

**Az SFD 1001 hajlékony
mágneselemez tároló
(Single Floppy Disk Drive)**

FELHASZNÁLÓI KÉZIKÖNYV

Commodore 64

AZ
SFD 1001 HAJLÉKONY
MÁGNESLEMEZES TÁROLÓ
(SINGLE FLOPPY DISK DRIVE)

FELHASZNÁLÓI KÉZIKÖNYVE

Felelős kiadó: Novotrade RT. igazgatója Renyi Gabor

Szerkesztés: Szatmari György
Tud. az. III. törvény 13. §-a alapján, a 136. §-a (1) bekezdése alapján

AGROÉPSZER GT. Nyomda Tsz: 109 85.

TARTALOMJEGYZÉK

1. FEJEZET: ÁLTALÁNOS ISMERTETÉS

Bevezetés	5
Általános jellemzők	5
Üzembe helyezés	5
A mágneslemezek kezelése	6
A helyes működés ellenőrzése	7

2. FEJEZET: A MÁGNESLEMEZES RENDSZER

A mágneslemez	8
A lemez operációs rendszer (DOS)	9
Csatorna, egység és funkciókód megadása	10
Összevont fájlnev megadás	10
Rövidített parancs megadás	11
Változók használata parancsoknál	11

3. FEJEZET: KÖZVETLEN DOS PARANCSON

Formai előírások	12
Szimbólumok	12
Mágneslemezre vonatkozó parancsok	13
HEADER	13
DIRECTORY/CATALOG	13
COLLECT	14
BACKUP	14
Fájlokra vonatkozó parancsok	14
DSAVE	14
DLOAD (Automatikus indítás)	14
RENAME	15
COPY	15
CONCAT	15
SCRATCH	15

4. FEJEZET: A DOS KEZELÉSE BASIC-BŐL

DOPEN #: 1. Szekvenciális fájlknál	16
2. Relatív fájlknál	16
APPEND #	16
DCLOSE #	17
PRINT #	17
INPUT #	17
GET #	17
RECORD #	18

5. FEJEZET: RELATÍV FÁJLOK

Relatív fájl létrehozása	19
Relatív fájl kibővítése	19
Adatok kezelése	20

6. FEJEZET: A DOS PROGRAMOZÁSA

Szimbólumok	21
Parancsok	21
Blokk lefoglalás	22
Blokk felszabadítás	22
Blokk olvasás	22
Blokk írás	22
Blokk végrehajtás	22
Puffer pozicionálás	22
RAM írás	
Egységcím átírása	23
RAM/ROM olvasás	23
RAM végrehajtás	23
Felhasználói parancsok	23

7. FEJEZET: SEKTOR FORMÁTUMOK

Szektor/sáv felosztás	25
BAM formátumok	25
1. BAM blokk	25
2. BAM blokk	25
3. BAM blokk	26
4. BAM blokk	26
Directory formátumok	26
Directory Header	26
Directory blokkok	27
Fájl adatblokk formátumok	27
Programfájl blokk	27
Szekvenciális (és relatív) adatblokk	27
Relatív fájl szektor-tábla	27

A) FÜGGELÉK: DOS PARANCSKÉSZLET	29
B) FÜGGELÉK: EGYSÉGSZÁM (CÍM) ÁTÁLLÍTÁSA	31
C) FÜGGELÉK: HIBAÜZENETEK	32

1. FEJEZET ÁLTALÁNOS ISMERTETÉS

BEVEZETÉS

Ez a kézikönyv a Commodore SFD 1001 típusú hajlékony mágneslemezes tároló („single floppy disk unit”) felhasználói számára készült.

Az adott számítógép felhasználói kézikönyvével és BASIC leírásával együtt a konfiguráció kezelésének megismeréséhez és hatékony alkalmazásához nyújt segítséget.

ÁLTALÁNOS JELLEMZŐK

A tároló a Commodore-64 és VIC-20 számítógépekhez és valamennyi BASIC 3.0 és BASIC 4.0 verziót alkalmazó CBM (Commodore Business Machines) és PET számítógéphez **IEEE-488 interfészen keresztül** közvetlenül csatlakoztatható.

Az SFD 1001-es tároló egy lemezegységet („drive”) foglal magába. Meghajtó és sáv pozicionáló mechanikát, motor vezérlést, írás/olvasás vezérlést és két író/olvasó fejet tartalmaz. Saját „intelligens” vezérléssel rendelkezik, így nem igényel külön memória területet a számítógép operatív tárába.

Adathordozóként 5 1/4” méretű kétoldalas, szoft szektor formátumozású hajlékony mágneslemez („Double Sided, Soft Sectorred Format Minifloppy Diskette”) alkalmazható. Egy lemezen oldalanként 77, összesen 154 sávon („track”) 254 adatbájtos szektorokban maximum 1 066 496 adatbajt (ill. karakter) tárolható. A sávok sűrűsége 100 TPI („Track Per Inch”).

ÜZEMBEHELYEZÉS

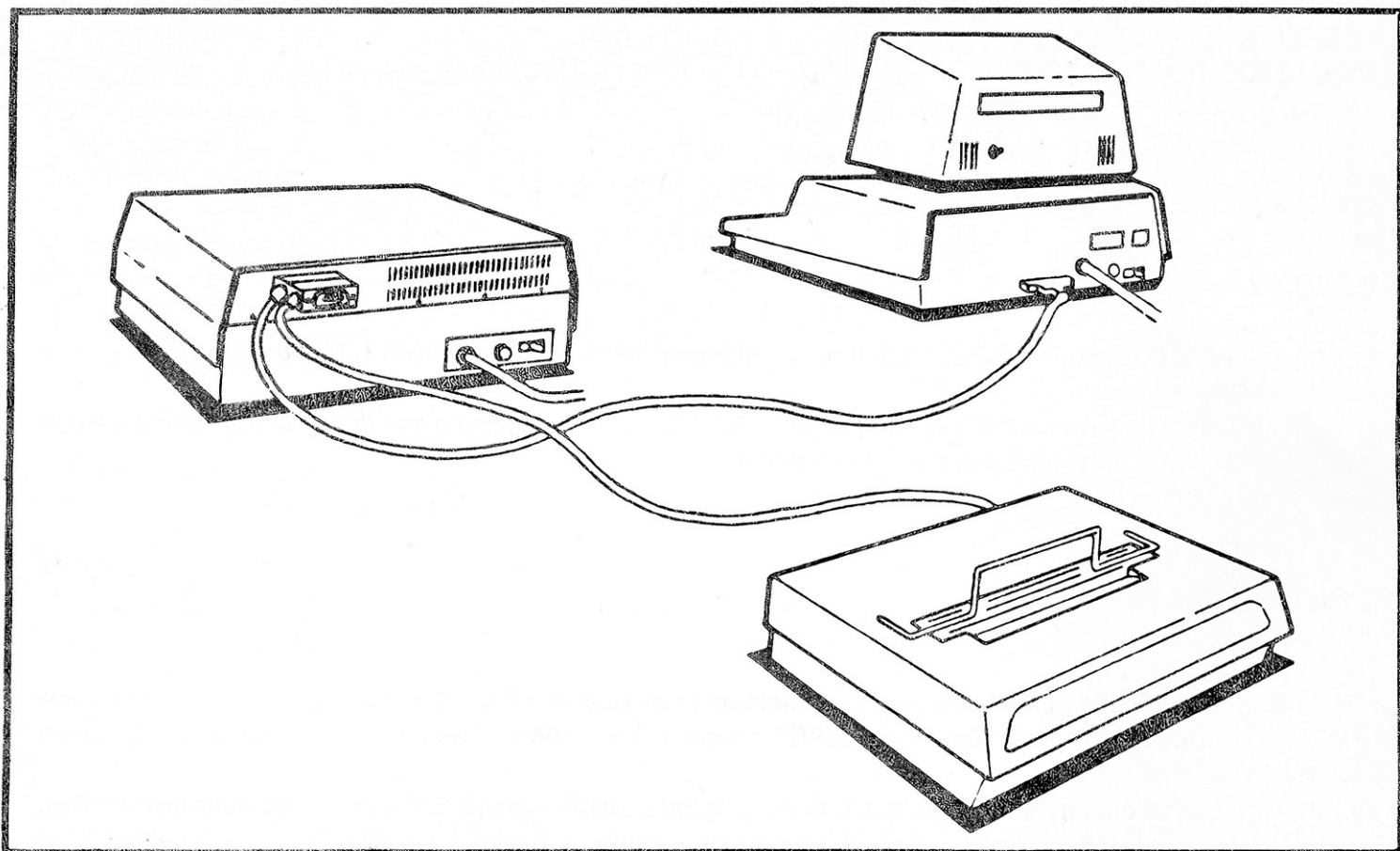
A mágneslemezes tároló ütésre, rázásra érzékeny berendezés, szállításnál, kicsomagolásnál óvatosan bánjunk vele.

