

Osborne McGraw-Hill

EGYSZERŰEN

DOS

KÖZÉPHALADÓ

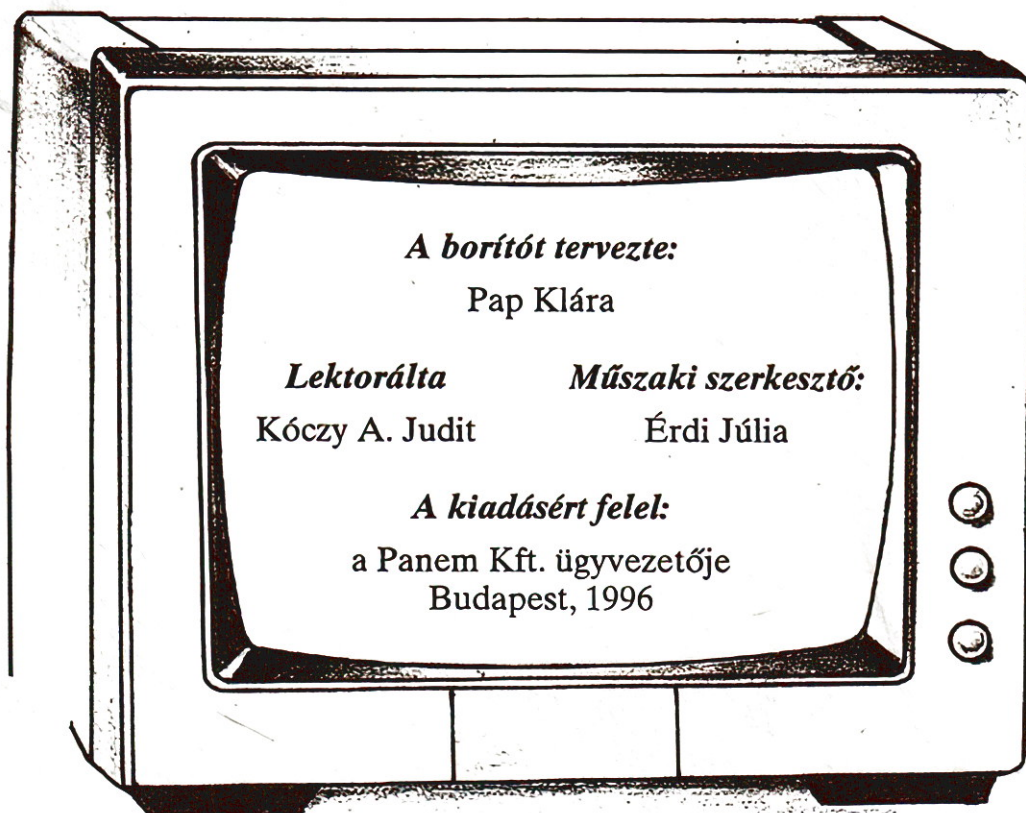
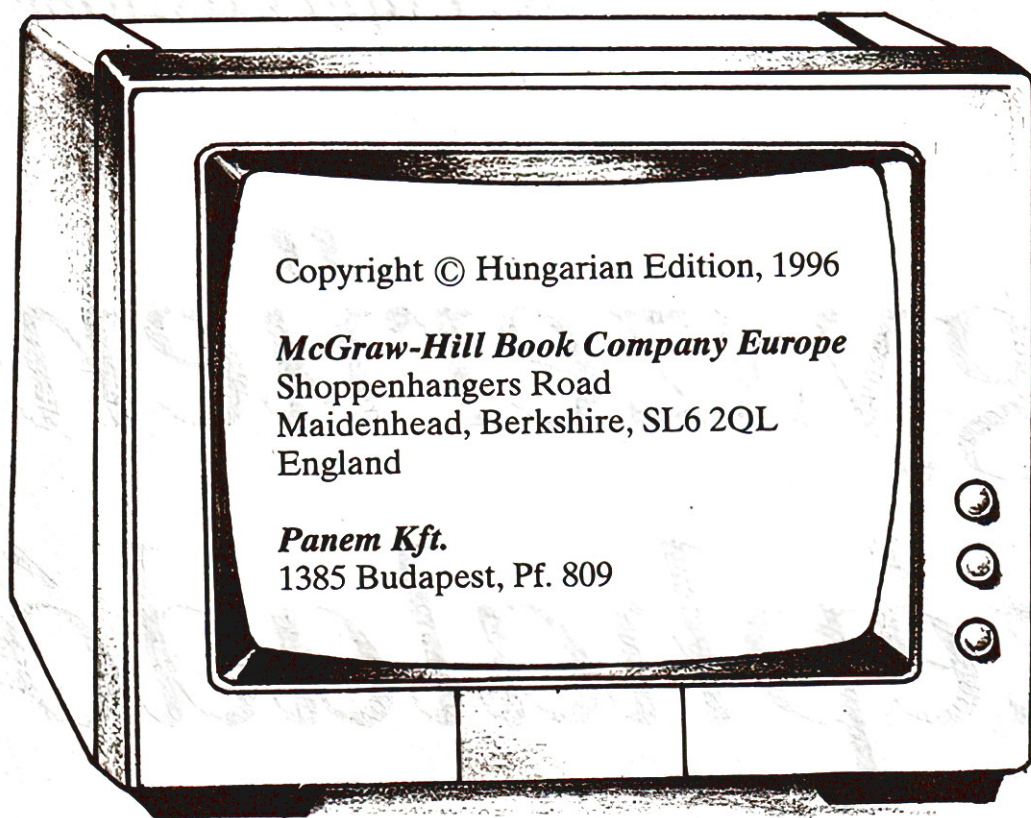
K. PAPP LÁSZLÓNÉ



*K. Papp Lászlóné*

*Egyszerűen  
középhaladó  
DOS*

*Panem–McGraw–Hill*



ISBN 963 545 077 X  
2. kiadás

A PANEM KÖNYVEK megrendelhetők a 06-30/488-488 hívószámú  
WESTEL 900 GSM mobiltelefonon, illetve az 1385 Budapest, Pf. 809 levélcímen.

# Tartalomjegyzék

<b>Bevezetés</b>	7
<b>0. Fejezet</b>	
<b><i>Ami az Egyszerűen DOS c. könyvből kimaradt</i></b>	9
Lemeznevek megadása a LABEL paranccsal	9
A lemeznév lekérdezése a VOL paranccsal	11
Eszköznevek a DOS-ban	12
A parancsok szintaxisának elemei	17
Ismétlés = Kulcs a sikerhez	19
Mit jelent?	20
<b>1. Fejezet</b>	
<b><i>Gépelési hibák javítása a parancssorban</i></b>	21
Az F1—F6 funkcióbillentyűk használata	22
Az INS és a DEL billentyű használata	27
A DOSKEY segédprogram	29
A DOSKEY utasítással létrehozható makrók	35
Ismétlés = Kulcs a sikerhez	37
Mit jelent?	38
<b>2. Fejezet</b>	
<b><i>Állománykezelő parancsok haladóknak</i></b>	39
Lemzemásolat készítése a DISKCOPY paranccsal	40
Lemezok tartalmának összehasonlítása a DISKCOMP paranccsal	46
Állományok másolása a XCOPY paranccsal	49
Két állomány tartalmának összehasonlítása a COMP utasítással	57
Állományok aktualizálása a REPLACE paranccsal	62
Ismétlés = Kulcs a sikerhez	67
Mit jelent?	67

### 3. Fejezet

#### ***Az állományok logikai védelme és a véletlenszerűen törölt állományok visszaállítása*** 68

Az állományok attribútumainak megtekintése és beállítása az ATTRIB paranccsal	68
Véletlenszerűen törölt állományok visszaállítása	73
Ismétlés = Kulcs a sikerhez	91
Mit jelent?	92

### 4. Fejezet

#### ***Nagyméretű állományok hajlékonylemezre másolása (avagy BACKUP/RESTORE, könyvek nélkül)*** 93

Előkészületek a BACKUP használatához	95
A BACKUP utasítás használata	97
A BACKUP legfontosabb kapcsolói	98
A BACKUP végrehajtása adott állomány, adott könyvtár, ill. az egész merevlemez esetén	100
A RESTORE parancs	104
A RESTORE parancs kapcsolói	105
Gyakoroljuk a BACKUP és a RESTORE parancs használatát!	107
Ismétlés = Kulcs a sikerhez	113
Mit jelent?	113

### 5. Fejezet

#### ***További tudnivalók a parancsállományokkal kapcsolatban*** 114

Parancsállomány paraméterek	116
Kilencnél több parancsállomány paraméter használata (SHIFT)	119
A parancsállományok egyszerű parancsai (REM, PAUSE, CALL)	120
A parancsállomány feltételes programozása, az IF parancs használata	126

Adott DOS parancs végrehajtása több állományra (a FOR parancs)	132
Ugrás végrehajtása a parancsállományon belül (a GOTO parancs)	134
Az AUTOEXEC.BAT állomány minimális és szokásos tartalma	136
Saját parancsállományok	137
Ismétlés = Kulcs a sikerhez	139
Mit jelent?	141

## 6. Fejezet

<b><i>A CONFIG.SYS állomány adott felhasználói igényekhez igazítása</i></b>	142
---	-----

Az utolsó érvényes meghajtó betűjelének előírása a LASTRDIVE-bejegyzéssel	143
Eszközmeghajtó programok telepítése a DEVICE= bejegyzéssel	144
Megjegyzések elhelyezése a CONFIG.SYS állományban	151
Az INSTALL= bejegyzés használata memóriarezidens programok betöltéséhez	152
A SHELL= bejegyzés használata a parancsfeldolgozó állományok nevének és helyének megadásához	154
A STACKS= bejegyzés használata a hardvermegszakítások fogadásának kezelésére	157
Ismétlés = Kulcs a sikerhez	158
Mit jelent?	159

## 7. Fejezet

<b><i>Lehetőségek a nemzeti sajátosságok figyelembevételére 5.0-s DOS használata esetén.</i></b>	160
--	-----

A dátum- és idő kijelzés formátumának beállítása a COUNTRY= bejegyzéssel	162
A kódlapok betöltése és aktivizálása	167
A kódlappal kapcsolatos információk lekérdezése	176

Konkrét helyzetek, konkrét példák	179
Ismétlés = Kulcs a sikerhez	183
Mit jelent?	183
<b>8. Fejezet</b>	
<b><i>Memóriakezelés az 5.0-s DOS használata esetén</i></b>	<b>184</b>
Memóriatípusok	184
Az éppen aktuális memóriahasználat lekérdezése a MEM paranccsal .	190
Memóriakezelés és sebességnövelés	194
Lehetőségek a rendelkezésre álló hagyományos memória jobb kihasználására	195
A számítógép működésének gyorsítása	202
Ismétlés = Kulcs a sikerhez	211
Mit jelent?	212
<b>A Függelék</b>	
<b><i>ERRORLEVEL értékek</i></b>	<b>213</b>
<b>B Függelék</b>	
<b><i>Kódlapok az 5.0-s DOS-ban</i></b>	<b>216</b>
<b>C Függelék</b>	
<b><i>Kódtáblázat a nemzeti sajátosságok figyelembevételére 5.0-s DOS használata esetén</i></b>	<b>223</b>
<b>D Függelék</b>	
<b><i>Hardver napló</i></b>	<b>224</b>
<b><i>Mutató</i></b>	<b>227</b>

# Bevezetés

A számítástechnikával foglalkozó szakirodalmat figyelve a DOS az a témakör, amellyel kapcsolatban magyarul is a legtöbb könyv jelent meg. Mi adta tehát a bátorságot ahhoz, hogy ennek ellenére egy újabb DOS könyv elkészítésére vállalkozzunk?

1. A DOS-hoz a Microsoft cég mind a mai napig nem készítette el a referencia és felhasználói kézikönyvek magyar nyelvű fordítását.
2. A DOS-szal foglalkozó legtöbb könyvre az a jellemző, hogy rögtön „belevág a közepébe”, vagyis sok olyan speciális szakkifejezést használ és alkalmaz, amelyet csak a számítástechnikai szakemberek és programozók értenek. Így ezek a könyvek a „mezítlábas” felhasználók számára nem nyújtanak segítséget.
3. AZ EGYSZERŰEN DOS c. könyv megjelenése után az Olvasók várták a folytatást, vagyis a középhaladók számára készített, azonban az EGYSZERŰEN sorozat hagyományaihoz alkalmazkodó második kötetet.

A könyv elkészítése során tehát a következő szempontokat vettük figyelembe:

Feltételezzük azt, hogy az Olvasó az EGYSZERŰEN DOS c. könyvet már elolvasta. Ennek ellenére azonban minden fejezet elején a MÁR TANULTUK részben újból ismertetjük azokat a legfontosabb fogalmakat, amelyek az adott fejezet megértéséhez feltétlenül szükségesek.

Az Egyszerűen DOS c. könyvben — a tárgyalt parancsok és alapfogalmak természetéből adódóan — még meg lehetett oldani a DOS verzióktól való függetlenséget, vagyis a leírtak a 3.3-as DOS verzióban ugyanúgy igazak, mint az 5.0-s DOS esetén. Ebben a könyvben azonban már ezt a szempontot két okból sem volt érdemes betartani:

1. A legtöbb IBM gépen az 5.0-s DOS-szal dolgoznak. (A könyv megjelenésének időpontjában már a 6.0-s DOS is megvásárolható!!)
2. A verziófüggetlenség érdekében nagy könnyelműség lett volna nem tárgyalni azokat a hatékony szolgáltatásokat, amelyekkel az 5.0-s DOS-t ellátták. Ennek értelmében tehát:
  - a könyvben szereplő valamennyi képernyőábra és üzenet az 5.0-s DOS-nak felel meg;
  - azok érdekében, akik nem az 5.0-s DOS-szal dolgoznak, minden parancs esetében megadtuk azt a legelső DOS verziót, amelytől kezdve a parancs a felhasználók rendelkezésére áll. Az itt alkalmazott DOS rövidítés azt jelzi, hogy a leírtak valamennyi DOS verzióra érvényesek.



A könyv elkészítésénél a legfontosabb szempont — igazodva az EGYSZERŰEN ... sorozat többi tagjához — a használhatóság és a közérthetőség volt. Ezt szolgálja a rendszerüzenetek és a képernyőábrák alatt közvetlenül elhelyezett magyar fordítás, valamint az ábrák és a példák. A kényelemnek azonban ára van! Az Olvasó csak akkor tudja igazán megérteni és megtanulni a leírta-  
kat, ha leül a számítógép mellé és mindent kipróbál, ill. a valóságban megta-  
pasztal. Zongorázni sem lehet megtanulni az asztalnál ülve. Ugyanez igaz a számí-  
tástechnikára is. Még ha először remegő kezekkel is, de használjuk a számí-  
tógépet és próbáljuk ki a könyvben tárgyalt parancsokat! Igazi tudást csak így szerezhetünk!

A könyvben tárgyalt parancsok összetettsége megkövetelte, hogy a paran-  
csok szintaxisát, vagyis a parancsok teljes alakját is megadjuk. A parancsok  
ezen formájától általában mindenki idegenkedik. Ennek rendszerint az az oka,  
hogy az Olvasók a jelölésrendszerben nem tudnak biztonságosan eligazodni.  
Pedig a szintaxis megértése a parancsok megfelelő használatához nagyon  
fontos. Ennek a sok esetben komoly problémának megyünk elébe akkor, ami-  
kor a 0. Fejezet táblázatában részletesen tárgyaljuk a DOS parancsok szintaxi-  
sának elemeit.

Végezetül fogadjanak meg egy tanácsot! Ha végképp nem értenek valamit,  
kérdézzék meg egy tapasztaltabb kollégájuktól. Ha azonban feltesznek egy  
kérdést, mindig legyenek felkészülve, hogy a választ esetleg az sem tudja, akit  
megkérdeztek. (Ennek ellenére választ még kaphatunk!) Nem mindenki szüle-  
tett oktatónak, és ezért nem biztos, hogy rendelkezik azzal a türelemmel és  
szókinccsel, amely a tudása átadásához szükséges. Azonkívül a számítástech-  
nika — sajnálatosan — azon tudományterületek közé tartozik, ahol az embe-  
rek rendszerint nem szívesen adják át azt az ismeretet, amelyet maguk is csak  
nehézségek árán szereztek meg. Mindezek tudatában igyekeztünk a könyvet  
úgy elkészíteni, hogy annak olvasása során minél kevesebbet kelljen az Olva-  
sónak kérdeznie.

Jó munkát és sok sikert kívánunk!

*K. Papp Lászlóné és Kóczy A. Judit*

# Ami az *EGYSZERŰEN DOS c.* könyvből kimaradt

**Már tanultuk:** port, CON eszköznév

A **port** a számítógép hátulján elhelyezett kommunikációs aljzat, amelyhez a külső eszközök (pl. nyomtató vagy egér) csatlakoztathatók.

A DOS minden egyes külső eszközhöz — így a monitorhoz, billentyűzethez vagy portokhoz — saját azonosítónevet, ún. eszköznevet rendel. A **CON**, amely ugyancsak egy eszköznév, aszerint utal a billentyűzetre vagy a monitorra, hogy az adott műveletben az forráshelyként (ekkor a billentyűzetet jelenti), vagy célhelyként (ekkor a monitort jelenti) szerepel.

Ebben a fejezetben két parancsot és egy olyan témakört tárgyalunk, amelyek alapvető fontosságúak, és tulajdonképpen még a kezdők számára készített DOS könyvbe tartozhatnának. A parancsok a lemezek kötetelnevezését lekérdező **VOL**, a lemezek kötetelnevezését beállító **LABEL**, a tárgyalni kívánt témakör pedig a DOS által használt eszköznevek ismertetése. Végezetül, de nem utolsó sorban a DOS parancsok szintaxisának elemeit is megadjuk.

## *Lemeznevek megadása a LABEL parancssal*

3.0

KÜLSŐ

A DOS lehetővé teszi, hogy a lemezekhez olyan azonosító nevet, másképpen kötetelnevezést (*volume label*) rendeljünk, amelyet a rendszer magára a lemez adatterületére ír rá.

A lemeznév megadására első alkalommal a **FORMAT** parancs futtatásakor

van lehetőségünk, ui. a FORMAT, amint végez a lemez formázásával, a következő üzenettel szólít fel a kötetelnevezés megadására: (csak 4.0 vagy magasabb DOS verzió esetén!)

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

```
Kötetelnevezés (11 karakter, ha nincs rá szükségünk, akkor az ENTER-t kell lenyomni)?
```

Mint látható, a szóközöket is beleszámítva maximálisan 11 karakterből álló név megadására van lehetőségünk, a következő karaktereket azonban kerülünk kell a név megadásakor:

```
* ? \ / ! . , ; : + = [ ] ( ) & ^ < > "
```

Könnyen előfordulhat, hogy formázáskor elmulasztjuk a lemeznév megadását — az előbbi üzenet után, a lemeznév megadása nélkül az ENTER-t nyomjuk le —, utólag azonban mégis szükségét érezzük azonosítónév hozzárendelését a lemezhez. Ehhez a LABEL parancsot a meghajtó betűjelével és az azonosítónévvel együtt kell futtatnunk. A LABEL parancs általános alakja a következő:

```
LABEL [meghajtó betűjele:] [kötetelnevezés]
```

Például ha az A: meghajtóban elhelyezett lemez nevét MUNKA1-re akarjuk beállítani, akkor a következő parancs végrehajtására van szükségünk:

```
C:\>LABEL A:MUNKA1 <ENTER>
```

Az A: meghajtóban elhelyezett lemez már meglévő nevének törlése a következő paranccsal érhető el:

```
C:\>LABEL A: <ENTER>
```

```
Volume in drive A is MUNKA1
```

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

```
Az A meghajtó kötetelnevezése MUNKA1
Kötetelnevezés (11 karakter, ha nincs rá szükségünk,
akkor az ENTER-t kell lenyomni)?
```

Ha a megjelenő üzenetre az ENTER gombot nyomjuk le, akkor a DOS mielőtt a lemez nevét törölné, még egy végső visszaigazolást vár:

Delete current volume label (Y/N)?

Az éppen aktuális kötetelnevezés törlése (I/N)?

A DOS tehát csak akkor törli a lemez éppen aktuális kötetelnevezését, ha az előbbi kérdésre Y, azaz igen választ kap.

A LABEL parancs segítségével a lemezekhez kötetelnevezés rendelhető, valamint a már meglévő kötetelnevezés módosítható vagy törölhető. A parancs után mindig annak a meghajtónak a betűjelét kell megadnunk, amelyre a műveletet végre akarjuk hajtani. Ha a meghajtó betűjelét elhagyjuk, akkor a parancs az éppen aktuális meghajtóra vonatkozik.

## *A lemeznév lekérdezése a VOL paranccsal*

2.0

**BELSŐ**

Ha csak a lemez nevét akarjuk lekérdezni, akkor a LABEL-nél — amely ugyancsak kijelzi az éppen aktuális kötetelnevezést — egyszerűbb és biztonságosabb a VOL parancs használata. Tegyük fel, hogy az A: meghajtóban elhelyezett lemezt a formázásakor a MUNKA1 névvel láttuk el. Ha most ezt a nevet le akarjuk kérdezni, akkor a következőket kell a parancssorba begépelnünk:

C:\>VOL A: <ENTER>

Erre a következő üzenetet kapjuk:

```
Volume in drive A is MUNKA1
```

Az A meghajtó kötetelnevezése MUNKA1

*Megjegyezzük, hogy a lemez neve a DIR paranccsal is lekérdezhető, ekkor azonban a lemez tartalmára vonatkozó további adatok — állománynevek és/vagy könyvtárnevek stb. — is megjelennek.*

A lemezek kötetelnevezésének lekérdezésére a DOS-ban a VOL parancs áll rendelkezésünkre. Ha a VOL után nem adjuk meg a meghajtó betűjelét, akkor az éppen aktuális meghajtó kötetelnevezését kapjuk meg. Például:

```
C:\>VOL <ENTER>
Volume in drive C is DISK_VOL2
```

A C meghajtó kötetelnevezése DISK\_VOL2

## ***Eszköznevek a DOS-ban***

Az EGYSZERŰEN DOS c. könyvben már tanultuk, hogy a lemezen minden egyes állományhoz egyedi azonosítónév tartozik. Bár a CON eszköznév a COPY paranccsal kapcsolatban már felmerült, arról azonban nem volt szó, hogy a DOS a számítógéphez tartozó minden egyes eszközhöz — monitor, billentyűzet, soros és párhuzamos adapter — is egyedi azonosítónevet rendel. Ebben az alfejezetben ezeket a speciális DOS eszközneveket tekintjük át.

### **A CON eszköz használata adatkivitelhez és adatbevitelhez**

Az EGYSZERŰEN DOS c. könyvben a parancsállományok létrehozásakor minden alkalommal a COPY paranccsal a CON eszköznevet alkalmaztuk annak érdekében, hogy az állományt közvetlenül a billentyűzetről — külön szövegszerkesztő használata nélkül — tudjuk létrehozni. Például írjuk be:

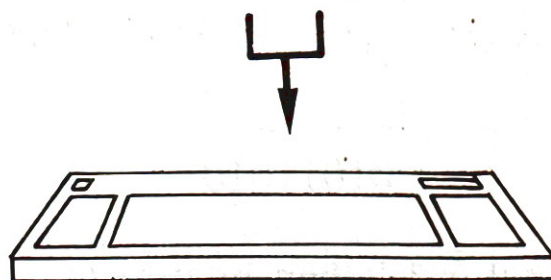
```
C:\>COPY CON PROBA.TXT <ENTER>
A CON eszköz adatbevitelhez és <ENTER>
adatkivitelhez használható. <ENTER>
^Z <ENTER>
    1 file(s) copied
C:\>
```

Ha a CON eszközt az állományok másolási műveletében forráshelyként szerepeltetjük — mint ebben az esetben is —, akkor a DOS a CON eszköznevet a billentyűzettel azonosítja.

A CON eszközt az állományok másolási műveletében azonban célhelyként is meg lehet adni. Ekkor azonban a CON eszköznév már a monitort jelenti.

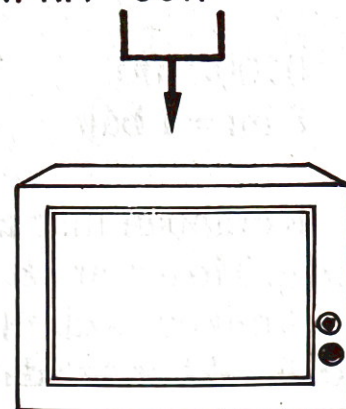
Jogosan merül fel ezek után a kérdés: honnan lehet tudni, hogy a CON a monitorra vagy a billentyűzetre vonatkozik? A választ könnyen megadhatjuk, ha megnézzük, hogy a CON a művelet forráshelyeként vagy célhelyeként szerepel-e. Ha a CON a művelet forráshelyeként szerepel, akkor a billentyűzetet, míg ha a művelet célhelyeként, akkor a monitort jelenti.

```
A:\> COPY CON FILENÉV. KIT
```



*CON eszköz, amely a billentyűzetre utal*

```
C:\> COPY FILENÉV. KIT CON
```



*CON eszköz, amely a monitorra utal*

## Párhuzamos és soros eszközök

Bár az EGYSZERŰEN DOS c. könyvben a számítógépdoboz bemutatásakor és a nyomtatók csatlakoztatásának tárgyalásakor már érintettük a párhuzamos és soros port fogalmát, ott azonban részletesen nem tárgyaltuk, hogy ezek a fogalmak pontosan mit is takarnak az adatátvitel szempontjából.

Amikor a számítógép a hozzá kapcsolt külső eszközökhöz pl. a nyomtatóhoz adatokat küld, akkor ezt szokásos esetben vagy a soros, vagy a párhuzamos adatátviteli mód egyikével végzi. Mit jelentenek ezek a fogalmak?

Mint tudjuk, a számítógép minden információt — számokat, karaktereket, programutasításokat — bináris számjegyeknek, másképpen biteknek nevezett egyesek és nullák sorozataként tárol. Ennek megfelelően az ABC minden egyes betűjéhez bináris, vagyis kettes számrendszerbeli számjegyek egyedi sorozata (egy kód) tartozik. Például:

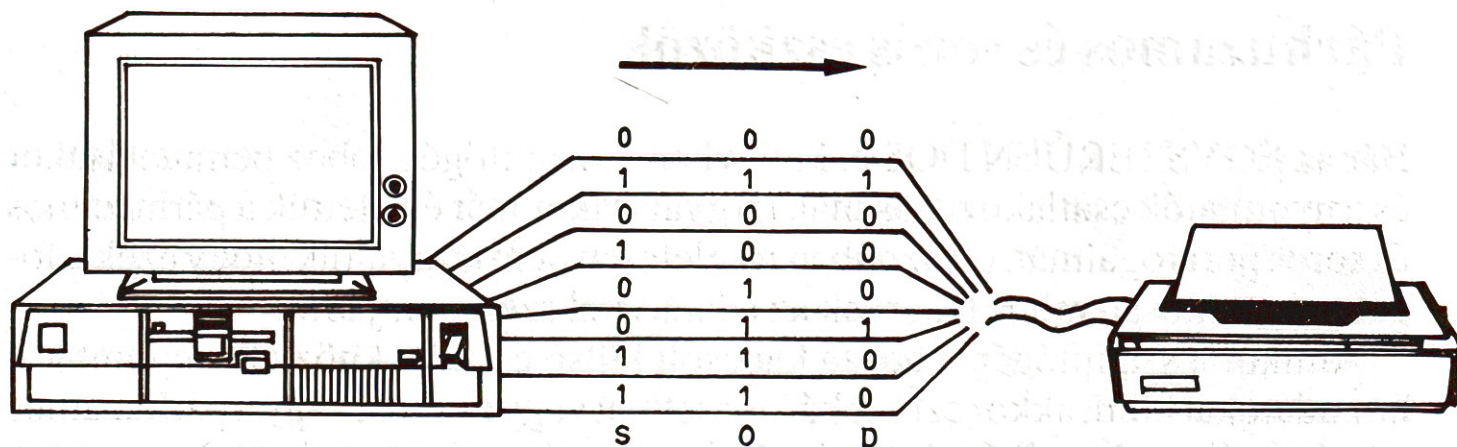
Betű	Bináris megfelelő
A	01000001
B	01000010
C	01000011
D	01000100
.	
.	
.	
Z	01011010

Annak érdekében, hogy a számítógép a számokat és karakterkódokat egységesen tudja kezelni, a bitekből 8 bitből álló csoportokat alakít ki. Ezeket a csoportokat nevezzük bájtnek. Például:

D  
01000100  
8 bit = 1 bájt

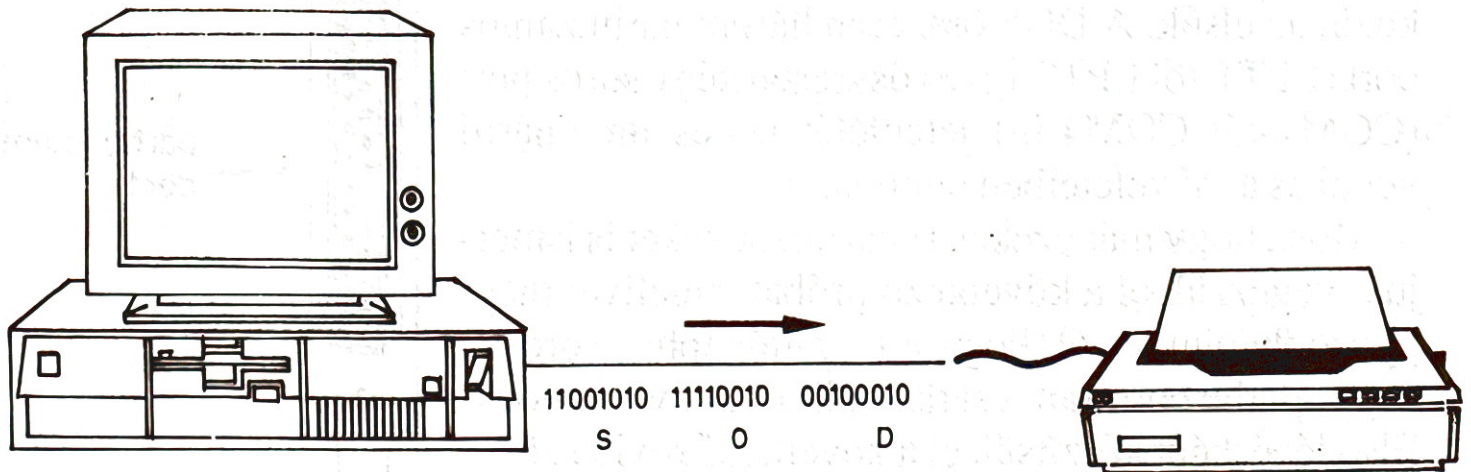
Korábban már tanultuk, hogy a bájt az információ egy karakterének felel meg. Most már azt is tudjuk, hogy miért!

Amikor a számítógép az ún. párhuzamos (*parallel*) eszközhöz küldi az adatokat, akkor az adatátvitelhez egyszerre nyolc adatvezetéket használ, ami azt jelenti, hogy egy teljes karakter átvitelét egyidejűleg tudja megoldani.



Párhuzamos adatátvitel

Ha a számítógép az adatokat soros (*serial*) eszközhöz továbbítja, akkor az adatátvitelhez csak egyetlen vezetékét használ, így egyszerre mindig csak 1 bit mennyiségű információ átvitelét tudja végrehajtani, mint ez a következő ábrán is látható.



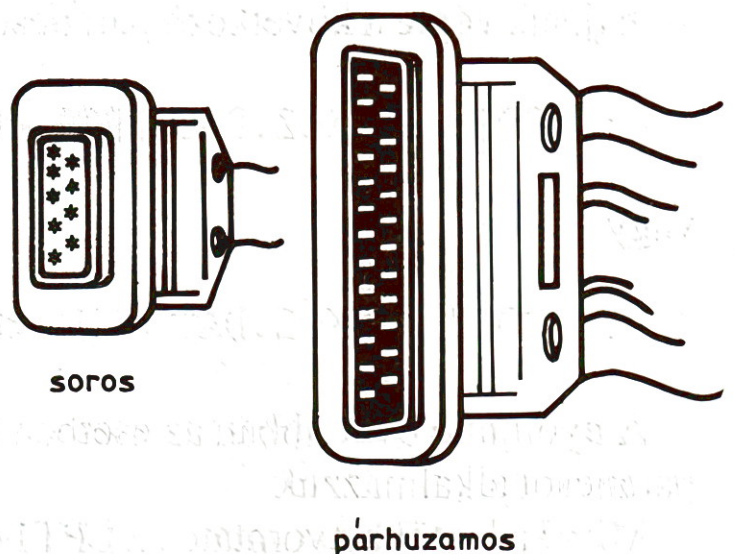
Soros adatátvitel

Soros eszköz esetében az adatátvitelhez csak egy vezetékét használ a számítógép, így a művelet tovább tart, mint a párhuzamos eszköz esetében.

A számítógéphez kapcsolható nyomtatók vagy soros, vagy párhuzamos csatlakoztatásra, vagyis adatátvitelre vannak felkészítve, ennek megfelelően tehát vagy párhuzamos, vagy soros eszközök. A felhasználónak tisztában kell lennie a nyomtató típusával ahhoz, hogy azt használni tudja.

A legegyszerűbben úgy lehet megállapítani a nyomtató típusát, hogy megnézzük azt a kábelt, amely azt a számítógéppel összeköti. (A fejezet végén a típus megállapítására még egy kis tesztet is bemutatunk.) A következő ábrán soros és párhuzamos nyomtatókábelek láthatók.

A legtöbb számítógép legalább egy párhuzamos és egy soros csatlakoztatási lehetőséggel, ún. porttal rendelkezik. A portok általában a számítógép hátulján helyezkednek el.

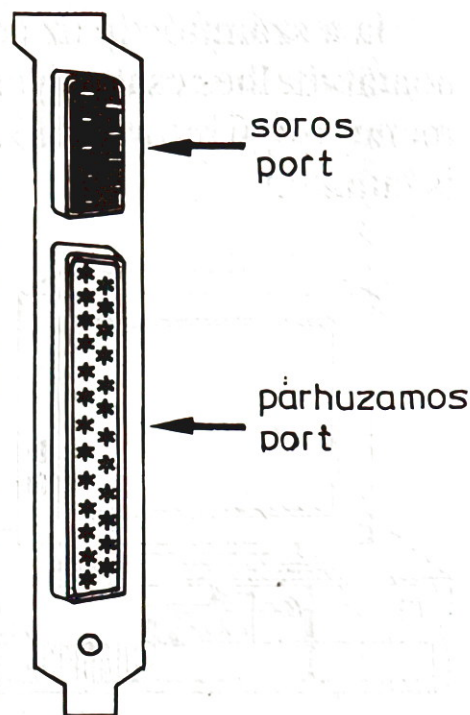


Soros és párhuzamos nyomtatócsatlakozók



A DOS-ban ezekhez a portokhoz is egyedi elnevezés tartozik. Az első párhuzamos port, vagyis kommunikációs aljzat az LPT1, az első soros port pedig a COM1 megnevezést viseli. Az LPT és a COM után az 1-es szám azt jelenti, hogy ezek a több lehetséges port közül az elsők. A DOS összesen három párhuzamos port (LPT1-től LPT3-ig) és összesen négy soros port (COM1-től COM4-ig) jelenlétét képes tudomásul venni és a műveleteiben támogatni.

Most, hogy már ezeket az eszközneveket is ismerjük, végezzük el a következő próbát, amellyel meggyőződhetünk arról, hogy a számítógéphez sorosan vagy párhuzamosan csatlakozik-e a nyomtatónk. Első lépésként készítsük el a következő rövid szöveges állományt:



*Soros és párhuzamos portok*

```
C:\>COPY CON ESZKOZ.DAT <ENTER>
Az LPT1 az első párhuzamos port. <ENTER>
A COM1 az első soros port. <ENTER>
^Z <ENTER>
  1 file(s) copied
C:\>
```

Kapcsoljuk be a nyomtatót, és győződjünk meg arról, hogy a csatlakoztatása a számítógéphez megtörtént. Ezután pedig — a feltételezett típustól függően — hajtsuk végre a következő parancsok egyikét:

```
C:\>COPY ESZKOZ.DAT LPT1 <ENTER>
```

vagy

```
C:\>COPY ESZKOZ.DAT COM1 <ENTER>
```

A nyomtató csak abban az esetben fog nyomtatni, ha a számára megfelelő parancsot alkalmazzuk.

Mivel a legtöbb nyomtató az LPT1 eszközhöz csatlakozik, a DOS-ban erre a portra az LPT1 helyett a PRN eszköznévvel is hivatkozhatunk. Például:

```
C:\>COPY ESZKOZ.DAT PRN <ENTER>
```

Végezetül meg kell említenünk, hogy az első soros port az AUX eszköznévvel is megadható.

A következő táblázat a DOS eszközneveit tartalmazza.

DOS eszköznév	Jelentés
CON	Az eszköznevet megadhatjuk a műveletben forrásként, akkor a billentyűzetet, vagy célhelyként, akkor a monitort jelenti
LPT1 vagy PRN	A számítógép első párhuzamos portja
LPT2	A számítógép második párhuzamos portja
LPT3	A számítógép harmadik párhuzamos portja
COM1 vagy AUX	A számítógép első soros portja
COM2	A számítógép második soros portja
COM3	A számítógép harmadik soros portja
COM4	A számítógép negyedik soros portja

Ezek az eszköznevek a könyvben még többször, és nemcsak a COPY paranccsal kapcsolatban fordulnak majd elő, ezért érdemes arra emlékezni, hogy itt szerepel egy táblázat, amelyben valamennyi DOS eszköznév jelentése megtalálható!

A DOS minden egyes eszközhöz saját azonosítónevet rendel. Ezeket az eszközneveket nemcsak az üzeneteiben, hanem bizonyos DOS parancsokban is használja, ill. a felhasználó számára használni engedi. A CON eszköznév aszerint jelenti a billentyűzetet vagy a képernyőt, hogy az adott műveletben forráshelyként vagy célhelyként szerepel. Az LPT1 vagy a PRN a számítógép első párhuzamos, míg a COM1 vagy AUX az első soros portját jelenti. További párhuzamos portnevek: LPT2, LPT3. További soros portnevek: COM2, COM3, COM4.

## *A parancsok szintaxisának elemei*

A szintaxis a parancsok leírásának helyes módját adja meg, beleértve a parancshoz használható opciók, kapcsolók vagy paraméterek leírásának és sorrendjének módját is. A szintaxisban alkalmazott jelöléseknek konkrét jelentése van, amelyet a következő táblázatban összefoglalunk. A szintaxisban kövér

— kiemelt — betűvel megadott részeket kötelezően és pontosan úgy kell alkalmazni, ahogy az ott szerepel! Lássunk először egy olyan mintasort, amelyben a szintaxis minden eleme előfordul:

```
minta[+r|-r][meghajtó_betűjele:][elérési_út]filenév.kit[...][opciók]
```

| | | | | | | |  
 | | | | | | | |  
 1. 2. 3. 4. 5. 6. 7. 8.

A megjelölt elemek jelentése a következő:

Sorszám	Elem	Jelentés
1.	minta	A parancs nevét jelenti.
2.	[ ]	Azt jelzi, hogy az adott tételt nem kötelező szerepeltetni. Amikor olyan információt adunk meg, amelyet a szintaxis szögletes zárójelekkel jelöl, akkor mindig csak magát az információt írjuk le. <b>A szögletes zárójeleket hagyjuk el!</b>
3.	!	Azt jelzi, hogy a két információ közül csak az egyiket szabad megadni. Például a BREAK [ON!OFF] parancs után vagy az ON, vagy az OFF kapcsoló alkalmazható. A kettő egyszerre nem (BREAK ON, BREAK OFF).
4.	meghajtó_betűjele:	A merev- vagy a hajlékonylemez meghajtó betűjelét jelenti. Ha a parancsban olyan állományt írunk elő (1. és 6. pont), amely nem az éppen aktuális meghajtóban helyezkedik el, vagy nincs előírva keresési helyként a PATH parancssal, akkor a meghajtó betűjelének előírásától nem tekinthetünk el.
5.	elérési_út	Azon könyvtár vagy könyvtárak nevét jelenti, amelyeken az operációs rendszernek végig kell haladnia ahhoz, hogy a parancsban megadott állományt (6. pont) megtalálja. Az elérési útra minden olyan esetben szükség van, ha 6.-tal je-

Sorszám	Elem	Jelentés
		lölt állomány nem a gyökérkönyvtárban helyezkedik el, vagy a hozzá vezető útvonalat a PATH parancsban nem írtuk elő.
6.	filenév. kit	Az állomány nevét jelenti. Az állománynév maximálisan 8 karakterből állhat, ez után következik a pont, majd pedig a maximálisan három karakterből álló kiterjesztés.
7.	...	Azt jelzi, hogy az előtte álló információt a parancsban többször meglehet ismételni. A szögletes zárójelhez hasonlóan ebben az esetben is mindig magát az információt kell megadni, vagyis ismételni, nem pedig magát a három pontot.
8.	opciók	Olyan paraméterekre vagy kapcsolókra utal, amelyek megadása nem kötelező. A kapcsolók előtt általában törtjel is áll. Például: /P.

## ***Ismétlés = Kulcs a sikerhez***

A LABEL parancs segítségével a lemezekhez kötetelnevezés rendelhető, valamint a már meglévő kötetelnevezés módosítható vagy törölhető. A parancs után mindig annak a meghajtónak a betűjelét kell megadnunk, amelyre a műveletet végre akarjuk hajtani. Ha a meghajtó betűjelét elhagyjuk, akkor a parancs az éppen aktuális meghajtóra vonatkozik.

A VOL parancs a lemezek kötetelnevezésének lekérdezésére használható. Bár ezt az adatot a DIR parancs is közli, ezután azonban még a lemezen levő könyvtárak és állományok nevét is a képernyőre írja. Ha viszont csak a kötetelnevezésre vagyunk kíváncsiak, akkor érdekesebb a VOL parancsot alkalmaznunk. A parancs után a meghajtó betűjelét kell előírni, ellenkező esetben a parancs az éppen aktuális meghajtóra vonatkozik.

A DOS minden egyes eszközhöz saját azonosítónevet rendel, amely eszközneveket nemcsak az üzeneteiben, hanem bizonyos DOS parancsokban is használja, ill. a felhasználó számára használni engedi. A CON eszköznév asze-

rint jelenti a billentyűzetet vagy a képernyőt, hogy az adott műveletben forrás-helyként vagy célhelyként szerepel. Az LPT1 vagy a PRN a számítógép első párhuzamos, míg a COM1 vagy AUX az első soros portot jelenti. További párhuzamos portnevek: LPT2, LPT3. További soros portnevek: COM2, COM3, COM4.

## *Mit jelent?*

**Bit vagy bináris szám.** A feldolgozás és a tárolás során a bit az a legkisebb információegység, amelyet a számítógép kezelni tud. Értéke 0 vagy 1 lehet.

**Bájt.** 8 bitből álló információegység, amely egy karakter — betű, számjegy vagy írásjel — tárolásához és feldolgozásához szükséges a számítógépben.

**Szintaxis.** 1. A parancsok leírásának helyes módja, beleértve a parancshoz használható opciók, kapcsolók vagy paraméterek leírásának és sorrendjének módját is. 2. Valamely programozási nyelv használatát meghatározó szabályrendszer.

# Gépelési hibák javítása a parancssorban

**Már tanultuk:** BACKSPACE, parancsállomány, F6 funkcióbillentyű, külső és belső DOS parancsok

A parancsok futtatásához a parancs nevét a DOS prompt után kell beírni, majd az ENTER billentyűt lenyomni. Gépelési hiba esetén — még az ENTER lenyomása előtt — a kurzortól balra álló karakter a **BACKSPACE** billentyű lenyomásával törölhető.

A **parancsállomány** olyan szöveges állomány, amely egy vagy több DOS parancs nevét tartalmazza. A parancsállományoknak **.BAT** kiterjesztéssel kell rendelkezniük. Ha a felhasználó a DOS prompthoz beírja egy parancsállomány nevét, akkor a DOS azt a lemezen megkeresi, és a benne felsorolt valamennyi parancsot azok előírásának sorrendjében végrehajtja.

Az **F6 funkcióbillentyű** arra utasítja a DOS-t, hogy az állomány vége karakterrel — ^Z — zárja le az állományt akkor, amikor valamely szöveges állományt a billentyűzetről közvetlenül begépelve, vagyis külön szövegszerkesztő program használata nélkül állítunk elő. (L. Egyszerűen DOS: Egyszerű parancsállományok készítése szövegszerkesztő program használata nélkül.)

A **belső DOS parancsok** azok a parancsok, amelyeket a **COMMAND.COM** állomány tartalmaz. Mivel a DOS az elindításakor ezt az állományt minden alkalommal automatikusan betölti, így ezek a parancsok is állandóan a memóriában vannak.

A **külső DOS parancsok** azok a parancsok, amelyek állományokban és lemezen foglalnak helyet. Ezeket a parancsokat tehát a DOS csak úgy tudja használni, ha előbb a lemezről betölti a memóriába. A külső DOS parancsokat rendszerint a merevlemezen a **C:\DOS** alkönyvtárban helyezik el, és ezt a könyvtárnevet az **AUTOEXEC.BAT** állomány **PATH** utasításában is előírják. Például **PATH C:\DOS;**

A parancsok begépelése során gyakran hibázunk, de az is előfordul, hogy a parancsok hatásának vagy opcióinak kipróbálása során többször akarjuk

ugyanazt a parancsot más kapcsolóval (minősítő karakterrel) vagy paraméterrel futtatni. Munkánkat jelentősen megkönnyíthetjük, ha a parancssor javítására, ill. módosítására a DOS által kínált szerkesztőbillentyűket, ill. 5.0-s DOS esetén a DOSKEY segédprogramot is használjuk.

Mielőtt azonban hozzákezdenénk a szerkesztőbillentyűk tárgyalásához, röviden tekintsük át, mi történik, ha egy DOS parancs vagy más futtatható program nevét a parancssorba beírjuk, és az ENTER gombot lenyomjuk:

1. A DOS ellenőrzi, hogy a beírt név szerepel-e a DOS parancsok vagy futtatható programok között. Ha igen, akkor megkísérli végrehajtani, ha nem, akkor hibaüzenetet küld a képernyőre.
2. A parancssorba begépelte karaktereket egy átmeneti tárolóban, ún. billentyűzetpufferben helyezi el.

A DOS szerkesztőbillentyűi ezen átmeneti tároló tartalmát használják fel. Lássuk hogyan!

## Az F1—F6 funkcióbillentyűk használata

### DOS

Tegyük fel, hogy a gyökérkatalógus tartalmára vagyunk kíváncsiak, az ehhez szükséges parancsot azonban — DIR — hibásan írtuk be a parancssorba:

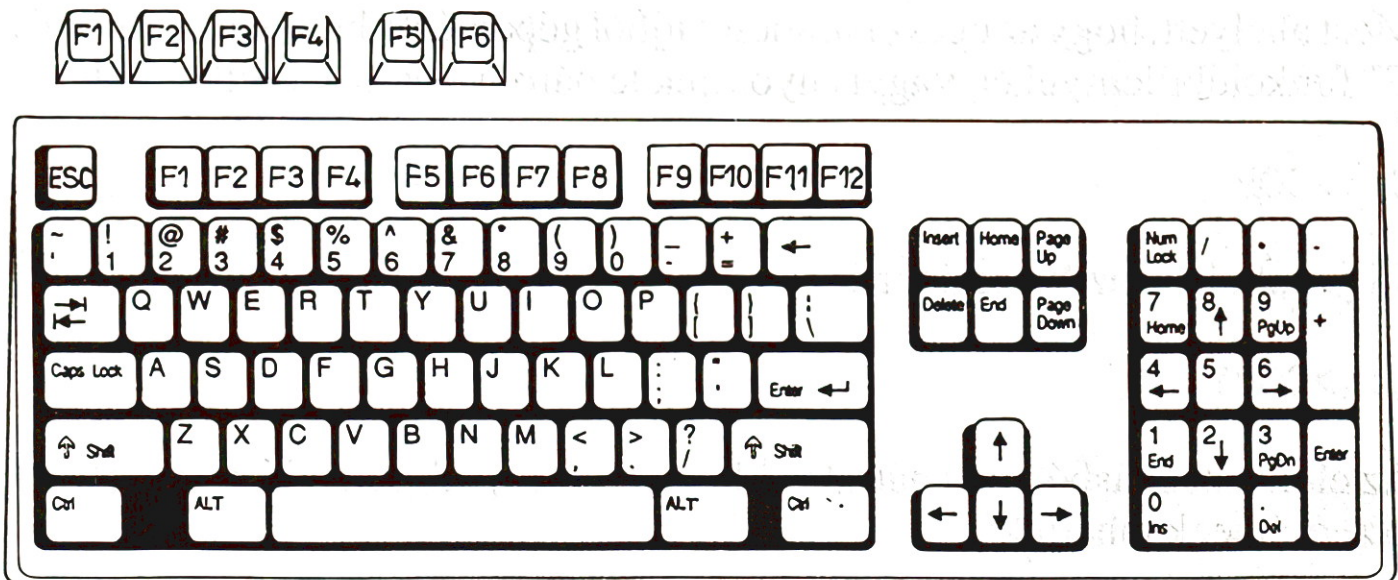
```
C:\>DIR <ENTER>
```

A DOS-tól a *Bad command or file name* (Téves parancs vagy állománynév) hibaüzenetet kapjuk, mivel ilyen néven sem a belső, sem a külső parancsai között nem talál megfelelőt. Mit tettünk ilyen esetben a korábbiakban? Begéveltük újból a parancsot, és lenyomtuk az ENTER gombot, hibáinkat pedig — még az ENTER lenyomása előtt — a BACKSPACE billentyűvel javítottuk. Most azonban nyomjuk le az F1 funkcióbillentyűt!

Erre a DOS a D karaktert jeleníti meg a parancssorban, mivel ezt tárolta első karakterként a billentyűzetpufferben:

```
C:\>D
```

Nyomjuk le ismét az F1 billentyűt! A DOS erre a parancssorba korábbiakban beírt következő karakter megjelenítésével válaszol:



Az F1—F6 funkcióbillentyűk

```
C:\>DI
```

Ha ezután ismét lenyomnánk az F1 billentyűt, a DOS a következő, vagyis a helytelenül gépelt betűt is bemásolná. Írjuk be tehát az *r* betűt, és indítsuk a parancsot az ENTER lenyomásával:

```
C:\>DIR <ENTER>
```

Tegyük fel, hogy amint a parancsot végrehajtottuk, máris meggondoltuk magunkat, és nem a teljes katalógus tartalmára vagyunk kíváncsiak, csak egy adott állomány jelenlétére. Amint tehát visszakaptuk a DOS promptot, nyomjuk le az F3 funkcióbillentyűt! Ekkor a DOS parancssorban a következő karakterek jelennek meg:

```
C:\>DIR
```

Tehát az F3 billentyű lenyomására a DOS az előzőleg végrehajtott parancssort az új parancssorba másolja be az ott éppen érvényes aktuális kurzorpozíciótól kezdve. Ezután máris beírhatjuk azt az állománynevet, amelyet ténylegesen kerestetni akarunk. De lássunk az F1 és az F3 használatára még egy példát!

Tegyük fel, hogy a hibásan begépelte és futtatott parancs a következő volt:

```
C:\>COPI C: A: <ENTER>
```



Most ahelyett, hogy az egész parancsot újból gépelnénk, használjuk az F1 és az F3 funkcióbillentyűket, vagyis nyomjuk le háromszor az F1 billentyűt:

```
C:\>COP
```

majd gépeljük be az Y karaktert!

```
C:\>COPY
```

Az előző utasításból még hátralévő karaktereket pedig az F3 lenyomásával egyszerre megkaphatjuk:

```
C:\>COPY C: A:
```

A parancs máris futtatható!

Fontos tudnivalók:

1. A DOS az F1 és az F3 funkcióbillentyűk használata során mindig attól a karakterpozíciótól kezdve másolja a billentyűzetpufferből az új parancssorba a karaktereket, amelyet a kurzor az új — éppen szerkesztett — parancssorban elfoglal. Tehát a kurzor éppen aktuális pozíciója a billentyűzetpufferben és a parancssorban azonos.
2. A JOBBRA kurzormozgató billentyűnek az F1-gyel, a BALRA kurzormozgató billentyűnek a BACKSPACE billentyűvel azonos a hatása.

Ha még nem nyomtuk le az ENTER billentyűt és észrevettük, hogy a parancssorban hibásan gépeltünk, egy további szerkesztési lehetőséget is felhasználhatunk. Tegyük fel, hogy a parancssor a következőket tartalmazza:

```
C:\>DATA 02-15-93
```

**Megjegyzés:** *Lehet, hogy az olvasó számítógépén a dátumot nem hónap-nap-év (angolul: mm-dd-yy) sorrendben kell megadni. A parancs zárójelben mindig kijelzi, hogy az adatokat milyen sorrendben várja. A DATE parancs használatakor tehát ezt mindenképpen vegyük figyelembe.*

Nyilvánvaló, hogy a parancssor hibás, mivel a parancs neve helyesen DATE. Nyomjuk le az F5 funkcióbillentyűt! Erre a DOS a parancssor végére @ jelet ír be:

```
C:\>DATA 02-15-93@
```

és a kurzort a D betű alá helyezi. Nyomjuk le az F1 billentyűt háromszor, majd írjuk be az E karaktert, végül pedig nyomjuk le az F3 gombot! Ekkor a parancssorban a következő lesz látható:

```
C:\>DATA 02-15-92@  
DATE 02-15-92
```

A parancs futtatásához ezután már csak az ENTER gombot kell lenyomnunk.

Most pedig tekintsük át röviden a kevésbé ismert és ritkán használt F2 és F4 funkciógombok használatát is. Ezen billentyűk alkalmazásában az a közös, hogy mindegyik lenyomva tartása alatt egy karaktert kell beírni. Különösebb magyarázat helyett egyszerűbb, ha rögtön kipróbáljuk ezeket a funkcióbillentyűket is! Tegyük fel, hogy az előzőleg futtatott parancssor a következőket tartalmazta:

```
C:\>DIR SAJAT
```

A parancs végrehajtása után tegyük fel, hogy ismét egy DIR utasítást akarunk végrehajtani, azonban most egy másik alkatalógusra. Nyomjuk le tehát az F2 gombot, és lenyomva tartva gépeljük be az S karaktert! Erre a következő parancssor jelenik meg a képernyőn:

```
C:\>DIR
```

Vagyis a DOS az F2 billentyűvel képzett kombináció hatására a billentyűzetpufferből azokat a karaktereket másolja az új parancssorba, amelyek közvetlenül megelőzik az F2-vel együtt beírt karaktert.

Az F4 funkcióbillentyű az F2-höz hasonlóan működik. Mindössze annyi az eltérés, hogy ebben az esetben a DOS az új parancssorba csak azokat a karaktereket másolja — beleértve a megadott karaktert is —, amelyek a funkcióbillentyű lenyomva tartása alatt beírt karakter után állnak az előző parancssorban. Próbáljuk ki ezt a funkcióbillentyűt is! Tegyük fel, hogy a parancssor a következőket tartalmazta:

```
C:\>DIR FORMAT
```

Ha most lenyomjuk az F4 billentyűt, és lenyomva tartjuk míg begépeljük az F karaktert, majd pedig lenyomjuk az F3 billentyűt, akkor a DOS a következőként válaszol:

C:\>FORMAT

### Megjegyzés:

1. Az F4 funkcióbillentyűt csak akkor érdemes használni, ha az utoljára indított parancssor több futtatható program nevét tartalmazza.
2. Az F2-vel ellentétben az F4 + karakter alkalmazása esetén az F1/F3 funkcióbillentyűkkel külön gondoskodnunk kell a karakterek megjelenítéséről.

### Az F1—F5 funkcióbillentyűk hatása

**F1** (vagy a JOBBRA kurzormozgató billentyű). Arra utasítja a DOS-t, hogy az éppen aktuális parancssorba az utoljára indított parancssorból azt a karaktert másolja be, amely a kurzor éppen aktuális pozíciójára esik.

