parancs végrehajtásakor.

Beállítása, például:

```
SET DIRCMD=/o/a/p
```

A **DIR** parancs, így paraméterek nélkül is ábécé sorrendben írja ki az összes (bármilyen attribútumú) fájlt, és ha megtelt az oldal, akkor egy billentyű leütése után folytatja a kiírást. Ha megadunk valamilyen paramétert, akkor az felülértékeli a **DIRCMD**-ben megadottakat.

TEMP - Környezeti változó (**TEMP**orary-ideiglenes). Akkor használjuk, ha a listát egy másik parancsnak átadjuk a (| - pipe) parancscsatoló használatával.

Beállítása, például:

```
SET TEMP=c:\text\temp
```

Példák a **DIR** parancs használatára:

A példákat egy **D:** nevű merevlemez **DOS** könyvtárán mutatom be, a listák lerövidítését függőleges pontsor jelzi. A **D:** meghajtó címkeje **DISK_CIMKE~** (11 karakter), a sorozatszáma **256D-0EF4**.

Kivonatos lista, melyben csak a nevek szerepelnek, soronként öt név. A meghajtóról viszont megkapunk minden információt:

```
D:\DOS>dir /w

Volume in drive D is DISK_CIMKE~
Volume Serial Number is 256D-0EF4
Directory of D:\DOS

ANSI.SYS      APPEND.EXE    ASSIGN.COM    ATTRIB.EXE    AUTOEXEC.BAT
COMMAND.COM   CONFIG.SYS    CONFIG.DOC    COUNTRY.SYS   DEBUG.EXE
DIRINFO       DISKCOPY.COM  DISPLAY.SYS   DOSKEY.COM    DRIVER.SYS
:             :             :             :             :
[D5]          EGA.SYS      EXE2BIN.EXE   FASTOPEN.EXE  FIND.EXE
XCOPY.EXE     [.]          [..]
              43 file(s)   537232 bytes
              4575232 bytes free

D:\DOS>
```

Lista név szerint sorba rendezve, és a képernyő beteltkor várakozás egy billentyű leütésre:

```
D:\DOS>DIR /o:n /p

Volume in drive D is DISK_CIMKE~
Volume Serial Number is 256D-0EF4
Directory of D:\DOS

ANSI      SYS      9029 91-04-09  5:00
APPEND    EXE      10774 91-04-09  5:00
ASSIGN    COM      6399 91-04-09  5:00
ATTRIB    EXE     15796 91-04-09  5:00
AUTOEXEC  BAT      662 92-06-01 13:14
COMMAND   COM     47845 91-04-09  5:00
CONFIG    SYS      416 92-06-01  7:47
CONFIG    DOC     7861 92-05-22 12:29
COUNTRY   SYS     17069 91-04-09  5:00
DEBUG     EXE     20634 91-04-09  5:00
DISKCOPY  COM     11793 91-04-09  5:00
DISPLAY   SYS     15792 91-04-09  5:00
DOSKEY    COM     5883 91-04-09  5:00
DRIVER    SYS     5409 91-04-09  5:00
D5        <DIR>    93-01-13 13:51
EGA       SYS     4885 91-04-09  5:00
EXE2BIN   EXE     8424 91-04-09  5:00
FASTOPEN  EXE     12050 91-04-09  5:00
FIND      EXE     6770 91-04-09  5:00
FORMAT    COM     32911 91-04-09  5:00
Press any key to continue . . .

(continuing D:\DOS)
FORMAT    TXT      1966 93-05-12  5:00
:
:
UNDELETE  EXE     13924 91-04-09  5:00
XCOPY     EXE     15804 91-04-09  5:00
.         <DIR>    92-08-19  8:03
..        <DIR>    92-08-19  8:03
              43 file(s)   537232 bytes
              4579328 bytes free

D:\DOS>
```

Lista jelzőbit szerint, csak a 'D'- könyvtár jelzőbittel rendelkezőket listázza, vagyis az alkönyvtárneveket:

(Figyeljük meg, a könyvtárak csak katalógus-bejegyzések, nem itt foglalnak helyet lemezen.)

```
D:\DOS>dir /a:d

Volume in drive D is DISK_CIMKE~
Volume Serial Number is 256D-0EF4
Directory of D:\DOS

D5          <DIR>    93-01-13  13:51
.           <DIR>    92-08-19   8:03
..          <DIR>    92-08-19   8:03
3 file(s)                   0 bytes
                          4579328 bytes free

D:\DOS>_
```

Lista az aktuális könyvtár alkönyvtáiról is:

(Figyeljük meg, hogy az oldalakra tagolás a ' /p ' paraméter elmaradása esetén is megtörténik, és a listák is ábécé szerint sorrendbe vannak rendezve, mert a DIRCMD környezeti változó tartalmazza ezt (DIRCMD=/o:n /p).)

```
d:\>DIR /s

Volume in drive D is DISK_CIMKE~
Volume Serial Number is 256D-0EF4

Directory of D:\DOS

ANSI      SYS      9029 91-04-09  5:00
APPEND    EXE      10774 91-04-09  5:00
:
:
XCOPY     EXE      15804 91-04-09  5:00
.         <DIR>    92-08-19  8:03
..        <DIR>    92-08-19  8:03
          43 file(s)  537232 bytes

Directory of D:\DOS\D5

ANSI      SYS      9029 91-04-09  5:00
APPEND    EXE      10774 91-04-09  5:00
APPNOTES  TXT      9701 91-04-09  5:00
:
:
XCOPY     EXE      15804 91-04-09  5:00
XXUNFRMT COM  18576 91-04-09  5:00
Press any key to continue . . .

(continuing D:\DOS\D5)
4201     CPI      6404 91-04-09  5:00
.         <DIR>    93-01-13 13:51
..        <DIR>    93-01-13 13:51
          108 file(s) 2239615 bytes

Total files listed:
          151 file(s) 2776847 bytes
                          4577280 bytes free

D:\DOS>_
```

Lista a állománynevekről egyéb információk nélkül:

```
D:\DOS>DIR /b
ANSI.SYS
APPEND.EXE
ASSIGN.COM
ATTRIB.EXE
AUTOEXEC.BAT
COMMAND.COM
CONFIG.SYS
CONFIG.DOC
COUNTRY.SYS
DEBUG.EXE
DIRINFO
:
:
...stb.
D:\DOS>_
```

Lista kisbetűkkel: (Továbbra is érvényesül a DIRCMD hatása!)

```
D:\DOS>DIR /L
Volume in drive D is DISK_CIMKE~
Volume Serial Number is 256D-0EF4
Directory of D:\DOS

ansi      sys      9029 91-04-09  5:00
append   exe     10774 91-04-09  5:00
assign    com      6399 91-04-09  5:00
attrib    exe     15796 91-04-09  5:00
:
:
xcopy     exe     15804 91-04-09  5:00
Press any key to continue . . .

(continuing D:\DOS)
.          <DIR>      92-08-19  8:03
..         <DIR>      92-08-19  8:03
          43 file(s)  537232 bytes
                               4575232 bytes free
D:\DOS>_
```

Az ismertetett paraméterek természetesen kombinálhatók is, de a példákban csak egyesével alkalmaztam azokat, hogy a különbségek jól megmutatkozzanak.

A kapott listákat nem csak a képernyőn nézhetjük meg, hanem át is irányíthatjuk a kimenetet - a listát - más egységre is. Például nyomtatóra, vagy akár egy fájlba is, amit egy szövegszerkesztővel megtekinthetünk, vagy megformázhatunk tetszőlegesen. Mindezekről részletesen az "*Állományok átírányítása*" című fejezetben olvashattok.

Műveletek könyvtárakkal

Ha a gyökérkönyvtárban (\) létrehozunk alkönyvtárakat, és abba belépünk, akkor ott is létrehozhatunk alkönyvtárakat. Ha ezt sokáig bonyolítjuk, akkor kapunk egy ágas-bogas könyvtárfát. A létrehozott alkönyvtárakban (MAKE DIRECTORY - MKDIR - MD), tudunk lépkedni, aktuális könyvtárat váltani (CHANGE DIRECTORY - CHDIR - CD), és kitörölni alkönyvtárakat (REMOVES DIRECTORY - RMDIR - RD).

MKDIR

MD

Könyvtárak létrehozása

MKDIR [*meghajtó:*] *elérési_út*

MD [*meghajtó:*] *elérési_út*

Könyvtárat lehet vele létrehozni.

Belső parancs.

meghajtó: A meghajtó betűjele, ahol létre akarunk hozni egy könyvtárat. Ha elmarad, akkor az aktuális meghajtón hozza létre. A meghajtónévhez tartozik a kettőspont (:) kiírása is.

elérési_út: Annak a könyvtárnak az elérési útja, ahová az új könyvtárat létrehozzuk, és az új könyvtárnév. Ha csak az új könyvtárnevet adjuk meg, akkor az aktuális könyvtárban hozza létre. (Ha azt sem, akkor mit akarunk tulajdonképpen ?)

A létrehozott könyvtárban két bejegyzést fogunk találni:

```
.      <DIR>    92-08-19  8:03
..     <DIR>    92-08-19  8:03
```

Az első - egy pont nevű könyvtár (.) - az aktuális könyvtár tartalmát mutatja, a második - kettő pont nevű (..) - pedig az úgynevezett szülőkönyvtár tartalmát.

CHDIR

CD

Könyvtársváltás

CHDIR [*meghajtó:*] [*elérési_út*]

CD [*meghajtó:*] [*elérési_út*]

Könyvtársváltásra és az aktuális könyvtár nevének kiírására alkalmas.

Belső parancs.

meghajtó: A meghajtó betűjele, ahol az aktuális könyvtárat meg akarjuk változtatni. Ha nem azon a meghajtón vagyunk, akkor is megváltoztatja az aktuális könyvtárat azon a meghajtón, de nem fogjuk azonnal látni, csak ha meghajtót váltunk. Ha nem adjuk meg ezt a paramétert, akkor az aktuális meghajtóra vonatkozik a parancs végrehajtása. A meghajtó névhez tartozik a kettőspont (:) kiírása is.

elérési_út: Annak a könyvtárnak az elérési útja, ahová lépni akarunk, azaz aktuálissá tesszük a megadott könyvtárat. Ha nem adjuk meg, akkor kiírja a meghajtó és a könyvtár nevét, annak a meghajtónak, amelyiket megadtuk, vagy az aktuálisét.

A CHDIR parancs megadásának formái:

CD Paraméter nélkül kiadva a parancsot, kiírja az aktuális meghajtó és az aktuális könyvtár nevét. Erre akkor lehet szükség, ha a készenléti jel nem a meghajtó és könyvtár nevét mutatja (Lásd PROMPT parancs!).

CD A: Hasonló az előző parancshoz, de itt a megadott meghajtó nevét és aktuális könyvtárát írja ki. Akkor használjuk, ha egy másik meghajtóra akarunk másolni, annak az aktuális könyvtárába, és annak helyességét ellenőrizzük le.

CD DOS Az aktuális meghajtón, az aktuális könyvtárban van egy 'DOS' nevű könyvtár, és ezt tesszük aktuálissá. Ha a megadott könyvtár nem itt van, akkor nem találja meg, hibaüzenetet ad rá. Ezt nevezzük *relatív útvonalmegadás*nak.

CD \IBM_PC\DOS\V3.30 Az aktuális meghajtón lép a megadott könyvtárba. Tetszőleges helyen állhatunk, mert a könyvtár neve egyértelműen meg van adva a gyökérkönyvtárból kiindulva.. Ezt nevezzük *abszolút útvonalmegadás*nak.

CD A:\BASIC\PRG A megadott meghajtón, a megadott könyvtárba lép. Ha ez a meghajtó nem az aktuális, akkor is fog könyvtárat váltani azon a meghajtón. Ha az aktuális meghajtón váltunk könyvtárat, akkor nem kell megadni a meghajtó nevét.

RMDIR

RD

Könyvtárak törlése

RMDIR [*meghajtó:*] *elérési_út*

RD [*meghajtó:*] *elérési_út*

Könyvtár törlésére alkalmas.

Belső parancs.

meghajtó: A meghajtó betűjele, ahol törölni akarunk egy könyvtárat. Ha nem azon a meghajtón vagyunk, akkor is megszünteti a könyvtárat. Ha elmarad a meghajtó megadása, akkor az aktuális meghajtón keresi a törlendő könyvtárat. A meghajtó névhez tartozik a kettőspont (:) kiírása is. FONTOS: Csak üres könyvtárat lehet törölni, amelyben csak az a kettő bejegyzés lehet, amit az MD parancs hozott létre:

([.] és [.]) !

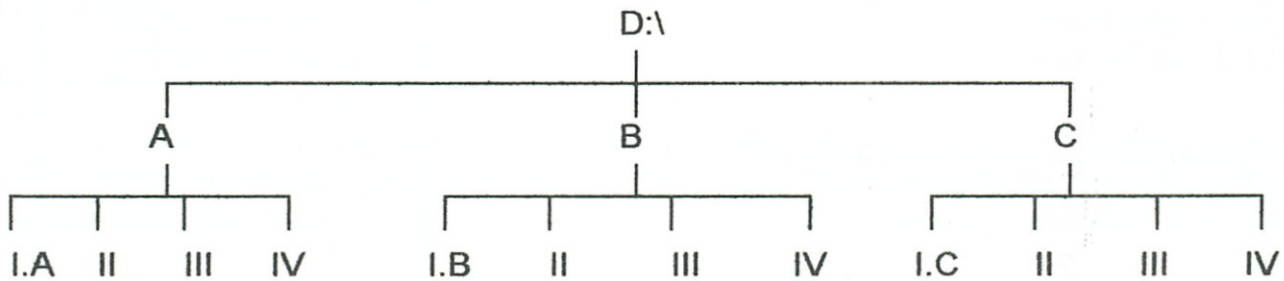
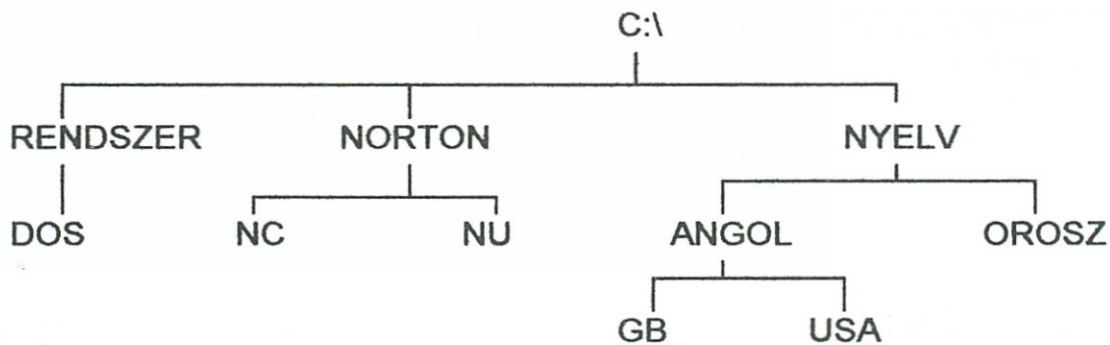
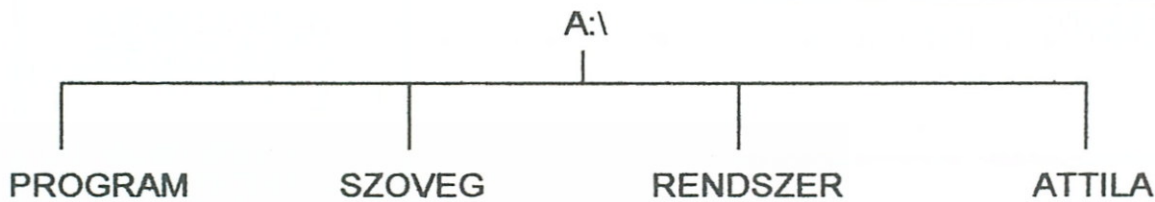
elérési_út: Annak a könyvtárnak a neve és az elérési útja, amelyiket ki akarjuk törölni.

Ezt a paramétert mindig kötelező megadni. Ha csak egy könyvtárnevet adunk meg, akkor az aktuális könyvtárban kell lennie a törlendő könyvtárnak.

Az aktuális könyvtárat nem lehet kitörölni. Ehhez vissza kell lépni, az aktuális könyvtár szülőkönyvtárába (' CD .. ' paranccsal). Ha a gyökérkönyvtárban állunk, akkor onnan nem tudunk visszalépni, mert annak nincs szülőkönyvtára, hiszen ez az első könyvtár a lemezen. Ebből következik, hogy a gyökérkönyvtár nem is törölhető ki.

Példák a CD - MD - RD parancsok használatára:

A példákban szereplő parancsok az alábbi meghajtókra és könyvtárstrukturákra vonatkoznak:



* A:\ C:\RENSZER D:\B\IV	* Az aktuális meghajtó. A jelenlegi készletléti jel: A:\>_
--------------------------------	--

```
A:\>cd program
```

```
A:\PROGRAM>_
```

* A:\PROGRAM C:\RENSZER D:\B\IV

```
A:\PROGRAM>cd \rendszer
```

```
A:\RENSZER>_
```

* A:\RENSZER C:\RENSZER D:\B\IV

```
A:\RENSZER>cd d:\a\i.a
```

```
A:\RENSZER>_
```

* A:\RENSZER C:\RENSZER D:\A\I.A
--

A:\RENDSZER> cd d:..	* A:\RENDSZER C:\RENDSZER D:\A
A:\RENDSZER>_	
A:\RENDSZER> D:	A:\RENDSZER C:\RENDSZER * D:\A
D:\A>_	
D:\A> cd .	A:\RENDSZER C:\RENDSZER * D:\A
D:\A>_	
D:\A> cd ..	A:\RENDSZER C:\RENDSZER * D:\
D:\>_	
D:\> rd a Invalid path, not directory or directory not empty	A:\RENDSZER C:\RENDSZER * D:\
D:\>_	
D:\> cd rendszer Invalid directory	A:\RENDSZER C:\RENDSZER * D:\
D:\>_	
D:\> md \dbase\clipper Unable to create directory	A:\RENDSZER C:\RENDSZER * D:\
D:\>_	

1. Lépünk be a PROGRAM könyvtárba:

Belépni egy könyvtárba annyit jelent, hogy az lesz a továbbiakban az aktuális könyvtár. Az itt látható DOS készenléti jel a **PROMPT \$P\$G** paranccsal hívható elő. A számítógép az aktuális könyvtárban keresi a PROGRAM nevű bejegyzést, aminek kiterjesztése nincs, és a 'D' jelzőbitje be van állítva, azaz könyvtár.

2. Lépünk át a RENDSZER könyvtárba:

Átlépni egy másik könyvtárba annyit jelent, hogy nem az aktuális könyvtárból nyílik az, így belépni nem tudunk, de az elérési útját megadhatjuk, vagy az aktuális könyvtárból, vagy a gyökérkönyvtárból indulva. Példánkban ez utóbbit láthatjuk. (Az átlépés másik módja: **CD ..\RENDSZER**)

3. Lépünk át a D: meghajtón a B\IV könyvtárból az A\I.A könyvtárba:

A jobb oldalon láthatjuk, hogy a D: meghajtón az aktuális könyvtár megváltozott, de a készenléti jel (PROMPT) nem. Az átlépést ismét a gyökérkönyvtárból indulva adtuk meg.

(Az átlépés másik módja: `CD ..\..\A\I.A`, ami akkor lenne előnyösebb, ha hosszabb lenne a gyökérkönyvtárból az út az aktuális könyvtárba.)

4. Lépünk át a D: meghajtón a szülőkönyvtárba:

Szülő könyvtárnak nevezzük azt a könyvtárat, amelyikből közvetlenül be tudunk lépni az adott könyvtárba.

5. Váltunk meghajtót:

Meghajtót váltani úgy lehet, hogy beírjuk a meghajtó betűjelét és egy kettőspontot (:). Ekkor az újonnan kijelölt meghajtóra lépve, annak aktuális könyvtárába lépünk. Amint láthatjuk a D: meghajtón az aktuális könyvtár az 'A', ami nem a kiindulási állapot. Tehát úgy is lehet könyvtárat váltani egy adott meghajtón, hogy nem is az az aktuális. Az előző parancsokat úgy is beírhattuk volna, hogy előbb átlépünk a D: meghajtóra, majd lépünk a könyvtárakban.

6. Ennek a parancsnak nincs semmi értelme, bár hiba nincs benne. Csupán arra utasítjuk a gépet, hogy lépjen abba a könyvtárba, ahol jelenleg is áll. Ugyanis a [.] nevű könyvtár az aktuális könyvtár adatait tartalmazza.

7. Lépünk vissza a szülőkönyvtárba:

Meghajtó betűjelet nem adunk meg, így az aktuális meghajtón lép vissza egyet a könyvtárfán.

8. Töröljünk ki egy könyvtárat ami nem üres:

A gép azonnal válaszol is: "Helytelen elérési út, nincs könyvtár, vagy a könyvtár nem üres". Ezek szerint ha egy nem létező könyvtárat akarunk kitörölni, akkor is ugyanezt az üzenetet kapjuk.

9. Lépünk be egy nem létező könyvtárba:

A gépünk válasza: "Érvénytelen könyvtár".

10. Hozzunk létre egy nem létező könyvtárban egy könyvtárat:

Mivel a szülőkönyvtár nem létezik, ezért abban nem is tud könyvtárat létrehozni. A gép válasza: "Nem képes létrehozni a könyvtárat".

A következő példákban láthatjuk, hogyan lehet létrehozni a C: meghajtón látható könyvtárstruktúrát, az A: meghajtó PROGRAM könyvtárába. Több módja is van egy-egy könyvtár létrehozásának, mindegyik jó, ha helyesen alkalmazzuk.

1. Először is célszerű arra a meghajtóra lépni, amelyiken a könyvtárakat létrehozzuk (A:).

2. Utána belépünk abba a könyvtárba, ahová létrehozzuk a struktúrát (A:\PROGRAM).

3. Szintenként haladva, előbb az aktuális könyvtárba kerülő könyvtárakat hozzuk létre, majd az abban lévőket.

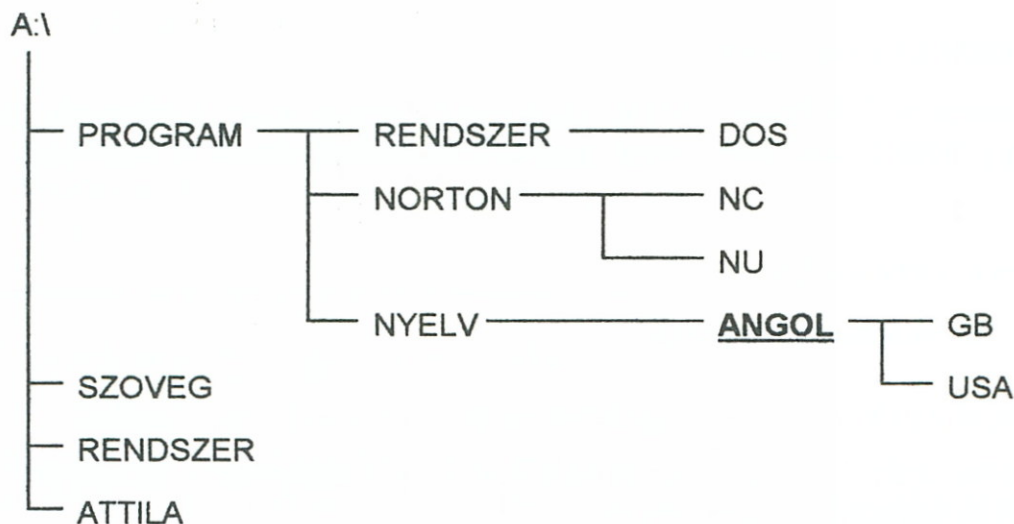
4. A PROGRAM könyvtárból hozzuk létre a DOS - NC - NU könyvtárakat is. (Nem lépünk át az új könyvtárakba.)

