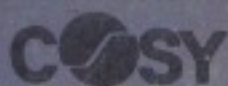
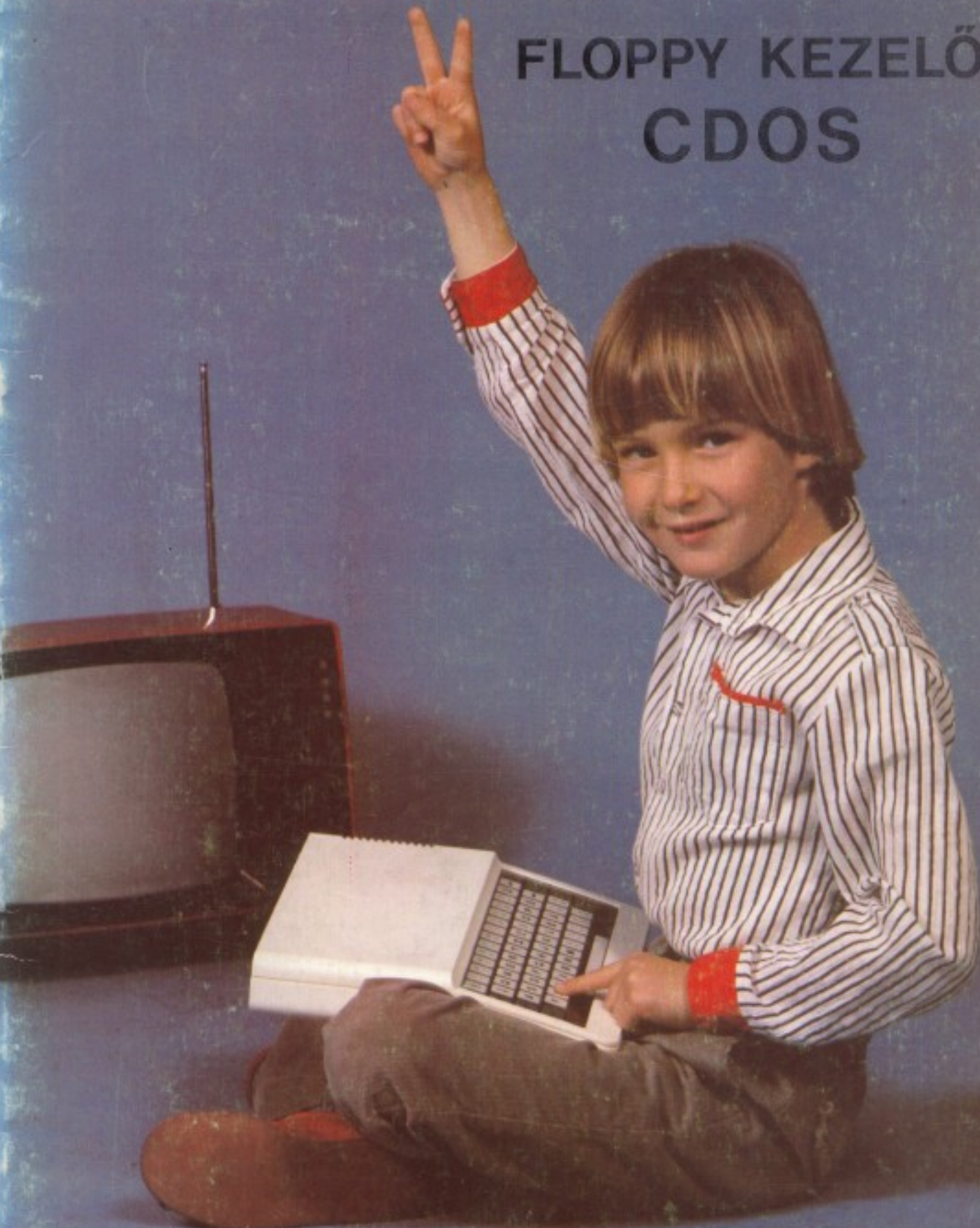


PRIMO FÜZETEK

FLOPPY KEZELŐ
CDOS



PRIMO FLOPPY KEZELŐ CDOS

**MTA-SZTAKI COSY MŰSZAKI FEJLESZTŐ LEÁNYVÁLLALAT
BUDAPEST 1985**

Szerzők

**BOROS LÁSZLÓ
MAROSI ISTVÁN
URBÁN ZOLTÁN**

Kiadja
az MTA—SZTAKI COSY Műszaki Fejlesztő Leányvállalat

A változtatás jogát fenntartjuk.