**F2 + egy karakter.** Arra utasítja a DOS-t, hogy az előző parancssorból a karaktereket a megadott karakterig — azt azonban már nem beleértve — az éppen aktuális parancssorba másolja.

**F3** Arra utasítja a DOS-t, hogy a karaktereket a kurzor aktuális pozíciójától kezdődően az előző parancssorból az új parancssorba másolja be.

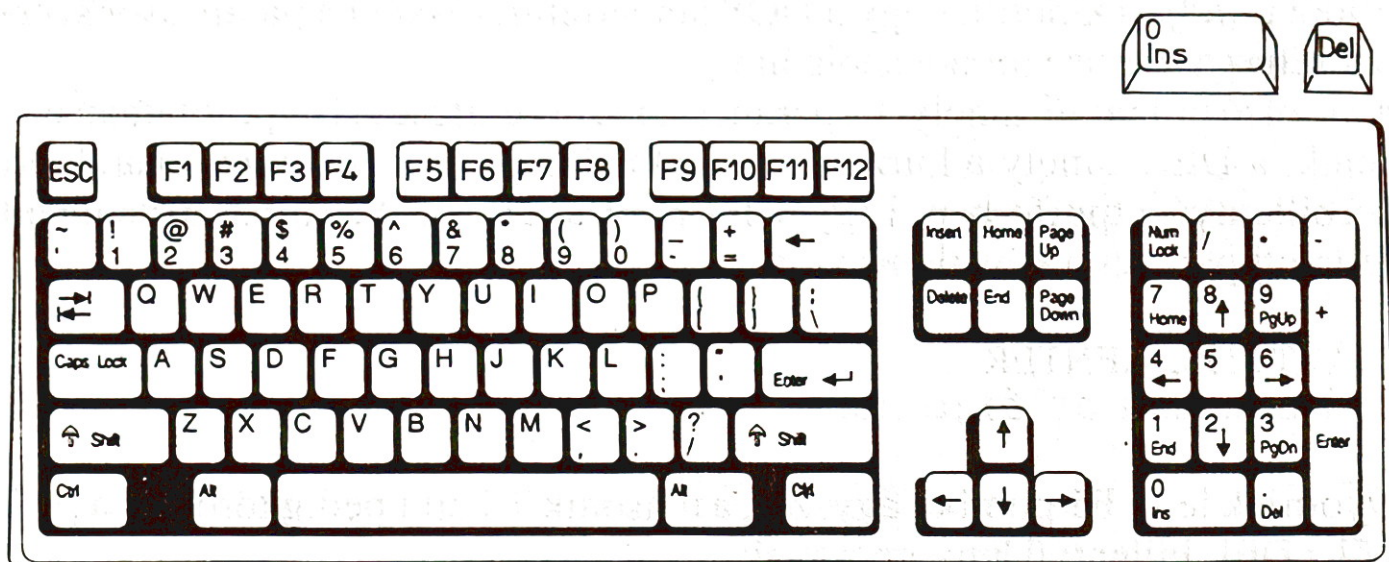
**F4 + egy karakter.** Arra nyújt lehetőséget, hogy az előző parancssorból a karaktereket a megadott karaktertől kezdődően — azt is beleértve — az F1/F3 billentyűk felhasználásával az aktuális parancssorba másolhassuk.

**F5** Az éppen aktuális parancssor átszerkesztését teszi lehetővé. Lenyomására a DOS a @ karaktert helyezi el az éppen aktuális parancssor végén, és a kurzort a következő sor elejére állítja. Ekkor a parancssor átszerkesztése az F1 és az F3 gombokkal kényelmesen elvégezhető.

**Megjegyzés:** Az F2 és az F4 funkcióbillentyűk és az utánuk beírt karakter billentyűkombinációt alkotnak, vagyis a funkcióbillentyűket lenyomva tartva kell a karaktereket begépelni. Erre a szövegben és a táblázatban az F2 + karakter, ill. F4 + karakter jelöléssel utaltunk.

## Az *INS* és a *DEL* billentyű használata

DOS



### A *DEL* és az *INS* billentyű

Eddig csak olyan eseteket tárgyaltunk, amikor a parancssorban félregépelte karaktert kellett kijavítanunk. Az is gyakran előfordul azonban, hogy a parancs nevéből egy betűt hagyunk el. Például:

```
C:\>FOMAT A:
```

Ha a parancsot futtatjuk, a rendszertől a *Bad command or file name* hibaüzenetet kapjuk. Hogyan tudunk most a leggyorsabban és legegyszerűbben javítani? Nyomjuk le kétszer az F1 gombot:

```
C:\>FO
```

majd nyomjuk le az *INS* billentyűt, amelynek hatására a DOS felülíró üzemmódból beszúró üzemmódba kapcsol. Írjuk be ezután az *R* karaktert, majd nyomjuk le az F3 gombot, amelyre a DOS a még hátralevő karaktereket is bemásolja a parancssorba, vagyis:

```
C:\>FORMAT A:
```

Az *INS* gomb tehát a billentyűzetpuffer használata során a felülírás és a beszúrás üzemmódjai között ki/bekapcsoló gombként működik. Nézzük meg,

mi történt volna akkor, ha az előbbi műveletsort az INS lenyomása nélkül végeztük volna el:

```
C:\>FORAT A:
```

Ebből mindjárt kiderül, hogy a DOS alapértelmezésként a parancssor szerkesztéséhez a felülíró módot szolgáltatja.

Az utolsó billentyű, amelyet a parancssor szerkesztésével kapcsolatban tárgyalunk, a DEL, amely a kurzor éppen aktuális pozíciójában levő karaktert törli a billentyűzetpufferben. Tegyük fel, hogy az előzőekben hibásan begépett és futtatott parancs a következő volt:

```
C:\>TTIME <ENTER>  
Bad command or file name
```

Nyomjuk le az F1 gombot egyszer, a második T betűt pedig töröljük a pufferből a DEL billentyű lenyomásával:

```
C:\>T
```

Ezután pedig alkalmazzuk az F3 billentyűt!

```
C:\>TIME
```

### **Az INS és a DEL billentyű használata a parancssor szerkesztéséhez**

Az INS a beszúró és felülíró üzemmódok között mint ki/be kapcsoló működik. Először lenyomva a karakterek beszúrását, újból lenyomva a karakterek felülírását teszi lehetővé a billentyűzetpuffer tartalmának parancssorba másolása során.

A DEL billentyű a kurzor éppen aktuális pozíciójában levő karaktert törli a billentyűzetpufferben.

## A DOSKEY segédprogram

**5.0****KÜLSŐ**

Az előbbieken tárgyalt funkciógombok a parancssor átszerkesztését nagymértékben megkönnyítik, van azonban egy nagy hátrányuk: mivel a billentyűzet pufferét használják, mindig csak egy, a legutolsó parancs átszerkesztését teszik lehetővé. A DOS 5-ös verziójában megtalálható DOSKEY segédprogrammal több futtatott DOS parancs követhető nyomon.

A DOSKEY külső DOS parancs, amely azonban a futtatása után mindaddig a memóriában marad, amíg a számítógépet ki nem kapcsoljuk. Futtassuk tehát!

```
C:\>DOSKEY <ENTER>  
DOSKey installed.
```

```
C:\>DOSKEY <ENTER>  
DOSKey telepítve.
```

**Megjegyzés:** A DOSKEY alapértelmezés szerint 512 bájt helyet foglal le a memóriában. Ha a DOS parancsok átlagos hossza 10 bájt, akkor a DOSKEY közelítőleg 50 DOS parancs tárolását tudja elvégezni. Ha ez a tárolóterület betelik, akkor a DOSKEY az elsőként kiadott parancsnevek törlésével biztosít helyet az új parancsoknak.

Ahhoz, hogy a DOSKEY szolgáltatásait megismerhessük, helyezzünk el néhány parancsot a pufferében. Futtassuk tehát egymás után a VER, DIR, DATE, TIME, CLS parancsot! Ezután nyomjuk le a FÖL kurzormozgató gombot! A parancssorban CLS lesz olvasható. Nyomjuk le ismét a FÖL kurzormozgató gombot! Erre a TIME parancs jelenik meg a parancssorban. Most pedig alkalmazzuk a LE kurzormozgató gombot! Erre ismét a CLS parancsot fogjuk látni, vagyis a FÖL és a LE kurzormozgató gombokkal az előzőekben végrehajtott parancsok listájában, a parancsokat tartalmazó pufferben lépkedhetünk felfelé és lefelé. Figyeljük meg, hogy a művelet során a DOSKEY az éppen aktuális parancsot mindig a parancssorba másolja. Ha a lista tetejére

vagy aljára érkezünk, és ismételten lenyomjuk a FÖL vagy a LE kurzormozgató gombot, akkor ezek lenyomását a DOSKEY egyszerűen figyelmen kívül hagyja.

Következő lépésként ismerkedjünk meg a PGUP és a PGDN gomb használatával! Legyen a parancssorban a TIME parancs, és nyomjuk le a PGUP gombot! Erre a VER parancs neve jelenik meg, mivel a PGUP arra utasítja a DOSKEY programot, hogy a pufferében elsőként tárolt — vagyis legelőször végrehajtott — DOS parancsot jelenítse meg. A PGDN hatása pont ellentétes, vagyis a pufferben utolsóként elhelyezett — vagyis legutoljára végrehajtott — parancs nevét jeleníti meg a parancssorban. Próbáljuk ki ezt a gombot is! A CLS parancs nevének kell megjelennie a parancssorban.

Az eddig tárgyalt gombok használata egyszerű és kényelmes. Mit tegyünk azonban akkor, ha már sok utasítást futtattunk, és nemcsak arra nem emlékszünk, hogy melyik volt az első és az utolsó, de azt sem tudjuk pontosan, hogy melyik utasításokat hajtottuk végre. Ekkor futtassuk a DOSKEY parancsot a (HISTORY, vagy rövidebben) H kapcsolóval, amelynek hatására a DOSKEY a pufferében tárolt valamennyi parancs nevét a képernyőre írja. Az előbbi példánál maradva ez a következő képernyő kialakulásához vezet:

```
C:\>DOSKEY/H <ENTER>
```

```
VER
```

```
DIR
```

```
DATE
```

```
TIME
```

```
CLS
```

```
DOSKEY/H
```

Előfordulhat, hogy a DOSKEY puffere sok parancsot tartalmaz. Ilyen helyzetben már érdemes az F7 és az F9 funkcióbillentyűt használni. Nyomjuk le az F7 gombot! Erre a következőket pillanthatjuk meg a képernyőn:

```
1: VER
```

```
2: DIR
```

```
3: DATE
```

```
4: TIME
```

```
5: CLS
```

```
6: DOSKEY/H
```

Tehát a DOSKEY az F7 lenyomására a pufferében tárolt parancsokat azonosító számokkal ellátva írja a képernyőre. Nyomjuk le ezután az F9 gombot! Erre a következő üzenetet olvashatjuk a parancssorban:

C:\>Line number:

C:\>A sor száma:

vagyis a DOSKEY az F9 lenyomása után a parancsokhoz hozzárendelt azonosító szám (sorszám) beírását várja. Írjuk be pl. az 1-es számot, és nyomjuk le az ENTER-t!

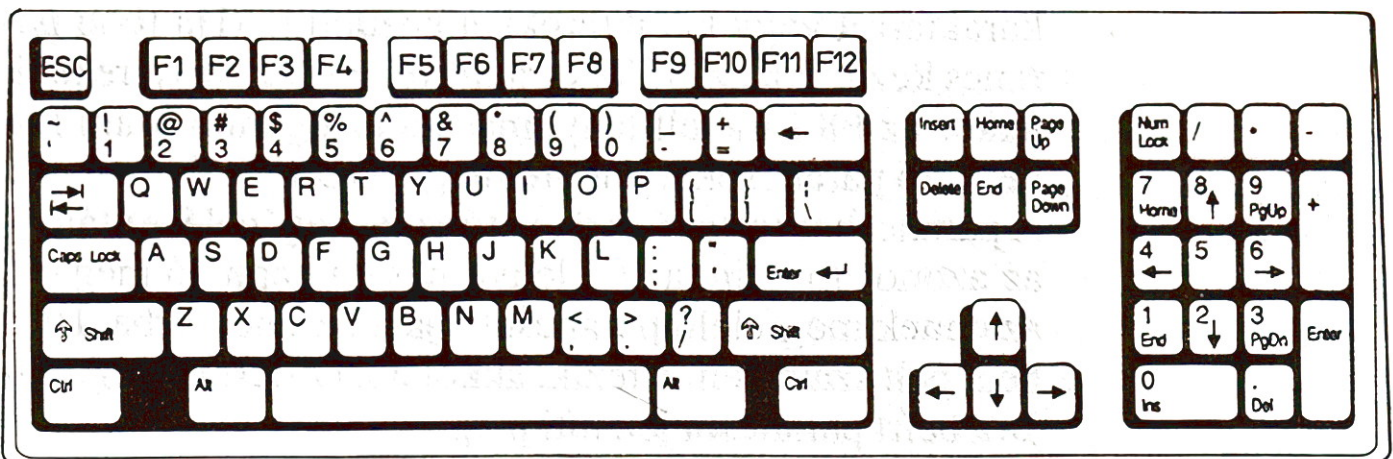
C:\>Line number:1 <ENTER>

Erre a parancssorban a VER parancs jelenik meg:

C:\>VER

Ha a parancsot futtatni akarjuk, csak le kell nyomnunk az ENTER billentyűt.

A DOSKEY program megkönnyíti a parancssor törlését is. A DOS ehhez a művelethez csak a BACKSPACE billentyűt biztosítja, ha azonban a DOSKEY programot használjuk, akkor az ESC gomb lenyomására az egész parancssor törlését érhetjük el.



Az ESC billentyű

Ha a végrehajtott parancsok száma húsznál is több — vagyis az összes parancs nevének listája már nem fér el egyszerre a képernyőre —, akkor az előbbi módszerek már nehézkesek. Ilyenkor hasznos segítséget jelent az F8 funkcióbillentyű. Használata a következő: írjunk be a parancs nevéből egy vagy



több betűt a parancssorba, majd nyomjuk le az F8 gombot! Ennek hatására a DOSKEY végignézi a parancsneveket tartalmazó pufferét és abban olyan parancsot keres, amely a beírt karakterrel vagy karakterekkel kezdődik. Például írjuk be a parancssorba a T karaktert, majd nyomjuk le az F8 gombot. Erre a DOSKEY a TIME parancsot fogja a pufferéből előhívni, tehát a parancssorba a következőket írja:

```
C:\>TIME
```

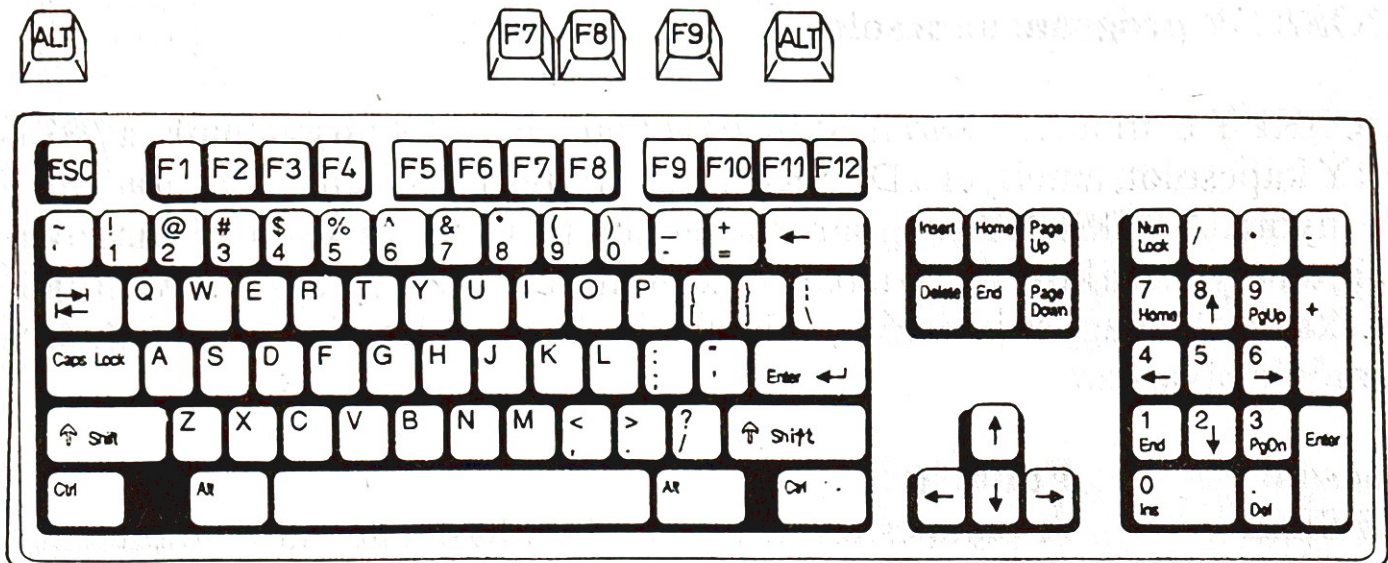
Különösebb zavart az sem okoz, ha a DOSKEY pufferében több olyan parancs is van, amely az F8 használata előtt beírt karakterrel vagy karakterekkel kezdődik. Az F8-at ismételten lenyomva a DOSKEY a következő olyan parancsot írja a parancssorba, amelyet a pufferben tárol és amely a megadott karakterrel (karakterekkel) kezdődik.

Munkánk során előfordulhat, hogy teljesen új feladatba akarunk fogni, így a DOSKEY által tárolt parancsnevekre már a továbbiakban nincs szükségünk. Az ALT—F7 kombinációt lenyomva a DOSKEY pufferének törlését érjük el.

A következő táblázatban a DOSKEY eddig bemutatott szolgáltatásait foglaljuk össze.

<b>Funkcióbillentyű</b>	<b>Művelet</b>
F7	A pufferben tárolt parancsok nevének megjelenítése a hozzájuk tartozó azonosító számmal együtt.
F8	A pufferben tárolt azon parancsnév megjelenítése, amelynek neve az F8 lenyomását megelőzően begépelte karakterrel vagy karakterekkel kezdődik. (Ha több parancs kezdődik az előírt karakterrel vagy karakterekkel, akkor az F8 ismételt lenyomására az egymás utáni következő parancsokat jeleníti meg a DOSKEY.)
F9	A parancshoz tartozó azonosító szám begépelése után — az azonosító szám az F7 lenyomására kapható meg —, az ennek megfelelő parancsot írja a parancssorba. Ha a begépelte szám nem létezik, akkor a DOSKEY a legutoljára beírt parancsot jeleníti meg.
ESC	Az éppen aktuális parancssor törlése.
ALT—F7	Az aktuális puffertartalom törlése.

A DOSKEY a parancssor szerkesztését az F1—F6 funkcióbillentyűkön, valamint a DEL, INS és BACKSPACE gombokon kívül további szerkesztőbillentyűkkel támogatja. A következő táblázatban ezeket a lehetőségeket foglaltuk össze.



Az F7—F9 funkcióbillentyűk és az ALT billentyű

### Billentyű

JOBBRA NYÍL

BALRA NYÍL

CTRL—JOBBRA NYÍL

CTRL—BALRA NYÍL

INS

DEL

HOME

END

### Művelet

A kurzort egy karakterpozícióval jobbra lépteti.

A kurzort egy karakterpozícióval balra lépteti.

A kurzort egy szóval jobbra lépteti.

A kurzort egy szóval balra lépteti.

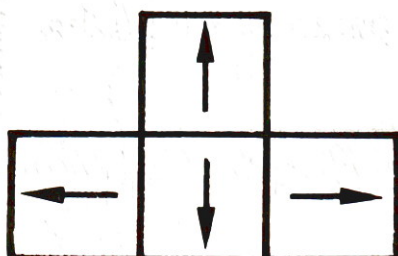
A beszúró és felülíró mód közötti átkapcsolás.

Karakter törlése a kurzor aktuális pozíciójában.

A kurzort a parancs első karakterére állítja.

A kurzort a parancs utolsó karakterére állítja.

INS	HOME	PG UP
DEL	END	PG DN



A DOSKEY szerkesztőbillentyűi

**Megjegyzés:** A DOSKEY segédprogram telepítése nélkül a JOBBRA nyíl az F1, a BALRA nyíl pedig a BACKSPACE gombbal azonos funkciót lát el. A DOSKEY telepítése után azonban az ezekhez a billentyűkhöz tartozó funkciók a táblázatban leírtaknak megfelelően megváltoznak.

## A DOSKEY program kapcsolói

A DOSKEY bemutatása során eddig egyetlen kapcsolót tárgyaltunk, a /HISTORY kapcsolót, amellyel a DOSKEY parancspufferének tartalma íratható a képernyőre. A DOSKEY programmal azonban további kapcsolók is használhatók, amelyeket a következő táblázatban tömören összefoglalunk. (A zárójelben a kapcsoló magyar jelentése, valamint — ha ilyen létezik — a rövidítésének módja olvasható.)

<b>Kapcsoló</b>	<b>Funkció</b>
/BUFSIZE (pufferméret)	A parancsnevek tárolásához használt puffer méretének előírását teszi lehetővé bájtokban. (Alapértelmezés: 512 bájt. Használata: DOSKEY /BUFSIZE = 1024.)
/HISTORY (történelem)	A parancsnevek tárolásához használt puffer tartalmát a képernyőre írja. (Rövidítése: /H)
/INSERT (beszúrás)	A beszúrás üzemmód beállítása alapértelmezésként.
/MACROS (makrók)	Az aktuális DOSKEY makródefiníciók megjelenítése. (Rövidítése: /M)
/OVERSTRIKE (felülírás)	A beszúrás mód letiltása.
/REINSTALL (újrategyítés)	A DOSKEY programból új másolatot telepít a memóriába.

### A DOSKEY program

A DOSKEY program az előzőekben végrehajtott parancsnevek visszahívását teszi lehetővé. A számítógép RAM memóriájában egy átmeneti tárolót (puffert) létesít, és a parancssorba begépelte és futtatott parancsokat itt tárolja. Ezen puffer méretétől függ tehát a DOSKEY-vel tárolt parancsok száma. A pufferben tárolt parancsok megjelenítésére a DOSKEY számos lehetőséget biztosít. Ezek közül a legegyszerűbb a FÖL és a LE kurzormozgató gombok használata, amelyekkel a pufferben tárolt parancsok lapozhatók végig.

## *A DOSKEY utasítással létrehozható makrók*

A DOSKEY program egyik leghasznosabb szolgáltatása a makrokészítés, amellyel a gyakran használt és hosszú DOS parancsnevek néhány karakterrel rövidíthetők. Ezek a makrók az EGYSZERŰEN DOS c. könyvben megismert parancsállományokra hasonlítanak, mivel mindegyikük azonosító névvel kell hogy rendelkezzen, amelyet a futtatáskor a parancssorba kell begépelni, és az ENTER gombot kell lenyomni. A parancsállományokhoz képest azonban van néhány lényeges eltérés is. Lássuk, melyek ezek!

- A DOSKEY a makrókat a RAM memóriában tárolja, így ezeket a DOS a belső parancsokkal azonos gyorsasággal tudja futtatni, ha azonban kikapcsoljuk a számítógépet, vagy a rendszert újból indítjuk, az addig definiált makróinkat is elveszítjük.
- A DOSKEY makrói csak egyetlen sorból állhatnak, de ebben a sorban több utasítás is elhelyezhető.

Minden további magyarázat helyett készítsük el a VERZIO elnevezésű makrót, amely a CLS és a VER parancsot fogja végrehajtani:

```
C:\>DOSKEY VERZIO=CLS $T VER <ENTER>
```

A makro definiálásához tehát a DOSKEY parancsnév után a makro azonosító nevét kell megadni, majd pedig az egyenlőségjel után azt a parancsot vagy parancsokat, amelyeket a makro nevének begépelésével futtatni akarunk. Az egyes parancsok nevét a \$T karakterekkel kell elválasztani egymástól.

A makrókat ugyanolyan egyszerű törölni, mint definiálni. Tegyük fel, hogy az előzőekben létrehozott VERZIO makróra már nincs szükségünk. Ekkor a parancssorban a következőket kell beírunk:

```
C:\>DOSKEY VERZIO= <ENTER>
```

Ha viszont valamennyi makrót egyszerre törölni akarjuk, akkor alkalmazzuk az ALT—F10 billentyűkombinációt!

Fontos megjegyezni, hogy a DOSKEY programmal létrehozott makrók a velük azonos nevű belső és külső DOS parancsokhoz, valamint parancsállományokhoz képest elsőbbséget élveznek. Például ha a kezdő felhasználót meg akarjuk akadályozni abban, hogy a FORMAT utasítással végzetes károkat okozzon a merevlemezünkön, készítsük el a következő makrót!

```
C:\>DOSKEY FORMAT=ECHO A FORMAT most nem működik!
```

Ezt követően a **FORMAT** utasítás futtatására a következő képernyő jelenik majd meg:

```
C:\>ECHO A FORMAT most nem működik!  
A FORMAT most nem működik!
```

## Fontos tudnivalók a makrókkal kapcsolatban:

Mint látható, a **DOSKEY** programmal nagyon egyszerű a makrók definiálása és létrehozása, van azonban néhány olyan alapvető tudnivaló, amelyet a felhasználóknak mindenképpen érdemes jól megjegyezniük!

1. A **DOSKEY** a makrókat ugyanabban a memóriapufferben tárolja, amelyet a végrehajtott parancsok tárolásához használ. Ha tehát sok makrót akarunk használni, akkor ezen puffer méretét a **/BUFSIZE** kapcsolóval növeljük meg!
2. Egymásba ágyazott makrók létesítésére nincs lehetőség, vagyis az egyik makro nem hívhatja meg a másik makrót!
3. A **DOSKEY** makrókat csak közvetlenül a billentyűzetről lehet futtatni. A makrók tehát parancsállományokból nem hajthatók végre!
4. A gyakran használt makrók definiálását érdemes az **AUTOEXEC.BAT** állományban elvégezni, így azok a rendszer elindítása után automatikusan és mindig a már megszokott módon állnak rendelkezésünkre.
5. A már elkészített makro átszerkesztése annak a **DOSKEY** parancsnak a módosítását jelenti, amellyel azt létrehoztuk. Ha tehát a makrót a **DOSKEY** a pufferében tárolja, akkor jelenítsük meg — felhasználva persze a rendelkezésre álló szerkesztési lehetőségeket — és végezzük el a módosítást. Ha a makrót parancsállomány tartalmazza, akkor végezzük el a parancsállományban a megfelelő módosítást, majd futtassuk újból a parancsállományt!

## A parancsállományok és a DOSKEY makrók összehasonlítása:

Mivel a DOSKEY utasítással létrehozható makrók a legjobban az egyetlen sort tartalmazó parancsállományokhoz hasonlíthatók, a következőkben a parancsállományokat és a makrókat hasonlítjuk össze.

	<b>DOSKEY makro</b>	<b>Parancsállomány</b>
Létrehozása	DOSKEY Név=parancsnév <ENTER>	Szövegszerkesztővel vagy a COPY CON utasítással
Törlése	DOSKEY Név= <ENTER>, vagy ALT—F10, ami valamennyi makrót törli	DEL(ETE) utasítás
Futtatása	Nevének beírása és <ENTER> (Csak a billentyűzetről futtatható.)	Nevének beírása és <ENTER>
Tárolásának helye	RAM memória	Lemez
Tartalma	Egyetlen sorban elhelyezett 1 vagy több DOS parancs	DOS parancsok, amelyeket egymás alatt külön-külön sorban kell elhelyezni

## *Ismétlés = Kulcs a sikerhez*

A DOS a parancssorban a gépelési hibákat többféle módon engedi javítani. Az ENTER gomb lenyomása előtt a karaktereket a BACKSPACE vagy a BALRA kurzormozgató gombbal törölhetjük.

Az F1—F4 funkcióbillentyűkkel azt a lehetőséget használhatjuk ki, hogy a DOS a parancssor tartalmát egy átmeneti tárolóban — pufferben — is elhelyezi. A leggyakrabban használt funkcióbillentyű az F1 és az F3, amelyek lenyomására a DOS a kurzor éppen aktuális pozíciójától kezdve egy, ill. az összes hátralevő karaktereket az új parancssorba írja.

Az INS a beszúró és felülíró mód közötti átkapcsolást biztosítja a pufferből a parancssorba másolás során, míg a DEL a kurzor éppen aktuális pozíciójában levő karaktert törli a pufferben.

A DOSKEY az 5.0-s DOS egyik segédprogramja, amely a parancssor szerkesztését, a korábban beírt és futtatott DOS parancsok éppen aktuális parancssorba írását, valamint egysoros parancsállományokhoz hasonló makrok definiálását támogatja.

## *Mit jelent?*

**Makro.** A felhasználó által definiált olyan karaktersorozat, amelyet futtatásakor a rendszer sokkal hosszabb karaktersorozattal cserél fel. A megvalósítástól függően a makronév parancsokat, de akár egész programrészeket is helyettesíthet.

**Puffer.** A számítógép RAM memóriájában létesített átmeneti adattároló hely. A puffereket leggyakrabban az adatbeviteli és adatkiviteli műveletek létesítik és használják.

# Állománykezelő parancsok haladóknak

**Már tanultuk:** COPY, helyettesítő karakterek (\*, ?), írásvédelmi címke és fül, sáv és szektor.

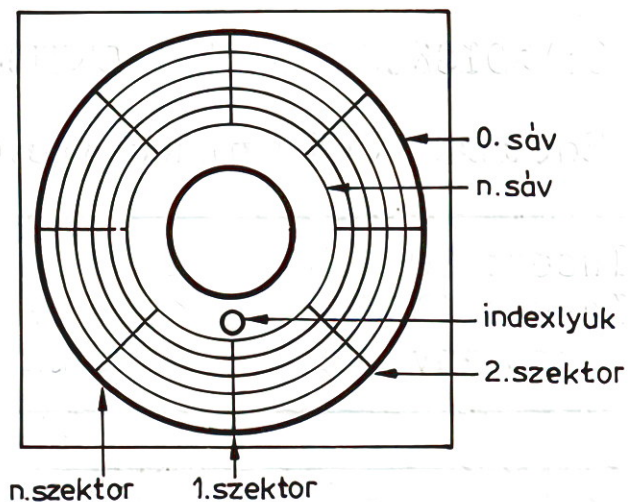
A **COPY paranccsal** az állományok az egyik lemezről vagy meghajtóról a másikra másolhatók. A parancs használatakor a másolandó állomány nevét (forrás) és a másolás céljául kijelölt helyet (cél) egyaránt elő kell írni.

A csillag (\*) a leggyakrabban használt **helyettesítő karakter**, amelynek hatására a DOS az előírt és az azt követő összes karakterpozícióban levő karaktereket figyelmen kívül hagyja akkor, amikor az előírásnak megfelelő állományneveket keres.

A kérdőjel (?) helyettesítő karakter arra utasítja a DOS-t, hogy az előírásnak megfelelő állománynevek keresésénél csak azt a karaktert ne vegye figyelembe, amelynek pozíciójában a kérdőjel áll. A csillag és a kérdőjel helyettesítő karakterek az állománynéven belül vegyesen is használhatók.

Az 5 1/4-es lemezen írásvédelmi bemetszés található, amelyet ha az **írásvédő címkével** leragasztunk, akkor a lemez tartalmának törlését, kiegészítését vagy megváltoztatását lehetetlenné tesszük. A 3 1/2-es lemez írásvédelmi lyukat tartalmaz, amely egy kis műanyag csúszka mozgatásával lezárható vagy kinyitható. A meghajtó csak akkor rögzíthet a lemezre információkat, ha a lyuk lezárt állapotban van. Ha viszont a lyuk nyitott, akkor a meghajtó a lemez tartalmát nem tudja megváltoztatni. Ezek tehát az ún. mechanikai írásvédelem eszközei.

A DOS a hanglemez barázdáihoz hasonló koncentrikus körökön, ún. sávokon, az ezek felosztásával keletkező és tortaszetre hasonlító szektorokban tárolja a lemezeken az információt. A következő ábrán a szektorok és a sávok elhelyezkedése látható lemezen:



Sávok és szektorok



# Lemezmásolat készítése a DISKCOPY paranccsal

DOS
KÜLSŐ

Programcsomagok (pl. WORD, FRAMEWORK stb.) megvásárlása után a legelső lépések egyike kell hogy legyen az eredeti lemez vagy lemezek biztonsági másolatainak elkészítése, vagyis a munkalemezek előállítás. Miért van erre szükség? Az eredeti lemezek védelmének érdekében. Ha ui. az eredeti (gyári) lemezeinket lemásoljuk, akkor azokat biztonságos helyre tehetjük el, és nem kell azokat a mindennapi munkánkba bevonnunk.

A DISKCOPY (lemezmásolat) paranccsal a biztonsági másolatok gyorsan és egyszerűen elkészíthetők, függetlenül attól, hogy az a számítógépes környezet, amelyben dolgozunk, egy vagy két hajlékonylemezes meghajtóval rendelkezik.

Ha annak a számítógépnek, amelyet használunk, két azonos méretű és kapacitású hajlékonylemez meghajtója van, akkor a másolandó lemezt helyezzük be az A meghajtóba — ez lesz a továbbiakban a forráslemez (*source*) —, és azt a lemezt, amelyre a másolatot akarjuk elkészíteni, a B meghajtóba — ez lesz a továbbiakban a céllemez (*target*). Az ennek megfelelő helyzet a következő oldalon lévő ábrán látható.

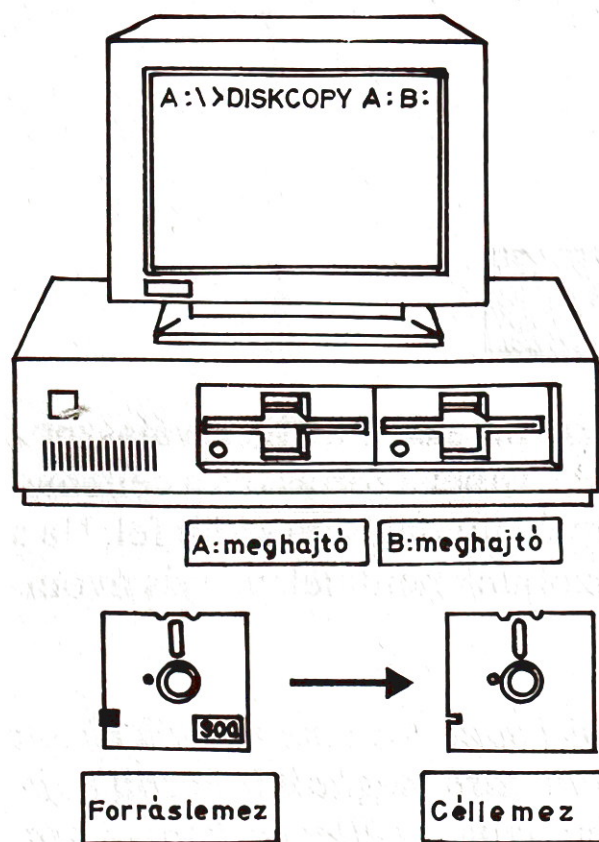
Ezután írjuk be a parancssorba a következőket:

```
C:\>DISKCOPY A: B: <ENTER>
```

Erre a DISKCOPY parancs a következőként válaszol:

<pre>Insert SOURCE diskette in drive A: Insert TARGET diskette in drive B: Press any key to continue...</pre>
---

<pre>Kérem a forráslemez behelyezését az A meghajtóba: Kérem a céllemez behelyezését a B meghajtóba: A folytatáshoz kérem bármelyik gomb lenyomását...</pre>
--



Lemezmasolat készítése kétmeghajtós rendszerben

**Fontos megjegyzések!** A DISKCOPY a céllemez tartalmát megkérdezés nélkül felülírja, és teljesen új lemezek esetében a lemez formázását is elvégzi. (A formázás műveletének végrehajtását a Formatting while copying — Formázás a másolás során — üzenet jelzi.) Különös gonddal járjunk el tehát akkor, amikor a lemezeket a meghajtókba helyezzük! Vagyis:

1. A másolandó eredeti lemezeinket minden esetben lássuk el mechanikai írásvédelemmel! Így még véletlenül sem tudjuk annak tartalmát felülírni, ha azt véletlenül a B meghajtóba helyezzük, vagyis ha céllemezként szerepeltetjük.
2. Mielőtt a lemezmasoláshoz hozzákezdénénk, gondoskodjunk megfelelő számú üres lemezről! Ha pedig olyan lemezt akarunk céllemezként használni, amely már információt tartalmazott, a DIR paranccsal előzetesen győződjünk meg annak tartalmáról!

A DISKCOPY egy billentyű lenyomására hozzákezd a lemezek másolásához, vagyis beolvassa az információt az A: meghajtóba helyezett lemezeről — ekkor az A: meghajtó jelzőfénye világít —, majd pedig ezt az információt a B: meghajtóba helyezett lemezre írja, ekkor a B: meghajtó jelzőfénye világít. Ezt

a műveletet mindaddig ismétli, amíg a másolással elkészül. Ekkor a következő üzenetet írja a parancssórhoz:

```
Copy another diskette (Y/N)?
```

```
Egy másik lemez másolása (igen/nem)?
```

Ha további lemezmásolatokat akarunk készíteni, akkor az igen válaszhoz nyomjuk le az Y billentyűt. Ekkor a DISKCOPY ismét a forrás- és a céllemez meghajtókba helyezése, majd egy billentyű lenyomására szólít fel. Ha a lemezmásolást be akarjuk fejezni, akkor válaszoljunk nemmel, vagyis nyomjuk le az N billentyűt.

**Fontos megjegyzés:** a DISKCOPY csak azonos kapacitású meghajtók között működik. Ha a parancsot mégis különböző kapacitású meghajtók között hajtjuk végre és a forráslemez kapacitása nagyobb, mint a céllemez kapacitása, akkor a DISKCOPY a

```
TARGET media has lower capacity than SOURCE
Continue anyway (Y/N)?
```

```
A CÉLKÉNT megjelölt médiának kisebb a kapacitása, mint a
FORRÁSÉ
Mindenképpen folytassam (I/N)?
```

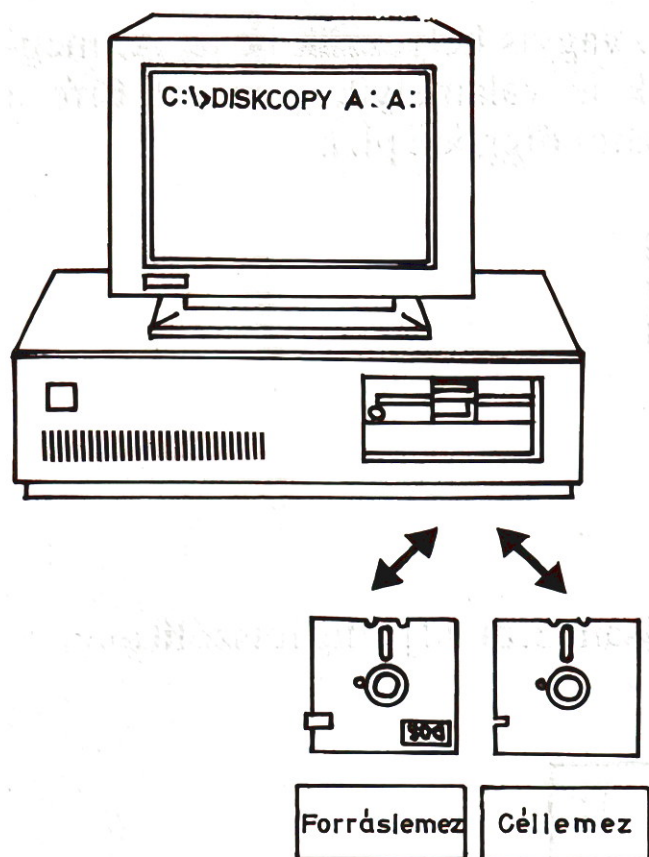
üzenetet küldi, és Y válasz esetén a másolásra kísérletet tesz. Ilyen esetben a DISKCOPY parancsot a következőképpen kell használni:

```
DISKCOPY A: A: <ENTER>
```

vagy

```
DISKCOPY B: B: <ENTER>
```

Ezek után azt az esetet tárgyaljuk, amikor a lemezmásoláshoz használt számítógép egyetlen hajlékonylemez meghajtóval rendelkezik. Az ennek megfelelő helyzet a következő ábrán látható.



*Lemezmásolat készítése egyetlen meghajtó esetén*

Ebben az esetben a DISKCOPY parancs a következőre módosul:

```
C:\>DISKCOPY A: A: <ENTER>
```

melyre a DISKCOPY parancs a következő üzenettel válaszol:

```
Insert SOURCE diskette in drive A:
Press any key to continue...
```

```
Kérem a forráslemez behelyezését az A meghajtóba:
A folytatáshoz kérem bármelyik gomb lenyomását...
```

**Megjegyzés:** A korábbiakban tárgyalt

```
DISKCOPY A: B:
```

*alakú parancs egyetlen hajlékonylemez meghajtóval rendelkező számítógépenél is használható.*

Járjunk el a felszólításnak megfelelően, vagyis helyezzük be az A: meghajtóba a másolandó lemezt, és nyomjuk le valamelyik gombot! Erre a DISKCOPY a másolandó lemez formátumától függően pl. a

```
Copying 40 tracks  
9 Sectors per Track, 2 Side(s)
```

```
40 sáv másolása  
9 szektor/sáv, 2 oldal
```

üzenet után hozzákezd az információ olvasásához, majd pedig felszólít a céllemez A: meghajtóba helyezésére:

```
Insert TARGET diskette in drive A:  
Press any key to continue...
```

Cseréljük ki tehát a meghajtóban a lemezeket, és nyomjuk le valamelyik gombot! Erre a DISKCOPY elvégzi a beolvasott információ lemezre írását. (A lemez méretétől és a lemezen tárolt adatok mennyiségétől függően a forrás- és a céllemez cseréjét esetleg többször is el kell végezni ahhoz, hogy a másolat elkészüljön.) A művelet végét azonban most is a következő üzenet jelzi:

```
Copy another diskette (Y/N)?
```

A szoftverek telepítőprogramjai a futásuk során rendszerint mindig egy adott lemez meghajtóba helyezésére szólítanak fel. Ezért nagyon fontos, hogy az elkészített biztonsági másolatot az eredeti lemezen olvasható címkefelírral lássuk el (pl. SYSTEM1 DISK stb.). Először írjuk meg a címkéket és csak azután ragasszuk fel a lemezre! A már lemezre ragasztott címkére írva ui. a hajlékonylemezen maradandó károsodásokat és adatvesztéseket okozhatunk!

## A DISKCOPY parancs

A DISKCOPY (lemezmásolás) külső DOS parancs, amellyel lemezmásolatok készíthetők. Ha a rendszer, amelyben a lemezmásolást végezzük, két hajlékonylemezes meghajtóval rendelkezik, akkor a következőképpen kell a DISKCOPY parancsot futtatnunk:

```
C:\>DISKCOPY A: B: <ENTER>
```

Ha rendszerünknek csak egyetlen hajlékonylemezes meghajtója, vagy két különböző kapacitású meghajtója van, akkor az előző parancs a következőképpen módosul:

```
C:\>DISKCOPY A: A: <ENTER>
```

*A leggyakoribb hibák a DISKCOPY parancs használata során:*

1. A rendszer a DISKCOPY parancsot — mivel az külső parancs — nem találja.  
*Védekezés:*  
— Ha nincs a parancs a merevlemezen, másoljuk rá a DOS lemezről!  
— A parancs futtatása előtt váltsunk át abba az alkönyvtárba, amely a parancsot tartalmazza, vagy írjuk elő a parancsot tartalmazó alkönyvtár elérési útját a PATH utasításban!
2. A forrás- és a céllemez összecserélése.  
*Védekezés:* A másolandó lemezeket fizikai írásvédelemmel kell ellátni.
3. A céllemez eredeti tartalmának akaratlan felülírása.  
*Védekezés:* A DISKCOPY parancshoz használjunk teljesen új, még formázatlan lemezt — a DISKCOPY szükség esetén a formázást is elvégzi —, vagy a művelet elvégzése előtt kérdezzük le a céllemez tartalmát.
4. Az elkészült másolatok lemezcímkéjének üresen hagyása.  
*Védekezés:* Amint a lemezmásolással elkészültünk, a lemezdobozban található üres címkék közül lássunk el egyet a forráslemezen olvasható felirattal, majd ragasszuk rá a másolatra.
5. A DISKCOPY-val együtt használható /V kapcsoló előírása esetén a parancs a másolás helyességét is ellenőrzi.

## *Lemezek tartalmának összehasonlítása a DISKCOMP paranccsal*

DOS

KÜLSŐ

A DISKCOMP (lemeztartalom összehasonlítás) parancs annyiban hasonlít az előzőekben tárgyalt DISKCOPY parancshoz, hogy két lemezzel működik, ez a parancs azonban a két lemez tartalmát hasonlítja össze szektorról szektorra haladva. A legtöbb felhasználó a DISKCOMP parancsot közvetlenül a lemez-másolat elkészülte után hajtja végre, így győződve meg a másolási művelet sikeres elvégzéséről. Ha a DISKCOMP a két lemez között eltérést talál, akkor a következő üzenetet írja a képernyőre:

```
Compare error on  
side n, track nn
```

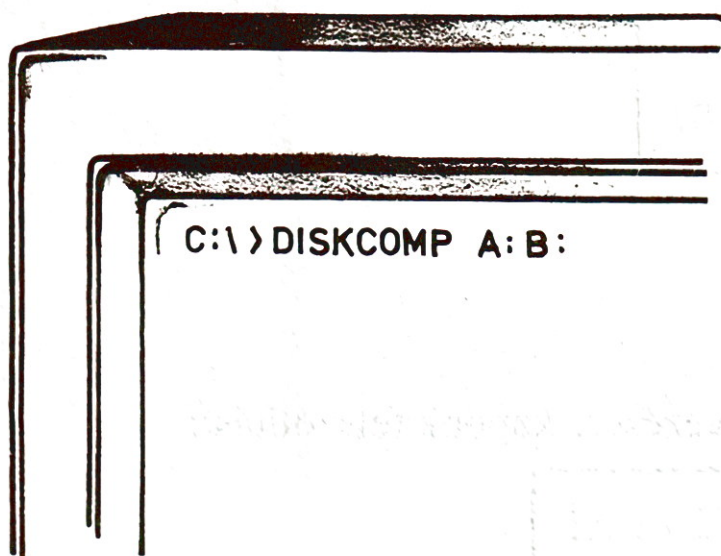
```
Összehasonlítási hiba az  
n-edik oldalon, az nn-edik sávban
```

Ha viszont a két lemez azonos, akkor ezt a parancs a következő üzenettel jelzi:

```
Compare OK  
Compare another diskette (Y/N)?
```

```
Az összehasonlítás rendben  
Egy másik lemez összehasonlítása (I/N)?
```

Ha tehát az összehasonlítást egy további lemezpár között folytatni akarjuk, akkor a kérdésre Y-nal, ellenkező esetben pedig N-nel válaszoljunk. Ezek után lássuk, hogyan kell a DISKCOMP parancsot futtatni!



*A DISKCOMP használata két hajlékonylemezes rendszerben*

Ha az a számítógépes környezet, amelyben dolgozunk, két azonos kapacitású meghajtóval rendelkezik, akkor a DISKCOMP parancsot a következőképpen kell indítanunk:

```
C:\>DISKCOMP A: B: <ENTER>
```

amelyre a DISKCOMP parancs a következő üzenettel válaszol:

```
Insert FIRST diskette in drive A:
Insert SECOND diskette in drive B:
Press any key to continue...
```

Kérem az ELSŐ lemez behelyezését az A meghajtóba:  
Kérem a MÁSDIK lemez behelyezését a B meghajtóba:  
A folytatáshoz kérem bármelyik gomb lenyomását...

Ha a számítógép, amellyel dolgozunk, csak egyetlen hajlékonylemezes meghajtóval rendelkezik, akkor a DISKCOMP a következőképpen futtatható:

```
C:\>DISKCOMP A: A: <ENTER>
```

amelyre a következő üzenetet kapjuk:

```
Insert FIRST diskette in drive A:
Press any key to continue...
```

Járjunk el a felszólításnak megfelelően! Ekkor a következő üzenet lesz olvasható a képernyőn:



```
Comparing 40 tracks  
9 sectors per track, 2 side(s)
```

```
40 sáv összehasonlítása  
9 szektor/sáv, 2 oldal
```

majd a második lemez meghajtóba helyezésére kapunk felszólítást:

```
Insert SECOND diskette in drive A:  
Press any key to continue...
```

Nyomjuk le tehát a kérésnek megfelelően valamelyik gombot. Ekkor a parancs az összehasonlítást elvégzi és a művelet eredményét a képernyőre írja.

**Fontos megjegyzés:** Előfordulhat, hogy a DISKCOMP a két lemezt különbözőnek érzékeli, bár azok ugyanazokat az állományokat tartalmazzák. Mint az előzőekben már említettük, a DISKCOMP a lemezek összehasonlítását szektorról szektorra haladva végzi el. Ha viszont a lemezek állományainak másolását a korábbiakban már megismert COPY paranccsal végeztük el, nem biztos, hogy az egyes szektorok tartalma a két lemez esetében azonos, még akkor sem, ha az állományok másolása egyébként sikeres volt.

## A DISKCOMP parancs

A DISKCOMP (lemezösszehasonlítás) külső DOS parancs, amely két lemez tartalmát hasonlítja össze szektorról szektorra haladva. Ha a parancs az összehasonlítás során eltérést talál, akkor annak helyét — oldal, sáv — kijelzi a képernyőre. A következő parancs az A: meghajtóban levő lemezt a B: meghajtóban levő lemezzel hasonlítja össze:

```
C\>DISKCOMP A: B: <ENTER>
```

Ha a parancsot egymeghajtós rendszerben használjuk, akkor az előbbi parancs a következőkre módosul:

```
C\>DISKCOMP A: A: <ENTER>
```

A leggyakoribb hibák a *DISKCOMP* parancs használata során:

1. A rendszer a *DISKCOMP* parancsot — mivel az külső parancs — nem találja.

Védekezés:

- Ha nincs a parancs a merevlemezen, másoljuk rá a DOS lemezről!
- A parancs futtatása előtt váltsunk át abba az alkönyvtárba, amely a parancsot tartalmazza, vagy írjuk elő a parancsot tartalmazó alkönyvtár elérési útját a *PATH* utasításban!

2. A *DISKCOMP* használata *COPY*-val másolt lemezek, ill. állományok esetében.

Védekezés: A *DISKCOMP*-ot csak akkor használjuk, ha a másolást a *DISKCOPY* paranccsal végeztük.

## Állományok másolása az *XCOPY* paranccsal

3.2
-----

KÜLSŐ
-------

Ez ideig csak olyan parancsokat tárgyaltunk, amelyek teljes lemezek tartalmának másolásával, ill. tartalmának összehasonlításával foglalkoztak. A napi munka során azonban a leggyakrabban egyes állományok vagy állományok bizonyos csoportjainak másolását kell elvégeznünk. Ezen művelethez eddig csak a *COPY* parancsot használtuk. Ha még hatékonyabban szeretnénk dolgozni, akkor mindenképpen érdemes megismerkednünk az *XCOPY* paranccsal is, amellyel

— az állomány létrehozásának és módosításának dátuma, vagy archív attribútuma alapján szelektív másolás hajtható végre;

— az állományok másolásával együtt a hozzájuk tartozó könyvtárszerkezet másolása is elvégezhető;

— sok állomány biztonságosan másolható a merevlemezről a hajlékonylemezre.

**Megjegyzés:** a DOS-ban minden egyes állományhoz ún. attribútumbajt tartozik, amely az állományra vonatkozó speciális információkat (pl. csak olvasható, archív) tárolja. Már a COPY CON-nal létrehozott állomány is rendelkezik attribútummal, amely az archív. Az archív attribútumot a rendszer azon állományok megjelölésére használja, amelyekről még nem készült biztonsági másolat, vagyis az újonnan létrehozott, ill. a biztonsági másolat elkészítése óta módosított állományok megjelölésére. Az XCOPY parancson kívül még a 4. Fejezetben tárgyalt BACKUP/RESTORE parancspár is felhasználja az archív attribútumot. Az attribútumok fajtáit, az attribútumbajt lekérdezését és módosítását a 3. Fejezetben részletesen tárgyaljuk.

Az XCOPY, ha azt a kapcsolói nélkül használjuk — a szelektív állománymásolást összesen nyolc különböző kapcsoló teszi lehetővé! —, a COPY paranccsal teljesen azonos módon működik, vagyis a forrásállományt a célállományba vagy a célként megjelölt helyre másolja. A parancs alkalmazásakor a forrásállományt és a célállományt vagy célhelyet egyaránt elő kell írni. A következő XCOPY parancs pl. a merevlemez WP alkönyvtárában levő RIPORT.LTR állományt az A: meghajtóban levő lemezre másolja:

```
C:\WP>XCOPY RIPORT.LTR A: <ENTER>
```

Az XCOPY parancs a helyettesítő karakterek használatát a COPY paranccsal azonos módon teszi lehetővé. Lássunk erre is egy példát! A következő parancs a merevlemez WP alkönyvtárában levő .LTR kiterjesztésű állományokat másolja az A: meghajtóban levő lemezre:

```
C:\WP>XCOPY *.LTR A: <ENTER>
```

A továbbiakban ismerkedjünk meg az XCOPY legfontosabb kapcsolóival! Elsőként a /S kapcsolót mutatjuk be, amelynek megadása esetén a DOS nemcsak az előírt, hanem az esetlegesen alatta elhelyezkedő alkönyvtárakat is létrehozza a célként előírt lemezen, és az ezekben található összes állomány másolását elvégzi, ha erre a megfelelő helyettesítő karakterekkel — \*.\* — utasítást adunk. Lássunk mindjárt egy példát a kapcsoló használatára! Tegyük fel, hogy az a lemez, amelyet az A: meghajtóba helyeztünk el, és a merevlemezre már meglévő és a gyökérkönyvtár alatt elhelyezkedő WP alkönyvtárba akarunk másolni, a következő könyvtárszerkezettel rendelkezik:

INDITO.DOC

CIMKE	RIPORT
CIMKE1.LTR	RIPORT_1.DOC
CIMKE2.LTR	RIPORT_2.DOC
CIMKE3.LTR	

```
C: \ > XCOPY A: *.* C:\WP\*.* /S /E
```

*Állományok másolása a könyvtárszerkezettel együtt*

Ha tehát ezeket az állományokat egyszerre és a könyvtárszerkezet megtartásával akarjuk a merevlemez WP alkönyvtárába másolni, akkor a következő előírást kell alkalmaznunk:

```
C: \> XCOPY A: *.* C:\WP /S <ENTER>
```

Az XCOPY az ENTER lenyomása után a másolás elvégzéséről következőként tájékoztat:

```
Reading source file(s) ...
A:\INDITO.DOC
A:\CIMKE\CIMKE1.LTR
A:\CIMKE\CIMKE2.LTR
A:\CIMKE\CIMKE3.LTR
A:\RIPORT\RIPORT 1.DOC
A:\RIPORT\RIPORT 2.DOC
6 File(s) copied
```

```
A forrásállomány(ok) olvasása...
.
.
.
6 állomány átmásolva
```

Győződjünk meg a másolási művelet elvégzésének eredményéről. Kérdezzük le tehát a C : meghajtó tartalmát!

```
C:\>DIR <ENTER>
```

A képernyőn valami hasonlóknak kell lennie:

```
Volume in drive C has no label
Volume Serial Number is 3210-12F4
Directory of C:\
DOS          <DIR>      01-01-91  12:00
CONFIG      SYS           145  02-01-91  13:15
AUTOEXEC    BAT           159  02-01-91  17:30
WP          <DIR>      04-04-93  10:25
4 file(s)   304 bytes
3423456 bytes free
```

Ha most a WP alkönyvtárba belépünk

```
C:\>CD WP <ENTER>
```

és a tartalmát a DIR-rel lekérdezzük

C:\WP>DIR <ENTER>

akkor a

```

.          <DIR>          04-04-93  10:25
..         <DIR>          04-04-93  10:25
INDITO    DOC            12      04-04-93  10:25
CIMKE     <DIR>          04-04-93  10:25
RIPORT    <DIR>          04-04-93  10:25
          5 file(s)          12 bytes
          3423456 bytes free
    
```

fogjuk ott megtalálni.