5. Itt sorba belépünk a létrehozott könyvtárakba, és ott hozzuk létre az újabbakat.

Lássuk hogyan is megy ez valójában:

```
D:\>a:
A:\RENDSZER>cd \program
A:\PROGRAM>md rendszer
A:\PROGRAM>md Norton
A:\PROGRAM>md nyelv
A:\PROGRAM>md rendszer\dos
A:\PROGRAM>md norton\nc
A:\PROGRAM>md norton\nu
A:\PROGRAM>cd nyelv
A:\PROGRAM\NYELV>md angol
A:\PROGRAM\NYELV>md orosz
A:\PROGRAM\NYELV>cd angol
A:\PROGRAM\NYELV\ANGOL>md gb
A:\PROGRAM\NYELV\ANGOL>md usa
A:\PROGRAM\NYELV\ANGOL>_
```

Most lássuk mi is lett valójában:



TREE

Könyvtárfelepítés rajza

TREE [*meghajtó*] [*elérési_út*] [/F] [/A]

A parancs a kért könyvtárstruktúrát rajzolja ki.

Külső parancs.

meghajtó: Az a meghajtó, amelyiken a kirajzolandó könyvtár van. Ha elmarad, akkor az aktuális meghajtón keres.

elérési_út: A kirajzolandó könyvtár helye és neve a könyvtárstruktúrában. Ha elmarad, akkor az aktuális könyvtárat rajzolja fel.

/f: Kiírja a könyvtár(ak)ban lévő fájlok (FILE) nevét is.

/a: Nem használ a könyvtárfa kirajzolásakor grafikus karaktereket (ASCII kód <128). Erre akkor van szükségünk, ha olyan kódlapot használunk, amelyik nem tartalmaz grafikus karaktereket, vagy speciális esetekben nyomtatáskor.

Példák:

A **/F** paraméterrel megjeleníti a kirajzolt könyvtárak fájljait is, egyéb információk nélkül:

```
D:\DOS>tree /f
Directory PATH listing for Volume DISK_CIMKE~
Volume Serial Number is 256D-0EF4
D:.\
| LABEL.EXE
| COMMAND.COM
| CONFIG.SYS
| CONFIG.DOC
|
| : VEM.SYS
|
|-----D5
|
|     DOSSHELL.EXE
|     DOSSHELL.HLP
|     COMMAND.COM
D:\DOS>_
```

A teljes könyvtárstruktúra képe:

```
D:\DOS>tree \
Directory PATH listing for Volume DISK_CIMKE~
Volume Serial Number is 256D-0EF4
D:\
|---DOS
|   |---D5
|---ARCHIVE
|---NORTON
|   |---NC
|   |---NG
|   |---NU
|---ME
|   |---DOC
|   |---HELP
|   |---MAC
|   |---SRC
|---TEMP
|---UTILS
|---WINDOWS
D:\DOS>_
```

Ezt a rajzolatot 'D:\>_' prompt mellett kiadott TREE (paraméter nélküli) paranccsal is megkaphatjuk. Paraméter nélkül az aktuális könyvtárat rajzolja fel:

```
D:\DOS>tree /a
Directory PATH listing for Volume DISK_CIMKE~
Volume Serial Number is 256D-0EF4
D:.\
\---D5

D:\DOS>_
```

Műveletek lemezekkel

VOL

Lemzcímke lekérdezés

VOL [*meghajtó* :]

A parancs kiírja a lemez címkéjét és sorozatszámát.

Belső parancs.

meghajtó: Annak a meghajtónak a betűjele kettősponttal(:), amelyiknek a címkéjére kíváncsiak vagyunk. Ha elmarad, akkor az aktuális meghajtóra vonatkozik. (VOLUME - kötet)

Példa:

```
D:\TEMP>vol

Volume in drive D is DISK_CIMKE~
Volume Serial Number is 256D-0EF4

D:\TEMP>_
```

LABEL

Lemzcímke állítás

LABEL [*meghajtó* :] [*címke*]

Meghajtók címkéje írható, törölhető vele.

Külső parancs.

meghajtó: Annak a meghajtónak a betűjele kettősponttal(:), amelyiknek a címkéjét (LABEL) meg akarjuk változtatni. Ha elmarad, akkor az aktuális meghajtóra vonatkozik.

címke: Egy tetszőleges karaktersorozat, ami az adott meghajtónévvel illetett háttértároló egység (hajlékony lemez vagy a merevlemez) neve lesz.

A címkében fel nem használható karakterek : 21 db

? / \ | . , ; : + = - " ^ & [] () < >

Megjegyzések:

- A címke felírásakor a kisbetűket nagyra konvertálja, még az ékezetes betűket is. Az aktuális kódlapot (látsd ott!) veszi figyelembe, de más kódlap használata esetén, más karakterek fognak megjelenni, már DIR parancsnál is. Ezért az ékezetes betűkkel csínján bánjunk, nehogy meglepetés érjen bennünket! - Ha ékezetes karaktereket is használtunk, akkor célszerű azonnal egy VOL parancsot is kiadni, és így megláthatjuk mi lett a címkénkből.

Példák:

```

D:\DOS>label
Volume in drive D is DISK_CIMKE~
Volume Serial Number is 256D-0EF4
Volume label (11 characters, ENTER for none)? abcd[*?]
Invalid characters in volume label

Volume label (11 characters, ENTER for none)? számítógép

Volume label (11 characters, ENTER for none)?

Delete current volume label (Y/N)? y

D:\DOS>vol
Volume in drive D is SZÁMÍTÓGÉP
Volume Serial Number is 256D-0EF4

D:\DOS>vol
Volume in drive D is SZ|M TαGÉP
Volume Serial Number is 256D-0EF4

D:\DOS>vol
Volume in drive D is has no label
Volume Serial Number is 256D-0EF4

D:\DOS>_

```

Megjegyzés a példákhoz:

1. Az első esetben tiltott karaktereket is beírtunk, erre válaszol a számítógépünk, hogy:

"Nem megfelelő karakterek a kötetcímkében".

 Itt jegyzem meg, hogy ha tizenegy karaktert már beírtunk, akkor utána minden leütésre sípoló hang lesz a válasz, mert csak 11 karakter írható be.
2. A másik változat szerint, csupa kisbetűvel írjuk be a címke nevét, és az utána következő VOL paranccsal nézzük meg mi lett belőle. Ugyanis a parancs a beírt címkénév betűit nagybetűre konvertálja az aktuális kódlap szerint.
3. A harmadik lehetőségünk, hogy nem írunk be semmit sem, és akkor a számítógép megkérdezi, hogy:

"Töröljem a jelenlegi kötetcímkét (Igen/Nem)?"
- 2/A. Ha olyan számítógépen írtuk fel a címkét ahol a 852-es kódlap volt az aktuális, és ugyan azon a gépen kérünk egy DIR parancsot, akkor semmi különöset nem látunk rajta, csupa nagy betűkből áll a címkénk.
- 2/B. Ha az aktuális kódlap például 437-es, akkor a megjelenő címkénk furcsa képet fog mutatni. (A kódlapokról bővebben a MODE parancsnál olvashatunk.)
- 2/C. Ha egy kötetnek nincs címkéje, akkor a számítógépünk a címke helyett azt írja ki, hogy:

'has no label' vagyis 'nincs címke'

CHKDSK

Lemezvizsgálat

CHKDSK [*meghajtó:*] [[*elérési_út*] *fájlnev*] [/F] [/V]

Az állományokat és azok foglaltsági táblázatát (FAT) ellenőrzi.

Külső parancs.

meghajtó: Annak a meghajtónak a betűjele, amelyiken az ellenőrizendő fájlok vannak. Ha elmarad, akkor az aktuális meghajtón lévő összes fájlt ellenőrzi.

elérési_út: A vizsgálandó fájlok helye a meghajtón. Csak akkor kell megadni, ha a *fájlnev* paramétert is megadjuk.

fájlnev: Az ellenőrizendő fájl vagy fájlok köre. Több fájl esetén helyettesítő karakterek is használhatók.

/f: Az elveszett és megtalált szektorokat írja fel a gyökérvényvtárba a 'FILEnnnn.CHK' nevű fájlba, ahol *nnnn* 0001-től kezdődő sorszám.

/v: Kiírja az éppen vizsgált állomány nevét az elérési úttal együtt.

A parancs a lemezen fizikai hibát nem keres (BAD SECTORS), csak logikait, de kiírja azokat is. Ha talál valamilyen hibát, akkor azt kiírja, de nem javítja ki, csak akkor, ha a */f* paramétert is használjuk. A parancs arra is használható, hogy információkat kapjunk a kötetten elhelyezkedő fájlokról, azok helyfoglalásáról, a szabad hely méretéről...stb.

Példa:

```
D:\DOS>chkdsk a:
Volume MORATI_USER created 06-03-1992 12:25p
Volume Serial Number is 1873-4819

    362496 bytes total disk space
      1024 bytes in 1 directories
      1024 bytes in 1 user files
    360448 bytes available on disk
      1024 bytes in each allocation unit
      354 total allocation units on disk
      352 available allocation units on disk

    655360 total bytes memory
    563184 bytes free

D:\DOS>
```

A kiírt adatok:

- 1 Teljes lemezkapacitás (360 kb, egy DS DD -s lemez).
- 2 Könyvtárak száma és helyfoglalása (1 db 1024 bájt).
- 3 Fájlok száma és helyfoglalása (1 db 1024 bájt).
- 4 Rendelkezésre álló lemezterület bájtokban.
- 5 Minden kiosztási terület (klaszter) mérete 1024 bájt.
- 6 A teljes lefoglalható terület klaszterekben mérve.
- 7 A szabad terület nagysága klaszterekben.

8 A teljes memória mérete.

9 A szabad memória mérete.

Az adatok lemezen való tárolásának a módja, hogy a lemezen lévő területet *szektorokra* osztják fel. (Egy szektor mindig 512 bájt. Ettől eltérő szektorméretet csak memórialemezen (RAMDRIVE.SYS - lásd "DOS alapismeretek II." -ben) lehet kialakítani.) Az egymás felett elhelyezkedő szektorokat *klasztereknek* nevezzük. Egy klaszter mérete hajlékony lemezen 1024 bájt, merevlemezen 4096 bájt vagy annak többszöröse. Ezeknek a területeknek a foglaltságát őrzi a *fájlok foglaltsági táblázata* (FAT - FILE ALLOCATIONS TABLE). Egy fájl felírásakor a DOS folyamatosan teleírja a szektorokat egy klaszteren belül, majd lép a következőre. Az állomány utolsó bájtjai a klaszteren belül, az egyik szektorba kerülnek. Az a klaszter, amelyikbe akár egyetlen bájt is került, az már foglaltnak van bejegyezve, függetlenül attól, hogy annak esetleg a 99 %-a még üres. A hátralévő üres helyeket a klaszteren belül, sajnos sehogyan sem tudjuk kihasználni.

Vizsgáljuk meg közelebbről ezt a témát: Ha egy 360 kilobájtos DS DD-s lemezen (DOUBLE SIDE DOUBLE DENSITY) van 354 db klaszter, ahol adatokat tárolhatunk. Ezen a lemezen például 354 db 1024 bájtos, vagy annál rövidebb fájl fér el. A lemezünk akkor lenne kihasználva jó hatásfokkal, ha minden program 1024 bájt hosszú lenne (100 %). Viszont ha rövid (pl.: BATCH fájlokat) - pl. 100 bájtos - programokat tartunk a lemezen, akkor a lemezünk kihasználtsága csak kb 10 %-os lesz ($354 * 100 = 35\ 400$ bájt a rendelkezésre álló 361 472 bájtból). Ennek az az oka, hogy egy fájl egy klasztert elfoglal akkor is, ha csak egy töredékén helyezkedik is el. (Az nagyon sok területet foglalna el, ha minden szektor foglaltságát külön jegyeznénk a lemezen.) Ha ugyanezt a számítást elvégezzük merevlemez esetén is, ott még szomorúbb képet kapunk. A kihasználatlan területeket pedig ebben az esetben fel sem tudjuk használni.

FORMAT

Lemezformázás

FORMAT meghajtó: [/V [:címke]] [/Q] [/U] [/B | /S] és
[/F:méret] vagy [/T:sáv /N: szektor] vagy [/I] [/4] [/8]

Lemezek formázására használjuk.
Külső parancs.

meghajtó: Annak a meghajtónak a betűjele és a kettőspont, amelyiket meg akarjuk formázni.
(Nem tanácsos eltéveszteni.) Megadása kötelező!

/v [:címke] Hatására a sikeres formázás végén címkét ad a lemeznek (VOLUME LABEL), ha nem adjuk meg, akkor pedig bekéri. A lemezcímkére vonatkozó korlátozások a LABEL parancsnál olvashatók. (Ha a /8 paramétert is használjuk, akkor nem adhatunk meg címkét. A /8 -at nem használjuk már, csak a DOS 2.0 előtti verzióinál kellett.)

/q: Már egyszer megformázott lemez újra formázását végzi el gyorsan (QUICK). Törli a foglaltsági táblázatot (FAT) és a gyökérkönyvtárat (ROOT DIRECTORY), a lemezhibákat viszont nem vizsgálja, és nem törli a lemezen lévő fájlokat sem, csak számunkra tűnik úgy, mintha letörölte volna. Ha valamilyen okból a gyors formázás nem végezhető el, akkor azt jelzi, és ha kérjük elvégezze az alacsony szintű formázást (LOW LEVEL FORMAT).

/u: A téves formázás utáni helyrehozatal lehetőségét veti el. (NO UNFORMAT INFORMATIONS) Arra utasítjuk, hogy ne mentse el azt a fájlt, amiből visszaállíthatná a lemez előző tartalmát.

- /b:** A sikeresen megformázott lemezen biztosítja a helyet a rendszerfájlok részére. Erre csak az előző verziókban volt szükség. Megadása esetén a lemezre kerül két fájl rendszerfájlnévvel, de ezek nem rendszerfájlok! A lemezen csak lefoglalja az üres területet.
- /s:** A sikeresen megformázott lemezre másolja fel szükséges rendszerfájlokat (IO.SYS, MSDOS.SYS, COMMAND.COM). A formázás előtt betölti a memóriába az aktuális meghajtó gyökérfájltárában található rendszerfájlokat, és a sikeres formázás után ezeket felmásolja a formázott lemezre. Ha nem talál rendszert az aktuális meghajtón, akkor üzenetet küld. (Lásd a példában!) A lemez ezután rendszerlemez, és erről indíthatjuk is a számítógépünket.
- /f:méret** Konkrétan megadjuk a lemez formázandó méretét bájtokban, ami összhangban kell hogy álljon a lemez típusával. A lemez méretét így abszolút értékben meghatározzuk, tehát a /t /n /1 /4 /8 paramétereket nem használhatjuk.
- /t:sávok** A formázandó lemez felépítését meghatározó egyik paraméter, a sávok száma. A lehetséges értékek közül a táblázat ad útmutatást. A lemez méretének ilyen megadása esetén a /n -t is meg kell adni, viszont nem használhatjuk a /f /1 /4 /8 paramétereket, mert azok egyértelműen meghatározzák a lemez méretét.
- /n:szektorok** A sávonkénti szektorok számát adjuk meg. A választható értékek közül a /f -nél levő táblázat ad segítséget. Ennél a megadási módnál a /t paramétert is használnunk kell, és nem használhatjuk a /f /1 /4 /8 paramétereket.
- /1:** A formázandó dupla sűrűségű lemezt egy oldalásra formázza meg. Az így elérhető lemezkapacitás 160-180 kilobájt oldalanként, attól függően, hogy 8 vagy 9 szektor kerül-e sávonként a lemezre (lásd /8 -nál!). Régebben lehetett kapni olyan lemezt, aminek csak az egyik oldalát lehetett biztonsággal megformázni.
- /4:** A formázandó lemez 5.25" méretű dupla sűrűségű (DS DD), amit 1.2 megabájtos meghajtóban formázunk. A paraméter mellett használható a /1 /8 , de nem használható a /t , /n , /f , mert így is elég jól be van határolva a leendő lemezkapacitás. Nem ajánlom viszont ezt a módszert, mert kompatibilitási problémák léphetnek fel. Az 1.2 MB-os meghajtóban formázott DD-s lemezt gyakran nem olvassa a 360-as meghajtó. Ugyanezt a paramétert kell használnunk akkor is, ha 1.44 MB -os meghajtóban 720 kB -os lemezt formázunk meg.
- /8:** A formázandó lemezre sávonként nem 9, csak 8 szektor kerül fel. Erre az MS-DOS 2.0 előtti verzióival való kompatibilitás miatt lenne szükségünk. Ma már az esetek nagy többségében erre nincs szükség. Az így formázott lemeznek lesz 180 helyett 160 vagy 360 helyett 320 kb kapacitása.

Az előbb ismertetett paramétereket csak értelemszerűen lehet használni. Vannak esetek, amikor kiadhatunk egy lemezre tőle idegen formázási parancsot is, de nem érdemes. Például egy DS DD -s lemez, ami elvileg maximum 360 kilobájtos lehet, megformázható 1.2 megabájtra is. Viszont a lemez fizikai felépítését tekintve nem képes ilyen sűrűségű adattárolásra. Ebből következik, hogy a programjaink pont akkor fognak elveszni, amikor azok a legjobban kellenének, vagy amikor a legnagyobb kárt okozzák. (Ez Murphy törvénye.) Tehát ne törekedjünk sohasem arra, hogy egyszerű módszerekkel kis kapacitású lemezre több adatot vigyünk fel, mint amennyit azon tárolni lehet. Lehetőség van viszont DS DD -s lemezt a rajta lévő felirat szerint 800 kB -ra megformázni. Ehhez szükségünk van például a 800.COM nevű segédprogramra, ami biztosítja, hogy hibamentesen megformázzuk a lemezt nagyobb

kapacitásra, mint 360 kB. Lehet más segédprogramot is használni, de a DOS parancsok nem biztosítanak lehetőséget az egyéb formázásra, annak ellenére, hogy a megadható paraméterek többet sejtetnek.

Ezek után tehát, ha megadjuk a lemez formázandó méretét a /f paraméterrel (pl.: 1.2 Mb-ra: /f:1.2), akkor ne adjuk meg a /1 -el egyoldalasnak, mert ezek egymásnak ellentmondanak. Vagy a /4 paraméterrel megadjuk, hogy a lemez egy 1.2-es meghajtóban formázott 360-as lemez lesz, és beírjuk utána a /n /t paramétereket, amelyek értékei például 1.44-es lemezre vonatkoznak. Semmi értelme, mert vagy a számítógép nem végzi el a parancs végrehajtását, vagy látszólagosan teszi azt meg.

A legáltalánosabb lemeztípusok a következők:

típus	lemez neve	Maximális kapacitása	Lemez méret
SS DD -	Single Side Double Density Egy Oldalas Dupla Sűrűségű	180 kilobájt	5¼"
DS DD -	Double Side Double Density Két Oldalas Dupla Sűrűségű	360 kilobájt	5¼"
DS HD -	Double Side High Density Két Oldalas Nagy Sűrűségű	1200 kilobájt (1.2 megabájt)	5¼"
DS DD -	Double Side Double Density Két Oldalas Dupla Sűrűségű	720 kilobájt	3½"
DS HD -	Double Side High Density Két Oldalas Nagy Sűrűségű	1440 kilobájt (1.44 megabájt)	3½"
DS .. -	Double Side ... Két Oldalas Nagy Kapacitású	2880 kilobájt (2.88 megabájt)	3½"

A megadható paraméterek:

(Egy szektor 512 bájt.)

megadható paraméter (/f)	lemezen összes		l e m e z		meghajtóegység típusa
	sáv (/t)	szektor (/n)	oldal	méret	
160	20	8	1	160 kB	360 kB 5¼"
180	20	9	1	180 kB	
320	40	8	2	320 kB	
360	40	9	2	360 kB	
1.2	80	15	2	1.2 MB	1.2 MB 5¼"
1200				1200 kB	
720	80	9	2	720 kB	720 kB 3½"
1.44	80	18	2	1.44 MB	1.44 MB 3½"
1440				1440 kB	
2.88				2.88 MB	
2880				2880 kB	

Példák:

```

D:\DOS>format a: /f:1.2 /b
a: Insert new diskette for drive A:
and press ENTER when ready...

Checking existing disk format.
Existing format differs from that specified.
This disk cannot be unformatted.
Proceed with Format (Y/N)? y
Formatting 1.2M
_10 percent completed.
Format complete.

Volume label (11 characters, ENTER for none)?_

1213952 bytes total disk space
73728 bytes used by system
1140224 bytes available on disk

512 bytes in each allocation unit.
2227 allocation units available on disk.

Volume Serial Number is 3B3D-11D1

Format another (Y/N)? n

D:\DOS>_

```

Az első hajlékony lemez formázás esete:

"Tegyéél be egy új lemezt az A: meghajtóba
és nyomd meg az **[ENTER]** billentyűt, akkor kész..."

"Vizsgálom a létező lemez formátumokat."

Ha sikerül neki leolvasni a lemez formátumát, akkor tud készíteni majd egy olyan fájlt,
amely segítségével később helyre tudja állítani a lemez eredeti formáját.

"Ez a forma különbözik a létező formáktól."

Nem sikerült, mert nem volt még használva a lemez, vagy más, esetleg számára
ismeretlen formátumban vannak az adatok felírva.

"Ez a lemez nem állítható helyre UNFORMAT-tal."

"Mégis kéred a formázást (Igen/Nem)?"

' y ' (YES) és ' n ' (NO) válaszokkal dönthetünk róla.

"Formátálás folyamatban van 1.2 megabájtra"

"... százalékával készen vagyok." Itt a számolás folyamatosan megy 1-től 99-ig.

"A formázás készen van." - közli mikor végzett.

"Kötetcímke (11 karakter, üss **[ENTER]**-t ha nem kell)?" Ha nem adjuk meg a
v paramétert, akkor a formázás után automatikusan bekéri. Próbaként írjuk be:

12345678901234567890

Ha nem kérünk címkét, akkor csak az **[ENTER]**-t üssük le. Akkor fog a DIR parancs hatására a "has no label." felirat megjelenni.