Felelős kiadó
MÓRICZ SÁNDOR
igazgató

Megjelent a GRAFO KIADÓI IRODA gondozásában
1000 példányban, 2,6 (A/5) ív terjedelemben
255—2/85

VTV REPROTECHNIKA

```

*****
*
* Commodore tip. soros perifériák kezelése *
*          PRIMO számítóséppel          *
*
*****

```

a CDOS nevű programot töltsük be a kazettáról a LOAD utasítás segítségével (lásd: PRIMO kézikönyv). Figyelem! Az esetleg benn lévő BASIC program kitörölődik a gépből!

A CDOS ellenőrzi a BASIC rendszert, nem megfelelő BASIC verzió esetén hibaüzenetet ad, egyébként egy fejléc kiírása után a megjelenő OK üzenet jelzi, hogy BASIC parancsállapotba kerülünk.

(Megjegyzés a betöltéssel kapcsolatban:

A program felköltözik a memória végére, a 40B1H címen található memóriavéscím alá. Ezután ezt az értéket módosítja, hogy a BASIC rendszer 'tudjon róla', valamint, hogy további önrelokálós gépi kódú programokat lehessen betölteni a szabad felső címtartományba.)

A program kiterjeszti a BASIC utasításkészletét, ezen új utasítások segítségével kezelhetjük a Commodore soros buszra kapcsolható perifériákat, pl. a 1541 floppy egységet, a MOM floppy egységet, az MPS-801 nyomtatót, stb.

A használható új BASIC parancsok leírásánál alkalmazott jelölések a következők:

- | | |
|----------|---|
| kif | Numerikus kifejezés. Legegyszerűbb esetben egy szám, de lehet tetszőleges bonyolult számértéket eredményező kifejezés is.
Pl.: 12, 5*I ^x -3, LEN(A\$), stb. |
| strkif | String értékű kifejezés. Legegyszerűbb esetben idézőjelek között álló szöveg, de lehet tetszőleges stringváltozók, stringműveletekből kialakított stringkifejezés is.
Pl.: "FILENEV", A\$+CHR\$(135), MID\$(A\$,2,1), stb. |
| var | Egy numerikus változó.
Pl.: A, FILE%, B\$(12,I), stb. |
| strvar | Egy stringváltozó.
Pl.: A\$, NAME\$, B\$(12,I), stb. |
| varlista | Egy darab változó (numerikus vagy string), illetőleg több változó (numerikus vagy string) vesszővel elválasztva. A numerikus változók, valamint a stringváltozók tetszőlegesen keverhetők egymással. |

- strlista** Egy darab stringváltozó, illetőleg több stringváltozó vesszővel elválasztva.
- printlista** kifejezés, illetve vesszővel vagy pontosvesszővel elválasztott kifejezések listája; lényegében az, amit a PRINT utasítás paraméteréül meg lehet adni (kivéve a \$X,Y alakú pozicionálást, valamint file-ba írás esetén a TAB(X) függvényt is.)

Az új BASIC parancsok közös jellegzetessége, hogy a CMD kulcsszóval kezdődnek, ez utal a Commodore névre. Valószínűleg a kezdetben előforduló hibák nagy részét az teszi majd ki, hogy elfeledkezünk kiírásáról. Erre tehát külön ügyeljünk, amire meg nem szokjuk. A kulcsszavak között a szóközök - hasonlóan a többi BASIC utasításhoz - nem szükségesek. A parancsok nagy része használható programutasításként, valamint közvetlen parancsként is. Ahol ez nem áll fenn, azt külön megjegyezzük!

A parancsok feloszthatók adminisztrációs, programkezelő, állománykezelő és nyomtatókezelő parancsokra.

1. Adminisztrációs parancsok

Az adminisztrációs parancsok a következők:

- CMD SET kif
Az egyes lemezkezelő utasítások a commodore-nál szokásostól eltérően nem igénylik, hogy bennük a lemez egység számát mindig kiírjuk (Vagyis nincs szükség mindenhol a ",8" -ra). Alaphelyzetben a 8-as egységet lehet használni. Ha mégsem ezt akarjuk, ezen utasítás segítségével térhetünk át pl. a 9-es egység kezelésére, majd szintén ezzel térhetünk vissza a 8-asra. Az egység szám 8..12 lehet. Egy lemezegységgel dolgozva használata nem szükséges.
- CMD SET ,kif
Alaphelyzetben a nyomtató kezelő utasítások a 4-es egység számú nyomtatót használják. Ha másik egységet akarunk használni, ezzel az utasítással térhetünk át arra. A kif értéke 4 vagy 5 lehet.
- CMD SET kif,kif
Az előző két utasítás egyesített formája. Az első kifejezés vonatkozik a lemezegységre, a második a nyomtatóra.
- CMD ERROR
A lemezegység hibacsatornáját olvassa le, majd az üzenetet kiírja a képernyőre. Az üzenet formája pl.:

21,READ ERROR,18,01

ahol az első szám a hiba kódja, a szöveg a hibaüzenet, a harmadik-negyedik paraméter pedig a lemez sávja és szektora, ahol a hiba történt. Bővebben lásd a VIC-1541 felhasználói kézikönyvében.

- CMD ERROR var, strvar, var, var

Ez az utasítás csak programból használható, közvetlen parancsként nem! Az előzővel azonos, kivéve, hogy az üzenetet nem a képernyőre írja, hanem a felsorolt változóknak adja értékül. Nem szükséges mind a négy változó: ha pl. csak kettőt írunk, (a másodiknak stringváltozónak kell lennie!) akkor az első kapja értékül a hibakódot, a második a hibaszöveget, a sáv és szektorszám pedig elvész.

- CMD RESET Alaphelyzetbe állítja a lemezesyséset (az egység parancs csatornájára való "I" parancs kiküldésével: lásd a VIC-1541 felhasználói kézikönyvét), valamint (logikailag) lezárja az esetleg nyitott csatornákat is. Ebben különbözik a

CMD PRINT "I"

parancstól (lásd később). Akkor érdemes kiadni, ha a lemezesyséssel valami probléma van. Ha erre nem javul meg, próbáljuk meg (ha van) a külső RESET gombot, ha erre sem, az egység ki- majd bekapcsolását. Ha más mindig rossznak találjuk, szóljunk szakembernek.

2. Programkezelő parancsok

Az egyes programkezelő parancsok a következők:

- CMD SAVE strkif

A memóriában található BASIC programot lemezre írja a strkif-nek megfelelő nevű PRG típusú file-ba. A file nevére vonatkozóan bővebb információkat a VIC-1541 felhasználói kézikönyvében találunk. Az ottaniakhoz képest egy fontos különbség, hogy a PRIMO-n a '@' karakternek a nagy 'E' betű felel meg, tehát helyesen a "név" nevű program módosítás utáni visszairását a lemezre a következőképpen tehetjük meg:

CMD SAVE "E:név"

Vigyázzunk arra is, hogy a kisbetűs valamint nagybetűs programnevek különböznek a lemezesység számára, tehát a "név" valamint "Név" nevű programok különböznek!

- CMD SAVE SCREEN strkif

A képernyő aktuális tartalmát írja a lemezre a strkif-nek megfelelő nevű PRG típusú file-ba. A parancsot főleg programból érdemes használni egy felrajzolt ábra elmentésére, de persze közvetlen parancsként is kiadható. A file

névére vonatkozóan lásd az előző bekezdést.

- CMD TEST strkif

A strkif-nek megfelelő nevű PRG típusú file-t hasonlítja össze a memóriában lévő programmal, illetőleg ha az adott néven a képernyőtartalmat mentettük lemezre, akkor az AKTUALIS képernyőtartalommal. Ebből következik, hogy képernyőtartalmat csak programból való kimentés és közvetlen utána végrehajtott CMD TEST esetén tudunk ellenőrizni, parancsként végrehajtva ugyanis a képernyőtartalom közben megváltozik.

Ha az ellenőrzés hibát nem talált, nem kapunk üzenetet, hiba esetén viszont azt is megtudjuk, mennyi a hibás byte-ok száma (teljesen 255-is). Bővebben erről is a hibakezelésről szóló fejezetben lesz szó.

- CMD LOAD strkif

Betölti a memóriába a strkif-nek megfelelő nevű PRG típusú file-t. A file lehet BASIC program, gépi kódú program, vagy elmentett képernyőtartalom is.

Ha a CMD LOAD utasítást programból adjuk ki és a betöltött file egy BASIC program, akkor az automatikusan elindul az első utasításánál. Gépi kódú programok akkor is elindulnak automatikusan, ha közvetlen paranccsal töltjük be őket. A PRIMO-n nem kell (és nem is lehet) gépi kódú programok betöltésénél az utasítás mögé írni a ",1"-et, a CMD LOAD e nélkül is "tudja", hogy mely programot mely címre kell betölteni.