**Megjegyzés:** Ha a másolás célhelyeként megjelölt C: meghajtó még a WP alkönyvtárat sem tartalmazza, és a művelet előírásakor a WP után nem adjuk meg a \jelet, akkor az XCOPY a következő üzenetet írja a képernyőre:

```

Does WP specify a file name
or directory name on the target
(F=File D=Directory)?
    
```

A WP a célként megjelölt helyen állománynevet vagy könyvtárnevet jelent (F=állomány D=könyvtár)?

*Mivel a WP könyvtárnév, ezért D-vel kell válaszolnunk.*

A /D kapcsoló előírásakor a dátumbélyeget hasznosíthatjuk a másolandó állományok körének előírásakor. A következőkben egy olyan utasítást mutatunk be, amely az A: meghajtóban elhelyezett lemezről csak az 1993. január 31-én, vagy ezután létrehozott vagy módosított állományokat másolja a merevlemez gyökérkönyvtárába:

C:\>XCOPY A:\*.\*/D:01-31-93

(Figyeljük meg, hogy a dátumot — a DATE paranccsal összhangban — hh-nn-éé sorrendben kell megadni!)

Ha a másolandó állományok körét sem helyettesítő karakterekkel, sem a létrehozás vagy módosítás dátuma alapján nem tudjuk előírni, ill. leszűkíteni, akkor nagyon hasznos szolgálatot tehet az XCOPY parancs /P kapcsolója. Ennek előírása esetén ui. mielőtt az XCOPY az adott állomány másolását elvégezné, annak nevét a képernyőre írja. Pl.:

```
FILENEV.KIT (Y/N) ?
```

vagyis a másolás elvégzéséhez a felhasználótól egy végső visszaigazolást vár. Ha tehát az adott állományt át akarjuk másolni, akkor az üzenetre az Y, ellenkező esetben pedig az N gomb lenyomásával válaszoljunk.

Gyakori feladat, hogy a merevlemez egyik alkönyvtárának teljes tartalmát hajlékonylemezre vagy lemezekre akarjuk másolni. Ha azonban az alkönyvtár tartalma nagyobb, mint ami egyetlen hajlékonylemezen elfér, akkor a DOS a másolást megszakítja és az

```
Insufficient disk space
```

```
A lemezen nem elegendő a hely.
```

hibaüzenetet írja a képernyőre. Ebben az esetben csak külön nyomozással deríthető ki, hogy melyik volt az az állomány, amelynek másolását a rendszer már nem tudta elvégezni, ill. mely állományok vannak már a lemezen. Az XCOPY parancsot a /M kapcsolóval használva, valamint a másolandó állományok teljes körét előzetesen az ATTRIB utasítással archívra beállítva — egyébként is ez az alaphelyzet — egy adott alkönyvtár tartalmát akár több lemezre is biztonságosan és egyszerűen át tudjuk másolni. Lássuk tehát, hogyan kell ezt a feladatot elvégezni!

Tegyük fel, hogy a merevlemez MUNKA elnevezésű alkönyvtárából valamennyi állományt hajlékonylemezre akarjuk menteni. Ha szükséges, elsőként az ATTRIB parancs — amely ugyancsak külső DOS parancs — felhasználásával állítsuk be archívra ezen állományok attribútumát:

```
C:\>ATTRIB +A \MUNKA\ *.* <ENTER>
```

majd adjuk ki az XCOPY parancsot a /M kapcsolóval:

```
C:\>XCOPY \MUNKA\*.* A: /M <ENTER>
```

Az /M kapcsoló arra utasítja az XCOPY utasítást, hogy csak azoknak az állományoknak a másolását végezze el, amelyek archív attribútummal rendelkeznek. Ha az XCOPY az állományt sikeresen átmásolta, akkor a forráshelyen törli annak archív attribútumát. Ha a hajlékonylemez megtelt, akkor a DOS az

Insufficient disk space

hibaüzenetet küldi. Ekkor a hajlékonylemezt új lemezzel kell kicserélnünk a meghajtóban, majd ismét a

```
C:\>XCOPY \MUNKA\*.* A: /M <ENTER>
```

parancsot kell végrehajtanunk. Mivel az XCOPY a /M kapcsoló használata esetén a már átmásolt állományok archív attribútumát törölte, pontosan tudja, hogy melyik állománytól kezdve kell a másolást folytatnia.

A következő táblázatban — a teljesség kedvéért — az XCOPY paranccsal használható összes kapcsolót megadjuk.

**Kapcsoló      Művelet**

- /A** Arra utasítja az XCOPY parancsot, hogy csak azokat az állományokat másolja a célként megjelölt helyre, amelyek archív attribútummal rendelkeznek.
- /D:hh-nn-éé** Arra utasítja az XCOPY parancsot, hogy csak a megadott, vagy az utáni dátumbélyeggel rendelkező állományokat másolja a célként megjelölt helyre.
- /E** Arra utasítja az XCOPY parancsot, hogy a célként megadott könyvtárban még akkor is hozza létre a forráshely könyvtárszerkezetét, ha az üres. (Csak a /S kapcsolóval együtt használható.)
- /M** Arra utasítja az XCOPY parancsot, hogy csak azokat az állományokat másolja a célként megjelölt helyre, amelyek archív attribútummal rendelkeznek, valamint sikeres másolás esetén törölje a forráshelyen a már átmásolt állományok archív attribútumát.
- /P** Arra utasítja az XCOPY parancsot, hogy az állomány másolását csak az FILENEV.KIT (Y/N)? üzenetre adott Y, vagyis igen válasz esetén végezze el.



- /S** A kapcsoló előírásának hatására a XCOPY nemcsak azokat az állományokat másolja át a célként megjelölt helyre, amelyek az aktuális katalógusban helyezkednek el, hanem a könyvtárszerkezettel együtt azokat is, amelyek az aktuális alkönyvtár alatt helyezkednek el.
- /V** Arra utasítja az XCOPY parancsot, hogy az állomány átmásolása után a forrás és a célállomány tartalmát hasonlítsa össze, vagyis azonnal győződjön meg a másolás helyességéről.
- /W** A kapcsoló hatására az XCOPY a másoláshoz csak akkor kezd hozzá, ha a felhasználó egy billentyűt lenyomott.

*Leggyakoribb hibák az XCOPY parancs használata során:*

1. A rendszer az XCOPY parancsot — mivel az külső parancs — nem találja.  
*Védekezés:*
  - Ha nincs a parancs a merevlemezen, másoljuk rá a DOS lemezről!
  - A parancs futtatása előtt váltsunk át abba az alkönyvtárba, amely a parancsot tartalmazza, vagy írjuk elő a parancsot tartalmazó alkönyvtár elérési útját a PATH utasításban.
2. Az XCOPY a forrásként előírt állományt vagy állományokat nem találja.  
*Védekezés:*
  1. Ellenőrizzük, hogy helyesen írtuk-e a forrásállomány(ok) nevét.
  2. Ha az XCOPY parancsot nem abból az alkönyvtárból hajtjuk végre, amely a másolandó állományt tartalmazza, ezért a teljes — a másolandó állományhoz vezető útvonalat is tartalmazó — állománynév előírására van szükség.

## **Az XCOPY (kibővített másolás) parancs**

Az XCOPY a parancssorba előírható kapcsolói segítségével az állományok szelektív másolását teszi lehetővé a dátumbélyeg — /D kapcsoló — vagy az archív attribútum — a /A vagy a /M kapcsoló — alapján. A /P kapcsoló hatására a parancs a másolást csak akkor végzi el, ha a felhasználó az Y gombot lenyomja, vagyis igen választ ad. A /S az előírt célhely alatti könyvtárakba is másol, a /E hatására pedig a célként előírt helyen szükség esetén a forráshely üres alkönyvtárait is létrehozza. A /V az XCOPY-t arra utasítja, hogy az éppen átmásolt állományt a forrással összehasonlítsa, a /W hatására pedig csak akkor kezd a másoláshoz, ha a felhasználó a billentyűzeten valamelyik gombot lenyomta.

## *Két állomány tartalmának összehasonlítása a COMP utasítással*

DOS
-----

KÜLSŐ
-------

A COMP (összehasonlítás) parancs külső DOS parancs, amellyel két állomány tartalmát lehet összehasonlítani. A parancs első paramétereként az első, második paramétereként a második összehasonlítandó állomány nevét kell megadni.



```
A:\> COMP A.DAT B.DAT
```

### *Két állomány összehasonlítása*

Tegyük fel, hogy az A.DAT és a B.DAT állományokat akarjuk összehasonlítani. Ha az állományok azonosak, akkor ezt a COMP a következő üzenettel jelzi:

Comparing A.DAT and B.DAT Files compare OK Compare more files (Y/N)?
--

Az A.DAT és a B.DAT összehasonlítása Az összehasonlítás rendben További állományok összehasonlítása (I/N)?
--

Ha az összehasonlítást másik két állománnyal folytatni akarjuk, akkor Y, ellenkező esetben pedig N beírásával válaszoljunk. Y válasz esetén a parancs a következő üzenetet írja a képernyőre:

Name of first file to compare:

Az első összehasonlítandó állomány neve:

vagyis az első összehasonlítandó állomány nevének megadását várja. Amint az állomány nevét beírtuk és az ENTER billentyűt lenyomtuk, a következő üzenet válik láthatóvá:

Name of second file to compare:

A második összehasonlítandó állomány neve:

Írjuk be tehát a második állomány nevét is. Ezután az

Option:

felirat jelenik meg, amellyel a parancs kapcsolók beírására ad lehetőséget. Ha nem akarunk kapcsolót megadni, akkor egyszerűen nyomjuk le az ENTER gombot!

Ez eddig nagyon egyszerű. De mi történik akkor, ha a két állomány eltér egymástól? Ezt egy konkrét példán fogjuk végigkísérni.

Első lépésként létrehozzuk azt a két állományt, amelyet össze akarunk hasonlítani.

```
C:\>COPY CON A.DAT <ENTER>
abcde<ENTER>
^Z<ENTER>
  1 file(s) copied
C:\>
```

Tehát ez az állomány a következő karaktersorozatot fogja tartalmazni:

```
abcde<ENTER>
01234 5
```

Alul a karakterek bájtban mért relatív eltolási értékeit (OFFSET) jelöltük.

```
C:\>COPY CON B.DAT <ENTER>
abcd<ENTER>
^Z<ENTER>
  1 file(s) copied
C:\>
```

Tehát ez az állomány a következő karaktersorozatot tartalmazza:

```
abcd<ENTER>
01234 5
```

Futtassuk ezután a COMP parancsot a két állományra:

```
C:\>COMP A.DAT B.DAT <ENTER>
```

Erre a következő üzenetet olvashatjuk a képernyőn:

```
Comparing C:A.DAT and C:B.DAT
Compare error at OFFSET 2
File 1 = 63
File 2 = 62
Compare error at OFFSET 3
File 1 = 64
File 2 = 63
Compare error at OFFSET 4
File 1 = 65
File 2 = 64
```

Az C:A.DAT és C:B.DAT összehasonlítása

Eltérés a 2 eltolási értéknél

1. állomány = 63
2. állomány = 62

Eltérés a 3 eltolási értéknél

1. állomány = 64
2. állomány = 63

Eltérés a 4 eltolási értéknél

1. állomány = 65
2. állomány = 64

Értelmezzük a fenti üzenetet! A COMP az első eltérést a 2 eltolási értéknél találta, ami rendben is van, hiszen a két állományban az első eltérés a c és a b karakternél, vagyis a 2 eltolási értéknél jelentkeznek. Ettől kezdve a két állomány már végig eltérő, erre utal a két további üzenet. A parancs az eltérés helyének megadásán kívül a két eltérő karakter ASCII kódját is megadja. (A b, c, d és e betűk ASCII kódja 62, 63, 64, 65.)

*Fontos megjegyzések a COMP használatával kapcsolatban:*

1. Ha az 5.0-s DOS-nál alacsonyabb verziójú DOS-szal dolgozunk, akkor a két összehasonlítandó állománynak azonos méretűnek kell lennie. Ellenkező esetben ui. a parancs a

```
Files are different sizes
```

```
Az állományok különböző méretűek
```

hibaüzenetet küldi, és az összehasonlítást nem végzi el.

2. Az eltolási értékeket, vagyis az eltérések helyét — bájtban értelmezve —, valamint az eltérő karakterek kódjait hexadecimális, vagyis tizenhatos számrendszerben adja meg a parancs. (Az eltolási értékek számozása is nullától indul!)
3. Ha a COMP tíz eltérésnél többet talál, akkor az összehasonlítást befejezi, és a következő üzenetet írja a képernyőre:

```
10 Mismatches - endig compare
```

```
10 eltérés - az összehasonlítás befejezve
```

*A COMP utasítás kapcsolói*

(Csak DOS 5.0 használata esetén!)

A DOS 5.0-ban a COMP utasítással már 5 különböző kapcsoló használható, amelyek a felhasználó kényelmét, ill. az utasítás hatékonyabb használatát szolgálják. A következő táblázatban ezeket a kapcsolókat ismertetjük.

<b>Kapcsoló</b>	<b>Művelet</b>
/A	A kapcsoló előírásának esetén a COMP az eltéréseket karakteres formában — nem pedig az alapértelmezés szerinti hexadecimális ASCII kódban — jelzi a képernyőre.
/C	Arra utasítja a COMP parancsot, hogy a két állomány összehasonlításakor a kis- és a nagybetűk között ne tegyen különbséget.
/D	A kapcsoló előírása esetén a COMP az eltéréseket tizes számrendszerben — nem pedig az alapértelmezés szerinti hexadecimális rendszerben — jelzi a képernyőre.
/L	Arra utasítja a COMP parancsot, hogy az eltérés helyét ne az alapértelmezés szerinti eltolási értékkel, hanem az eltérést tartalmazó sor számával jelezze.
/N= sorok száma	Arra utasítja a COMP parancsot, hogy az összehasonlítást csak az egyenlőségjel után megadott mennyiségű sorban — az állomány első n sorában — végezze el.

### *Leggyakoribb hibák a COMP parancs használata során*

1. A rendszer a COMP parancsot, mivel az külső parancs, nem találja.

#### *Védekezés:*

A COMP parancs futtatása előtt váltsunk át abba az alkönyvtárba, amely a parancsot tartalmazza, vagy írjuk elő a parancsot tartalmazó alkönyvtár elérési útját a PATH utasításban.

2. A COMP az összehasonlítandó állományokat nem találja.

#### *Védekezés:*

1. Ellenőrizzük, hogy helyesen írtuk-e az állományok nevét!

2. Ha a COMP parancsot nem abból az alkönyvtárból hajtjuk végre, amely az összehasonlítandó állományokat tartalmazza, a teljes — az útvonalat is tartalmazó — állományazonosító előírására van szükség.

## **A COMP (összehasonlítás) parancs**

A COMP külső DOS parancs, amellyel két állomány tartalma hasonlítható össze. A parancs nemcsak az eltérés bájtban mért helyét (OFFSET), hanem az eltérő karakterek hexadecimális ASCII kódját is megadja. A következő parancs a REGI.TXT állományt az UJ.TXT állománnyal hasonlítja össze:

```
C:\>COMP REGI.TXT UJ.TXT
```

## Állományok aktualizálása a REPLACE parancssal

### 3.2

#### KÜLSŐ

Az állományokkal végzett munka során két olyan speciális helyzet is létezik, amelyet sem a COPY, sem az XCOPY parancs nem tud kezelni. Az első az, amikor a céllemezen már meglévő állományokat a forráslemez olyan állományaival akarjuk kicserélni/felülírni, amelyeknek azonos a nevük. Például elképzelhető, hogy az adat- vagy szöveges állományainkból több példánnyal rendelkezünk (hajlékonylemezek, merevlemezen), amelyeket különböző időpontban hoztunk létre, és a tartalmuk is különböző. Tehát csak az állományok neve azonos, a tartalmuk már nem. A zavar és a félreértések elkerülése végett érdemes ezt az állapotot minél előbb megszüntetni. A legegyszerűbb és leggyorsabb megoldást a REPLACE adja.

A másik eset — amely többé-kevésbé pont az ellentéte az előzőnek — akkor adódik, amikor a célként megjelölt lemezt olyan állományokkal akarjuk kiegészíteni, amelyek ott még nem léteznek.

Alapértelmezés szerint, vagyis ha a REPLACE parancs után csak a forrás- és a célhelyet adjuk meg, akkor a REPLACE csak azokat az állományokat másolja a forráslemezről a céllemezre, amelyek mind a két lemezen léteznek. Másként fogalmazva: ha egy adott állomány csak a forráslemezen szerepel, akkor azt a REPLACE alapértelmezés szerint nem másolja át.

Lássunk erre mindjárt egy példát! Tegyük fel, hogy az A: meghajtóban elhelyezett forráslemez tartalma a következő:

```
C:\>DIR A:
Volume in drive A has no label
Volume Serial Numbet is 0D6C-17DA
Directory of A:\
PELDA_1   TXT      2496      6-11-92      1:13p
PELDA_2   TXT     28208      6-16-92     11:53a
PELDA_3   TXT     11009      3-12-93     11:26a
PELDA_4   TXT       225      3-12-93     11:19a
          4 file(s) 41738 bytes
          318464 bytes free
```

Továbbá tételezzük fel, hogy a SAJAT elnevezésű alkatalógus, amelyben az állományokat ki akarjuk cserélni, tartalma a következő:

```
C:\>DIR \SAJAT <ENTER>
Volume in drive C has no label
Volume Serial Number is 3210-12F4
Directory of C:\SAJAT
.           <DIR>          3-12-93    11:15a
..          <DIR>          3-12-93    11:15a
PELDA_1    TXT      22     3-12-93    11:18a
PELDA_2    TXT      24     3-12-93    11:17a
PELDA_3    TXT      25     3-12-93    11:19a
           5 file(s)  71 bytes
           5038080 bytes free
```

Most pedig futtassuk a REPLACE utasítást:

```
C:\>REPLACE A: *.* \SAJAT <ENTER>
```

Erre a REPLACE a következőket írja a képernyőre:

```
Replacing C:\SAJAT\PELDA_1.TXT
Replacing C:\SAJAT\PELDA_2.TXT
Replacing C:\SAJAT\PELDA_3.TXT
3 file(s) replaced
```

```
A C:\SAJAT\PELDA_1.TXT kicserélése
A C:\SAJAT\PELDA_2.TXT kicserélése
A C:\SAJAT\PELDA_3.TXT kicserélése
3 állomány kicserélve
```

Kérdezzük le ezután ismét a C: meghajtó SAJAT alkatalógusának tartalmát!



```
C:\>REPLACE A: *.* \SAJAT
```

*A REPLACE parancs használata*

```
C:\>DIR \SAJAT <ENTER>
Volume in drive C has no label
Volume Serial Number is 3210-12F4
Directory of C:\SAJAT
.          <DIR>          3-12-93    11:15a
..         <DIR>          3-12-93    11:15a
PELDA_1   TXT          2496      6-11-92    1:13p
PELDA_2   TXT         28208     6-16-92    11:53a
PELDA_3   TXT         11009     3-12-93    11:26a
          5 file(s)    41713 bytes
          4999168 bytes free
```

Láthatjuk, hogy a katalógusban továbbra is három állomány szerepel, azonban már azzal a mérettel és dátum/időbélyeggel, amellyel az A : meghajtóban.

Most pedig nézzük, hogyan lehet elvégezni a második feladatot, vagyis amikor a céllemez az ott még nem szereplő állományokkal akarjuk kiegészíteni. Ekkor a REPLACE parancsot az /A kapcsolóval kell használni. Az előbbi példához visszatérve ha tehát a SAJAT elnevezésű alkönyvtárat a PELDA\_4.TXT állománnyal akarjuk kiegészíteni, akkor a következő parancsot kell futtatnunk:

```
C:\>REPLACE A: *.* \SAJAT /A <ENTER>
```

amelyre a következő üzenetet olvashatjuk a képernyőn:

```
Adding C:\SAJAT\PELDA_4.TXT
1 files(s) added
```

```
A C:\SAJAT\PELDA4.TXT hozzáadása
1 állomány hozzáadva
```

**Fontos megjegyzés:** a *REPLACE* paranccsal az állományok aktualizálása és célhely kiegészítése az ott még nem szereplő állományokkal nem végezhető el egyszerre! A *REPLACE* parancs alapértelmezés szerint csak azokkal az állományokkal dolgozik, amelyek a forrás- és a célhelyen egyaránt szerepelnek. A parancs a célhelyen szereplő állományokat felülírja a forráshelyen azonos névvel szereplő állományokkal. Az /A kapcsoló esetén viszont azokat az állományokat másolja a forráshelyről a célhelyre, amelyek ott még nem szerepelnek.

A *REPLACE* paranccsal együtt használható kapcsolók közül még kiemeljük a /S kapcsolót, amelynek hatására a parancs a céllemezén az éppen aktuális könyvtár alatti könyvtárakban is elvégzi az állományok felülírását. Az előbbi példához visszatérve a

```
C:\>REPLACE A: *.* C: /S <ENTER>
```

parancs hatására a *REPLACE* a merevlemez összes könyvtárát végignézi, és ha ezek bármelyikében a

```
PELDA_1.TXT
PELDA_2.TXT
PELDA_3.TXT
PELDA_4.TXT
```

állományokkal találkozik, akkor azokat felülírja.

Fontos megjegyezni, hogy a /S kapcsoló csak a céllemezre vonatkozik. A következő táblázatban a *REPLACE* kapcsolóit foglaljuk össze:

Kapcsoló	Művelet
/A	Arra utasítja a <i>REPLACE</i> parancsot, hogy csak azokat az állományokat másolja a céllemezre, amelyek ott még nem szerepelnek.

<b>Kapcsoló</b>	<b>Művelet</b>
/P	A kapcsoló hatására a REPLACE a cseréket vagy kiegészítéseket csak akkor végzi el, ha a felhasználó jóváhagyja.
/R	Arra utasítja a REPLACE parancsot, hogy csak azokat az állományokat cserélje ki, amelyek csak olvasható attribútummal rendelkeznek.
/S	A kapcsoló hatására a REPLACE a céllemezen az aktuális könyvtár alatti könyvtárakban is elvégzi a csereműveletet.
/U	Hatására a REPLACE a céllemezen csak azokat az állományokat cseréli ki, amelyek régebbiek, mint a cseréléshez felhasznált forrásállományok.
/W	A kapcsoló előírásának esetén a REPLACE mindaddig nem végzi el a cserét, amíg a felhasználó le nem nyom egy billentyűt.

*A leggyakoribb hibák a REPLACE parancs használata során:*

1. A rendszer a REPLACE parancsot, mivel az külső parancs, nem találja.

*Védekezés:*

A REPLACE parancs futtatása előtt váltsunk át abba az alkönyvtárba, amely a parancsot tartalmazza, vagy írjuk elő a parancsot tartalmazó alkönyvtár elérési útját a PATH utasításban!

2. A REPLACE a kicserélendő állományokat nem találja.

*Védekezés:*

1. Ellenőrizzük, hogy helyesen írtuk-e az állományok nevét, valamint a hozzájuk vezető útvonalat! (Ha a REPLACE parancsot nem abból az alkönyvtárból hajtjuk végre, amely az összehasonlítandó állományokat tartalmazza, a teljes — az útvonalat is tartalmazó — állományazonosító előírására van szükség.)

## **A REPLACE parancs**

A REPLACE (csere) parancs a lemezeinken már meglévő állományok aktualizálására használható. A REPLACE parancs alapértelmezés szerint a forráslemezen levő állományokkal a velük azonos néven szereplő állományokat a céllemezen felülírja. A /A kapcsoló hatására felülírást nem végez, csak a céllemez kiegészítését, a /U kapcsoló alkalmazása esetén pedig csak azokat az állományokat írja felül, amelyek a forráslemez állományainál régebbiek.

## *Ismétlés = Kulcs a sikerhez*

A DISKCOPY parancs segítségével egy- vagy kétmeghajtós rendszerben a lemezmásolás műveletét tudjuk elvégezni. Kétmeghajtós rendszerben a parancs csak akkor működik, ha a lemezek mérete azonos, ellenkező esetben a parancsot úgy kell használni, mintha egymeghajtós környezetben dolgoznánk.

A DISKCOMP két lemez tartalmát hasonlítja össze szektorról szektorra haladva. A parancsot rendszerint a DISKCOPY-val végrehajtott lemezmásolást követően használjuk.

Az XCOPY, kapcsolói különböző beállításával szelektív állománymásolást tesz lehetővé.

A COMP két állomány összehasonlítását teszi lehetővé.

A REPLACE segítségével a régi és már nem aktuális állományok az új verzióval írhatók felül, valamint az alkönyvtárak az ott nem szereplő állományokkal egészíthetők ki.

A parancsok mindegyike külső DOS parancs, így a használatukhoz a külső parancsokra vonatkozó szabályokat kell betartani.

## *Mit jelent?*

**Attribútum.** Valamely adattételt, állományt, képernyőkaraktert, változót leíró jellemző. Például az állomány rendelkezhet olyan attribútummal, amely azt rejtettnek vagy csak olvashatónak definiálja. A képernyőn megjelenített karakterekhez tartozhat olyan attribútum, amely a karakterek színét vagy fényességét definiálja.

**Eltolási érték (*offset*).** Megadott kiindulási pozíciótól, ún. báziscímtől mért távolság a memóriában. Az eltolási értékeket rendszerint a bájtok számában adják meg.

**Tizenhatos vagy hexadecimális számrendszer.** Az 16-os szám hatványain alapuló számrendszer, amely számok kifejezéséhez a 0—9 számjegyeket és az A, B, C, D, E, F karaktereket használja. (Az A értéke 10, a B értéke 11 ... az F értéke 15.)

# Az állományok logikai védelme és a véletlenszerűen törölt állományok visszaállítása

**Már tanultuk:** csak olvasható állomány, rejtett állomány, lemezformázás

A **csak olvasható állomány** (*read only*) tartalmát a DOS el tudja olvasni, annak módosítását vagy törlését azonban nem engedi.

**Rejtett állománynak** (*hidden*) azt az állományt nevezik, amely bár rajta van a lemezen, a könyvtárlistában azonban nem jelenik meg. Az állományok elrejtésének az a célja, hogy a felhasználók még véletlenül se tudják ezeket az állományokat törölni vagy felülírni. (A DOS rejtett állományai: IO.SYS, MSDOS.SYS)

**Lemezformázáskor** a lemezt a számítógépes alkalmazásra készítjük fel. A művelet a FORMAT parancs futtatásával végezhető el. A FORMAT a lemezt sávoknak és szektoroknak nevezett területekre osztja.

## Az állományok attribútumainak megtekintése és beállítása az ATTRIB paranccsal

3.0

KÜLSŐ

Az állományok létrehozásakor vagy módosításakor a DOS az állományhoz idő- és dátumbélyeget rendel. Ezeket az adatokat a DIR utasítással, vagyis a könyvtár tartalmának lekérdezésekor, meg tudjuk jeleníteni a képernyőn. Van azonban egy olyan jellemző is, az ún. attribútum, amit a DOS az állományhoz

rendel, de a könyvtár tartalmának listázásakor nem írja a képernyőre. A rendszer az állományhoz rendelt attribútum alapján dönti el, hogy az törölhető-e, módosítható-e, menthető-e lemezre a BACKUP utasítással, vagy a nevét meg kell-e jelenítenie az állományok nevének listázásakor. Az állományokhoz a következő attribútumok tartoznak:

Csak olvasható (*read only*), amely megakadályozza a DOS-t abban, hogy az állomány tartalmát módosítsa vagy törölje. Az ilyen attribútummal rendelkező állománnyal csak olyan utasítások hajthatók végre, amelyek annak tartalmát nem befolyásolják (pl. TYPE, PRINT).

Archív (*archive*), amellyel a DOS azt jelzi, hogy az adott állományról még nem készült biztonsági másolat, vagyis az állomány archiválandó. A DOS az újonnan létrehozott állományokat automatikusan ezzel az attribútummal látja el. Biztonsági másolat a BACKUP utasítással készíthető, amelyet a 4. Fejezetben ismertetünk, azonban az előző fejezetben tárgyalt XCOPY parancsnál is kihasználható az állományok archív attribútuma.

Rejtett (*hidden*), amely arra utasítja a DOS-t, hogy a katalógus listázásakor a rejtett attribútummal rendelkező állományok nevét ne írja a képernyőre, vagyis ily módon rejtse el a felhasználó szeme elől.

Rendszer (*system*), ami azt jelzi, hogy az adott állomány speciális DOS rendszerállomány. Ezt az attribútumot fenn kell tartani a DOS-nak.

Az állományok attribútumai tehát a DIR parancs használatával nem tudhatók meg. Erre külön parancsot hoztak létre, az ATTRIB parancsot, amellyel nem csak lekérdezhetők az állományok attribútumai, hanem a kívánt attribútum be is állítható.

Tegyük fel, hogy olyan rendszerben dolgozunk, amelyben a C: meghajtó gyökérkönyvtára tartalmazza a DOS rejtett állományait. Ha erre a könyvtárra kiadjuk az ATTRIB parancsot,

```
C:\>ATTRIB *.* <ENTER>
```

a következőhöz hasonló lista jelenik meg a képernyőnkön:

A	SHR	C:\IO.SYS
A	SHR	C:\MSDOS.SYS
A		C:\COMMAND.COM
A		C:\CONFIG.SYS
A		C:\AUTOEXEC.BAT

Az ATTRIB, mint látható, az állomány neve előtt ún. attribútumkarakterekkel jelöli az adott állomány attribútumait. Az állományok tehát egyszerre több

attribútummal, ún. attribútumkombinációval is rendelkezhetnek. Ha az állomány neve előtt semmilyen karakter nem jelenik meg, akkor az adott állományhoz semmilyen attribútum nincs beállítva. A következő táblázat az attribútumkaraktereket és a jelentésüket tartalmazza.

Karakter	Jelentés	Megjegyzés
A	Archív	
R	Csak olvasható	
S	Rendszerállomány	Az attribútum beállítása vagy törlése csak 5.0-s DOS esetén lehetséges.
H	Rejtett állomány	Az attribútum beállítása vagy törlése csak 5.0-s DOS esetén lehetséges.

A 2. Fejezetben a DISKCOPY parancs tárgyalásánál, valamint az EGYSZERŰEN DOS c. könyvben a DEL parancs tárgyalásakor már kitértünk a hajlékonylemezek mechanikai írásvédelmére, amellyel azok tartalma a véletlenszerű törléstől vagy felülírástól megóvható. Ilyen mechanikai írásvédelem a merevlemezek esetében nem létezik. Az ATTRIB parancs segítségével csak olvasható attribútummal láthatjuk el azokat a fontos állományokat, amelyek tartalmát nem akarjuk megváltoztatni, viszont a védelmüket biztosítani kívánjuk. Hozzuk létre a PROBA.TXT rövid szöveges állományt az ATTRIB parancs működésének kipróbálásához:

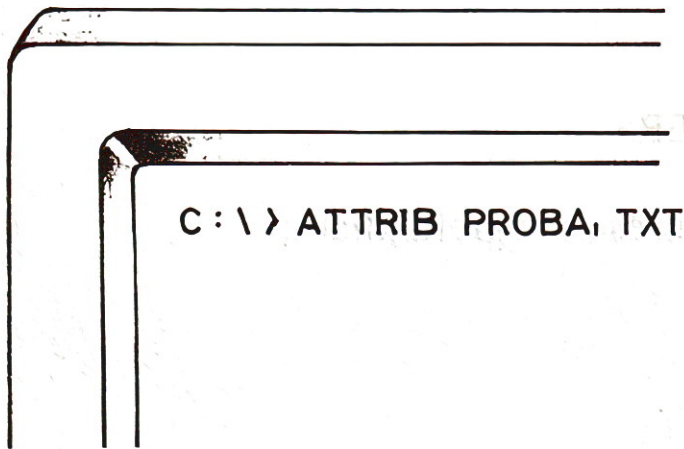
```
C:\>COPY CON PROBA.TXT <ENTER>
Az ATTRIB parancs kipróbálása. <ENTER>
^Z <ENTER>
1 File(s) copied
C:\>
```

Ezután az ATTRIB paranccsal kérdezzük le az állomány attribútumait!

```
C:\ATTRIB PROBA.TXT <ENTER>
A C:\PROBA.TXT
```

Mint tudjuk, a DOS az állományokhoz az A, vagyis az archív — pontosabban archiválható — attribútumot rendeli, ha azokat módosítjuk vagy létrehozunk, így jelezve, hogy biztonsági másolat az állományról még nem készült.

Az állományok attribútumának beállításakor az attribútumkarakterek előtt a +, törlésekor a – karaktert kell alkalmazni. Példaként állítsuk be az előbbi állomány attribútumát csak olvashatóra:



Az *ATTRIB* parancs használata

```
C:\>ATTRIB +R PROBA.TXT <ENTER>
```

majd kérdezzük le ismét az állomány attribútumait!

```
C:\>ATTRIB PROBA.TXT <ENTER>  
A R C:\PROBA.TXT
```

Látható tehát, hogy az archív attribútum megmaradt, és az állomány most már a csak olvasható attribútummal is rendelkezik. Most, hogy a *PROBA.TXT* állományt már védelemmel láttuk el, tegyünk kísérletet a törlésre:

```
C:\>DEL PROBA.TXT <ENTER>
```

Erre a DOS-tól a következő üzenetet kapjuk:

```
C:\>DEL PROBA.TXT <ENTER>  
Access denied
```

```
C:\>DEL PROBA.TXT <ENTER>  
Hozzáférés visszautasítva
```

Ahhoz tehát, hogy a csak olvasható állományt törölni tudjunk, meg kell szüntetnünk a csak olvasható attribútumát.



Végezzük el ezt a műveletet:

```
C:\>ATTRIB -R PROBA.TXT <ENTER>
```

Az előbbi DEL parancs ezután már sikeresen végrehajtható:

```
C:\>DEL PROBA.TXT <ENTER>
```

```
C:\>
```

**Megjegyzés:** A 2. Fejezetben az XCOPY parancs ismertetésekor arra is láthattunk példát, hogyan kell az állományok attribútumát archívra beállítani.

Az ATTRIB az állománynevek előírásakor helyettesítő karakterek alkalmazását is támogatja, valamint 3.3-as vagy e feletti DOS verzió esetén a /S kapcsoló előírását is lehetővé teszi. A /S kapcsoló hatására az ATTRIB nem csak a megadott, hanem az az alatti könyvtárakra is kiterjeszhető. A következő utasítás pl. a merevlemez összes állományának attribútumát archívra állítja be:

```
C:\>ATTRIB +A C:\*.* /S <ENTER>
```

### Megjegyzések:

1. Bár az ATTRIB paranccsal az 5.0-s DOS esetén az S attribútum is beállítható, ill. törölhető, ezen attribútum alkalmazását tartsuk fenn a DOS rendszer számára.
2. Sok felhasználó egyszerűen azért nem használja az ATTRIB parancsot, mert nem tudja, melyek azok az állománytípusok, amelyeket védelemmel kell és érdemes ellátnia. Az általános elv az, hogy a felhasználói programokat és egyéb fontosnak ítélt alkalmazói és segédprogramokat — a programállományok .EXE vagy .COM kiterjesztéssel rendelkeznek —, ill. a már nem módosított adatállományokat érdemes csak olvashatóra beállítani.

*Leggyakoribb hibák az ATTRIB parancs használata során:*

1. A rendszer az ATTRIB parancsot, mivel az külső parancs, nem találja.  
Védekezés:  
Az ATTRIB parancs futtatása előtt váltsunk át abba az alkönyvtárba, amely a parancsot tartalmazza, vagy írjuk elő a parancsot tartalmazó alkönyvtár elérési útját a PATH utasításban.
2. Az ATTRIB a parancssorban előírt állományt vagy állományokat nem találja.

*Védekezés:*

1. Ellenőrizzük, hogy helyesen írtuk-e az állományok nevét.
2. Ha az ATTRIB parancsot nem abból az alkönyvtárból hajtjuk végre, amely az előírt állományokat tartalmazza, a teljes — az útvonalat is tartalmazó — állományazonosító előírására van szükség.

## Az ATTRIB parancs

Az ATTRIB parancs segítségével a paraméterként megadott állomány vagy állományok attribútumai lekérdezhetőek, ill. beállíthatók. Az attribútumok a következők lehetnek: A: archív, R: csak olvasható, S: rendszer, H: rejtett. Az attribútum beállításához annak betűjele előtt a +, törléséhez a – karaktert kell szerepeltetni. Az ATTRIB parancs megfelelő alkalmazásával nemcsak a merevlemez megfelelő állományainak védelme, de egyéb DOS parancsok — pl. XCOPY, BACKUP, RESTORE végrehajtása — előtt alkalmazva az állományok gyorsabb és hatékonyabb kezelése oldható meg.

## *Véletlenszerűen törölt állományok visszaállítása*

Még a leggondosabb felhasználók életében is legalább egyszer előfordul, hogy olyan állományokra hajtják végre a DELETE parancsot, vagy olyan lemezt formáznak, amelyek értékes és fontos információkat, esetleg többhetes munka eredményét tartalmazták. Bár a korábbiakban már tárgyalt mechanikai írásvédelem, vagy a csak olvasható attribútum alkalmazása ezektől a balesetektől megvédenek, a mindennapos munkában résztvevő állományokat és lemezeket azonban a legritkább esetben szokták ilyen védelmekkel ellátni. A következő tanácsot ezért mindenképpen érdemes megfogadni:

A napi munka utolsó lépéseként mindig készítsünk másolatot azokról az állományokról, amelyekkel éppen dolgozunk.

De mit tehetünk akkor, ha már megtörtént a baj, és az állományokról még biztonsági másolatunk sincs? 5.0-s DOS használata esetén a MIRROR, UNDELETE, ill. UNFORMAT utasítások segítségével vissza tudjuk állítani az állományokat, ha azonban nem 5.0-s DOS verzióval dolgozunk, akkor sem kell megijednünk, létezik ui. több olyan segédprogram, amely ugyancsak képes az állományok visszaállítására.

## Az állományok védelmének előkészítése a MIRROR utasítással

5.0
-----

KÜLSŐ
-------

A MIRROR olyan speciális program, amely a futtatása után a memóriában marad, és a lemezre ráírja azokat a kulcsfontosságú információkat, amelyeket az UNFORMAT és az UNDELETE parancs a visszaállítás során felhasznál. Elsőként tehát azt vizsgáljuk meg, hogyan működik a MIRROR utasítás!

### *Kulcsinformációk tárolása*

A DOS annak érdekében, hogy a lemezszektorok tartalmát nyomon tudja követni, az ún. állományallokációs táblázatban (*FAT*) nyilvántartást vezet a lemez tartalmáról. Ezen táblázatban olyan adatok szerepelnek, amely alapján a DOS meg tudja állapítani, hogy a lemez melyik szektora tartalmaz már adatokat, mely szektorokba lehet még írni, és mely szektorokat jelölte a FORMAT — lemezkárosodás miatt — használhatatlannak. Ezt a táblázatot, amely ugyancsak a lemezen található, a DOS térképként használja, amikor az állományokat a lemezen elhelyezi.

Tekintsük át, milyen műveleteket végez el a FORMAT utasítás, amikor a lemezt formázza, ill. a DELETE utasítás, amikor az állományt törli!

A FORMAT a lemez formázásának első lépéseként új allokációs táblázatot készít. Ha ilyen már létezik a lemezen, akkor azt felülírja. Nem írja azonban felül a lemez adatokat tartalmazó területét. A FORMAT parancs végrehajtása után a DOS a lemezen tárolt adatokhoz az állományallokációs táblázat hiányában nem tud hozzáférni. Most pedig lássuk, hogyan működik a DELETE parancs!

A DOS az állományok lemezre írásakor az állományallokációs táblázatban egy vagy több bejegyzést helyez el, amelyek alapján az állományt a lemezen vissza tudja keresni. Ezenkívül minden egyes állományhoz ún. katalógusbejegyzést is rendel, amely az állomány nevét, méretét, dátum- és időbélyegét — ezeket az adatokat láthatjuk akkor, amikor a DIR-rel a könyvtár tartalmát lekérdezzük —, az attribútumot, valamint az állományallokációs táblázatban az adott állományhoz tartozó kezdő bejegyzést tartalmazza. Amikor valamelyik állományra a DEL utasítást végrehajtjuk, akkor a DOS az állományhoz tartozó adatokat nem törli a lemezről. Ehelyett az allokációs táblázatban az állományhoz tartozó bejegyzést úgy módosítja, hogy az adott állomány által elfoglalt

helyet felszabadítja más állományok számára, a katalógusbejegyzésben pedig az állomány nevének első betűjét a  $\sigma$  karakterrel írja felül. (Így tehát pl. a PELDA.TXT névből a DEL hatására  $\sigma$ ELDA.TXT lesz.) Ha ezután a katalógust a DIR utasítással listázzuk, a DIR a  $\sigma$  karakterből tudni fogja, hogy az állományt már töröltük, így annak nevét a listában már nem szerepelteti.

Most pedig lássuk, hogyan hasznosítja ezeket a jellemzőket a MIRROR utasítás!

A MIRROR — alapértelmezés szerint — az állományallokációs táblázatot, valamint a lemez gyökérkatalógusát a lemez adatterületére másolja, azaz ezen adatokról a lemezre egyfajta biztonsági másolatot készít. Ha a FORMAT utasítással az allokációs táblázatot felülírjuk, a másolat továbbra is megmarad a lemezen. Ezt használja fel az UNFORMAT utasítás akkor, amikor a lemezt eredeti állapotába állítja vissza.

A MIRROR-nak létezik egy olyan kapcsolója, amelynek hatására a parancs az állományok törlését nyomon követő szoftvert is telepíti. Ez a szoftver a törölt állományokra vonatkozó kulcsinformációkat — bejegyzés az állományallokációs táblázatban, valamint a katalógusban — tárolja. Az UNDELETE utasítás ezeket az információkat használja fel akkor, amikor az állományt a törlése után visszaállítja.

### Fontos megjegyzések:

- Valahányszor új állományt hozunk létre, vagy a már meglévő állományt módosítjuk, a DOS az allokációs táblázatot automatikusan aktualizálja. Ha tehát ez a táblázat módosul, akkor az értékei már nem egyeznek meg azzal a korábbi táblázattal, amelynek másolatát a MIRROR a lemezen elhelyezte. Ha tehát ezt követően hajtjuk végre az UNFORMAT utasítást, akkor az nem az aktuális információkkal dolgozik majd. Bár az UNFORMAT még ebben az esetben is valószínűleg a lemez adatainak legnagyobb részét vissza fogja állítani, bizonyos információk elveszhetnek. Tehát érdemes a MIRROR parancsot időnként újból futtatni. A MIRROR a lemezinformációt a MIRROR.FIL elnevezésű állományban tárolja.*
- Ha véletlenül törölünk egy vagy több állományt, akkor az állomány(ok) visszaállításáig ne írjunk semmilyen információt a lemezre! Amikor ui. az állományt töröljük, akkor a DOS az allokációs táblázatban az állományhoz tartozó bejegyzéseket újra használatra alkalmasnak jelöli meg. Ha tehát ezután a lemezre információt írunk, akkor a DOS már ezeket a lemezterületeket is felhasználhatja az új információk tárolásához, felülírva esetleg azokat az adatokat, amelyek a visszaállítandó állományhoz tartoznak. Az UNDELETE parancs a már felülírt információ visszaállítására nem képes!*

## A MIRROR használata

Ha a MIRROR utasítást a következők szerint futtatjuk, akkor az a lemez allokációs táblázatának éppen aktuális adataiból és a gyökérkatalógusból másolatot készít a lemezre.

```
C:\>MIRROR <ENTER>
```

Fontos tudni, hogy a fenti utasítás hatására az állományok törlését nyomon követő szoftver, amely az UNDELETE parancshoz szükséges, még nem töltődik be. Ezt csak úgy érhetjük el, ha a fenti parancsot a /T kapcsolóval futtatjuk. Például a következő parancs hatására a MIRROR a C: meghajtón 200 törölt állomány nyilvántartását hajtja végre:

```
C:\>MIRROR/TC-200 <ENTER>
```

A kapcsoló után szereplő karakterek jelentése a következő:

- C      Annak a meghajtónak a betűjele, ahol a rendszer a törölt állományok nyilvántartását el fogja végezni.
- 200   A kapcsoló hatására betöltött memóriarezidens program maximálisan 200 törölt állomány nyilvántartását tudja végrehajtani.

Az utasítás futtatása után a MIRROR a következő üzenetet írja a képernyőre:

```
Creates an image of the system area.  
Drive C being processed.  
The MIRROR process was successful.  
Deletion-tracking software being installed.  
The following drives are supported:  
Drive C - 200 files saved.
```

```
Installation complete.
```

```
A rendszerterület leképezése.  
A C meghajtó feldolgozása.  
A MIRROR eljárás sikeresen végrehajtva.  
Törlés-nyomonkövetés szoftver telepítve.  
A következő meghajtók támogatása:  
C meghajtó - 200 állomány tárolva.
```

```
A telepítés befejezve.
```

Az első három sor arról nyújt információt, hogy a MIRROR a másolatot az allokációs táblázatról és a gyökérkatalógusról sikeresen elkészítette. Az utolsó négy sorból pedig azt tudhatjuk meg, hogy a MIRROR az állományok törlését nyilvántartó szoftvert sikeresen telepítette.

A MIRROR utasítást érdemes az AUTOEXEC.BAT állományban elhelyezni, így a parancs automatikus futtatását biztosítva a rendszer indítását követően.

## Akaratlanul formázott lemez visszaállítása az UNFORMAT utasítással

**5.0****KÜLSŐ**

Az UNFORMAT parancsot a legjobban a használat során lehet megismerni. Ezért a következőkben az A: meghajtóba helyezett gyakorlólemez formázás utáni visszaállítását fogjuk elvégezni. Első lépésként vegyünk elő egy olyan lemezt, amelynek tartalmára már BIZONYOSAN nincs szükségünk — használhatunk új lemezt is! —, és végezzük el a formázását, vagyis futtassuk a következő parancsot:

```
C:\>FORMAT A: <ENTER>
```

Amint a rendszer a lemez formázását elvégezte, hozzuk létre rajta a következő alkatalógust:

```
C:\>MD A:\PELDA <ENTER>
```

és készítsük el a következő állományt:

```
C:\>COPY CON A:SAJAT.TXT <ENTER>
```

```
Most az UNFORMAT parancs használatát <ENTER>
```

```
próbáljuk ki. <ENTER>
```

```
^Z <ENTER>
```

```
1 File(s) copied
```

```
C:\>
```

Ezt követően futtassuk a MIRROR parancsot az A: meghajtóra:

```
C:\>MIRROR A: <ENTER>
```

A MIRROR a művelet elvégzését a következő üzenettel jelzi:

```
Crates an image of the system area.  
Drive A being processed.  
The MIRROR process was successful.
```

Ezután formázzuk ismét a lemezt! (EZ A VÉLETLENSZERŰ FORMÁZÁS!)

```
C:\>FORMAT A: <ENTER>
```

és amint a rendszer a művelettel végzett, kérjünk listát a lemez tartalmáról a DIR paranccsal:

```
C:\>DIR A: <ENTER>
```

Erre a következő üzenetet olvashatjuk a képernyőn:

```
Volume in drive A has no label  
Volume Serial Number is 0E57-17E1  
Directory of A:\  
File not found
```

```
Az A meghajtónak nincs kötetelnevezése  
A kötet sorozatszám 0E57-17E1  
Az A:\ könyvtár  
Állomány nem található
```

Lássuk, mit tud kezdeni ezzel a lemezzel az UNFORMAT parancs!

```
C:\>UNFORMAT A: <ENTER>
```

A parancs futtatása után a következő üzenet jelenik meg a képernyőn:

Insert disk to rebuild in drive A:  
and press ENTER when ready.

Helyezük be a visszaállítandó lemezt az A: meghajtóba,  
majd nyomjuk le az ENTER gombot!

Járjunk el a felszólításnak megfelelően. Erre az UNFORMAT a következő-  
höz hasonló üzenettel felel:

Restores the system area of your disk by using the image  
file created by the MIRROR command.

WARNING!!      WARNING!!

This command should be used only to recover from inadver-  
tent use of the FORMAT command or the RECOVER command. Any  
other use of the UNFORMAT command may cause you to lose  
data! Files modified since the MIRROR image file was  
created may be lost.

Searching disk for MIRROR image.

The last time the MIRROR or FORMAT command was used at  
11:31 on 03-01-93.

The prior time the MIRROR or FORMAT command was used at  
11:30 on 03-01-93.

If you wish to use the last file as indicated  
above, press L. If you wish to use the prior  
file as indicated above, press P. Press ESC  
to cancel UNFORMAT.

A lemez rendszerterületének visszaállítása a MIRROR pa-  
ranccsal létrehozott tükörképállomány felhasználásá-  
val.

FIGYELEM!!      FIGYELEM!!

Ezt a parancsot csak a FORMAT vagy a RECOVER parancs  
akaratlan használata esetén szabad használni! Ha az UN-  
FORMAT parancsot bármilyen más esetben alkalmazzuk, a



lemezen adatvesztést okozhatunk! Azok a módosítások, amelyeket a MIRROR paranccsal létrehozott tükörképállomány létrehozása után jönnek létre a lemezen, elveszhetnek.

A MIRROR tükörképállományának keresése.

A MIRROR vagy a FORMAT parancs legutolsó használatának időpontja 14:31 03-01-93.

A MIRROR vagy a FORMAT parancs utolsó előtti használatának időpontja 14:31 03-01-93.

Ha a fentiekben megjelölt utolsó állományt akarjuk használni, akkor az L gombot nyomjuk le. Ha a fentiekben megjelölt utolsó előtti állományt akarjuk használni, akkor a P gombot nyomjuk le. Az UNFORMAT művelet felfüggesztéséhez az ESC gombot kell lenyomni.

Jelen esetben az UNFORMAT két tükörállományt talált a lemezen. Az első az, amely a MIRROR parancs futtatása után jött létre, a másodikat viszont a FORMAT parancs állította elő. Ezt az esetet a későbbiekben tárgyaljuk. Az UNFORMAT tehát a szöveg szerint az L vagy a P karakter beírását várja, és ennek megfelelően a visszaállításhoz a legutolsó vagy az utolsó előtti tükörállományt fogja használni. Írjuk be tehát az L karaktert! Erre az UNFORMAT a következő üzenettel válaszol:

```
The MIRROR image file has been validated.  
Are you sure you want to update the system area of your  
drive A (Y/N)?
```

```
A MIRROR tükörképállomány érvényesítve.  
Bizonyosan aktualizálni akarja az A: meghajtó rendszer-  
területét (I/N)?
```

Az igen válaszhoz írjuk be az Y karaktert, amelyre a következő üzenetet kapjuk:

```
The system area of drive A: has been rebuilt.  
You may need to restart the system.
```

Az A: meghajtó rendszerterülete újraépítve.  
Tanácsos a rendszert újraindítani.

A művelet sikeres elvégzése könnyen ellenőrizhető, ha lekérdezzük a lemez katalógusát:

```
C:\>DIR A: <ENTER>
Volume in drive A has no label
Volume serial number is 0D6C-17DA
Directory of A:\
PELDA    <DIR>                03-01-93    2:30p
SAJAT    TXT                      22    03-01-93    2:35p
MIRROR   BAK                      6144   03-01-93    2:37p
MIRROR   FIL                      6144   03-01-93    2:38p
         4 file(s)  12341 bytes
         347136 bytes free
```

Érdeemes megfigyelni, hogy a korábbiakban létrehozott alkönyvtáron és a szöveges állományon, valamint a kulcsinformációkat tartalmazó MIRROR.FIL állományon kívül még a MIRROR.BAK állomány is szerepel a listában. Ennek az az oka, hogy amikor a MIRROR vagy a FORMAT parancsot futtatjuk, a már meglévő MIRROR állományt a rendszer .BAK kiterjesztéssel látja el, és ezután hozza létre az új MIRROR.FIL állományt.

Már a bevezető részben is utaltunk rá, hogy ha a lemez tartalmát a MIRROR parancs futtatását követően módosítjuk, akkor a MIRROR által az állományallokációs táblázatról készült másolat nem felel meg a valós helyzetnek. Próbáljuk ki ebben a helyzetben is az UNFORMAT viselkedését! Első lépésként egy új állomány létrehozásával módosítsuk a lemez tartalmát!

```
C:\>COPY CON A:PROBA 2.TXT <ENTER>
Ez az új állomány. <ENTER>
^Z <ENTER>
    1 File(s) copied
C:\>
```

Az UNFORMAT parancs /J kapcsolójával biztosítjuk, hogy MIRROR parancs állományait tároltuk, és az azonos lemezre másolt információval. Most ezt fogjuk kipróbálni!

```
C:\>UNFORMAT A: /J <ENTER>
```

Az utasítás végrehajtásának első lépéseként most is a visszaállítandó lemez A: meghajtóba helyezése és a MIRROR tükörképállomány kiválasztására kapunk felszólítást. Az L vagy a P karakter beírása után — mivel az állományok különböznek —, a parancs a következőket írja a képernyőre:

```
The MIRROR image file has been validated.
The system area does not agree with the
MIRROR image file.
```

```
A MIRROR tükörképállomány érvényesítve.
A rendszerterület nem egyezik a
MIRROR tükörképállománnyal.
```

Aktualizáljuk tehát a MIRROR.FIL állományt a MIRROR parancs futtatásával, majd ismételjük meg az előbbi parancsot!

```
C:\>MIRROR A: <ENTER>
C:\>UNFORMAT A: /J <ENTER>
```

Mivel most már a MIRROR.FIL állomány tartalma az éppen aktuális értékeknek felel meg, a következő üzenetet kapjuk:

```
The system area of drive A: has been verified
to agree with the MIRROR image file.
```

```
Az A: meghajtó rendszerterületének tartalma
igazoltan egyezik a MIRROR tükörképállomány tartalmá-
val.
```

### ***Biztonságos formázás***

Az előző példában az UNFORMAT által küldött képernyőképből már kiderült, hogy az 5.0-s DOS FORMAT parancsa is létrehoz egy MIRROR állományt, mielőtt a lemez formázását elvégezné. Hacsak kifejezetten másként

nem rendelkezünk, a DOS 5.0-s verziójának FORMAT utasítása ún. biztonságos formázást végez, vagyis a lemez adattartalmát nem semmisíti meg valamint a kulcsinformációkról — a MIRROR paranccsal azonos módon — a lemezre másolatot készít. Ha tehát a FORMAT parancsot olyan lemezre adjuk ki, amely már tartalmaz információt, akkor a FORMAT — 5.0-s DOS esetén! — a következő üzenettel jelzi, hogy az UNFORMAT parancs használatához szükséges információkat lemezre írja:

```
Saving UNFORMAT information
```

```
Az UNFORMAT-hoz szükséges információ elmentése
```

Ha a formázandó lemez már annyira telített, hogy nem képes további információk tárolására, akkor a FORMAT, mielőtt a formázási művelethez hozzáfekedne, a következő információt jeleníti meg:

```
Drive A: Error Insufficient space for the MIRROR image
file.
There was an error creating the format recovery file.
This disk cannot be unformatted.
Proceed with the Format (Y/N)?
```

```
Az A: meghajtóban levő lemezen a hely a MIRROR állomány
számára nem elegendő.
Hiba a formázás visszaállításhoz szükséges állomány lét-
rehozásában.
Ezt a lemezt nem lehet a formázás után visszaállítani.
Folytassam a formázás végrehajtását? (I/N)
```

Vagyis a parancs jelzi, hogy nem tudja létrehozni a MIRROR állományt és a felhasználótól visszaigazolást vár a formázás folytatására.

**Fontos megjegyzés:** Előfordulhat, hogy valamely lemez tartalmát véglegesen és biztonságosan meg akarjuk semmisíteni, tehát úgy, hogy azt már ne lehessen visszaállítani. Az 5.0-s DOS esetén a FORMAT paranccsal együtt használt /U kapcsolóval ez is elérhető. A következő táblázatban az UNFORMAT paranccsal együtt használható kapcsolókat foglaljuk össze.