A tájékoztató adatok kiírása következik most:

"1 213 952 bájt az összes lemez hely"
 " 73 728 bájtot használ ebből a rendszer"
 "1 140 224 bájt az ami rendelkezésre áll a lemezen"
 " 512 bájt minden rendelkezésre álló terület"
 " 2 227 darab rendelkezésre álló terület van a lemezen" (Szektorokban értendőek)
 "A kötet sorozatszámja 3B3D-11D1"
 "Formázást kérsz még (Igen/Nem)?" Ebben az esetben a paraméterek változatlanok minden egyes lemez formázására, még a lemez címkéje is, ha megadjuk előre (/V).

Most nézzük meg egy **DIR** paranccsal mi került rá a lemezünkre.

```
D:\DOS>dir a: /a

Volume in drive A is 12345678901
Volume Serial Number is 3B3D-11D1
Directory of A:\

IO      SYS      35000  93-03-01  10:09
MSDOS   SYS      38000  93-03-01  10:09
      2 file(s)      73000 bytes
      1140224 bytes free

D:\DOS>_
```

A /a paramétert azért kell megadni, mert a rendszerfájlok S, H, R (SYSTEM, HIDDEN, READ ONLY) jelzőbitekkel rendelkeznek, és a normál **DIR** parancs nem listázza ki. Amint láthatjuk a megadott kötetcímkékből csak 11 karakter maradt meg.

A listázott fájlok - IO.SYS, MSDOS.SYS néven bejegyzett állományok - nem rendszerfájlok, erről a lemezről nem indítható a rendszer! Csak a helyet foglalják egy későbbi időpontban felkerülő rendszer számára.

```
D:\DOS>format a: /f:1.44 /b
Insert new diskette for drive A:
and press ENTER when ready...

Parameters not compatible.
Format terminated.

D:\DOS>_
```

A helytelen paramétermegadás egyik esete:

Lemez formázási parancsot adunk ki az A: meghajtóra, ami egy 1.2 -es hajlékony lemezes meghajtó. Méretnek 1.44 megabájtot adunk meg, és a /b paraméter biztosítja a helyet a később felkerülő rendszerfájlok számára.

"Tegyéél be egy új lemezt az A: meghajtóba
és nyomd meg az **[ENTER]** billentyűt, akkor kész..."

"A paraméterek nem megfelelőek. a formázás leállítva."

```

D:\DOS>format a: /q /u /s
Insert new diskette for drive A:
and press ENTER when ready...

Checking existing disk format.
QuickFormatting 1.2M
Format complete.
System transferred

Volume label (11 characters, ENTER for none)? [adatok]
Invalid Volume ID.
Volume label (11 characters, ENTER for none)? adatok

1213952 bytes total disk space
124928 bytes used by system
1089024 bytes available on disk

512 bytes in each allocation unit.
2127 allocation units available on disk.

Volume Serial Number is 155F-11D5

QuickFormat another (Y/N)? n

D:\DOS>_

```

Rendszerlemez készítése gyors formázással:

A gyors formázásnál (/q) a parancs törli a foglaltsági táblázatot (FAT) és a gyökérfájlytárát, de nem bántja a lemezen tárolt adatokat. Ha a /u paramétert nem használnánk, akkor lehetne UNFORMAT paranccsal a lemez eredeti állapotát visszaállítani a formázás után. A parancs végrehajtásakor leolvassa a lemez formátumát, de nem írja fel azt, így helyreállítani nem lehet már. A forma leolvasását azért is végzi el, mert ha a lemeznek más a formátuma, akkor nem lehet rajta gyors formázást végezni. A /s hatására viszi fel a lemezre a rendszerfájlokat.

"Gyors formázás 1.2 megabájtra folyamatban"

"A formázás készen van." - mondja mikor végzett.

"A rendszer átmásolva"

"Kötetcímke (11 karakter, üss [ENTER]-t ha nem kell)?" Ha nem adjuk meg a /v paramétert, akkor a formázás után automatikusan bekéri.

Próbaként írjuk be: [adatok]

Válasz: "Nem megfelelő karakterek a kötetcímkében".

Helyesen: "adatok", és így már elfogadja.

Utána a lemez adatait írja ki a már ismert formátumban. Végül megkérdezi:

"Gyorsformázást kérsz még (Igen/Nem)?"

```

D:\DOS>format a: /v:címke
Insert new diskette for drive A:
and press ENTER when ready...

Checking existing disk format.
Formatting 1.2M
Invalid media or Track 0 bad - disk unusable.
Format terminated.
Format another (Y/N)? n

D:\DOS>_

```

Ez a formázásunk sem fog eseménytelenül lezajlani. Lemezünkről ki fog derülni, hogy nem használható semmire sem a továbbiakban.

A formázás a szokott módon indul 1.2 megabájtra, majd a nulladik sáv formázásakor hiba jelentkezik:

"Érvényelen adathordozó eszköz vagy a nulladik sáv hibás - a lemez használhatatlan."

A formázás megszakad, és ha nem akarunk újra próbálkozni, akkor vissza is kapjuk a készenléti jelet.

```
D:\DOS>format a: /4 /1
Insert new diskette for drive A:
and press ENTER when ready...

Checking existing disk format.
Existing format differs from that specified.
This disk cannot be unformatted.
Proceed with Format (Y/N)? y
Formatting 180K
Format complete.
Unable to write BOOT.
Format terminated.
Format another (Y/N)? n

D:\DOS>vol a:

Volume in drive A is adatok
Volume Serial Number is 2314-11DF

D:\DOS>_
```

Most formázzunk meg egy 360 kilobájtos lemezt 1.2-es meghajtóban, egy oldalásra. Tétélezzük fel, a lemezt már használtuk valamire, és véletlenül az írásvédő nyílása le van ragasztva, így persze nem tud a DOS a lemezre írni.

A kezdeti üzenetek már ismerősek, egy billentyű leütése után indul a formázás. Leolvassa a lemez formátumát, és közli, hogy ez nem egyezik meg a megadott formátummal, így nem használható az UNFORMAT parancs. Megkérdezi, hogy ezek után is kéred-e a formázást. Igen, indulhat a formázás, amit 180 kB-ra el is végez, látszólagosan. **Figyelem!** A lemezünk le van ragasztva, így arra nem lehet írni, tehát leformázni sem lehet. Kicsit későn ugyan, de erre a számítógépünk is rájön:

"Nem tudom írni a betöltő szektort" (BOOT - betölt)

"A formázás leállítva"

"Akarsz még formázni (Igen/Nem)?"

(Ugyanezekkel a paraméterekkel.) Ha nem, akkor kilép a parancsból.

Most nézzük meg VOL paranccsal megmaradt-e a lemez címkéje? Ha igen, akkor a FORMAT parancs nem nyúlt a lemezünkhöz.

```

D:\DOS>format a: /s
Insert new diskette for drive A:
and press ENTER when ready...

Checking existing disk format.
Saving UNFORMAT information.
Drive A error. Could not open the MIRROR image file.
There was an error creating the format recovery file.
This disk cannot be unformatted.
Proceed with Format (Y/N)? y
Verifying 1.2M
 16 percent completed.
Format complete.
System transferred

Volume label (11 characters, ENTER for none)?

 1213952 bytes total disk space
 124928 bytes used by system
4294842368 bytes in bad sectors
 1213952 bytes available on disk

 512 bytes in each allocation unit.
 2371 allocation units available on disk.

Volume Serial Number is 244F-11E4

Format another (Y/N)? n

D:\DOS>_

```

Most ismét egy speciális esetet nézzünk meg. Formázzunk meg egy 360-as (DS DD 5¼"-os) lemezt, egy 1.2 MB-os meghajtóban 1.2 megabájtos méretűre. Már írtam ennek a veszélyeiről, most nézzük meg milyen galibát tud okozni egy helytelen paraméter megadása. Egy billentyű leütésére indul a FORMAT parancs.

"Leolvasom a lemez formátumát"

"Elmentem az UNFORMAT információkat"

"Nem tudom megnyitni a MIRROR fájlt"

"Itt volt egy hiba a formátum helyreállításához szükséges fájl létrehozásánál"

"Ezt a lemezt nem lehet helyreállítani"

"Folytassam formázással (Igen/Nem) ?" Y (YES-Igen)

"Ellenőrzés folyamatban" (formázás!)

Az ellenőrzés végig megy 1..100 százalékig. Miután a formázást befejezte, a /s paraméter hatásra a rendszerfájlokat átmásolja az új lemezre. A /v paramétert nem adtuk meg, így bekéri a címkét, amit most nem adunk meg.

A lemez adatait olvashatjuk a továbbiakban:

A teljes formázható lemez méret

Ahány bájtot a rendszerfájlok elfoglalnak

A hibás szektorok száma (úgy 429 millió körül). Egy kicsit fucsa értéket kaptunk, nem? Ennyi szektor több doboz lemezen sincsen. Ugyanis a program arra nem képes, hogy a lemez formázható méretét leolvassa. Hogy egy adat azért nem olvasható vissza, mert a lemez formázható kapacitása jóval kisebb a megadott értékeknél, vagy egyszerűen csak gyenge minőségű, az nem dönthető el a programból (jelenleg.)

SYS**Új rendszerállományok felmásolása**

SYS [*meghajtó_1*] [*elérési_út*] *meghajtó_2*:

A rendszer fájlokat viszi fel az adott lemezre.

Külső parancs.

meghajtó_1: Annak a meghajtónak a betűjele és a kettőspont, amelyikről az új rendszert kívánjuk felvinni. Ha elmarad, akkor az aktuális meghajtón keresi a rendszerfájlokat.

elérési_út: A felvinni kívánt rendszerfájl könyvtárának a neve és annak elérési útja. Ha elmarad, akkor az aktuális meghajtó gyökérkönyvtárában keresi.

meghajtó_2: Annak a meghajtónak a betűjele és a kettőspont, amelyekre a rendszerállományokat fel akarjuk másolni. A fájlokat a DOS a gyökérkönyvtárba teszi. Megadása kötelező, és nem keverendő össze a két meghajtómegadás!

Több célból is használhatjuk a parancsot:

1. Ha egy lemezt nem a /s paraméterrel formáztunk meg valamilyen okból. Ha egy adatokat tároló lemezt kinevezünk rendszerlemeznek.
2. Ha a rendszerlemezen lévő rendszerfájlok megsérültek.
3. Ha vírusosak voltak a rendszer állományok, akkor célszerű mindenképpen felmásolni egy újat, még ha a vírusirtó program azt is mondta, hogy sikeresen kiirtotta a vírust. Ugyanis nem tudhatjuk, hogy nem sérült-e meg valamilyik, előzőleg fertőzött állományunk.

Példa:

```
C:\>sys c: a:
System transferred

C:\>sys a: d:
Insert system disk in drive A:\
Press any key to continue . . .
^C

C:\>copy *.txt a:\
SETUP.TXT
BOOTLOG.TXT
LOGFILE.TXT
HPSETLOG.TXT
      4 file(s) copied

C:\>dir a:

Volume in drive A is DISK_144-ES
Volume Serial Number is 345F-16DB
Directory of A:\

BOOTLOG   TXT           1243  93-11-12   16:48
HPSETLOG  TXT           2218  93-11-13   11:36
LOGFILE   TXT          89530  93-11-14    0:08
SETUP     TXT          41724  92-03-10    3:10
      4 file(s)           134715 bytes
                               1321984 bytes free

C:\>sys c: a:
System transferred

C:\>_
```


Az első példa, egy szabályos parancshasználatot mutat be. A merevlemez (C:) gyökérkönyvtárában található rendszerállományokat másoljuk ki az első hajlékonylemez egység (A:) gyökérkönyvtárába.

A második példában a helyzet fordított, de a parancs lefutása is különbözik az előzőtől. Ugyanis ezen a lemezen (A:) nincs DOS, így felszólít bennünket a számítógép, hogy:

"Helyezd be a rendszerlemezt az A: meghajtó gyökérkönyvtárába
Nyomj meg egy tetszőleges billentyűt a folytatáshoz..."

Innen most egy **[CTRL] + [C]** -vel lépünk ki, mert éppen nincs kéznél rendszerlemezünk.

Most másoljunk fel néhány állományt a lemezre, a következő példa bemutatásához. Példánkban egy 1.44 MB -os lemezt használtunk. A **DIR** parancs által szolgáltatott adatokból egyértelműen kiderül, hogy nincs rendszer a lemezen, még rejtett állapotban sem.

Utána használjuk a **SYS** parancsot, és ezzel bizonyítást nyert, hogy a rendszerállományok utólag is fémásolhatók a lemezre. Így a **FORMAT** parancsnál a /b paraméternek semmi haszna sincs.

detonáló lemez COMMAND.COM-ot is

DISKCOPY

Lemezmásolás

DISKCOPY [meghajtó_1: [meghajtó_2:]] [/I] [/V]

Hajlékony lemezek komplett másolását végzi el.

Külső parancs.

meghajtó_1: A másolandó lemez meghajtójának a betűjele.

meghajtó_2: Ebben a lemezegységben lesz a másolat lemez.

/I: Csak az első oldalát másolja a lemezeknek.

/V: A másolásakor ellenőrzi a másolatot.

A másolás formátumát a másolandó lemez határozza meg. Ha szükséges, akkor a másolatlemez előbb megformázza. A másolatlemez csak vele egyező formátumú lehet. (1.2-es lemezt ne próbáljunk 360-asra lemásolni, mert nem fog sikerülni!)

A parancsot csak hajlékony lemezekre (FLOPPY DISK) lehet használni.

A parancs a másolást szektoronként végzi el, és annyit olvas be a memóriába, amennyi szabad helyet fel tud használni. Ezért az 1.2 megabájtos lemezeket csak több menetben tudja lemásolni.

A *meghajtó_1* és *meghajtó_2* lehet azonos is, hacsak egy meghajtó van a számítógépünkben. Ekkor a lemezeket cserélni kell időnként, amire folyamatosan felhívja a figyelmünket.

A *meghajtó_2* paramétert nem kötelező megadni, ha elmarad, akkor az aktuális meghajtó lesz az.

Ha a *meghajtó_1* paramétert sem adjuk meg, akkor mind a kettő az aktuális lesz. Ebből következik, hogy paraméter nélkül csak akkor adható ki a parancs, ha az aktuális meghajtó az A: vagy a B: lemezegység.

Példák:

```
D:\TEMP>diskcopy a: a:

Insert SOURCE diskette in drive A:
Press any key to continue . . .
Copying 80 tracks
15 sectors per track, 2 side(s)

Insert TARGET diskette in drive A:
Press any key to continue . . .
:
:
:
Volume Serial Number is 10D9-1B2A

Copy another diskette (Y/N)?_

D:\TEMP>_
```

Másoljuk le egy lemez tartalmát egy másikra, egy olyan számítógépen, ahol csak egy hajlékony lemezes meghajtó van:

"Helyezd be az A: meghajtóba a forráslemezt"
 "Nyomj egy tetszőleges billentyűt a folytatáshoz..."
 "Másolom 80 sáv
 15 szektor/sáv, 2 oldal"
 "Helyezd be az A: meghajtóba a cél lemezt"
 "Nyomj egy tetszőleges billentyűt a folytatáshoz..."

A továbbiakban ez a művelet sor addig ismétlődik, ameddig a lemez tartalmát át nem másolta. Annyi adatot tud egyszerre beolvasni, amennyi a rendelkezésére álló szabad memória. Általában egy 360 -as lemezt egy menetben átmásol, egy 1.2 -es lemezt pedig három menetben.

Ha a parancsot nem hajlékony lemezes egységre adjuk ki, hanem például merevlemezre, akkor:

```
D:\TEMP>diskcopy d: d:

Invalid drive specification
Specified drive does not exist
or is non-removable

D:\TEMP>_
```

"Nem megfelelő meghajtómegadás
 A megadott meghajtó nem létezik,
 vagy nem cserélhető a lemez."

Műveletek fájlokkal

COPY

Fájlok másolása

COPY [/A/B] *forrás* [+ [/A/B] *forrás*] [+...] [*cél* [/A/B] [/V]]

COPY *forrás* [/A/B] [+ *forrás* [/A/B]] [+...] [*cél* [/A/B] [/V]]

Fájlok sokféle másolását végzi el.

Belső parancs.

forrás: [*meghajtó*:] [*elérési_út*] *fájlnev*

meghajtó: Annak meghajtónak a betűjele és a kettőspont, amelyiken a másolandó fájl(ok) van(nak). Ha nem adjuk meg, akkor szokás szerint az aktuális meghajtón keresi azokat.

elérési_út: Annak könyvtárnak a neve az elérési úttal, ahol a másolandó fájl(ok) van(nak). Ha nem adjuk meg, akkor az aktuális könyvtárban fogja keresni a másolandó fájlokat.

fájlnev: A másolandó fájl vagy fájlok köre. Helyettesítő karaktereket is lehet használni. Ezt a paramétert minden esetben meg kell adni. Gondoljunk csak bele: ha nem adjuk meg, hogy mit akarunk valahová másolni, a számítógép honnan találja ki? Enélkül semmi értelme sincs a parancsnak.

Ne tévesszen meg senkit sem például az, hogy a `C:\>copy . a:↵` (copy, szóköz, pont, szóköz, a, kettőspont, enter) parancs az összes állományt másolja a C: meghajtó gyökérkönyvtárából, az A: meghajtó aktuális könyvtárába. Látszólag nem adtuk meg a másolandó fájlok körét, mégis mindent másol. Ugyanis a pont karakter magában azonos a .* megadással, ami nem más, mint *fájlnev* megadás.

cél: [*meghajtó*:] [*elérési_út*] [*fájlnev*]

meghajtó: Annak meghajtónak a betűjele és a kettőspont, amelyikre a másolandó fájl(ok) kerül(nek). Ha nem adjuk meg, akkor az aktuális meghajtóra kerülnek.

elérési_út: Annak a könyvtárnak a neve az elérési úttal, ahová a másolandó fájl(ok) kerül(nek). Ha nem adjuk meg, akkor az aktuális könyvtárba fog kerülni.

fájlnev: A másolandó fájl vagy fájlok régi vagy új neve. Helyettesítő karaktereket is lehet használni. Ezt a paramétert sem kell mindig megadni. Ha nem adjuk meg, akkor az aktuális forrásfájl neve lesz az új fájl neve.

/a: Az állományt, (ASCII) szöveges állományként kezeli, és a fájl végét a **[CTRL] + [Z]** (ALT 26) kód fogja jelezni. Ha a */a* paraméter a fájlnevek előtt van, akkor minden fájlra vonatkozik addig, amíg nem talál egy */b* -t. Ha a */a* paraméter a fájlnevek után áll, akkor annak hatása az előtte közvetlenül álló és minden következő olyan fájlra érvényes, ami után nem áll a */b*. Tehát a */a* után álló fájlok közül az utolsóra, amelyik után már a */b* áll, már a */b* hatása vonatkozik.

Ez egy kicsit bonyolultnak tűnhet, de majd példákon keresztül jobban meg lehet érteni. Hasonló a **PRINT** parancsnál megadható */p* */c* paraméterekhez.

/b: Az állományt, bináris fájlként kezeli, így például a **[CTRL] + [Z]** nem jelenti a fájl végét. A fájlt tényleges mérete és adatai alapján kezeli, és nem próbál vezérlő kódokat keresni benne. A hatása hasonlóan érvényes a /a paraméternél leírtakkal. Ez az alapértelmezésbeli másolási módszer.

/v: Hatására a másolás helyességét fogja ellenőrizni. (VERIFY) A kiírt adatokat szektoronként visszaolvassa, és ha nem egyezik, akkor újra és újra próbálja sokszor, és ha továbbra sem sikerül, akkor az a szektor hibás, már nem használható (kopott mágnesezésű?), bejelöli hibásnak a DOS, és egy másik szektorba próbál írni. Sokat használt régi meghajtóknál célszerű használni, de nem kötelező.

A parancsban legalább kettő paramétert meg kell adni, de a másolandó fájl nevét azt mindig. Ha csak a másolandó fájl nevét adjuk meg, akkor a cél fájl neve és helye is ugyanaz lesz. Ebben az esetben arra utasítjuk, hogy másolja a fájlt saját magára (amit nem enged). De ennek mi az értelme? Semmi. A másolandó állomány nem lehet nulla hosszúságú. Ilyen eset akkor állhat elő, amikor írunk egy állományt, de valamilyen hiba miatt nem zárjuk le azt. Ugyanis a lezárás után tudja csak az állomány hosszát felírni. Ha ez elmarad, akkor a fájl-foglaltsági táblázatba (FAT) sem kerül bejegyzésre, így az állományunkat gyakorlatilag elveszettnek nyilváníthatjuk. Viszont a fájl nevének bejegyzése megmarad nulla hosszal. Ebből következik, hogy a fájlunk nem is fájl, csak egy bejegyzés a könyvtárban, így másolni sincs mit.

Példák:

```
F:\SZOVEG>copy f:\szoveg\xcopy.txt f:\subdir\xcopy.txt
1 file(s) copied

F:\SZOVEG>copy xcopy.txt subdir
1 file(s) copied

F:\SZOVEG>copy xcopy.txt NOFILE
1 file(s) copied

F:\SZOVEG>_
```

A **COPY** parancs szabályos használata: Minden paramétert megadtunk, pontosan meghatároztuk, hogy mit akarunk másolni, honnan és hová, és mi legyen az új másolatnak a neve. Ha így használjuk a parancsot, akkor biztos mindig az fog történni, amit szeretnénk. Persze ez elég hosszadalmas eljárás, és már némi kis gyakorlattal a hat megadott paraméterből négyet elhagyhatunk nyugodtan, akkor is jól fog működni a parancs. A parancs megadásakor azok a paraméterek maradhatnak el, amelyeket az "aktuálissal" lehet helyettesíteni. Az "aktuális meghajtó" a forrásban és a célon is szerepelhet. Az "aktuális könyvtár" csak az egyik helyen lehet (hacsak nem akarjuk megduplázni a fájlunkat), és ha a célon lévő fájlnevet sem adjuk meg, akkor az a forrásfájl nevével lesz azonos. Ha összeszámoljuk, akkor ez négy elhagyható paraméter.

A második példában pont nem azt a kettő paramétert adtuk meg, amelyiket kellett volna. A 'SUBDIR' könyvtár nem a 'SZOVEG' könyvtárnak az alkönyvtára, így ezt a nevet a parancs fájlnevének fogja tekinteni.

A parancsban megadott 'NOFILE' viszont alkönyvtára a 'SZOVEG' -nek, így a fájlunkat át fogja másolni oda.

```
F:\SZOVEG>copy xcopy.txt \subdir
1 file(s) copied

F:\SZOVEG>_
```

A második példában megadottak így néznek ki helyesen: Itt három paramétert kellett megadni, a fájl nevét, és a célkönyvtár nevét (SUBDIR) és elérési útját (\).

```
F:\>copy SUBDIR\xcopy.txt
1 file(s) copied

F:\>attrib xcopy.txt +h

F:\>copy SUBDIR\xcopy.txt
Access denied - F:\XCOPY.TXT
file(s) copied

F:\>_
```

Az aktuális könyvtár most a gyökérkönyvtár. Most ide másoljuk az 'XCOPY.TXT' fájlt. Most is kettő paramétert adtunk meg, a könyvtár nevét (az aktuális könyvtárban található 'SUBDIR' alkönyvtár) és a fájl nevét.