Kicsomagolásnál győződjünk meg valamennyi tartozék meglétéről:

1. SFD 1001 típusú mágneslemezes tároló
2. Felhasználói kézikönyv
3. „TEST/DEMO” mágneslemez
4. IEEE-488 interfész kábel

A berendezést kikapcsolt állapotban csatlakoztassuk a 220 voltos hálózathoz, és az interfész kábellel a kikapcsolt számítógéphez!

(További IEEE-488 interfészű perifériák — pl. mátrix nyomtató — használata esetén azok egyikének interfész kábelét a mágneslemezes tárolóhoz csatlakoztatva, láncba „fűzzük fel” a perifériákat.)



1. ábra A mágneslemez tároló csatlakoztatása

A berendezések összekapcsolása után kapcsoljuk be a számítógépet, és győződjünk meg helyes működéséről!

Kapcsoljuk be a mágneslemezegység hálózati kapcsolóját! A készülék előlapján levő jelzőlámpák (LED-ek) kétszer felvillannak, a kétszínű, tápfeszültség bekapcsolást jelző (POW/ERR) LED pedig égve marad. (A POW/ERR LED zöld színe normál bekapcsolt állapotot – „POWER ON”, piros színe hibát – „ERROR”, jelez).

Ha valamennyi LED folyamatos, vagy villogó fényt ad, vagy pedig a POW/ERR LED 15 másodpercnél hosszabb ideig piros marad, kapcsoljuk ki a készüléket! Kis várakozás után újból kapcsoljuk be! A hibás állapot ismétlődése esetén kapcsoljuk ki a berendezéseket, és forduljunk a szervizhez.

A MÁGNESLEMEZEK KEZELÉSE

A hajlékony mágneslemez („floppy diskette”) sérülékeny, de helyes kezelés esetén megbízható, hosszú élettartamú adathordozó. Bánjunk vele óvatosan, ne erőltessük a meghajtó egységbe. Tartsuk távol elektromágneses eszközöktől, mint amilyen a villanymotor, transzformátor, televízió vagy video monitor. Sohase tegyük rá nehéz tárgyakat, csészét, üveget, könyvet. Használaton kívül tartsuk a saját papír borítójában.

A tárolt adatok végleges elvesztését megelőzendő, mágneslemezeinkről készítsünk tartalék másolatot („Backup Copy”), és azt tartsuk biztonságos helyen.

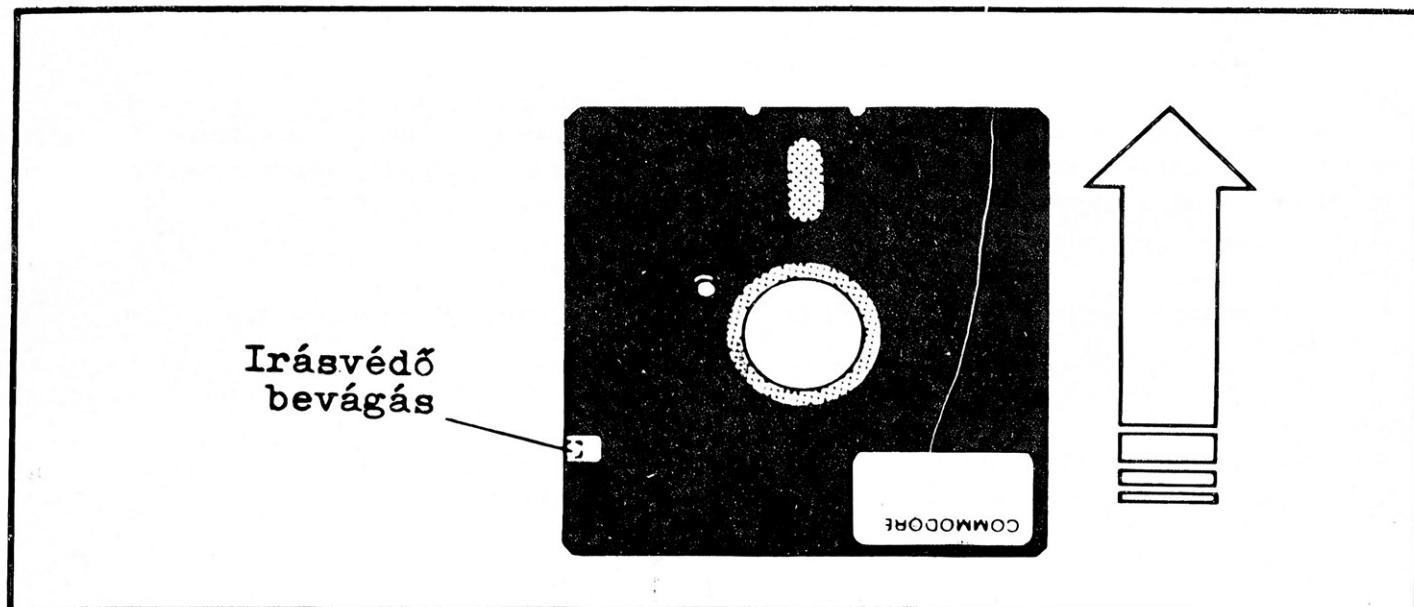
Bármelyik „soft sector” formátumú, egyszeres, vagy kétszeres felírási sűrűségű („Single-Density”, „Double-Density”) mini-floppy lemez használható (a dupla sűrűségű az ajánlott). A HUB központosító gyűrűs lemeztípusok nem ajánlottak.

A tárolt adatok törlését, átírását az írásvédő bevágás leragasztásával előzhetjük meg. Felírás esetére a leragasztást el kell távolítani!

A lemezre ragasztott címkére csak speciális (puha) íróeszközzel írjunk!

A mágneslemezt a tárolóegységbe helyezés előtt igazítsuk meg úgy, hogy tokjában középpontosan helyezkedjen el. A lemez berakása – a tárolóegység mechanikájától függően – az alábbiak szerint történik:

1. A zöld LED-es mechanikába csúsztassuk be a lemezet, amíg kis kattanással helyére ugrik, majd könnyedén, de határozottan csukjuk rá a fedelet.
 2. A piros LED-es mechanikába a lemez becsúsztatása után a fedél két lépésben zárható le.
- A lemez kivételéhez mindkét típusnál nyissuk ki a fedelet, és a lemezt könnyedén húzzuk ki a nyílásból.



2. ábra A mágneslemez behelyezése a mechanikába

A HELYES MŰKÖDÉS ELLENŐRZÉSE

Az első üzembe helyezés után helyezzük be a TEST/DEMO lemezt a meghajtó egységbe, csukjuk le a fedelet és billentyűzzük be az alábbi parancsokat:

- BASIC 3.0 verziónál:
LOAD „PERFORMANCE”, 8
RUN
- BASIC 4.0 verziónál:
DLOAD „PERFORMANCE”
RUN

A számítógép betölti és lefuttatja a lemezen levő teszt (ellenőrző) programot. A teszt program „worst case” (legrosszabb eset) adat- és időrögzítési feltételekkel ellenőrzi a mágneslemez tároló funkcióit.

A program kiírja a képernyőre a lehetséges opciókat és instrukciókat, illetve az egyes tesztek eredményeit. A tesztprogram hibátlan lefutása garantálja a berendezés normál feltételek melletti biztonságos működését. Ha valamelyik teszt hibát jelez, forduljunk a szervizhez!