**Kapcsoló Művelet**

- /J** Arra utasítja az UNFORMAT utasítást, hogy az éppen aktuális gyökérkatalógus és allokációs táblázat tartalmát a MIRROR paranccsal létrehozott MIRROR.FIL állomány tartalmával összehasonlítsa. A kapcsoló alkalmazása esetén a parancs csak összehasonlítást végez. A lemezt nem állítja vissza.
- /L** A kapcsoló előírásának esetén az UNFORMAT minden egyes állomány és könyvtár nevét a képernyőre írja, amelyet a lemezen megtalál.
- /P** A kapcsoló alkalmazásának esetén az UNFORMAT az üzeneteit a képernyőn kívül a nyomtatóra is küldi.
- /TEST** A kapcsoló alkalmazása esetén az UNFORMAT kijelzi a képernyőre, hogyan állítja majd vissza a lemezt anélkül, hogy azt ténylegesen visszaállítaná.
- /U** Arra utasítja az UNFORMAT parancsot, hogy a lemezt a kulcsinformációkat tartalmazó MIRROR állomány nélkül állítja vissza.

*A leggyakoribb hibák az UNFORMAT parancs használata során:*

1. A rendszer az UNFORMAT parancsot, mivel az külső parancs, nem találja.  
*Védekezés:*  
Az UNFORMAT parancs futtatása előtt váltsunk át abba az alkönyvtárba, amely a parancsot tartalmazza, vagy írjuk elő a parancsot tartalmazó alkönyvtár elérési útját a PATH utasításban!
2. A UNFORMAT parancsot a /U kapcsolóval hajtottuk végre, ami a teljes lemeztartalom felülírását jelenti.  
*Védekezés:*  
A /U kapcsolót csak igazán indokolt esetben használjuk!

## A MIRROR parancs

A MIRROR (tükör) parancs a lemez állományallokációs táblázatát és a gyökérkatalógus tartalmát a MIRROR.FIL elnevezésű állományba a lemezre írja. Az itt tárolt ún. kulcsinformációk alapján az UNFORMAT vissza tudja állítani a lemez tartalmát az akaratlanul végrehajtott FORMAT parancs után. A rendszer a MIRROR.FIL állomány aktualizálását nem végzi automatikusan. Erről a felhasználónak kell gondoskodnia a MIRROR parancs ismételt futtatásával.

A MIRROR parancsot a /T kapcsolóval (pontosabban: /Tmeghajtó[-állományok száma]) futtatva olyan memóriarezidens program betöltése érhető el, amely az előírt meghajtó lemezéről törölt állományokról nyilvántartást vezet. Ezt az információt az UNDELETE parancs használja fel akkor, amikor a véletlenszerűen törölt állományokat visszaállítja.

## Az UNFORMAT parancs

Az UNFORMAT parancs segítségével a véletlenszerűen formázott lemez tartalma nyerhető vissza. Az UNFORMAT a visszaállításhoz a MIRROR parancs által létrehozott és kulcsinformációkat tartalmazó MIRROR.FIL állományt használja. Az /U kapcsoló előírásának esetén az UNFORMAT a MIRROR.FIL állomány használata nélkül állítja vissza a lemezt.

## Törölt állományok visszaállítása az UNDELETE parancssal

5.0

KÜLSŐ

Miként az előzőekben már említettük, a MIRROR parancsot a /T kapcsolóval futtatva olyan memóriarezidens szoftver betöltése érhető el, amely a törölt állományokról nyilvántartást vezet. Ezt a nyilvántartást használja fel az

UNDELETE parancs akkor, amikor a törölt állományokat visszaállítja. A parancs használatát most is konkrét eseten keresztül fogjuk bemutatni.

Első lépésként — ha eddig még nem tettük meg — futtassuk a következő MIRROR parancsot:

```
C:\>MIRROR /TC-200 <ENTER>
```

A kapcsoló után szereplő karakterek jelentése:

C Annak a meghajtónak a betűjele, ahol a rendszer a törölt állományok nyilvántartását el fogja végezni.

-200 A kapcsoló hatására betöltött memóriarezidens program 200 törölt állomány nyilvántartását fogja elvégezni. (Lehetséges értékek: 1—999)

Következő lépésként hozzuk létre a SAJAT elnevezésű alkönyvtárat, amelybe váltsunk is át:

```
C:\>MD \SAJAT <ENTER>
```

```
C:\>CD \SAJAT <ENTER>
```

A SAJAT elnevezésű alkönyvtárban hozzuk létre a következő szöveges állományokat:

```
C:\SAJAT>COPY CON PROBA_1.TXT <ENTER>
```

```
Ez az első állomány. <ENTER>
```

```
^Z <ENTER>
```

```
1 File(s) copied
```

```
C:\SAJAT>COPY CON PROBA_2.TXT <ENTER>
```

```
Ez a második állomány. <ENTER>
```

```
^Z <ENTER>
```

```
1 File(s) copied
```

```
C:\SAJAT>COPY CON PROBA_3.TXT <ENTER>
```

```
Ez a harmadik állomány. <ENTER>
```

```
^Z <ENTER>
```

```
1 File(s) copied
```

```
C:\SAJAT>
```

A DIR parancs felhasználásával győződjünk meg arról, hogy az állományok ténylegesen létrejöttek:

```
C:\SAJAT>DIR <ENTER>
Volume in drive C has no label
Volume Serial Number is 3210-12F4
Directory of C:\SAJAT
.                <DIR>          03-03-93  7:35p
..               <DIR>          03-03-93  7:35p
PROBA_1  TXT                22  03-03-93  7:40p
PROBA_2  TXT                24  03-03-93  7:45p
PROBA_3  TXT                25  03-03-93  7:55p
          5 file(s)                71 bytes
          424256 bytes free
```

Ezután töröljük a lemezről a PROBA\_1.TXT állományt,

```
C:\SAJAT>DEL PROBA_1.TXT <ENTER>
```

majd az alkönyvtár tartalmának listázásával győződünk meg az állomány tényleges törléséről:

```
C:\SAJAT>DIR <ENTER>
Volume in drive C has no label
Volume Serial Number is 3210-12F4
Directory of C:\SAJAT
.                <DIR>          03-03-93  7:35p
..               <DIR>          03-03-93  7:35p
PROBA_2  TXT                24  03-03-93  7:45p
PROBA_3  TXT                25  03-03-93  7:55p
          4 file(s)                49 bytes
          3389440 bytes free
```

Állítsuk vissza az állományt az UNDELETE felhasználásával:

```
C:\SAJAT>UNDELETE PROBA_1.TXT <ENTER>
```

Erre az UNDELETE parancs a következő üzenetet írja a képernyőre:



```

Directory: C:\SAJAT
File Specifications: PROBA 1. TXT
  Deletion-tracking file contains 1 deleted files.
  Of those, 1 files have all clusters available,
              0 files have some clusters available,
              0 files have no clusters available.

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

Using the deletion-tracking file ...
Searching deletion-tracking file ...
PROBA 1 TXT 22 03-03-93 7:40p...A Deleted:
03-03-93 7:40p
All of the clusters for this file are available.
Undelete (Y/N)?

```

```

Könyvtár: C:\SAJAT
Állomány specifikáció: PROBA 1. TXT
  A törlés-nyomkövetés állomány 1 törölt állományt tar-
  talmaz, amelyek közül
              1 állománynál valamennyi klaszter elérhető,
              0 állomány néhány klasztere elérhető
              0 állomány egyetlen klasztere sem érhető el
MS-DOS könyvtár 1 törölt állományt tartalmaz,
amelyek közül 1 állomány állítható vissza.

A törlés-nyomkövetés állomány használata
A törlés-nyomkövetés állomány keresése
PROBA 1 TXT 22 03-03-93 7:40p...A Törölve:
03-03-93 7:40p
Az állomány valamennyi klasztere elérhető.
Visszaállítsam (I/N)?

```

Az igen válaszhoz tehát írjuk be az Y karaktert, amelyre a rendszer a következőkkel felel:

```
File successfully undeleted
```

```
Az állomány sikeresen visszaállítva.
```

Ha a törléskor helyettesítő karaktert is alkalmaztunk, és így egyszerre több állományt töröltünk, akkor azokat az

```
C:\SAJAT>UNDELETE *.* <ENTER>
```

paranccsal egyszerre is visszaállíthatjuk. Ilyen esetben azonban az UNFORMAT minden egyes állomány visszaállításához visszaigazolást vár a felhasználótól:

```
All of the clusters for this file are available. Undelete (Y/N)?
```

```
Az állomány számára minden klasztere elérhető.  
Visszaállítsam (I/N)?
```

Ha az UNDELETE-hez az /ALL kapcsolót is megadjuk, akkor a parancs az állományokat megkérdezés nélkül állítja vissza. Az utasítás használata során a /LIST kapcsoló is hasznos szolgálatot tehet. Ekkor az UNDELETE a törölt állományok nevét a képernyőre írja, a visszaállítást azonban nem végzi el.

*A leggyakoribb hibák az UNDELETE parancs használata során:*

1. A rendszer az UNDELETE parancsot, mivel az külső parancs, nem találja.

*Védekezés:*

Az UNDELETE parancs futtatása előtt váltsunk át abba az alkönyvtárba, amely a parancsot tartalmazza, vagy írjuk elő a parancsot tartalmazó alkönyvtár elérési útját a PATH utasításban.

2. Az UNDELETE parancs használata előtt az érintett lemezre írási műveletet végzünk.

*Védekezés:*

Az UNDELETE parancsot a DELETE végrehajtása után azonnal futtatni kell, ellenkező esetben ui. nem garantálható az állományok visszaállítása.

## Az UNDELETE parancs

Az UNDELETE parancs segítségével a véletlenül törölt állományok állíthatók vissza. Az UNDELETE parancs a MIRROR /T, futtatásával, de anélkül is használható.

## Lemezek és állományok visszaállítása a MIRROR használata nélkül

Az előző alfejezetekben már utaltunk rá, hogy az UNDELETE és az UNFORMAT parancs a MIRROR használata nélkül is működtethető. Most erre mutatunk be egy példát, felhasználva az előző alfejezetben létrehozott PROBA 1.TXT, PROBA 2.TXT és PROBA 3.TXT állományokat. A CTRL—ALT—DEL lenyomásával indítsuk újra a rendszerünket, de most ne futtassuk a MIRROR parancsot! Váltunk át viszont a SAJAT elnevezésű alkönyvtárba, és töröljük az abban található három állományt.

```
C:\>CD \SAJAT <ENTER>  
C:\SAJAT>DEL *.* <ENTER>
```

Ezután pedig futtassuk az UNDELETE parancsot:

```
C:\SAJAT>UNDELETE *.* <ENTER>
```

Ekkor a képernyőn a következő felirat jelenik meg:

```
File Specifications: *.*  
Deletion-tracking file not found.  
  
MS-DOS directory contains      3 deleted files.  
Of those,      3 files may be recovered.  
  
Using the MS-DOS directory.  
PROBA_1  TXT  22 03-03-93      7:40p...A UNDELETE (Y/N)?
```

A korábbiakban már említettük, hogy a DELETE parancs hatására a DOS az állomány nevének első karakterét a  $\sigma$  karakterrel írja felül. Ezt a karaktert az UNDELETE most kérdőjellel jelzi. Ha tehát a kérdésre az Y beírásával válaszolunk, akkor az UNDELETE parancs a

Please type the first character for ?ROBA\_1.TXT:

Kérem a ?ROBA\_1.TXT esetén az első karaktert beírni:

üzenetet írja a képernyőre. Írjuk be a P karaktert, amelyre a parancs az állomány visszaállítását elvégzi, és ezt a képernyőn is jelzi.

File successfully undeleted.

Az állomány visszaállítva.

Ezután az UNDELETE a fentivel azonos módszerrel hozzálát a két további állomány visszaállításához. A művelet elvégzését már nem részletezzük. Gyakorlásként az Olvasóra bízunk.

Most pedig lássuk, mit tehetünk akkor, ha a lemez visszaállítását akarjuk elvégezni az UNFORMAT utasítással. Ha a művelethez nem áll rendelkezésre a MIRROR.FIL állomány, akkor csak egyet tehetünk. Futtassuk a parancsot a /U kapcsolóval. Ebben az esetben — csak 5.0-s DOS esetén!!! — az UNFORMAT azt az információt fogja felhasználni, amelyet a FORMAT a formázási művelet elvégzése előtt a lemezre írt.

## ***Ismétlés = Kulcs a sikerhez***

Az ATTRIB parancs segítségével a paraméterként megadott állomány vagy állományok attribútumai lekérdezhetők, ill. beállíthatók. Az attribútumok a következők lehetnek: A — archív, R — csak olvasható, S — rendszer, H — rejtett. Az attribútum beállításához annak betűjele előtt a +, törléséhez a — karaktert kell szerepeltetni. Az ATTRIB parancs megfelelő alkalmazásával

nemcsak a merevlemez megfelelő állományainak logikai védelme, de egyéb DOS parancsok — pl. XCOPY, BACKUP, RESTORE — előtt alkalmazva az állományok gyorsabb és hatékonyabb kezelése is megoldható.

A MIRROR parancs a lemez állományallokációs táblázatát és a gyökérkatalógus tartalmát a MIRROR.FIL elnevezésű állományba menti el. Az itt tárolt ún. kulcsinformációk alapján az UNFORMAT vissza tudja állítani a lemez tartalmát az akaratlanul végrehajtott FORMAT parancs után. A rendszer a MIRROR.FIL állomány aktualizálásáról automatikusan nem gondoskodik. Ezt a műveletet tehát a felhasználónak kell elvégeznie a MIRROR parancs ismételt futtatásával.

A MIRROR parancsot a /T kapcsolóval (pontosabban: /Tmeghajtó[-állományok száma]) futtatva olyan memóriarezidens program betöltése érhető el, amely az előírt meghajtó lemezéről törölt állományokról nyilvántartást vezet. Ezt az információt az UNDELETE parancs használja fel akkor, amikor a véletlenszerűen törölt állományokat visszaállítja.

Az UNFORMAT parancs segítségével a véletlenszerűen formázott lemez tartalma nyerhető vissza. Az UNFORMAT a visszaállításhoz a MIRROR parancs által létrehozott és kulcsinformációkat tartalmazó MIRROR.FIL állományt használja fel. Az /U kapcsoló előírásának esetén viszont az UNFORMAT a MIRROR.FIL állomány használata nélkül állítja vissza a lemezt.

Az UNDELETE parancs, amely a MIRROR /T futtatásával, de az nélkül is használható, a véletlenül törölt állományok visszaállítását szolgálja. Ha a parancsot a MIRROR /T előzetes végrehajtása nélkül alkalmazzuk, akkor az állományok visszaállítását csak egyesével — a parancs kérdéseire adott megfelelő válaszokkal — tudjuk elérni.

## *Mit jelent?*

**Állományallokációs táblázat (FAT).** A lemezen tárolt állományok helyének nyilvántartása klaszterenként.

**Klaszter.** A lemezszektoroknak olyan csoportja, amelyet az operációs rendszer egyetlen egységként kezel.

# ***Nagyméretű állományok hajlékonylemezre másolása (avagy BACKUP/RESTORE könnyek nélkül)***

## ***Bevezető gondolatok a BACKUP/RESTORE használatával kapcsolatban***

A BACKUP/RESTORE összetartozó parancspár, amellyel lemezekről, alkönyvtárakról vagy akár egyetlen állományról is biztonsági másolatot készíthetünk. A BACKUP az állományok lemezre írását, vagyis a biztonsági másolatok elkészítését végzi, míg a RESTORE a BACKUP paranccsal létesített biztonsági másolatokat visszatölti a mentés eredeti helyére. Bár a BACKUP parancs hajlékonylemezről is elkészíti a biztonsági másolatot, igazi felhasználási területe azonban a merevlemez tartalmának mentése hajlékonylemezre.

Nagyon sok felhasználó idegenkedik a BACKUP/RESTORE parancsok használatától, a következő okok miatt:

- Felesleges időpazarlásnak tartja a biztonsági másolatok készítését.
- Nehézkesnek ítéli a BACKUP/RESTORE használatát.
- Sikertelenül próbálkozott a BACKUP/RESTORE használatával.

Nézzük meg ezeket az okokat kicsit részletesebben!

1. Miért van egyáltalán szükség az állományok és a lemezek biztonsági másolatainak elkészítésére? A kérdésre ezúttal egy újabb kérdéssel felelünk: Milyen munkáinkat veszítenénk el akkor, ha mondjuk a következő nap reggelén nem működne a számítógépünkbe épített merevlemez? Akik már átéltek ilyet, azoknak nem kell különösebben magyarázni ezt az érzést, akik pedig még nem, azok gondolják végig, és ennek szellemében olvassák végig ezt a fejezetet!

A 2. Fejezetben már tárgyaltuk a DISKCOPY parancsot, amellyel a hajlékonylemezekről készíthető biztonsági másolat. Milyen utasítást használhatunk azonban az akár 20 Mb-ot is tartalmazó merevlemez biztonsági másolatának elkészítéséhez, vagy egy olyan állomány lemezre írásához, amelynek mérete meghaladja a hajlékonylemez méretét?

A DOS ezen feladat megoldásához a BACKUP és a RESTORE parancsokat szolgáltatja, és ha csak a DOS áll a rendelkezésünkre, akkor ezeket a parancsokat vagyunk kénytelenek használni. (Manapság a felhasználók az állományok lemezre írásához és tárolásához többféle tömörítő és egyéb kiszolgáló programot is használnak.)

2. A BACKUP és a RESTORE parancshoz több kapcsoló is tartozik, és a parancsok használatában rendszerint az adja a nehézséget, hogy ezeket a kapcsolókat a felhasználók nem, vagy nem a megfelelő esetben használják. A fejezetben a parancsok kapcsolóit részletesen és használat közben is tárgyaljuk.

3. Mi lehet az oka annak, hogy a BACKUP-pal mentett állományokat nem tudjuk visszaállítani, vagyis megghiúsul az állományok visszatöltése?

1. Amikor a BACKUP az állományokat hajlékonylemezre írja, akkor a legelső lemez esetében az állományok adatrészét a BACKUP.001 állományban, az állományokhoz vezető útvonalat pedig a CONTROL.001 állományban tárolja. (Tehát az első BACKUP lemez csak ezt a két állományt fogja tartalmazni.) Ezenkívül a lemez kötetelnevezését is BACKUP 001-re változtatja. Ha a biztonsági mentéshez nem elég egy lemez, akkor a második lemezen a BACKUP.002 és a CONTROL.002 állományokat hozza létre, a lemez kötetelnevezése pedig BACKUP 002 lesz, és így tovább. Ezek az állományok tehát a BACKUP program speciális adatállományainak tekintendők, amelyeket csak és kizárólag a RESTORE parancs tud olvasni, ill. amelyekből csak a RESTORE parancs tudja az állományokat a DOS többi parancsnak is használható állapotba visszaállítani.

2. Ha az állományok biztonsági másolatainak elkészítéséhez 3.3-as DOS verzióhoz tartozó BACKUP parancsot használtunk, akkor a visszatöltés-kor is 3.3-as DOS verziójú RESTORE parancsot kell használnunk, vagyis a *BACKUP-nak és a RESTORE-nak azonos DOS verziójúnak kell lennie.*

## *Előkészületek a BACKUP használatához*

A BACKUP parancs futtatása előtt a következő előkészületeket kell feltétlenül elvégezni:

1. Meg kell határoznunk a biztonsági másolathoz szükséges lemezek számát.
2. Előre formáznunk kell a felhasználandó lemezeket.

A biztonsági másolathoz szükséges lemezek számának meghatározásához, akár az egész merevlemezről, akár annak csak egy alkönyvtáráról van is szó, nagyon jól használható a CHKDSK (lemezellenőrzés) külső DOS parancs, amelyet mindegyik DOS verzió tartalmaz. A parancs a lemez allokációs táblázatának megvizsgálásával többek között a következő adatokat adja meg a lemezről:

- a lemezen levő szabad, felhasznált, valamint megrongálódott hely méretét bájtban;
- a rejtett állományok számát;
- a teljes és a szabad memória méretét.

Próbáljuk ki a parancsot saját rendszerünkben is! Futtassuk tehát a következő parancsot!

```
C:\>CHKDSK <ENTER>
```

Ekkor a következőhöz hasonló információk jelennek meg a képernyőn:

```
Volume Serial Number is 3210-12F4
42366976 bytes total disk space
 3004416 bytes in 5 hidden files
  176128 bytes in 83 directories
35559424 bytes in 1383 user files
 3627008 bytes available on disk

    2048 bytes in each allocation unit
   20687 total allocation units on disk
    1771 available allocation units on disk
 655360 total bytes memory
570784 bytes free
```



A kötet sorozatszám 3210-12F4

42366976	bájt a lemezen levő összes hely
3004416	bájt 5 rejtett állományban
176128	bájt 83 könyvtárban
35559424	bájt 1383 felhasználói állományban
3627008	bájt a rendelkezésre álló hely a lemezen
2048	bájt az egyes allokációs egységekben
20687	darab az összes allokációs egység a lemezen
1771	rendelkezésre álló allokációs egység a lemezen
655360	bájt a teljes memória mérete
570784	bájt szabad

Az adatok lekérdezése után a következő lépés a foglalt bájtok számának meghatározása:

Felhasznált bájtok száma = a lemezen levő összes hely — rendelkezésre álló bájtok

Számítsuk ki ezt az értéket az első CHKDSK parancs futtatása után megjelenített adatokkal:

Felhasznált bájtok száma = 42366976 — 3627008

Felhasznált bájtok száma = 38739968

Most, hogy ezt az adatot már ismerjük, kiszámíthatjuk a biztonsági mentéshez szükséges hajlékonylemezek számát, ha tudjuk, hogy az egyes lemeztípusok milyen tárolási kapacitással rendelkeznek. A következő táblázat ezeket az adatokat tartalmazza:

Lemeztípus	Lemez méret ["]	Tárolási kapacitás [bájt]
Egyoldalas, duplasűrűségű (SS DD)	5 1/4	179712
Kétoldalas, duplasűrűségű (DS DD)	5 1/4	362496
Kétoldalas, duplasűrűségű (DD)	3 1/2	730112
Kétoldalas, négyszeres sűrűségű (DS HD)	5 1/4	1213952
Kétoldalas, nagysűrűségű (HD)	3 1/2	1457664

Az adatok birtokában már könnyen kiszámíthatjuk a biztonsági másolat elkészítéséhez szükséges lemezek számát:

Lemezek száma = Felhasznált bájtok száma/tárolási kapacitás

Ha pl. a számítógép kétoldalas, duplasűrűségű lemezek fogadására alkalmas meghajtóval rendelkezik, akkor a szükséges lemezek száma — az előbbiekben kiszámított adat birtokában — a következő lesz:

Lemezek száma =  $38739968 / 362496$

Lemezek száma = 106.87

Lemezek száma = 107

A fenti számítást egy adott alkönyvtár vagy egy adott állomány esetében is hasonlóan kell elvégezni. A felhasznált bájtok számát ezekben az esetekben a DIR parancs segítségével lehet megtudni, ill. kiszámítani.

A BACKUP használatát megelőző második előkészítő lépés a lemezek formázása. Bár a 3.3-as és az e feletti DOS verziók esetében a /F kapcsoló előírására a BACKUP a lemez formázását is elvégzi, ezzel azonban jelentősen megnő a biztonsági másolatok elkészítésének ideje. Érdemesebb ezért a lemezek formázását előzetesen elvégezni.

## ***A BACKUP utasítás használata***

<b>2.0</b>
<b>KÜLSŐ</b>

Bár a BACKUP paranccsal a hajlékonylemezek biztonsági másolata is elkészíthető, ez a parancs azonban elsődlegesen a merevlemez tartalmának hajlékonylemezre írását végzi. Ehhez igazodva a továbbiakban minden példa esetében a merevlemez — C: meghajtó — tartalmát írjuk az A: meghajtóban levő hajlékonylemezre.

A BACKUP parancs legáltalánosabban használt alakja a következő:

```
C:\>BACKUP C:\*.* A: /S <ENTER>
```

Mint látható, a BACKUP parancs esetében is először a forráshelyet kell előírni — ez most a C: meghajtón levő összes állomány —, majd pedig a célhelyet, amely ebben az esetben az A: meghajtó, ill. az ebben elhelyezett lemezek. A /S kapcsoló arra utasítja a BACKUP parancsot, hogy az előírt könyvtár alatt elhelyezkedő összes könyvtár állományairól is készítse el a biztonsági másolatot. Mivel az előbbi esetben az előírt könyvtár a gyökér, így a parancs a teljes merevlemez tartalmát hajlékonylemezekre írja. Ha a merevlemez sok állományt tartalmaz, akkor a BACKUP csak egynél több hajlékonylemeze tudja a másolási műveletet elvégezni. (Az előbbi példa esetében a számítások szerint 107 lemeze!!!)

A BACKUP parancs két további fontos tulajdonsága:

1. Azoknál az állományoknál, amelyekre a biztonsági mentést végrehajtotta, törli az archív bitet. A DOS ui. az újonnan létrehozott vagy módosított állományokat automatikusan archívra, vagyis archiválandóra állítja be.
2. A lemezeire írt BACKUP.001 CONTROL.001 stb. állományokat csak olvasható attribútummal látja el.

A BACKUP parancs igazán rugalmasan csak a megfelelő kapcsolókkal együtt alkalmazható, ezért következő lépésként a BACKUP leggyakoribb kapcsolóit soroljuk fel.

## ***A BACKUP legfontosabb kapcsolói***

A BACKUP parancs a forrás- és a célhely megjelölésén kívül összesen hét különböző kapcsolóval segíti a biztonsági másolathoz felhasználandó állományok körének definiálását. Mivel a megfelelő kapcsolókkal sok időt, fáradságot és lemezt takaríthatunk meg, ezért érdemes ezekkel a kapcsolókkal megismerkedni. A következő táblázat a BACKUP kapcsolóit tartalmazza.

**Megjegyzés:** Ha az adott kapcsoló a 2.0 feletti DOS verziókban alkalmazható, akkor ezt külön jelezzük.

- | <b>Kapcsoló</b> | <b>Művelet</b>  |
|-----------------|---|
| /A              | A kapcsoló arra utasítja a BACKUP parancsot, hogy a céllemezén már elhelyezett BACKUP-állományokat kiegészítse azokkal, amelyeket a parancs forráshelyében előírtunk. A kapcsolónak két fontos tulajdonsága van: <ol style="list-style-type: none"> <li>1. Csak olyan lemezre adható ki, amely már BACKUP lemez.</li> <li>2. A kapcsoló hatására a BACKUP mechanikusan hozzáírja a BACKUP állományok végéhez a megjelölt állományokat. Az állományok kicserélését, ha a BACKUP állomány már tartalmaz azokból korábbi verziót, nem végzi el!</li> </ol> |
| /D:hh-nn-éé     | Arra utasítja a BACKUP parancsot, hogy csak azokról az állományokról készítse el a biztonsági másolatot, amelyek létrehozása, vagy módosítása a megadott napon, ill. azután történt.  |
| /F              | A kapcsoló hatására a BACKUP a céllemezeken formázását is elvégzi. Használatához 3.3-as, vagy e feletti DOS verzió szükséges. 4.0-s DOS esetén a kapcsoló után még a formázandó lemez mérete is — 160/180/320/360/720/1200/1440/2880 — előírható, pl. /F: 360.  |
| /L:naplófile    | A kapcsoló alkalmazása esetén a BACKUP egy speciális naplóállományba — az állomány alapértelmezése szerinti neve: BACKUP.LOG — feljegyzi azoknak az állományoknak a nevét, amelyekről a biztonsági másolatot elkészíti, valamint annak a lemeznek a számát, amely az adott állományt tartalmazza. A BACKUP a művelet időpontját is rögzíti.   |
| /M              | Arra utasítja a BACKUP parancsot, hogy csak azokról az állományokról készítsen biztonsági másolatot, amelyek archív attribútummal rendelkeznek. Ha az /M kapcsolót az /A kapcsoló nélkül használjuk, akkor a DOS törli a BACKUP lemezen már meglévő állományokat!   |
| /S              | A kapcsoló előírásának esetén a BACKUP a forráskönyvtár alatti összes alkönyvtár állományairól is elkészíti a biztonsági másolatot.   |
| /T:hh:mm:ss     | Arra utasítja a BACKUP parancsot, hogy csak azokról az állományokról készítse el a biztonsági másolatot, amelyek létrehozása vagy módosítása a megadott időpontban, ill. azután történt. A kapcsoló használatához 3.3-as, vagy e feletti DOS verzió szükséges!!!  |

A kapcsolók használatát a következő alfejezet konkrét példáin keresztül fogjuk bemutatni, ill. begyakorolni.

## ***A BACKUP végrehajtása adott állomány, adott alkönyvtár, ill. az egész merevlemez esetén***

Bár a BACKUP parancsot rendszerint az egész merevlemez biztonsági másolatának elkészítéséhez használják, a mindennapi munka során azonban a leggyakrabban olyan helyzetek adódnak, amikor csak egy adott alkönyvtár új vagy módosított állományait, vagy egyetlen, de igen nagy méretű állomány biztonsági másolatát akarjuk a merevlemezről a hajlékonylemezekre átvinni. Elsőként egy adott állomány — legyen ez a gyökérkönyvtár alatt elhelyezkedő SAJAT alkönyvtárban levő PELDA\_1.TXT állomány — biztonsági másolatának elkészítését mutatjuk be.

**Megjegyzés:** *a parancs futtatása előtt ne feledkezzünk meg a biztonsági másolat elkészítéséhez szükséges lemezek számának meghatározásáról. (L. Előkészületek a BACKUP használatához c. alfejezet.)*

```
C:\>BACKUP C:\SAJAT\PELDA_1.TXT A: <ENTER>
```

Ekkor a BACKUP parancs a következőket írja a képernyőre:

```
Insert backup diskette 01 in drive A:
```

```
WARNING! Files in the target drive  
A:\ root directory will be erased  
Press any key to continue...
```

```
Kérem helyezze az 01 számú lemezt az A: meghajtóba!
```

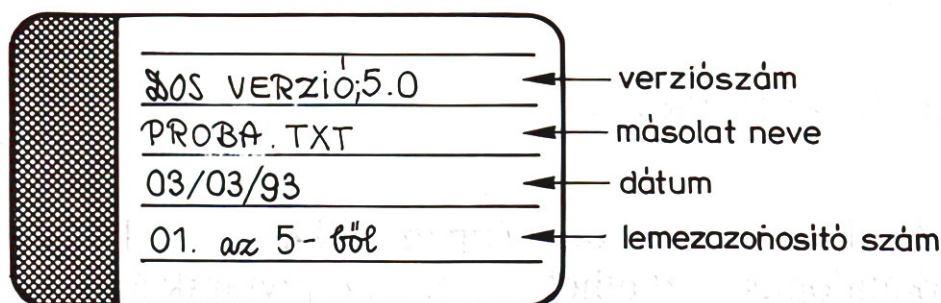
```
FIGYELMEZTETÉS! Az A:\ céllemez gyökérkatalógusában  
lévő állományok törlődnek.
```

```
A folytatáshoz kérem egy gomb lenyomását...
```

1. A fenti üzenet azt jelenti, hogy a BACKUP mielőtt a biztonsági mentéshez hozzákezdene, törli a céllemez tartalmát. A BACKUP tehát azon a leme-

- zen, amelyre a biztonsági mentést készítjük, semmilyen más állományt nem tűr meg, csak az általa elkészített BACKUP és CONTROL állományokat! Ha tehát véletlenül olyan lemezt helyeztünk a meghajtóba, amely számunkra fontos állományokat tartalmaz, a fenti üzenet megjelenése után a CTRL—BREAK lenyomásával lépünk ki a BACKUP műveletből.
2. Figyeljük meg, hogy a BACKUP az 01 számú lemez A: meghajtóba helyezésére szólít fel, vagyis a biztonsági másolathoz felhasznált lemezeket 1-től kezdve beszámozza, és ezt az adatot a lemezre mint kötetelnevezést ráírja. Amikor a lemezekről az állományokat a RESTORE parancs segítségével visszatöltjük, mindig megadott sorszámú lemez meghajtóba helyezésére kapunk felszólítást. Nagyon fontos tehát, hogy a biztonsági másolatok készítése során a lemezek címkéire feljegyezzük azt a sorszámot, amit a BACKUP a képernyőre kijelez. (Ha ezt elfelejtettük, akkor kérdezzük le a lemez kötetelnevezését a DIR vagy a VOL paranccsal.)

Mielőtt a meghajtóba helyeznénk a formázott lemezt, készítsük el a címkét, vagyis lássuk el a következő felirattal:



*Címkekészítés a BACKUP parancs használata során*

majd helyezük be az A: meghajtóba, és nyomjuk le valamelyik gombot!  
A BACKUP erre a

```
*** Backing up files to drive A:***
Diskette Number: 01
PELDA_1.TXT
```

```
*** Az állományok biztonsági mentése az A:
meghajtóba: ***
Lemezszám: 01
PELDA_1.TXT
```

felirattal jelzi, hogy a biztonsági mentéshez hozzákezdett, majd az állomány méretétől függően a második, ill. a további lemezek meghajtóba helyezésére szólít fel (a képernyő már ismerős):

```
Insert backup diskette 02 in drive A:  
WARNING! Files in the target drive  
A:\ root directory will be erased  
Press any key to continue...
```

Ha a biztonsági másolatot pl. a SAJAT elnevezésű alkönyvtárra akarjuk elkészíteni, akkor az előbbi parancsot a következőre kell módosítanunk:

```
C:\>BACKUP C:\SAJAT\*.* A: <ENTER>
```

Ha viszont a fenti műveletet az egész merevlemezre akarjuk kiterjeszteni, akkor a következő BACKUP parancsot kell használnunk:

```
C:\>BACKUP C:\*.* A: /S <ENTER>
```

A parancsnál a /S kapcsoló előírására mindenképpen szükség van, hogy a biztonsági másolat a gyökérkatalógus alatt elhelyezkedő könyvtárak állományairól is elkészüljön.

*A leggyakoribb hibák a BACKUP parancs használata során:*

1. A rendszer a BACKUP parancsot, mivel az külső parancs, nem találja.

*Védekezés:*

A BACKUP parancs futtatása előtt váltsunk át abba az alkönyvtárba, amely a parancsot tartalmazza, vagy írjuk elő a parancsot tartalmazó alkönyvtár elérési útját a PATH utasításban!

2. A BACKUP parancs használata előtt nem végezzük el az előkészítő műveleteket (lemezek számának meghatározása, lemezformázás).

3. A parancs előírásakor elfeledkezünk a forrás- és/vagy a célhely előírásáról.

*Védekezés:*

A parancsnak a biztonsági másolat elkészítéséhez mind a forrás-, mind a célhely előírására szüksége van, így ennek figyelembevételével javítsuk ki a parancs paramétereit.

4. A biztonsági másolatok elkészítésekor nem írjuk fel a lemezekre a BACKUP azonosító számait.

*Védekezés:*

A biztonsági másolat elkészítéséhez felhasznált lemezeket a következő adatokat tartalmazó címkével kell ellátni:

DOS verziószám

biztonsági másolat azonosító neve (állománynév/alkönyvtár/meghajtó)

a BACKUP művelet dátuma

a BACKUP lemezazonosító száma

## A BACKUP parancs

A BACKUP parancs a merevlemezen tárolt állományok biztonsági másolatának elkészítését végzi. A parancs használatát megelőzően a CHKDSK parancs segítségével meg kell határozni a művelethez szükséges hajlékonylemezek számát, valamint érdemes elvégezni ezek formázását is. A BACKUP használatakor a forráshelyet — amely a teljes merevlemez, annak egy alkönyvtára, de egyetlen állomány is lehet — és e célhelyet egyaránt elő kell írni. A BACKUP parancshoz összesen hét különböző kapcsoló rendelhető. Ezek közül külön kiemeljük a /S kapcsolót, amelynek alkalmazása esetén a BACKUP az előírt forráskönyvtár alatti összes könyvtár állományairól is biztonsági másolatot készít. Az /L kapcsolót azért érdemes alkalmazni, mert így a biztonsági másolathoz automatikusan naplóállomány készül. A /M és a /A kapcsoló a rendszeresen elvégzett biztonsági másolatok gazdaságos és gyors elkészítését teszi lehetővé.



## A RESTORE parancs

2.0
-----

KÜLSŐ
-------

A RESTORE paranccsal a biztonsági másolatok állományai visszatölthetők ugyanarra a helyre, ahonnan a biztonsági másolat elkészítése történt. Ha a BACKUP parancsot a /L kapcsolóval használtuk, akkor a biztonsági másolat-hoz naplóállomány is tartozik. A naplóállomány alapértelmezés szerinti neve: BACKUP.LOG. Tartalmát kinyomtatva megtudhatjuk, hogy a biztonsági másolat mikor készült, és milyen állományokat tartalmaz. Példaként állítsuk vissza az előbbiekben elmentett PELDA\_1.TXT állományt!

```
C:\>RESTORE A: C:\SAJAT\PELDA_1.TXT <ENTER>
```

Mint látható, ennél a parancsnál is először a forráshelyet, majd pedig a célhelyet kell előírni.

A RESTORE támogatja helyettesítő karakterek alkalmazását is, ha tehát az előbbi példához igazodva a SAJAT elnevezésű alkönyvtár állományait akarjuk visszaállítani, akkor a következő parancsot gépeljük be:

```
C:\>RESTORE A: C:\SAJAT\*. * <ENTER>
```

Végül azt a parancsot mutatjuk be, amellyel a merevlemez teljes tartalmának biztonsági másolata visszaírható:

```
C:\>RESTORE A: C:\*. * /S <ENTER>
```

A /S kapcsoló előírására a RESTORE a mentett könyvtárszerkezetet is visszaállítja.

## *A RESTORE parancs kapcsolói*

A RESTORE parancs a forrás- és a célhely megjelölésén kívül összesen kilenc különböző kapcsolóval teszi lehetővé a biztonsági másolat visszatöltésének minél hatékonyabb és jobb elvégzését. Mivel a megfelelő kapcsolókkal sok időt, fáradságot és lemezt takaríthatunk meg, ezért érdemes ezekkel a kapcsolókkal megismernedni. A következő táblázat a RESTORE kapcsolóit tartalmazza.

### **Kapcsoló      Művelet**

- /A:hh-nn-éé** Arra utasítja a RESTORE parancsot, hogy csak azokat az állományokat állítsa vissza, amelyek létrehozása, vagy módosítása a megadott napon, ill. az után történt.
- /B:hh-nn-éé** Arra utasítja a RESTORE parancsot, hogy csak azokat az állományokat állítsa vissza, amelyek létrehozása, vagy módosítása a megadott napon, ill. az előtt történt.
- /D** Ez a kapcsoló csak 5.0-s DOS esetén alkalmazható, és arra utasítja a RESTORE parancsot, hogy a visszaállítandó állományok nevét a képernyőre írja anélkül, hogy azokat ténylegesen visszaállítaná.
- /E:hh:mm:ss** Arra utasítja a RESTORE parancsot, hogy csak azokat az állományokat állítsa vissza, amelyek létrehozása vagy módosítása a megadott időpontban, ill. azelőtt történt.
- /L:hh:mm:ss** Arra utasítja a RESTORE parancsot, hogy csak azokat az állományokat állítsa vissza, amelyek létrehozása vagy módosítása a megadott időpontban, ill. az után történt.
- /M** A kapcsoló alkalmazása esetén a RESTORE csak azokat az állományokat állítja vissza, amelyek a legutolsó BACKUP művelet óta módosultak.
- /N** A kapcsoló alkalmazása esetén a RESTORE csak azokat az állományokat állítja vissza, amelyeket a merevlemez nem tartalmaz.
- /P** A RESTORE a kapcsoló előírásának esetén a felhasználótól külön engedélyt kér a csak olvasható vagy az utolsó BACKUP művelet óta megváltozott állományok visszaállításához.
- /S** A kapcsoló előírásának hatására a RESTORE az alkönyvtárakban levő állományokat is visszaállítja.

*A leggyakoribb hibák a RESTORE parancs használata során:*

1. A rendszer a RESTORE parancsot, mivel az külső parancs, nem találja.

*Védekezés:*

A RESTORE parancs futtatása előtt váltsunk át abba az alkönyvtárba, amely a parancsot tartalmazza, vagy írjuk elő a parancsot tartalmazó alkönyvtár elérési útját a PATH utasításban.

2. A parancs előírásakor megfelelünk a forrás- és/vagy a célhely előírásáról.

*Védekezés:*

A parancsnak a biztonsági másolat visszaállításához mind a forrás-, mind a célhely előírására szüksége van, így ennek figyelembevételével javítsuk ki a parancs paramétereit.

3. A visszaállítási művelet során nem azt a lemezt helyezük az A: meghajtóba, amelynek azonosító számát a parancs a képernyőre kijelzi.

*Védekezés:*

A BACKUP művelet végrehajtásakor a lemezek címkéjére rá kell írni azt az azonosító számot, amelyet a parancs a képernyőre ír.

## A RESTORE parancs

A RESTORE (visszaállítás) parancs a BACKUP paranccsal lemezre írt állományok visszaállítását végzi. A parancs használatakor a forráshelyet — ami rendszerint az A: meghajtó — és a célhelyet — ami rendszerint a me-revlemez egésze, esetleg egy alkatalógusa, nagyméretű állomány esetén pedig maga az állománynév — egyaránt meg kell jelölni. A visszaállítási művelet során mindig azt a lemezt kell a meghajtóba helyezni, amelynek sorszámát a parancs a képernyőre írja. (A lemezek sorszámát a BACKUP szolgáltatja.) A következő parancs a lemezekon levő összes állományt a C: meghajtóra írja vissza:

```
C:\>RESTORE A: C:\*.* /S <ENTER>
```

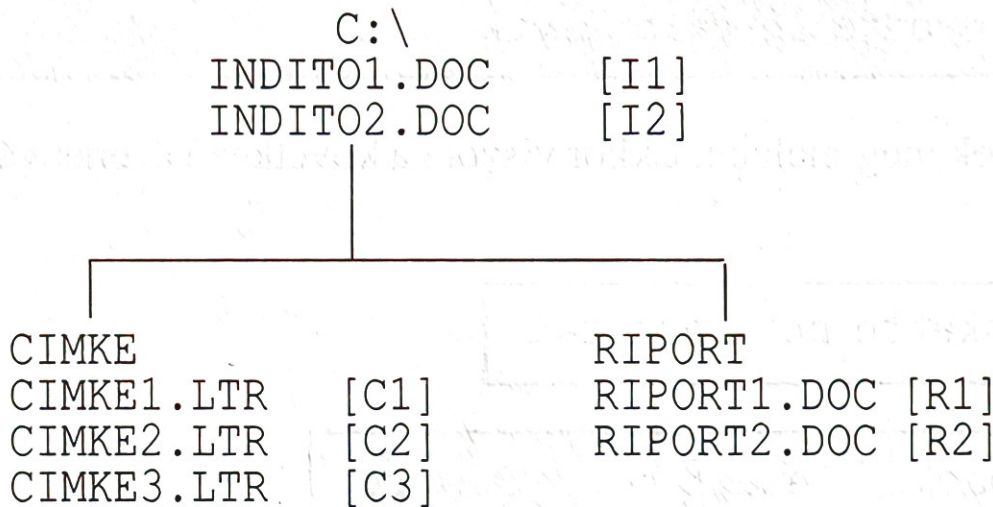
## *Gyakoroljuk a BACKUP és a RESTORE parancs használatát!*

A BACKUP és a RESTORE parancsot általában azért nem használják a felhasználók, mert még soha nem próbálták, vagy mert valaki egyszer azt mondta, hogy a parancsok hibásan működnek, mivel a RESTORE-ral „nem lehet” az állományokat visszaállítani.

A következő gyakorlatsor végrehajtásával adjunk esélyt magunknak a parancsok megismerésére és begyakorlására.

*A kiinduló helyzet legyen a következő:*

1. Hozzuk létre a következő szerkezetet a C: meghajtón! (Az állományok tartalmát szögletes zárójelek között adjuk meg. Persze, ettől el lehet térni.)



2. Vegyünk elő egy üres, formázott lemezt, és írjuk rá a következő állományt:

```
SAJAT.SAJ [SS]
```

Most pedig kezdjük hozzá a feladatokhoz! 5.0-s DOS használata esetén a gyakorlatok elvégzését a DOSKEY program használata nagymértékben megkönnyíti. Próbáljuk meg az eredményt még az előtt kitalálni, mielőtt a szövegben elolvassuk!

1. Váltunk át a WP alkönyvtárba, és kérdezzük le a létrehozott állományok attribútumát!

```
C:\WP>ATTRIB *.* /S <ENTER>
```

Mindegyik állományt most hoztuk létre, tehát valamennyi állomány az A, vagyis az archív attribútummal rendelkezik.

2. Tegyük fel, hogy a hajlékonylemezen elhelyezett SAJAT.SAJ állományra szükségünk van, egyébként azonban a lemez más állományt nem tartalmaz. Azt gondolva tehát, hogy a BACKUP állományok számára elég hely áll rendelkezésre, a következő parancsot hajtjuk végre:

```
C:\WP>BACKUP C:\WP A: /A <ENTER>
```

Erre a következő üzenetet írja a BACKUP a képernyőre:

```
Insert last backup diskette in drive A:  
Strike any key when ready
```

```
Kérem az utolsó backup lemez A: meghajtóba helyezését!  
Ha készen van, nyomja le valamelyik gombot!
```

Járjunk el a kérésnek megfelelően. Ekkor viszont a következő üzenet válik láthatóvá:

```
Last backup diskette not inserted
```

```
Nem az utolsó backup lemezt helyeztük be.
```

*Fontos következtetések:*

1. Az /A kapcsoló csak olyan lemezre használható, amely már tartalmaz BACKUP állományokat!
2. A SAJAT.SAJ állományt meg akarjuk őrizni, akkor másoljuk át egy másik lemezre.
3. Az /A kapcsoló törlése után hajtjuk végre ismét a fenti parancsot!

```
C:\WP>BACKUP C:\WP A: <ENTER>
```

Most először a már korábban tárgyalt és a lemeztartalom törlésére vonatkozó figyelmeztető üzenetet kapjuk. Nyomjuk le tehát valamelyik gombot, hogy a BACKUP a műveletet elvégezhesse!

A BACKUP, mint az üzenetekből kiderül, csak két állomány — INDITO1.DOC és INDITO2.DOC — biztonsági másolatát készíti el, mert ha másként nem írjuk elő, akkor csak az alapértelmezés szerinti könyvtárban dolgozik. Az A: meghajtóban elhelyezett lemez csak a BACKUP.001 és a CONTROL.001 állományt tartalmazza. Az ott eredetileg elhelyezett SAJAT.SAJ állományt tehát a BACKUP törölte. Az ATTRIB paranccsal az INDITO1.DOC és az INDITO2.DOC állomány attribútumát lekérdezve

```
C:\WP>ATTRIB INDITO?.*<ENTER>
```

tapasztalni fogjuk, hogy azok nem rendelkeznek attribútummal, tehát a BACKUP ezek archív attribútumát is törölte. Ha viszont az ATTRIB parancsot az A: meghajtóra adjuk ki

```
C:\WP>ATTRIB A: *.* <ENTER>
```

akkor látni fogjuk, hogy a két BACKUP állomány viszont az archívon kívül az R, vagyis a csak olvasható attribútummal is rendelkezik.

#### 4. Következő lépésként próbáljuk ki a /S kapcsoló hatását!

```
C:\WP>BACKUP C: A: /S <ENTER>
```

Mivel a /S kapcsolót is megadtuk, a BACKUP nemcsak az alapértelmezés szerinti, hanem az az alatt elhelyezkedő alkönyvtárak állományairól is elkészítette a biztonsági másolatot. A lemez előző tartalmát törölte, és ott új BACKUP és CONTROL állományokat hozott létre. Figyeljük meg továbbá, hogy most a BACKUP parancsban a forráshely előírásakor csak a C: meghajtót szerepeltettük. Ezt azért tehettük meg, mert ha másként elő nem írjuk, akkor a BACKUP a biztonsági másolat elkészítéséhez az aktuális könyvtárat használja, amely a prompt szerint az előbb a C:\WP volt. Ez viszont pont a számunkra megfelelő alkönyvtár.

Ha az ATTRIB parancsot a létrehozott egész könyvtárrendszerre futtatjuk

```
C:\WP>ATTRIB *.* /S
```

látni fogjuk, hogy egyik állomány sem rendelkezik már archív attribútummal.

#### 5. Most hogy már a mind a hét állományról van biztonsági másolatunk, töröljük a merevlemezen az összes létrehozott állományt,

```
C:\WP>DEL *.* <ENTER>
C:\WP>CD CIMKE <ENTER>
C:\WP\CIMKE>DEL *.* <ENTER>
C:\WP\CIMKE>CD \WP\RIPORT <ENTER>
C:\WP\RIPORT>DEL *.* <ENTER>
```

váltsunk vissza a WP alkönyvtárba, és állítsuk vissza az állományokat a lemeztől a RESTORE paranccsal!

```
C:\WP\RIPORT>CD.. <ENTER>
C:\WP>RESTORE A: C:\ *.* <ENTER>
```

A BACKUP-tól az

```
Insert backup diskette 01 in drive A:
Strike any key when ready.
```

üzenet után a

```
*** Files were backed up 04-12-1993 ***
*** Restoring files from drive A: ***
Diskette 01
```

tájékoztató üzenetek után a

```
Warning! No files were found to restore
```

```
Figyelmeztetés! A visszaállításhoz állományok nem
találhatók
```

figyelmeztető üzenetet kapjuk, mivel a visszaállítás helyeként nem azt az alkönyvtárat adtuk meg, ahonnan az állományok mentése történt — C:\WP — hanem a gyökérkönyvtárat. Javítsuk tehát a parancsot ennek megfelelően!

```
C:\WP>RESTORE A: C:\WP\ *.* <ENTER>
```

Bár a C:\WP\\*.\* előírás arra a szándékra utal, hogy valamennyi állományt vissza akarjuk állítani, a RESTORE — a megfelelő kapcsoló nélkül — mindig csak az adott alkönyvtárra vonatkozik. Mivel semmilyen kapcsolót nem írtunk elő, a parancs csak azt a két állományt állította vissza, amelyek a WP alkönyvtárban voltak. Nézzük meg viszont a két visszaállított állomány attribútumát!

```
C:\WP>ATTRIB *.* <ENTER>
```

A visszaállított állományok — mint látható — semmilyen attribútummal nem rendelkeznek.

6. Állítsuk ezután vissza az összes állományt a hajlékonylemezről!

```
C:\WP>RESTORE A: C:\WP\*.* /S <ENTER>
```

7. Módosítsuk az INDITO1.DOC és az INDITO2.DOC állomány tartalmát!

```
C:\WP>COPY CON INDITO1.DOC <ENTER>
```

```
I11 <ENTER>
```

```
^Z
```

```
1 file(s) copied
```

```
C:\WP>COPY CON INDITO2.DOC <ENTER>
```

```
I22 <ENTER>
```

```
^Z
```

```
1 file(s) copied
```

```
C:\WP>
```

8. Most pedig próbáljuk ki először az /A kapcsoló hatását!

```
C:\WP>BACKUP C:\WP A: /A /S <ENTER>
```

Mivel a /A kapcsolót előírtuk, a BACKUP az

```
Insert last backup diskette in drive A:  
Strike any key when ready
```

üzenetet küldi és a lemezen már meglévő BACKUP.001 állományt a két állománnyal kiegészíti. Győződjünk meg erről magunk is!



```
C:\WP>TYPE A:\BACKUP.001 <ENTER>
```

A megjelenő képernyőről leolvashatjuk, hogy az állomány tartalma a következő:

```
I1I2C1C2C3R1R2
I11
I22
C1C2C3R1R2
```

Tehát a BACKUP.001 az INDITO állományokból a két változatot, a többiből pedig 2-2 példányt tartalmaz.

De mi lesz akkor, ha ezt a BACKUP állományt majd visszaállítjuk? Az INDITO1.DOC és az INDITO2.DOC melyik verzióját kapjuk vissza, hiszen ugyanazzal a névvel egy alkönyvtárban csak egy állomány szerepelhet!

#### 9. Töröljük tehát a merevlemezen a két INDITO állományt!

```
C:\WP>DEL *.* <ENTER>
```

majd végezzük el a visszaállítást!

```
C:\WP>RESTORE A: C:\WP\ *.* <ENTER>
```

A RESTORE mind a négy INDITO állományt visszaállítja, ha azonban azok tartalmát a TYPE paranccsal egyesével lekérdezzük, látni fogjuk, hogy a C:\WP alkönyvtárban csak a két legutoljára létrehozott INDITO állomány szerepel.

#### 10. A visszaállított állományok — mint tudjuk — semmilyen attribútummal nem rendelkeznek, ezért a /M kapcsoló kipróbálásához, amely csak az archív attribútummal rendelkező állományok mentését végzi el, állítsuk be archívra a két INDITO állományt!

```
C:\WP>ATTRIB +A *.* <ENTER>
```

majd pedig futtassuk a következő parancsot:

```
C:\WP>BACKUP C:\WP A: /A /M /S <ENTER>
```

A kapcsolók sorrendje tetszőleges!

Bár a /S kapcsolót is előírtuk, a parancs mégis csak két állománnyal egészítette ki a BACKUP.001 állományt, mivel csak ezek rendelkeztek archív attribútummal. A /M kapcsoló használatánál külön felhívjuk a figyelmet a /A használatára. Az előbbi parancsot ui. a /A kapcsoló nélkül futtatva a BACKUP a lemezt előzetesen törli, és az új BACKUP.001 állományban csak a két archív attribútummal rendelkező állományt helyezi el. A BACKUP parancs előbbi alakja rendszeresen végrehajtott — pl. a napi munka végén — biztonsági másolatok készítésére alkalmas.

11. Végül 5.0-s DOS használata esetén még érdemes kipróbálni a RESTORE parancs /M kapcsolós változatát is, amely csak a két legutoljára mentett állomány visszaállítását teszi lehetővé.

```
C:\>RESTORE A C: WP\ *.* /M <ENTER>
```

## ***Ismétlés = Kulcs a sikerhez***

A BACKUP a merevlemez egészéről, vagy annak előírt helyéről a hajlékonylemezre biztonsági másolatot készít. A biztonsági másolat elkészítése során speciális formátumot használ, amelyet csak a RESTORE parancs tud értelmezni, így a hajlékonylemezre elkészített biztonsági másolat visszaállítását csak a RESTORE parancs tudja elvégezni. Ehhez a művelethez tehát sem a COPY, sem az XCOPY parancs nem alkalmazható. A /L kapcsoló alkalmazása esetén a BACKUP a másolással együtt egy speciális naplóállományt is elhelyez a lemezen — alapértelmezés szerinti állománynév: BACKUP.LOG —, amely a lemez azonosító számát, a tárolt állományok nevét és a művelet időpontját tartalmazza. A következőkben azt a két parancsot adjuk meg, amelyekkel a teljes merevlemez tartalma menthető, ill. visszaállítható:

```
C:\>BACKUP C:\ *.* A: /S <ENTER>
```

```
C:\>RESTORE A: C:\ *.* /S <ENTER>
```

## ***Mit jelent?***

**Nagyméretű állomány.** Olyan állomány, amely méreténél fogva a DOS állománykezelő parancsai által nem kezelhető.

# *További tudnivalók a parancsállományokkal kapcsolatban*

**Már tanultuk:** Parancsállomány, AUTOEXEC.BAT

**Parancsállomány:** olyan szöveges állomány, amely általában több DOS és speciális, csak parancsállományban használható parancsot tartalmaz. A parancsállományoknak .BAT kiterjesztésűeknek kell lenniük. Ha a felhasználó a DOS prompt után egy parancsállomány nevét írja be, akkor a DOS az abban szereplő parancsokat futtatja.

**AUTOEXEC.BAT:** speciális parancsállomány, amelyet a DOS az operációs rendszer betöltése után a **gyökérfiókban** — **de csak itt!** — automatikusan keres. Ha az állományt megtalálja, akkor a benne foglalt parancsokat egymás után — vagyis azok előírásának sorrendjében — futtatja. Az AUTOEXEC.BAT állományban azokat a parancsokat érdemes elhelyezni, amelyeket minden egyes rendszerindítás után futtatni akarunk. Ha a DOS az AUTOEXEC.BAT állományt nem találja meg, akkor a már ismert DATE és TIME parancsot automatikusan indítja.