Az ATTRIB paranccsal rejtjük el az imént bemásolt fájlunkat. Most ismételjük meg az ablakunk első parancsát. Normál esetben a fájlunkat szó nélkül felülírná a mostani parancs. Ha viszont a "H" jelzőbitje be van állítva, akkor nem engedi felülírni, és ezért mondja azt, hogy a "Hozzáférhetőség lehetetlen".

A DIR paranccsal nézzük meg mi van a könyvtárunkban. Viszont ha a '/a' paramétert nem használjuk, akkor nem fogjuk látni a keresett fájlt, mert rejtett fájl.

```
F:\SUBDIR>dir \ /a

Volume in drive F is DISK_CIMKE~
Volume Serial Number is 256D-0EF4
Directory of F:\

SUBDIR    <DIR>    93-03-03    9:41
SZOVEG    <DIR>    93-03-03    9:44
.         <DIR>    93-01-21   15:21
..        <DIR>    93-01-21   15:21
XCOPY TXT  8843 93-03-03    9:31
          5 file(s) 8843 bytes
          1804288 bytes free

F:\SUBDIR>_
```

Váltsunk könyvtárat, és lépjük be a 'SUBDIR' -be. Miután láttuk, hogy a fájlunk a gyökérkönyvtárban megvan, másoljuk be most ide. Az eddigi gyakorlatnak megfelelően azok a paraméterek elmaradhatnak, amelyek az "aktuálissal" helyettesíthetők. A mostani példában: aktuális meghajtó gyökérkönyvtárában (\) található 'XCOPY.TXT' fájlt másoljuk. A célból semmit sem adtunk meg, így az az aktuális meghajtó, aktuális könyvtárába kerül, ugyanazzal a névvel mint a forrás volt.

```
F:\SUBDIR>copy \xcopy.txt
File not found - F:\XCOPY.TXT
0 file(s) copied

F:\SUBDIR>_
```

A következő példa előtt nézzük meg az aktuális könyvtárban található 'TXT' kiterjesztésű fájlokat, azok hosszát.

```
F:\SZOVEG>dir *.txt

Volume in drive F is DISK_CIMKE~
Volume Serial Number is 256D-0EF4
Directory of F:\SZOVEG

DISKCOPY TXT      432 93-03-02  12:41
TEXT      TXT      140 93-03-02  10:35
UNDELETE  TXT     5213 93-03-02  10:13
XCOPY     TXT     8843 93-03-03   9:31
          4 file(s)  14628 bytes
                   1802240 bytes free

F:\SZOVEG>_
```

Most példát láthatunk arra, hogy hogyan lehet fájlokat összefűzni a '+' jellel és egy új név alatt elmenteni. (Ha a TYPE paranccsal megnézzük az eredményt, akkor láthatjuk, hogy egy olyan fájl keletkezett, amiben a felsorolt fájlok tartalma szerepel.)

```
F:\SZOVEG>copy diskcopy.txt +text.txt +undelete.txt +xcopy.txt ujfile. $$$

DISKCOPY.TXT
TEXT.TXT
UNDELETE.TXT
XCOPY.TXT
1 file(s) copied

F:\SZOVEG>_
```

Az új fájl adatait a DIR paranccsal nézhetjük meg. A hossza pedig az összefűzött fájlok hosszának összege, plusz egy bájt fájlvégjel (END OF FILE = EOF)

```
F:\SZOVEG>dir ujfile. $$$

Volume in drive F is DISK_CIMKE~
Volume Serial Number is 256D-0EF4
Directory of F:\SZOVEG

UJFILE $$$  14629 93-03-26  8:27
          1 file(s)  14629 bytes
                   1785856 bytes free

F:\SZOVEG>_
```

A **COPY** parancs egyik speciális formája, hogy a *forrás* a billentyűzet, a *cél* pedig egy fájl. Rövidebb batch fájlokat így is írhatunk, nem kell bonyolult szövegszerkesztőt néhány sor miatt betölteni. Az aktuális sorban tudjuk még javítani a beírt szöveget, de ha az **[ENTER]**-t megnyomtuk, akkor már nem tudunk visszalépni az előző sorba. A fájl végén a **[CTRL] + [Z]** billentyűt vagy az **[F6]**-ot nyomjuk meg, majd az **[ENTER]** hatására a fájlt lezárja, lemezre írja a billentyűzetpuffer tartalmát, és kiírja a DOS készenléti jelet.

```
F:\SZOVEG>copy con levél.txt
A billentyuzetrol geperek egy levelet
bele egy faljba, ekezetek nelkul.
De írhatnám ékezetekkel is, csak
lassú így az ALT -al előcsalni őket egyesével.
A fájl vége CTRL + Z lesz:
^Z
      1 file(s) copied

F:\SZOVEG>_
```

A **'LEVÉL.TXT'** fájlunk adatait a **DIR** parancs mutatja meg. A **DIR** a fájlnevet nagybetűkkel írja ki. (Az 'é' betű az **[ALT] + [1] [3] [0]** billentyűkre jön elő. Lenyomjuk az **[ALT]**-ot, a jobb oldali számbillentyűzetten beírjuk a **[1] [3] [0]** számokat, és csak most engedjük fel az **[ALT]**-ot. A fájl nevében használt ékezetes betűknek csak egy nagy hátránya van: lassú mindig az **[ALT]**-on keresztül egyesével beírni a betűket.

```
F:\SZOVEG>dir levél.txt

Volume in drive F is DISK_CIMKE~
Volume Serial Number is 256D-0FE4
Directory of F:\SZOVEG

LEVÉL  TXT      184 93-03-26  8:36
      1 file(s)    184 bytes
                1781760 bytes free

F:\SZOVEG>_
```

Az **[ALT]**-os billentyűzésnek a hátránya még, hogy ha nincs még kellő gyakorlatunk benne, akkor lehet hogy előbb engedjük fel az **[ALT]**-ot, és így más kódot viszünk be mint szeretnénk. Példánkban az "ALT-130" -ből így lett "ALT-13", ami nem más mint az **[ENTER]**. Ezért kaptunk hibaüzenetet, mert csak félig írtuk be azt a parancsot, amit akartunk.

```
F:\SZOVEG>copy lev
File not found - LEV
      0 file(s) copied

F:\SZOVEG>_
```

```
F:\SZOVEG>copy levél.txt con
A billentyuzetrol gepenek egy levelet
bele egy faljba, ekezetek nelkul.
De írhatnám ékezetekkel is, csak
lassú így az ALT -al előcsalni őket egyesével.
A fájl vége CTRL + Z lesz:
      1 file(s) copied

F:\SZOVEG>_
```

A **COPY** parancs másik speciális formája, hogy a fájlt másoljuk ki a konzolra, vagyis a képernyőre. A parancs ebben a formájában hasonlít a **TYPE** parancsra. Annyi a különbség, hogy a végén kiírja az **"1 file(s) copied"** szöveget.

A következő példa már pontos mása a **TYPE** parancsnak. A fájl tartalmát a képernyőre, mint külső egységre másoljuk, és a parancs üzeneteit a **NUL** egységre irányítjuk, azaz a semmibe. Hogy érthetőbb legyen, írjuk be példaként ugyanezt a parancsot, de a **NUL** helyett írjunk **PRN** -t. A fájl tartalma a képernyőre másolódik, míg a **COPY** parancs üzenete(i) a nyomtatón fognak megjelenni.

```
F:\SZOVEG>copy levél.txt con >nul
A billentyuzetrol gepenek egy levelet
bele egy faljba, ekezetek nelkul.
De írhatnám ékezetekkel is, csak
lassú így az ALT -al előcsalni őket egyesével.
A fájl vége CTRL + Z lesz:

F:\SZOVEG>_
```

A billentyűzetet és a nyomtatót így lehet összekötni egy parancs segítségével. Másolunk a billentyűzetről a nyomtatóra. Egy sor bevitele után fogja kiírni a beírt szöveget a nyomtatóra. Akkor hasznos ez a módszer, ha csak egy-két szót kell kiíratni egy nyomtatványra számítógéppel. Ilyenkor hosszadalmas lenne egy szövegszerkesztőt behívni.

```
F:\SZOVEG>copy con prn
Címzett: Tanuló diákság
Cím: Győr
Tárgy Számítástechnika oktatás
^Z
1 file(s) copied

F:\SZOVEG>_
```

```
F:\SZOVEG>copy con prn
^Z
      0 file(s) copied

F:\SZOVEG>_
```

A fenti példánkban nem írtunk be egyetlen karaktert sem, csak az EOF -ot, így nincs is mit másolni. Ezért írta ki, hogy: **"0 file(s) copied"**.

ATTRIB

.Jelzőbit állítás

ATTRIB [+S|-S] [+H|-H] [+R|-R] [+A|-A]
 [*meghajtó*:] [*elérési_út*] *fájlnev*] [/S]

A fájlok és könyvtárak jelzőbitjeit lehet állítani vele.
 Külső parancs.

+S|-S, +H|-H, +R|-R, +A|-A

A jelzőbiteket (ATTRIBUTUM) állíthatjuk (+), illetve törölhetjük (-), paraméter nélkül pedig listázhatjuk. A használható jelzőbitek:

- H - Rejtett (HIDDEN)
- S - Rendszer (SYSTEM)
- R - Csak olvasható (READ ONLY)
- A - Elmentendő fájl (ARCHIVE)

meghajtó: *elérési_út fájlnev*

Az fájl, vagy azok a fájlok, amelyeknek a jelzőbitjeit állítani szeretnénk. Ha a *meghajtót* és/vagy az *elérési_út* paramétereket elhagyjuk, akkor az aktuálisat feltételezi a számítógép. Használhatunk helyettesítő karaktereket is (* ?).

/s: A paraméter hatására a megadott könyvtár alkönyvtáraiban (SUBDIRECTORY) is végrehajtja a parancsot a számítógép.

Ha a "H" jelzőbitet használjuk, akkor az alapértelmezett DIR parancs nem listázza ki a fájlt. Az "S" jelzőbit, a rendszerfájlok jelzőbitje, nincs különösebb jelentősége. Az "R" jelzőbit használata esetén a DEL paranccsal nem tudjuk törölni a programot. Ahhoz, hogy letörölhessünk egy ilyen fájlt, előbb az ATTRIB paranccsal törölni kell az "R" jelzőbitjét, és csak utána tudjuk törölni a fájlt. Nem az illetéktelenek ellen véd, hanem a véletlen törlés ellen. Bár az 5.00-ás verziójú DOS -ban nem sok jelentősége van, mert a fájlok az UNDELETE paranccsal visszahozhatók.

Az "A" jelzőbitet minden fájl megkapja akkor, amikor létrehozzuk, és archiválás után akkor, amikor módosítjuk. Az előző DOS verziókban volt jelentősége, amikor még a fájlokat csak a BACKUP - RESTORE és az XCOPY paranccsal lehetett elmenteni (archiválni). Ma már jobbnál jobb programok segítenek ebben, az adatok tömörítésével, az archív állományok könnyebb, komfortosabb kezelésével.

```
D:\TEXT>attrib UTIL -r
Not resetting hidden file D:\TEXT\UTIL

D:\TEXT>attrib UTIL
H   D:\TEXT\UTIL

D:\TEXT>attrib UTIL -h

D:\TEXT>
```

Töröljük a UTIL könyvtár 'R' jelzőbitjét. Csak az a hiba, hogy nincs beállítva, így nincs mit törölni. A hibaüzenetet pedig azért kaptuk, mert rejtett állománynak nem állíthatjuk a jelzőbitjeit, csak akkor, ha töröltük a 'H' jelzőbitet.

Nézzük meg a UTIL nevű könyvtár jelzőbitjeit. Mint láthatjuk a 'H' (rejtett) van beállítva.

Töröltük a UTIL könyvtár 'H' jelzőbitjét.

```
D:\TEMP>attrib *.*?x
A      D:\TEMP\MODE.XXX
A      D:\TEMP\FIND.TXX

D:\TEMP>attrib *.*?x +rhsa
Parameter format not correct - +rhsa

D:\TEMP>attrib *.*?x +r h s a
Too many parameters - s

D:\TEMP>attrib *.*?x +r h

D:\TEMP>attrib *.*?x
A R    D:\TEMP\MODE.XXX
A R    D:\TEMP\FIND.TXX

D:\TEMP>_
```

```
D:\TEMP>attrib *.*?x -r +h +s -a

D:\TEMP>attrib *.*?x
SH     D:\TEMP\MODE.XXX
SH     D:\TEMP\FIND.TXX

D:\TEMP>_

D:\TEMP>attrib *.*?x +a /s
Not resetting hidden file D:\TEMP\MODE.XXX
Not resetting hidden file D:\TEMP\FIND.TXX

D:\TEMP>
```

A további parancsaink minden olyan fájlra vonatkoznak, melynek a kiterjesztése három karakter, és 'x' betűre végződik (*.*?x).

Most megnézzük ezeknek a fájloknak a beállított jelzőbitjeit. Az 'A' jelzőbitje van mindkét fájlra beállítva.

Utána a helytelen paramétermegadásra láthatunk egy példát:

A paraméter formája nem megfelelő - +rhsa"

A paramétereket egyesével kell megadni!

Majd megint elkövettünk egy hibát a parancs megadásakor: nem adtuk meg az egyes jelzőbiteknél azt, hogy törölni vagy beállítani akarjuk az adott bitet.

"Túl sok paraméter - s"

Az üzenet kicsit megtévesztő, mert nem csak az 'S' jelzőbit a sok - illetve hiányos - , hanem a 'HSA' jelzőbitek.

Az előző példát bizonyítjuk most be egy példán keresztül: A parancs az 'R' jelzőbitet be fogja állítani, de a 'H' -t nem.

Utána megnéztük meg melyik jelzőbit változott.

```
D:\TEMP>attrib *.*?x -h -s
D:\TEMP>attrib *.*?x +a /s
D:\TEMP>attrib *.*?x +h +s
D:\TEMP>_
```

Az ATTRIB parancs helyes használatára láthatunk egy példát.

Nézzük meg az eredményt.

Töröljük a megadott fájlok 'A' jelzőbitjét, és az alkönyvtárakban is végezzük el a parancsot. A kapott hibaüzenet arra vonatkozik, hogy a megadott fájlok rejtett - 'H' jelzőbittel rendelkező - fájlok, ezért nem állíthatjuk a többi jelzőbitjét.

Töröljük a megadott fájlok 'H' és 'S' jelzőbitjét.

Most elvégezhetjük az archív jelzőbit beállítását.

A művelet elvégzése után célszerű a többi jelzőbitet az eredeti formájába visszaállítani. Az előző és a mostani parancsot kiadhattuk volna egyszerre is.

XCOPY

Csoportos másolás

XCOPY *forrás* [*cél*] [/A | /M] [/D:*dátum*] [[/S] [/E]] [/V] [/P] [/W]

Fájlok csoportos másolását végzi el.

Külső parancs.

forrás: [*meghajtó*:] [*elérési_út*] [*fájlnév*]

meghajtó: Annak a meghajtónak a betűjele és a kettőspont, amelyiken a másolandó fájl(ok) van(nak).

elérési_út: Annak a könyvtárnak a neve az elérési úttal, ahol a másolandó fájl(ok) van(nak).

fájlnév: A másolandó fájl vagy fájlok köre. Helyettesítő karaktereket is lehet használni. Ha nem adjuk meg, akkor a szokásostól eltérően minden fájlt átmásol (*.*) .

cél: [*meghajtó*:] [*elérési_út*] [*fájlnév*]

meghajtó: Annak a meghajtónak a betűjele és a kettőspont, amelyekre a másolandó fájl(ok) kerül(nek).

elérési_út: Annak a könyvtárnak a neve az elérési úttal, ahová a másolandó fájl(ok) kerül(nek).

fájlnév: A másolandó fájl vagy fájlok régi vagy új neve. Helyettesítő karaktereket is lehet használni. Ezt a paramétert sem kell mindig megadni. Ha nem adjuk meg akkor az aktuális forrásfájl neve lesz az új fájl neve is.

A parancs logikája a **COPY** parancshoz hasonlóan:

"Másolás honnan mit hová milyen_névre."
XCOPY d:\..\..\ fájlnév d:\..\..\ fájlnév

A parancsban legalább kettő paramétert meg kell adni a hat megadható közül, értelemszerűen. Gondoljuk ki magyarul mit is szeretnénk csinálni, és csak le kell fordítani angolra. A többi megadható paraméterrel tudjuk "komfortosabbá" tenni a parancsot.

/a vagy **/m**

Csak azokat az állományokat másolja, amelyeknek az 'A' - mint archív jelzőbitje be van állítva. A két paraméter azt dönti el, hogy a másolás során a fájlok archív bitjével mi történjen:

/a: Nem változtatja meg. Akkor szoktuk használni, ha több példányban archíváljuk az állományokat. (AVAILABLE - meglévő)

/m: Törli az archív bitet. Akkor használjuk, ha másolással archíváljuk a fájlokat, jelezve, hogy már el van mentve biztonságos közegre az állomány. (MODIFY - módosítás)

/d:dátum

A másolás csak a megadott dátum után létrejött, vagy módosított állományokat másolja át. Ennek a paraméternek a használatával naprakészen tarthatjuk a biztonsági állományainkat (DATE - dátum).

*** /s:** Az alkönyvtárakból (SUBDIRECTORY) is másolja a fájlokat. Egyébként csak a megadott könyvtárból másolna.

*** /e:** Csak akkor használhatjuk a paramétert, ha a **/s** -t is használjuk. A paraméter hatására az olyan alkönyvtárakat is átmásolja, amelyek üresek (EMPTY). Egyébként nem másolná át. Vannak olyan esetek, amikor egy program például az ideiglenes állományait egy alkönyvtárban hozza létre, de a program lefutása után törli azokat. Viszont a munkája során szüksége van egy ilyen - általában üres - alkönyvtárra is. Ha a programot az üres alkönyvtárak nélkül másolnánk át, akkor a másolat nem működne helyesen.

/v: A másolás helyességét ellenőrzi (VERIFY) a parancs a paraméter hatására. Ugyanis ha sérült, piszkos, sokat használt lemezre írjuk ki az adatainkat, akkor azokat nem biztos, hogy vissza tudjuk még valaha is olvastatni. A **/v** hatására viszont a kiírt adatokat szektoronként visszaolvassa, és ha nem egyezik, akkor többször megpróbálja, és ha továbbra sem sikerül, akkor azt a szektort bejelöli hibásnak a DOS, és egy másik szektorba próbál írni. Sokat használt meghajtóknál célszerű használni, de nem kötelező.

/p: A (PROMPT - kiértésk) paraméter hatására minden másolandó fájl neve mellé kiírja a "(Y/N)?" -t, és egyesével bekéri azt, hogy szeretnénk-e másolni vagy sem. Az 'Y' (YES-igen) és az 'N' (NO-nem) válaszokat adhatjuk. Akkor szoktuk használni, ha több fájlt másolunk (például: *.* azaz minden fájlt), lehet a fájlok között olyan is, amit mégsem akarunk átmásolni.

/w: A paraméter hatására a parancs kiadása után kapunk egy üzenetet, és egy billentyű leütésére indul a másolás (WAIT - várakozás). A paraméternek gyakorlatilag nem sok értelme van, mert addig, amíg nem készítettük elő a lemezeket a másoláshoz, addig ne üssük le az **[ENTER]** billentyűt, azaz ne indítsuk el a parancsot.

Másolás közben folyamatosan tájékoztatást kapunk a másolás alakulásáról. Először beolvassa a számítógép memóriájába a forrásfájlokat, majd kiírja azokat a célmeghajtóra. Beolvasáskor a "**Reading file(s) ...**" üzenetet kapjuk, íráskor pedig a kiírt fájl neve jelenik meg. Egyszerre annyi fájlt tud beolvasni, amennyit a rendelkezésre álló memória megenged. A másolás végén kiírja a számítógép a másolt fájlok számát.

Ha nem másolunk egy fájlt sem, - például **/p** paraméter adta lehetőségre minden kérdésre nemleges választ adunk, akkor a következő üzenetet kapjuk:

```
FILE NOT FOUND - ??????????.???
0 FILE(S) COPIED
```

Példák:

A következőkben ismertetésre kerül egy komplex példasor, amelyben darabonként átmásoljuk az összes alkönyvtárat és a fájlokat. Minden parancs másképpen fog kinézni, hogy felismerhető legyen bennük a különbség, és az egyes parancsok funkciója, a paraméterek hatása a parancsra. Figyeljük meg a fájlok dátumát és jelzőbitjeit is. Bemutatom a kiindulási könyvtárak tartalmát és azt, hogy a másolások után mi lesz belőle. A fájlok jelzőbitjeit is időnként tájékoztatásul kiíratjuk az **ATTRIB** parancssal.

A kiindulási állapot:

```
F:\>dir /s
Volume in drive F is DISK_CIMKE~
Volume Serial Number is 256D-0EF4

Directory of F:\

SUBDIR    <DIR>      93-03-03    9:41
SZOVEG    <DIR>      93-03-03    9:44
.         <DIR>      93-01-21   15:21
..        <DIR>      93-01-21   15:21
4 file(s)    0 bytes

Directory of F:\SUBDIR

NOFILE    <DIR>      93-03-03   10:08
USERKE    <DIR>      93-03-03   10:08
VANBENNE  <DIR>      93-03-03   10:08
.         <DIR>      93-03-03    9:41
..        <DIR>      93-03-03    9:41
DISKCOPY  TXT        432 93-03-02   12:41
MEM       TXT        4779 93-03-02   12:14
PRINT    TXT         205 93-03-03    5:58
TEXT     TXT         140 93-03-02   10:35
UNDELETE TXT        5213 93-03-02   10:13
XCOPY    TXT        8843 93-03-03    9:31
11 file(s) 19612 bytes
```

```

Directory of F:\SUBDIR\NOFILE
.          <DIR>      93-03-03  10:08
..         <DIR>      93-03-03  10:08
          2 file(s)      0 bytes

Directory of F:\SUBDIR\USERKE
.          <DIR>      93-03-03  10:08
..         <DIR>      93-03-03  10:08
          2 file(s)      0 bytes

Directory of F:\SUBDIR\VANBENNE
.          <DIR>      93-03-03  10:08
..         <DIR>      93-03-03  10:08
PRINT     TXT       205 93-03-03  5:58
TEXT      TXT       140 93-03-02  10:35
          4 file(s)     345 bytes

Directory of F:\SZOVEG
.          <DIR>      93-03-03  9:44
..         <DIR>      93-03-03  9:44
          2 file(s)      0 bytes

Total files listed:
          25 file(s)    19957 bytes
                          145408 bytes free

F:\>_

```

Most másoljuk át a 'SZOVEG' könyvtárba az 1993.március 3.-a után módosított és létrehozott fájlokat, ellenőrzéssel, és az átmásolt fájloknak töröljük az archív bitjét. Próbáljuk ki a /w paraméter hatását is. A parancs végrehajtása előtt nézzük meg az eredeti könyvtár listáját, hogy a parancsoknak melyik fájlokat kell átmásolnia.

```

F:\SUBDIR>xcopy *.* f:\szoveg /m /v /w /d:1993/03/03
Press any key to begin copying file(s)
Reading source file(s)...
XCOPY.TXT
PRINT.TXT
2 File(s) copied

F:\SUBDIR>_

```

A /w paraméter hatására kapjuk az első sorban lévő üzenetet, és a számítógép egy billentyű leütésére vár.