2. FEJEZET A MÁGNESLEMEZES RENDSZER

A MÁGNESLEMEZ

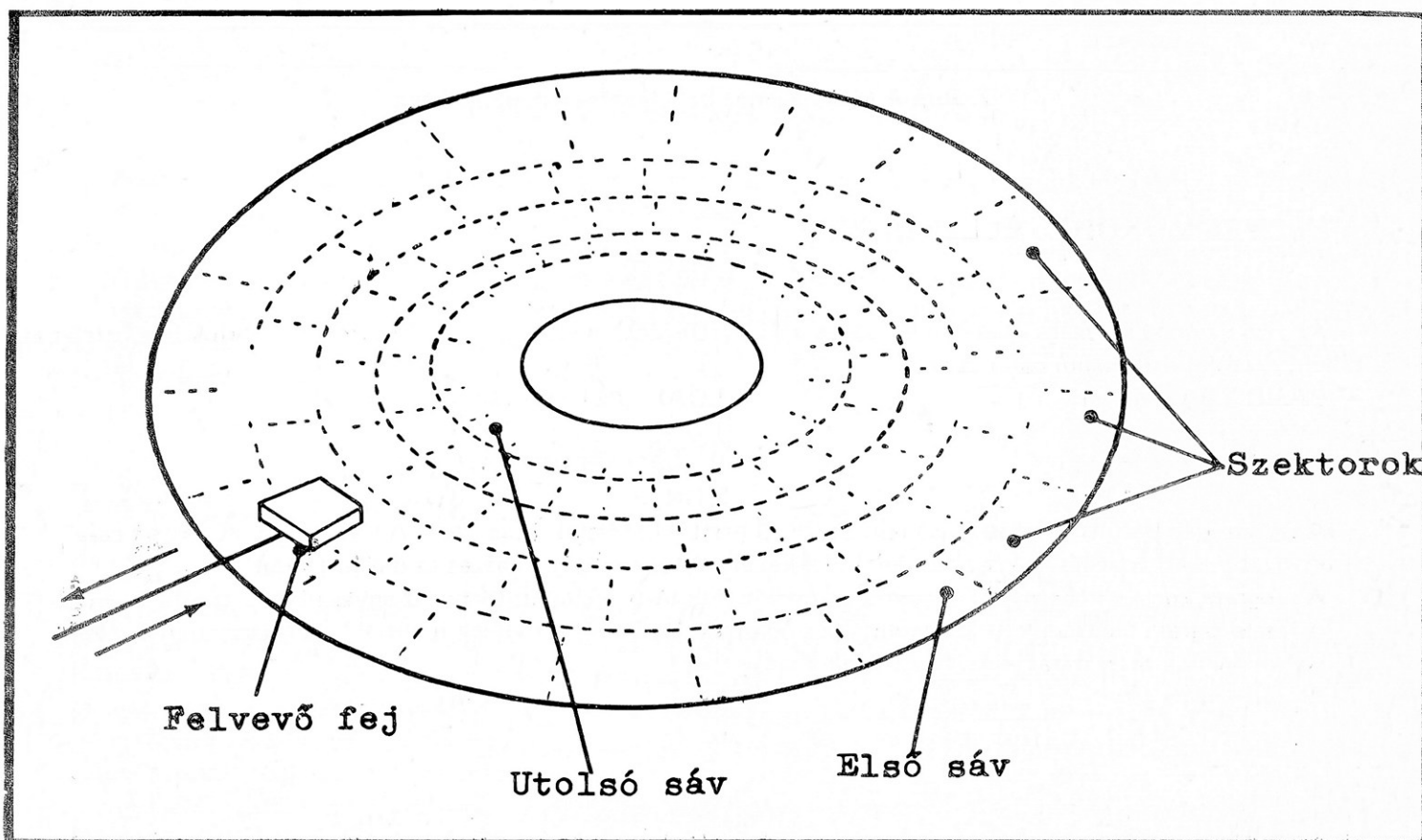
Egy mágneslemezen sok ezer (1 066 496) byte tárolható. A bájtok programutasítások, szövegek karakterei, vagy tetszőleges adatok lehetnek. A mágneslemeze az információk a lemezfelület mágneses mezőinek megváltoztatásával írhatók fel, a mágnesszalaghoz hasonló módon.

A mágnesszalagot egy adat beolvasásához a szalag elejétől kezdve kell leolvasni. Ezt nevezzük szekvenciális (soros) hozzáférésnek.

Egy lemezegység közvetlenül a kívánt pozícióra állhat és leolvashatja a kívánt adatot. Ezt véletlen, vagy közvetlen hozzáférésnek nevezzük.

(A Commodore lemezegység szekvenciális hozzáféréssel működhet, de a leolvasást a lemez bármelyik pozícióján kezdheti.)

A lemezen az adatokat sávokon („track”) tároljuk. A sávok szektorokból állnak. Minden szektor 256 bájtot tartalmaz. Ebből az első kettő a rendszer számára szükséges információkat tartalmazza, így szektoronként 254 bájtot áll a felhasználó rendelkezésére.



3. ábra A mágneslemezen levő sávok és szektorok

A logikai összetartozó adatok rendezett halmazát fájlnek („file”) nevezzük. Egy fájl kisebb logikai egységekre, rekordokra, változókra bontható.

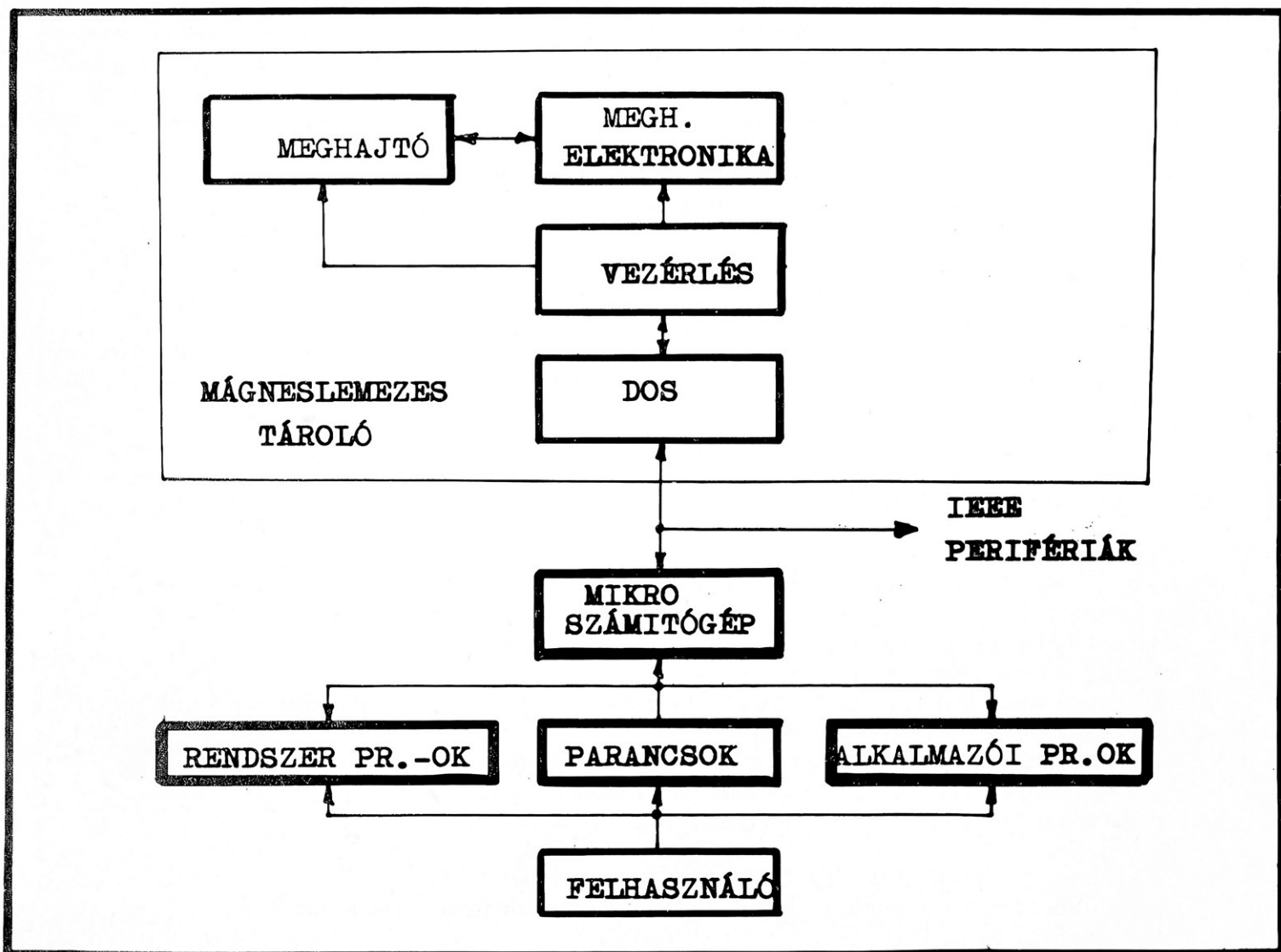
Egy lemezen tárolt fájl (lemez-fájl) tetszőleges méretű (max. 1 066 496) bájt lehet. (Az adatok fizikailag a lemezsávokon, szektorokban tárolódnak!)

A LEMEZ OPERÁCIÓS RENDSZER (DOS)

A Commodore SFD 1001 mágneslemez tároló saját mikroprocesszorral, RAM és ROM tárral rendelkező intelligens periféria, amely megkíméli a számítógépet (és programozót) az adatkezelés bonyolult részleteivel való foglalkozástól.

A fájlok kezelésével kapcsolatos részfeladatokat, az aktuális sáv és szektor nyilvántartását, írását, olvasását egy, a mágneslemez egységben állandóan futó programrendszer, a lemez operációs rendszer (DOS: Disk Operating System) végzi, amelynek parancskészlete a számítógép flexibilis BASIC utasításrendszerével magas szintű fájl kezelési lehetőségeket biztosít a felhasználó számára.

Az adatátvitelt a billentyűzetről beírt, vagy programban szereplő paranccsal kezdeményezhetjük. A számítógép által a lemez egységhez küldött parancsokat a DOS értelmezi, és a szükséges műveleteket önállóan végrehajtja.



4. ábra A mágneslemez rendszer blokkdiagramja

A DOS egyidejűleg öt lemez fájlhoz való hozzáférést tesz lehetővé. A Commodore BASIC tíz megnyitott fájl tud egyidejűleg kezelni (például öt lemez fájl, és öt egyéb — nyomtató, modem stb. — fájl).