Fontos ügyelnünk arra, hogy pl. a "program" nevű file-t nem tudjuk betölteni a CMD LOAD "PROGRAM" utasítással, mivel a diszk különbséget tesz a kis és nagybetűk között! A file névére vonatkozóan bővebb információkat a VIC-1541 felhasználói kézikönyvében találunk.

A speciális "\$" név (illetőleg "\$:" kezdetű név) esetén a PRIMO-nál is a lemez katalógusa (illetőleg annak egy részlete) töltődik a memóriába (bővebben lásd a VIC-1541 felhasználói kézikönyvet), természetesen ezt nem érdemes programként futtatni, csupán kilistázni. Ha nem a nyomtatóra akarunk listázni, érdemesebb a CMD \$ utasítást használni (lásd: CMD \$).

- CMD \$

Segítségével a memóriában lévő BASIC program elrontása NÉLKÜL nézhetjük meg a lemez teljes katalógusát. A parancs beolvassa a katalógust a BASIC változók mögötti szabad területre, majd a képernyőre listázza azt. Közben - hasonlóképpen a BASIC LIST parancs működéséhez - egy tetszőleges billentyűt kell nyomnunk, hogy fusson a lista. Ha egy részletét jobban akarjuk szemlélteni, csak el

kell engedni a billentyűt. a BREAK gomb megérintésével a katalógus listázása abbahagyható.

Ez az utasítás általában célszerűbb, mint a CMD LOAD "\$", hisz nem törli ki a memóriában lévő BASIC programot. Akkor nem felel csak meg, ha nem minden katalógus-tételt, hanem csak bizonyos mintára illeszkedőket akarunk megnézni, illetve ha a katalógust (vagy részletét) ki akarjuk nyomtatni.

- CMD KILL strkif

Törli a lemez katalógusából a strkif-nek megfelelő nevű PRG típusú file-t, illetve file-okat. A

CMD KILL "PR?BA*"

parancs például minden olyan file-t töröl a lemezből, aminek a neve "PR"-rel kezdődik, majd egy tetszőleges (de pontosan egy darab) karakter áll, aztán "BA" következik, és végződése tetszőleges (több karakter hosszúságú is lehet). A letörölt file-ok számát megtudhatjuk, ha a parancs használata után kiadjuk a CMD ERROR parancsot. Ekkor a lemezegység üzenetében (ez most nem hibára utal!) a harmadik paraméter a letörölt file-ok száma lesz. (bővebben lásd a VIC-1541 felhasználói kézikönyvében.)

MOM floppy egység használatakor törléskor a file fizikailag nem törlődik, csupán megjegyzi a katalógusban a DOS, hogy már 'nem létezik'. Ezért előfordulhat az, hogy bár csak néhány file-t látunk a katalógusban, a diszk mégis hibát jelez, hogy megtelt a katalógus. Ilyenkor segít a

CMD PRINT "V"

parancs (nagybetűs "V"). Ennek hatására fizikailag is törlődnek a file-ok. (A CMD PRINT utasításról később lesz szó.)

- CMD NEW strkif

A parancssal kétféle műveletet is végezhetünk. Egyrészt letörölhetjük a lemez teljes katalógusát pl. a

CMD NEW "ÜRES LEMEZ"

parancssal. Ekkor az összes file törlődik, a katalógus új neve pedig "ÜRES LEMEZ" lesz. Másrészt egy teljesen új lemezt formattálhatunk vele, pl.

CMD NEW "FORMATTALT LEMEZ, ID"

Formattálás nélkül a lemezt a diszk nem képes elolvasni, mert nincsenek rajta a szükséges ütemjelek, stb. Egy lemezt életében (ha a rajta lévő jelek meg nem sérülnek) általában egyszer kell csak formattálni. Az előző parancs hatására a lemez azonosítója a két karakteres "ID" lesz (ez minden egyes blokkba bekerül), a katalógus neve pedig "FORMATTALT LEMEZ".

3. Állománykezelő Parancsok

Az állománykezelő parancsok használhatóak (általában) SEQ típusú file-ok létrehozására, írásra, olvasásra; illetve a lemezeység intelligens tulajdonságainak kihasználására.

Először a file-ok (illetve csatornák) megnyitásának négyféle módját ismertetjük. A PRIMO-n egyszerre 10 csatorna lehet nyitva; ez elegendő 2 lemezeység összes csatornájának megnyitására.