Ellenőrizzük a **DIR** paranccsal, hogy tényleg átmásoltuk-e a kiválasztott fájlokat:

```
F:\SUBDIR>dir \szoveg

Volume in drive F is DISK CIMKE~
Volume Serial Number is 256D-0EF4
Directory of F:\SZOVEG

.           <DIR>          93-03-03   9:44
..          <DIR>          93-03-03   9:44
PRINT      TXT           205 93-03-03   5:58
XCOPY      TXT          8843 93-03-03   9:31
           4 file(s)      9048 bytes
           131072 bytes free

F:\SUBDIR>_
```

Most nézzük meg az **ATTRIB** paranccsal, hogyan változtak meg a jelzőbitek. Az átmásolt fájloknak az archív bitje törlődött.

```
F:\SUBDIR>attrib
F:\SUBDIR\XCOPY.TXT
A      F:\SUBDIR\UNDELETE.TXT
A      F:\SUBDIR\TEXT.TXT
A      F:\SUBDIR\MEM.TXT
A      F:\SUBDIR\DISKCOPY.TXT
       F:\SUBDIR\PRINT.TXT

F:\SUBDIR>_
```

Azok a fájlok, amelyeket még nem másoltunk át, azok 2.-ai keltezésűek. Most másoljuk át azokat is:

```
F:\SUBDIR>xcopy *.* \szoveg /m /d:1993/03/02
Reading source file(s)...
UNDELETE.TXT
TEXT.TXT
MEM.TXT
DISKCOPY.TXT
       4 File(s) copied
F:\SUBDIR>_
```

Most lássuk, hogyan változtak meg a jelzőbitek:

```
F:\SUBDIR>attrib
       F:\SUBDIR\XCOPY.TXT
       F:\SUBDIR\UNDELETE.TXT
       F:\SUBDIR\TEXT.TXT
       F:\SUBDIR\MEM.TXT
       F:\SUBDIR\DISKCOPY.TXT
       F:\SUBDIR\PRINT.TXT

F:\SUBDIR>_
```

Ha most megpróbálunk másolni, akkor megbizonyosodhatunk arról, hogy a /m paraméter hatására, csak azokat másolja át, amelyeknek az archív jelzőbitje be van állítva.

```
F:\SUBDIR>xcopy *.* \szoveg /m
        0 File(s) copied

F:\SUBDIR>_
```

Adjuk meg a dátum mellett az időt is, aminél frissebb fájlokat akarunk másolni.

```
F:\SUBDIR>xcopy *.* f:\szoveg /m /d:1993/03/02 11:00
Invalid number of parameters
        0 File(s) copied

F:\SUBDIR>_
```

A válasz: "Nem megfelelőek a paraméterben a számok."

Láthatjuk, minden fájl megvan, nem azért nem másolja át mert eltűntek az állományok, hanem a jelzőbiték állása miatt:

```
F:\SUBDIR>dir

Volume in drive F is DISK CIMKE~
Volume Serial Number is 256D-0EF4
Directory of F:\SUBDIR

NOFILE      <DIR>      93-03-03   10:08
USERKE      <DIR>      93-03-03   10:08
VANBENNE    <DIR>      93-03-03   10:08
.           <DIR>      93-03-03    9:41
..          <DIR>      93-03-03    9:41
DTSKCOPY   TXT        432 93-03-02   12:41
MEM        TXT        4779 93-03-02   12:14
PRINT      TXT        205 93-03-03    5:58
TEXT       TXT        140 93-03-02   10:35
UNDELETE   TXT        5213 93-03-02   10:13
XCOPY      TXT        8843 93-03-03    9:31
        11 file(s)      19612 bytes
                               114688 bytes free

F:\SUBDIR>_
```

Véletlenül kimaradt a célkönyvtár a parancsból. A számítógép válasza:
"Nem tudja végrehajtani a ciklikus másolást."

Azért mondja, hogy ciklikus másolás, mert a célt nem adtuk meg, így a célon a fájlnevek a forrás neveivel megegyeznek. De a könyvtárak neveit sem adtuk meg, így azok az aktuális könyvtárra vonatkoznak. Végeredményben a megadott fájlokat ugyanoda másolnánk, mint ahol vannak:

```
F:\SUBDIR>xcopy *.* /m
Cannot perform a cyclic copy
0 File(s) copied

F:\SUBDIR>_
```

Megpróbáltunk az aktuális könyvtárból másolni, de ott már nem maradt olyan fájl, amit archiválni kellene. (Persze a /a paraméterrel bármennyiszer tudnánk másolni, mert az nem változtatja meg az archív jelzőbitet.) Most akkor másoljuk át az aktuális könyvtár alkönyvtárait, és az abban fellelhető fájlokat:

```
F:\SUBDIR>xcopy *.* \szoveg /m /s /e
Reading source file(s)...
VANBENNE\PRINT.TXT
VANBENNE\TEXT.TXT
2 File(s) copied

F:\SUBDIR>_
```

Visszalépve a gyökérkönyvtárba, nézzük meg hogyan alakultak a fájlok jelzőbitjei.

```
F:\SUBDIR>cd\

F:\>attrib /s
F:\SUBDIR\VANBENNE\PRINT.TXT
F:\SUBDIR\VANBENNE\TEXT.TXT
F:\SUBDIR\XCOPY.TXT
F:\SUBDIR\UNDELETE.TXT
F:\SUBDIR\TEXT.TXT
F:\SUBDIR\MEM.TXT
F:\SUBDIR\DISKCOPY.TXT
F:\SUBDIR\PRINT.TXT
A      F:\SZOVEG\VANBENNE\PRINT.TXT
A      F:\SZOVEG\VANBENNE\TEXT.TXT
A      F:\SZOVEG\XCOPY.TXT
A      F:\SZOVEG\UNDELETE.TXT
A      F:\SZOVEG\TEXT.TXT
A      F:\SZOVEG\MEM.TXT
A      F:\SZOVEG\DISKCOPY.TXT
A      F:\SZOVEG\PRINT.TXT

F:\>_
```

Most pedig lássuk a másolásaink eredményét:

```
F:\SUBDIR>dir \ /s

Volume in drive F is DISK_CIMKE~
Volume Serial Number is 256D-0EF4

Directory of F:\

SUBDIR      <DIR>    93-03-03    9:41
SZOVEG      <DIR>    93-03-03    9:44
.           <DIR>    93-01-21   15:21
..          <DIR>    93-01-21   15:21
            4 file(s)      0 bytes

Directory of F:\SUBDIR

NOFILE      <DIR>    93-03-03   10:08
USERKE      <DIR>    93-03-03   10:08
VANBENNE    <DIR>    93-03-03   10:08
.           <DIR>    93-03-03    9:41
..          <DIR>    93-03-03    9:41
DISKCOPY TXT    432 93-03-02   12:41
MEM         TXT    4779 93-03-02   12:14
PRINT      TXT    205 93-03-03    5:58
TEXT       TXT    140 93-03-02   10:35
UNDELETE   TXT    5213 93-03-02   10:13
XCOPY      TXT    8843 93-03-03    9:31
            11 file(s)  19612 bytes

Directory of F:\SUBDIR\NOFILE

.           <DIR>    93-03-03   10:08
..          <DIR>    93-03-03   10:08
            2 file(s)      0 bytes

Directory of F:\SUBDIR\USERKE

.           <DIR>    93-03-03   10:08
..          <DIR>    93-03-03   10:08
            2 file(s)      0 bytes

Directory of F:\SUBDIR\VANBENNE

.           <DIR>    93-03-03   10:08
..          <DIR>    93-03-03   10:08
PRINT      TXT    205 93-03-03    5:58
TEXT       TXT    140 93-03-02   10:35
            4 file(s)      345 bytes

Directory of F:\SZOVEG

NOFILE      <DIR>    93-03-10   13:08
USERKE      <DIR>    93-03-10   13:08
VANBENNE    <DIR>    93-03-10   13:08
.           <DIR>    93-03-03    9:44
..          <DIR>    93-03-03    9:44
DISKCOPY TXT    432 93-03-02   12:41
MEM         TXT    4779 93-03-02   12:14
PRINT      TXT    205 93-03-03    5:58
TEXT       TXT    140 93-03-02   10:35
UNDELETE   TXT    5213 93-03-02   10:13
XCOPY      TXT    8843 93-03-03    9:31
```

```

        11 file(s)      19612 bytes

Directory of F:\SZOVEG\NOFILE
.           <DIR>      93-03-10   13:08
..          <DIR>      93-03-10   13:08
        2 file(s)           0 bytes

Directory of F:\SZOVEG\USERKE
.           <DIR>      93-03-10   13:08
..          <DIR>      93-03-10   13:08
        2 file(s)           0 bytes

Directory of F:\SZOVEG\VANBENNE
.           <DIR>      93-03-10   13:08
..          <DIR>      93-03-10   13:08
PRINT    TXT        205 93-03-03    5:58
TEXT     TXT        140 93-03-02   10:35
        4 file(s)       345 bytes

Total files listed:
        42 file(s)      39914 bytes
                        100352 bytes free

F:\SUBDIR>_

```

DEL**ERASE**

Állományok törlése

DEL [*meghajtó*] [*elérési_út*] *fájlnev* [/P]
ERASE [*meghajtó*] [*elérési_út*] *fájlnev* [/P]

Egy vagy több fájlt lehet vele törölni.
 Belső parancs.

meghajtó: Az a meghajtó, amelyiken a törlendő fájlok vannak. Ha elmarad, akkor az aktuális meghajtón keres.

elérési_út: Annak a könyvtárnak az elérési útja és neve, amelyiken a törlendő fájlok vannak. Ha a könyvtár elérési útja elmarad, akkor az aktuális könyvtárban keresi azt, és abban a törlendő fájlokat. Ha a könyvtár neve is elmarad, akkor a fájlokat aktuális könyvtárban keresi.

fájlnev: A törlendő fájl neve vagy fájlok csoportja. Csak egy fájl vagy egy fájlcsoport jelölhető ki. (Ha elmarad, akkor mit akarsz törölni?) A fájlok csoportját a helyettesítő karakterekkel lehet kijelölni (* ?). A '*' helyettesít minden további karaktert, a '?' pedig csak azt, amelyiknek a helyén áll. Például:

DEL a*.*cc

Töröl minden 'a' betűvel kezdődő olyan fájlt, amely kiterjesztésének az első karaktere tetszőleges, de a 2. és 3. az 'c'. A fájl neve bármilyen hosszú is lehet.

DEL AT93??01.D*

Töröl minden olyan fájlt, amelyik nevének az 1.-4. karaktere 'AT93', a 7.-8. '01', de az 5.-6. az teljesen tetszőleges lehet, a kiterjesztése pedig 'D' betűvel kezdődjön, de bármivel folytatódhat. Pl.:

```
at930101.dat, at930201.dat, at930301.dat, ...
at930101.def, at930201.def, at930301.def, ...
at930101.dbf, at930201.dbf, at930301.dbf, ...
at930101.dix, at930201.dix, at930301.dix, ...
```

/p: (PROMPT - kiértésít) A törlés előtt megkérdezi minden kijelölt fájlról, hogy törölheti-e. Használata akkor célszerű, ha egyszerre több fájlt jelöltünk ki. (Például: **DEL at*.*** , de az "atari.exe"-t nem akarjuk törölni, csak az "AT+dátum.*" adatbázisokat. /Lásd az előző példában!/).

A **DEL** és az **ERASE** parancsok ugyanúgy hajtják végre a törlést, gyakorlatilag az egyik felesleges is. (Más rendszerekkel való kompatibilitás miatt van meg az **ERASE** parancs.) Általában a **DEL** parancsot használjuk, mert az a rövidebb.

Példák:

```
D:\TEMP>del
Required parameter missing

D:\TEMP>del *.1 /p
File not found

D:\TEMP>attrib unformat.txt +h

D:\TEMP>del *.* /p      {ami azonos a "DEL . /p"-vel}

D:\TEMP\BORITEK.$$1, Delete (Y/N)? y
D:\TEMP\FORMAT.TXT, Delete (Y/N)? n
D:\TEMP\UNFORMAT.TXT, Delete (Y/N)? y
Access denied.

D:\TEMP>_
```

Ne adjunk meg törlendő fájlnevet: A válasz "Kötelező a hiányzó paraméter" , vagyis ami hiányzik a fájl neve, azt kötelező megadni.

Adjunk meg biztosan nem létező fájlnevet:

És a számítógép "Fájlt nem találok" -választ ad.

Hajtsunk végre egy valós törlést is: De vigyázva, nehogy kárt tegyünk. Használjuk a **/p** -t, így biztos nem törölünk le véletlenül olyan fájlokat, amire még szükségünk lehet. A törlések előtt a számítógép megkérdezi minden fájlról, hogy "Töröljem (Igen/Nem) ?" , amire válaszolhatunk ' Y ' -t (YES), vagy ' N ' -t (NO) is, ha meggondoltuk magunkat.

UNDELETE

Törölt állományok visszahozatala

UNDELETE [*meghajtó: elérési_út*] [*fájlnev*] [/LIST | /ALL] [DT | DOS]

Törölt fájlok helyreállítását lehet vele megkísérelni.

Külső parancs.

meghajtó: \elérési_út A helyreállítandó fájlok pontos helye a könyvtárfán. Csak együtt adható meg, vagy mind a kettő elmarad. Akkor viszont az aktuális könyvtárban keres.

fájlnev A helyreállítandó fájl vagy fájlok neve. Helyettesítő karaktereket is lehet használni. Ha elmarad, akkor megnézi a megadott könyvtárban a törölt fájlokat, és mindegyiknél egyesével kiírja, hogy helyreállítható-e, vagy sem. Ha megmenthető, akkor bekéri a fájl nevének első betűjét, és visszateszi a fájlt a könyvtár bejegyzései közé.

/LIST A paraméter hatására a könyvtárban fellelhető összes törölt fájlt kilistázza (LIST), megjelölve (**) azokat, amelyeket már nem lehet helyreállítani. Ennek segítségével megnézhetjük, hogy elveszetteknek hitt programjaink tényleg elvesztek-e.

/ALL Minden (ALL) helyreállítható állományt visszahoz, és az első karakterüket '#' karakterrel helyettesíti, amit a **REN** paranccsal átnevezhetünk. Ha ilyen nevű állomány már létezik, akkor a következő karakterek közül választ a sorból:
#&-0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ

A */ALL* és a */LIST* paramétereket együtt használni nem lehet.

/DT fájlokat a **MIRROR** által létrehozott állományban lévő információk alapján próbálja meg helyreállítani. (A **MIRROR** parancs a "*DOS alapismeretek II.*" -ben található.)

/DOS A paraméter hatására a fájlokat a megadott könyvtárban lévő információk alapján próbálja meg helyreállítani.

A */DT* és a */DOS* paramétereket együtt használni nem lehet.

Példák:

```
D:\TEXT>undelete /list
Directory: D:\TEXT
File Specifications: *.*

Deletion-tracking file not found.

MS-DOS directory contains 2 deleted files.
Of those, 1 files may be recovered.

Using the MS-DOS directory.

      ?TATUS   ME       727 93-03-02  8:58  ....
      ** ?ABCODER $$$ 12068 93-02-26 13:45  ...A

"***" indicates the first cluster of the file
is unavailable and cannot be recovered
with the UNDELETE command.

D:\TEXT>_
```

Az első példában megnézzük milyen helyreállítandó fájlok vannak az aktuális könyvtárban. A kiírt üzenetek általában ismétlődnek, így azokat csak egyszer ismertetem.

Először annak a könyvtárnak a nevét írja ki, amelyikben keresni fogja a fájlokat.

Utána a helyreállítandó állomány vagy állományok nevét írja ki. Ha nem adtunk meg semmit sem, akkor alapértelmezés szerint '*.*', azaz minden fájlt feldolgoz.

"Törleszt-nyomkövetés fájlt nem találtam" Első megközelítésben ezt keresi, és ha nincs, akkor nézi meg a könyvtárbejegyzéseket. Ezt a fájlt a **MIRROR** parancs hozza létre, a törölt fájlok pontosabb nyilvántartása végett. (Erről a "DOS alapismeretek II."-ben olvashattok.)

"Az MS-DOS könyvtár tartalmaz 2 törölt fájlt,
amiből 1 fájlt lehetséges helyreállítani."

"Használok az MS-DOS könyvtárat" Használhatná a 'törleszt-nyomkövetés' fájlt is, de nem adtuk meg neki, no meg nincs is. (c:)

Az egyik helyreállítható fájlról közöl információkat: A neve első betűje ismeretlen, a többi **'TATUS.ME'**. A méretét (727 bájt) és utolsó módosításának idejét is megtudhatjuk (1993.03.02 és 8:58-kor). A fájlnek a jelzőbitjeit is kiírja, amik a törlés előtt voltak érvényben. (Itt most egyik sem volt beállítva, de a másoknak az archív bitje igen.)

Ennek a fájlnek az adatai előtt láthatunk **'**'** -ot. Ez jelzi, hogy a fájlnek az első klasztere nem áll rendelkezésünkre, és nem lehet helyreállítani az **UNDELETE** paranccsal."

```
D:\TEXT>undelete /all
Directory: D:\TEXT
File Specifications: *.*
Deletion-tracking file not found.
MS-DOS directory contains 2 deleted files.
Of those, 1 files may be recovered.
Using the MS-DOS directory.

      ?TATUS  ME      727 93-03-02 8:58  ....
      File successfully undeleted.
** ?ABCODER $$$ 12068 93-02-26 13:45 ...A
      Starting cluster is unavailable.
      This file cannot be recovered
with the UNDELETE command. Press any key to continue.
D:\TEXT>_
```

Most állítsuk helyre az összes fájlt, amit csak lehet ebben a könyvtárban: A kiírások már ismerősek, (ha nem, akkor nézd meg az előző példát!) következhet a helyreállítás:

Kiírja a megtalált, törölt fájlt az ismert adataival. "A fájlt sikeresen helyreállítottam."

A következő helyreállítandó fájl **'**'** jelzésű volt, azaz nem tudja már visszahozni. (lásd fent!) Ha most egy **DIR #*.*/A** parancsot kiadunk, akkor láthatjuk meg a helyreállított állományok listáját. Esetünkben csak a **'#TATUS.ME'**, amit a **RENAME** paranccsal újjá varázsolhatunk: **REN #tatus.me status.me**

```

D:\TEXT>del #tatus.me
D:\TEXT>undelete

Directory: D:\TEXT
File Specifications: *.*
Deletion-tracking file not found.
MS-DOS directory contains 2 deleted files.
Of those, 1 files may be recovered.
Using the MS-DOS directory.

    ?TATUS ME      727 93-03-02 8:58  ....
    Undelete (Y/N)? y
Please type the first character for ?TATUS .ME:s
File successfully undeleted.

    ** ?ABCODER $$$ 12068 93-02-26 13:45  ...A
Starting cluster is unavailable.
This file cannot be recovered
with the UNDELETE command. Press any key to continue.
D:\TEXT>_

```

Most mielőtt használnánk a parancsot, töröljük le újra a 'STATUS.ME' fájlunkat (vagy azt, amelyik fájlra te kísérletezel) azért, hogy az UNDELETE parancsot kipróbálhassuk paraméter nélkül is. A parancs a szokásos információkat írja ki megint. Most viszont a helyreállítható fájlról megkérdezi: "Helyreállítsam (Igen/Nem)?"

Ha igen, akkor: "Kérem beírni az első karakterét a ?TATUS .ME fájlra:"

A következőre leütött billentyű (**[S]**) lesz az első jegye a helyreállított fájl nevének.

"A fájlt sikeresen helyreállítottam."

A '******' jelzi, hogy a fájlra "Az első klasztere nem áll rendelkezésünkre, és nem lehet helyreállítani az UNDELETE paranccsal. Egy billentyű leütése után folytatom" Mivel nincs több törölt fájl a megadott könyvtárban, így be is fejezi, visszaadja a DOS készenléti jelet.

RENAME

REN

Állományok átnevezése

RENAME [*meghajtó:*] [*elérési_út*] *régi_fájlnev új_fájlnev*

REN [*meghajtó:*] [*elérési_út*] *régi_fájlnev új_fájlnev*

Fájlok átnevezésére szolgál.

Belső parancs.

meghajtó: Az meghajtó, amelyiken az átnevezendő fájl van. Ha nem adjuk meg, akkor az aktuális meghajtón van a keresett fájl.

elérési_út: Az átnevezendő fájlok könyvtárának neve és elérési útja. Ha nem adjuk meg, akkor az aktuális könyvtárban kell lennie az átnevezendő fájloknak.

régi_fájlnev: Az átnevezendő fájl vagy fájlok neve. Használhatunk dzsókerkaraktereket is.

új_fájlnev: Az átnevezésre kerülő fájl vagy fájlok új neve. Itt is használhatunk helyettesítő karaktereket.

Példák:

```
D:\TEMP>ren fc.txt fc.txt
Duplicate file name or file not found

D:\TEMP>ren fc.txt fc.doc

D:\TEMP>ren d:\dos\readme.txt olvasd.el

D:\TEMP>ren e:\adatok\at930101.* jan.*

D:\TEMP>_
```

Ha az átnevezéssel nem is nevezzük át a fájlt, akkor mi értelme van a parancsnak? A számítógép válasza:

"Megkettőzött fájlnev vagy nem találok a fájlt"

Tehát, ha olyan fájlnevet adunk meg, ami nem létezik, akkor is ugyanezt az üzenetet kapjuk.

Általában nem szándékosan adjuk meg rosszul a szükséges paramétereket, hanem tévedésből, vagy hiányos ismereteink miatt. Az ilyen kettős hibaüzeneteknél meg kell vizsgálni mind a két hiba lehetőségét.

A következő egy szabályos átnevezés, az aktuális könyvtárban lévő fájl esetében.

Utána egy másik szabályos átnevezési parancs látható, teljes paramétermegadással.

Fájlok csoportos átnevezésére láthatunk példát, ahol minden **at930101** nevű (tetszőleges kiterjesztésű) fájl **jan** nevű lesz, és a kiterjesztéseket rendre átörökli:

at930101.dat	⇒	jan.dat
at930101.def	⇒	jan.def
at930101.dbf	⇒	jan.dbf
at930101.dix	⇒	jan.dix

Távoli fájlok elérése, környezeti változók

Mint ismeretes, a belső parancsok bármely könyvtárból indíthatók, de a külső parancsok, és a többi futtatható fájl csak abból, amelyikben van. Ezen a helyzeten segít nekünk a **PATH** (útvonal), az **APPEND** (hozzáfűz) és a **FASTOPEN** (gyorsnyitás) parancs. Ezek megfelelő használatával meg tudjuk gyorsítani a fájlok indításakor eltelő időt.