A DOS a lemez egy részét a lemezen tárolt fájlokról szóló információ tárolására használja. Itt található a diszken levő fájlok neveinek katalógusa a „*Directory*”. Ugyancsak itt tárolja a DOS a foglalt és a szabad szektorok listáját a „szabad helyek térképét” (*BAM: Block Availability Map*).

A BAM-ot a mágneslemez első használata előtti formátumozásnál hozza létre és írja fel a lemezre a DOS. Egy formátumozott lemeznek az egységbe helyezésekor a DOS automatikusan átmásolja a BAM-ot az egység memóriájába. Egy fájl felírásakor, módosításakor, törlésekor a DOS megfelelően módosítja a memóriában és a lemezen levő BAM-ot.

CSATORNA, EGYSÉG ÉS FUNKCIÓKÓD MEGADÁSA

A számítógép és a perifériák közötti adatátvitelhez egy adatátviteli utat, „csatornát” kell megnyitni a számítógép és a periféria között. A mágneslemez tároló esetében ezt a *DOPEN #* BASIC utasítással, vagy paranccsal lehet megtenni. Az adatátvitel befejezésekor a megnyitott csatornát *DCLOSE #* utasítással le kell zárni.

A csatorna sorszámát *0-255* között adhatjuk meg. A csatorna sorszámát, „logikai fájl szám”-nak is szokás nevezni. A logikai fájl számot a *DOPEN #*, *DCLOSE #*, *INPUT #*, *GET #*, *RECORD #*, *PRINT #* utasításoknál a fájl névvel és a fizikai eszköz számmal együtt adjuk meg.

A számítógéphez kapcsolt perifériák saját fizikai eszköz (vagy *egység*) számmal rendelkeznek. A mágneslemez egységek egység száma *8-15* között lehet. Az egység számot parancs paraméterként adjuk meg a hozzáférni kívánt fizikai eszköz azonosítására.

A Commodore mágneslemez egységek sorszámát gyárilag 8-ra állítják be. Megváltoztatása a *C) Függelék* szerint, vagy 6. fejezetben leírtak szerint lehetséges.

Ha az egység számot nem adjuk meg, a DOS 8-as egység számot feltételez (default egység szám”).

A logikai fájl számon és az egység számon kívül *lehetőség van 0-15 közötti „másodlagos címek” megadására is*. Ezek tulajdonképpen *funkció kódok*, amelyekkel megadjuk a lemezegységnek a végrehajtandó műveletet.

A *0* címet (DLOAD-dal) programfájl beolvasására,

az *1* címet (DSAVE-vel) programfájl felírására használjuk.

A *2-14* címeket adatfájlok elérésére alkalmazzuk.

A *15* címet speciális „parancs csatorna” címként használjuk (OPEN-nel), speciális lemezegység műveletek végrehajtásához és a lemezegység állapotinformációinak lekérdezéséhez.

ÖSSZEVONT FÁJLNÉV MEGADÁS

A mágneslemez parancsokban szereplő *fájl név paraméter megadásánál összevont megadást is alkalmazhatunk*. Erre a kérdőjel (?) és a csillag (*) karaktereket használhatjuk fel.

A kérdőjellel jelölt pozícióban tetszőleges karakter szerepelhet, így például a

K? S.SRC fájl névvel megadott parancs valamennyi

három karakter hosszúságú, K-val kezdődő, és S-sel végződő azonosítójú, SRC típusú fájlra vonatkozik. (Pl.: KIS.SRC, KOS.SRC, KAS.SRC, KKS.SRC stb.).

A csillag egy karaktorsorozat végén a további karakterek elhanyagolását eredményezi, így például a PR*.SRC minden PR-rel kezdődő, tetszőleges hosszúságú azonosítójú SRC típusú fájlra vonatkozik.

A kérdőjel és a csillag kombináltan is alkalmazható. Például:

??? PUT*.SRC

Az összevont megadást a RENAME, DSAVE parancsoknál ne használjuk!

A DLOAD és DOPEN parancsoknál alkalmazva a legelső megfelelő fájlra fog vonatkozni.

RÖVIDÍTETT PARANCS MEGADÁSA

A parancsokat *rövidített formában* is megadhatjuk. Elegendő a más parancsoktól, vagy BASIC kulcsszavaktól való megkülönböztetéshez szükséges számú karaktert megadni. A rövidítés utolsó karakterét a SHIFT billentyűvel együtt kell megadni. Például:

CATALOG helyett cA (c, és SHIFT/A)

PRINT helyett pR (p, SHIFT/R)

(Program listázásnál a teljes parancs listázódik!)

VÁLTOZÓK HASZNÁLATA PARANCSOKNÁL

A programban szereplő *parancsok paramétereit* nemcsak konstansokkal (explicit formában), hanem *változókkal is megadhatjuk*. Például:

DOPEN # 1, d1, „Fizetés” helyett

DOPEN # 1, (dN), (A\$) is írható.

Billentyűzetről közvetlenül kiadott parancsoknál a változókkal történő megadás nem alkalmazható!

3. FEJEZET KÖZVETLEN DOS PARANCSONK

A közvetlen (direkt) DOS parancsokat általában közvetlenül a billentyűzetről kiadva hajtjuk végre, de BASIC program utasításaiként is megadhatók.

FORMAI ELŐÍRÁSOK

1. A parancsokat és kiegészítő kulcsszavakat nagybetűvel írjuk.
2. Az idézőjelben levő adatok változókat jelölnek, amelyet a felhasználó határoz meg. (Az idézőjeleket is meg kell adni!)
3. A szögletes zárójelben levő adatok opcionális paramétereket **jelölnek**, amelyek megadásáról a felhasználó dönt.
4. A szögletes zárójelen belüli aláhúzott részeket az opcionális paraméterek használatakor pontosan a leírt karakterekkel kell megadni.
5. A hegyes zárójelben szereplő változókat a felhasználó határozza meg.

A formai előírások azonosak a BASIC 4.0 verzióéval. (Korábbi BASIC verziók megfelelő parancsait az *A) Függelék tartalmazza.*)

SZIMBÓLUMOK

Az alábbiakban ismertetjük a parancsok leírásával leggyakrabban előforduló szimbólumokat.

SZIMBÓLUM	PÉLDA	FUNKCIÓ
"< név > "	"JÁTÉK"	Egy lemez, vagy fájl neve.
"< fájl név > "	"SAKK"	Egy fájl neve.
"< forrás fájl > "	"REGI"	Egy másolandó, vagy átnevezendő fájl eredeti neve.
"< cél fájl > "	"UJ"	Egy átmásolt, vagy átnevezett fájl új neve.
< egység >	8	Egy periféria címe, egység száma (lemezegységénél: 8–15).
< meghajtó >	0	A mágneslemez tároló meghajtó egységének száma (mindig 0).
< forrás meghajtó >	0	Az eredeti fájlt tartalmazó meghajtó egység száma (mindig 0).
< cél meghajtó >	0	Az új (átmásolt) fájl tárolására kijelölt meghajtó egység száma (mindig 0).
< azonosító >	BB	Két karakteres lemez azonosító.
< fájl szám >	3	Logikai fájl (csatorna) száma (0–255).
< cím >	15	„Másodlagos cím”: funkció kód (0–15).
< változó >	A\$	Egy BASIC változó.
< rekord hossz >	128	Fizikai rekord hossz (relatív fájlnál).

Ha nem adunk meg egység számot, akkor a 8-as című egység, ha nem adjuk meg a meghajtó számát, úgy a 0-ás meghajtó választódik ki („default”).

MÁGNESLEMEZRE VONATKOZÓ PARANCSONK

HEADER

A mágneslemez első használat előtti inicializálására, formátumozására szolgál. Az adatblokkokat nullával tölti ki, felírja a sáv- és szektorcímekeket, a lemezkatalógust (Directory), és az üres blokkok térképét (BAM). A lemeze esetleg előzőleg felírt információkat törli. Emiatt a parancs kiadásakor a rendszer megkérdezi, hogy:

"Are You Sure?" (Biztos Ön Benne? — hogy inicializálni óhajt?)

Csak Y (Yes — Igen) válasz esetén hajtódik végre a parancs.

Általános forma:

HEADER "<név>" [onu <egység>] [,i< azonosító>]

Például a

HEADER "JÁTÉKOK" onu 8,i01

parancs a 8-as egységcímű mágneslemezes egységben levő mágneslemezt inicializálja, és a JÁTÉKOK nevet és a 01 azonosítót írja fel rá.

A lemez neve maximum 16 karakter hosszú lehet, az azonosító két karakter kell legyen.

Új lemezeknél kötelező a parancs használata!

Rövidített forma: hE

DIRECTORY, vagy CATALOG

Ezek a parancsok kilistázzák a lemezen levő fájlok katalógusát a képernyőre.

A listázás a szóköz billentyűvel megállítható és újraindítható.