## *Speciális — csak, ill. első sorban parancsállományban alkalmazható — DOS parancsok*

A DOS-nak nyolc olyan parancsa van, amelyeket **csak, ill. első sorban** parancsállományokban szokás alkalmazni. A következő táblázat ezeket a parancsokat tartalmazza.

A parancsok mindegyike belső parancs!

Parancs	Elvégzett művelet	Megjegyzés
CALL	Az egyik parancsállományból a másik parancsállomány futtatása.	<b>3.3</b> -as vagy magasabb DOS verzió esetén
ECHO	Üzenetek megjelenítése a képernyőn, ill. a parancsnevek képernyőre írásának letiltása.	<b>2.0</b> vagy magasabb DOS verzió esetén
FOR *	Parancs vagy parancsok végrehajtásának ismételtetése.	<b>2.0</b> vagy magasabb DOS verzió esetén
GOTO	A parancsállomány végrehajtásának folytatása a GOTO után megadott helytől (címkétől).	<b>2.0</b> vagy magasabb DOS verzió esetén
IF	Parancs feltételhez kötött végrehajtása.	<b>2.0</b> vagy magasabb DOS verzió esetén.
PAUSE	A parancsállomány végrehajtásának ideiglenes felfüggesztése.	A parancsot <b>valamennyi</b> DOS verzió támogatja
REM	Emlékeztető megjegyzések elhelyezése a parancsállomány utasításai között.	A parancsot <b>valamennyi</b> DOS verzió támogatja
SHIFT	Kilencnél több parancssor paraméter alkalmazásának lehetővé tétele.	<b>2.0</b> vagy magasabb DOS verzió esetén

\* A FOR parancsot nemcsak a parancsállományokban, hanem — más formátummal!! — a DOS parancssorban is lehet alkalmazni.

Ezen speciális parancsok közül az **EGYSZERŰEN** DOS c. könyvben csak az ECHO ON/OFF, ill. az @ECHO parancsot tanultuk. Ebben a fejezetben azonban a többi paranccsal is megismerkedünk. Mielőtt azonban a parancsok tárgyalásához hozzákezdénénk, a parancsállományok egy további sajátosságát, a parancsállomány paramétereit tárgyaljuk.

**Fontos megjegyzés:** Ebben, és a következő három fejezetben a parancsok bonyolultsága, ill. a teljesség igénye miatt a parancsok szintaxisát is minden esetben megadjuk. A szintaxis összetevőinek magyarázatát a 0. Fejezetben tárgyaljuk részletesen.

## Parancsállomány paraméterek

### 2.0

A különböző DOS parancsok használatakor már többször előfordult, hogy a parancs neve után paramétereket is megadtunk, ill. meg kellett adnunk. Így volt ez a COPY parancsnál is, amelynél első paraméterként a forrás-, második paraméterként a célállományt kell előírni. Felmerül a kérdés, hogy a saját magunk készítette parancsállományok esetében nincs-e lehetőség ilyen megoldásra? Hiszen ezzel az adott parancsállomány használhatósága nagymértékben megnőne. Hogy miért? A következő példákban ezt érthetjük meg.

A parancsállományok elsődleges célja a felhasználó gépelési munkájának csökkentése. Készítsük el tehát a T.BAT parancsállományt, amely a SAJAT alkönyvtárban levő PROBA\_1.TXT állomány tartalmát írja a képernyőre:

```
C:\SAJAT>COPY CON T.BAT <ENTER>
TYPE C:\SAJAT\PROBA_1.TXT <ENTER>
^Z <ENTER>
    1 File(s) copied
C:\SAJAT>
```

Ha tehát a T.BAT parancsállományt elkészítettük, a PROBA\_1.TXT képernyőre íratásához a következőket kell a parancssorba írni:

```
C:\SAJAT>T <ENTER>
```

Ennek a parancsállománynak azonban van egy komoly szépséghibája. Csak a PROBA\_1.TXT állomány képernyőre íratásához használható. Mennyivel hatékonyabb lenne ez a parancsállomány, ha a T után, vagyis a parancsállomány neve után meg lehetne adni annak a szöveges állománynak a nevét, amelyet a képernyőre akarunk íratni. A DOS ezt is lehetővé teszi, a parancsállomány nevének beírását követően ui. összesen kilenc különböző paraméter megadására nyújt lehetőséget. De hogyan lehet ezekhez a paraméterekhez magából a parancsállományból hozzáférni? Tudnunk kell, hogy a DOS ezekhez a paraméterekhez a következő változókat automatikusan hozzárendeli:

- %0 A parancsállomány nevét tartalmazza
- %1 Az első parancsállomány paramétert tartalmazza
- %2 A második parancsállomány paramétert tartalmazza

%9 A kilencedik parancsállomány paramétert tartalmazza

Példaként vegyük a következő parancssort:

```
C:\SAJAT>SBAT AA BB CC <ENTER>
```

A DOS a parancsállomány nevét — SBAT —, valamint az AA, BB, CC paramétereket rendre a %0, %1, %2, %3 változókhoz rendeli hozzá.

%0 Aktuális értéke: SBAT

%1 Aktuális értéke: AA

%2 Aktuális értéke: BB

%3 Aktuális értéke: CC

A parancsállomány paraméterek ismeretében módosítsuk úgy az előbbi T.BAT állományt, hogy az a parancssorban megadott állomány tartalmát írja a képernyőre:

```
C:\SAJAT>COPY CON T.BAT <ENTER>
```

```
TYPE %1 <ENTER>
```

```
^Z <ENTER>
```

```
1 File(s) copied
```

```
C:\SAJAT>
```

A fenti T.BAT állomány elkészítése után pl. a PROBA\_1.TXT képernyőre íratásához csak a következőket kell tennünk:

```
C:\SAJAT>T PROBA_1.TXT <ENTER>
```

Itt jegyezzük meg, hogy a DOS parancssorában megadott argumentumokat szóközökkel, tabulátorkarakterrel, vesszővel vagy pontosvesszővel kell és lehet egymástól elválasztani. Ennek értelmében tehát

EZ-EGY-PARAMÉTER

csak **egy** paraméternek számít.

Lássunk még egy példát a parancsállomány paraméterek használatára.

A következő parancsállomány a fontos állományok másolásakor nyújt hasznos szolgálatot:

```
C:\SAJAT>COPY CON BIZTOS.BAT <ENTER>
COPY %1 %2 <ENTER>
COMP %1 %2 <ENTER>
^Z <ENTER>
    1 File(s) copied
C:\SAJAT>
```

A parancsállomány működése a következő: a BIZTOS után megadott első állományt a másodikként megadott állományba másolja, majd pedig a két állomány tartalmát összehasonlítja. Például a

```
C:\SAJAT>BIZTOS LABEL.COM LABEL.BAK <ENTER>
```

futtatásakor az %1 változó a LABEL.COM, a %2 változó a LABEL.BAK értéket veszi fel. A DOS a parancsállományban előírtaknak megfelelően először a LABEL.COM állományról másolatot készít — ez lesz a LABEL.BAK —, majd pedig a két állomány tartalmát összehasonlítja.

Utolsó példaként egy olyan parancsállományt fogunk elkészíteni, amely a paramétereket a képernyőre írja:

```
C:\SAJAT>COPY CON MUTAT.BAT <ENTER>
@ECHO %0 %1 %2 %3 %4 %5 %6 %7 %8 %9 <ENTER>
^Z <ENTER>
    1 File(s) copied
C:\SAJAT>
```

**Először paraméterek nélkül futtassuk a parancsállományt!**

```
C:\SAJAT>MUTAT <ENTER>
MUTAT
C:\SAJAT>
```

Mivel nem adtunk meg paramétert, csak a %0 változó kapott értéket, amely a parancsállomány nevét tartalmazza, ezért az ECHO csak ezt írhatta a képernyőre. Most pedig adjunk meg a MUTAT után néhány paramétert is!

```
C:\SAJAT>MUTAT 11 22 33 44 55 66 77 <ENTER>
```

Erre a következő feliratot fogjuk látni:

```
MUTAT 11 22 33 44 55 66 77  
C:\SAJAT>
```

Adjunk meg végül a MUTAT parancsállomány paramétereként egy szöveget!

```
C:\SAJAT>MUTAT EZ EGY SZÖVEG <ENTER>
```

Ekkor a képernyőn a

```
MUTAT EZ EGY SZÖVEG
```

felirat jelenik meg. Megjegyezzük, hogy az utolsó futtatással a %0, %1, %2 és %3 változók kaptak értéket.

## Parancsállomány-paraméterek

A DOS a parancsállományok neve után paraméterek megadását is megengedi. A megadott paramétereket automatikusan az %1, ..., %9 változókhoz rendeli hozzá, a parancsállományon belül tehát így kell és szabad ezekre hivatkozni. A %0 mindig a parancsállomány nevét jelenti. A paramétereket a DOS csak akkor tekinti különbözőnek, ha azokat a szóköz, tabulátor, vessző vagy pontosvessző karakterekkel választjuk el egymástól.

## *Kilencnél több parancsállomány paraméter használata (SHIFT)*

Mint az előző alfejezetben olvashattuk, a DOS a %1—%9 parancsállomány paraméterek alkalmazását automatikusan támogatja. Ha azonban kilencnél több parancsállomány paraméter használatára kényszerülünk, akkor a SHIFT parancsot kell használnunk. A SHIFT olyan belső parancs, amelynek hatására a DOS az egyes paramétereket egy pozícióval balra lépteti. Lássunk mindjárt egy példát a SHIFT használatára!



```
C:\>COPY CON LEPTET.BAT <ENTER>
@ECHO OFF <ENTER>
ECHO %0 %1 %2 %3 %4 %5 %6 %7 %8 %9 <ENTER>
SHIFT <ENTER>
ECHO %0 %1 %2 %3 %4 %5 %6 %7 %8 %9 <ENTER>
^Z <ENTER>
    1 File(s) copied
C:\>
```

A parancsállományt futtatva

```
C:\>LEPTET 1 2 3 4 5 <ENTER>
```

a következőket olvashatjuk a képernyőn:

```
LEPTET 1 2 3 4 5
1 2 3 4 5
C:\>
```

Tehát ha a DOS a SHIFT paranccsal találkozik, akkor az egyes paramétere-  
ket egy hellyel balra lépteti, amelynek eredményeképpen az %1-es parancsál-  
lomány paraméter a %0-ba, a %2-es az %1-esbe lép és így tovább.

## ***A parancsállományok egyszerű parancsai (REM, PAUSE, CALL)***

A bonyolultabb parancsállományokat érdemes megjegyzésekkel dokumen-  
tálni, mert így a későbbiekben is bármikor könnyen kideríthetjük egy-egy sor  
értelmét. A megjegyzések a REM (emlékeztető) parancs felhasználásával  
helyezhetők el a parancsállományban. A REM parancs általános alakja a kö-  
vetkező:

```
REM [szöveg]
```

vagyis a REM után lehet elhelyezni a megjegyzés szövegét. Ha a DOS a REM  
paranccsal találkozik, akkor magát a parancsszót, valamint az utána következő  
és vele egy sorban szereplő szöveget — a szöveg maximum 123 karakterből  
állhat — figyelmen kívül hagyja. A következő példa a REM használatát mutat-  
ja be.

```
C:\SAJAT>COPY CON PROBA.BAT <ENTER>
REM ***** <ENTER>
REM Az ECHO OFF parancs hatására <ENTER>
REM a DOS a parancsállományból indított <ENTER>
REM parancssorokat nem írja a képernyőre. <ENTER>
REM ***** <ENTER>
```

```
ECHO OFF <ENTER>
```

```
REM ***** <ENTER>
REM Az ECHO után megadott szöveget a DOS <ENTER>
REM a parancsállomány végrehajtásakor <ENTER>
REM a képernyőre írja. <ENTER>
REM ***** <ENTER>
```

```
ECHO Ez az ECHO parancs! <ENTER>
```

```
REM ***** <ENTER>
REM Ha az ECHO OFF parancs előtt a @ karaktert <ENTER>
REM is elhelyezzük, akkor még az ECHO OFF <ENTER>
REM parancs képernyőre írását is letiltjuk. <ENTER>
REM (3.3 vagy ezt követő DOS esetén.) <ENTER>
REM ***** <ENTER>
```

```
@ECHO OFF <ENTER>
```

```
REM ***** <ENTER>
REM üres sor elhelyezése: 5.0-s DOS esetén <ENTER>
REM az ECHO után egy pont megadásával; korábbi <ENTER>
REM DOS verzióknál az ECHO után az ALT lenyomva <ENTER>
REM tartása alatt a numerikus billentyűzeten <ENTER>
REM a 255 beírásával. <ENTER>
REM ***** <ENTER>
```

```
ECHO Ez egy <ENTER>
```

```
ECHO. <ENTER>
```

```
ECHO üres sor. <ENTER>
```

```
^Z <ENTER>
```

```
1 File(s) copied
```

```
C:\SAJAT>
```

Ha a parancsállományt futtatjuk,

```
C:\SAJAT>PROBA.BAT <ENTER>
```

a következőket olvashatjuk a képernyőn:

```
C:\SAJAT>ECHO OFF
Ez az ECHO parancs!
Ez egy
üres sor.
C:\SAJAT>
```

Mint látható, a REM után elhelyezett sorok a képernyőn nem jelennek meg. Csak a parancsállomány belső dokumentálására használandók!

**Fontos megjegyzés:** A REM parancsnév után a szöveget nem szabad azonnal elkezdni. **Egy szóközt kötelező hagyni** — mint minden parancsszó után —, ellenkező esetben ui. a DOS nem képes a parancsot értelmezni, amelyet a következő hibüzenettel jelez:

```
Bad command or file name
```

```
Hibás parancs vagy fájlnev
```

A következő egyszerű parancs a PAUSE (szünet), amellyel a parancsállomány végrehajtása egy billentyű lenyomásáig ideiglenesen szüneteltethető. Mikor van erre szükség? Például ha a parancsállomány végrehajtása során a felhasználóval bizonyos feladatokat — pl. a nyomtató bekapcsolása, vagy a meghajtóban a lemez kicserélése — akarunk végrehajtani. Próbáljuk ki ezt a parancsot is!

```
C:\SAJAT>COPY CON SZUNET.BAT <ENTER>
PAUSE Bekapcsoltad a nyomtatót?
^Z <ENTER>
1 File(s) copied
C:\SAJAT>
```

A parancsállományt futtatva a következők olvashatók a képernyőn:

```
C:\SAJAT>PAUSE Bekapcsoltad a nyomtatót?  
Press any key to continue ...
```

```
C:\SAJAT>PAUSE Bekapcsoltad a nyomtatót?  
A folytatáshoz kérem valamelyik gomb lenyomását...
```

Nyomjuk le tehát valamelyik gombot, melyre a DOS — mivel a parancsállomány több utasítást nem tartalmaz — a parancsállomány végrehajtását befejezi és visszaadja a promptot.

Az ECHO OFF vagy az @ECHO OFF a PAUSE után megadott üzenet képernyőre írását is letiltja. Ha tehát az előbbi SZUNET.BAT állományt az @ECHO OFF paranccsal kiegészítjük

```
C:\SAJAT>COPY CON SZUNET.BAT <ENTER>  
@ECHO OFF  
PAUSE Bekapcsoltad a nyomtatót?  
^Z <ENTER>  
1 File(s) copied  
C:\SAJAT>
```

akkor bármennyire is várjuk, csak a

```
Press any key to continue ...
```

```
A folytatáshoz kérem valamelyik gomb lenyomását
```

üzenetet fogjuk látni a képernyőn. Mi adja tehát ebben az esetben a megoldást?

Ha olyan parancsállományt készítünk, amely sok utasítást tartalmaz, ezért az @ECHO OFF paranccsal már az első sorban letiltjuk a parancssorok képernyőre való kijelzését, viszont a PAUSE parancsot is használni akarjuk, a megjelenítendő szöveget az ECHO-val írassuk a képernyőre:

```
C:\SAJAT>COPY CON SZUNET.BAT <ENTER>
@ECHO OFF <ENTER>
ECHO Bekapcsoltad a nyomtatót? <ENTER>
PAUSE <ENTER>
^Z <ENTER>
    1 File(s) copied
C:\SAJAT>
```

Parancsállományok készítésére rendszerint akkor kerül sor, ha több feladat végrehajtását néhány billentyű — a parancsállomány nevének begépelése — lenyomására akarjuk korlátozni. Könnyen előfordulhat tehát, hogy új parancsállományt készítve egy másik parancsállomány tartalmára is szükségünk van. Ilyen esetben ahelyett, hogy a parancsokat ismételten begépelnénk, a CALL segítségével az adott parancsállományt meghívhatjuk, vagyis beleágyazhatjuk abba a parancsállományba, amit éppen készítünk. Tegyük fel, hogy a képernyőt letörlő és a DOS verzió számát kijelző VERZIO.BAT állományból az előbbiekben elkészített SZUNET.BAT parancsállományt akarjuk meghívni.

```
C:\SAJAT>COPY CON VERZIO.BAT <ENTER>
CLS <ENTER>
CALL SZUNET <ENTER>
VER <ENTER>
^Z <ENTER>
    1 File(s) copied
C:\SAJAT>
```

Az új VERZIO.BAT állományt futtatva a következőket láthatjuk a képernyőn:

```
C:\SAJAT>CALL SZUNET
Bekapcsoltad a nyomtatót?
Press any key to continue ...
```

És amint valamelyik gombot lenyomjuk:

```
C:\SAJAT>VER
MS-DOS Version 5.00
C:\SAJAT>
```

Bizonyára felmerült az olvasóban a kérdés, hogy mi történik akkor, ha a CALL parancsot elhagyjuk és csak annak az állománynak a nevét adjuk meg, amelyet meg akarunk hívni. Próbáljuk ki ezt a változatot is!

```
C:\SAJAT>COPY CON VERZIO.BAT <ENTER>
CLS <ENTER>
SZUNET <ENTER>
VER <ENTER>
^Z <ENTER>
    1 File(s) copied
C:\SAJAT>
```

A parancsállományt futtatva azt tapasztaljuk, hogy a VER parancs végrehajtására már nem kerül sor. Vagyis ha az egyik parancsállományból a másik parancsállományt a CALL nélkül hívjuk meg, akkor a DOS annak utasításait végrehajtja, az eredeti parancsállományba azonban már nem tér vissza.

A **REM** parancs segítségével a parancsállományokban megjegyzések és magyarázatok helyezhetők el. A DOS a parancsállomány végrehajtásakor a REM-mel kezdődő sorokat figyelemen kívül hagyja. Többsoros megjegyzéseket csak úgy lehet elhelyezni, hogy a sorok elején a REM parancsot ismételtelen előírjuk.

A **PAUSE** parancs előírására a DOS a parancsállomány végrehajtását csak akkor folytatja, ha a felhasználó a billentyűzeten egy gombot lenyom. A PAUSE után üzenet is megadható, amely viszont csak akkor jelenik meg a képernyőn, ha a parancsállományban a parancssor megjelenése nincs letiltva, azaz ha a PAUSE sort megelőzően az ECHO OFF parancsot nem tartalmazza.

A **CALL** parancs az egyik parancsállományból másik parancsállomány meghívását teszi lehetővé.

## *A parancsállomány feltételes programozása, az IF parancs használata*

A DOS, hacsak kifejezetten másként nem utasítjuk, a parancsállományba foglalt utasításokat egymás után hajtja végre. Könnyen előfordulhat azonban, hogy egy adott műveletet csak akkor akarunk végrehajtani, ha bizonyos feltételek teljesülnek. Ezt az IF parancs használatával érhetjük el. Általános alakja:

IF feltétel parancs

Ha tehát az IF után megfogalmazott feltétel igaz, akkor a DOS az előírt parancsot végrehajtja, ellenkező esetben viszont a parancsállomány következő sorára ugrik és azt hajtja végre. Az IF után a következő három feltétel egyike szerepelhet:

1. Valamely állomány létezésének ellenőrzése
2. Két karakterlánc azonosságának ellenőrzése
3. Vizsgálat a parancsállomány IF-et megelőző sorában szereplő utasítás végrehajtásával kapcsolatban

Vegyük sorra részletesen a három feltételt:

1. Ha valamilyen művelet végrehajtását egy adott állomány létezéséhez akarjuk kötni, akkor a következő alakú IF parancsot kell alkalmaznunk:

```
IF EXIST [meghajtó_betűjele:][elérési_út]filenév[.kit]  
parancs
```

(Az IF EXIST magyar jelentése: ha létezik.)

Az állománynév előírásakor helyettesítő karakterek is alkalmazhatók. Lásunk mindjárt egy példát az IF EXIST használatára! Tegyük fel, hogy az A: meghajtón levő PROBA\_1.TXT állományt, ha az létezik, át akarjuk másolni az éppen aktuális meghajtóra. Ezt a következő IF EXIST parancs tudja megvalósítani:

```
IF EXIST A:PROBA_1.TXT COPY A:PROBA_1.TXT *.*
```

2. Az IF után alkalmazható második feltételtípus két karakterlánc azonosságának vizsgálata és ettől függően a megadott parancs végrehajtása. Ebben az esetben az IF parancs alakja a következő:

```
IF karakterlánc1 == karakterlánc2 parancs
```

Ezen feltételtípus igazi haszna csak akkor jelentkezik, ha a parancsállománynak a parancssorból paramétereket is átadunk, amelyeket ezzel az IF feltétellel tudunk megvizsgálni. Különösebb magyarázat helyett lássunk mindjárt néhány példát!

Tegyük fel, hogy ugyanazt a számítógépet három alkalmazottunk használja, és mindegyikük számára üzenetet akarunk hagyni. INDITO névvel tehát a következő parancsállományt készítjük el:

```
C:\SAJAT>COPY CON INDITO.BAT <ENTER>
@ECHO OFF <ENTER>
IF %1 == MARI ECHO Fejezd be a tegnapi levelet. <ENTER>
IF %1 == LACI ECHO Menj el a postára. <ENTER>
IF %1 == FERI ECHO Hívjál fel telefonon. <ENTER>
^Z <ENTER>
    1 File(s) copied
C:\SAJAT>
```

Ha az alkalmazott az INDITO parancsot a nevével együtt futtatja, pl.

```
C:\SAJAT>INDITO MARI <ENTER>
```

a következő felirat jelenik meg a képernyőn:

```
Kérlek, fejezd be a tegnapi levelet.
C:\SAJAT>
```

Most pedig paraméter nélkül futtassuk az INDITO.BAT állományt!

```
C:\SAJAT>INDITO <ENTER>
```

Erre a képernyőn a következő hibüzenetek jelennek meg:



```
Syntax error
Syntax error
Syntax error
C:\SAJAT>
```

A *syntax error* magyarul szintaxis hibát jelent. Miért kaptuk ezt az üzenetet?

Az IF-nek az összehasonlításához két karakterláncra van szüksége. Ha viszont csak egy karakterlánc áll rendelkezésére — mint ebben az esetben —, akkor a fenti hibaüzenetet küldi. Az IF parancs ezen változatáról azonban még azt is tudnunk kell, hogy a karakterláncok megadásakor az idézőjelek használatát nem várja el kötelezően. (A korábbi példa ezt igazolja.) Ha azonban a %1 változót idézőjelek közé tesszük, és a parancsállományt paraméter megadása nélkül futtatjuk, akkor az IF az ún. üres karakterláncot — tehát karakterláncot! — hasonlít össze karakterláncsal. Ahhoz tehát, hogy a fenti hibaüzeneteket elkerüljük az INDITO parancsállomány paraméter nélküli futtatásakor, végezzük el a következő módosításokat:

```
C:\SAJAT>COPY CON INDITO.BAT <ENTER>
@ECHO OFF <ENTER>
IF '%1' == 'MARI' ECHO Fejezd be a tegnapi levelet. <ENTER>
IF '%1' == 'LACI' ECHO Menj el a postára. <ENTER>
IF '%1' == 'FERI' ECHO Hívjál fel telefonon. <ENTER>
^Z <ENTER>
    1 File(s) copied
C:\SAJAT>
```

Futtassuk ezután az INDITO parancsállományt!

```
C:\SAJAT>INDITO <ENTER>
C:\SAJAT>
```

Mint látható, a hibaüzeneteket már kiküszöböltük, a felhasználó számára azonban hasznos lenne a tájékoztatás a parancsállomány használatáról. A megoldást a következő sor adja:

```
IF '%1' =='' DOS_PARANCS
```

Az INDITO.BAT állomány első IF soraként tehát a következőt érdemes beírni:

```
.IF '%1' =='' ECHO Az INDITO után a neved begépelését  
kérem!
```

**Megjegyzés:** A karakterlánc jelölésére az idézőjelen kívül a macskaköröm is használható. Egyedül az a lényeges, hogy ugyanazon a soron belül a két egyenlőségjel jobb és bal oldalán egységes legyen a jelölés. Ennek értelmében tehát az előbbi INDITO.BAT állomány a következő is lehet:

```
C:\SAJAT>COPY CON INDITO.BAT <ENTER>  
@ECHO OFF <ENTER>  
IF "%1" == "MARI" ECHO Fejezd be a tegnapi levelet. <ENTER>  
IF '%1' == 'LACI' ECHO Menj el a postára. <ENTER>  
IF "%1" == "FERI" ECHO Hívjál fel telefonon. <ENTER>  
^Z <ENTER>  
1 File(s) copied  
C:\SAJAT>
```

3. Vannak olyan DOS parancsok — programok —, amelyek a futásukat befejezve egy belső DOS változót állítanak be aszerint, hogy szabályosan vagy valamilyen program által el nem hárítható külső hiba következtében fejezték-e be a működésüket. Ha a program hibamentesen futott le, akkor ez az ún. hibaváltozó — amely a DOS-ban az ERRORLEVEL nevet viseli — 0 értékű, ellenkező esetben nullától eltérő értéket kap. Bár a hibaváltozó nullától eltérő értékének ismerete bizonyos esetekben fontos lehet, általában csak az ERRORLEVEL zérus értékére szokás vizsgálatot végezni. A hibaváltozó értékét a következő IF paranccsal lehet megvizsgálni:

```
IF ERRORLEVEL n parancs
```

Az ERRORLEVEL a parancsállomány IF-et megelőző sorában végrehajtott parancs sikeres vagy sikertelen futásának megfelelően kap értéket, amelyet az *n* helyébe beírt értékkel hasonlít össze az IF parancs. Ha az ERRORLEVEL nagyobb vagy egyenlő — a nagyobb szót külön hangsúlyozzuk! — az *n* helyébe beírt értékkel, akkor az összehasonlítás igaz eredményt ad, és az előírt

DOS parancs végrehajtásra kerül. Ellenkező esetben viszont a parancsállományt a DOS az IF után következő sorral folytatja. A bevezető után lássunk egy példát!

A 2. Fejezetben tárgyaltuk a DISKCOPY parancsot, amely lemezmásolatok készítését teszi lehetővé. Ez a parancs a számára megszabott művelet elvégzésétől függően az ERRORLEVEL-t a 0—4 értékek valamelyikére állítja be.

**Megjegyzés:** Az *A függelék* minden olyan DOS parancsot tartalmaz, amely az ERRORLEVEL értékét a futása során beállítja. Az ERRORLEVEL értékeket és azok értelmezését ugyancsak megadjuk.

Ezt figyelembe véve tehát a következő parancsállományt lehet elkészíteni:

```
C:\SAJAT>COPY CON MASOLAS.BAT <ENTER>
@ECHO OFF <ENTER>
DISKCOPY A: B: <ENTER>
IF ERRORLEVEL 0 ECHO Sikeres másolás!
IF ERRORLEVEL 1 ECHO Nem végzetes lemezhiba!
IF ERRORLEVEL 2 ECHO A Ctrl-C-t lenyomta a felhasználó!
IF ERRORLEVEL 3 ECHO Végzetes lemezhiba!
IF ERRORLEVEL 4 ECHO Érvénytelen lemezmeghajtó!
^Z <ENTER>
C:\SAJAT>
```

**Megjegyzés:** A parancsállomány elkészítése során vigyázzunk a szintaxisra. Az IF ERRORLEVEL után ui. nincs egyenlőségjel, csak szóköz!

Futtassuk a MASOLAS parancsállományt, és amikor a DISKCOPY a forráslemez meghajtóba helyezésére szólít fel, nyomjuk le a Ctrl—C billentyűkombinációt, vagyis szakítsuk meg a művelet végrehajtását! Ekkor a DISKCOPY az ERRORLEVEL értékét 2-re állítja be. Ez az érték azonban az IF ERRORLEVEL 0 és az IF ERRORLEVEL 1 feltételek vizsgálatakor is igaz eredményt ad, mivel a 2 mind a 0-nál, mind az 1-nél nagyobb. Az ERRORLEVEL és az *n* összehasonlításakor ui. igaz eredményt ad az is, ha az ERRORLEVEL az *n*-nél nagyobb! Így tehát a parancsállomány a feltételt teljesítő valamennyi!! üzenetet a képernyőre írja, vagyis:

```
Sikeres másolás!  
Nem végzetes lemezhiba!  
A Ctrl-C-t lenyomta a felhasználó!
```

Ez nagyon zavaró, hiszen csak az utolsó üzenet felel meg a valóságnak. A következőkben ezért megadjuk, hogyan kell egy bizonyos ERRORLEVEL értékre a vizsgálatot elvégezni:

```
IF ERRORLEVEL 1 IF NOT ERRORLEVEL 2 ECHO Egy  
IF ERRORLEVEL 2 IF NOT ERRORLEVEL 3 ECHO Kettő  
IF ERRORLEVEL 3 IF NOT ERRORLEVEL 4 ECHO Három
```

(Az első sor szavakkal kifejezve a következő:

Ha az ERRORLEVEL értéke 1, vagy 1-nél nagyobb és az ERRORLEVEL nem egyenlő 2-vel, vagy ennél nagyobb értékkel — tehát 1 az értéke —, akkor ECHO Egy.)

Gyakorlásképpen érdemes az előbbi MASOLAS parancsállományt ezzel a feltételrendszerrel módosítani, majd kipróbálni a futtatás után a Ctrl—C lenyomását.

**Megjegyzés:** Az *IF ERRORLEVEL* feltételt gyakran használják *GOTO* paranccsal együtt. Erről még a későbbiekben lesz szó.

### A NOT operátor

Az IF parancs használata során bizonyos esetekben pont az ellentétét kell vizsgálni annak, amit az adott IF feltétel megfogalmaz. Például az

```
IF NOT EXIST PROBA_1.TXT COPY A:PROBA_1.TXT *.*
```

parancs hatására a DOS csak akkor végzi el a PROBA\_1.TXT állomány másolását, ha az éppen aktuális könyvtárban nem szerepel. A NOT operátort mind a három IF változat esetében közvetlenül az IF után kell és lehet elhelyezni.

## Az IF parancs

Az IF EXIST parancs hatására a DOS megvizsgálja, hogy az előírt állomány létezik-e. Ha létezik, akkor a megadott parancsot végrehajtja, ellenkező esetben viszont a parancsállomány végrehajtását az IF után következő sortól folytatja.

Az IF ERRORLEVEL parancs arra utasítja a DOS-t, hogy az IF előtti sorban álló parancs által beállított ERRORLEVEL értékére vizsgálatot végezzen. Ha az ERRORLEVEL értéke nagyobb vagy egyenlő az IF ERRORLEVEL után megadott értékkel, akkor a DOS a megadott parancsot végrehajtja, ellenkező esetben viszont az IF-et követő sortól folytatja a parancsállomány végrehajtását.

Az IF karakterlanc1 == karakterlanc2 parancs a két karakterlanc azonoságától függően hajtja végre az IF végén szereplő DOS parancsot. A karakterlancok helyett az %1, ..., %9 változók is alkalmazhatók. A parancs megfelelő működéséhez a karakterlancokat és a %1, ..., %9 változókat is idézőjelek között kell szerepeltetni. Például:

```
IF '%1' == '' ECHO Nem adtad meg a paramétert!
```

## *Adott DOS parancs végrehajtása több állományra (A FOR parancs)*

A FOR parancs — amelynek segítségével a parancsok ismételt végrehajtása érhető el — általános alakja:

```
FOR %%változó IN (állományok_halmaza) DO parancs
```

ahol a változó egy karakter, amelyhez a DOS az állományok halmazából a ciklus minden egyes ismétlése során egy állománynevet rendel. Az állományok halmazát egy vagy több DOS állomány alkothatja. Az állományok megadása során helyettesítő karakterek használata is megengedett. Parancsként azt a DOS parancsot kell előírni, amelyet az egyes állományokra akarunk végrehajtani. Ezek után lássuk, hogyan működik a FOR parancs! Induljunk ki a következőkből:

```
FOR %%I IN (AA.TXT BB.TXT CC.TXT) DO PRINT %%I
```

A FOR parancs végrehajtásának első lépéseként az **I** változó az állományok halmazában elsőként szereplő állománynevet fogja tartalmazni. Tehát amikor a

```
PRINT %%I
```

végrehajtására sor kerül, akkor valójában a DOS a

```
PRINT AA.TXT
```

parancsot hajtja végre. A PRINT parancs elvégzése után a FOR megvizsgálja, hogy az állományok halmazában szerepel-e még valamelyik állomány neve. Mivel szerepel, ezért elvégzi az **I** értékadását, és így a

```
PRINT BB.TXT
```

parancs jön létre. A művelet elvégzése után az eljárás még egyszer megismétlődik, majd mivel az állományok halmaza nem tartalmaz már új nevet, a FOR parancs végrehajtása befejeződik. Az állományneveket nem szükséges előzetesen rögzíteni, mivel ide a parancssorban megadott paraméterek is átvehetők. Például:

```
C:\SAJAT>COPY CON PELDA_1.BAT <ENTER>  
@ECHO OFF <ENTER>  
FOR %%I IN (%1 %2) DO TYPE %%I <ENTER>  
^Z <ENTER>  
C:\SAJAT>
```

Az állományok körének előírásánál helyettesítő karakterek is alkalmazhatók. Erre is mutatunk egy példát az alábbiakban elkészíthető PELDA\_2.BAT-ban:

```
C:\SAJAT>COPY CON PELDA_2.BAT <ENTER>  
@ECHO OFF <ENTER>  
FOR %%I IN (*.BAT) DO TYPE %%I <ENTER>  
^Z <ENTER>  
C:\SAJAT>
```

A FOR parancs arra utasítja a DOS-t, hogy a parancs végén megadott utasítást az IN után előírt állományokra egyesével végrehajtsa. Például:

```
FOR %%I IN (*.TXT) DO PRINT %%I
```

## *Ugrás végrehajtása a parancsállományon belül (A GOTO parancs)*

Bár az IF paranccsal a feltételhez kötött parancsvégrehajtás megvalósítható, a FOR paranccsal pedig az előírt állományok körére a megadott parancs hajtható végre, előfordulhatnak olyan helyzetek, amikor egy adott feltétel teljesülése esetén a parancsállomány tartalmának végrehajtását egy adott, címkével azonosított helytől kezdve akarjuk folytatni. Ennek a műveletnek az eszköze a GOTO, amelynek általános alakja a következő:

```
GOTO CIMKE
```

```
·
·
·
:CIMKE
```

vagy

```
:CIMKE
```

```
·
·
·
GOTO CIMKE
```

A GOTO után megadott CIMKE tetszőleges név vagy azonosító lehet. A rendszer a parancsállomány feldolgozását a GOTO után szereplő CIMKE-vel jelölt sortól folytatja. Például a

```
GOTO CIMKE
DATE
TIME
:CIMKE
VER
```

esetben a DOS a DATE és a TIME parancsot nem hajtja végre. A GOTO sorból egyenesen a :CIMKE sorra ugrik, és a VER parancsot végzi el.

**Fontos megjegyzés:** Azt a sort, amelyben a címke szerepel, vagyis ahová az ugrást végre akarjuk hajtani, kettősponttal kell kezdeni. Címkeként tetszőleges hosszúságú név, vagy azonosító szerepelhet, a DOS azonban ebből csak az első nyolc karaktert veszi figyelembe.

A GOTO jól használható az ERRORLEVEL értékek kezeléséhez is. A következő példa ezt illusztrálja!

```
C:\SAJAT>COPY CON MASOLAS.BAT <ENTER>
@ECHO OFF <ENTER>
DISKCOPY A: B: <ENTER>
IF ERRORLEVEL 1 GOTO HIBA <ENTER>
GOTO RENDBEN <ENTER>
:HIBA <ENTER>
ECHO Hiba a másolási művelet során! <ENTER>
:RENDBEN <ENTER>
^Z <ENTER>
C:\SAJAT>
```

Ha tehát a hibaváltozó értéke 1 vagy ennél nagyobb, akkor a parancsállomány végrehajtása a HIBA címkétől, 0 érték esetén viszont a RENDBEN címkétől folytatódik.

A GOTO parancs hatására a DOS a parancsállomány végrehajtását attól a sortól folytatja, amely a GOTO után megadott címkét tartalmazza. A címkével megjelölt sort kettősponttal kell kezdeni, mert a DOS csak innen tudja, hogy címkével van dolga. Címkeként tetszőleges hosszúságú szó vagy változónév választható, a DOS azonban ebből csak az első nyolc karaktert veszi figyelembe.



## Az *AUTOEXEC.BAT* állomány minimális és szokásos tartalma

Mint a bevezetőben említettük, az *AUTOEXEC.BAT* állomány speciális parancsállomány, mivel:

1. A DOS a betöltése után a gyökérkönyvtárban automatikusan keresni kezdi.
2. A neve és a helye rögzített:
  - a gyökérkönyvtárban lehet csak elhelyezni,
  - az *AUTOEXEC.BAT* elnevezést kell viselnie.

Ezen okok miatt tehát az *AUTOEXEC.BAT* állományban a rendszerindítás után minden alkalommal végrehajtandó DOS parancsokat kell, és érdemes elhelyezni. Ezen szempontok figyelembevételével a következőkben az *AUTOEXEC.BAT* minimális és szokásos tartalmára mutatunk be egy-egy példát.

Az *AUTOEXEC.BAT* minimális tartalma:

```
PROMPT $P$G  
PATH C:\;C:\DOS;C:\SAJAT
```

**Megjegyzés:** a *PATH* parancs után természetesen minden olyan könyvtárat — elérési utat — érdemes előírni, amelyben a parancsok automatikus keresését el akarjuk érni.

Az *AUTOEXEC.BAT* szokásos tartalma:

```
@ECHO OFF  
DATE  
TIME  
PROMPT $P$G  
PATH C:\;C:\DOS;C:\SAJAT  
KBHUN  
REM Meghajtó- és könyvtár váltás  
D:  
CD D:\MUNKA  
REM Egérmeghajtó program. CSAK AKKOR, HA EZT A MŰVELETET  
NEM A REM CONFIG.SYS ÁLLOMÁNYBAN VÉGEZZÜK  
C:\MOUSE
```

DOSKEY  
MIRROR

REM A DATE és a TIME futtatására csak akkor van szükség,  
REM ha a rendszerünk erről automatikusan nem gondos-  
REM kodik. A magyar ékezetes billentyűzetet defi-  
REM niáló programot – KBHUN – csak abban az esetben  
REM kell futtatni, ha nem használjuk ki az 5.0-s DOS által  
REM kínált magyar nyelvű billentyűzetet. (Az ide vonat-  
REM kozó parancsokat a 7. Fejezetben tárgyaljuk.)  
REM A DOSKEY segédprogrammal létrehozott makrókat  
REM is az AUTOEXEC.BAT állományban érdemes elhelyezni!  
REM Az 5.0-s DOS verzió a korábbi verziókhöz képest új,  
REM és a memóriakezeléssel kapcsolatos AUTOEXEC.BAT-ban  
REM elhelyezendő parancsait a 8. Fejezetben tárgyaljuk.  
REM Itt adjuk meg az ennek megfelelő AUTOEXEC.BAT  
REM állományt is.

## Saját parancsállományok

Saját parancsállományokat mindig az adott feladatnak megfelelően érdemes és kell elkészíteni, így ezek javasolt vagy szokásos tartalmára nehéz konkrét példákat mutatni. A következőkben azonban megadunk két ügyes parancsállományt, amelynek az olvasó biztosan hasznát veheti. Gyakorlásképpen nézze alaposan végig a parancsállományok tartalmát, és még azok futtatása előtt próbálja értelmezni a sorokat!

*Az ENTER lenyomását a sorok végén már nem jelezzük!*

1. Állományok áthelyezése az egyik alkönyvtárból a másikba  
(Az 5.0-s DOS verzióban erre már külön parancs — MOVE — van!)

```

C:\>COPY CON MOZGAT.BAT
@ECHO OFF
CLS
IF '%1' == '' GOTO HIBA
IF '%2' == '' GOTO HIBA
XCOPY %1 %2 /V
ECHO Töröljem a(z) %1-t?
ECHO Ha a törlésre nincs szüksége, nyomja le a Ctrl-C
ECHO kombinációt!
PAUSE
DEL %1
GOTO VEGE
:HIBA
ECHO Az állomány(ok) áthelyezéséhez a MOZGAT.BAT pa-
ECHO rancsállományt a következőképpen kell futtatni:
ECHO MOZGAT állománynév célhely. Állománynévként
ECHO azt az állományt kell megadni, amit át akarunk he-
ECHO lyezni. Célhelyként meghajtó vagy meghajtó és al-
ECHO könyvtár írható elő.
:VEGE
^Z
    1 File(s) copied
C:\>

```

Ha a törlést a Ctrl—C lenyomásával letiltjuk, akkor a rendszertől a következő üzenetet kapjuk:

Terminate batch job (Y/N)?

Parancsállomány feldolgozásának befejezése (I/N)?

A kérdésre igen válasszal felelve az állomány törlése elkerülhető.

2. A módosított és újonnan létrehozott állományokról biztonsági másolat készítése (félíg automatikus módszer).

**Megjegyzés:** A parancsállomány a FRAMEWORK (FW) integrált programcsomag használatát tételezi fel és az archív attribútummal rendelkező állományokról készít biztonsági másolatot.

```
C:\>COPY CON MUNKA.BAT
@ECHO OFF
REM Magyar ékezetes billentyűzetet definiáló program
MAGYAR
FW
CLS
ECHO Kérem a biztonsági másolathoz felhasználandó lemezt
ECHO az A: meghajtóba helyezni!
PAUSE
CLS
XCOPY C:\FW3\ *.* A: /S /M /P
CLS
^Z
    1 File(s) copied
C:\>
```

**Megjegyzés:** Az XCOPY parancsot és a kapcsolóit a 2. Fejezetben részletesen tárgyaltuk.

Ha a céllemez a másolás során betelik, akkor cseréljük ki egy másik formázott lemezre a parancssorba pedig gépeljük be a következőket:

```
XCOPY C:\FW3\ *.* A: /S /M /P
```

A saját célra készített parancsállományokat vagy a kifejezetten erre a célra létesített alkönyvtárban — pl. C:\BATCH —, vagy pedig abban a könyvtárban érdemes elhelyezni, amelyben azt felhasználjuk.

## ***Ismétlés = Kulcs a sikerhez***

A parancsállomány olyan állomány, amely DOS és speciális, csak parancsállományokban használható parancsokat tartalmaz. A DOS parancsállományainak .BAT kiterjesztéssel kell rendelkezniük. Ha a felhasználó a DOS prompt után egy parancsállomány nevét beírja, akkor a rendszer azt a lemezen megkeresi és az abban szereplő parancsokat egymás után végrehajtja. (A keresést

abban az esetben, ha az aktuális lemezegység aktuális könyvtárában nem találja, a PATH parancsban felsorolt útvonalakon is elvégzi.) A parancsállományok között kiemelt szerepe van az AUTOEXEC.BAT állománynak, amelynek neve rögzített, és amelyet a DOS az elindítása után a rendszerlemez gyökérkönyvtárában automatikusan keres. Ebben az állományban tehát azokat a parancsokat kell és érdemes elhelyezni, amelyeket minden rendszerindítás után automatikusan futtatni akarunk.

A parancsállományok neve után paraméterek is megadhatók, amelyeket a DOS automatikusan az %1, ..., %9 változókhöz rendel hozzá, a parancsállományon belül tehát így kell és szabad ezekre hivatkozni. A %0 mindig a parancsállomány nevét tartalmazza. A paramétereket a DOS csak akkor tekinti különbözőnek, ha azokat a szóköz, tabulátor, vessző vagy pontosvessző karakterrel választjuk el egymástól.

A DOS-ban összesen nyolc olyan parancs van, amelyeket csak, ill. elsősorban parancsállományok elkészítéséhez szoktak használni. Ezekkel a parancsokkal már programokhoz hasonló parancsállományok készíthetők.

Elsőként a REM parancsot tárgyaltuk, amely segítségével a parancsállományokban megjegyzések és magyarázatok helyezhetők el. A DOS a parancsállomány végrehajtásakor a REM-mel kezdődő sorokat figyelmen kívül hagyja. Többsoros megjegyzéseket csak úgy lehet készíteni, hogy minden megjegyzést tartalmazó sor elején a REM parancsot ismételten előírjuk.

A PAUSE előírására a DOS a parancsállomány végrehajtását csak akkor folytatja, ha a felhasználó egy billentyűt lenyom. A PAUSE után megadható üzenet viszont csak akkor jelenik meg a képernyőn, ha a parancsállományban a PAUSE sort megelőzően az ECHO OFF parancs nem szerepel.

A CALL az egyik parancsállományból a másik parancsállomány meghívását teszi lehetővé.

Az IF-fel feltételhez kötött parancsvégrehajtás valósítható meg. Három alakja van:

— Az IF EXIST hatására a DOS megvizsgálja, hogy az előírt állomány létezik-e. Ha létezik, akkor a megadott parancsot végrehajtja, ellenkező esetben viszont a parancsállomány végrehajtását az IF után következő sortól folytatja.

— Az IF ERRORLEVEL arra utasítja a DOS-t, hogy az IF előtti sorban álló parancs által beállított ERRORLEVEL értékre vizsgálatot végezzen. Ha az ERRORLEVEL nagyobb vagy egyenlő, mint az IF ERRORLEVEL után megadott érték, akkor a DOS a megadott parancsot végrehajtja, ellenkező esetben viszont az IF-et követő sortól folytatja a parancsállomány végrehajtását.

— Az IF karakterlánc1 == karakterlánc2 parancs a két karakterlánc azonoságától függően hajtja végre az IF végén szereplő DOS parancsot. A karakter-

láncok helyett az %1, ..., %9 változók is alkalmazhatók. A parancs megfelelő működéséhez a karakterláncokat, valamint a %1, ..., %9 változókat is idézőjelek között kell szerepeltetni.

A FOR arra utasítja a DOS-t, hogy a parancs végén megadott utasítást az előírt változóértékekre egyesével végrehajtsa.

A GOTO hatására a DOS a parancsállomány végrehajtását attól a sortól folytatja, amely a GOTO után megadott címkét tartalmazza. A címkével megjelölt sor kettősponttal kell hogy kezdődjön, mert a DOS csak innen tudja, hogy címkét kell feldolgoznia. A címke hosszúsága tetszőleges, a DOS azonban csak az első nyolc karaktert veszi figyelembe.

A SHIFT parancs arra utasítja a DOS-t, hogy az egyes parancsállomány paramétereiket egy pozícióval balra léptesse. Ennek értelmében tehát a SHIFT hatására a %1-es paraméter azt fogja tartalmazni, amit korábban a %2-es paraméter tartalmazott, a %3-as azt, amit korábban a %2-es és így tovább. Ha a parancssorban kilencnél több paramétert adunk meg, akkor a SHIFT végrehajtására a 9-ediket követő paramétert a %9 fogja tartalmazni.

## *Mit jelent?*

**Karakterlánc (string).** Idézőjelek között elhelyezett alfanumerikus karakterek sorozata. A karakterláncba tartozó számokat is szöveggként kezeli a rendszer.

**Változó.** Olyan paraméter, amely a parancs vagy a program futtatásakor kap értéket.

# A CONFIG.SYS állomány adott felhasználói igényekhez igazítása

**Már tanultuk:** CONFIG.SYS állomány, memóriarezidens program

Ha a DOS a **CONFIG.SYS** állományt a rendszer elindításakor a gyökérkönyvtárban megtalálja, akkor annak bejegyzései szerint alakítja ki a környezetet a memóriában. Ezt a műveletet ezért konfigurálásnak is nevezik. Ha az állomány nem létezik, akkor a DOS ehhez a művelethez a saját alapértelmezés szerinti konfigurációs értékeit és feltételezett eszközkészletét használja fel. Ha a CONFIG.SYS állományt létrehozunk, vagy annak tartalmát módosítjuk, akkor a rendszert újból el kell indítanunk, mert annak tartalmát a DOS csak egyszer, a rendszerindításkor vizsgálja meg.

A **memóriarezidens programok** a betöltésük után a memóriában maradnak és így bármikor és gyorsan hozzáférhetők.

**Fontos megjegyzések:** *avagy még mielőtt bármihez is hozzákezdnenék!*

1. *Mivel a CONFIG.SYS állományban elhelyezett bejegyzéseket a DOS rendszerindításkor automatikusan felhasználja, hibás bejegyzés esetén könnyen előfordulhat, hogy egyáltalán nem tud a rendszer elindulni. Fogadjuk meg ezért a következő két tanácsot:*

— *Mielőtt a CONFIG.SYS állomány tartalmát módosítanánk, a már meglévő és jól működő állományról készítsünk lemezre másolatot!*

C:\>COPY CONFIG.SYS A:CONFIG.SAJ <ENTER>

— *A FORMAT parancs felhasználásával készítsük el a rendszer betöltésére alkalmas lemezt, amelyről — ha a rendszerünket a hibás CONFIG.SYS állomány megfagyasztotta — a DOS-t újból el tudjuk indítani:*

```
C:\>FORMAT A: /S <ENTER>
```

2. A rendszer a CONFIG.SYS állományt mindig az AUTOEXEC.BAT állomány előtt dolgozza fel. A CONFIG.SYS bejegyzéseinek végrehajtásakor a DOS tehát még nincs birtokában azoknak az ismereteknek — elérési útvonalaknak —, amelyeket a PATH paranccsal az AUTOEXEC.BAT állományban előírtunk. Ennek megfelelően tehát a bejegyzések elkészítésekor — ha azokban állománynevek is szerepelnek, és ezek az állományok nem az éppen aktuális meghajtó gyökérkönyvtárában vannak — mindig adjuk meg az állományokhoz vezető teljes útvonalat is.

## Az utolsó érvényes meghajtó betűjelének előírása a LASTDRIVE= bejegyzéssel

### 3.0

Az EGYSZERŰEN DOS c. könyvben már felsoroltuk a CONFIG.SYS állományban megadható bejegyzéseket, részletesen viszont csak a FILES= és BUFFERS= bejegyzéseket tárgyaltuk. Most elsőként a LASTDRIVE= bejegyzést ismertetjük, amely lehetővé teszi a DOS által támogatandó legutolsó logikai meghajtó betűjelének előírását. Mint tudjuk, a DOS az egyes fizikai meghajtókra az A:, B:, C: stb. karakterekkel enged hivatkozni. Ez az adott lemezegység betűjele, az ún. meghajtónév. A DOS alapértelmezés szerint csak az A:, B:, C:, D:, E: meghajtót támogatja.

**Megjegyzés:** 5.0-s DOS verzió használata esetén az alapértelmezés mindig a felhasznált utolsó logikai meghajtó betűjelét követő betű az angol abc szerint.

Az A és a B karakter mindig a hajlékonylemez meghajtók azonosítását szolgálják. A C, partíciónált merevlemez meghajtó esetében — l. FDISK parancs — a D meghajtónevek is foglaltak lehetnek. Ha ezenkívül még további logikai meghajtókat, vagy ún. virtuális lemezt — ezt a 8. Fejezetben tárgyaljuk — akarunk létrehozni, akkor ehhez már alapértelmezés szerint csak az E karaktert használhatjuk, szükségessé válhat tehát a LASTDRIVE= bejegyzéssel további lemezazonosítók használatba vételének jelzése.

A LASTDRIVE= (legutolsó meghajtó) bejegyzés általános alakja a következő:



LASTDRIVE=X

ahol az X helyett (a meghajtó betűjele) az A—Z karakterek bármelyike — az angol abc-t feltételezve — adható meg. A következő bejegyzés hatására pl. a DOS az A:, B:, C:, D:, E:, F: meghajtót azonosító karaktert fogja támogatni:

LASTDRIVE=F

A CONFIG. SYS állományban elhelyezendő LASTDRIVE= bejegyzés a legutolsó érvényes meghajtó betűjelének előírását teszi lehetővé. A bejegyzés általános alakja:

LASTDRIVE=X

ahol az X az A—Z karakterek bármelyike lehet.

## ***Eszközmeghajtó programok telepítése a DEVICE= bejegyzéssel***

A számítógépes környezetbe tartozó hardver eszközökkel — legyen az a billentyűzet, a nyomtató, a monitor, a lemez meghajtó, az egér stb. — a DOS csak akkor tud kapcsolatot teremteni, ha ezt speciális szoftverek, ún. eszközmeghajtó programok lehetővé teszik. A leggyakoribb hardver eszközök esetében, mint pl. a nyomtató vagy a billentyűzet, a DOS ezeket a speciális programokat automatikusan biztosítja és telepíti. Ha azonban a számítógépes környezetet további hardver eszközökkel bővítjük, pl. egeret vásárolunk, akkor az eszközhöz tartozó és vele együtt lemezen szállított eszközmeghajtó program telepítéséről már magunknak kell gondoskodnunk. Az eszközmeghajtó program telepítése tehát az a művelet, amikor a DOS számára ezt a szoftvert hozzáférhetővé tesszük. Másként fogalmazva: a DOS tudomására hozzuk az adott eszköz létezését, és megadjuk a használatának módját.

A CONFIG.SYS állomány DEVICE= bejegyzésének beállításával az eszközmeghajtó program telepítését végezhetjük el a rendszer indításakor. A bejegyzés általános alakja a következő:

```
DEVICE=[meghajtó_betűjele][elérési_út]filenév[.kit]
```

ahol a filenév.kit az eszközmeghajtó programot tartalmazó állomány neve.

**Megjegyzés:** az eszközmeghajtó állomány előtt opcionálisan megadható a hozzá vezető teljes útvonal is. Erre persze csak akkor van szükség, ha ez az állomány nem a gyökérkönyvtárban van.

Az eszközmeghajtó állományoknak rendszerint .SYS kiterjesztése van.

## Az egér telepítése

A mindennapi életben az egyik leggyakrabban előforduló feladat az egér telepítése, ezért ezt tárgyaljuk elsőként. Az egérrel együtt a lemezen kétféle egérmeghajtó programot szállítanak. Az egyik neve rendszerint MOUSE.COM vagy MOUSE.EXE (esetleg MSMOUSE.COM, MSMOUSE.EXE, ill. GMOUSE.COM, GMOUSE.EXE). Ebben az esetben nem is az állomány azonosító nevére hívjuk fel az Olvasó figyelmét, hanem a kiterjesztésre (.COM, .EXE). Ez ui. az a program, amelyet a parancssorból kell futtatni ahhoz, hogy az egeret használni lehessen. Mivel az egér telepítésének műveletét minden alkalommal el kell végezni, valahányszor a rendszert elindítjuk, az .EXE vagy .COM kiterjesztésű egérmeghajtó program legjobb helye az AUTOEXEC.BAT állományban van.

Az egér telepítésének másik módja az ugyancsak lemezen szállított MOUSE.SYS eszközmeghajtó állomány felhasználása. Ezt a meghajtóprogramot viszont már a CONFIG.SYS állományban a DEVICE= bejegyzéssel kell telepíteni.

```
DEVICE=[meghajtó_betűjele:][elérési_út]MOUSE.SYS
```

### Fontos megjegyzések:

1. Az egér telepítéséhez csak az egyik módszert kell és szabad használni, tehát vagy az .EXE .COM kiterjesztésű állományt futtatjuk, vagy a CONFIG.SYS állományt a fenti bejegyzéssel egészítjük ki, majd a rendszert újból indítjuk.
2. A CONFIG.SYS állomány DEVICE= bejegyzésével csak a .SYS kiterjesztésű egérmeghajtó állományt szabad előírni. Ha itt az .EXE kiterjesztésű állományt adjuk meg, akkor a rendszert teljesen megbénítjuk, és még a DOS telepítését is újból kell kezdenünk!!!