Ha a **PATH** parancsot használjuk, akkor a programjainkat nem csak az aktuális könyvtárból indíthatjuk, hanem más távoli könyvtárakból is.

Az **APPEND** parancs segít abban, hogy a távoli könyvtárakban lévő adatbázisainkat is használhassuk.

A **FASTOPEN** parancs pedig nyilvántartást készít az egyszer már elindított fájlok helyéről, és ha máskor is használjuk, akkor már nem fogja a DOS keresni, hanem azonnal tudni fogja, hogy honnan indítsa el.

A **PATH** és az **APPEND** parancsok a *keresési könyvtárakat környezeti változó*ban tartják. *Változónak* a számítástechnikában azokat a névvel ellátott adatokat nevezzük, amelyek a program futása során értéküket megváltoztathatják. *Környezeti változók* pedig azok, amelyek a DOS környezetében használhatók, annak használatát segítik, az értéküket pedig a **SET** (beállít) paranccsal lehet megváltoztatni.

A *keresési könyvtárak* tárolása ';' -vel elválasztott könyvtárak elérési útjának és nevének felsorolásából áll. A munkánk során hozzátehetünk, elvehetünk belőle, igényeink szerint változtathatjuk a tartalmát. A keresés a változóban felsorolt könyvtárakon megy végig, amíg nem jár eredménnyel. Az igazsághoz hozzátartozik az is, hogy a változó értékének megváltoztatása lehetséges ugyan, de nem olyan kényelmes, mint gondolnánk. Egyszerűbb esetben csak a változó tartalmának teljes lecserélése lehetséges egy újra, de egy megfelelő *batch-fájl* (csomó, köteg - itt kötegelt parancs fájl) futtatásával hozzáírhatunk újabb könyvtárneveket a felsoroláshoz. Törölni viszont nem tudunk a felsorolt listából.

PATH

Keresési könyvtárlista

PATH [[*meghajtó*:] *elérési_út* [;...]] [;]

A keresési útvonalat lehet megváltoztatni, kiírni vele.

Belső parancs.

Ha *paraméter nélkül* adjuk ki a parancsot, akkor kiírja a változó aktuális értékét.

elérési_út: Annak a könyvtárnak a neve és elérési útja, amelyben az indítható fájljaink vannak. Az elérési utat a gyökérkönyvtártól kell megadni.

meghajtó: Az elérési út pontosabb meghatározása céljából a meghajtó nevét is megadhatjuk. Ha elmarad, akkor - nagyon fontos !!! - az aktuális meghajtóra vonatkoznak a megadott könyvtárelérési utak. Erre akkor van szükség, ha több meghajtón is dolgozunk egyszerre.

A keresés sorban megy végig a felsorolt könyvtárakon, amíg az eredménnyel nem jár. Ha olyan meghajtót is talál a felsoroltak között, ami nem is létezik, akkor minden alkalommal, amikor nem az aktuális könyvtárból indítunk egy programot, ki fogja írni, hogy:

Invalid drive specifications. (Nem létező meghajtó megadása)

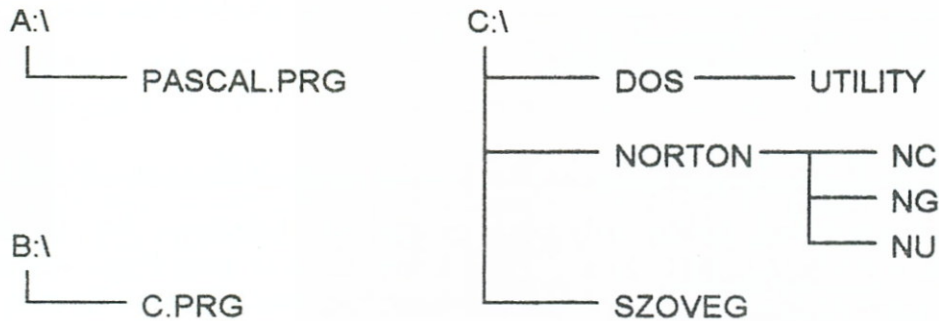
Persze a keresést folytatja tovább a többi felsorolt könyvtárban. Ha a keresés nem járt eredménnyel, akkor lehet, hogy a fájl nem is létezik, vagy lehet, hogy rosszul írtuk be a nevét. Ekkor írja ki azt a beírt parancsunkra, hogy:

Bad command or file name. (Rossz parancs vagy fájl név)

Egyszerre több keresési könyvtár megadás is lehet a **PATH** -ban, amelyeket ';' -el kell egymástól elválasztani. Az utolsó könyvtár megadása után nem kötelező ';' -t megadni. A **PATH** változó tartalmának teljes törlése a '**PATH**;' paranccsal történik.

A hossza maximum 127 bájttal lehet, bár meggondolandó, hogy ha minden parancs kiadásánál, majdnem az egész merevlemezt végig járja a parancs, milyen lassan fognak a parancsaink végrehajtódni.

Példák:



A **PATH** parancs paraméter nélkül használva, kiírja a **PATH** környezeti változó értékét.

```

C:\>path
PATH=c:\dos;c:\norton

C:\>_
  
```

Itt egy helytelen megadásra láthatunk példát: A keresési úton megadtunk olyan könyvtárakat is, amelyek nem a gyökérkönyvtárból indulnak. A **UTILITY** könyvtár megadásra nem fog hibát jelezni, de nem is jó. Viszont azoknál a meghajtóknál, amelyek nem léteznek, hibaüzenetet fog adni (**K: X: Y:**).

```

C:\>path \norton;utility;k:\;x:\dos;y:\public

C:\>_
  
```

Most az előző **PATH** -t felhasználva írjunk be egy nem létező parancsot. A DOS végig nézi a belső parancsok listáját, az aktuális könyvtárban lévő fájlokat, és ha nem talál egyezést a megadott paranccsal, akkor végig megy az elérési úton. Ott persze találni fog néhány olyan meghajtót, amelyek nem érhetők el. Ezekre hibaüzenettel fog reagálni. Miután végignézte a keresési útvonalat, és a keresés eredménytelen volt, újabb hibaüzenettel leáll.

```

C:\>dfkjh
Invalid drive specification
Invalid drive specification
Invalid drive specification

Bad command or file name

C:\>_
  
```

Ez a parancs a **PATH** törlésére szolgál. Ez után program, külső parancs, csak az aktuális könyvtárból indítható, illetve távoli programok indítása esetén az elérési utat is meg kell adni. (Amint látható ez elég hosszadalmas eljárás minden parancs kiadásánál.)

```

C:\>path ;

C:\>path
No path.

C:\>\dos\utility\scan108.exe a:

C:\>_
  
```

Itt egy helyes **PATH** megadást láthatunk, több meghajtót is használunk egyszerre.

```
C:\>path c:\dos;c:\norton;a:\pascal.prg;b:\c.prg
C:\>path \dos;\norton;\pascal.prg;\c.prg

C:\>_
```

Ez a példa viszont az előző példa helytelen megadása. Ugyanis, ha az aktuális meghajtó a **C:**, akkor csak a **PASCAL.PRG** és a **C.PRG** könyvtárakat nem fogja megtalálni, mert honnan találja ki a számítógép, hogy hol vannak ezek a könyvtárak. Viszont, ha az aktuális meghajtó a **D:** (a másik merevlemez), akkor egyetlen keresési úton lévő programunk sem fog elindulni.

APPEND

Adatállományok keresési útja

APPEND [/E] [/X [:ON | OFF]] [/PATH:ON | /PATH:OFF]

APPEND [[meghajtó:] elérési_út [;...]]

APPEND [;]

Távoli könyvtárakban lévő adatállományok megnyitását biztosítja.

Külső parancs.

elérési_út: Annak a könyvtárnak a neve és elérési útja, amelyben az adatfájljaink vannak. Az elérési utat a gyökérkönyvtártól kell megadni.

meghajtó: Az elérési út pontosabb meghatározása céljából a meghajtó nevét is megadhatjuk. Ha elmarad, akkor - nagyon fontos !!! - az aktuális meghajtóra vonatkoznak a megadott könyvtár elérési utak. Erre akkor van szükség, ha több meghajtón is dolgozunk egyszerre.

/x /x:on /x:off (Alapértelmezésben **/x:off**) A parancs az első indítás alkalmával rezidensen beül a memóriába, és folyamatosan nyomköveti a munkánkat. Ha használjuk ezt a paramétert, akkor a megadott keresési utat több megszakítási funkciónál is használni fogja. Egyszerűbben, több parancsnál tudjuk használni a keresési utak adta előnyöket. (A megszakítási rendszerbe még korai lenne belebonyolódni, azért nem fejtem ki bővebben.)

/path:on /path:off (Alapértelmezésben **/path:on**) A programok, az olyan állományok megnyitásánál is használhatják a keresési utakat, amelyeknél meg van adva az elérési út is az állomány nevéhez. Ez az **ON** esete, az alapértelmezés, ha nem adjuk meg a paramétert.

/e: A keresési utat egy **APPEND** nevű környezeti változóban fogja tárolni, ha használjuk ezt a paramétert.

Az előző három paramétert csak a rendszer indulása utáni első alkalommal használhatjuk. Ekkor keresési utakat nem lehet megadni. A program, mint írtam rezidensen beül a memóriába, és a használatához szükséges paramétereket ekkor kell megadni. A keresési utak listája ezek után adható meg, változtatható tetszőlegesen, de az alapbeállításokat már nem módosíthatjuk. A parancsot több formában használhatjuk, például:

```
APPEND /e /x:on /path:off
```

```
APPEND a:\adatok;c:\szoveg
APPEND ;
APPEND
```

1. A rendszer indulása után az első **APPEND** parancsunk valahogy így nézhet ki, ha meg akarjuk változtatni az alapértelmezésbeli beállításokat.
2. Az előbbi változtatásokat ha kérjük, ha nem, ezt a formát mindig használhatjuk a keresési utak megadására. Ha az előző parancsot kiadtuk, akkor viszont tanácsos is használni, mert különben fölöslegesen foglalja a helyet a memóriában.
3. A keresési utak törlését mutatja be ez a példa.
4. Kiírja a megadott keresési utak listáját.

FASTOPEN

Állományok gyors megnyitása

FASTOPEN *meghajtó*: [=n] [;...] [/X]

Meggyorsítja az állományok elérését a merevlemeze(ke)n.

Külső parancs.

meghajtó: Annak a meghajtónak a betűjele, amelyiken a gyakran használt állományaink vannak. Több meghajtót is meg lehet adni, de egyet kötelező, mert itt nincs alapértelmezett meghajtó.

[=] n Egy pozitív egész szám, ami a feljegyzendő állományok száma. Alapértelmezésben ez 48, csak akkor kell megadni, ha ezen az értéken változtatni akarunk. Az egyenlőség jelet nem kötelező kiírni. A megadható értékek: 10-999.

/x Az állományok adatait a kiterjesztett memóriában jegyzi fel. (Ez a 640 kB feletti memóriaterület. Lásd "*DOS alapismeretek II.*")

Minden feljegyzés 48 bájtot foglal le a memóriából.

A program úgy működik, hogyha egy programot betöltünk a merevlemezezről, akkor annak nevét, helyét és egyéb szükséges adatait feljegyzi a memóriába. Ha másodszor is használjuk a programot, akkor már nem fogja keresgéléssel tölteni az időt, hanem tudni fogja azonnal, hogy honnan töltse be. Ezzel tudja meggyorsítani a programok betöltését, de csak merevlemezezről.

A program rezidens a memóriában, és kb 3 kilobájtot foglal le. Ha van két merevlemezeünk, és mindegyiknek 100-100 bejegyzést lefoglalunk, akkor sem éri el a memória foglalása a 13 kb-ot. Ez nem olyan nagy terület, hogy ne érné meg lefoglalni. Persze attól függ hányszor indítjuk a programokat. Ha egész munkánk alatt csak egy-két programot használunk, akkor fölöslegesen foglalnánk le a memóriát.

Használata:

```
FASTOPEN c:=100 d:10
```

```
FASTOPEN c: d:
```

SET

Környezeti változók beállítása

SET [változó= [füzér]]

A DOS környezeti változóinak értékét lehet megváltoztatni vele.

Belső parancs.

változó: Egy DOS környezeti változó (ENVIRONMENT VARIABLES) neve. Az egyenlőség jelet mindig kötelező kiírni. Ez választja el a változó nevét az értékétől.**füzér:** Egy tetszőleges karakterfüzér (STRING), ami a változó értéke lesz. Hogy mit tartalmazzon, azt az adott program, parancs dönti el, amelyik használja a változót.

Egy környezeti változó törlését a **SET változó=** paranccsal tehetjük meg. Egész pontosan a semmit rendeljük hozzá a változó nevéhez. A változók aktuális értékeinek kiírása a paraméter nélküli **SET** paranccsal lehetséges. Már van néhány ismert környezeti változónk, például:

parancs	változó=értéke (például)
PROMPT	PROMPT=\$P\$G
DIR	DIRCMD=/O:N /A /P
DIR	TEMP=C:\TEXT\TEMP
PATH	PATH=C:\DOS;C:\UTILITY
APPEND	APPEND=A:\ADATOK;C:\SZOVEG

Persze ezt a sort tovább bővíthetjük saját használatú változókkal is. Például:

SET USER=Móricz Attila

Ezt egy batch-fájlból az **ECHO Üdvözöllek %USER%** paranccsal kiírathatjuk a rendszer indulásakor, és akkor minden rendszerinduláskor üdvözölni fog a számítógép bennünket.

A környezeti változók részére alapértelmezésben 256 bájttal áll a rendelkezésre. Ha ezt a területet túl akarjuk lépni, akkor az:

Out of environment space

"A környezeti változóknak a helye elfogyott" üzenetet kapjuk. Ebben az esetben, vagy lemondunk a változó használatáról, vagy törölünk néhány olyan változót, amit már nem használunk.

Az állományok megjelenítése különböző helyeken

Ahhoz, hogy egy állomány tartalmát megnézhessük, azon változtathassunk, általában egy szövegszerkesztőre van szükségünk. Ennek és a szerkesztendő programnak a betöltése időt vesz igénybe. Vannak olyan esetek, amikor csak megtekinteni akarjuk a fájl tartalmát, de változtatni nem akarunk rajta. Ekkor elegendő egy olyan parancs, ami kiírja az állományt a monitorra vagy a nyomtatóra. A következőkben ismertetésre kerülnek azok a parancsok, amelyekkel a programjainkat megjeleníthetjük. Ehhez azonban előbb meg kell ismerkednünk a DOS-ban használható perifériákkal, adatátírási módszerekkel. Ugyanis egy parancs, ha kiírja a kívánt adatokat alapértelmezésben a monitorra, akkor mi azt át is irányíthatjuk

más eszközre is (pl: LPT1 /első nyomtató/). Vannak viszont olyan programok is, amelyek bemenő adatokat várnak alapértelmezésben a billentyűzetről.

Ezt is átirányíthatjuk úgy, hogy az adatait például fájlból fogja kapni. Ha pedig két olyan parancsot használunk, amelyiknél az egyik adatokat ír ki, a másik pedig adatokat vár, amelyeket átalakítva ír ki, akkor beszélünk a parancsok összefűzéséről. Ha az adatokat valamely külső egységre akarjuk irányítani, akkor vegyük figyelembe a számítógépünk hardverkiépítését. Tehát ne akarjunk az LPT4 -re írni, ha csak egy monokróm hercules videokártya van a gépünkben, és csak az azon lévő LPT1 nyomtatócsatlakozóra köthetünk nyomtatót. Tehát az ismertetésre kerülő eszköznevek skálája lehet bővebb is, és szűkebb is, egy adott számítógépen. (Eltérés általában a COM és LPT kimenetekben lehet.)

Állományok átirányítása

Fájl, CON, NUL, PRN, LPTn, AUX, COMn

Azok az eszközök, ahová egy program kimenetét irányíthatjuk.

Nem parancsok, de a COMMAND.COM dolgozza fel.

Fájl: Ez nem eszköznév, hanem egy állomány neve.

Lehet egy parancsnak ki- és bemenete is egyaránt.

CON: CONSOLE (magyarban kiálló rész, itt más)

Bemenetkor: a billentyűzetről jön az adat

Kimenetkor: a képernyőre megy az adat.

NUL: NULLa (semmi) Nem kell kiírás.

PRN: PRINTER (nyomtató) PRN = LPT1

LPTn: LOCAL PRINTER (helyi nyomtató)

n=1,2,3...hardver függő.

AUX: AUXILIARY (mellék-, itt kommunikációs csatorna)

COMn: COMMUNICATION (információ közlés) COM1 = AUX

Bármilyen eszköz, műszer lehetséges.

Az átirányítást végző jelek:

' > ' Nagyobb jel (GREATER THAN SIGN), a kimenetet irányítja át. Ha a kimenet fájl, akkor létrehozza azt, ha már van, akkor pedig felülírja az újonnan kijövő adatokkal.

' >> ' Kettő darab nagyobb jel (DOUBLE GREATER THAN SIGN), csak fájlok esetén van értelmezve. Már meglévő fájlhoz írja hozzá a kimenő adatokat. Ha nem létezik, akkor létrehozza.

' < ' Kisebb jel (LESS THAN SIGN), egy parancs bemenetére irányítja a fájl tartalmát. Csak fájlok esetén van értelmezve, mert bemenetként csak a fájlnev vagy a billentyűzet (ez az alapértelmezés) szerepelhet.

' | ' Vagy jel (PIPE - cső, itt adatfolyamot jelképez), kettő vagy több parancs adatait kapcsolja össze. Az elsőnek a kimenő adatait küldi a másik parancs bemenetére, majd az a feldolgozott adatokat tovább küldi. Maximum öt parancs kapcsolható így össze.

Példák:

```
DIR >PRN
DIR \also\*.txt >c:\levelek.lst
DIR \masodik\*.txt >>c:\levelek.lst
```

A DIR paranccsal listát készítünk az aktuális könyvtár fájljairól. A listát nem a képernyőre kérjük, mert például túl hosszú, és át szeretnénk tanulmányozni, hanem a nyomtatóra irányítjuk, és papírra írjuk ki. Utána az ELSO nevű alkönyvtárat listázzuk ki egy fájlba, ami még nem létezik, most hozzuk létre. (LEVELEK.LST) A MASODIK nevű alkönyvtár tartalmát hozzáírjuk az előbb létrehozott fájlhoz.

```
DATE <rovid.txt >>dolgozok. $$$
TIME <rovid.txt >>dolgozok. $$$
```

```
COPY CON rovid.txt
COPY CON LPT1
```

```
PROGRAM.EXE >COM2
PROGRAM.EXE <AUX >PRN
```

```
TYPE adatok.txt >LPT2
TYPE adatok.txt | SORT | MORE
SORT < adatok.txt > rendezve.dat
SORT <adatok.txt> rendezve.dat
SORT<adatok.txt>rendezve.dat
```

Batch-ben:

```
ECHO. | DATE >>dolgozok. $$$
ECHO. | TIME
```

A DATE parancs a dátumot várja bemenetként, vagy ha nem akarjuk megadni, akkor csak egy **[ENTER]** billentyűre vár. A ROVID.TXT csak egy 13-as kódú karaktert tartalmazzon. Akkor a parancs bemenetére irányított **[ENTER]** -t fogadja el, és a kimenetét (a dátum kiírását) pedig egy fájlba írjuk ki. Ha TIME paranccsal is ugyanígy járunk el, akkor kapunk egy pontos nyilvántartást a számítógép bekapcsolásainak vagy újraindításainak időpontjairól.

A COPY parancs egyik speciális formáját láthatjuk itt. Másolunk a konzolról (CON - bemenetkor a billentyűzet) a ROVID.TXT fájlba. A parancs kiadása után írjuk be a szöveget, amit a **[CTRL] + [Z]** billentyűkkel beírt fájl- végjellel (END OF FILE - EOF) fejezhetünk be. Utána **[ENTER]**-t nyomunk, és ekkor másolja át a DOS a beírt szöveget a fájlba, amit az "1 file(s) copied." üzenettel fog a tudomásunkra hozni.

Ha az előző példán egy kicsit változtatunk, akkor közvetlenül a nyomtatóra is írhatunk. Pontosabban a másolás nem fájlba megy, hanem a nyomtatóra (LPT1).

Egy tetszőleges programot indítunk, aminek a kimenő adatait a COM2 -es kimenetre irányítjuk, ahová egy másik számítógép vagy egy speciális berendezés csatlakozik.

Ez a példa is hasonló az előzőhöz, csak itt az **AUX** -ról kapja a bemenő adatokat a programunk, és a nyomtatóra írja ki az eredményeket (Az **AUX** vagy **COM1** lehet például egy egér /**MOUSE**/).

Az **ADATOK.TXT** állomány tartalmát íratjuk ki az **LPT2** -es nyomtatóra. Hasonló eredményre jutnánk a **COPY ADATOK.TXT LPT2** paranccsal is.

A parancsok sorbafűzésére láthatunk egy példát, ahol az **ADATOK.TXT** sorai a **TYPE** parancs kimenetéről a **SORT** sorbarendező program bemenetére kerül. Ezt a **SORT** parancs sorbarendezi (felparaméterezhető!). Ennek a kimenetét átirányítjuk a **MORE** parancs bemenetére, ami képernyőoldalakra tagolja a kiírandó szöveget. Ha megtelik a képernyő oldala, akkor egy tetszőleges billentyű leütésére vár, majd folytatja a kiírást.

A **SORT** parancs másik formája, hogy bemenetére kell küldeni a sorbarendezendő fájlt, a kimenetet pedig egy fájlba irányítjuk. A szóközöket ebben az esetben tetszőlegesen osztjuk el, mert a '**< >**' jelek nem szerepelhetnek a fájlnevében, így ott automatikusan elválasztja a paramétereket a DOS.

VIGYÁZAT! Rejtélyesnek tűnő hibára adhat okot, ha a '**< >**' jeleket sormintának használjuk.
Például:

ECHO Üss <ENTER> vagy [SPACE] billentyűt, ha kész!

Látszólag ártalmatlan kiírás, az **[ENTER]** billentyű neve '**< >**' jelek közé van téve, a szóközé (**[SPACE]**) pedig '**||**'. Míg a második esetben semmi hiba nincs, az "ENTER" szó az **ECHO** bemenetére van irányítva, így annak fájlnevének kellene lennie, a kimenet pedig a **VAGY** nevű fájlba van irányítva. Az **ENTER** nevű fájl persze nem létezik, mert nem is arra szántuk, de mégis hibaüzenetet kapunk válaszul.