A RUN/STOP billentyű törli a listát.

("Üzleti" billentyűzetnél a listázás leállítására a kettőspont, újraindítására a 3, 6, 9 billentyűk egyike, lassítására a balra mutató nyíl szolgál. A kettőspontot lenyomva tartva a RUN/STOP-pal egyenként lehet a sorokat kiíratni.)

DIRECTORY [onu <egység>]

vagy:

CATALOG [onu <egység>]

Például a

CATALOG d0 onu 9

parancs a 9-es mágneslemezes tárolóban levő lemez katalógusát listázza ki.

A lista első sorában („fejléc”) a meghajtó egység száma, a lemez neve, azonosítója és a DOS verziószáma jelenik meg.

Például: 0 "JÁTÉKOK"012A

A további sorokba a fájlok blokkhossza, neve és típusa íródik ki.

Például: 1 "TEVE BOOT" PRG
 33 "TEVE" PRG

.

.

.

Fájl típusok: PRG: program
 SEQ: szekvenciális
 REL: relatív
 USR: felhasználói

A lista utolsó sora a szabad (üres) blokkok számát adja meg.

Például: 11 BLOCKS FREE

A CMD parancs használatával a katalógus nyomtatóra is kiírható. (Lásd a nyomtató kézikönyvében!)

Rövidített forma: diR vagy cA

COLLECT

A programhiba miatt megnyitva maradt fájlok, valamint az összes felhasználói fájl törlését hajtja végre, és a lemez tartalmának megfelelő új BAM-ot ír föl.

```
COLLECT [onu <egység>]
```

Például a

```
COLLECT
```

Újragenerálja a lemez BAM-ját, és törli az összes megnyitott és USR típusú fájlt.

Rövidített forma: coL

BACKUP

Átmásolja a teljes "forrás" lemezt a "cél" lemezre.

```
BACKUP d <forrás megh. > [onu <egység>] TO d <cél megh. > [onu <egység>]
```

Például a

```
BACKUP d0 TO d0 onu 9
```

parancs a 8-as egységben levő lemezről másolatot készít a 9-es egységben levő lemezre.

Rövidített forma: bA

FAJLOKRA VONATKOZÓ PARANCSONOK

DSAVE

A számítógép RAM-jában levő programot a lemezre írja ("menti") a megadott nevű PRG típusú fájlba. Ha a fájl-nak első karaktereként -t adunk meg, úgy az azonos néven már esetleg meglévő fájl felülírósik; egyébként hiba-jelzés generálódik, és a program nem kerül felírásra.

```
DSAVE [onu <egység>], "[ <fájlnév >"]
```

Például:

```
DSAVE, "MATRIXMUV"
```

Rövidített forma: dS

DLOAD

Betölti a megadott lemez-fájlt a számítógép RAM-jába, ugyanakkor lezárja az összes megnyitott fájlt és a parancs-csatornát.

```
DLOAD [onu <egység>], "[ <fájlnév >"]
```

Például:

```
DLOAD d0 onu 9, "SPIRAL"
```

Rövidített forma: dL

Megjegyzés:

1. A DLOAD "*" parancs a katalógus első helyén levő fájlt tölti be. Ha ez nem PRG típusú, hibajelzés íródik ki.

Automatikus indítás ("Autostart")

A BASIC 4.0 verziónál a SHIFT és a RUN/STOP billentyűk lenyomására a 8-as egységben levő lemez katalógus szerinti első fájlja betöltődik a számítógép RAM-jába, és végrehajtódik. (Ha az első fájl nem PRG típusú, hibaüzenet íródik ki.)

(Ezt a lehetőséget pl. egy menü (választéklista) megadására lehet használni.)

RENAME

Egy, a lemezen levő fájl nevének megváltoztatására szolgál. A fájl tartalma nem változik. A megadott új névvel azonos nevű fájl nem lehet a lemezen. Összevont névmegadás nem megengedett.

RENAME [onu <egység>], "<eredeti név>" TO "<új név>"

Például:

```
RENAME "AUTO" TO "TRABANT"
```

Rövidített forma: reN

COPY

A kiválasztott program, – vagy szekvenciális fájlokat a megadott néven átmásolja a megadott lemezre. A forrás lemezre a forrás fájl nevével azonos nevű fájl nem másolható! (Egy lemezen több azonos nevű fájl nem lehet.)

COPY [onu <egység>], "<forrás név>" TO [onu <egység>], "<cél név>"

Például:

```
COPY, "ADAT1" TO d0 onu 9, "ADAT 1"
```

Megjegyzések:

1. Összevont névmegadás megengedett, így egy paranccsal több fájl is másolható.
2. A "cél név" helyére " " jelet írva a fájlok az eredeti nevükkel másolandók át.

Rövidített forma: coP

CONCAT

Hozzákapcsolja a cél fájl végéhez a forrás fájlt. Az eredeti forrás fájl is megmarad. Csak szekvenciális (SEQ) fájloknál használható.

Az összevont névmegadás nem megengedett.

CONCAT [onu <egység>], "<forrás fájl>" TO [onu <egység>], "<cél fájl>"

Például:

```
CONCAT, "KEDD" TO d0, "HETFO"
```

Rövidített forma: conC

SCRATCH

Törli a megadott fájlokat. Összevont névmegadás megengedett, de körültekintéssel alkalmazzuk! A parancs az *Are You Sure?* üzenetre adott "Y" válasz esetén hajtódik csak végre!

SCRATCH [onu <egység>], "<fájlnév>"

Például:

```
SCRATCH d0 onu 9, "JATEK"
```

Rövidített forma: sC

4. FEJEZET A DOS KEZELÉSE BASIC-BŐL

Az alábbiakban a szekvenciális és relatív adatfájlok kezelésére szolgáló parancsok kerülnek ismertetésre. A parancsok közvetlenül, vagy BASIC programutasításokként használhatók.

DOPEN

Egy fájl megnyitását, hozzáférésre való előkészítését végzi.

1. Szekvenciális fájlok megnyitása

Az olvasásra megnyitott fájl nevének létező fájlra kell hivatkoznia. Ellenkező esetben, vagy ha már megnyitott fájlra hivatkozunk, hibaüzenet generálódik.

Ha írásra nyitunk meg nem létező fájlt, úgy a megadott névvel új fájl jön létre.

Megnyitás olvasásra: (Read)

DOPEN # <fájl szám> [onu <egység>], " <fájlnév> ", R

Például: DOPEN # 1, "LELTAR", R

Megnyitás írásra: (Write)

DOPEN # <fájl szám> [onu <egység>] L, " <fájlnév> ", W

Például: DOPEN # 3, "ALLOESZKOZ", W

2. Relatív fájlok megnyitása

A relatív fájlok minden rekordja tetszőleges sorrendben érhető el. (Az elérni kívánt rekord számát a RECORD # paranccsal adhatjuk meg.)

A relatív fájlokat írásra és olvasásra egyidejűleg lehet megnyitni. Egy már létező fájl megnyitásakor a rekord hosszát nem szükséges megadni.

Általános forma:

DOPEN # <fájl szám> [onu <egység>], " <fájlnév> " [, L <rekord hossz>]

Például: DOPEN # 6, "FIZETES", L80

A DOS által egyidejűleg megnyitott fájlok száma függ azok típusától. Maximálisan 5 SEO, vagy 1 REL és 3 SEO, vagy 2 REL és 2 SEO, vagy 3 REL típusú fájl lehet egyidőben megnyitva.

Rövidített forma: dO#

APPEND

Egy már létező szekvenciális fájlt nyit meg írásra úgy, hogy a PRINT # utasítással felírt adatok a fájl végére kerülnek.

APPEND # <fájl szám> [onu <egység>], " <fájlnév> "

Például: APPEND # 5, "CÍMEK"

Rövidített forma: aP #

DCLOSE

Lezárja a megadott, előzőleg megnyitott fájlt. Fájl szám nélkül kiadva valamennyi megnyitott fájlt lezárja.

DCLOSE # < fájl szám > lezárja a megadott fájlt
DCLOSE lezárja az összes megnyitott fájlt

Például a

DCLOSE # a 25-ös logikai fájl számmal megnyitott fájlt lezárja

Rövidített forma: dC #

Ha a lemez kivétele előtt nem zárjuk le a megnyitott fájlokat, elveszhetnek a tárolt adatok!

PRINT

Adatok beírására szolgál egy előzőleg már (írásra) megnyitott fájlba.

PRINT # < fájl szám > [, változó] [, változó] . . . [, változó]

A változók elválasztására vesszőt, pontosvesszőt, vagy szóközt használunk. A karakteres változók közé nem szükséges elválasztó karaktert írni. A legelső változó előtt vesszőnek kell állnia!

Az egy PRINT # utasításban szereplő változók tartalma a lemezre egyetlen egyesített karakterfűzéként íródik fel, amelyet kocsi vissza (CR) és soremelés (LF) vezérlő karakter zár le. A BASIC 4.0 verziónál a soremelés karakter csak a 127-nél nagyobb fájl számú utasításoknál íródik fel!