## A leggyakoribb eszközmeghajtó állományok

A DOS számos eszközmeghajtó állománnyal rendelkezik. Első lépésként kérdezzük le, hogy a saját rendszerünk ezek közül melyeket tartalmazza, tehát ha a DOS állományok a C: meghajtóban levő \DOS alkönyvtárban vannak, akkor írjuk be a következő parancsot:

```
C:\>DIR \DOS\*.SYS <ENTER>
```

Ekkor a következőhöz hasonló lista jelenik meg a képernyőnkön:

```
Volume in drive C has no label
Volume serial number is 1A35-5F1B
Directory of C:\DOS
ANSI             SYS             9029      04-09-91   12:00
COUNTRY         SYS             17069     04-09-91   12:00
DISPLAY         SYS             15792     04-09-91   12:00
DRIVER          SYS             5409      04-09-91   12:00
EGA             SYS             4885      04-09-91   12:00
HIMEM           SYS             11552     04-09-91   12:00
KEYBOARD        SYS             34697     04-09-91   12:00
PRINTER         SYS             18804     04-09-91   12:00
RAMDRIVE        SYS             5873      04-09-91   12:00
SMARTDRV        SYS             8335      04-09-91   12:00
                10 file(s)
                11552 bytes
                40249344 bytes free
```

**Fontos megjegyzés!** Bár a *COUNTRY.SYS* és a *KEYBOARD.SYS* állomány *.SYS* kiterjesztésűek, ezért a fenti listában megjelennek, azonban *NEM* eszközmeghajtó programok. Nema *DEVICE=* bejegyzéssel használандók. Ha ezt a hibát mégis elkövetjük, akkor a DOS nem indul el! A *COUNTRY.SYS* állomány használata a *COUNTRY=* bejegyzéshez, a *KEYBOARD.SYS* állomány használata pedig a *KEYB* parancshoz kapcsolódik. Mivel ezt a bejegyzést, ill. parancsot a 7. Fejezetben tárgyaljuk, ezért ezen *.SYS* állományok ismertetésére is ott kerül majd sor

A következőkben a *DEVICE=* bejegyzéssel használható fenti eszközmeghajtó állományok célját és jellemzőit foglaljuk össze.

## ANSI.SYS

**2.0**

Az ANSI.SYS felhasználásával a képernyő és a billentyűzet sokkal hatékonyabban programozható. Azok az alkalmazói és felhasználói programok, amelyek az ANSI.SYS szolgáltatásait igénybe veszik, a telepítésük során vagy automatikusan elhelyezik a

```
DEVICE=[meghajtó_betűjele:][elérési_út]ANSI.SYS
```

bejegyzést a CONFIG.SYS állományban, vagy ezen művelet elvégzésére felszólítják a felhasználót.

## DISPLAY.SYS

**3.3**

Ha a számítógépet arra akarjuk rávenni, hogy ne csak az angol, hanem egyidejűleg más nyelv karakterkészletének használatát is lehetővé tegye, akkor a DISPLAY.SYS felhasználásával az ún. kódlapváltást — másképpen fogalmazva: kódlapok közötti átkapcsolást — kell előírnunk. A kódlap definiálja ui. az adott nyelv vagy nyelvterület esetében használt karakterkészletet. A bejegyzés általános alakja a következő:

```
DEVICE=[meghajtó_betűjele:][elérési_út]DISPLAY.SYS  
CON[:]=(típus[, [hwcp] [, (n,m) ]])
```

A típus lehet: MONO, CGA, LCD, EGA

A hwcp, vagyis a hardver kódlap lehet 437, 850, 852, 860, 863, 865

A kódok jelentése a következő:

- 437 USA kódlap
- 850 többnyelvű kódlap
- 852 szláv (csak 5.0-s DOS esetén) kódlap
- 860 Portugál kódlap
- 863 Francia kódlap
- 865 Norvég kódlap

Az n a kiegészítő kódlapok számát jelenti. Értéke hardvertől függően 0—12 lehet.

Az *m* az oldalanként biztosított segéd-karakterkészletek számát jelenti. (Ez EGA esetén 2, LCD esetén 1 lehet.)

**Fontos megjegyzés:** *Ez az eddig tárgyalt első olyan bejegyzés, amelynél egymástól vesszőkkel elválasztva többféle adatot (típust, kódokat) lehet előírni. Előfordulhatnak azonban olyan esetek, amikor a kódok közül nem kell mind-egyiket megadnunk. Ekkor a meg nem adott kód vagy kódok helyett csak a vesszőt kell szerepeltetni. Például:*

```
DEVICE=DISPLAY.SYS CON:=(EGA,,1)
```

*Ezt a bejegyzést akkor használják, ha a számítógépes környezet EGA monitort tartalmaz, és az angol mellett pl. a magyar ékezetes karakterkészlet használatát is lehetővé akarják tenni.*

*A fenti bejegyzés szokásos alakja még:*

```
DEVICE=DISPLAY.SYS CON:=(EGA,437)
```

*A most tárgyalt bejegyzésre a 7. Fejezetben még visszatérünk.*

## DRIVER.SYS

### 3.2

Ha a rendszerünkhöz a számítógép dobozán kívül elhelyezkedő ún. külső lemezmeghajtó egység is tartozik, akkor a DOS-nak ezen egység kezeléséhez a DRIVER.SYS meghajtóprogramra van szüksége. A bejegyzés általános alakja a következő:

```
DEVICE=[meghajtó_betűjele:][elérési_út]DRIVER.SYS/  
D:ddd[/T:ttt][/S:ss][/H:hh][/C][/F:f]
```

amelyben a kapcsolók jelentése:

*/D:ddd* a fizikai meghajtószám. Értéke 0—255 lehet. Az A a 0, a B az 1, a harmadik fizikai meghajtó, amelynek külsőnek kell lennie 3, stb. Az első merev-lemez meghajtóhoz — C — a 128-as érték tartozik!

*/T:ttt* az oldalankénti sávok száma (alapértelmezés: 80)

*/S:ss* a sávonkénti szektorok száma (alapértelmezés: 9)

/H:hh az oldalankénti fejek száma (alapértelmezés: 2)

/C jelzi a DOS számára, hogy az adott meghajtó a reteszének kinyitásáról, ill. becsukásáról tud jelzést küldeni.

/F:f az eszköz típusának megadása. Alapértelmezés: 2. Az f lehetséges értékei:

- 0 160Kb/180Kb-os vagy 320Kb/360Kb-os meghajtó
- 1 1.2 Mb-os meghajtó
- 2 720 Kb-os, 3 1/2 hüvelykes meghajtó
- 3 8 hüvelykes, egyszeres sűrűségű meghajtó
- 4 8 hüvelykes, duplasűrűségű meghajtó
- 5 Merevlemez meghajtó
- 6 Mágnesszalagos adatmentő/visszatöltő (sztrímer)
- 7 1.44 Mb-os hajlékonylemez meghajtó vagy optikai egység (5.0-s DOS!)
- 9 2.88 Mb-os hajlékonylemez meghajtó (5.0-s DOS!)

Ha külső meghajtóegységet vásárolunk, a dokumentációban a működtetéshez szükséges DEVICE=DRIVER.SYS bejegyzésnek szerepelnie kell!

## PRINTER.SYS

### 3.3

Azoknál a nyomtatóknál, amelyek a kódlapváltást támogatják, a megfelelő PRINTER.SYS bejegyzést el kell helyezni a CONFIG.SYS állományban. A bejegyzés általános alakja a következő:

```
DEVICE=[meghajtó_betűjele:][elérési_út]PRINTER.SYS
LPTn=(típus[, [hwcp] [,n]])
```

Az LPTn esetében az n helyébe 1, 2 vagy 3 írható attól függően, hogy melyik fizikai nyomtatóporthoz csatlakozik párhuzamosan a nyomtató.

**Megjegyzés:** Az LPT ún. eszköznév, amelyet a 0. Fejezetben részletesen tárgyaltunk.

A típus, amely a nyomtató típusát megadó kódot jelenti, a következő lehet:

4201	IBM 4201 Proprinter, IBM 4202 Proprinter XL
4208	IBM Proprinter 4207 vagy 4208
5202	IBM QuietWriter III Model 5202

A hwcp a hardver kódlap azonosító számát jelenti. Értéke 437, 850, 852, 860, 863, 865 lehet.

Az n a további előkészített kódlapok számát jelenti. Értéke 0—12 lehet.

A nyomtatóhoz tartozó dokumentációnak a nyomtató megfelelő működtetéséhez szükséges PRINTER.SYS bejegyzést tartalmaznia kell! Példaként az LPT1-hez csatlakoztatott és francia kódlapot használó Proprinter működtetéséhez szükséges DEVICE=PRINTER.SYS bejegyzést adjuk meg:

```
DEVICE=PRINTER.SYS LPT1:=(4201,863,1)
```

A 7. Fejezetben a most tárgyalt bejegyzésre még visszatérünk.

## HIMEM.SYS

### 4.0

A 8. Fejezetben részletesen tárgyalt kibővített (extended) memória használatát biztosító meghajtóprogram.

## RAMDRIVE.SYS

### 5.0

(A 3.0-tól 4.0-ig ugyanezt a funkciót a VDISK.SYS meghajtóprogram látja el.) A 8. Fejezetben részletesen tárgyalt virtuális lemez vagy RAM meghajtó létrehozását biztosító meghajtóprogram.

## SMARTDRV.SYS

### 5.0

A 8. Fejezetben részletesen tárgyalt cache memória vagy gyorsítótár létesítését biztosító meghajtóprogram.

## EGA.SYS

### 5.0

Ha EGA monitort használunk, és egyszerre több programot akarunk futtatni, és ezek között könnyen át akarunk kapcsolni, akkor az EGA.SYS meghajtóprogram telepítéséről is gondoskodnunk kell.

**Megjegyzés:** *Több program egyidejű használatát az 5.0-s DOS verzióban a Shell Options menüjének Enable Task Swapper opciója teszi lehetővé.*

Bár nem .SYS kiterjesztésű, azonban a meghajtóprogramok közé tartozik és a DEVICE= bejegyzéssel kell telepíteni az 5.0-s DOS használata esetén rendelkezésre álló és a kiterjesztett memóriát kibővített memória esetén szimuláló EMM386.EXE programot is, amelynek használatát a 8. Fejezetben tárgyaljuk.

## A DEVICE= bejegyzés

A DOS minden eszközzel — pl. billentyűzet, monitor — eszközmeghajtó programok segítségével veszi fel a kapcsolatot. A DOS a billentyűzet, monitor, merev- és hajlékonylemez meghajtók, valamint portok esetében beépített eszközmeghajtókkal rendelkezik, azonban a további eszközök — pl. külső meghajtó — telepítéséhez külső, azaz lemezzel betöltött eszközmeghajtó programot kell igénybe vennie. Ennek eszköze a CONFIG.SYS állományban elhelyezendő DEVICE= bejegyzés, amely után megadott eszközmeghajtó programot a DOS betölti a memóriába, és az adott eszköz vezérléséhez felhasználja.

## Megjegyzések elhelyezése a CONFIG.SYS állományban

### 4.0

Az AUTOEXEC.BAT állomány tárgyalása során már megismerkedhetett az Olvasó a REM paranccsal, amely megjegyzések elhelyezését teszi lehetővé a parancsállományban. A 4.0-s DOS verziótól kezdődően ugyanez a lehetőség



már a CONFIG.SYS állományban is rendelkezésre áll. A megjegyzések különösen azoknál a bejegyzéseknél hasznosak, amelyek „létfonosságú” vagy speciális célt szolgálnak.

A REM parancsnak még egy fontos felhasználási területe van. Ha valamilyik bejegyzésre átmenetileg nincs szükségünk, akkor az előtt a REM parancsot elhelyezve a bejegyzés letiltható. Például:

```
REM DEVICE=ANSI.SYS
```

### Fontos megjegyzés!

*A REM után a szöveget nem szabad azonnal elkezdni. Ne feledkezzünk meg a parancsszó utáni szóköz begépeléséről! Ellenkező esetben ui. a DOS a REM parancsot nem ismeri fel, és erre a következő üzenettel hívja fel a figyelmet:*

```
Unrecognized command in CONFIG.SYS
Error in CONFIG.SYS line...
```

```
Beazonosíthatatlan parancs a CONFIG.SYS-ben.
Hiba a CONFIG.SYS ... sorában.
```

## ***Az INSTALL= bejegyzés használata memóriarezidens programok betöltéséhez***

### **4.0**

A DOS már alapértelmezés szerint is több programot a memóriában tart — pl. a belső DOS parancsokat is tartalmazó COMMAND.COM állományt —, hogy azok használatát minél gyorsabbá és könnyebben kezelhetővé tegye.

A CONFIG.SYS állományban elhelyezendő INSTALL= bejegyzés segítségével a memóriában tartott állományok köre a következő négy programmal bővíthető tovább:

FASTOPEN.EXE	3.3
KEYB.COM	3.3
NLSFUNC.EXE	3.3
SHARE.EXE	3.0

Milyen feladat végezhető el ezekkel a programokkal?

### **FASTOPEN.EXE**

A DOS ahhoz, hogy az adott programot a memóriába be tudja tölteni és futtatni tudja, előbb meg kell keresnie a lemezen. Mivel azonban a lemez — akár hajlékony, akár merev — fizikai eszköz, az ezzel kapcsolatos írási és olvasási műveleteket az egyéb műveletekhez képest csak több nagyságrenddel lassabban tudja a számítógép elvégezni. A FASTOPEN arra utasítja a DOS-t, hogy a már megnyitott állományok helyére vonatkozó adatokat a memóriában, egy erre a célra létesített külön pufferben tárolja. A program alkalmazásával tehát csökkenthető a már megnyitott állományok ismételt betöltéséhez szükséges idő. A program használatát a 8. Fejezetben részletesen tárgyaljuk.

### **KEYB.COM**

Annak érdekében, hogy a DOS használata során ne csak az angol karakterkészletet és billentyűzet elrendezést lehessen használni, a rendszer lehetővé teszi különböző billentyűsablonok alkalmazását. A KEYB parancs ezen sablonok memóriába töltését teszi lehetővé. A KEYB.COM program használatát és a magyar nyelvű billentyűzet beállításához szükséges egyéb parancsokat — 5.0-s DOS esetén — a 7. Fejezetben tárgyaljuk.

### **NLSFUNC.EXE**

Ha a kódlapváltást a CHCP paranccsal akarjuk végrehajtani, akkor ehhez a DOS-nak előzetesen az NLSFUNC parancs futtatására van szüksége.

### **SHARE.EXE**

Hálózatba kapcsolt számítógépek esetén a SHARE paranccsal lehetőség adható arra, hogy ugyanahhoz az állományhoz egy időben több felhasználó férheszen hozzá.

Az INSTALL= bejegyzés általános alakja a következő:

```
INSTALL=[meghajtó_betűjele:] [elérési_út] filenév.kit [paraméterek]
```

A `filenév.kit` helyett a fentiekben megadott négy állománynév közül a megfelelőt, paraméterként pedig a megadott állományhoz tartozó paramétert kell előírni. Például ha minden egyes rendszerindításkor el akarjuk érni, hogy a DOS a C: meghajtón az 50 legutoljára használt katalógusbejegyzést nyilvántartsa, akkor a következő sort helyezük el a `CONFIG.SYS` állományban:

```
INSTALL=FASTOPEN.EXE C:=50
```

A bejegyzés feltételezi, hogy a `FASTOPEN.EXE` állomány a rendszerlemez gyökérkönyvtárában van.

Az `INSTALL=` bejegyzést a `CONFIG.SYS` állományba helyezük el a DOS működésének hatékonyabbá tételére. Ez további memóriarezidens programok telepítését is lehetővé teszi.

## *A SHELL= bejegyzés használata a parancsfeldolgozó állomány nevének és helyének megadásához*

### 2.0

Ahhoz, hogy a DOS rendszer működőképes és az adott felhasználói környezethez igazított legyen, a következő három állománynak kell a merevlemez gyökérkönyvtárában lennie a két rejtett állományon (`IO.SYS`, `MSDOS.SYS`) kívül:

```
COMMAND.COM  
CONFIG.SYS  
AUTOEXEC.BAT
```

Az `AUTOEXEC.BAT` és a `CONFIG.SYS` állományokról eddig már többször és részletesen beszéltünk. Ismerkedjünk meg egy kicsit részletesebben a `COMMAND.COM` állománnyal is!

A parancsfeldolgozónak vagy parancsprocesszornak is nevezett COMMAND.COM állomány felelős a prompt megjelenítéséért, a begépett parancsok feldolgozásáért, valamint ez az állomány tartalmazza a DOS belső parancsait (pl. DATE, TIME, COPY stb.). A DOS működéséhez tehát három nélkülözhetetlen állomány szükséges, az IO.SYS és MSDOS.SYS rejtett állományok és a COMMAND.COM. Ha a lemez formázásakor az /S kapcsolót is megadjuk, akkor a DOS ezeket az állományokat automatikusan a formázandó lemezre másolja, vagyis a rendszer betöltésére alkalmas lemezt készít. Ha a DOS betöltését olyan lemezzől kíséreljük meg, amely ezeket az állományokat nem tartalmazza, akkor a következő hibaüzenetet kapjuk:

```
Non-System disk or disk error  
Replace and press any key when ready
```

```
Nem rendszerlemez vagy lemezhiba.  
Cserélje ki, és ha készen van, nyomja le valamelyik  
gombot!
```

Mivel a rendszer ezeket az állományokat csak és kizárólag a gyökérkönyvtárban keresi, ezeket itt kell elhelyezni!

A merevlemez rendszerint úgy szervezik meg, hogy a külső DOS parancsokat, valamint a COMMAND.COM állományt is a DOS vagy ehhez hasonló nevű alkönyvtárban helyezik el. Viszont az előbbieken leírtak értelmében a COMMAND.COM állomány másolatát a gyökérkönyvtárban is el kell helyezni. Ennek következtében tehát a COMMAND.COM rendszerint két példányban szerepel a merevlemezen. A SHELL= bejegyzés felhasználásával a DOS arra utasítható, hogy a COMMAND.COM állományt ne a gyökérkönyvtárban, hanem a megadott alkönyvtárban keresse. Például ha a COMMAND.COM állomány közvetlenül a gyökér alatt elhelyezkedő DOS alkönyvtárban van, akkor a CONFIG.SYS állományban a következő SHELL= bejegyzést érdemes szerepeltetni:

```
SHELL=C:\DOS\COMMAND.COM C:\DOS /P
```

A COMMAND.COM után szereplő C:\DOS a parancsállomány helyét adja meg, a /P kapcsoló előírására pedig azért van szükség, mert enélkül a DOS nem futtatja automatikusan a gyökérkönyvtárban elhelyezkedő AUTOEXEC.BAT állományt.

A 3.2-es vagy magasabb DOS verzió esetén a SHELL= bejegyzés még arra is felhasználható, hogy a DOS környezet számára fenntartott hely méretét előírjuk vele. Erre azért lehet szükség, mert ha a rendszerünkben memóriarezi-dens szoftvereket helyezünk el, akkor a DOS környezet szükség esetén nem tud tovább terjeszkedni. A megoldást a /E kapcsoló használata adja, amellyel a DOS környezet számára biztosítandó hely foglalható le. (Lehetséges érték: 160-tól — ez az alapértelmezés — 32768 bájt.) Például a

```
SHELL=C:\DOS\COMMAND.COM C:\DOS /E:4096 /P
```

bejegyzés azon kívül, hogy megadja a parancsfeldolgozó helyét — C:\DOS — a környezet számára 4096 bájtot foglal le.

A SHELL= bejegyzéssel a DOS által használandó parancsfeldolgozó neve és helye adható meg. A DOS alapértelmezésként a COMMAND.COM állományt használja parancsértelmezőként, a SHELL= bejegyzéssel azonban — szükség esetén — más parancsértelmező is megadható. Ha a COMMAND.COM parancsértelmező a számunkra megfelel, akkor a SHELL= bejegyzés alkalmazására csak a következő két esetben van szükségünk:

1. A COMMAND.COM állomány nem a gyökérkönyvtárban van. Például:

```
SHELL=C:\DOS\COMMAND.COM C:\DOS /P
```

A /P kapcsoló azért kell, hogy a DOS a gyökérkönyvtárban elhelyezett AUTOEXEC.BAT állományt is futtassa.

2. Az alapértelmezés szerint biztosított hely nem elegendő a DOS környezet számára. Például:

```
SHELL=COMMAND.COM /E:1024 /P
```

## ***A STACKS= bejegyzés használata a hardver megszakítások fogadásának kezelésére***

Annak érdekében, hogy a számítógép gyorsabban és hatékonyabban működhessen, a hardver eszközök (pl. meghajtó) közül a legtöbb bizonyos feladatokat önállóan végez el, és ha a feladattal végzett, akkor megszakítja a DOS működését, és közli vele, hogy a feladatot befejezte. Például ha a DOS arra utasítja az egyik meghajtót, hogy az abban elhelyezett lemezre írási műveletet végezzen, akkor a DOS maga továbbra is folytatja a feldolgozási és felügyeleti feladatok végzését, mialatt a meghajtó a lemezre rögzíti az információt. Amikor azonban a meghajtó a feladatot elvégezte, akkor megszakítja a DOS működését, és ezt közli vele.

Ha a DOS-hoz egy ilyen megszakítás érkezik, akkor ideiglenesen felfüggeszti az éppen elvégzett feladatot, és a folytatáshoz szükséges adatokat a RAM memória egy speciális területén, az ún. veremben (*stack*) tárolja. Bizonyos esetekben előfordulhat, hogy a DOS-hoz egyszerre több megszakítás érkezik, mint amennyit az kezelni tud. Ekkor további veremekben további adminisztrációs adatok tárolását kell elvégeznie. Ha azonban a vermek alapértelmezés szerinti száma és/vagy mérete ezen átmeneti adattárolási műveletek elvégzését nem teszi lehetővé, akkor a DOS a következő üzenetet írja a képernyőre:

```
Fatal: Internal Stack Failure, System Halted
```

```
Végzetes hiba: Belső veremhiba, a rendszer megállítva
```

Ha tehát ilyen üzenetet kapunk, akkor a CONFIG.SYS állományban a veremek számát és méretét növelnünk kell. A bejegyzés általános alakja a következő:

```
STACKS=n, s
```

ahol az *n* vermek számát (megadható érték: 0, valamint 8—64), az *s* pedig a vermek méretét (megadható érték: 0, valamint 32—512) jelenti.

Alapértelmezés:

PC XT esetén: 0,0

A többinél: 9,128

Mivel ez a bejegyzés a RAM memóriából foglal el helyet, ezen értékeket a helytakarékosság érdekében 0-ra szokták beállítani:

```
STACKS=0,0
```

A STACKS= bejegyzés arra utasítja a DOS-t, hogy a hardver megszakítások kezeléséhez több memóriát tartson fenn. Erre a bejegyzésre a következő két esetben van szükség:

1. PC AT gépeknél az alapértelmezés szerinti érték — 9,128 — 0,0-ra állítása helytakarékossági szempontokból.
2. Az alapértelmezés szerinti méret növelése a következő hibüzenet esetén:

```
Fatal: Internal Stack Failure, System Halted
```

## ***Ismétlés = Kulcs a sikerhez***

A LASTDRIVE= bejegyzés a legutolsó érvényes meghajtó betűjelének előírását teszi lehetővé. A bejegyzés általános alakja:

```
LASTDRIVE=X
```

ahol az X az A—Z karakterek bármelyike lehet.

A DOS minden eszközzel — pl. billentyűzet, monitor — eszközmeghajtó programok felhasználásával veszi fel a kapcsolatot. A DOS a billentyűzet, monitor, merev- és hajlékonylemez meghajtók, valamint portok esetében beépített eszközmeghajtókkal rendelkezik, a további eszközök azonban — pl. külső meghajtó — telepítéséhez külső, azaz lemezzel betöltött eszközmeghajtó programot kell igénybe vennie. A telepítés eszköze a DEVICE= bejegyzés, amely után megadott eszközmeghajtó programot a DOS a memóriába betölti, és az adott eszköz vezérléséhez felhasználja.

A REM parancs megjegyzések elhelyezését, valamint az adott helyzetben szükségtelen bejegyzések végrehajtásának letiltását szolgálja.

Az INSTALL= bejegyzés a DOS működésének hatékonyabbá tételét szolgálja, mivel további memóriarezidens programok telepítését teszi lehetővé.

A SHELL= bejegyzéssel az operációs rendszer által használandó parancsfeldolgozó neve és helye adható meg. A DOS alapértelmezésként a COMMAND.COM állományt használja parancsértelmezőként. Ha tehát ez a parancsértelmező a számunkra megfelel, akkor a SHELL= bejegyzés alkalmazására csak a következő két esetben lehet szükség:

1. A COMMAND.COM állomány helyének előírása, ha az nem a gyökérkönyvtárban van.
2. A DOS környezet számára alapértelmezés szerint biztosított hely növelése.

A STACKS= bejegyzés arra utasítja a DOS-t, hogy a hardver megszakítások kezeléséhez több memóriát tartson fenn. Erre a bejegyzésre a következő két esetben van szükség:

1. PC AT gépeknél az alapértelmezés szerinti érték — 9,128 — 0,0-ra állítása memóriatakarékosági szempontokból.
2. Az alapértelmezés szerinti méret növelése a következő hibaüzenet esetén:

```
Fatal: Internal Stack Failure, System Halted
```

A fejezetben tárgyalt valamennyi bejegyzés ún. konfigurációs parancs, amelyeket csak és kizárólag a CONFIG.SYS állományban elhelyezve lehet végrehajtani.

## *Mit jelent?*

**Kódlap (code page).** A kódlapok hasonló módon működnek a számítógépben, mint az írógépen az írógépfej. Ha pl. a gépíró cirill betűkkel akar gépelni, akkor kicseréli az írógépfejet. Az ennek megfelelő művelet a számítógépben a kódlapváltás.

**Megszakítás (interrupt).** Programvezérlés ideiglenes átadása hardver eszköztől származó jel vagy programutasítás által. A megszakítási program elvégzése után a megszakított program futtatása folytatódik.

**Verem (stack).** A RAM memóriának az a speciálisan kezelt területe, amely a számítógép működése során értékek ideiglenes tárolását szolgálja.



# *Lehetőségek a nemzeti sajátosságok figyelembevételére 5.0-s DOS használata esetén*

## *Bevezetés*

**Fontos megjegyzés!** A fejezetben tárgyalt parancsokat és utasításokat csak akkor alkalmazzuk, ill. a CONFIG.SYS és az AUTOEXEC.BAT tartalmát csak akkor módosítjuk, ha a fejezetet már alaposan végigolvastuk. Az itt tárgyalt parancsoknak és bejegyzéseknek sem a sorrendje, sem a paraméterezése nem választható tetszőlegesen. Az utolsó alfejezetben konkrét helyzetekhez konkrét példákat is megadunk, amelyeket az Olvasó könnyen a saját igényeihez igazíthat, ha már minden bejegyzésnek és paraméternek pontosan tudja az okát és a célját. Ehhez viszont a fejezetben leírtak alapos figyelmes átolvasása szükséges.

Az IBM számítógépek és ezek operációs rendszere, a DOS széles körű elterjedésével igény jelentkezett arra, hogy ezek a gépek ne csak az angolszász országokban szokásos dátum- és idő kijelzési módokat, pénznemet, gépeléskor csak az angol abc betűit, valamint az angolszász országokban használatos írógép-billentyűzet elrendezést, hanem más országok — pl. a német vagy a francia, 5.0-s DOS esetében pedig a magyar! —, szokásait, ill. speciális karaktereit is engedjék használni.

A továbbiakban az **5.0-s DOS** kínálta lehetőségeket tárgyaljuk, azonban ahol lehet, megemlítjük azokat a lehetőségeket is, amelyek az 5.0-nál alacsonyabb DOS verzió esetén állnak a felhasználók rendelkezésére.

A nemzeti sajátosságok figyelembevételére a következő beállítások végezhetők el:

— A dátum, idő és a pénznem kijelzésének módja, a karakterek abc-be rendezéséhez felhasznált sorrend és az állományneveknél alkalmazható karakterkészlet kialakítása az adott ország igényeinek megfelelően.

— Az éppen aktuális karakterkészlet és annak billentyűzeten való elrendezése az adott ország szokásainak megfelelően.

— Karakterkészlet — (kódlap) — váltás annak érdekében, hogy olyan karakterek is gépelhetőek, megjeleníthetőek, ill. kinyomtathatók legyenek, amelyeket az alapértelmezés szerinti kódlap nem tartalmaz.

A DOS, amikor szöveggel dolgozik, maximálisan 256 különböző karaktert tud használni. Az adott időpontban alkalmazott adott karakterkészletet kódlapnak nevezzük.

Kódlappal valamennyi számítógép rendelkezik. Ez az ún. hardver kódlap. A DOS tehát — ha más nem áll a rendelkezésre — magától értetődően ezt tekinti aktívnak, és ezt használja. Ha olyan nyelven akarunk gépelni, amelynek karaktereit a hardver kódlap tartalmazza, akkor „mindössze” annyi a teendők, hogy a COUNTRY= bejegyzés megfelelő beállításával, valamint a megfelelően paraméterezett KEYB parancs végrehajtásával a rendszert átállítsuk az adott nyelvre.

De ha pl. a speciális magyar karakterekre van szükségünk, vagyis az é, á, ó, ő, ú, ö, ü, í, ű, karakterek kis- és nagybetűs alakjára, akkor pl. a 437-es kódlap már nem felel meg, mivel ez a fenti karakterek közül többet nem tartalmaz. Ilyen esetben szükséges a kódlapváltás, vagyis a karakterkészlet cseréje. Ehhez az 5.0-s DOS a **B Függelékben** megadott kódlapokat biztosítja. Ezek az ún. előkészített kódlapok, amelyeket a DOS lemezen .CPI kiterjesztésű állományok tartalmaznak.

Az 5.0-s DOS összesen 24 ország sajátosságait tudja figyelembe venni, ehhez azonban mindig az adott országhoz tartozó, és a **C Függelék** táblázatában összefoglalt ország, billentyűzet, alapértelmezés szerinti vagy alternatív kódot kell alkalmazni.

Magyarországhoz a táblázat szerint a következő kódok tartoznak:

Országkód: 036

Billentyűkód: HU

Alapértelmezés szerinti kódlap: 852

Alternatív kódlap: 850

Országkódként rendszerint az adott ország telefon-előhívó száma szerepel, így Magyarországnál a 036.

A táblázatban található alapértelmezés szerinti, ill. alternatív kódlapok a COUNTRY= bejegyzés, ill. a KEYB parancs használatához kötődnek, így ezekre a fogalmakra a megfelelő helyen még visszatérünk. Arra azonban már most felhívjuk a figyelmet, hogy adott országhoz mindig a táblázatban — l. **C Függelék** — megadott kódlapokat kell alkalmazni. Ha ui. a DOS-t adott ország esetében a táblázattól eltérő kód használatára próbáljuk kényszeríteni, akkor a következő üzenetet kapjuk:

```
Invalid country code or code page
Error in CONFIG.SYS line...
```

```
Érvénytelen országcód vagy kódlap.
Hiba a CONFIG.SYS ... sorában.
```

## *A dátum- és idő kijelzés formátumának beállítása a COUNTRY= bejegyzéssel*

### 3.0

A COUNTRY= bejegyzéssel és a COUNTRY.SYS programmal a dátum- és idő kijelzés formátuma, a pénznem szimbóluma, az állományok abc-be rendezésekor a sorrend, valamint az állomány- és könyvtárnevekben alkalmazható karakterek köre definiálható. CONFIG.SYS állományban elhelyezendő bejegyzés általános alakja:

```
COUNTRY=xxx[, [yyy] [, [meghajtó_betűjele:] [elérési_út]filenév.kit]]
```

Az xxx a **C Függelék** táblázata szerint az adott országhoz tartozó országkód.

Az yyy a **C Függelék** táblázata szerint az adott országhoz megadott kódlapok egyikének azonosító száma.

A filenév.kit annak az állománynak a neve, amely az adott országra jellemző formátumot tartalmazza. Ez az állomány rendszerint a COUNTRY.SYS.

A COUNTRY= bejegyzés Magyarország esetében — ha a COUNTRY.SYS állomány a C: meghajtó gyökérkatalógusában van — pl. a következő lehet:

```
COUNTRY=036
```

amely a következő dátum- és idő kijelzési formátumot állítja elő:

Dátum	Idő
1993-04-15	10:25:00,00

Ha azonban a COUNTRY.SYS állomány a C:\DOS alkönyvtárban van, akkor az állományhoz vezető útvonal előírásáról sem szabad megfeledkezünk, vagyis:

```
COUNTRY=036, , C:\DOS\COUNTRY.SYS
```

Mivel az országkód után nem adtuk meg a kódlap számát, ezért a DOS úgy értelmezi, hogy azt az adott országkódhoz tartozó — **1. C Függelék** — alapértelmezés szerinti kódlapot kívánjuk használni. Ha viszont valamilyen oknál fogva a táblázatban feljánlott alternatív kódlapot kívánjuk használni, akkor a fenti bejegyzést a következőképpen kell módosítanunk:

```
COUNTRY=036, 850, C:\DOS\COUNTRY.SYS
```

### Fontos megjegyzések:

1. *Adott országkódhoz csak a táblázatban szereplő alapértelmezés vagy alternatív kód használható.*
2. *Ha a COUNTRY= bejegyzést a CONFIG.SYS állományban elhelyezzük, akkor ezzel az alapértelmezés szerinti, vagy alternatív kódlapot aktív státuszra állítjuk be. Ha viszont ez az alapértelmezés szerinti vagy alternatív kódlap nem azonos a rendszerben rendelkezésre álló és automatikusan aktív hardver kódlapmal, akkor a megfelelő kódlapot előbb be kell töltenünk a számítógép memóriájába és aktivizálnunk kell. Ezeket a műveleteket a későbbiekben még részletesen tárgyaljuk.*

*Az 5.0-s DOS-t megelőző verziók Magyarország kódját nem tartalmazzák, ezért a COUNTRY= bejegyzés beállításánál olyan ország kódját érdemes megadni, amely a nálunk szokásos éé-hh-nn formátumot használja. Például:*

```
COUNTRY=002, , C:\DOS\COUNTRY.SYS
```

*(A 002 a kanadai-francia országkód.)*

A COUNTRY= bejegyzés, amelyet a CONFIG.SYS állományban kell elhelyezni, lehetővé teszi a felhasználónak, hogy az adott ország szokásainak megfelelő dátum- és időkieljelezést, pénznem szimbólumot, az állományok abc-be rendezésekor a sorrendet, valamint az állomány- és könyvtárnevekben alkalmazható karakterek körét definiálja.

A COUNTRY=036 bejegyzéssel a következő formátumok alakíthatók ki Magyarországon:

Megnevezés	Formátum	Példa
Ország	Országkód: 036 Ország jele: HU	
Idő	24 órás kijelzés: óra, perc elkülönítve Órák előtt nulla: nincs	21.59 9.59
Dátum	Rövidített dátum: ÉHN, külön Nulla a napok előtt: Igen Nulla a hónapok előtt: Igen Század jelzés: Igen Teljes dátumforma: éééé. hhhh. n.	1999.09.09 1999. szeptember 9.
Számok	Ezresek után köz Tizedesek jelzése: vesszővel	1 234,56
Pénznem	Pénznem szimbólum: Ft az összeg után köz az összeg és a szimbólum között Két tizedes pontosságig Negatív érték jele: mínuszjel a szám előtt	1 234,56 Ft -1 234,56 Ft
Egyéb	Felsorolás jelölése: pontosvesszővel	A;B;1;2;3

## A KEYB program használata

3.3
-----

KÜLSŐ
-------

Az adott országra jellemző karakterek körének definiálása és azok elrendezése módja a billentyűzetten a KEYB programmal állítható be. A program általános alakja a következő:

```
KEYB [xx[, [yyy][, [meghajtó betűjele:][elérési_út]
filenév.kit]]][ /E ][ /ID:nnn ]
```

Az xx a **C Függelék** táblázata szerinti billentyűkód.

Az yyy a **C Függelék** táblázatában megadott kódlap azonosító száma. A filenév.kit annak az állománynak a neve, amely az adott országra jellemző billentyűelrendezést tartalmazza. Ez az állomány alapértelmezés szerint a **KEYBOARD.SYS**.

A /E kapcsoló előírása azt jelenti a DOS számára, hogy kibővített billentyűzetet használunk 8086-os számítógéppel.

A /ID:nnn kapcsoló olyan országoknál használandó, ahol ugyanazon nyelv esetében a DOS többféle billentyűkiosztást is lehetővé tesz. Magyarországra ez nem vonatkozik.

A KEYB tehát egy program, amely a parancssorból futtatható, mivel azonban minden rendszerindítás után szükségünk van rá, érdemes inkább az AUTOEXEC.BAT állományban elhelyezni. A KEYB program elindítására a DOS egy további lehetőséget is kínál a CONFIG.SYS állományban elhelyezendő INSTALL= bejegyzéssel. Ebben az esetben viszont az előbbi szintaxis a következőre módosul:

```
INSTALL=[[meghajtó betűjele:]elérési út]KEYB.COM[xx[, [yyy] [,
[meghajtó betűjele:] [elérési út]filenév.kit]]] [/E]
[/ID:nnn]
```

A KEYB.COM után szereplő paraméterek és kapcsolók jelentése az előzőekben tárgyalt KEYB parancsnál megadottakkal azonos.

Tegyük fel, hogy magyar karaktereket és magyar billentyűzet-elrendezést szeretnénk, valamint hogy gondoskodtunk a táblázatban alapértelmezésként megjelölt kódlap — 852 — betöltéséről és aktivizálásáról. Ekkor feltételezve, hogy a KEYBOARD.SYS állomány a C: meghajtó — pontosabban fogalmazva a rendszert indító lemez vagy meghajtó — gyökérkatalógusában van, a következőket kell a parancssorba beírni:

```
C:\>KEYB HU <ENTER>
```

Ha a KEYBOARD.SYS állomány pl. a C:\DOS alkönyvtárban van, akkor a végrehajtandó parancs a következő lesz:

```
C:\>KEYB HU, , C:\DOS\KEYBOARD.SYS <ENTER>
```

A kódlapot nem adtuk meg, ezért áll a parancsban egymás után a két vessző. Ezzel azt jelezzük a rendszer felé, hogy a már rendelkezésre álló, aktív és a HU kódnál alapértelmezésként nyilvántartott kódlap használatát várjuk el.

A KEYB parancsnál is lehetőség van az ún. alternatív kódlap használatára. Például ha a Magyarországnál megadott 850-es alternatívára van szükségünk, akkor az előbbi parancsot a következőre kell módosítanunk:

C:\>KEYB HU,850,C:\DOS\KEYBOARD.SYS <ENTER>

### Fontos megjegyzések:

1. Adott billentyűkódhoz csak a **C Függelék** táblázata szerinti alapértelmezés, ill. alternatív kód jöhet számításba.
2. A KEYB parancsnál megadható alternatív kódlappal csak jelezzük a rendszer felé, hogy az adott billentyűkóddal az alternatívaként felajánlott kódlapot kívánjuk használni. Ez azonban nem jelenti azt, hogy a kódlapot már használhatjuk is. Ehhez ui. a kódlapot be kell tölteni, és aktivizálni kell.

A KEYB parancsot paraméterek és kapcsolók nélkül futtatva az éppen aktuális billentyűkód és kódlapszám tudható meg. A parancsnak ezt a formáját a „Kódlappal kapcsolatos információk értelmezése” c. alfejezetben tárgyaljuk.

A KEYB program a betöltött és aktivizált kódlap és a hardver kódlap közötti átkapcsolást is lehetővé teszi. A Ctrl—Alt—F1 kombinációval a hardver kódlapra lehet átváltani, míg a Ctrl—Alt—F2 a betöltött és aktivizált kódlapra való visszaváltás valósítható meg. Mindjárt adódik a kérdés: Miért van erre szükség? Vegyük példaként a magyar ékezetes billentyűzetet!



### Magyar ékezetes billentyűzet

Mivel a magyar abc több karaktert tartalmaz, mint az angol, ezért az összes speciális magyar karaktert (ű, Ű, í, Í, ö, Ö, ó, Ó, ü, Ü, é, É, á, Á) csak úgy lehet egyidejűleg és közvetlenül a billentyűzetről gépelni, ha azokat bizonyos írásjel karaktereket ábrázoló billentyűkre (pl. ;, ', [, ], -, =, ', \) definiálják rá. Ha tehát ezekre a karakterekre van szükségünk, akkor a magyar billentyűzetről az angol billentyűzetre kell átváltanunk. Majd ha ismét a magyar karakterekre van szükségünk, akkor a magyar billentyűzetre kell visszaváltanunk. A fenti két billentyűkombináció ezt az átkapcsolást teszi lehetővé.

**Fontos megjegyzés!** Mivel Magyarország sajátosságait csak az 5.0-s DOS tudja figyelembe venni, korábbi DOS verziókkal tehát a rendszer kínálta eszközökkel a magyar ékezetes billentyűzet nem állítható elő, ezért a magyar ékezetes billentyűzet létrehozásához külön programot kell futtatni. A program neve pl. kbhun.exe vagy keybhun.exe. Mivel programról van szó, ezért azt a parancssorból kell végrehajtani, de parancsállományban is — nem feltétlenül az AUTOEXEC.BAT-ban — el lehet helyezni. A program alkalmazása után a magyar ékezetes karakterek rendszerint azonnal rendelkezésre állnak. Általában ezek a programok is igazodnak ahhoz a szabályhoz, hogy az angol billentyűzetre a Ctrl—Alt—F1 kombinációval, a magyar billentyűzetre pedig a Ctrl—Alt—F2 kombináció lenyomásával lehet átváltani.

A KEYB programmal a már betöltött és aktív kódoldalon belül az éppen felhasznált karakterkészlet és annak billentyűzeten való elrendezése választható meg. Például ha az 5.0-s DOS által felkínált magyar billentyűzetet és karakterkészletet akarjuk használni, valamint a speciális magyar karaktereket tartalmazó 852-es kódlap betöltéséről és aktivizálásáról már gondoskodtunk, akkor a következő KEYB parancsra van szükségünk, feltételezve, hogy a KEYBOARD.SYS állomány a C:\DOS alkönyvtárban van:

```
KEYB HU, , C:\DOS\KEYBOARD.SYS
```

## A kódlapok betöltése és aktivizálása

A DOS — egyéb beállítások hiányában — alapértelmezésként mindig a billentyűzetbe, monitorba és nyomtatóba beépített hardver kódlapot használja. Ha azonban olyan nyelven akarunk gépelni, amelynek karaktereit ez a hardver kódlap nem tartalmazza — így van ez a magyar nyelv esetében is —, a hat előkészített kódlap — l. **B Függelék** — közül kell a megfelelőt a számítógép memóriájába betöltenünk és aktivizálnunk.

Az előkészített kódlapokat .CPI kiterjesztésű állományok tartalmazzák a DOS lemezen. A memóriába több kódlap is betölthető, aktív azonban mindig csak egy lehet.

Adott országhoz csak azok a kódlapok használhatók, amelyeket a **C Függelék** táblázata alapértelmezés szerinti vagy alternatív feliratú oszlopa tartalmaz. Így Magyarországhoz vagy a 852-es vagy a 850-es kódlap használható. Ha



tehát ezen kódlapok egyikét — vagy mindkettőt — használni akarjuk, akkor a következő műveleteket kell végrehajtanunk:

1. A képernyőt és a billentyűzetet elő kell készítenünk a kódlap fogadására. A művelet végrehajtásának eszköze a CONFIG.SYS állományban elhelyezett megfelelő DEVICE=DISPLAY.SYS bejegyzés.
2. A megfelelő MODE parancs futtatásával be kell töltenünk a memóriában azt vagy azokat a kódlapokat, amelyeket használni akarunk.
3. A megfelelő MODE parancs futtatásával a használni kívánt — és már betöltött — kódlapot aktivizálnunk kell.

Megjegyzés: A DOS a kódlap aktivizálását az NLSFUNC és CHCP parancspár alkalmazásaival is lehetővé teszi.

Most pedig lássuk részletesen a műveleteket!

## A képernyő és a billentyűzet, vagyis a CON eszköz előkészítése a kódlapok fogadására

Mint tudjuk, hardver kódlappal minden számítógépes környezet rendelkezik. Ez határozza meg a számítógépben alapértelmezésként használható karakterkészletet. Ha azonban olyan nyelven szeretnénk gépelni, amelynek karaktereit ez a hardver kódlap nem tartalmazza — pl. magyar nyelven! —, akkor az adott országhoz tartozó megfelelő kódlapot vagy kódlapokat telepítenünk kell. Ezen művelet első lépéseként a DISPLAY.SYS eszközmeghajtó program CONFIG.SYS állományban való elhelyezésével monitort és a billentyűzetet — CON eszközt — elő kell készítenünk a kódlapok fogadására.

A bejegyzés általános alakja a következő:

```
DEVICE=[meghajtó_betűjele:][elérési_út]DISPLAY.SYS
CON [:]=(típus[, [hwcp][,n]])
```

Típusként a rendelkezésre álló monitor típusát — EGA vagy LCD — kell megadni. A VGA monitorok esetében is EGA-t kell előírni. Ha ezt a paramétert elhagyjuk, akkor a DOS magától is képes arra, hogy a rendszerünkben rendelkezésre álló monitor típusát lekérdezze.

### Megjegyzések:

1. Bár típusként a bejegyzésben CGA vagy a MONO is rögzíthető, ennek azonban nincs hatása, mivel ezek a monitorok nem teszik lehetővé a kódlapváltás műveletét.

## 2. Az EGA, VGA stb. rövidítések jelentése a fejezet végén található meg.

A hwcp-vel megjelölt paraméter, vagyis hardver kódlap azt a kódlapot jelenti, amely a rendszerünkbe van építve. A hwcp leggyakoribb értéke 437.

### Fontos megjegyzés:

Ha a magyar karakterkészletet tartalmazó 852-es kódlapot akarjuk használni, és a rendszerünkben 437-es hardver kódlap van, akkor ennél a bejegyzésnél ne rögzítsük a 437-es értéket, ui. ez a kódlap a magyar országgóddal és billentyűkóddal összeférhetetlen. Ha ezt mégis megteesszük, a KEYB programtól a következő hibaüzenetet kapjuk:

```
One or more CON Code Pages invalid for given keyboard code
```

Egy vagy több érvénytelen CON kódlap a megadott billentyűkódhoz.

Az n helyébe a betölteni kívánt kódlapok darabszámát kell megadni. A rendszer ui. annyi kódlap számára foglal helyet, amilyen értéket itt előírunk. VGA és EGA monitor esetén n értéke 1-től 6-ig terjedhet, LCD esetén azonban csak 1 lehet.

A bejegyzés szintaxisára külön felhívjuk az Olvasó figyelmét! A paraméte-  
reket ui. zárójelek közé kell foglalni, és egymástól vesszővel kell elválasztani. A vesszőkre pedig akkor is szükség van, ha az adott paramétert nem adjuk meg, az utána következőt viszont igen.

A bejegyzésnek van egy másik alakja is, amely az egyes kódlapok által támogatandó részkarakterkészlet számát teszi előírhatóvá. Ezt az alakot itt nem tárgyaljuk.

### Lássunk a bejegyzésre egy példát!

```
DEVICE=C:\DOS\DISPLAY.SYS CON=(EGA,437,1)
```

### Értelmezzük a parancsot!

Ha a DISPLAY.SYS állomány nem a gyökérkönyvtárban van, akkor a hozzá vezető útvonalat elő kell írni, pl. C:\DOS.

A felhasznált monitor típusa EGA. (VGA esetén is EGA-t kell használni!)

A következő paraméter a hardver kódlap száma.

Az utolsó paraméter 1, ami azt jelenti, hogy a rendszer 1 előkészített kódlap számára foglal helyet. Ismételten hangsúlyozzuk, hogy itt nem a használni kívánt kódlap azonosító számát, hanem csak a betölteni kívánt kódlapok darabszámát kell megadni.

Ahhoz, hogy a billentyűzettel és a monitorral egy vagy több kódlapot használni tudjunk, a DISPLAY.SYS eszközmeghajtó program CONFIG.SYS állományban való elhelyezésével a monitort és a billentyűzetet — CON eszközt — elő kell készítenünk a kódlapok fogadására. Például ha EGA monitorunk van, és egy előkészített kódlap számára akarunk helyet biztosítani, valamint a DISPLAY.SYS állomány a C:\DOS alkönyvtárban van, akkor a következő bejegyzést kell elhelyeznünk a CONFIG.SYS állományban:

```
DEVICE=C:\DOS\DISPLAY.SYS CON=(EGA,,1)
```

Ezzel a bejegyzéssel betöltendő kódlap számára csak a helyet foglaljuk le. A betöltést és az aktivizálást még külön el kell végeznünk.

## A nyomtató előkészítése a kódlapok használatára

Vannak már olyan nyomtatók is, amelyeknél lehetőség nyílik a DOS kínálta kódlapok használatára. A monitorhoz hasonlóan azonban itt is az előkészítésnek kell lennie az első lépésnek. A művelet eszköze a PRINTER.SYS eszközmeghajtó program telepítése a DEVICE= bejegyzéssel a CONFIG.SYS állományban. Először lássuk a bejegyzés általános alakját!

```
DEVICE=[meghajtó_betűjele:][elérési_út]  
PRINTER.SYS LPTx=(típus[, [hwcp][,n]])
```

Az LPTx azt a párhuzamos portot jelenti, amelyhez csatlakoztatott nyomtatón a kódlapváltást végre akarjuk hajtani.

Típusként, amely a használatban levő nyomtató típusát adja meg, a következő értékek adhatók meg:

4201 IBM Proprinter II és III, 4201 és 4202 modellek

4208 IBM Proprinter X24E 4207-es modell

IBM Proprinter XL24E 4208-as modell

5202 IBM Quietwriter III 5202-es modell

A hwcp a nyomtató hardver kódlapjának számát jelenti. Az n helyébe ebben az esetben is a használni kívánt kódlapok darabszámát kell megadni. Például:

```
DEVICE=C:\DOS\PRINTER.SYS LPT1=(4201,437,1)
```

A bejegyzés azt feltételezi, hogy a DISPLAY.SYS állomány nem a gyökérkönyvtárban van, ezért tartalmazza a hozzá vezető útvonalat is, C:\DOS. A felhasznált nyomtató típusa 4201. A következő megadott paraméter a hardver kódlap száma, vagyis a 437. A legutolsó paraméter az 1-es, amivel 1 előkészített kódlapnak foglalunk helyet.

**Megjegyzés:** A kódlapváltást nem minden nyomtatótípus teszi lehetővé. Az erre vonatkozó részleteket a nyomtató kézikönyvéből lehet megtudni.

Vannak már olyan nyomtatók, amelyeknél a monitorhoz és a billentyűzethez hasonlóan lehetséges az 5.0-s DOS által kínált kódlapok használata. Ezt a műveletet a CONFIG.SYS állományban elhelyezett DEVICE=PRINTER.SYS bejegyzéssel kell előkészíteni. Például:

```
DEVICE=PRINTER.SYS LPT1=(4201,437,1)
```

## A CHCP parancs használatának előkészítése: az NLSFUNC parancs futtatása

3.3

KÜLSŐ

Az NLSFUNC parancsot csak a következő esetekben kell alkalmazni:

1. A már betöltött kódlap aktivizálása a későbbiekben tárgyalt CHCP parancssal.
2. Kódlapváltás a rendszer újraindítása nélkül.

Az NLSFUNC parancs futtatására a DOS a következő három lehetőséget kínálja:

- Futtatás a parancssorból.
- Elhelyezés az AUTOEXEC.BAT állományban.

**Fontos megjegyzés:** Az *NLSFUNC* parancs végrehajtásának időpontja nem tetszőleges, *ui.* sorrendben meg kell előznie a kódlap betöltésére és aktivizálásra vonatkozó parancsokat. Ha tehát a parancsot az *AUTOEXEC.BAT* állományban helyezzük el, akkor erre külön ügyeljünk!

— Futtatás a *CONFIG.SYS* állományból az *INSTALL=* bejegyzés felhasználásával. Például:

```
INSTALL=C:\DOS\NLSFUNC.EXE
```

**Fontos megjegyzés!** A *CONFIG.SYS* állományt a rendszer előbb dolgozza fel, mint az *AUTOEXEC.BAT* állományt. Így ha az *NLSFUNC* parancsot az *INSTALL=* bejegyzés felhasználásával futtatjuk, akkor az előbbi feltételnek automatikusan eleget teszünk.

Ha valamennyi eszközre egyszerre akarjuk a kódlapváltás műveletét végrehajtani, vagy a rendszer újraindítása nélkül akarunk kódlapot váltani, akkor az *NLSFUNC* parancs maradjon! Ha azonban a kódlapváltást csak egyetlen eszköz esetében akarjuk elérni, akkor ne hajtsuk végre az *NLSFUNC* parancsot!

Most pedig lássuk hogyan kell a kódlapot a memóriába betölteni!

## A kódlap betöltése a memóriába a MODE paranccsal

<b>5.0</b>
<b>KÜLSŐ</b>

**Megjegyzés:** Bár a MODE parancsot már az 1.0-s DOS is tartalmazta, az itt tárgyalt alak csak az 5.0-s DOS-ban érvényes.

Az eddig ismerttetett parancsok és bejegyzések a kódlap fogadását előkészítették, nem végezték el azonban annak betöltését. Most tehát ezt a műveletet tárgyaljuk.

A kódlapok betöltését a MODE CODEPAGE PREPARE paranccsal kell elvégezni. A parancsnak, amely egyébként MODE CP PREP-nek rövidíthető, a következő az általános alakja:

```
MODE eszköz CODEPAGE
PREPARE=((yyy[...]) [meghajtó_betűjele:] [elérési_út]filenév.CPI)
```

Eszközként mindig azt a DOS eszköznevet kell előírni, ahová a kódlapot be akarjuk tölteni. Így tehát a monitor és a billentyűzet esetén a CON, a nyomtató esetén pl. az LPT1 eszköznév jöhet számításba.

Az yyy a kódlap azonosító száma. Ha több kódlapot akarunk betölteni, akkor azokat itt kell felsorolnunk egymás után. Mint a szintaxisból is látszik, a kódokat ebben az esetben nem kell vesszővel elválasztani egymástól, de zárójelbe kell foglalni.

Végül azt a .CPI kiterjesztésű állományt — valamint ha az állomány nem a gyökérkönyvtárban van, akkor a hozzá vezető utat — kell előírni, amely a kódlapot tartalmazza. Az EGA/VGA kódlapokat az EGA.CPI, az LCD kódlapot az LCD.CPI, a nyomtatókra vonatkozó kódlapokat pedig értelemszerűen a 4201.CPI, 4208.CPI és az 5202.CPI állományok tartalmazzák.

A következő parancs pl. a 852-es kódlapot tölti be a C:\DOS alkönyvtárban elhelyezkedő EGA.CPI állományból:

```
MODE CON CP PREP=((852)C:\DOS\EGA.CPI)
```

Figyeljük meg, hogy az eszköz neve — CON — a MODE után következik, valamint hogy a paramétereket zárójelbe kell foglalni. Ha a DEVICE=DISPLAY.SYS bejegyzéssel egynél több kódlap számára foglaltunk helyet, akkor azok betöltését is most tudjuk elvégezni. Tegyük fel, hogy a

CONFIG.SYS állományban a következő DEVICE=DISPLAY.SYS bejegyzés szerepel:

```
DEVICE=C:\DOS\DISPLAY.SYS CON=(EGA,,2)
```

Ezzel tehát a monitor és a billentyűzet számára két előkészített kódlapnak foglaltunk helyet. A két kódlapot viszont a következő MODE parancs tölti be:

```
MODE CON CP PREP=((852 850)C:\DOS\EGA.CPI)
```

A nyomtató esetén is a fentiekhez hasonlóan kell elvégezni a kódlap vagy kódlapok betöltését, csak arra kell figyelni, hogy a MODE után a megfelelő eszköznevet, paraméterként pedig a megfelelő .CPI állományt használjuk. Például:

```
MODE LPT1 CP PREP=((852 850)C:\DOS\5202.CPI)
```

A kódlap memóriába való betöltéséhez a megfelelő eszköznévvel ellátott MODE CODEPAGE PREPARE parancsot kell végrehajtani. Például ha a CON eszközzel kapcsolatban a 852-es kódlapra van szükségünk, akkor a következő MODE parancsot kell alkalmaznunk:

```
MODE CON CP PREP=((852)C:\DOS\EGA.CPI)
```

## A kódlap aktivizálása

Miután a kódlap betöltését előkészítettük és azt be is töltöttük, már csak aktivizálnunk kell. Erre a DOS két lehetőséget kínál:

1. A CHCP parancs végrehajtása, amellyel a kódlap valamennyi eszköz esetében egyszerre aktivizálható. A parancs általános alakja:

```
CHCP [nnn]
```

ahol az nnn az aktivizálni kívánt kódlap számát jelenti. Például a

```
CHCP 852
```

parancs a 852-es kódlapot aktivizálja.