A bejelentkező batch-fájlokban használható a következő két parancs. Az első a dátumot jegyzi fel egy fájlba, hasonlóan 2. példában látottakhoz. Ugyanis az **ECHO** parancs speciális formája az '**ECHO.**' ami egy **[ENTER]** -rel (szövegben soremeléssel) egyezik meg. Itt a **DATE** parancs bemenetére irányítjuk az '**ECHO.**' kimenetét, és a **DATE** parancs kimenetét a '**DOLGOZOK. \$\$\$**' nevű fájlba.

A **TIME** paranccsal is hasonlóképpen járunk el, csak itt a kimenet a képernyő lesz. Vagyis bejelentkezéskor kiírja a pontos (?) időt (azoknál a számítógépeknél, ahol a dátum és az idő tárolását tartós elem biztosítja).

Az adatok nyomtatóra irányításának van egy másik módja is a DOS -ban. A **[CTRL]+[P]** billentyűk lenyomásával bekapcsoljuk a "két eszközre írás" funkciót. Ekkor ugyanis a képernyőre kerülő adatok a nyomtatón is megjelennek. Ennek az állapotnak a **[CTRL]+[P]** újbóli megnyomásával vethetünk véget. Akkor viszont, ha nincs a számítógépünkhöz az **LPT1** -re nyomtató kötve, akkor kerülhetünk bajba. Ugyanis a képernyőre menő összes bájtot megpróbálja a nyomtatóra is kiírni, de ez nem sikerül neki. Ekkor adja a :

```
Write fault error device PRN
Abort, Retry? _
```

```
"A PRN eszközön írási hiba van
Megszakítás, Újra? _"
```


Erre a kérdésre az **[A]** és **[R]** billentyűkkel válaszolhatunk, amit viszont szintén meg akar jeleníteni a DOS a nyomtatón. Ez továbbra sem sikerül neki, így továbbra is kapjuk a hibaüzeneteket, akármit is válaszolunk rá. Szabadulást az ördögi körből a **[CTRL] + [P]** újbóli megnyomása jelent. Ezután már az **[A]**, mint "Abort" billentyűvel kiszálhatunk a végtelennek tűnő hibaüzenet-áradatból.

Az adatok nyomtatóra irányításának másik módja, a képernyő tartalmának a nyomtatóra küldése. Kiíratunk valamit a képernyőre, majd a **[PRINT SCREEN]** megnyomásával a teljes szöveges képernyő tartalma kiíródik a nyomtatóra is. Ha grafikus ábrát akarunk megjeleníteni a nyomtatón, akkor be kell tölteni először a GRAPHICS.COM-ot, és azután van rá lehetőségünk. Itt viszont nem kerülünk csapdába, ha nincs nyomtatónk, mert erre egy sípoló hang figyelmeztet bennünket.

Az állományok megjelenítése

TYPE

Kiírja az állományok tartalmát

TYPE [*meghajtó:*] [*elérési_út*] *fájlnev*

A megadott fájl tartalmát írathatjuk ki vele.

Belső parancs.

meghajtó: *elérési_út fájlnev* A kiírandó állomány neve, és helye a könyvtárfán. A helyének a meghatározásakor az maradhat el, amelyik az aktuálissal helyettesíthető. (Ha az aktuális meghajtó, aktuális könyvtárban van a fájl, akkor csak a fájl nevét kell megadni.) A fájlnev megadása kötelező, enélkül a parancsnak nincs értelme.

A megadott fájl tartalmát írja ki alapértelmezésben a képernyőre. A kimenet (a fájl tartalma) átirányítható más eszközre.

Ha a kiírandó szöveg túl hosszú, akkor lefut a képernyőről, és csak akkor áll le a képernyő görgetése, ha kiírta az utolsó bájtot is. Ezen a problémán több módon is segíthetünk:

- Használjuk a **[CTRL] + [S]** vagy a **[PAUSE]** billentyűket. (Ebben az esetben gyorsan kell megnyomni a parancs kiadása után, mert különben lefut a képernyőről a szöveg eleje.)
- Használjuk a **MORE** parancsot, ami oldalakra tördeli a szöveget, és minden oldal aljára kiírja a "**-MORE-**" feliratot, és egy billentyű leütésére vár, majd megint kiír egy oldalt.
- A kimenetet nyomtatóra irányítjuk, így a kinyomtatott szöveget kényelmesen elolvashatjuk, kijavíthatjuk a hibákat, és akkor fordulunk ismét a számítógéphez, amikor már tudjuk mit kell csinálnunk. A szemünket is jobban kíméljük így, és egy hosszabb szöveg is jobban áttekinthető, mint a képernyőn.
- A **TYPE** parancs kimenetének fájlba irányítása nem más, mint a fájl megduplázása, egy **COPY Régi_név Új_név** paranccsal azonos.
- Ha a kimenet olyan típusú szöveg, ami sorberendezhető, akkor a **SORT** paranccsal már sorberendezve íratjuk ki a fájlt monitorra, nyomtatóra, esetleg fájlba. Ebben az esetben nem csak a fájl megduplázásáról van szó, hanem a rendezett formájának egy új név alatti elmentéséről.

Szűrő parancsok

Azokat a parancsokat nevezzük *szűrő parancsoknak*, amelyeknek a bemenetére irányított fájlban valamilyen szűrést vagy átalakítást végeznek, és az eredményeket a kimenetükre adják ki. Az ilyen parancsoknak megadható az, amit a bemenetére irányítunk, és az a hely, ahová a kimenetét irányítjuk. Mind a kettőnél az alapértelmezett hely a CON eszköz. Azaz *bemenetnél a billentyűzet, kimenetnél a képernyő*. Itt a bemenetnek a billentyűzetet használni nem sok értelme van, így az általában egy állomány. A kimenetet viszont, ha nem irányítjuk át, akkor a képernyőre jön.

A parancsok ismertetésénél a kimenetet nem adjuk meg, bemenetnek pedig egy állományt feltételezünk.

Ha a szűrő parancsok használatánál a ' | ' (PIPE) parancssatolót használjuk, akkor be kell állítani a **TEMP környezeti változót**. Meg kell adni benne egy könyvtár nevét és helyét. Ide fog szűréskor egy ideiglenes állományt létrehozni, amit a parancs végrehajtása után le is töröl. (Ezt meg is nézhetjük, ha az UNDELETE parancssal helyreállítjuk az állományt.)

MORE

Képernyőoldalakra tördelés

MORE < [*meghajtó*] [*elérési_út*] *fájlnev*
parancs | MORE

A megjelenítendő állományt oldalakra tagolva írja ki.
 Külső parancs.

meghajtó: *elérési_út fájlnev* A kiírandó állomány neve, és helye a könyvtárfán. A helyének a meghatározásakor az maradhat el, amelyik az aktuálissal helyettesíthető. (Ha az aktuális meghajtó, aktuális könyvtárban van a fájl, akkor csak a fájl nevét kell megadni.) A fájlnev megadása kötelező, enélkül a parancsnak nincs értelme.

A parancs a bemenetére irányított fájl oldallakra tördeli, és csak egy oldalnyit jelenít meg a kívánt eszközön. Utána vár egy billentyű leütésére, majd folytatja tovább a kiírást. A billentyűre akkor is vár, ha a kimenetét tovább irányítjuk nyomtatóra, vagy fájlba. Bár ezekben az esetekben nem célszerű használni ezt a parancsot.

SORT

Sorbarendezés

SORT [/R] [/+N] [< *forrás*] [> *cél vagy* | *szűrő*]
 [*parancs* |] SORT [/R] [/+N] [> *cél vagy* | *szűrő*]

A bemenetére érkező adatokat, a megadottak szerint sorbarendezi.
 Külső parancs.

/r: Fordított sorrendben rendezi a bemenetére érkező adatokat. A rendezést az **ASCII kódtáblázat** szerint végzi, és nem a magyar ábécé szerint.

/+n: A rendezést a megadott (pozitív egész számú) karaktertől kezdi. Majd a példákon keresztül láthatjuk mi az előnye ennek a paraméternek.

< *forrás:* [*meghajtó:*] [*elérési_út*] *fájlnev*

A fájl neve és helye a könyvtárán, amelyiknek a sorait sorba szeretnénk rendezni. Forrásként nem szoktunk mást megadni. Lehet még elvileg külső (COMX) csatornáról is várni az adatokat, de ez nem jellemző. Ha nem adunk meg semmit forrásként, akkor a billentyűzetről várja az adatokat. Hasonló módon mint a COPY CON *fájlnev* parancsban, a bevitt adatokat le kell zárni [CTRL] + [Z] fájlvégjellel.

> *cél:* a/ [*meghajtó:*] [*elérési_út*] *fájlnev*
 b/ Külső egység: PRN vagy LPTX vagy COMX
 c/ Alapértelmezésben - a CON egység - a monitor.
 (Akkor, ha nem adjuk meg egyiket sem.)

| *szűrő:* Egy másik szűrő parancsot adunk meg a kimenet további feldolgozása céljából. (Például: |MORE) Erről bővebben a példákban olvashattok.

A parancs másik megadási formája a jellemzőbb. Egy parancs vagy külső program kimenetét csatoljuk rá a SORT bemenetére, és a rendezett adatokat a ' cél ' -ra küldjük. Ilyen parancs lehet például a TYPE , FIND , DIR .

De használhatunk olyan parancsokat, programokat is, amelyek adatokat bocsájtanak a kimenetükre. A többi szűrőnél is használhatjuk az előző parancsokat, kiegészítve a SORT paranccsal a felsorolást.

Példák:

```
D:\DOS>type text.txt
retes .lap
piskota .lap
torta .lap
adatok .dat
holnap .dat
mai .dat
run .exe
sorter .com
temp .$$$
szoveg .txt

D:\DOS>_
```

```
D:\DOS>type text.txt |sort
adatok .dat
holnap .dat
mai .dat
piskota .lap
retes .lap
run .exe
sorter .com
szoveg .txt
temp .$$$
torta .lap

D:\DOS>_
```

Állomány megjelenítésének egyik egyszerű módja: Mindegyik példában a TEXT.TXT fájlt fogjuk használni. A kimenetet nem adtuk meg, így az alapértelmezésben a képernyő lesz. Használhatnánk a MORE parancsot is példaként, de ez a szöveg nincs 24 sor se. (Te keress a merevlemezeden olyan TXT vagy DOC kiterjesztésű fájlt, aminek a hossza nagyobb mint 3000 bájt. Ezzel ki tudod majd próbálni: TYPE *fájlnev* | MORE) Ha nem találsz, akkor írd be:

```
C:\>DIR \ /S >konyvtar. $$$
C:\>TYPE konyvtar. $$$ | MORE
C:\>MORE < konyvtar. $$$
```

A ' DIR \ /S ' parancs ad olyan hosszú listát, hogy csak több képernyő oldalra fog csak kiférni.

Most nézzük meg ezt az állományt sorba-rendeve: Az első karaktertől kezdve rendeztük sorba (név szerint).

```
D:\DOS>type text.txt |sort/r
torta .lap
temp .$$$
szoveg .txt
sorter .com
run .exe
retes .lap
piskota .lap
mai .dat
holnap .dat
adatok .dat

D:\DOS>_
```

Most ugyanez a listát nézzük meg fordított sorrendben. A /r paraméter hatására a rendezettség az *ASCII kódtábla* szerint hátról előre irányban értendő.

```
D:\DOS>type text.txt|sort /+10
sorter .com
adatok .dat
holnap .dat
mai .dat
run .exe
retes .lap
piskota .lap
torta .lap
szoveg .txt
temp .$$$

D:\DOS>_
```

Rendezhetjük a fájlunk adatait kiterjesztés szerint is. Olyan adatállományoknál, ahol az adatok adott pozíción találhatók, ott alkalmazhatjuk ezt a módszert. A /+10 paraméterrel adjuk meg, hogy a tizedik pozíciótól kezdve rendezze az állomány rekordjait.

```
D:\DOS>type text.txt |sort /+10 /r
temp .$$$
szoveg .txt
torta .lap
piskota .lap
retes .lap
run .exe
mai .dat
holnap .dat
adatok .dat
sorter .com

D:\DOS>_
```

Most az előző módszert alkalmazva rendezzük az állományunkat kiterjesztés szerint fordított sorrendbe.

```
D:\DOS>sort /r /+10 <text.txt
temp .$$$
szoveg .txt
torta .lap
piskota .lap
retes .lap
run .exe
mai .dat
holnap .dat
adatok .dat
sorter .com

D:\DOS>_
```

A **SORT** parancs másik megjelenési formája is ugyanazt az eredményt szolgáltatja. Előre került a **SORT** parancs, utána jön a bemenetére irányított fájl megadása. Ilyenkor nem hoz létre ideiglenes állományt, így egy teljesen betelt merevlemezen is használhatjuk a parancsot, míg az előző esetekben nem. Hasznos ez az alkalmazás hálózati környezetben is, ahol például előfordulhat, hogy nincs sehol írási jogunk a merevlemezre. Olyankor csak ez a módszer használható.

FIND

Keresés fájlban

FIND [/V] [/C] [/N] [/I] "karakterek" [forrás(ok)] [>cél]

parancs | **FIND** [/V] [/C] [/N] [/I] "karakterek" [>cél]

A forrásban megkeresi a megadott karaktereket.

Külső parancs.

/v: Kiírja azokat a sorokat, amelyek NEM tartalmazzák a keresett karaktersort.

/c: A keresett karaktersor előfordulási számát írja ki.

/n: Kiírja azokat a sorokat, amelyekben megtalálta a keresett karaktersort, annak sorszámával együtt.

/i: Nem tesz különbséget a kereséskor a kis- és a nagy- betűk között.

"karakterek" Az a karaktersor amelyet keresünk a megadott forrásban.

forrás: [meghajtó:] [elérési_út] fájlnev

Annak a fájlnak, vagy azoknak a fájloknak a neve és helye a könyvtárfán, amelyekben keressük a megadott karakterersorozatot. Forrásként nem szoktunk mást megadni. Lehet még elvileg külső (COMX) csatornáról is várni az adatokat, de ez nem jellemző. Ha nem adunk meg semmit forrásként, akkor a billentyűzetről várja az adatokat. Hasonló módon mint a COPY CON fájlnev parancsban, a bevitt adatokat le kell zárni [CTRL] + [Z] fájlvég jellel.

>cél: a/ [meghajtó:] [elérési_út] fájlnev

b/ Külső egység: PRN vagy LPTX vagy COMX

c/ Alapértelmezésben - a CON egység - a monitor.

(Akkor ha nem adjuk meg egyiket sem.)

|szűrő: Egy másik szűrő parancsot adunk meg a kimenet további feldolgozása céljából. (Például: |SORT) Erről bővebben a példákban olvashattok.

Példák:

```
D:\TEMP>type text.txt
retes .lap
piskota .lap
torta .lap
holnap .dat
mai .dat
adatok .dat
run .exe
sorter .com
temp .$$$
szoveg .txt
date_uj .txt
adatok .doc

D:\TEMP>_
```

Először nézzük meg a 'TEXT.TXT' fájl tartalmát, amiben a kereséseket fogjuk folytatni.

```
D:\TEMP>find
FIND: Required parameter missing

D:\TEMP>_
```

A parancsot, ha paraméterek nélkül adjuk ki, akkor kiírja, hogy:

"Szüksége van a hiányzó paraméterre"

```
D:\TEMP>find "doc" /c
0

D:\TEMP>_
```

Ha a kapcsolókat nem a keresendő karaktersor előtt adjuk meg, akkor kicsit furcsán működik a parancs. Akkor úgy veszi, hogy nincs bemenő adat, így abban nem is talál semmit. (0)

```
D:\TEMP>find /c "doc"
document.doc
hali. $$$
tartalom.doc
^Z
2

D:\TEMP>_
```

Csak azt írja ki, hányszor találta meg a keresett "doc" karaktersort. A bemenő adatok helyét nem adtuk meg, így azt alapértelmezésben a (CON) billentyűzetről fogja várni. Most egy kicsit elgondolkozhatunk azon, hogy ennek meg mi az értelme. Ha egyszer úgy kell begépelni a szöveget, akkor látjuk, hogy hol van benne a keresett 'doc'. A szöveg bevitele egyébként, hasonlít a 'COPY CON ...' formára, és a fájl végét itt is [CTRL] + [Z] -vel (^Z) kell lezárni. Az [ENTER] után kiírja, hogy kétszer találta meg a keresett szövegrészt.

```
D:\TEMP>find "doc"
document.txt
document.txt
hali. $$$
tartalom.doc
tartalom.doc
^Z

D:\TEMP>_
```

Ha elhagyjuk a '/c' paramétert, akkor nem azt írja ki, hogy hányszor találta meg, hanem azokat a sorokat, amelyekben megtalálja a keresett szövegrészt. Ezért ha ugyanazokat a példákat adjuk meg mint az előbb, akkor a sor bevitele után azonnal kiírja még egyszer a sort, ha megtalálta benne a keresendőt.

Most a következőkben ugyanazokat a példákat mutatom be, csak a paraméterek fognak változni, így jól elkülöníthetők az egyes paraméterek hatásai.

```
D:\TEMP>find /c "dat" text.txt

----- TEXT.TXT: 5

D:\TEMP>_
```

A megadott fájlban a keresett karaktersor előfordulási számát adja meg. Kiírja a fájl nevét és mellé a számot.

```
D:\TEMP>find /v "dat" text.txt
```

```
----- TEXT.TXT
retes .lap
piskota .lap
torta .lap
run .exe
sorter .com
temp .$$$
szoveg .txt
```

```
D:\TEMP>_
```

Kiírja azon sorokat, amelyekben nincs benne a keresett karaktersor.

```
D:\TEMP>find /n "dat" text.txt
```

```
----- TEXT.TXT
[4]holnapi .dat
[5]mai .dat
[6]adatok .dat
[11]date_uj .txt
[12]adatok .doc
```

```
D:\TEMP>_
```

Kiírja annak a sornak a számát, amelyikben a keresett karaktersort megtalálta, és utána sor tartalmát.

```
D:\TEMP>find /i "DAT" text.txt
```

```
----- TEXT.TXT
holnapi .dat
mai .dat
adatok .dat
date_uj .txt
adatok .doc
```

```
D:\TEMP>_
```

Most vegyük észre, hogy a 'DAT' csupa nagybetűvel van írva, így azt nem lenne szabad kiírnia mint találatot. Viszont a '/i' paraméter arra utasítja, hogy ne tegyen különbséget kis- és nagybetű között.

```
D:\TEMP>find "DAT" text.txt
```

```
----- TEXT.TXT
```

```
D:\TEMP>_
```

Az előző példát igazolja ez a példa, most nem adtuk meg a '/i' paramétert, így nem is talál a fájlban nagybetűs 'DAT' karaktersort.

```
D:\TEMP>find "dat" text.txt
```

```
----- TEXT.TXT
holnapi .dat
mai .dat
adatok .dat
date_uj .txt
adatok .doc
```

```
D:\TEMP>_
```

Viszont ha kisbetűvel írjuk be a keresett karaktersort, akkor már meg fogja találni.

```
D:\TEMP>type text.txt | find "dat" | sort /r >prn
```

```
mai .dat
holnap .dat
date_uj .txt
adatok .doc
adatok .dat
```

```
D:\TEMP>_
```

(Ha nincs nyomtatónk, akkor irányítsuk egy állományba.)

Most bonyolítsuk meg egy kicsit a parancsot: A **TYPE** parancssal kiíratjuk a **'TEXT.TXT'**-t. A kimenő adatokat, rácsatoljuk a **FIND** parancsra, ami már csak azokat a sorokat fogja kiírni, amelyekben megtalálja a **'dat'** karaktersort. Ezt a kimenetet tovább kapcsoljuk a **SORT** parancs bemenetére. A **'/r'** paraméter utasítja arra, hogy a bejövő adatokat fordított sorrendben írja ki. Most végre az összes olyan sort megkaptuk fordított sorrendbe szedve, amelyekben szerepel a **'dat'** szócska. A kapott eredményeket legvégül a nyomtatóra irányítjuk.

Az állományok megjelenítése nyomtatón

Már több módszert is láttunk arra, hogyan lehet állományokat nyomtatón megjeleníteni. Ezek a módszerek az adatok nyomtatóra irányításán alapultak. Vannak olyan parancsok is, amelyek kifejezetten a nyomtatást hivatottak elősegíteni. Ezek a **PRINT**, és a **GRAPHICS**.

A billentyűzeten pedig a **[PRINT SCREEN]** billentyű szolgál az aktuális képernyőtartalom kinyomtatására.

A nyomtatással ebben a könyvben, csak a DOS -t érintő részével foglalkozunk, a nyomtatók vezérlésével, karakterkészletek, nyomtatási formák állításával, a *"DOS alapismeretek II."* című könyvben olvashattok. A nyomtatások megismerésének azért olyan nagy a jelentősége, mert az egész számítógépes munkánkra akkor tesszük fel a koronát, amikor a munkánk eredményeit papíron megörökítjük. A papírra kinyomtatott levelek, tablók, számítási eredmények megjelenését már egy egyszerű olcsó nyomtatón is változatosan tudjuk megjeleníteni, ha van némi kis jártasságunk a nyomtatók kezelésében. Egy levélnél az *esztétikus megjelenés* az elsődleges, míg egy számítási eredményeket tartalmazó tábló kinyomtatásakor a *nagyobb nyomtatási sebességre* való törekedés a fontosabb. A nyomtatás sebessége és az esztétikai megjelenése egymással fordítottan arányosak, az egyik csak a másik rovására mehet.

A DOS -ban a nyomtatás nincs elkényeztetve jobbnál jobb parancsokkal. Sajnos elég szegényes a kínálat a külső egységek kezelését illetően. Viszont már egy programnyelv alapfokú megismerése után sok hasznos, kis programcskával tudjuk a DOS hiányosságait pótolni.

PRINT

Nyomtatás

PRINT [/D:eszköz] [/B:pméret] [/U:jel_1] [/M:jel_2] [/S:jel_3] [/Q:darab] [/T] [/C] [/P]
 [[meghajtó:] [elérési_út] fájlnev] [...]

Állományok kinyomtatását végzi a háttérben.

Külső parancs.

/d:eszköz A külső nyomtató eszköz (DRIVE) nevét adhatjuk meg, ami lehet: LPTx , COMx , AUX , alapértelmezésben LPT1 (PRN). A parancs első kiadásakor kell megadni, ha elmarad, akkor bekéri a számítógép, és ez után a nyomtatásoknál már nem lehet megadni ezt a paramétert.

/b:pméret A belső puffer (BUFFER) mérete bájtokban, amelyben a nyomtatás előtt közvetlenül az adatokat tárolja. Ha nem adjuk meg, akkor az alapértelmezésbeli értéke 512 bájt, a megadható maximum 16384 bájt. Ha nagy a puffer mérete, akkor a nyomtatás, a közben folyó munkától függetlenül gyorsan fog végbe menni, és az a 16 kilobájt puffer nem foglal túl nagy memóriát.

/u:jel_1 Megadhatjuk a maximális várakozási óraciklusok számát. Ha a nyomtató foglalt a megadott ideig, akkor a munka nem kerül nyomtatásra. Alapértelmezésben 1 maximum 255 lehet.