Rövidített forma: pR #

INPUT

Adatokat olvas be egy olvasásra megnyitott fájlból, és rendeli értéként az utasításban szereplő változókhoz,

INPUT # < fájl szám >, < változó > [, < változó >] . . . [, < változó >]

A változók előtt vesszőnek kell állnia.

Például: INPUT # 8, B\$, C\$

Egy fájlban szereplő adat max. 80 karakter hosszúságú karakterfűzék, vagy számadat lehet, amelyet kocsi vissza (CR), vagy soremelés (LF), vagy vessző, vagy null-karakter (CHR\$(0)) határol.

Rövidített forma: iN #

GET

Egy olvasásra megnyitott fájlból egyetlen adatkaraktert olvas be és rendel a megadott változóhoz.

GET # < fájl szám >, < változó >

Például: GET # 9, B

Karakteres adat beolvasásakor a helyes adatkezeléshez a bináris nullát (null-karaktert) CHR\$(0) formára kell átalakítani.

Például: GET # 1 1,C\$: IF B\$ = " " THEN B\$ = CHR\$(0)

RECORD

Relatív típusú fájlknál a közvetlenül elérni kívánt rekord (és azon belüli bájtt) kijelölésére szolgál. Ha a bájtt pozíciót nem adjuk meg, úgy a rekord első bájttja jelölődik ki.

RECORD # < fájl szám >, < rekord szám > [, < bájtt pozíció >]

Például:

RECORD # 10,20,15 után a GET # 10, A\$

a 10-es számmal megnyitott fájl 20. rekordjának 15. karakterét rendeli az A\$ változóhoz.

Rövidített forma: reC #

5. FEJEZET RELATÍV FAJLOK

A relatív fájlkezelési módszer lehetővé teszi az adatrekordok közvetlen elérését. Ehhez az adott rekord fájlban belüli sorszámát (a fájl kezdetéhez viszonyított relatív pozícióját) kell megadni. Ez a módszer jelentősen lecsökkenti a rekordok elérési idejét.

A rekordok hossza egy fájlban belül azonos, és a fájl létrehozásakor kell (1–254 bájt között) megadni. A rekordok fizikailag adatblokkokban (szektorokban) tárolódnak. A rekordokat CHR\$ (13) kódú kocszi vissza (CR) karakter zárja le. Az üres rekordok első bájtja FF, a többi 00 értékű. A részlegesen kitöltött rekordok üres bájtjai 00 értékűek.

Az adatblokk első két bájtja a következő adatblokk sáv- és szektorcímét tartalmazza. A fájl utolsó adatblokkjában a sávcím 00 értékű. Egy adatblokk 254 adatbájtot tartalmaz. Minthogy a rekordok az adatblokkokban folyamatosan helyezkednek el, így az 1, 2, 127, 254-től eltérő hosszúságú rekordok két adatblokkba szétosztva is elhelyezkedhetnek.

Egy fájl adatblokkjainak címeit a DOS külön szektorokban, úgynevezett szektor-táblákban ("side sector") tárolja. Hat-hat ilyen szektor-tábla alkot egy-egy csoportot. Egy szektortábla 120 adatblokk címét és a csoport hat szektor táblájának a címét (sajátját is) tartalmazza.

Az SFD 1001-es egységnél egy "szuper-szektor-tábla" 127 hatos szektor-táblázat csoport első szektorának címét tartalmazhatja. Így egy fájl elvileg maximum $127 \times 6 \times 120 \times 254$ bájt hosszúságú lehet.

Az adat és tábla-szektorok részletes felépítését a 7. fejezet ismerteti.

RELATÍV FAJL LÉTREHOZÁSA

Egy relatív fájl első megnyitásakor a programozónak ajánlatos egy inicializálási szekvenciát végrehajtania. Ez a fájl megnyitása után a használni kívánt legmagasabb sorszámú rekordba való írással valósítható meg. Ezáltal a DOS lefoglalja a lemezen a teljes fájl helyét, és a későbbi felírásoknál gyorsabb hozzáférést tesz lehetővé.

Például a

```
DOPEN # 1, d0, "FILE 3", L80
RECORD # 1, 200
PRINT # 1, CHR$(255)
DCLOSE #
```

utasításokkal egy 200 rekordot tartalmazó, 80 bájt rekordhosszúságú (még üres) adatfájlt hozhatunk létre. (A fenti eljárás nem kötelező, de gyorsabb hozzáférést tesz lehetővé.)

RELATÍV FAJL KIBŐVÍTÉSE

Egy létező fájl kibővítését a létrehozásához hasonló módon érhetjük el. A RECORD # utasításban az eredeti legnagyobb rekord-sorszám helyett az új, legutolsó rekord sorszámát kell megadni.

Egy már létező fájl megnyitásakor a rekordhosszat nem szükséges megadni, de ha megadjuk, úgy meg kell egyeznie a fájl létrehozásakor megadott hosszal.

(A DOS a fájlok létrehozásakor, kibővítésekor az aktuális szektor-táblázatokat automatikusan létrehozza, illetve módosítja. A programozónak ezzel nem kell foglalkoznia.)

ADATOK KEZELÉSE

Íráskor vagy olvasáskor a RECORD # utasítást használjuk az elérni kívánt rekord, és az azon belül elérni kívánt bájtpozíciójának megadására.

Például a

```
DOPEN # 1, d0, "FILE 3"  
RECORD # 1, 25, 10  
PRINT # 1, "BUDAPEST"  
DCLOSE # 1
```

utasítások a 8-as ("default") egységszámú tárolóban levő "FILE 3" nevű fájl 25. rekordjának 10. bájtpozíciójától kezdődően a (10. helyre még nem!) felírjuk a BUDAPEST karaktereket és a 19. bájtpozíciójába CHR\$(13), kocsivissza (CR) lezáró karakter íródik.

A rekord és a bájtpozíció paraméter változóval is megadható.

Például a

```
DOPEN # 1, d0, "FILE 3"  
R = 25 : B = 14  
RECORD # 1, R, B  
INPUT # 1, A$  
DCLOSE # 1
```

utasításokat az A\$ változóba a BUDAPEST karaktereket olvassák be.

Ha egy relatív fájl rekordjait szekvenciálisan akarjuk kezelni, a RECORD utasítást nem kell megadni. Ekkor a fájl megnyitásakor az első rekord lesz hozzáférhető, és minden PRINT # (ill. INPUT #) utasítás után a következő rekord jelenik meg.

6. FEJÉZET A DOS PROGRAMOZÁSA

A mágneslemez egység saját vezérlővel és RAM tárral rendelkezik. Egy DOPEN # BASIC utasítással történő fájl-megnyitáskor a DOS egy vagy két adatpuffert és egy munkaterületet jelöl ki a megnyitott adatcsatornához.

Egy OPEN utasításban megadott 15-ös másodlagos címmel (funkció kóddal) a felhasználó egy "parancs csatornát" nyithat meg, és PRINT # utasításokkal parancsokat küldhet ki a DOS-nak. A parancs csatornát CLOSE utasítással kell lezárni.

```
OPEN 15, 8, 15
PRINT # 15, ...
:
:
CLOSE 15
```

Az alábbiakban csak a PRINT # utasításokat tárgyaljuk, az OPEN és CLOSE utasításokat nem írjuk külön ki.

SZIMBÓLUMOK

cs:	csatornaszám, a másodlagos címmel azonos
mh:	meghajtó száma (0)
s:	sávszám (1–154)
sz:	szektorszám (0–112)
p:	puffer mutató pozíciója
ac:	alacsonyabb címbyte CHR\$(n) formában
mc:	magasabb címbyte (CHR\$(n) formában
k:	karakterek száma (CHR\$(n) formában (1–34)
a:	az aktuális adat CHR\$(n) formában
x:	a Felhasználói Táblán belüli index

PARANCSOK

"B-A:"mh;s;sz	blokk lefoglalás	("Block-Allocate")
"B-F:"mh;s;sz	blokk felszabadítás	("Block-Free")
"B-R:"cs;mh;s;sz	blokk olvasás	("Block-Read")
"B-W:"cs;mh;s;sz	blokk írás	("Block-Write")
"B-E:"cs;mh;s;sz	blokk végrehajtás	("Block-Execute")
"B-P:"cs;p	puffer pozícionálás	("Puffer-Pointer")
"M-W"acmcka	RAM írás	("Memory-Write")
"M-R"acmc	RAM olvasás	("Memory-READ")
"M-E"acmc	RAM végrehajtás	("Memory-Execute")
"Ux:"cs;mh;s;sz	felhasználói parancsok	("USER")

BLOKK LEFOGLALÁS

Például a

```
PRINT # 15,"B-A:"0;10;5
```

utasítás a lemez 10. sávjának 5. szektorát (blokkját) lefoglalja, a BAM-ba mint foglalt szektort jegyzi be.