**Fontos megjegyzések:**

1. A *CHCP*-vel végrehajtott kódlapváltás csak a következő műveletek után szerepelhet:
    - A megfelelő *COUNTRY*= bejegyzés elhelyezése a *CONFIG.SYS* állományban, szükség esetén előírva a *COUNTRY.SYS* állomány helyét is.
    - Az *NLSFUNC* parancs futtatása.
    - Az aktivizálni kívánt kódlap betöltése.
  2. Csak olyan eszközöknél lehet kódlapváltást végrehajtani, amelyek ezt lehetővé teszik. Például a *MONO* és a *CGA* monitor csak hardver kódlappal rendelkeznek és kódlapváltásra nem alkalmasak.
  3. A kódlapszámoknak összhangban kell lenniük az országgóddal és a billentyűkóddal. Tehát akármilyen kódlapot nem lehet betölteni és aktivizálni, csak azt, amelyik az országgódokat tartalmazó táblázatban — **C Függelék** — az adott országgód és billentyűkód sorában szerepel.
2. A *MODE* eszköz *CODEPAGE SELECT* parancs felhasználásával egyetlen eszközre aktivizálható az előzetesen már betöltött kódlap. Ebben az esetben az *NLSFUNC* parancs előzetes végrehajtására nincs szükség. A parancs általános alakja a következő:

*MODE* eszköz *CODEPAGE SELECT*=yyy

Eszközként érvényes DOS eszköznevet — *CON*, *LPT1*, *LPT2*, *LPT3* — kell előírni, az yyy helyén pedig az aktivizálni kívánt kódlap azonosító számát kell megadni.

Lássunk erre is egy példát!

*MODE LPT1 CP SELECT*=850

Ez a parancs az *LPT1*-hez csatlakoztatott nyomtatónál a 850-es kódlapot aktivizálja.

(Mint látható, a *CODEPAGE* ebben az esetben is *CP*-nek rövidíthető.) Ha a 850-es kódlapot előzetesen nem töltöttük be, akkor a parancstól a következő hibaüzenetet kapjuk:

Invalid code page

Érvénytelen kódlap



Ezzel tehát valamennyi olyan utasítást, bejegyzést és meghajtóprogramot áttekintettük, amely egy adott ország nemzeti sajátosságainak figyelembevételéhez szükséges, még akkor is, ha a számítógépes környezetben rendelkezésre álló hardver kódlap nem tartalmazza azokat a karaktereket, amelyek az adott nyelv használatához szükségesek.

Ha a már betöltött kódlapot valamennyi eszköz esetében egyszerre akarjuk aktivizálni, akkor a CHCP parancsot alkalmazhatjuk. Például:

```
CHCP 850
```

A MODE eszköz CODEPAGE SELECT paranccsal a már betöltött kódlap egyetlen eszköz esetében aktivizálható. Például:

```
MODE CON CP SELECT=852
```

A megfelelő beállítások elvégzéséhez a meglévő állapotot kell lekérdezni. Ehhez a DOS három eszközt — parancsot — kínál, amelyeket a következőkben röviden áttekintünk.

## *A kódlapmal kapcsolatos információk lekérése*

A DOS a kódlapmal kapcsolatos információk lekérézéséhez a következő három parancsot kínálja:

```
CHCP  
KEYB  
MODE
```

Ezen parancsok paraméterezett változatait már az előző fejezetekben részletesen tárgyaltuk. Ha viszont ezeket a parancsokat paraméterek nélkül futtatjuk, akkor a kódlapmal kapcsolatos információkhoz juthatunk. További magyarázat helyett inkább hajtsuk végre egymás után a parancsokat. Kiinduló helyzetként tételezzük fel, hogy az adott számítógépes környezetben nem történt még semmilyen beállítás, tehát csak a hardver kódlap áll rendelkezésre, és sem a COUNTRY= bejegyzést, sem a KEYB parancsot nem alkalmaztuk még.

Ekkor a CHCP parancsot futtatva az aktív hardver kódlap száma kapható meg. Például:

```
C:\>CHCP <ENTER>  
Active code page: 437
```

Az aktív kódlap: 437

Ez nyilvánvalóan a hardver kódlap száma, mivel a rendszer ezt mint alapértelmezést automatikusan aktívra állítja be. Itt jegyezzük meg, hogy a CONFIG.SYS állományban elhelyezett és megfelelően paraméterezett COUNTRY= az adott országhoz tartozó alapértelmezés szerinti kódlapot aktív státuszra állítja be még akkor is, ha azt még be sem töltöttük. Ha tehát a CONFIG.SYS állomány pl. a

```
COUNTRY=036,,C:\DOS\COUNTRY.SYS
```

bejegyzést tartalmazza, akkor a fenti CHCP parancs a következő üzenetet adja:

```
Active code page: 852
```

Ez viszont azt a téves elképzelést sugallja, mintha ez a kódlap már használható is lenne, vagy ez lenne a hardver kódlap száma. Ne essünk bele ebbe a csapdába! A hardver kódlap számát a CHCP paranccsal csak úgy kaphatjuk meg, ha a fenti bejegyzést letiltva

```
REM COUNTRY=036,,C:\DOS\COUNTRY.SYS
```

újából indítjuk a rendszert, és a CHCP parancsot ismételten végrehajtjuk.

Ezután lássuk milyen üzenetet kapunk a KEYB parancstól, amely a billentyűzetre és a monitorra vonatkozó kódlap lekérdezésére alkalmas. Ha a KEYB parancsot még nem futtattuk, akkor a

```
C:\>KEYB <ENTER>
```

vagyis a parancs végrehajtása a következő üzenetet eredményezi:

KEYB has not been installed  
Active code page not available from CON device

```
A KEYB program telepítése még nem történt meg.  
Aktív kódlap nem áll rendelkezésre a CON eszközből.
```

Ha viszont az egyik kódlapot már betöltöttük és aktivizáltuk, valamint a KEYB parancsot is futtattuk, akkor a KEYB paraméterek nélküli használata pl. a következő üzenet megjelenését eredményezi:

```
Current keyboard code: HU code page: 852  
Current CON code page: 852
```

```
Az éppen aktuális billentyűkód: HU kódlap: 852.  
Az éppen aktuális CON-ra vonatkozó kódlap: 852.
```

Végül futtassuk a legrészletesebb információt nyújtó MODE parancsot is!

Kiinduló helyzetként most is azt feltételezzük, hogy a rendszerben még semmilyen beállítást — COUNTRY= bejegyzés, KEYB parancs — nem hajtottunk végre.

```
C:\>MODE <ENTER>  
Status for device LPT1:  
LPT1: not rerouted  
Retry=NONE  
Codepage operation not supported on this device  
Status for device LPT2:  
LPT2: not rerouted  
Status for device LPT3:  
LPT3: not rerouted  
Status for device CON:  
Codepage operation not supported on this device  
Status for device COM1:  
Retry=NONE  
Status for device COM1:  
Retry=NONE  
C:\>
```

```
Az LPT1 eszközre vonatkozó állapot:  
LPT1: nincs átirányítva  
Újra próbálkozás=TÖRÖLVE  
Ezzel az eszközzel kapcsolatban egyetlen kódlap sincs  
támogatva  
Az LPT2 eszközre vonatkozó állapot:  
LPT2: nincs átirányítva  
Az LPT3 eszközre vonatkozó állapot:  
LPT3: nincs átirányítva  
A CON eszközre vonatkozó állapot:  
Ezzel az eszközzel kapcsolatban egyetlen kódlap sincs  
támogatva  
A COM1 eszközre vonatkozó állapot:  
Újra próbálkozás=TÖRÖLVE  
A COM2 eszközre vonatkozó állapot:  
Újra próbálkozás=TÖRÖLVE
```

Most, hogy már ismerjük a kódlappal kapcsolatos információk lekérdezésének eszközeit, lássunk néhány példát a fejezetben bemutatott bejegyzések és parancsok alkalmazására!

## ***Konkrét helyzetek, konkrét példák***

Ebben az alfejezetben olyan konkrét példákat mutatunk be, amelyek alapján az Olvasó a saját környezetét is be tudja majd állítani. Bizonytalanság esetén érdemes elolvasni még egyszer azt az alfejezetet, ahol az adott témát tárgyaltuk. Ne feledkezzünk meg arról sem, hogy a bejegyzések, ill. parancsok végrehajtásának sorrendje nem teljesen tetszőleges. Továbbá van néhány olyan parancs is (pl. a KEYB vagy az NLSFUNC), amelyet az INSTALL= bejegyzés felhasználásával a CONFIG.SYS állományban is el lehet helyezni. Ekkor a következőkre kell ügyelnünk:

1. A parancsokat csak egy helyen, tehát vagy az AUTOEXEC.BAT-ban, vagy a CONFIG.SYS-ben szerepeltessük.
2. A rendszer a CONFIG.SYS állományban szereplő bejegyzéseket mindig előbb dolgozza fel, mint az AUTOEXEC.BAT állomány parancsait. Ha tehát az INSTALL= bejegyzéssel a parancsot a CONFIG.SYS állományba tesszük át, viszont az adott parancsot az AUTOEXEC.BAT-ban szereplő

programok végrehajtása után kell elvégeznünk, akkor hibás sorrendet alakíthatunk ki, bármennyire is igyekszünk.

Külön felhívjuk az Olvasó figyelmét, hogy a bejegyzések szintaxisában minden alkalommal opcionálisan szerepel a megfelelő meghajtóprogramhoz vezető útvonal ([meghajtó\_betűjele:][elérési\_út]). Az útvonal bár opcionális, minden olyan esetben rögzíteni kell azonban, amikor az adott meghajtóprogram nem a gyökérkönyvtárban van.

Mielőtt a konkrét példákra rátérnénk, a következő táblázatban rögzítjük a kódlap betöltéséhez, aktivizálásához és a nemzeti sajátosságok figyelembevételéhez szükséges parancsok sorrendjét.

Parancs/Bejegyzés	Művelet	Hely
1. COUNTRY= DEVICE=DISPLAY.SYS DEVICE=PRINTER.SYS	Az országcód beállítása Helyfoglalás a kódlap számára	CONFIG.SYS CONFIG.SYS CONFIG.SYS
2. MODE CON CP PREP MODE LPT1 CP PREP	Kódlap(ok) betöltése az eszközökre	AUTOEXEC.BAT AUTOEXEC.BAT
...		
3. MODE CON CP SELECT MODE LPT1 CP SELECT	A használni kívánt kódlap aktivizálása minden eszköz esetében	AUTOEXEC.BAT AUTOEXEC.BAT
...		
4. KEYB	Billentyűkód beállítása	AUTOEXEC.BAT

1. Átváltás a magyar nyelv dátum- és idő kijelzési szokásaira, karakterkészletére, valamint billentyűzetelrendezésére  
Kiindulás: EGA (lehet VGA is!!) monitor  
Betöltött és aktualizált 852-es kódlap

CONFIG.SYS:

·  
·  
·

COUNTRY=036,,C:\DOS\COUNTRY.SYS

·  
·  
·

AUTOEXEC.BAT:

.  
.  
.

KEYB HU,,C:\DOS\KEYBOARD.SYS

.  
.  
.

vagy:

CONFIG.SYS:

.  
.  
.

COUNTRY=036,,C:\DOS\COUNTRY.SYS

INSTALL=C:\DOS\KEYB.COM HU,,C:\DOS\KEYBOARD.SYS

.  
.  
.

2. Egy előkészített kódlap betöltése, aktivizálása és azon beállítás a magyar nyelv sajátosságainak figyelembevételére

Kiindulás: EGA (lehet VGA is!!) monitor, 437-es hardver kódlappal

Cél: A 852-es kódlap betöltése és aktualizálása, valamint a nyelvi sajátosságok (karakterkészlet, billentyűzet, dátumkijelzés) figyelembevétele

CONFIG.SYS:

.  
.  
.

COUNTRY=036,,C:\DOS\COUNTRY.SYS

DEVICE=C:\DOS\DISPLAY.SYS CON=(EGA,,1)

.  
.  
.

AUTOEXEC.BAT:

.  
.  
.

MODE CON CP PREP=((852)C:\DOS\EGA.CPI)

MODE CON CP SELECT=852

KEYB HU,,C:\DOS\KEYBOARD.SYS

.  
.  
.

**Megjegyzés:** Mint tudjuk, a KEYB parancsot az INSTALL= bejegyzéssel a CONFIG.SYS állományban is el lehetne helyezni, így azonban a KEYB végrehajtására még a kódlap betöltése és aktualizálása előtt kerülné sor. Ha tehát mégis így járnánk el, a DOS-tól hibaüzenetet kapnánk.

A magyar billentyűzetről az angolra a Ctrl—Alt—F1 kombinációval tudunk átváltani, a visszaváltást pedig a Ctrl—Alt—F2 kombináció teszi lehetővé.

3. Két előkészített kódlap betöltése, az egyik aktivizálása és azon beállítás a magyar nyelv karaktereinek és szokásainak figyelembevételére.

**Fontos megjegyzés!** A C Függelék táblázata szerint a magyar nyelv esetében alternatívaként csak a 850-es kódlap jöhet számításba, ezért másodiknak csak ezt lehet betölteni.

Kiindulás: EGA (lehet VGA is!!) monitor, 437-es hardver kódlappal

Cél: A 852-es és a 850-es kódlap betöltése, a 852-es kódlap aktivizálása, valamint a nyelvi sajátosságok (karakterkészlet, billentyűzet, dátumkijelzés) figyelembevétele

CONFIG.SYS:

·  
·  
·

COUNTRY=036,,C:\DOS\COUNTRY.SYS

DEVICE=C:\DOS\DISPLAY.SYS CON=(EGA,,2)

·  
·  
·

AUTOEXEC.BAT:

·  
·  
·

MODE CON CP PREP=((852 850)C:\DOS\EGA.CPI)

MODE CON CP SELECT=852

KEYB HU,,C:\DOS\KEYBOARD.SYS

·  
·  
·

## *Ismétlés = Kulcs a sikerhez*

A nemzeti sajátosságok figyelembevételére az 5.0-s DOS a következő beállításokat teszi lehetővé:

- A dátum, idő és a pénznem kijelzésének módja, a karakterek abc-be rendezéséhez felhasznált sorrend és az állományneveknél alkalmazható karakterkészlet kialakítása az adott ország igényeinek megfelelően. A művelet eszköze a megfelelően paraméterezett COUNTRY= bejegyzés.
- Karakterkészlet- (kódlap-) váltás annak érdekében, hogy olyan karakterek is gépelhetők, megjeleníthetők, ill. kinyomtathatók legyenek, amelyeket a hardver vagy az éppen aktív kódlap nem tartalmaz. Ehhez először a kódlapot az adott eszközre be kell tölteni (pl. MODE CON CP PREP=), majd pedig aktivizálni kell (pl. MODE CON CP SELECT=).
- Az éppen aktuális karakterkészlet és annak billentyűzeten való elrendezése az adott ország szokásainak megfelelően, amely a már rendelkezésre álló kódlapon a megfelelő KEYB parancs futtatásával érhető el.

## *Mit jelent?*

**MONO.** Az angol MONOCHROME szó rövidítéséből származó megjelölés. A kifejezést olyan monitorokra használják, amelyek csak egyetlen szín, pl. feketén fehér, megjelenítésére képesek.

**CGA.** Az angol Color Graphics Adapter (színes grafikus adapter) szavak rövidítéséből származó megjelölés, amelyet alacsonyabb színfelbontású monitorok megjelölésére használnak. (25 vízszintes sor 16 színnel.)

**EGA.** Az angol Enhanced Graphics Adapter (kibővített grafikus adapter) szavak rövidítéséből származó megjelölés, amelyet nagyobb színfelbontású monitorok megjelölésére használnak. (64 színt tartalmazó palettából 16 szín kiválasztása és a CGA-nál nagyobb grafikus felbontás.)

**VGA.** Az angol Video Graphics Adapter (video grafikus adapter) szavak rövidítéséből származó megjelölés, amelyet már több különböző grafikus üzemmód és 262144 színt tartalmazó palettából akár 256 szín kiválasztására tettek alkalmassá.

**LCD.** Az angol Liquid Crystal Display (folyadékkristályos kijelző) szavak rövidítéséből származó megjelölés. A kifejezést olyan monitorok esetében használják, amelyek a hagyományos tv-képernyőktől eltérő módszerrel jelenítik meg a karaktereket.



# Memóriakezelés az 5.0-s DOS használatára esetén

**Már tanultuk:** memória, RAM

**Memória:** Amikor a számítógép futtat egy programot, akkor azt előbb a lemeztől (akár hajlékony, akár merevlemez) az operatív memóriába kell beolvasnia. Az adott program végrehajtása után egy másik program tölthető a memóriába.

**RAM** (Random Access Memory = közvetlen elérésű tár): A programok futását lehetővé tevő memória, amelynek tartalma a számítógép kikapcsolásakor elvész.

## Memóriatípusok

A felhasználók számára készített újabb és újabb alkalmazói programok egyre több szolgáltatást nyújtanak, egyre jobban kidolgozott, részletesebb képernyőképet jelenítenek meg. Persze ennek az az ára, hogy növekszik a programok mérete is. Így jelentkezik az igény arra, hogy a számítógépekbe beépített memóriát minél jobban ki lehessen használni, ill. a már meglévőt bővíteni lehessen.

Az 5.0-s DOS egyik legfontosabb szolgáltatását azok az új parancsok és CONFIG.SYS állományban elhelyezhető bejegyzések jelentik, amelyek a hagyományos memória jobb kihasználását, ill. a hagyományos memória bővítését jelentő ún. kiterjesztett és kibővített memória kezelést teszik lehetővé.

Elsőként ismerkedjünk meg az 5.0-s DOS által kezelhető memóriatípusokkal:

- hagyományos,
- kibővített (extended),
- kiterjesztett (expanded).

Ez utóbbi két fogalmat nagyon gyakran felcserélik, összetévesztik, sőt még az is előfordul, hogy a két fogalmat egymás helyett használják. A bonyodalmat

fokozza, hogy a kibővített memória különböző területeihez — a használatától függően — saját azonosító nevek tartoznak. Ebben az alfejezetben tehát a fogalmak tisztázása az elsődleges célunk.

## Hagyományos memória

Hagyományos memóriának azt a minimális 256 Kb-át, maximálisan 640 Kb-át méretű ún. elsődleges memóriát nevezik, amelyet minden PC (így a régebben készült XT-k is) tartalmaz. A DOS és a végrehajtott programok ezt a memóriát speciális parancsok igénybevétele nélkül használják. A 2. Fejezetben már tárgyalt CHKDSK paranccsal ezen hagyományos memória méretéről és az éppen rendelkezésre álló szabad bájtok számáról kaphatunk információt. A DOS — ha másként nem utasítjuk — a hagyományos memória legalsó részén helyezi el magát. Ha ezt követően a számítógépbe memóriarezidens szoftvereket töltünk be, akkor azok közvetlenül a DOS terület felett foglalnak helyet. Ilyen memóriarezidens program pl. az 1. Fejezetben megismert DOSKEY, a 3. Fejezetben megismert MIRROR, vagy pl. a 6. Fejezetben bemutatott DEVICE= bejegyzéssel telepíthető eszközmeghajtó állományok. A memóriarezidens szoftverek elhelyezése után megmaradó memória mérete adja annak a legnagyobb programnak a méretét, amelyet az adott számítógépben futtatni lehet.

## A kibővített (extended) memória

### *Rövidítése: XMS*

A kibővített memória a hagyományos memória folytatásaként képzelhető el. Talán úgy a legegyszerűbb ezt a fogalmat megjegyezni, ha elképzeljük, amint a már meglévő 640 Kb-át hagyományos memóriánkat külön kártya behelyezése nélkül magán az alaplapon bővítjük tovább. Ennek a bővítésnek lényeges állomásai vannak. Lássuk melyek ezek!

1. Az első állomás az, amikor a bővítéssel az 1024 Kb-átos (1 Mb-át!) határhoz érünk. A 640 Kb-átot az 1024 Kb-átból kivonva 384 Kb-átot kapunk. A legtöbb 80 286-os vagy 80 386-os számítógép már alapértelmezésként 640 Kb-át hagyományos és 384 Kb-át kibővített memóriával rendelkezik. Ez a bővítés annyira elterjedt és természetes, hogy ezt a memóriát rendszert nem is nevezik bővített memóriának. Csak azt a részt illetik ezzel az el-

nevezéssel, amely az 1 Mb-ot felett következik. Erre a 384 Kb-ot viszont a felső memóriaterület (upper memory area) elnevezést alkalmazzák.

Mire használható ez a memóriaterület:

- hardver eszközök kezelését végző programok tárolására;
- a kiterjesztett memória használatához szükséges 64 Kb-otos ún. oldalkeret tárolására (l. a kiterjesztett memóriát tárgyaló következő alfejezetet!);
- megfelelő DOS parancsok és bejegyzések alkalmazása esetén DOS segédprogramok (pl. a MIRROR, DOSKEY), eszközmeghajtó programok (pl. MOUSE, RAMDRIVE, SMARTDRV) vagy a nemzeti sajátosságok figyelembevételét támogató KEYB program tárolására és futtatására. A felső memóriaterülettel kapcsolatban jelentkezik az első speciális fogalom, a felső memóriablokk (Upper Memory Block rövidítve: UMB). A felső memóriablokkok azok a memóriaterületek, amelyek a hardver eszközök működtetéséhez szükséges programok és a kiterjesztett memória működtetéséhez szükséges 64 Kb-otos oldalkeret elhelyezése után szabadon megmaradnak. Ezek a területek használhatók fel — a megfelelő DOS parancsok és bejegyzések végrehajtása után — az előbbieken felsorolt programok — pl. MIRROR, DOSKEY stb. — tárolására. Az elmondottakat a következő oldalon lévő ábrán szemléltetjük.

2. A második állomás az, amikor az 1024 Kb-othoz további 64 Kb-otos bővítést rendelünk. Ezt a területet, amely speciális DOS parancsokkal különleges célra használható, magas memóriaterületnek (High Memory Area, rövidítve: HMA; vagy csak egyszerűen HIGH.) nevezik. Ez tehát az 1024-es határ után következő első igazi kibővített memóriaterület, saját azonosító névvel.
3. A harmadik állomás a 80 286-os gép esetében maximálisan 16 Mb-ot, 80 386-os vagy 80 486-os gép esetében maximálisan 4 Gb-ot (Gigabajt) méretű kibővített memóriaterület, amely csak a megfelelő kezelőprogrammal használható. Ez a program biztosítja többek között azt is, hogy ugyanazt a memóriaterületet egyidejűleg különböző programok ne használják. A program neve HIMEM.SYS.

**Fontos megjegyzés:** A 5.0-s DOS-hoz tartozó kibővített memóriát kezelő HIMEM.SYS program kiszorítja azt a HIMEM programot, amelyet a 3.0-s Microsoft Windows verzió tartalmaz. Ha a felhasználó mind a két programmal rendelkezik, akkor csak az egyiket szabad használnia.

1024 Kbyte = 1 Mbyte

960 Kbyte

832 Kbyte

768 Kbyte

640 Kbyte

0 Kbyte

ROM BIOS
UMB, amely megfelelő DOS parancsok és segítségével bejegyzések és rezidens programok tárolására alkalmas
A kiterjesztett memória használatához szükséges 64 Kbyte-os oldalkeret
A video számára fenntartott hely
Hagyományos szabad memória. Ezt használhatják a programok.
Betöltött memóriarezidens programok. PI. ANSI.SYS, COMMAND.COM stb.
A DOS alapterülete

F  
E  
L  
S  
Ő  
  
M  
E  
M  
Ó  
R  
I  
A  
  
H  
A  
G  
Y  
O  
M  
Á  
N  
Y  
O  
S

Memó-  
ria

*A hagyományos memória és a felső memóriaterület*

## A kiterjesztett (expanded) memória

### Rövidítése: EMS

A kiterjesztett memória a számítógépbe egy további kártya elhelyezését és a kártyával együtt szállított, annak használatát biztosító program telepítését jelenti. Ebben az esetben tehát úgy képzelhetjük el a bővítést, mint a vízszintesen elhelyezkedő alaplappól kiálló (kiterjeszkedő) önálló memóriakártyát.

Azok a programok, amelyeket kiterjesztett memória használatára készítettek fel, nem képesek a kártyán levő információhoz egyszerre és közvetlenül hozzáférni. A kártyán levő információ ui. 16 Kb-át méretű összetartozó egységekben, ún. lapokban (page) helyezkedik el. Amikor az adott alkalmazói program a kiterjesztett memóriában tárolt információkra tart igényt, akkor a memóriát kezelő program a megfelelő oldalt bemásolja, másképpen fogalmazva leképezi a felső memóriaterületen elhelyezkedő és 64 Kb-át méretű ún. oldalkeretbe (l. az előző alfejezetet). A kiterjesztett memóriát használó alkalmazói program tehát csak az oldalkeretből tudja átvenni az adatokat. A szemközti két ábrán az elmondottakat szemléltetjük.

A kiterjesztett memóriának az az előnye, hogy a hagyományos PC-ben — 8086/8088 — is elhelyezhető. Hátránya viszont az, hogy csak adatok tárolására alkalmas és használata, a 16 Kb-átos lapozgatás miatt, viszonylag lassú. További hátrányt jelent, hogy azok az alkalmazói programok, amelyeket ezen memória használatára készítettek fel, nem minden esetben férnek össze a memóriakártyát „üzemeltető” meghajtóprogrammal.

A kiterjesztett memóriát korábban fejlesztették ki, és így hamarabb elterjedt, mint a kibővített memória. Sok olyan alkalmazói program van forgalomban, amely tudja használni. Ezért ha olyan 80 386-os vagy 80 486-os proceszorral rendelkező számítógépes környezetben dolgozunk, amely csak kibővített memóriával rendelkezik, akkor az 5.0-s DOS EMM386.EXE meghajtóprogramjával ez a memória kiterjesztett memóriaként működtethető.

Az 5.0-s DOS a következő három alapvető memóriatípus használatát teszi lehetővé:

- hagyományos
- kibővített (extended)
- kiterjesztett (expanded)

A hagyományos memória az a maximálisan 640 Kb-át méretű memória, amelyet minden PC tartalmaz.

A kibővített memória, amelynek maximális mérete 4 Gb-át lehet, a hagyományos memória folytatásaként képzelhető el. A memória használatához speciális meghajtóprogram (HIMEM.SYS) telepítésére van szükség.

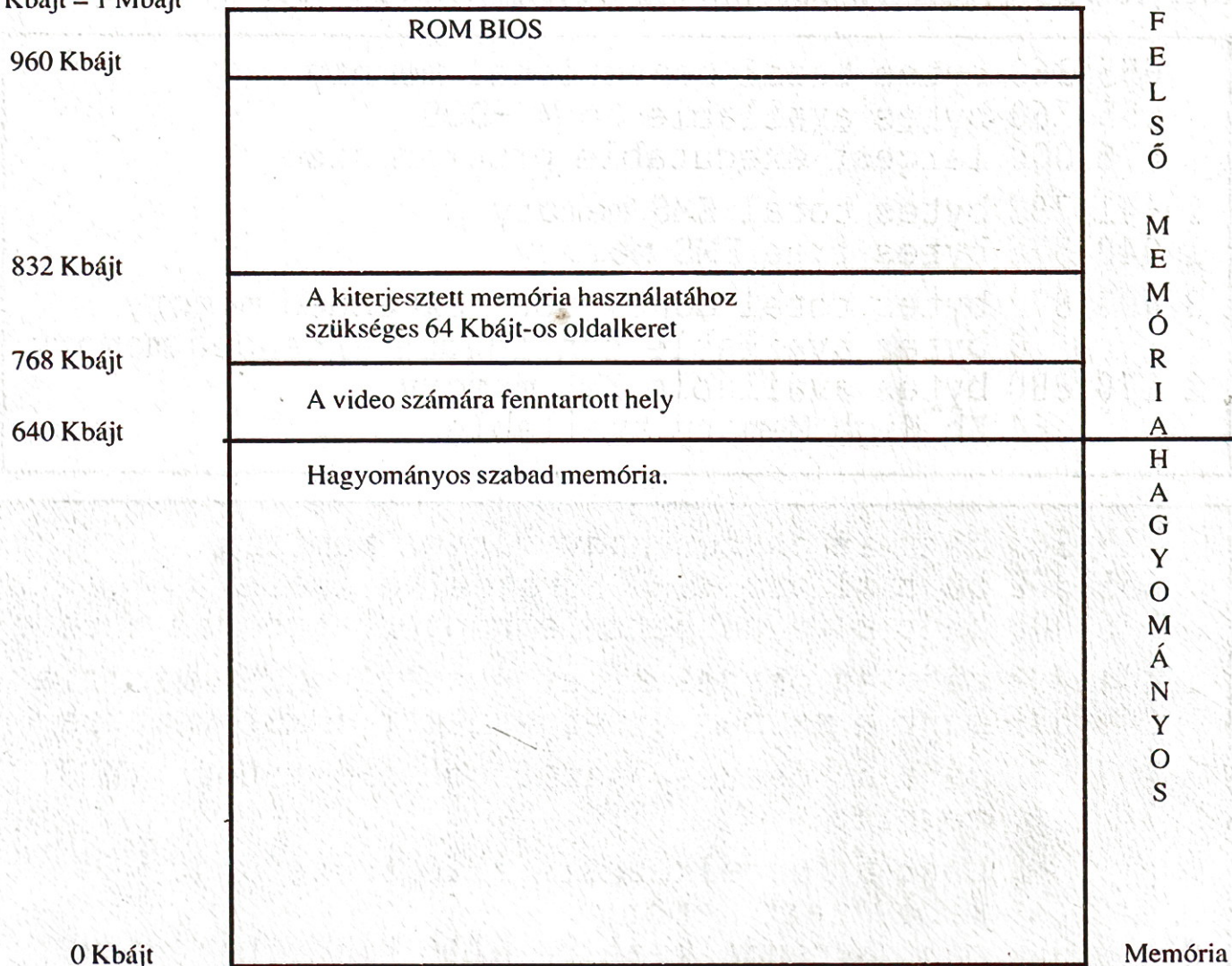
A kiterjesztett memóriát, amelynek maximális mérete 32 Mb-át, külön meghajtóprogrammal külön kártyán szállítják. Ez a memória bár a hagyományos PC-kben (8088/8086) is elhelyezhető, csak adatok tárolására alkalmas és csak 16 Kb-átos egységekben lapozgatva kezelhető.

n-ik oldal
"
"
"
5. oldal
4. oldal
3. oldal
2. oldal
1. oldal
0. oldal

Mindegyik oldal 16 Kbájt méretű és csak adatok tárolására alkalmas!

*A kiterjesztett memóriát kezelő program a memóriát, amelynek maximális mérete 32 Mbájt lehet, 16 Kbájtos lapokra osztja fel*

1024 Kbájt = 1 Mbájt



*A kiterjesztett memóriát kezelő szoftver a 16 Kbájtos lapokat 64 Kbájtos oldalkeretbe másolja be*

Most, hogy megismerkedtünk a témához tartozó fontos fogalmakkal, gyakorlásképpen nézzük meg, saját rendszerünk milyen memóriával vagy memóriákkal rendelkezik.

## *Az éppen aktuális memóriahasználat lekérdezése a MEM paranccsal*

<b>4.0</b>
------------

<b>KÜLSŐ</b>
--------------

A MEM parancs legegyszerűbb alakja az, amikor kapcsolók nélkül futtatjuk. Elsőként tehát így hajtsuk végre a parancsot!

C:\>MEM <ENTER>

melyre a következőhöz hasonló üzenet jelenik meg a képernyőnkön:

```

655 360 bytes total convetional memory
655 360 bytes available to MS-DOS
576 000 largest executable program size
1 441 792 bytes total EMS memory
1 048 576 bytes free EMS memory
3 407 872 bytes total contiguous extended memory
    0 bytes available contiguous extended memory
2 170 880 bytes available XMS memory
    64 Kb High Memory available

```

```

655 360 bájt az összes hagyományos memória
655 360 bájt áll az MS-DOS rendelkezésre
576 000 bájt a futtatható legnagyobb program mérete
1 441 792 bájt az összes kiterjesztett (EMS) memória
1 048 576 bájt a szabad kiterjesztett (EMS) memória
3 407 872 bájt az összes összefüggő kibővített (XMS)
    memória
    0 bájt a rendelkezésre álló összefüggő
    kibővített memória
2 170 880 bájt a rendelkezésre álló kibővített (XMS)
    memória
    64 Kbájt felső memóriaterület áll rendelkezésre

```

A MEM parancs tehát kapcsolók nélkül futtatva az adott rendszerben rendelkezésre álló memóriákról és azok aktuális használatáról nyújt információt, valamint kijelzi a legnagyobb futtatható program méretét is. A fenti üzenet természetesen más akkor, ha az adott rendszerben nincs EMS vagy XMS memória. A MEM paranccsal megjelenített előbbi alapadatokra mindenképpen szükségünk van, ha a rendszerünkben valamilyen memóriabeállítást akarunk végezni.

A MEM paranccsal három különböző kapcsoló használható, a /P-nek rövidíthető /PROGRAM, a /C-nek rövidíthető /CLASSIFY és a /D-nek rövidíthető /DEBUG, de egyszerre mindig csak egy alkalmazható. A /P kapcsoló hatására a memóriába éppen betöltött programok helyéről — (Address), nevéről (Name), méretéről (Size) és típusáról (Type) kapunk információt. Mielőtt a /P kapcsolót kipróbálnánk, a következő tartalmat tételezzük fel a CONFIG.SYS és az AUTOEXEC.BAT állományokban:

CONFIG.SYS:

```
FILES=40
BUFFERS=30
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE 1024 RAM
DOS=HIGH,UMB
DEVICE=C:\WINDOWS\MOUSE.SYS
BREAK ON
COUNTRY=036,,C:\DOS\COUNTRY.SYS
SHELL=COMMAND.COM /E:1024 /P
DEVICE=C:\DOS\DISPLAY.SYS CON=(EGA,,1)
```

AUTOEXEC.BAT:

```
@ECHO OFF
PROMPT $P$G
PATH C:;C:\DOS;C:\UTIL;C:\WINDOWS
SET TEMP=C:\WINDOWS\TEMP
DOSKEY
MIRROR
MODE CON CP PREP=((852)C:\DOS\EGA.CPI)
MODE CON CP SELECT=852
KEYB HU,,C:\DOS\KEYBOARD.SYS
```



C:\>MEM /P <ENTER>

parancs végrehajtása a következő képernyőt eredményezi:

Address	Name	Size	Type
000000		000400	Interrupt Vector
000400		000100	ROM Communication Area
000500		000200	DOS Communication Area
000700	IO	000A60	System Data
001160	MSDOS	0013D0	System Data
002530	IO	00D990	System Data
	HIMEM	0004A0	DEVICE=
	EMM386	0020D0	DEVICE=
	MOUSE	0042B0	DEVICE=
	DISPLAY	002060	DEVICE=
		000820	FILES=
		000100	FCBS=
		003E60	BUFFERS=
		0001C0	LASTDRIVE=
		000740	STACKS=
00FED0	MSDOS	000040	System Program
00FF20	COMMAND	000940	Program
010870	MSDOS	000040	- Free -
0108C0	COMMAND	000400	Environment
010CDO	MEM	000080	Environment
010D60	DOSKEY	001020	Program
011D90	KEYB	001840	Program
0135E0	MEM	0176F0	Program
02ACE0	MSDOS	075300	- Free -
09FFF0	SYSTEM	028010	System Program
0C8010	MSDOS	003FC0	- Free -
0CBFE0	SYSTEM	002020	System Program
0CE010	MSDOS	001FE0	- Free -
655360 bytes total conventional memory			
655360 bytes available to MS-DOS			
576000 largest executable program size			

Address	Name	Size	Type
	1441792 bytes total EMS memory		
	1048576 bytes free EMS memory		
	3407872 bytes total contiguous extended memory		
	0 bytes available contiguous extended memory		
	2170880 bytes available XMS memory		
	64 Kb High Memory available		

A memóriahelyek és a méretek hexadecimális számrendszerben vannak. Láthatjuk, hogy ebben az esetben a korábbiakban már lekérdezett alapinformációkon kívül a hagyományos memóriába betöltött programok és bejegyzések helyéről és terjedelméről is részletes információt kapunk. A Name, vagyis a programok nevét, ill. a Type vagyis a típus oszlopot tekintve érdemes megfigyelni, hogyan követik egymást a DOS áapterületét képező ún. megszakításvektor — Interrupt Vector —, a ROM és a DOS kommunikációs területek, majd a rejtett állományokként szereplő IO.SYS és MSDOS.SYS állományok. Ezután következnek a CONFIG.SYS állomány éppen aktuális tartalmának megfelelően a meghajtóprogramok és egyéb bejegyzések, majd az AUTOEXEC.BAT állományban előírt és betöltött memóriarezidens programok (pl. MOUSE vagy a DOSKEY).

A MEM parancsot kapcsolók nélkül futtatva az adott rendszerben rendelkezésre álló memóriákról és azok éppen aktuális használatáról kapunk információt, továbbá megtudjuk a legnagyobb futtatható program méretét is. A paranccsal három különböző kapcsoló használható, a /P-nek rövidíthető /PROGRAM, a /C-nek rövidíthető /CLASSIFY és a /D-nek rövidíthető /DEBUG, de egyszerre mindig csak egy alkalmazható. A /P kapcsoló hatására a memóriába éppen betöltött programok helyéről, nevéről, méretéről és típusáról kapunk információt. A /C kapcsoló a hagyományos memóriába és a felső memóriaterületbe betöltött programok méretét és a használatuk módját jelzi ki. A /D kapcsoló az éppen betöltött programok és meghajtóprogramok állapotáról ad információt.

## *Memóriakezelés és sebességnövelés*

Az 5.0-s DOS több olyan programot és bejegyzést tartalmaz, amelyekkel nemcsak a kibővített memória vehető használatba, hanem amelyekkel a rendszer működése gyorsítható, ill. a rendelkezésre álló hagyományos memóriában a futtatható programok számára nagyobb hely biztosítható.

Tegyük fel, hogy a MEM parancs végrehajtásával megállapítottuk, hogy a számítógépben kibővített memória van. Akkor elsőként ennek használatát kell biztosítanunk a HIMEM.SYS memóriakezelő program telepítésével. A művelet elvégzéséhez, ha a HIMEM.SYS program a C:\DOS alkönyvtárban van, a következő bejegyzést szükséges elhelyezni a CONFIG.SYS állományban:

```
DEVICE=C:\DOS\HIMEM.SYS
```

**Fontos megjegyzés:** *Mivel ez a program a kibővített memória kezelését biztosítja, a bejegyzés helye ezért nem tetszőleges! Meg kell előznie ui. minden olyan programot, amely a kibővített memóriára vonatkozik, vagy ezt használja. Ha a számítógépben kibővített memória áll a rendelkezésünkre, akkor a fenti bejegyzést a CONFIG.SYS állomány legelején kell és érdemes elhelyezni.*

A kibővített memória használatba vétele után a további bejegyzéseket, parancsokat mindig annak figyelembevételével kell megadni, hogy mi a célunk:

1. a rendelkezésre álló hagyományos memóriában minél nagyobb helyet biztosítani;
2. a rendszer működését gyorsítani.

A két szempont bizonyos fokig egymással szemben áll. Ahhoz ui., hogy a rendszerünket gyorsíthassuk, olyan programokat kell futtatnunk, amelyek a hagyományos memóriából foglalnak el kisebb vagy nagyobb helyet. A következőkben a DOS új szolgáltatásainak megismerésében ezt a két szempontot vesszük alapul.

## ***Lehetőségek a rendelkezésre álló hagyományos memória jobb kihasználására***

Valamennyi program közös tulajdonsága az, hogy a futásukhoz hagyományos memóriára van szükség. Már tudjuk, hogy a DOS a rendszerindítás után a hagyományos memória alsó részébe helyezi el magát. E fölött helyezkednek el a CONFIG.SYS állomány bejegyzéseinek megfelelően a meghajtóprogramok és memóriarezidens programok, valamint az AUTOEXEC.BAT állomány előírásai szerint betöltött olyan egyéb memóriarezidens programok, mint pl. a DOSKEY. Ami tehát ezen programok betöltése után marad, csak az a hely használható fel a különböző alkalmazói programok végrehajtásához. Ha viszont ez kevésnek bizonyul, akkor a következőképpen szabadíthatunk fel helyet a hagyományos memóriában:

1. A DOS futtatása a kibővített memóriában.
2. A mindenképpen szükséges memóriarezidens programok és eszköz-meghajtó programok futtatása a felső memóriaterületben. (80 386-os vagy 80 486-os gép esetén.)
3. A CONFIG.SYS-ben és AUTOEXEC.BAT-ban az adott helyzetben, nem feltétlenül szükséges sorok letiltása a REM paranccsal, ill. minden esetben a lehető legtakarékosabb paraméterezés.

Ezek után lássuk részletesen a gyakorlati teendőket!

### **A DOS futtatása a kibővített memóriában**

Ha kibővített memóriával is rendelkezünk, akkor annak első 64 Kb-átja, az ún. magas memóriaterület (High Memory Area) — felhasználható arra, hogy a DOS-t nagyrészt ide betöltve nagyobb hagyományos memóriaterületet nyerjünk. A művelethez a következő bejegyzéseket kell a CONFIG.SYS állományban elhelyezni:

```
DEVICE=C:\DOS\HIMEM.SYS  
DOS=HIGH
```

**Fontos megjegyzések:**

1. *Az első bejegyzés a kibővített memória használatát teszi lehetővé, ezért meg kell előznie minden olyan bejegyzést, amely erre a memóriára vonatkozik, így a DOS-t a kibővített memóriában elhelyező DOS=HIGH bejegyzést is.*
2. *Ahhoz, hogy a DOS=bejegyzés érvénybe lépjen, a rendszert újból kell indítanunk.*

**A felső memóriaterület használata****Az eszközmeghajtó programok futtatása a felső memóriablokkokban**

80 386-os, ill. 80 486-os számítógépek esetén az 5.0-s DOS DEVICEHIGH=bejegyzésének, ill. LOADHIGH parancsának felhasználásával lehetővé válik, hogy meghajtóprogramokat, ill. memóriarezidens programokat az UMB-ben hajtsunk végre. Mielőtt azonban hozzákezdünk a részletek tárgyalásához, már előzetesen meg kell jegyeznünk, hogy vannak olyan programok, amelyek a felső memóriaterületen nem futtathatók. Melyek ezek? A választ csak úgy adhatjuk meg, ha kipróbáljuk. Ha az adott program a felső memóriaterületre betöltve nem futtatható, akkor a rendszer vagy egyáltalán el sem indul, vagy pedig az adott program végrehajtása a számítógép működését megbénítja. Ezek tudatában tehát csak a következő szempontok figyelembevételével, ill. betartásával kezdünk hozzá a programok és bejegyzések felső memóriablokkokba telepítéséhez:

1. Kiindulásként a számítógép tökéletesen kell hogy működjön, különös tekintettel a benne elhelyezett kiterjesztett, vagy kibővített memóriára.
2. A DOS telepítése megfelelő legyen. (Semmilyen hibáüzenet, semmilyen felesleges bejegyzés vagy parancs futtatása nem engedhető meg.)
3. Valamennyi meghajtóprogram és memóriarezidens program a már meglévő állapotában helyesen működjön.
4. Készítsünk rendszerlemezt a FORMAT /S parancs végrehajtásával arra az esetre, ha valamilyen hiba következtében a rendszerünk megbénulna.
5. Készítsünk lemezmasolatot az AUTOEXEC.BAT és a CONFIG.SYS állományról.
6. Egyszerre mindig csak egy eszközmeghajtót vagy memóriarezidens programot helyezünk el a felső memóriaterületre, és csak ha már meggyőződünk róla, hogy rendben fut, akkor próbálkozzunk újabb program áthelyezésével.

Most pedig azokat az előkészítő műveleteket tárgyaljuk, amelyek ahhoz szükségesek, hogy a DOS hozzáférhessen a felső memóriaterülethez. Másként fogalmazva, hogy a hagyományos és a felső memóriaterület között kapcsolatot teremtsen, valamint ezt a területet kezelni tudja. Ezen előkészítő műveletek a következő bejegyzések elhelyezését igénylik a CONFIG.SYS állományban:

**Fontos megjegyzés:** A bejegyzéseket az itt megadott sorrendben kell egymás után elhelyezni.

### 1. A kibővített memória használatbavétele

```
DEVICE=C:\DOS\HIMEM.SYS
```

Ha tehát eddig ezt a bejegyzést nem tartalmazza a CONFIG.SYS állomány, akkor most írjuk be.

### 2. Kapcsolatteremtés a hagyományos és a felső memóriaterület között.

```
DOS=UMB
```

vagy

```
DOS=HIGH, UMB
```

vagy

```
DOS=LOW, UMB
```

Mint láthatjuk, a művelet eszköze a már előzőekben tárgyalt DOS=bejegyzés, most azonban más a paraméterezése. A DOS=bejegyzés általános alakja ui. a következő:

```
DOS=HIGH|LOW [, UMB|NOUMB]
```

```
DOS=[HIGH, |LOW, ]UMB|NOUMB
```

A szintaxisban a függőleges elválasztóvonal azt jelenti, hogy a két paraméter közül csak az egyiket lehet választani, a szögletes zárójel pedig — mint szokásosan — azt, hogy az adott paraméter megadható. A HIGH|LOW paraméterpár a DOS helyét írja elő. A HIGH hatására a DOS a kibővített, míg a LOW hatására a hagyományos memóriába tölti magát. Az UMB|NOUMB paraméterpár viszont arra utal, hogy a DOS létesítsen-e kapcsolatot a hagyományos és a felső memóriaterület között. Ha az UMB paramétert használjuk, akkor ez a kapcsolat létrejön. Az alapértelmezés viszont a NOUMB, amely ezt a kapcsolatot megszakítja. Tehát ha azt akarjuk, hogy a DOS a kibővített memóriaterületre helyezze magát, valamint hogy a két memória között a kapcsolat is létre-

jöjjön, akkor a következő bejegyzést kell beírunk a CONFIG.SYS állományba:

```
DOS=HIGH,UMB
```

### 3. A felső memóriaterület kezelése

Ahhoz, hogy a felső memóriaterület használatbavételét elérjük, a következő bejegyzésre van szükség a CONFIG.SYS állományban:

```
DEVICE=C:\DOS\EMM386.EXE NOEMS
```

Lehet, hogy most feltesszük magunkban a kérdést, mit keres itt az EMM386.EXE memóriakezelő program? Eddig még nem említettük, azonban tudnunk kell, hogy ez a program kettős célt szolgál. Ugyanis nemcsak kiterjesztett memóriát tud a kibővített memóriában szimulálni, hanem 80386-os vagy a 80486-os számítógép esetében a felső memóriaterületet is képes kezelni. E szolgáltatás igénybevételéhez azonban a NOEMS vagy a RAM kapcsoló egyikét meg kell adni. A NOEMS kapcsolót alkalmazva a felső memóriaterület kezelése érhető el, míg a RAM kapcsolót előírva a program a felső memóriaterületet is kezeli, de kiterjesztett memóriát is szimulál. Például a Microsoft Windows használata esetén a RAM kapcsolóra van szükség.

Az előkészítő műveletek tárgyalása után pedig lássuk, hogy milyen bejegyzés hatására helyezi el a DOS pl. az ANSI.SYS meghajtóprogramot a felső memóriaterületre:

```
DEVICEHIGH=C:\DOS\ANSI.SYS
```

A meghajtóprogramok felső memóriaterületre helyezéséhez tehát a DEVICEHIGH= bejegyzés használata szükséges.

Itt kell megjegyeznünk még a következőket is:

1. Amennyiben a megadott eszközmeghajtó program számára nem áll rendelkezésre elegendő hely a felső memóriablokkban, akkor a rendszer azt a hagyományos memóriába tölti be.
2. Ha a DEVICEHIGH= bejegyzést nem előzi meg a CONFIG.SYS állományban a DOS=UMB bejegyzés, akkor a rendszer a meghajtóprogramokat a hagyományos memóriába telepíti, mintha csak a DEVICE= bejegyzést használtuk volna.
3. A DOS=HIGH,UMB bejegyzés megadása előtt a DEVICE=C:\DOS\HIMEM.SYS bejegyzésnek szerepelnie kell.

4. A HIMEM.SYS és az EMM386.EXE memóriakezelő programokra vonatkozó DEVICE= bejegyzéseknek a többi DEVICE= bejegyzést meg kell előzniük.

### A memóriarezidens programok futtatása a felső memóriablokkban

Ahhoz, hogy valamely programot a felső memóriaterületen tudjunk futtatni, az LH-nak rövidíthető LOADHIGH belső DOS parancsot kell használnunk. A parancs általános alakja a következő:

```
LOADHIGH[meghajtó_betűjele:] [elérési_út] filenév.kit [paraméterek]
```

Paraméterként a program futtatásához szükséges és egyébként a program neve után a parancssorban megadott paraméterek írhatók elő. Ahhoz, hogy a DOS a felső memóriaterületen elhelyezett programot végre tudja hajtani, ugyanazokra az előkészítő műveletekre van szüksége, mint az előzőekben tárgyalt DEVICEHIGH= bejegyzés esetében, vagyis:

```
DEVICE=C:\DOS\HIMEM.SYS  
DOS=UMB  
DEVICE=C:\DOS\EMM386.EXE NOEMS
```

Mivel a LOADHIGH belső parancs, ezért ezt vagy a parancssorból, vagy az AUTOEXEC.BAT állományból lehet futtatni. A felső memóriaterületre azokat a programokat érdemes elhelyezni, amelyekre minden rendszerindítás után szükségünk van, de nem akarunk ezekre a hagyományos memóriából helyet „pazarolni”. A következőkben néhány ilyen példát, ötletet mutatunk be:

```
LOADHIGH C:\DOS\DOSKEY.COM  
LOADHIGH C:\DOS\MIRROR.COM  
LOADHIGH C:\DOS\NLSFUNC.EXE  
LOADHIGH C:\DOS\KEYB.COM
```

Végezetül tekintsük át, hogy a CONFIG.SYS állományban milyen sorrendben kell szerepelniük a memóriákat kezelő programoknak:

1. HIMEM.SYS
2. Ha a rendszerünk kiterjesztett memóriával rendelkezik, akkor az a memóriát kezelő program, amelyet a kártyával együtt kaptunk.



3. Minden olyan eszközmeghajtó program, amely a kibővített memóriát használja (pl. DOS=HIGH,UMB).
4. EMM386.EXE

### Fontos megjegyzések!

- a) *Ha az EMM386.EXE programot kiterjesztett memória szimulálására, valamint a felső memóriaterület kezeléséhez akarjuk felhasználni, akkor a RAM kapcsolót kell a program neve után megadnunk.*
  - b) *Ha az EMM386.EXE programot csak a felső memóriaterület kezelésére alkalmazzuk, akkor a NOEMS kapcsolóra van szükség.*
  - c) *Ne szimuláljunk az EMM386.EXE programmal kiterjesztett memóriát akkor, ha valódi kiterjesztett memóriával rendelkezünk.*
5. Minden olyan eszközmeghajtó, amely a kiterjesztett memóriát használja.
  6. Minden olyan eszközmeghajtó, amely a felső memóriaterületet használja.
  7. A többi eszközmeghajtó program és egyéb bejegyzés.

Ez a kötelező és a teljes sorrend. A programok és bejegyzések közül természetesen mindig csak az adott helyzetnek és igényeknek megfelelőket szabad alkalmazni.

## Lehetőségek a CONFIG.SYS és az AUTOEXEC.BAT állomány „karcsúsítására”

Mivel most az a célunk, hogy a lehetőségeken belül minél nagyobb legyen a programok futtatásához rendelkezésre álló hagyományos memóriaterület, az előbbi pontokban tárgyaltakon kívül nem szabad figyelmen kívül hagynunk azokat a lehetőségeket sem, amelyek a CONFIG.SYS és az AUTOEXEC.BAT állomány karcsúsítása kínál.

### Takarékos megoldások a CONFIG.SYS állományban

A következőkben pontokba szedve azokat a szempontokat foglaltuk össze, amelyek figyelembevételével a CONFIG.SYS állomány „nagytakarítása” végezhető el.

1. Tekintsük át a bejegyzéseket, és csak azokat hagyjuk meg, amelyekre az adott helyzetben ténylegesen szükségünk van. A feleslegesnek ítélt bejegyzéseket tiltsuk le a REM paranccsal. (A 6. Fejezetben a REM-et részletesen tárgyaltuk.)

2. A rendszer gyorsabb működését szolgáló és a következő alfejezetben tárgyalt SMARTDRV és RAMDRIVE optimalizáló programokra vonatkozó DEVICE= bejegyzéseket csak akkor hagyjuk meg, ha ezekre a gyorsítási lehetőségekre ténylegesen szükségünk van.
3. Csökkentsük a BUFFERS= bejegyzésnél beállított értéket 10—15-re, ha csak nem alkalmazzuk a SMARTDRV programot. (A rendszer minden egyes pufferhez kb. 500 bájt foglalt le a hagyományos memóriából!)
4. A STACKS= bejegyzést a 6. Fejezetben tárgyaltak figyelembevételével a következőként rögzítsük:

```
STACKS=0,0
```

5. Csökkentsük a LASTDRIVE= bejegyzéshez beállított értéket J-re vagy K-ra, ill. olyan betűre, amelyre még ténylegesen szükségünk van a meghajtók betűjelének előírásakor. Ne állítsunk be automatikusan Z-t, mivel minden egyes logikai meghajtó 100 további bájt lefoglalását jelenti a hagyományos memóriából.
6. Ha a CONFIG.SYS állomány tartalmazza az FCBS= bejegyzést, akkor az legyen a következő:  

```
FCBS=1
```
7. A DEVICEHIGH= bejegyzés felhasználásával helyezzünk át meghajtóprogramokat a felső memóriaterületre!

### *Takarékos megoldások az AUTOEXEC.BAT állományban*

Lássuk, mit tehetünk annak érdekében, hogy az AUTOEXEC.BAT állomány is a lehető legtömörebb legyen:

1. Nézzük át alaposan a parancsokat, és csak azokat tartsuk meg, amelyekre az adott munkánál ténylegesen szükségünk van. A feleslegesnek ítélt parancsok végrehajtását tiltsuk le a REM utasítással. Például a MOUSE.COM parancsot csak akkor futtassuk, ha az egeret ténylegesen használnunk kell.
2. A LOADHIGH paranccsal tegyük át a memóriarezidens programokat a felső memóriaterületre!

**Az 5.0-s DOS a hagyományos memória jobb kihasználására a következő lehetőségeket kínálja:**

1. A DOS futtatása a kibővített memóriában a DOS= bejegyzés felhasználásával.
2. A mindenképpen szükséges memóriarezidens programok és eszköz-meghajtó programok futtatása a felső memóriaterületben a LOAD-HIGH program, ill. a DEVICEHIGH= bejegyzés alkalmazásával.
3. A CONFIG.SYS-ben és AUTOEXEC.BAT-ban az adott helyzetben nem feltétlenül szükséges sorok letiltása a REM paranccsal, ill. minden esetben a lehető legtakarékosabb paraméterezés.