/m:jel_2 Megadhatjuk vele egy karakter kinyomtatásához szükséges maximális óraciklusok számát 1-255 -ig, alapértelmezésben az értéke 2.

/s:jel_3 Megadhatjuk, hogy hány óraciklusonként foglalkozzon a DOS a nyomtatással és a végrehajtandó feladatokkal. Alapértelmezésben 8, a megadható értéke 1-255 -ig tart. Ha növeljük az értékét, akkor gyorsul a nyomtatás, és az éppen aktuális feladat végrehajtása pedig lassul.

Az előző három paraméterben a megadható értéket próbáljuk ki több változatban is, és csak akkor alkalmazzuk "éles" -ben, amikor már biztosak vagyunk azok használatában. Ugyanis ha olyan paraméterértékeket adunk meg, ami túlzottan lelassítaná a munkánkat a nyomtatás javára olyankor, amikor a nyomtatás nem is olyan sürgős, akkor már azon változtatni, csak körülményesen lehet. Fordított esetben sem válik a javunkra a parancs használata.

/q:darab Megadhatjuk a nyomtatási sorba (QUEUE) kerülő fájlok számát 4-32 -ig. Alapértelmezésben ez 10.

Az előbb ismertetett paramétereket - /d /b /u /m /s /q - csak a parancs első kiadásakor lehet megadni. Ha nem adjuk meg - a /d kivételével - , akkor az alapértelmezésbeli értéküket fogják felvenni. Egyszerű esetben a parancs használatához nem szükséges megadni a paramétereket, el lehet fogadni a megadott alapértelmezésbeli értékeket. (A parancs egyszerű használata: **PRINT level1.txt**)

/t: Törli a nyomtatási sorból az összes nyomtatási munkát, a nyomtatás alatt állót is.

/p Nyomtatási munkákat fűz a nyomtatási sorba. Nem kell használni, általában csak akkor, ha egy paranccsal akarunk a nyomtatási sorba fájlokat fűzni és kitörölni.

/c Nyomtatási munkákat töröl a nyomtatási sorból. A kapcsolót a kitörölni kívánt fájl neve előtt és után is lehet használni.

A **/p** és **/c** paraméterek együttes használata esetén, olyan jól meg lehet bonyolítani a paraméterek használatát, hogy mire kiigazodunk rajta a nyomtató végez az összes nyomtatási feladattal. Célszerűbb előbb törölni a törlendő állományokat, és utána hozzáfűzni a kinyomtatandó újabb állományokat. Mind a két paraméterre egyaránt áll az, hogy például a **/c** paraméter, az előtte közvetlenül álló fájlra érvényes, és az azt követő összes fájlra, addig amíg egy **/p** paramétert nem talál, de a **/p** az előtte álló fájlra is érvényes már. Ha valaki ennek ellenére mégis egy parancsban akarja használni mind a két paramétert, akkor most ismertetem néhány példán keresztül:

```
D:\TEXT>print /t
Errors on list device indicate that it
may be off-line. Please check it.
PRINT queue is empty

D:\TEXT>_
```

Az első parancsban töröljük a nyomtatási sorban lévő munkákat. Viszont jelentkezett is egy gyakori hiba:

"Hibát észlelek a nyomtató meghajtón, ki van kapcsolva. Kérem változtasson rajta. A nyomtatási sor üres."

Az első üzenetet azért küldte, mert nem volt a nyomtató az adatok fogadására alkalmas, bekapcsolt (ON-LINE) állapotban. A másik üzenet arra vonatkozott, hogy a nyomtatási kéréseket - ha voltak - kitörölte a sorból, és a jelenlegi állapotban a sorban nincs kérés, vagyis üres.

```
D:\TEXT>print header.$$1 nc.mnu

D:\TEXT\HEADER.$$1 is currently being printed
D:\TEXT\NC.MNU is in queue

D:\TEXT>_
```

Két fájlt ki akarunk nyomtatni. Beteszi a kéréseket a nyomtatási sorba, és az elsőt elkezdki kinyomtatni.

```
D:\TEXT>print header.$$1 /c dosko.txt bevezeto.txx /p
File not in PRINT queue - D:\TEXT\DOSKO.TXT

D:\TEXT\NC.MNU is currently being printed
D:\TEXT\BEVEZETO.TXX is in queue

D:\TEXT>_
```

Most a nyomtatás alatt álló 'HEADER.\$\$1' és a 'DOSKO.TXT' nevű fájlok törlésére, és a 'BEVEZETO.TXX' fájl nyomtatási sorba való állítására adtunk parancsot. A 'D:\TEXT\DOSKO.TXT' fájl viszont nem létezik (FILE NOT ...), így azt nem tudja a nyomtatási sorba állítani. Miután elvégezte a kiadott műveleteket, a nyomtatási sorban lévő fájlokról ad egy listát. Az első kérés áll nyomtatás alatt.

```
D:\TEXT>print nc.mnu /c dosko.ttt /p
D:\TEXT\BEVEZETO.TXX is currently being printed
D:\TEXT\DOSKO.TTT is in queue
D:\TEXT>_
```

A mostani parancsunk kissé ellentmondásosnak tűnik, hisz a /c paraméter hatása a közvetlen előtte állóra él, és minden utána következőre addig, amíg nem talál egy /p paramétert. Akkor viszont az előtte álló már a /p -hez tartozik. Nos, a definíció szerint, a 'DOSKO.TTT' mindkét paraméter hatása alatt áll. Ebben az esetben a hátrébb álló paraméter felülírja az előző hatását, vagyis az említett fájlt a sorba állítjuk, az 'NC.MNU' -t pedig törölni akarjuk onnan.

```
D:\TEXT>print /c dosko.ttt bevezeto.txx doskonyv.doc /p
D:\TEXT\DOSKONYV.DOC is currently being printed
D:\TEXT>_
```

A mostani parancsunkkal pedig töröljük az 'DOSKO.TTT'-t, a 'BEVEZETO.TXX'-et, de a 'DOSKONYV.DOC' -ra már a /p paraméter fejti ki a hatását, tehát azt a sorba beállítjuk.

```
D:\TEXT>print header.$$1 nc.mnu
D:\TEXT\DOSKONYV.DOC is currently being printed
D:\TEXT\HEADER.$$1 is in queue
D:\TEXT\NC.MNU is in queue
D:\TEXT>_
```

A mostani parancsunkkal két újabb fájlt fűzünk a nyomtatási sorba, hogy további a parancsokat legyen hol bemutatni.

```
D:\TEXT>print /p nc.mnu bevezeto.txx doskonyv.doc /c
D:\TEXT\HEADER.$$1 is currently being printed
D:\TEXT\NC.MNU is in queue
D:\TEXT\NC.MNU is in queue
D:\TEXT\BEVEZETO.TXX is in queue
D:\TEXT>_
```

Itt ismét a /p és /c paraméterek használatát bonyolítjuk: A /p után álló fájlokat állítjuk nyomtatási sorba, amelynek addig tart az érvényessége, ameddig nem talál egy /c -t. Akkor viszont az előtte álló fájl már annak a hatása alatt áll. Az eredményekből láthatjuk, hogy az NC.MNU -t és a BEVEZETO.TXX -et a sorba állítottuk, a DOSKONYV.DOC-t pedig töröltük a nyomtatási sorból.

Most pedig nézzük meg az elsőként kiadandó **PRINT** parancs javasolt formáját, paraméterezését:

```
C:\>PRINT /d:PRN /b:16384 /s:16 /q:32
```

/d:PRN Adjuk meg a használni kívánt nyomtató helyét.
(Bekéri nyomtatás előtt, ha nem adjuk meg.)

/b:8192 Ha a parancsnak mindig sikerül megtölteni ezt a puffert, akkor még a lemezműveletek alatt is folyamatosan fog tudni nyomtatni. Ha ugyanis kiürül a puffer, akkor olvasni szeretné a nyomtatás alatt álló fájlt a lemezről. Viszont mi, közben olyan programot használunk, ami gyakran végez a lemezen műveleteket, akkor a **PRINT** parancs nem tudja a további adatokat olvasni, és a nyomtatás akadozva fog menni. A javasolt érték 16 kilobájt, ami elég kis terület a memóriában, de mégis gyorsan megy így a nyomtatás. Általánosságban elmondható, hogy a rezidens programoknál mindig azt kell elsősorban mérlegelni, hogy mennyi helyet foglal a memóriában, és ki lesz-e használva kellőképpen.

/s:16 Akkor tanácsos ezt az értéket használni, ha az előbb említett eset állna fenn. Egyébként az alapértéket lehet használni.

/q:32 Az előbbi okfejtést itt is alkalmazhatjuk, mert a sorban álló fájloknak csak a neve, helye és néhány adata van tárolva a nyomtatási sorban, így azok még nagy számban sem foglalnak nagy helyet. Éppen ezért nem szükséges spórolni vele.

Nem adtam meg a parancsban a **/u:jel_1** és a **/m:jel_2** paramétereket. Az oka az, hogy ezek alapértelmezett értékei mellett is, egy tetszőleges típusú EPSON nyomtató, hiba nélkül tud nyomtatni. Ha valaki nagyon régi elavult nyomtatót használ, ami jóval kisebb nyomtatási sebességgel dolgozik, akkor lehet ezeken az értékeken változtatni. De akkor is próbálgatással keressük meg azt az értéket, ami úgy-ahogy megfelel a nyomtatónak.

ÖSSZEFOGLALVA AZ EDDIG LEÍRTAKAT:

Először használjuk mindig a **PRINT** első megadására szolgáló parancsot, amit az **'AUTOEXEC.BAT'** fájlba elhelyezhetünk, így nincs rá gond minden rendszerinduláskor. A **'CONFIG.SYS'**-be is tehetjük, de ennek megadását csak a könyv második része tartalmazza majd.

Amikor nyomtatásra használjuk a **PRINT** parancsot, akkor előbb adjuk meg a sorból kitörendő fájlkat a **/c** paraméter után, majd egy újabb paranccsal adjuk meg a sorba állítandó fájlkat listáját. Ha követjük ezt az elvet, akkor nem lesz gondunk a fájlkat nyomtatásával, és áttekinthetőbbek lesznek a parancsaink, miáltal kevesebbet fogunk hibázni.

A parancs megadásánál olvasható, hogy a nyomtatás a háttérben zajlik. Ez gyakorlatilag azt jelenti, hogy a nyomtatási kérés egy nyomtatási sorba kerül, és onnan a megszakításon keresztül kerül kinyomtatásra, miközben mi úgy dolgozunk a számítógépünkön, mintha nem is a mi gépünk vezérelné a nyomtatást.

GRAPHICS

Grafikus képernyőnyomtatás

GRAPHICS [*típus*] [*fájl_specifikáció*] [/R] [/B] [/LCD] [/PRINTBOX:STD | /PRINTBOX:LCD]

Grafikus és/vagy színes képernyő tartalmának kinyomtatását segíti elő.

Külső parancs.

típus: A használt nyomtató típusa, nem kötelező megadni, csak akkor ajánlott, ha olyan típusú nyomtatónk van, ami szerepel a felsorolásban. (Az EPSON tús nyomtatóknál nem kell megadni ezt a paramétert.)

A megadható típusnevek, és nyomtató típusok:

COLOR1	IBM PC Színes nyomtató fekete szalaggal.
COLOR4	IBM PC Színes nyomtató piros-zöld-kék szalaggal.
COLOR8	IBM PC Színes nyomtató cián-bíbor-sárga-fekete szalaggal.
HPDEFAULT	HP PCL sornyomtató
DESKJET	HP Deskjet nyomtató
GRAPHICS	IBM Personal Graphics Printer, Proprinter, Quitwriter nyomt.
GRAPHICSWIDE	IBM Personal Graphics Printer, széles kocsival.
LASERJET	HP Laser Jet.
LASERJETII	HP Laser Jet II.
PAINTJET	HP PaintJet.
QUITJET	HP QuitJet.
QUITJETPLUS	HP QuitJet Plus.
RUGGEDWRITER	HP RuggedWriter.
RUGGEDWRITERWIDE	HP RuggedWriter, széles kocsival.
THERMAL	PC -hez kapcsolható hőpapiros nyomtató.
THINKJET	HP ThinkJet.

fájl_specifikáció Eddig még nem használtam ezt a szakkifejezést, mert ez nem más mint egy angol szó magyar kiejtéssel leírva. Viszont szabatosan, tömören elnevezi a megadandó paramétert. Ami nem más, mint a *meghajtó: elérési út fájlnev* hármas.

A megszokott paraméterek, amelyek oly gyakran fordulnak elő a DOS -ban. A fájl, amit meg lehet adni, az pedig a nyomtatók adatait tartalmazza. Nem kötelező, alapértelmezésben a 'GRAPHICS.PRO' állományt keresi a DOS, az aktuális könyvtárban, vagy abban a könyvtárban, ahol a 'GRAPHICS.COM' állomány van.

/r (REVERSE - fordított) A képernyő tartalmát úgy nyomtatja ki, ahogyan az ott látszik, azaz sötét háttérrel, világos karakterekkel.

/b (BACKGROUND - háttér) Színes háttérrel használ. Csak a *COLOR4* és a *COLOR8* esetén használható a paraméter.

/lcd A képernyő kinyomtatásánál, az LCD szerinti 8 * 8 -as karakter arányokat használja, a CGA képernyő karakterarányai helyett.

`/p[rint]b[ox]:lcd` vagy `/p[rint]b[ox]:std`

A nyomtatáskor más és más karakterarányokat használ az *LCD* és az *STD* paraméterek esetén. Az első megadás azonos a */lcd* paraméterrel. Próbáljuk ki mindkét nyomtatást, és meg fogjuk látni a különbséget a két paraméter között.

Ne felejtjük el, hogy ez a program a grafikus képernyő kinyomtatását támogatja, és a **[SHIFT] + [PRINT SCREEN]** billentyűket kell megnyomni.

Ha egy 320 * 200 -as felbontású képernyőt nyomtatunk ki kétszínű nyomtatón, akkor az a nyomtatáskor a szürke három árnyalatát fogja használni.

Ha egy 640 * 200 -as, vagy ennél nagyobb felbontású képet nyomtatunk, akkor a nyomtatón az 90°-os elforgatással jelenik meg.

A *PostScript* típusú nyomtatón nem használható ez a funkció.

A parancs rezidensen bent marad a memóriában, így ott állandóan helyet foglal. Csak akkor aktiváljuk, ha olyan programot használunk, amelynél képernyőnyomtatást akarunk alkalmazni.

Bepillantás egy batch-fájlba

A parancsfájl (BATCH - köteg, kötegelt parancsok) nem más, mint belső és külső parancsok sorozata, amelyeket gyakran kell ugyanabban a sorrendben végrehajtani. Hogy ezeket a parancsokat ne kelljen minden egyes alkalommal beírni, összegyűjthetjük őket egy fájlba. Ez lesz a parancs fájl, a batch-fájl, aminek a kiterjesztése BAT. Ha beírjuk ennek a batch fájlnek a nevét, akkor a 'COMMAND.COM' soronként végrehajtja a parancsokat a fájl végéig.

Az egyik speciális batch-fájl az 'AUTOEXEC.BAT', aminek annak a meghajtónak a gyökérkönyvtárában kell lennie, amelyikről a rendszert indítjuk (Általában az A: vagy a C:). Ekkor a parancsfeldolgozó 'COMMAND.COM' megkeresi az említett meghajtó gyökérkönyvtárában, és ha megtalálja, akkor lefuttatja. Ha nem találja meg, akkor sem jelez hibát, hanem bekéri a dátumot, a pontos időt és kiírja a készenléti jel alapértelmezett formáját (C>_).

A batch-fájlok létrehozhatók a legegyszerűbb módon, a 'COPY CON prog.bat' paranccsal, vagy tetszőleges szövegszerkesztővel, például a DOS 'EDIT' paranccsal.

Vannak olyan belső parancsok, amelyeket általában csak batch-fájlokban szoktunk használni, vagy csak ott használhatók. Ilyenek például a CALL, ECHO, FOR, GOTO, PAUSE, REM, SHIFT, IF. Ezek részletes ismertetése, használatuk bemutatása már csak a "*DOS alapismeretek II.*" -ben fog napvilágot látni. Egyszerűek, könnyen megtanulhatók, és némi kis befektetéssel hasznunkra fordíthatjuk a batch-fájlokat.

Minimális programnyelv ismerettel pedig, akár újabb parancsokat is gyárthatunk magunknak, amelyek a mi igényeinknek megfelelő módon fognak működni. Ezeket is beírhatjuk a batch fájljainkba, hogy azokat jobba, komfortosabbá varázsolják.

Most egy lehetséges 'AUTOEXEC.BAT' fájl mutatok be, néhány általában előforduló paranccsal. A belső parancsok minden gépen hiba nélkül lefutnak, viszont a külső parancsokat csak akkor használjuk, ha a mi merevlemezünkön is megvannak. Az elérési útját pedig értelemszerűen a saját viszonyainkhoz kell igazítani.

1. Nézzük meg a TYPE paranccsal a fájl tartalmát.
2. Most futtassuk le úgy, hogy az első sora: @ECHO OFF
3. Most pedig használjuk az ECHO ON -t.

1) D:\TEMP>type autoexec.bat

```
a @echo off
b Prompt $p$g
c Path D:\dos\d5;d:\;c:\;D:\norton\nc;
d set comspec=c:\COMMAND.COM
e set dircmd=/a /o:gne /p
f d:\util\NET5
g graphics
h copy i:la.com e:
i ver
j echo. | date | find "is"
k echo. | time | find "is"
l I:Login
m d:
n rem c:\keyhunma
o norton\nc\nc
```

D:\TEMP>_

2) D:\TEMP>autoexec

```
f NetWare V3.10 - Workstation Shell for PC DOS V5.x
(C) Copyright 1990 Novell, Inc. All Rights Reserved.
NetWare Workstation Shell has already been loaded.
```

```
h 1 file(s) copied
```

```
i MS-DOS Version 5.00
```

```
j Current date is Fri 1993-03-26
```

```
k Current time is 9:06:08,27
```

```
l Enter your login name:
```

```
:
:
```

```
o The Norton Commander, Copyright (C) 1990, 91, 92, Peter Norton Computing, INC.
```

D:\TEMP>_

3) D>autoexec.bat

```
D>Prompt $p$g

D:\TEMP>Path D:\dos\d5;d:\;c:\;D:\norton\nc;

D:\TEMP>set comspec=c:\COMMAND.COM

D:\TEMP>set dircmd-/a /o:gne /p

D:\TEMP>d:\util\NET5
NetWare V3.10 - Workstation Shell for PC DOS V5.x
(C) Copyright 1990 Novell, Inc. All Rights Reserved.
NetWare Workstation Shell has already been loaded.

D:\TEMP>graphics

D:\TEMP>copy i:la.com e:
1 file(s) copied

D:\TEMP>ver

MS-DOS Version 5.00

D:\TEMP>echo. | date | find "is"
Current date is Fri 1993-03-26




D:\TEMP>echo. | time | find "is"
Current time is 9:10:00,99

D:\TEMP>I:Login
Enter your login name:
:
:
D:\TEMP>d:

D:\TEMP>rem c:\keyhunma

D:\TEMP>\norton\nc\nc
The Norton Commander, Copyright (C) 1990, 91, 92, Peter Norton Computing, INC.
D:\TEMP>_
```


Ajánlat az LSI Oktatóközpont számítástechnikai kiadványaiból

Dr Kovács Magda: 32 bites mikroprocesszorok 80386 I.	189.-
Móricz Attila: Novell hálózati alapismeretek I. Felhasználóknak	.-
Móricz Attila: Novell hálózati alapismeretek II. Rendszergazdáknak	.-
Bartha Gábor: ACT! for Windows	564.-
Boér-Dóra-Fenyő-Seres: Az IBM PC belső felépítése	499.-
Fekete-Nagy-Gregorics-Szenes:	
Bevezetés a Mesterséges Intelligenciába	274.-
Honti-Honti: CorelDRAW! 2.01 for Windows	399.-
Bartha-Honti: CorelDRAW! 3.0 for Windows	699.-
Bartha-Bartha: CorelDRAW! 3.0 a csapattagok	599.-
Bartha Gábor: CorelDRAW! 4.0 kézikönyv 	1018.-
Csábi-Dallos: DOS 5.0 és kapcsolat a Windows 3.0 -val	890.-
Dallos-Fábián-Zonbor: DOS 6 kézikönyv	809.-
Hargittai-Kaszanyiczki: A DOS titkai	541.-
Kaszanyiczki László: DR DOS 6.0 kapcsolat a Windows 3.0 -val	490.-
Kaszanyiczki László: Egér programozása	249.-
Dr kovács Magda: Egyszerűen a mikroszámítógépről	499.-
Klucs-Koleszár: Excel for Windows 3.0	443.-
Klucs-Koleszár: Excel for Windows 4.0	560.-
Klucs-László: Excel 4.0 lépésről lépésre 1	274.-
Klucs-László: Excel 4.0 lépésről lépésre 2	236.-
Klucs-László: Excel 4.0 lépésről lépésre 3	227.-
Klucs-László: Excel 4.0 lépésről lépésre 4	225.-
Klucs-László: Excel 4.0 lépésről lépésre 5	200.-
• Hargittai-Kaszanyiczki: Grafikák készítése IBM PC -n	290.-
• Hargittai Péter: Harvard Graphics	590.-
Kelevitz Ferenc: MS ACCES for Windows	399.-
Bartha Attila: Norton ... for DOS	499.-
Bartha Attila: Norton Utilities 6.01	599.-
Hold Gábor: Page Maker 4.0 for Windows	599.-
Gál-Dallos: Quick BASIC 	469.-
Gál István: Quick BASIC Szubrutinyűjtemény	292.-
Nyéki-Nagy: Turbo BASIC	453.-
• Benkő-Poppe: Turbo C++ 	493.-

Pirkó József: Turbo Pascal 5.5	599.-
Pirkó József: Turbo Pascal 6.0 & for Windows	799.-
Pirkó József: Turbo Pascal 7.0	599.-
Hargittai-Kaszanyiczki: Visual BASIC for Windows	599.-
Hargittai-Kaszanyiczki: Visual C++	599.-
• Mörk Péter: Word for Windows 2.0	290.-
Dr Kovács Magda:	
Mikroszámítógép-mikroelektronikai értelmező szótár I.-III. kötet (English, Deutsch, Español, Esperanto, Magyar)	820.-
Dr Kovács Magda:	
Mikroszámítógép-mikroelektronikai értelmező szótár I.-III. kötet	699.-
Balogh-Berkes-Kovács:	
Számítógépes távközlés telematikai szolgálatai	463.-

 Lemez melléklet

Az áraink az ÁFA-t nem tartalmazzák.

Oktatási intézményeknek 30% kedvezmény!!!

Megrendelhető:

LSI Oktatóközpont
1033 Budapest Hévízi út 6/e

Telefon: 188-9751

Ára: 420 Ft



LSI Oktatóközpont

1033 Budapest Hévízi út 6/e.
Tel: 188-9751

Viszonteladókat is kiszolgálunk!