Ha már lefoglalt blokkot próbálunk lefoglalni, hibajelzés generálódik. A hibajelzés a parancs csatornán ("hiba-csatorna") olvasható ki.

```
OPEN 15,8,15
```

```
INPUT # 15,HS,HU$,S,SZ
```

```
CLOSE 15
```

ahol: HS = Hiba száma,

HU\$ = Hiba üzenet,

S = Sáv

SZ = Szektor

A sáv és szektor a következő szabad blokkra mutat!

BLOKK FELSZABADÍTÁS

Például a

```
PRINT # 15,"B-F:"0;10;15
```

utasítás felszabadítja – a BAM-ba mint szabad blokkot jegyzi be – a lemez 10. sávjának 5. szektorát.

Figyelem: ennél a parancsnál különös gondossággal járjunk el, nehogy a BAM, vagy "Directory" valamelyik blokkját "szabadítsuk fel"!

BLOKK OLVASÁS

Például a

```
PRINT # 15,"B-R:"5;0;18;1
```

utasítás az 5. csatorna puffterületére olvassa be a lemez 18. sávjának 1. szektorát (blokkját).

A pufferból GET #, vagy INPUT # utasítással olvasható ki az adatblokk.

BLOKK ÍRÁS

Például a

```
PRINT # 15,"B-W:"5;0;18;1
```

utasítás az 5. csatorna puffertjét a lemez 18. sávjának első szektorába írja.

BLOKK VÉGREHAJTÁS

Például a

```
PRINT # 15,"B-E:"6;0;1;10
```

utasítás egy, a felhasználó által (gépi kódban, a lemez vezérlő számára) írt programot beolvas a lemez 1. sávjának 10. szektorából a 6. csatorna adatpufferjába, és végrehajtja azt. (A program utolsó utasítása RTS kell legyen!)

PUFFER POZICIONÁLÁS

Például a

```
PRINT # 15,"B-P:"2;10
```

utasítás a 2. adatcsatorna puffertjének mutatóját ("pointer") a 10. pozícióra állítja. (A mutató értéke 0–255 között változhat.)

RAM ÍRÁS

Maximum 34 bájtt írható vele a két első (alacsonyabb, majd magasabb) értékkel megadott címtől kezdődő RAM területre. Ezzel speciális rutinok tölthetők be (és hajthatók végre az "M-E", vagy "Ux:" parancsokkal.) Az egyes értékeket CHR\$(n) formában kell megadni.

Például a

```
PRINT # 15,"M-W"CHR$(128)CHR$(0)CHR$(2)CHR$(5)CHR$(8)
utasítás a 0 128 (decimális) címtől kezdve 2 adatot (5 és 8) ír be a RAM-ba.
```

EGYSÉGCÍM ÁTÍRÁSA

A lemezegység fizikai címe szoftveresen az alábbi módon írható át:

```
OPEN 15,rc,15
```

```
PRINT # 15,"M-W"CHR$(12)CHR$(0)CHR$(2)CHR$(uc + 32)CHR$(uc + 64)CLOSE 15
```

ahol rc = régi cím, uc = új cím.

(A hardveres címváltoztatást a B) Függelék ismerteti!)

RAM/ROM OLVASÁS

Az utasításban megadott (alacsonyabb, majd magasabb) RAM (vagy ROM) címről olvas egy bájtot a "hibacsatorna" pufferjébe. Ez egy GET # utasítással olvasható ki a (15-ös) hibacsatornából.

Például a

```
PRINT # 15,"M-R",CHR$(128)CHR$(0)
a RAM0 128 (decimális) című bájtyát olvassa ki.
```

RAM VÉGREHAJTÁS

Végrehajtja a megadott címen kezdődő programot.

Például a

```
PRINT # 15,"M-E"CHR$(128)CHR$(0)
a 0 128 (decimális) címre adja a vezérlést. A programnak RTS utasítással kell befejeződnie.
```

FELHASZNÁLÓI PARANCSONK

Az U1-U9 parancsok UA-UI módon is jelölhetők, az U: helyett pedig UJ is írható.

U1, vagy UĀ

Egy teljes 256 byte-os adatblokkot (a sáv és szektor mutatóval együtt) olvas be a pufferba. Például a

```
PRINT # 15,"U1:"5;0;18;10
utasítás a lemez 18. sávjának 10. szektorát olvassa be az 5. csatorna pufferjébe.
```

U2, vagy UB

Egy puffer tartalmát írja fel a lemezre. Például a

```
PRINT # 15,"U2:"5;0;18;10
utasítás az 5. adatcsatorna pufferjének tartalmát felírja a lemez 18. sávjának 10. szektorába.
```

U3–U9, vagy UC–UI

Ezek a parancsok ugró (JUMP) vezérlésátadó utasítást hajtanak végre az alábbi *hexadecimális* címekre:

U3, UC	:	1300
U4, UC	:	1303
U5, UE	:	1306
U6, UF	:	1309
U7, UG	:	130C
U8, UH	:	130F
U9, UI	:	10F0 (NMI)

U:, vagy UJ

Ez a parancs szoftver RESET funkciót eredményez, a tápfeszültség bekapcsolásához hasonló inicializálást végez.

7. FEJEZET SZEKTOR FORMÁTUMOK

A mágneslemez sávjai koncentrikusan helyezkednek el. A legkülső sáv a legkisebb (0.), a legbelső a legmagasabb (77.) sorszámú.

A szektorok száma sávonként nem egyenlő. A külső sávokon több, a belső sávokon kevesebb szektor van. Néhány szektort a DOS saját céljaira, a BAM, a Directory, a Directory Header tárolására használ.

Az alábbiakban a sávok és szektorok felosztását, és a szektorok formátumát adjuk meg.

SZEKTOR/SÁV FELOSZTÁS

SÁV SORSZÁM	SZEKTOR/SÁV
1.–39., ill. 78.–116.	29
40.–53., ill. 117.–130.	27
54.–64., ill. 131.–140.	25
65.–77., ill. 141.–154.	23

BAM FORMÁTUMOK

1. BAM BLOKK: 38. sáv 00. szektor

BÁJT	ADAT	INFORMÁCIÓ
0–1	38–03	2. BAM blokk címe
2	67	DOS 2.7 azonosító: ASCII "C"
3	00	
4	01	Nyilvántartott legkisebb sorszámú sáv
5	51	Legnagyobb sorszámú sáv (+ 1)
6–255		Szabad blokkok bit-térképe (0.–50. sávoké)

2. BAM BLOKK: 38. sáv 03. szektor

BÁJT	ADAT	INFORMÁCIÓ
0–1	38–06	3. BAM blokk címe
2	67	DOS 2.7 azonosító: ASCII "C"
3	00	
4	51	Nyilvántartott legkisebb sorszámú sáv
5	101	Legnagyobb sorszámú sáv (+ 1)
6–255		Szabad blokkok bit-térképe (51.–100. sávoké)

3. BAM BLOKK: 38. sáv 06. szektor

BÁJT	ADAT	INFORMÁCIÓ
0-1	38-09	4. BAM blokk címe
2	67	DOS 2.7 azonosító: ASCII "C"
3	00	
4	51	Nyilvántartott legkisebb sorszámú sáv
5	151	Legnagyobb sorszámú sáv (+ 1)
6-255		Szabad blokkok bit-térképe (101.-150. sávoké)

4. BAM BLOKK: 38. sáv 09. szektor

BÁJT	ADAT	INFORMÁCIÓ
0-1	39-01	Első Directory blokk címe
2	67	DOS 2.7 azonosító: ASCII "C"
3	00	
4	151	Nyilvántartott legkisebb sorszámú sáv
5	155	Legnagyobb sorszámú sáv (+ 1)
6-25		Szabad-blokkok bit-térképe (151.-154. sávoké)
26-255	00	

A szabad blokkok bit-térképe sávonként 5 bájtton tartja nyilván a szektorokat. A szabad szektort "1" érték jelzi. Egy bájt tehát 8 szektor foglaltság bitjét tartalmazza fordított sorrendben. (Például egy adott sáv első bájtjának 7. bitje a 0. szektor, 0. bitje a 7. szektor; vagy az ötödik bájt 7. bitje a 24. szektor, 0. bitje a 31. szektor foglaltságát jelzi.)

DIRECTORY (katalógus) FORMÁTUMOK

DIRECTORY HEADER: 39. sáv 00. szektor

BÁJT	ADAT	INFORMÁCIÓ
0-1	38-00	Első BAM blokk címe
2	67	DOS 2.7 azonosító: ASCII "C"
3	00	
4-5	00	
6-21		A lemez neve
22-23	160	SHIFT/SPACE karakterek
24-25		A lemez azonosítója
26	160	SHIFT/SPACE
27-28	50, 67	DOS verzió és formátum azonosítója: ASCII "2C"
29-32	160	SHIFT/SPACE
33-255	00	

DIRECTORY BLOKKOK: 39. sáv 01.–29. szektor

BÁJT	INFORMÁCIÓ
0–1	A következő Directory blokk címe
2	A fájl típusa
3–4	Az első fájl blokk címe
5–20	A fájl neve
21–22	Relatív fájl "szuper-szektor-tábla" címe
23	Rekord hossz (relatív fájlnál)
24–27	
28–29	Áthelyezési cím
30–31	A fájl blokkjainak száma
32–255	7(*32 bájt) fájl leírás (a 0–31-hez hasonlóan)

Egy Directory blokk 8 fájl leírását tartalmazza. Az utolsó blokk 0. bájtja 00 értékű.