## *A számítógép működésének gyorsítása*

Már a korábbiakban is említettük, hogy a számítógép általános működéséhez képest több nagyságrenddel lassabb azon műveletek végrehajtása, amelyek a lemez — legyen az akár a merevlemez is — írásával és olvasásával kapcsolatosak. Ha tehát olyan programokat — pl. adatbázis-kezelő rendszert — használunk, amelyek nagyon gyakran írják, ill. olvassák a lemezt, mindenképpen érdemes igénybe venni a DOS kínálta gyorsítási lehetőségeket még akkor is, ha ennek következtében a hagyományos memóriából nagyobb helyet foglalunk el. Ebben az alfejezetben ezeket a lehetőségeket és az ide kapcsolódó fogalmakat tekintjük át.

### **A BUFFERS= bejegyzés használata**

Bár a BUFFERS= bejegyzést már az EGYSZERŰEN DOS c. könyvben is tárgyaltuk, mint a számítógép működésének gyorsítását szolgáló egyik legegyszerűbb eszközt, most is megemlítjük.

A CONFIG.SYS állományban elhelyezendő BUFFERS= bejegyzéssel — mint tudjuk — azon pufferek (átmeneti vagy ideiglenes tárolóhelyek) száma írható elő, amelyeket a DOS az állományműveletek végrehajtásához fenn tart. A bejegyzéssel előírható értéknek az 1—99 tartományba kell esnie, meg kell azonban jegyeznünk, hogy a beállítandó optimális érték mindig az adott

számítógépes környezet merevlemezének méretétől függ. Ugyanis pl. egy 40 Mbájtos merevlemez esetében hiába írunk elő 50-es értéket. Ezzel csak feleslegesen foglalunk le több hagyományos memóriát, a rendszerünk nem lesz gyorsabb. A következő táblázatban — a beállítás megkönnyítésére — adott merevlemez méretekhez a leghatékonyabban használható pufferméreteket adjuk meg:

Merevlemez mérete	Pufferek száma
40 Mbájtos és ennél kisebb	20
40-től 79 Mbájtig	30
80-tól 119 Mbájtig	40
120 Mbájtos felett	50

Tehát ha a számítógépben a merevlemez 20 Mbájtos, akkor a következő bejegyzésre van szükség:

```
BUFFERS=20
```

**Megjegyzés:** Ha a CONFIG.SYS állományban nem helyezzük el a BUFFERS= bejegyzést, akkor a DOS az általa kiszámított alapértelmezés szerinti értékkel dolgozik. Ezen alapértelmezés szerinti érték meghatározásánál azonban a DOS csak a hagyományos memória méretét veszi figyelembe. A merevlemez méretével nem számol. Ezért érdemes a DOS alapértelmezés-ként kiszámított minimális értékét a fenti táblázatnak megfelelően felülírni.

## A FASTOPEN program használata

### 3.3

### KÜLSŐ

A 6. Fejezetben már említett FASTOPEN program segítségével a gyakran használt állományok megnyitásához szükséges idő csökkenthető, mivel a program a futtatása után a már megnyitott állomány nevét és a hozzá vezető útvonalat egy speciális átmeneti tárolóban, a memóriában tartja. Így az adott állomány következő megnyitásakor már az itt — vagyis a hagyományos memóriában — elhelyezett adatokkal tud dolgozni, így az állomány megnyitásának műveletét gyorsabban tudja elvégezni. A FASTOPEN program tehát olyan alkalmazói programok (pl. adatbázis-kezelő programok) használata

esetén nyújt hasznos szolgálatot, amelyek mindig ugyanazokat az állományokat, azonban azokat gyakran nyitják meg.

A FASTOPEN program szintaxisa a következő:

FASTOPEN meghajtó\_betűjele:

[[=]n][meghajtó\_betűjele:[[=]n][...]][/X]

vagy

```
INSTALL= [[meghajtó_betűjele:]elérési_út]FASTOPEN.EXE
meghajtó_betűjele:[[=]n][meghajtó_betűjele:[[=]n][...]]
[/X]
```

A FASTOPEN.EXE után álló meghajtó betűjele annak a merevlemez meghajtónak a betűjele, amelyre a FASTOPEN programmal a nyilvántartást el akarjuk érni. Mint a szintaxisból látható, egynél több merevlemez meghajtó neve is előírható.

Az n FASTOPEN által nyilvántartható állományok maximális számát jelenti. Alapértelmezés: 48. Megadható érték: 10-től 999-ig. Az /X kapcsoló előírására a program a nevek és az útvonalak átmeneti tárolóját a kiterjesztett memóriában hozza létre.

A FASTOPEN program végrehajtására a következő négy lehetőségből választhatunk:

- futtatás a parancssorból,
- végrehajtás az AUTOEXEC.BAT állományból,
- az INSTALL= bejegyzés felhasználásával futtatás a CONFIG.SYS állományból,
- 5.0-s DOS esetén: a FASTOPEN.EXE program futtatása a felső memóriaterületen a LOADHIGH parancs felhasználásával.

Lássunk néhány példát is az elmondottakra!

1. A FASTOPEN futtatása a CONFIG.SYS állományból:

```
INSTALL=C:\DOS\FASTOPEN.EXE C:=40
```

2. Végrehajtás az AUTOEXEC.BAT állományból vagy a parancssorból:

```
LOADHIGH C:\DOS\FASTOPEN.EXE C:=40
```

**Fontos megjegyzés:** A FASTOPEN program által nyilvántartott állományok számát, vagyis az n értékét mindig a merevlemez méretéhez kell igazítani, mégpedig a következőképpen:

*Mbájtok száma = nyilvántartott állományok száma.*

*Ennek értelmében tehát 40 Mbájtos merevlemez esetében az  $n$  értékét 40-re célszerű rögzíteni.*

A FASTOPEN program futtatásával csökkenthető a gyakran használt állományok megnyitásához szükséges idő, mivel a program a már megnyitott állományok helyét a memóriában az erre a célra létesített speciális pufferban tartja nyilván.

## Gyorsítótár — Cache memória — létesítése a SMARTDRV.SYS meghajtóprogrammal

### 4.0

A gyorsítótárat vagy az ún. cache memóriát létesítő SMARTDRV program használata olyan számítógépek esetében jöhet számításba, amelyek merevlemezrel és kiterjesztett vagy kibővített memóriával rendelkeznek. A gyorsítótár segítségével jelentősen csökkenthető a merevlemezre való írásra, ill. olvasásra fordított idő.

A SMARTDRV program a telepítése után a kiterjesztett vagy a kibővített memóriából bizonyos területet a saját céljaira adatok tárolására foglal le. Ezt a memóriát nevezik cache memóriának vagy gyorsítótárnak. Ha a program végrehajtása után a merevlemezre írási vagy olvasási műveletet hajtunk végre, akkor az információ mozgatásához a SMARTDRV ezt a memóriát használja. A SMARTDRV program nagy előnye, hogy a cache memória tartalmának lemezre írásáról automatikusan gondoskodik, így nem kell attól félnünk, hogy a számítógép kikapcsolásakor az adatainkat elveszítjük.

A SMARTDRV program telepítését a DEVICE= bejegyzés felhasználásával a CONFIG.SYS állományban kell elvégezni. A bejegyzés általános alakja a következő:

```
DEVICE=[meghajtó_betűjele:]  
[elérési_út]SMARTDRV.SYS[i_méret][m_méret][/A]
```

Az `i_méret` paraméterrel a cache memória mérete állítható be. A megadható érték 128—8192 Kbájt lehet, az optimális tartomány azonban: 256—2048 Kbájt. 256 Kbájtnál kisebbet — egyébként ez az alapértelmezés is — nincs értelme beállítani, mert ennél kisebb cache memória esetén a SMARTDRV program már nem tud hatékonyan dolgozni. Viszont annak ellenére, hogy 8 Mbájt a beállítható maximális érték, 2 Mbájt az a felső optimális határ, amelynél nagyobb érték előírása már nem jelent számottevő sebességnövelést.

Az `m_méret` paraméterrel a cache memória minimális mérete adható meg. Erre a paraméterre csak olyan esetben van szükség, ha a számítógépben futtatott programról tudjuk, hogy a cache memória méretét képes nullára is beállítani (ilyen pl. a Windows).

A `/A` kapcsoló hatására a SMARTDRV.SYS a cache memóriát a kiterjesztett memóriában létesíti. Ha ezt a kapcsolót nem adjuk meg, akkor a gyorsítótár a kibővített memóriában jön létre.

### Fontos megjegyzések:

1. *Az 5.0-s DOS-ban szereplő SMARTDRV program hatálytalanítja azt a SMARTDRV programot, amelyet a Microsoft Windows 3.0-s verziója tartalmaz. Ha tehát mind a két programmal rendelkezünk, akkor csak az egyiket szabad használni.*
2. *Mivel a SMARTDRV program a cache memóriát a kiterjesztett, ill. a kibővített memóriában hozza létre, a memóriák használatához szükséges előkészítő műveleteket még a SMARTDRV program telepítése előtt el kell végezni.*
3. *A SMARTDRV program telepítése esetén a CONFIG.SYS állományban a `BUFFERS=` bejegyzést a következő értékre rögzítsük:*

```
BUFFERS=3
```

Lássunk ezek után néhány példát a SMARTDRV.SYS program telepítésére!

1.

```
DEVICE=SMARTDRV.SYS
```

A bejegyzés azt feltételezi, hogy a SMARTDRV.SYS program az éppen aktuális meghajtó gyökérkatalógusában van. Mivel sem az `i_méret` paramétert sem az `/A` kapcsolót nem alkalmaztuk, ezért a fenti bejegyzés hatására az alapértelmezés szerinti 256 Kbájt cache memória a kibővített memóriában létesül.

A DOS a memória létrehozását a következő üzenettel jelzi:

```
Microsoft SMARTDrive Disk Cache version 3.13  
Cache size: 256K in Extended Memory  
Room for 30 tracks of 17 sectors each  
Minimum cache size will be 0K
```

Microsoft SMARTDrive Disk Cache program 3.13-as verzió  
A cache memória mérete: 256K, amely a kibővített memóriában létesül.  
30 sáv – mindegyikben 17 szektor – számára van hely.  
A cache memória minimális mérete 0K.

2.

```
DEVICE=C:\DOS\SMARTDRV.SYS 1024
```

A bejegyzés hatására a DOS a SMARTDRV.SYS programot a C: meghajtó DOS alkönyvtárában keresi. A program a maximálisan 1024 Kb-ot méretű cache memória számára a kiterjesztett memóriában foglalja le a helyet. A memória létrehozását a következő üzenet jelzi:

```
Microsoft SMARTDrive Disk Cache version 3.13  
Cache size: 1024K in Extended Memory  
Room for 120 tracks of 17 sectors each  
Minimum cache size will be 0K
```

3.

```
DEVICE=C:\DOS\SMARTDRV.SYS 1024 /A
```

A bejegyzés az /A kapcsoló miatt a 1024 Kb-ot méretű cache memóriát a kiterjesztett memóriában hozza létre.

**Fontos megjegyzés:** Ha a számítógépen csak kibővített memória van, amely kiterjesztett memóriaként az EMM386.EXE programmal működtethető, akkor ne használjuk az /A kapcsolót, vagyis ne létesítsünk cache memóriát a szimulált kiterjesztett memóriában. Bár a cache memória ebben az esetben is létrejön, azonban a kiterjesztett memória lapozgatós kezelése miatt nem csökken lényegesen a merevlemezzel kapcsolatos írási és olvasási műveletekre fordított idő.



A SMARTDRV olyan optimalizáló program, amely a kiterjesztett vagy kibővített memóriában létrehozott átmeneti tár — cache memória — létesítésével jelentősen tudja csökkenteni azt az időt, amely a merevlemez írásához és olvasásához szükséges. A SMARTDRV program nagy előnye, hogy a cache memória tartalmának lemezre írásáról automatikusan gondoskodik, így a felhasználónak nem kell tartania a számítógép kikapcsolásakor az ott tárolt adatok elvesztésétől.

## RAM memória létesítése a RAMDRIVE.SYS meghajtóprogrammal

### 5.0

Ha sok szabad memóriánk van, legyen az hagyományos, kiterjesztett vagy kibővített memória, és sok ideiglenes állománnyal dolgozunk, akkor érdemes igénybe venni a DOS kínálta azon lehetőséget, hogy a memória egy részét lemezegységként működtetve, a munkaállományokat itt tároljuk. A memória bizonyos részének ilyen célú felhasználását RAM meghajtónak vagy virtuális lemeznek nevezik. Mivel a meghajtó a memóriában jön létre, a következő két fontos tulajdonsága van:

- sokkal gyorsabban működik, mint a fizikai lemezmeghajtók,
- tartalma a számítógép kikapcsolása után elvész.

A virtuális lemez telepítéséhez a RAMDRIVE.SYS meghajtóprogramot a DEVICE= bejegyzés felhasználásával a CONFIG.SYS állományban kell megadnunk. A bejegyzés általános alakja a következő:

```
DEVICE=[meghajtó_betűjele:][elérési_út]RAMDRIVE.SYS
[Lemez_méret Szektorméret]
[Katalógusbejegyzések_száma][ /E! /A]
```

A lemezméret paraméterrel a virtuális lemez mérete adható meg Kbájtban. Az alapértelmezés 64 Kbájt. Lehetséges értékek: 16—4096 Kbájt.

A szektorméret paraméter a virtuális lemez szektorainak bájtban megadott méretét jelenti. Alapértelmezés: 512. Megadható értékek: 128, 256, 512. Ha a szektorméretet előírjuk, akkor a lemezméret paramétert is meg kell adnunk. Bár a szektorméret értéke a fentiek szerint választható meg, javasolt azonban az alapértelmezés szerinti érték használata.

A katalógusbejegyzések száma paraméterrel a virtuális lemez gyökérkönyvtárában nyilvántartott állománynevek és könyvtárnevek együttes száma írható elő. Alapértelmezés: 64. Megadható érték: 2—1024. Ha a katalógusbejegyzések számát meg akarjuk adni, akkor az előző két paramétert is elő kell írunk.

A /E kapcsoló használatára akkor van szükség, ha a virtuális lemezt a kibővített memóriában akarjuk létrehozni.

A /A kapcsoló hatására a virtuális lemez a kiterjesztett memóriában jön létre.

### Megjegyzések:

- Mint a szintaxisból is látható, a két kapcsoló közül mindig csak az egyik alkalmazható. Ha nem adunk meg kapcsolót, akkor a DOS a virtuális lemezt a hagyományos memóriában létesíti.*
- A kibővített, ill. kiterjesztett memória használatbavételét a megfelelő meghajtóprogram(ok) telepítésével még a virtuális lemez létesítése előtt el kell végeznünk.*

Lássunk néhány példát a virtuális lemez létesítésére!

- Ha a RAMDRIVE.SYS meghajtóprogram a C:\DOS alkönyvtárban van és 64 Kb-ot — ez az alapértelmezés — méretű virtuális lemezt akarunk a hagyományos memóriában létesíteni, akkor a CONFIG.SYS állományt a következő bejegyzéssel kell kiegészítenünk:

```
DEVICE=C:\DOS\RAMDRIVE.SYS
```

Ahhoz, hogy a DOS a virtuális lemezt létrehozza, a rendszert újból kell indítanunk. Ekkor a DOS a következőhöz hasonló üzenettel jelzi a virtuális lemez létrejöttét:

```
Microsoft RAMDrive version 3.06 virtual disk D:  
Disk size: 64K  
Sector size: 512 bytes  
Allocation unit: 1 sectors  
Directory entries: 64
```

```
Microsoft RAMDrive verzió 3.06 A D: a virtuális lemez  
Lemez méret: 64K  
Szektor méret: 512 bájt  
Allokációs egység: 1 szektor  
Katalógusbejegyzések száma: 64
```

A meghajtó betűjeleként a DOS mindig a következő rendelkezésre álló karaktert, pl. a D-t használja. A virtuális lemez a hagyományos meghajtókkal azonos módon használható. Például a virtuális lemezen — D: meghajtó — tárolt állományok listázásához a következő parancs szükséges:

```
C:\>DIR D: <ENTER>
```

2. Ha a hagyományos memóriával takarékoskodni akarunk; ezért pl. 1024 Kbájt virtuális lemezt a kibővített memóriában kívánjuk létrehozni, akkor a következő bejegyzést kell alkalmaznunk:

```
DEVICE=C:\DOS\RAMDRIVE.SYS 1024 /E
```

3. Tegyük fel, hogy a 2048 Kbájt méretű virtuális lemezt a kiterjesztett memóriában akarjuk létrehozni, és a lemez gyökérkatalógusában a bejegyzések számát 1024-re akarjuk korlátozni. Ehhez a következő bejegyzés szükséges:

```
DEVICE=C:\DOS\RAMDRIVE.SYS 2048 512 1024 /A
```

A virtuális lemez, másképpen RAM meghajtó a hagyományos, kiterjesztett vagy kibővített memóriában létrehozható látszólagos lemezegység, amelynek tartalma a számítógép kikapcsolásakor elvész. Virtuális lemezt akkor érdemes létesíteni, ha viszonylag sok szabad memóriával rendelkezünk és az éppen futtatott program sok állományt használ. Ezen állományokat érdemes a virtuális lemezen tárolni, mivel így az azokat felhasználó alkalmazói program futása sokkal gyorsabbá válik.

## Ismétlés = Kulcs a sikerhez

A következő táblázatban a fejezetben tárgyalt bejegyzések és programok legfontosabb jellemzőit és célját tekintjük át. A programok megadásának sorrendje egyben azok előírásának, ill. futtatásának sorrendjét is jelenti.

Program-név	Elhelyezésének helye	Memória, amelyet felhasznál	Memória, amelyre hatással van	Célja
HIMEM.SYS	CONFIG.SYS	Kis mennyiségű hagyományos memória	Kibővített memória	A kibővített memória használatát teszi lehetővé. Megadja a lehetőséget hogy ott bizonyos programokat futtatva a hagyományos memóriát felszabadítsuk.
DOS=HIGH	CONFIG.SYS	Magas memóriaterület	Hagyományos memória	A DOS nagy részét a magas memóriaterületbe töltve a hagyományos memóriát felszabadítja.
DOS=HIGH, UBM	CONFIG.SYS	Magas memóriaterület	Hagyományos memória	Kapcsolatteremtés a hagyományos és a felső memóriaterület között, valamint a DOS nagy részének a magas memóriaterületre helyezése.
EMM386.EXE	CONFIG.SYS	Kibővített memória	Kibővített memória	Kiterjesztett memória szimulálása a kibővített memóriában.
EMM386.EXE NOEMS	CONFIG.SYS	Kibővített memória	Felső memória	A felső memóriaterület kezelése.
EMM386.EXE RAM	CONFIG.SYS	Kibővített memória	Felső memória	A felső memóriaterület kezelése és a kiterjesztett memória szimulálása a kibővített memóriában.
DEVICEHIGH=	CONFIG.SYS	Felső memória	Hagyományos memória	Meghajtóprogramok telepítése a felső memóriaterületre, ezzel a hagyományos memóriára felszabadítása.

LOADHIGH	Parancssor/ AUTOEXEC.BAT	Felső memória	Hagyományos memória	Memóriarezidens programok futtatása a felső memóriaterületen ezzel a hagyományos memória felszaba- dítása.
BUFFERS=	CONFIG.SYS	Hagyományos	—	Az állományműveletek végrehajtásához fel- használható ideiglenes tárolóhelyek számának megfelelő beállításával a számítógép működé- sének gyorsítása.
FASTOPEN.EXE	Parancssor/ AUTOEXEC.BAT	Hagyományos /X kapcsoló esetén: hagyomá- nyos és kiterjesz- tett	—	A gyakran használt ál- lományok megnyitásá- hoz szükséges idő csökkentése, vagyis a számítógép működésé- nek gyorsítása.
INSTALL= FASTOPEN.EXE	CONFIG.SYS	Hagyományos /X kapcsoló esetén: hagyomá- nyos és kiterjesz- tett	—	A gyakran használt ál- lományok megnyitásá- hoz szükséges idő csökkentése, vagyis a számítógép működésé- nek gyorsítása.
SMARTDRV.SYS	CONFIG.SYS	Hagyományos és kibővített me- mória /A kap- csoló esetén: Hagyományos és kiterjesztett memória	—	A merevlemez írásá- hoz, ill. olvasásához szükséges idő csökken- tése a kibővített vagy kiterjesztett memóriá- ban létrehozott átmeneti tár — cache memória — létesítésével.
RAMDRIVE.SYS	CONFIG.SYS	/E kapcsoló esetén: Hagyo- mányos és kibő- vített /A kapcsoló esetén: Hagyo- mányos és kiter- jesztett	—	Állományok tárolása.

## Mit jelent?

**Gbájt.** A Gbájt 1024 Mbájtval egyenlő. 1 Mbájt pedig 1024 • 1024 bájtval, azaz 1048576 bájtval egyenlő.

# **ERRORLEVEL értékek**

Ebben a függelékben azokat a DOS parancsokat adjuk meg, amelyek a végrehajtásuk után kilépési értéket állítanak elő, vagyis amelyek az ERRORLEVEL változónak értéket adnak.

<b>Változó</b>	<b>Kilépési érték</b>	<b>Jelentés</b>
<b>BACKUP</b>	0	Sikeres művelet
	1	Állomány a biztonsági mentéshez nem található
	2	Bizonyos állományok biztonsági mentését nem lehetett elvégezni
	3	A felhasználó a Ctrl—C lenyomásával a parancs végrehajtását felfüggesztette
	4	A parancs végrehajtásának befejezése hiba következtében
<b>DISKCOMP</b>	0	A lemezek azonosak
	1	A lemezek eltérők
	2	A felhasználó a Ctrl—C lenyomásával a parancs végrehajtását felfüggesztette
	3	Lemezhiba
	4	Érvénytelen szintaxis, nem elegendő a memória vagy érvénytelen meghajtó előírása
<b>DISKCOPY</b>	0	Sikeres másolási művelet
	1	Nem végzetes olvasási vagy írási hiba a másolási művelet során
	2	A felhasználó a Ctrl—C lenyomásával a parancs végrehajtását felfüggesztette
	3	Végzetes írási vagy olvasási hiba
	4	Érvénytelen meghajtó, érvénytelen szintaxis vagy a memória nem elegendő
<b>FORMAT</b>	0	Sikeres formázás
	3	A felhasználó a Ctrl—C lenyomásával a parancs végrehajtását felfüggesztette
	4	Fatális hiba

	5	A felhasználó az N karakterrel válaszolt a Proceed with Format (Y/N)? üzenetre
GRAFTABL	0	A táblázat sikeresen betöltve
	1	A táblázat egy már meglévőt cserélt ki
	2	Állományhiba megakadályozta az új táblázat betöltését
	3	Érvénytelen parancssor paraméter előírása
	4	Helytelen DOS verzió; 5.0-s DOS verzió szükséges
KEYB	0	Billentyűzetdefiníciós állomány sikeresen betöltve
	1	Érvénytelen billentyűkód, kódlap vagy szintaxis használata
	2	Érvénytelen, vagy hiányzik a billentyűzet definiáló állomány
	4	Hiba a CON eszközzel való kommunikáció során
	5	A kívánt kódlap nem lett előkészítve
REPLACE	0	Valamennyi állomány sikeresen kicserélve
	2	A forrásállományok nem találhatóak
	3	A forrás vagy a célútvonal nem található
	5	Állomány-hozzáférés megsértése
	8	A memória nem elegendő a művelet elvégzéséhez
	11	Érvénytelen szintaxis használata a parancssorban
	15	Érvénytelen az előírt meghajtó (Az 5.0-s DOS-ban már nem szerepel.)
	22	Helytelen DOS verzió (Az 5.0-s DOS-ban már nem szerepel.)
RESTORE	0	Sikeres visszaállítás
	1	Nem található az előírásnak megfelelő visszaállítandó állományok
	3	A felhasználó a Ctrl—C lenyomásával a parancs végrehajtását felfüggesztette
	4	Végzetes feldolgozási hiba
SETVER	0	A művelet sikeres befejezése
	1	Érvénytelen kapcsoló használata a parancssorban

	2	A felhasználó érvénytelen állománynevet írt elő
	3	A memória nem elegendő a parancs végrehajtásához
	4	Érvénytelen verziószám formátum használata
	5	A SETVER nem tudta megtalálni az előírt bejegyzést a verziótáblában
	6	A SETVER.EXE program nem található
	7	A felhasználó által előírt meghajtó érvénytelen
	8	Túl sok parancssor-paraméter előírása
	9	Hiányzó parancssor paraméterek
	10	Hiba a SETVER.EXE beolvasása során
	11	A SETVER.EXE állomány megsérült
	12	Az előírt SETVER.EXE állomány nem támogatja a verziótáblát
	13	A verziótáblában nincs hely az új bejegyzés számára
	14	Hiba a SETVER.EXE írásakor
XCOPY	0	Az állományok sikeres másolása
	1	A másoláshoz nem található állomány
	2	A felhasználó a Ctrl—C lenyomásával a parancs végrehajtását felfüggesztette
	4	A memória nem elegendő, érvénytelen a meghajtó, vagy hibás a szintaxis
	5	A másolást fatális lemezírási hiba akadályozta meg



*B Függelék*

# ***Kódlapok az 5.0-s DOS-ban***

Ebben a függelékben azt a hat kódlapot adjuk meg, amelyeket a nemzeti sajátosságok figyelembevételéhez fel lehet használni. Külön felhívjuk az Olvasó figyelmét a 437-es kódlapra, amely a leggyakrabban előforduló hardver kódlap, valamint a 852-es és a 850-es kódlapra, amelyek a magyar ékezetes billentyűzet definiálásához mint alapértelmezés szerinti ill. alternatív kódlapként használandók.

0	32	64	Q	96	'	128	Ç	160	á	192	L	224	α		
1	☒	33	!	65	A	97	a	129	ü	161	í	193	⊥	225	β
2	☒	34	"	66	B	98	b	130	é	162	ó	194	T	226	Γ
3	♥	35	#	67	C	99	c	131	â	163	ú	195	†	227	Π
4	♦	36	\$	68	D	100	d	132	ä	164	ñ	196	-	228	Σ
5	♣	37	%	69	E	101	e	133	à	165	Ñ	197	†	229	σ
6	♠	38	&	70	F	102	f	134	ã	166	ë	198	†	230	μ
7	•	39	'	71	G	103	g	135	ç	167	º	199		231	τ
8	◼	40	(	72	H	104	h	136	ê	168	¿	200	⊥	232	ϑ
9	◇	41	)	73	I	105	i	137	ë	169	ƒ	201	ƒ	233	θ
10	◼	42	*	74	J	106	j	138	è	170	¬	202	≡	234	Ω
11	♂	43	+	75	K	107	k	139	ï	171	½	203	≡	235	δ
12	♀	44	,	76	L	108	l	140	î	172	¼	204		236	ω
13	♯	45	_	77	M	109	m	141	ì	173	ï	205	=	237	ϕ
14	♮	46	.	78	N	110	n	142	ñ	174	«	206		238	€
15	*	47	/	79	O	111	o	143	â	175	»	207	±	239	Π
16	▶	48	0	80	P	112	p	144	É	176	▒	208	≡	240	≡
17	◀	49	1	81	Q	113	q	145	æ	177	▒	209	≡	241	±
18	↕	50	2	82	R	114	r	146	ff	178	▒	210	π	242	≥
19	!!	51	3	83	S	115	s	147	ô	179		211	⊥	243	≤
20	¶	52	4	84	T	116	t	148	ö	180	†	212	⊥	244	∫
21	§	53	5	85	U	117	u	149	ò	181	†	213	ƒ	245	J
22	-	54	6	86	V	118	v	150	û	182		214	π	246	÷
23	±	55	7	87	W	119	w	151	ù	183	π	215		247	≈
24	↑	56	8	88	X	120	x	152	ÿ	184	¬	216	†	248	°
25	↓	57	9	89	Y	121	y	153	ö	185		217	J	249	.
26	→	58	:	90	Z	122	z	154	Ü	186		218	ƒ	250	.
27	←	59	;	91	[	123	{	155	ç	187	≡	219	▒	251	√
28	└	60	<	92	\	124		156	£	188	≡	220	▒	252	n
29	↔	61	=	93	]	125	}	157	¥	189	≡	221	▒	253	z
30	▲	62	>	94	^	126	~	158	℞	190	J	222	▒	254	■
31	▼	63	?	95	_	127	△	159	f	191	¬	223	▒	255	□

0		32		64	Ⓔ	96	•	128	Ç	160	á	192	L	224	ó
1	Ⓔ	33	†	65	Ⓐ	97	a	129	ü	161	í	193	Ⓙ	225	ß
2	Ⓚ	34	"	66	B	98	b	130	é	162	ó	194	T	226	ô
3	♥	35	#	67	C	99	c	131	â	163	ú	195	†	227	ò
4	♦	36	\$	68	D	100	d	132	ä	164	ñ	196	—	228	õ
5	♣	37	%	69	E	101	e	133	à	165	Ñ	197	†	229	õ
6	♠	38	&	70	F	102	f	134	ã	166	•	198	ã	230	µ
7	•	39	'	71	G	103	g	135	ç	167	•	199	Ã	231	þ
8	◼	40	(	72	H	104	h	136	ê	168	¿	200	Ⓛ	232	þ
9	◊	41	)	73	I	105	i	137	ë	169	Ⓔ	201	Ⓙ	233	ú
10	◼	42	*	74	J	106	j	138	è	170	¬	202	Ⓛ	234	û
11	♂	43	+	75	K	107	k	139	ÿ	171	½	203	Ⓙ	235	ù
12	♀	44	,	76	L	108	l	140	î	172	¼	204	†	236	ý
13	♯	45	_	77	M	109	m	141	ì	173	•	205	—	237	ÿ
14	♮	46	.	78	N	110	n	142	Ā	174	◀	206	†	238	·
15	✱	47	/	79	O	111	o	143	Ă	175	▶	207	Ⓔ	239	˘
16	▶	48	0	80	P	112	p	144	É	176	▤	208	δ	240	·
17	◀	49	1	81	Q	113	q	145	æ	177	▥	209	Ⓔ	241	±
18	†	50	2	82	R	114	r	146	Ⓔ	178	▧	210	Ê	242	=
19	!!	51	3	83	S	115	s	147	ô	179		211	Ë	243	¾
20	¶	52	4	84	T	116	t	148	ö	180	†	212	È	244	¶
21	§	53	5	85	U	117	u	149	ò	181	Á	213	•	245	§
22	—	54	6	86	V	118	v	150	û	182	Â	214	Í	246	÷
23	±	55	7	87	W	119	w	151	ù	183	À	215	Î	247	˘
24	↑	56	8	88	X	120	x	152	ÿ	184	Ⓔ	216	Ï	248	•
25	↓	57	9	89	Y	121	y	153	ö	185	†	217	J	249	˘
26	→	58	:	90	Z	122	z	154	Ü	186		218	Γ	250	•
27	←	59	;	91	[	123	{	155	ø	187	¶	219	▣	251	•
28	└	60	<	92	\	124		156	£	188	Ⓛ	220	▣	252	•
29	↔	61	=	93	]	125	}	157	Ⓔ	189	Ç	221	•	253	•
30	▲	62	>	94	^	126	˘	158	×	190	¥	222	•	254	▣
31	▼	63	?	95	_	127	△	159	f	191	Γ	223	▣	255	▣

0		32		64	Q	96	'	128	Q	160	á	192	L	224	Ó
1	☐	33	!	65	A	97	a	129	ü	161	í	193	l	225	ö
2	☐	34	"	66	B	98	b	130	é	162	ó	194	T	226	ô
3	♥	35	#	67	C	99	c	131	â	163	ú	195	†	227	ón
4	♦	36	\$	68	D	100	d	132	ä	164	ŕ	196	-	228	ń
5	♣	37	%	69	E	101	e	133	û	165	q	197	†	229	ň
6	♠	38	&	70	F	102	f	134	é	166	ž	198	ǎ	230	š
7	•	39	'	71	G	103	g	135	ç	167	ž	199	ǎ	231	š
8	☐	40	(	72	H	104	h	136	ł	168	ę	200	u	232	ř
9	◊	41	)	73	I	105	i	137	ë	169	ę	201	π	233	ú
10	☐	42	*	74	J	106	j	138	ő	170		202	u	234	ř
11	♂	43	+	75	K	107	k	139	ó	171	z	203	π	235	ű
12	♀	44	,	76	L	108	l	140	î	172	č	204	π	236	ý
13	♯	45	_	77	M	109	m	141	ž	173	š	205	=	237	ý
14	♮	46	.	78	N	110	n	142	ǎ	174	«	206	π	238	ț
15	✳	47	/	79	O	111	o	143	ć	175	»	207	⊗	239	,
16	▶	48	0	80	P	112	p	144	é	176	☐	208	đ	240	-
17	◀	49	1	81	Q	113	q	145	ĺ	177	☐	209	đ	241	"
18	↕	50	2	82	R	114	r	146	í	178	☐	210	ď	242	'
19	!!	51	3	83	S	115	s	147	ô	179		211	ë	243	˘
20	¶	52	4	84	T	116	t	148	ö	180	†	212	ǎ	244	˘
21	§	53	5	85	U	117	u	149	ľ	181	á	213	ň	245	š
22	-	54	6	86	V	118	v	150	ĩ	182	â	214	í	246	+
23	±	55	7	87	W	119	w	151	š	183	ě	215	î	247	˘
24	↑	56	8	88	X	120	x	152	ś	184	š	216	ě	248	•
25	↓	57	9	89	Y	121	y	153	ö	185	π	217	J	249	..
26	→	58	:	90	Z	122	z	154	ü	186	π	218	Γ	250	•
27	←	59	:	91	[	123	{	155	ř	187	π	219	■	251	ú
28	└	60	<	92	\	124		156	ř	188	π	220	■	252	ř
29	↔	61	=	93	]	125	}	157	ł	189	ž	221	†	253	ř
30	▲	62	>	94	^	126	~	158	x	190	ž	222	ú	254	■
31	▼	63	?	95	_	127	△	159	č	191	└	223	■	255	

0		32		64	Ⓔ	96	'	128	Ç	160	á	192	Ⓕ	224	α
1	☒	33	!	65	Ⓐ	97	a	129	ü	161	í	193	Ⓖ	225	β
2	Ⓢ	34	"	66	Ⓑ	98	b	130	é	162	ó	194	Ⓗ	226	Γ
3	♥	35	#	67	Ⓒ	99	c	131	â	163	ú	195	Ⓙ	227	Π
4	♦	36	\$	68	Ⓓ	100	d	132	ã	164	ñ	196	—	228	Σ
5	♣	37	%	69	Ⓔ	101	e	133	à	165	ñ	197	†	229	σ
6	♠	38	&	70	Ⓕ	102	f	134	Á	166	º	198	‡	230	μ
7	•	39	'	71	Ⓖ	103	g	135	ç	167	º	199		231	γ
8	◼	40	(	72	Ⓖ	104	h	136	ê	168	¿	200	Ⓖ	232	δ
9	◇	41	)	73	Ⓖ	105	i	137	Ê	169	Ò	201	Ⓖ	233	θ
10	◼	42	*	74	Ⓖ	106	j	138	è	170	¬	202	Ⓖ	234	Ω
11	♂	43	+	75	Ⓖ	107	k	139	ì	171	½	203	Ⓖ	235	δ
12	♀	44	,	76	Ⓖ	108	l	140	Ô	172	¾	204	Ⓖ	236	ω
13	♯	45	_	77	Ⓖ	109	m	141	ì	173	¡	205	=	237	∅
14	♭	46	.	78	Ⓖ	110	n	142	Ã	174	«	206	Ⓖ	238	€
15	✳	47	/	79	Ⓖ	111	o	143	Â	175	»	207	Ⓖ	239	∩
16	▶	48	0	80	Ⓖ	112	p	144	É	176	▒	208	Ⓖ	240	≡
17	◀	49	1	81	Ⓖ	113	q	145	À	177	▒	209	Ⓖ	241	±
18	↕	50	2	82	Ⓖ	114	r	146	È	178	▒	210	Ⓖ	242	≥
19	!!	51	3	83	Ⓖ	115	s	147	ô	179		211	Ⓖ	243	≤
20	¶	52	4	84	Ⓖ	116	t	148	õ	180		212	Ⓖ	244	∫
21	Ⓢ	53	5	85	Ⓖ	117	u	149	ò	181		213	Ⓖ	245	∫
22	—	54	6	86	Ⓖ	118	v	150	Ú	182	Ⓖ	214	Ⓖ	246	÷
23	±	55	7	87	Ⓖ	119	w	151	ù	183	Ⓖ	215	Ⓖ	247	≈
24	↑	56	8	88	Ⓖ	120	x	152	ì	184	Ⓖ	216	Ⓖ	248	°
25	↓	57	9	89	Ⓖ	121	y	153	õ	185	Ⓖ	217	Ⓖ	249	•
26	→	58	:	90	Ⓖ	122	z	154	Ü	186	Ⓖ	218	Ⓖ	250	•
27	←	59	;	91	Ⓖ	123	{	155	ç	187	Ⓖ	219	■	251	√
28	└	60	<	92	Ⓖ	124		156	£	188	Ⓖ	220	■	252	∥
29	↔	61	=	93	Ⓖ	125	}	157	ù	189	Ⓖ	221	■	253	z
30	▲	62	>	94	^	126	~	158	Ŕ	190	Ⓖ	222	■	254	■
31	▼	63	?	95	_	127	△	159	Ó	191	Ⓖ	223	■	255	■

0		32		64	Ⓔ	96	·	128	Ç	160	ı	192	Ł	224	α
1	☒	33	!	65	Ⓐ	97	a	129	ü	161	´	193	ł	225	β
2	☓	34	"	66	Ⓑ	98	b	130	é	162	ó	194	ṽ	226	Γ
3	♥	35	#	67	Ⓒ	99	c	131	â	163	ú	195	†	227	Π
4	♦	36	\$	68	Ⓓ	100	d	132	Â	164	¨	196	—	228	Σ
5	♣	37	%	69	Ⓔ	101	e	133	à	165	˘	197	‡	229	σ
6	♠	38	&	70	Ⓕ	102	f	134	ŋ	166	3	198	‡	230	μ
7	•	39	'	71	Ⓖ	103	g	135	ç	167	-	199	‡	231	τ
8	◐	40	(	72	Ⓖ	104	h	136	ê	168	î	200	ℓ	232	ϖ
9	◑	41	)	73	Ⓖ	105	i	137	ë	169	ı	201	ℓ	233	θ
10	◒	42	*	74	Ⓖ	106	j	138	è	170	ı	202	ℓ	234	Ω
11	♂	43	+	75	Ⓖ	107	k	139	ÿ	171	½	203	ℓ	235	δ
12	♀	44	,	76	Ⓖ	108	l	140	î	172	¼	204	ℓ	236	ω
13	♃	45	_	77	Ⓖ	109	m	141	=	173	¾	205	=	237	Ϙ
14	♄	46	.	78	Ⓖ	110	n	142	À	174	◀	206	ℓ	238	€
15	♁	47	/	79	Ⓔ	111	o	143	Û	175	▶	207	ℓ	239	Π
16	▶	48	0	80	Ⓖ	112	p	144	É	176	▒	208	ℓ	240	≡
17	◀	49	1	81	Ⓖ	113	q	145	È	177	▒	209	ℓ	241	±
18	↕	50	2	82	Ⓖ	114	r	146	Ê	178	▒	210	ℓ	242	≥
19	!!	51	3	83	Ⓖ	115	s	147	Ô	179		211	ℓ	243	≤
20	ŋ	52	4	84	Ⓖ	116	t	148	Ë	180	†	212	ℓ	244	ƒ
21	Û	53	5	85	Ⓖ	117	u	149	Ï	181	†	213	ℓ	245	J
22	—	54	6	86	Ⓖ	118	v	150	û	182	ℓ	214	ℓ	246	÷
23	±	55	7	87	Ⓖ	119	w	151	ú	183	ℓ	215	ℓ	247	≈
24	↑	56	8	88	Ⓖ	120	x	152	Ï	184	†	216	†	248	◦
25	↓	57	9	89	Ⓖ	121	y	153	Ô	185	ℓ	217	J	249	•
26	→	58	:	90	Ⓖ	122	z	154	Ü	186	ℓ	218	ı	250	•
27	←	59	;	91	[	123	{	155	ç	187	ŋ	219	■	251	√
28	└	60	<	92	\	124		156	ç	188	ŋ	220	■	252	n
29	↔	61	=	93	]	125	}	157	Ù	189	ŋ	221	■	253	z
30	▲	62	>	94	^	126	~	158	Ô	190	J	222	■	254	■
31	▼	63	?	95	_	127	△	159	ƒ	191	ı	223	■	255	

0		32		64	Q	96	'	128	Ç	160	á	192	L	224	α
1	☺	33	!	65	A	97	a	129	ü	161	í	193	⊥	225	β
2	☹	34	"	66	B	98	b	130	é	162	ó	194	T	226	Γ
3	♥	35	#	67	C	99	c	131	â	163	ú	195	†	227	Π
4	♦	36	\$	68	D	100	d	132	ä	164	ñ	196	—	228	Σ
5	♣	37	%	69	E	101	e	133	à	165	Ñ	197	†	229	σ
6	♠	38	&	70	F	102	f	134	ã	166	ª	198	†	230	μ
7	•	39	'	71	G	103	g	135	ç	167	º	199	‡	231	τ
8	◼	40	(	72	H	104	h	136	ê	168	¿	200	⊥	232	ϑ
9	◊	41	)	73	I	105	i	137	ë	169	ƒ	201	‡	233	θ
10	◼	42	*	74	J	106	j	138	è	170	ƒ	202	⊥	234	Ω
11	♂	43	+	75	K	107	k	139	ÿ	171	½	203	‡	235	δ
12	♀	44	,	76	L	108	l	140	î	172	¼	204	‡	236	ω
13	♪	45	-	77	M	109	m	141	ì	173	ï	205	=	237	φ
14	♫	46	.	78	N	110	n	142	Ë	174	«	206	‡	238	€
15	✱	47	/	79	O	111	o	143	Ā	175	⊗	207	⊥	239	Π
16	▶	48	0	80	P	112	p	144	É	176	▒	208	⊥	240	≡
17	◀	49	1	81	Q	113	q	145	æ	177	▒	209	‡	241	±
18	↕	50	2	82	R	114	r	146	ff	178	▒	210	π	242	≥
19	!!	51	3	83	S	115	s	147	ô	179		211	⊥	243	≤
20	¶	52	4	84	T	116	t	148	ö	180	†	212	⊥	244	∫
21	§	53	5	85	U	117	u	149	ò	181	†	213	ƒ	245	J
22	-	54	6	86	V	118	v	150	û	182	‡	214	π	246	÷
23	±	55	7	87	W	119	w	151	ú	183	π	215	‡	247	≈
24	↑	56	8	88	X	120	x	152	ÿ	184	ƒ	216	†	248	•
25	↓	57	9	89	Y	121	y	153	ö	185	‡	217	J	249	•
26	→	58	:	90	Z	122	z	154	ü	186	‡	218	ƒ	250	•
27	←	59	;	91	[	123	{	155	ø	187	π	219	▒	251	J
28	└	60	<	92	\	124		156	£	188	⊥	220	▒	252	n
29	↔	61	=	93	]	125	}	157	Ø	189	⊥	221	▒	253	z
30	▲	62	>	94	^	126	~	158	R	190	⊥	222	▒	254	■
31	▼	63	?	95	_	127	△	159	f	191	ƒ	223	▒	255	

*C Függelék*

# ***Kódtáblázat a nemzeti sajátosságok figyelembevételére 5.0-s DOS használata esetén***

<b>Ország/területegység vagy nyelv</b>	<b>Országkód</b>	<b>Billentyűkód</b>	<b>Alapértelmezés szerinti kódlap</b>	<b>Alternatív kódlap</b>
Belgium	032	be	850	437
Brazília	055	br	850	437
Kanadai-francia	002	cf	863	850
Csehország	042	cz	852	850
Szlovákia	042	sl	852	850
Dánia	045	dk	850	865
Finnország	358	su	850	437
Franciaország	033	fr	850	437
Németország	049	gr	850	437
Magyarország	036	hu	852	850
Nemzetközi angol	061		437	850
Olaszország	039	it	850	437
Latin-Amerika	003	la	850	437
Hollandia	031	nl	850	437
Norvégia	047	no	850	865
Lengyelország	048	pl	852	850
Portugália	351	po	850	860
Spanyolország	034	sp	850	437
Svédország	046	sv	850	437
Svájc (francia)	041	sf	850	437
Svájc (német)	041	sg	850	437
Egyesült Királyság	044	uk	437	850
Egyesült Államok	001	us	437	850
Jugoszlávia	038	yu	852	850



# ***Hardver napló***

Új programok telepítése, az adott konfiguráció bővítése, valamint a megfelelő memóriakezelés kialakítása vagy a nemzeti sajátosságok miatt szükséges beállításokhoz az adott hardver rendszer jellemzőit ismerni kell. A konfigurációt egyértelműen azonosító adatokat érdemes az ún. hardver naplóban nyilvántartani. A következőkben ezen napló tartalmának kialakításához adunk ötleteket. (A napló természetesen szükség esetén tovább bővíthető.)

A számítógép típusa: 88/86/286/386/486

Sorozatszám:

Gyártó cég:

A gyártás időpontja:

A garancia lejártának időpontja:

Felhasználó neve: (A gépért felelős személy neve)

További felhasználók neve: (Akiket a konfigurálás változásairól még értesíteni kell.)

Segédprocesszor: van/nincs

Valós idő óra: van/nincs

Az óra elemcseréjének időpontja:

Hagyományos (standard) memória mérete: 512K/640K

Kiterjesztett (expanded) memória mérete: ami a rendszerben rendelkezésre áll, vagy nulla

Kibővített (extended) memória: ami a rendszerben rendelkezésre áll, vagy nulla

A kiterjesztett memória létesítésének módja: ..... típusú bővítőkártya a hozzá tartozó ..... nevű meghajtóprogrammal

Az A: meghajtó mérete: 360K/720K/1.2Mb/1.44Mb

A B: meghajtó mérete: 360K/720K/1.2Mb/1.44Mb

A merevlemez mérete: 20Mb/40Mb/120Mb

A merevlemez meghajtón létesített partíciók: C,D,E stb.

Az egér típusa: PC Mouse, MS Mouse, Genius Mouse stb.

Az egeret meghajtó szoftver: AUTOEXEC.BAT-ban: pl. mouse.com  
CONFIG.SYS-ben: pl. a mouse.sys

Az egér csatlakoztatásának helye: COM1

A modem típusa: Hayes 1200 smartmodem stb.

A modem csatlakoztatásának helye: COM2, COM3 stb.

Monitorhoz tartozó meghajtókártya típusa: MDA, MDA/Hercules, CGA, EGA, MCGA, PGA, VGA, XVGA (SVGA), XGA, LCD

A monitor típusa: MDA Monokróm nagyfelbontású, Színes a CGA kártyához, Színes az EGA kártyához, Multiszinkron stb.

A nyomtató típusa: HP Laserjet III/EPSON LX-800 stb.

(Ide nem feltétlenül az adott nyomtató típusát kell beírni, hanem azt a típust, amelyet a különböző alkalmazói programok telepítésekor meg kell adni. Például előfordulhat, hogy egy adott program akkor kezeli le megfelelően az EPSON LX-800-as nyomtatót, ha azt IBM Proprinter-nek adjuk meg a telepítés során.)

Nyomtató csatlakoztatásának helye: LPT1

Fontkazetta: Lézernyomtató esetén meg kell adni, hogy melyik az éppen használatban levő külső fontkazetta

Magyar nyelvű karakterkészlet előállításának módja: Vagy külön meghajtóprogrammal (esetleg külön EPROM-mal), vagy az 5.0-s DOS szolgáltatásainak igénybevételevel

Egyéb külső eszköz: HP rajzgép

Csatlakoztatásának helye:

Kétállású kapcsolók beállításának módja a nyomtatónál (DIP kapcsolók):

modemnél:

monitorkártyán:

alaplapon:

# Mutató

## A

@ECHO OFF 123  
Allokációs táblázat 92  
ATTRIB parancs 68, 69, 73, 91  
    használatának gyakori hibái 72  
Attribútum 67  
    archív 69  
    csak olvasható 69  
    rejtett 69  
    rendszer 69  
Attribútumbájt 50  
AUTOEXEC.BAT 114, 140  
    minimális tartalma 136  
    szokásos tartalma 136—137  
AUX 17, 20  
Azonosítónév 19  
Átmeneti tároló 22

## B

BACKSPACE 21, 37  
BACKUP parancs 93, 113  
    használata 97, 103, 107—113  
    használatának gyakori hibái 102  
    kapcsolói 98—99  
BALRA kurzormozgató gomb 37  
Bájt 14, 20  
Belső DOS parancs 21  
Billentyűzetpuffer 22  
Bit 14, 20  
Biztonságos formázás 82  
BUFFERS= bejegyzés 201, 202

## C

Cache memória 205—208  
CALL parancs 124, 125, 140  
CGA 183

CHCP parancs 177  
COMP parancs 57, 61, 67  
    használatának gyakori hibái 61  
    kapcsolói 60  
COM1 16, 20  
CON 9, 12, 20  
CONFIG.SYS 142  
COPY PARANCS 39  
COUNTRY=bejegyzés 162—163  
Csak olvasható állomány 68

## D

DATE parancs 24  
DEL billentyű 27, 28, 37  
DELETE parancs 74  
DEVICE= bejegyzés 144, 151  
DEVICEHIGH= bejegyzés 196,  
    198, 201  
DISKCOMP parancs 46, 48, 67  
    használatának gyakori hibái 49  
DISKCOPY parancs 40, 45, 67  
    használatának gyakori hibái 45  
DOS eszköznév 17  
DOS prompt 21  
DOS= bejegyzés 195, 197, 202  
DOSKEY segédprogram 29, 38  
    kapcsolói 34  
    FÖL/LE gomb 29  
    PGDN gomb 30  
    PGUP gomb 30  
    szerkesztőbillentyűi 33  
DOSKEY makro definiálása 35  
DOSKEY makro törlése 35

## E

ECHO OFF 123

EGA 183  
Egértelapítás 145  
Eltolási érték (offset) 67  
EMM386.EXE 198, 200  
ERRORLEVEL értékek 213—215  
Eszközmeghajtó állományok  
    146—151  
    ANSI.SYS 147  
    DISPLAY.SYS 147  
    DRIVER.SYS 148  
    EGA.SYS 150  
    HINEM.SYS 150  
    RAMDRIVE.SYS 150  
    SMARTDRV.SYS 150  
Expanded I. Kiterjesztett memória  
Extended I. Kibővített memória

**F**  
FASTOPEN program 203—205  
FASTOPEN.EXE 153  
FCBS= bejegyzés 201  
Felső memóriaterület 185  
FOR parancs 132, 134, 141  
FORMAT parancs 74  
    rendszerlemez készítéséhez  
    142—143  
Funkcióbillentyű 22, 26, 37

**G**  
GOTO parancs 134—135, 141  
Gyorsítótár 205—208

**H**  
Hardver kódlap 161  
Hardver napló 224—225  
Helyettesítő karaktercsillag 39  
    kérdőjel 39  
High Memory Area I. Magas memóriaterület  
HIMEM.SYS 194, 197

**I**  
IF parancs 126, 132, 140  
    IF ERRORLEVEL 129, 132  
    IF EXIST 126, 132  
    IF karakterlánc == karakterlánc 2,  
    127, 132  
INS billentyű 27, 28, 37  
INSTALL= bejegyzés 152, 154, 158

**K**  
Karakterlánc 141  
Kettes számrendszer 14  
KEYB parancs 164—167, 177  
KEYB.COM 153  
Klaszter 92  
Kötetelnevezés 9, 19  
Kódlap 159, 161, 216—222  
    aktivizálása 167  
    betöltése 167  
    fogadásának előkészítése CON  
    eszköznél 168  
    fogadásának előkészítése nyomta-  
    tónál 170  
Kódlapváltás 161  
Kódtáblázat 223  
Kulcsinformáció 74  
Külső DOS parancs 21

**L**  
LABEL parancs 9—10, 19  
Lap 188  
LASTDRIVE= bejegyzés  
    143—144, 158  
LCD 183  
Lemezformázás 68  
Lemeztípusok 96  
LOADHIGH parancs 196, 199, 201  
LPT1 16, 20

## M

- Magas memóriaterület 186
- Makro 38
- Mechanikai írásvédelem 39
  - írásvédelmi címkével 39
  - írásvédelmi lyukkal 39
- Megszakítás 159
- MEM parancs 190—193
  - kapcsolói 191
- Memória 184
  - hagyományos 185, 188
  - kibővített 185—187, 188
  - kiterjesztett 187—189
- Memóriarezidens program 142
- MIRROR utasítás 74, 75, 76, 85, 92
- MODE parancs 178—179
  - Kódlap aktivizálása 174—176
  - Kódlap betöltése 173—174
- MONO 183

## N

- Nagyméretű állomány 113
- Nemzeti sajátosságok figyelembevétele 160, 180, 183
- NLSFUNC parancs 171—172
- NLSFUNC.EXE 153
- NOT operátor 131

## O

- Oldalkeret 188

## P

- Parancsállomány paraméter 116, 119, 140
- Parancsállomány 114, 116, 139
- PAUSE parancs 122, 125, 140
- Párhuzamos eszköz 13, 14
- Párhuzamos port 16
- Port 9
- PRN 16

## Puffer 38

## R

- RAM meghajtó 208—210
- RAM 184
- RAMDRIVE.SYS 201, 208—210
- Rejtett állomány 68
- REM parancs 140
  - a CONFIG.SYS-ben 151—152, 158, 200
  - az AUTOEXEC.BAT-ban 120, 125, 201
- REPLACE parancs 62, 66, 67
  - használatának gyakori hibái 66
  - kapcsolói 65
- RESTORE parancs 93, 104, 106, 107—113
  - használatának gyakori hibái 106
  - kapcsolói 105

## S

- Saját parancsállományok 137
- SHELL= bejegyzés 154—156, 159
- SHIFT parancs 119, 141
- SMARTDRV.SYS 201, 205—208
- Soros eszköz 13, 15
- Soros port 16
- Speciális DOS parancsok 114—115
- STACKS= bejegyzés 201
- Szintaxis 17, 20

## T

- Tizenhatos (hexadecimális) számrendszer 67

## U

- UNDELETE utasítás 85, 90, 92
  - használatának gyakori hibái 89
  - kapcsolói 89

**UNFORMAT** utasítás 77, 85, 92  
használatának gyakori hibái 84  
kapcsolói 83, 91  
**Upper Memory Block** I. Felső memóriaterület

## **V**

**Változó** 141  
**Verem** 159  
**VGA** 183

**Virtuális lemez** 208—210  
**Visszaállítás a MIRROR** használata nélkül 90  
**VOL** 11, 19

## **X**

**XCOPY** parancs 49, 56, 67  
használatának hibái 56  
kapcsolói 55

Minden jog fenntartva, jelen könyvet, ill. annak részeit tilos reprodukálni, adatrögzítő rendszerben tárolni, bármilyen formában vagy eszközzel — elektronikus, mechanikus, fényképeszeti úton vagy más módon — közölni a kiadó engedélye nélkül.

Készítette a Kaposvári Nyomda Kft. – 151304 – Felelős vezető: Mike Ferenc

650,- Ft

● Kezdő

● Középhaladó

● Haladó

Minden DOS verzióhoz

Ha Ön már rendelkezik alapvető számítógépes ismeretekkel pl. olvasta már az Egyszerűen DOS c. könyvet -, és meg kíván ismerkedni olyan hatékony DOS parancsokkal és segédprogramokkal, amelyek mindennapi munkáját hatékonyabbá és gyorsabbá teszik, akkor olvassa el ezt a könyvet. Ha Ön, bár gyakran hallja, mégsem tudja, mit jelentenek azok a „rejtélyes” fogalmak, mint kiterjesztett/kibővített memória, RAM meghajtó vagy felső memóriablokk, vagy azon ügyencek közé tartozik, akik saját parancsállományokat kívánnak készíteni, akkor ezt a könyvet el kell olvasnia.

A sorozatban már megjelent:

Egyszerűen DOS 6-os

Egyszerűen WINDOWS

Egyszerűen EXCEL

Egyszerűen WordPerfect

Egyszerűen WORD for WINDOWS

Egyszerűen 1-2-3

Egyszerűen WordPerfect for Windows

Egyszerűen WORD

Egyszerűen MIT JELENT?

Egyszerűen INTERNET

ISBN 963 545 077 X



9 789635 450770