A fájl típusok kódjai:	\$00	Üres (törölt)
	\$01	SEQ
	\$02	PRG
	\$03	USR
	\$04	REL

A fájl lezárásakor a típuskódhoz \$80 érték adódik hozzá.

FÁJL ADATBLOKK FORMÁTUMOK

PROGRAMFÁJL BLOKK

BÁJT	FUNKCIÓ
0–1	Következő program blokk címe
2–255	A program bájtjai. A fájl végét (EOF) három 00 értékű bájt jelzi.

SZEKVENCIAÁLIS ÉS (RELATÍV) ADAT BLOKK

BÁJT	FUNKCIÓ
0–1	Következő szekvenciális (relatív) blokk címe. A sávcím 00 értéke jelzi az utolsó blokkot.
2–255	Adat bájtok; az egyes adatokat CHR\$(13) zárja le.

RELATÍV FÁJL SEKTOR-TÁBLA

BÁJT	FUNKCIÓ
0–1	Következő szektor-tábla címe.
2	
3	Rekordhossz
4–5	1. szektor-tábla címe

6-7	2. szektor-tábla címe
8-9	3. szektor-tábla címe
10-11	4. szektor-tábla címe
12-13	5. szektor-tábla címe
14-15	6. szektor-tábla címe
16-255	Adatblokkok (max. 120) címei

A FÜGGELÉK DOS PARANCSKÉSZLET

A táblázat a teljes parancsot, rövidítését, a BASIC verzió és a következő sorban a parancs paramétereit sorolja föl.

	PARANCS	RÖVIDÍTÉS	BASIC VERZIÓ
1.	APPEND # < fájl szám> [onu <egység>], "<fájlnév>"	aP #	4.0
2.	BACKUP d < meghajtó> [onu <egység>] TO d < meghajtó> [onu <egység>] PRINT # 15 "d[uplicate]< cél meghajtó> = < forrás meghajtó>"	bA pR15	4.0 3.0
3.	CATALOG DIRECTORY [onu <egység>] LOAD "\$ < meghajtó>", < egység>	cA diR IO	4.0 4.0 3.0
4.	COLLECT [onu <egység>] PRINT # 15 "v[alidate]	coL pR	4.0 3.0
5.	CONCAT [onu <egys.>], "<f. név>" TO [onu <egység>], "<f.npv>"	conC	4.0
6.	COPY [onu <egys.>], "<forrás f.>" TO [onu <egys.>], "<cél fájl>" PRINT # 15 "c[opy] < megh. > : <cél fájl> = < megh. > : <forrás fájl>	coP pR	4.0 3.0
7.	DCLOSE # [<fájlszám>] CLOSE [<fájlszám>]	dC # clO	4.0 3.0
8.	DLOAD [onu <egység>], "<fájl név>" ;LOAD; ;IO; LOAD ["< meghajtó> : <fájlnév>", <egység>]	dL IO	4.0 3.0
9.	DSAVE [onu <egység>], "<fájlnév>" SAVE "[: <fájlnév>", <egység>	dS sA	4.0 3.0
10.	DOPEN # <fájlszám> [onu <egys.>], "<fájlnév>" [,R], W], L<rek. hossz>] OPEN <fájlszám>, <egység>, <cím>, "< megh. > : <f.név>" [,R], W], [L + CHR\$(<r.h.>)]	dO # oP	4.0 3.0

11.	HEADER "⟨név⟩" [onu ⟨egység⟩][,i ⟨azonosító⟩] PRINT # 15 "nlew] +⟨név⟩[,⟨azonosító⟩]"	hE pR	4.0 3.0
12.	INPUT # ⟨fájl szám⟩ , ⟨változó⟩ [,⟨változó⟩] ... [,⟨változó⟩]	iN #	
13.	RECORD # ⟨fájl szám⟩ , ⟨rekord szám⟩ [,⟨byte pozíció⟩] PRINT # 15 "p"CHR\$ (⟨f.sz. + 96⟩)CHR\$ (rek.sz.alsó b. >) CHR\$ (⟨rek.sz.felső b. >) CHR\$ (⟨byte pozíció⟩)	reC # pR	4.0 3.0
14.	RENAME [onu ⟨egység⟩,] "⟨eredeti név⟩" TO "⟨új név⟩" PRINT # 15 "r[ename] : ⟨új név⟩ = ⟨eredeti név⟩"	reN pR	4.0 3.0
15.	SCRATCH [onu ⟨egység⟩,] "⟨fájl név⟩" PRINT # 15 "s[cratch] ⟨megh. > : ⟨fájl név⟩ [,⟨megh. > : ⟨fájl név⟩] ..."	sC pR	4.0 3.0

Megjegyzés: A BASIC 3.0-ban a (3.) LOAD parancs törli a RAM-ot!
A katalógus betöltés után LIST paranccsal jeleníthető meg.

B. FÜGGELÉK EGYSÉGSZÁM (CÍM) ATÁLLÍTÁSA

A gyárilag beállított 8-as egységszámot szoftver úton a 6. fejezetben leírtak szerint lehet átírni.

A tápfeszültség bekapcsolása után a 8-as cím lesz újból érvényes.

Az egységszámot véglegesen hardver úton, az *UE1* pozíciójú integrált áramkör lábaihoz menő vezetékek megszakításával lehet megváltoztatni.

EGYSÉGSZÁM	22. LÁB	23. LÁB	24. LÁB
8	0	0	0
9	0	0	1
10	0	1	0
11	0	1	1
12	1	0	0
13	1	0	1
14	1	1	0
15	1	1	1

A "0" a vezeték változatlanul hagyását, az "1" a vezeték megszakítását jelöli.

A megszakított vezetékek újbóli átkötésével az eredeti egységszám visszaállítható.

(A vezetékek megszakítása helyett az *UE1* áramkör megfelelő lábainak kihajlítása is alkalmazható.)

C. FÜGGELÉK HIBAÜZENETEK

SZÁMA	HIBAÜZENET	JELENTÉSE
0	OK	: Nincs hiba
1		: SCRATCH végrehajtva (nem hiba!)
2-19		
20	READ ERROR	: a megadott blokk nem található
21	READ ERROR	: szinkron karakter (hardver) hiba
22	READ ERROR	: a megadott blokk nincs felírva
23	READ ERROR	: hibás ellenőrző összeg (adatnál)
24	READ ERROR	: hibás adatbyte (hardver hiba)
25	WRITE ERROR	: írás ellenőrzés hiba
26		
27	READ ERROR	: ellenőrző összeg hiba (Header-nél)
28-29		
30	SYNTAX ERROR	: hibásan megadott parancs
31	SYNTAX ERROR	: illegális parancs (nem az 1. pozícióból)
32	SYNTAX ERROR	: 58 karakternél hosszabb parancs
33	SYNTAX ERROR	: illegális fájlnev megadás
34	SYNTAX ERROR	: hibás névmegadás (pl. ":" hiányzik)
35-38		
39	SYNTAX ERROR	: illegális parancs
40-49		
50	RECORD NOT PRESENT	: nincs ilyen rekord (bővítésnél nem hiba!)
51	OVERFLOW IN RECORD	: a rekordhossznál hosszabb adatsor
52	FILE TOO LARGE	: túl magas rekordsorszám megadása
53-59		
60	WRITE FILE OPEN	: a fájl nem lett lezárva
61	FILE NOT OPEN	: a fájl (hozzáférés előtt) nem lett megnyitva
62	FILE NOT FOUND	: a fájl nem található
63	FILE EXIST	: ilyen nevű fájl már létezik
64	FILE TYPE MISMATCH	: hibás típus megadás
65	NO BLOCK	: a kijelölt blokk már foglalt ("B-A"-nál)
66	ILLEGAL TR., SECT.	: illegális sáv- és sektorszám
67	ILLEGAL SYST. TOR S	: illegális rendszer-, sáv- vagy sektorszám
68-69		
70	NO CHANNEL	: a megadott csatorna foglalt
71	DIRECTORY ERROR	: hiba a BAM-ban
72	DISK FULL	: a lemez megtelt
73		
74	DRIVE NOT READY	: a meghajtó nem üzemkész (vagy nincs)
75	FORMAT SPEED ERROR	: formátumozásnál fordulatszám instabil
76	CONTROLLER ERROR	: hardver hiba a vezérlőben