- CMD CREATE kif, strkif
Létrehoz az aktuális lemezeységen egy strkif-nek megfelelő nevű, SEQ típusú file-t. (Az aktuális lemezeység az, amit a CMD SET kif utasítással utoljára beállítottunk.) A file-hoz a továbbiakban a kif sorszámú csatornán keresztül tudunk hozzáférni a CMD PRINT #kif (lásd később) utasítással. Pl. a
CMD CREATE 3, "ADATOK"
parancs létrehozta a 8-es lemezeységen (hacsak mást nem állítottunk be) az "ADATOK" nevű SEQ típusú file-t, amihez a hármas csatornán keresztül tudunk hozzáférni, míg a
CMD SET 9: CMD CREATE 4, "ADATOK"
parancs létrehozta a 9-es lemezeységen az "ADATOK" nevű SEQ típusú file-t, amihez a négyes csatornán keresztül tudunk hozzáférni. A kif értéke 2..14 lehet (lásd a VIC-1541 felhasználói kézikönyvet).

Amíg legalább 1 csatorna nyitva van, a lemezeység piros LED-je folyamatosan világít. Nagyon fontos, hogy ekkor ne vegyük ki a lemezt az egységből, mert ennek hatására elvesz az összes ezen csatornába (ezen file-ba) írt adat. A lemezt csak akkor vehetjük ki, ha már a csatornát lezártuk a CMD CLOSE kif, vagy a CMD CLEAR parancsok egyikével (lásd később).

- CMD CONT kif, strkif
Az utasítás a MOM floppy egységgel nem használható! Megnyitja az aktuális lemezeységen (lásd előző bekezdés) a strkif-nek megfelelő nevű, SEQ típusú file-t a kif sorszámú csatornán keresztüli hozzáféréssel. A kif értéke itt is 2..14 lehet. A file-hoz a CMD PRINT #kif parancsokkal férhetünk hozzá (lásd ott), a file-ba írt adatok hozzáfűződnek a file eddigi végéhez, a korábban beírt adatok mögé, a file-nak természetesen már korábban léteznie kell a lemezen.
- CMD OPEN kif, strkif
Megnyitja az aktuális lemezeységen már létező, strkif-nek megfelelő nevű SEQ típusú file-t olvasásra a kif (2..14) sorszámú csatornán keresztül. A file-hoz a CMD INPUT #kif vagy a CMD GET #kif utasításokkal

férhetünk hozzá (lásd később).

- CMD DEF kif, strkif

Ez az utasítás felel meg a Commodore BASIC
 OPEN kif, kif, kif, strkif (pl. OPEN 2,8,2,"NAME,S,R")
 utasításának. Az előző három megnyitási mód ezen megnyi-
 tási mód speciális eseteinek számít, ugyanis:
 CMD OPEN 11,"számok" ≡ CMD DEF 11,"számok,SEQ,READ"
 CMD CREATE 8,"betűk" ≡ CMD DEF 8,"betűk,SEQ,WRITE"
 CMD CONT 6,"kódok" ≡ CMD DEF 6,"kódok,SEQ,APPEND"

Ezzel az utasítással lehetőség nyílik arra, hogy ne csak szekvenciális (SEQ típusú) file-okat nyithassunk meg, hanem tetszőleges egyéb (pl. PRG típusú) file-t, valamint használhassuk a VIC-1541 adta további megnyitási lehetőségeket (lásd a felhasználói kézikönyvet).

Az utasítás szintén az aktuális lemezegység kif számú csatornájának megnyitására használható. A csatornához a CMD INPUT #kif vagy a CMD GET #kif, illetőleg a CMD PRINT #kif utasításokkal férhetünk hozzá (attól füg-
 gően, hogy olvasásra, vagy írásra nyitottuk meg a csator-
 nát).

Miután egy csatornát (file-t) megnyitottunk, használni kívánjuk azt: írásra nyitáskor (CMD CREATE illetve CMD CONT) írni akarunk bele, olvasásra nyitáskor pedig (CMD OPEN) olvasni akarunk belő-
 le. Ezt a következő utasításokkal tehetjük meg:

- CMD PRINT #kif, printlista

Az utasítással a kif csatornába (vagyis a megfelelő
 CMD CREATE kif, strkif által a lemezen létrehozott strkif
 nevű file-ba) tudunk adatokat írni. (A PRINT-et rövidít-
 hetjük kérdőjellel is.) Ha pl. a
 CMD PRINT #12,"két szám",45,20,"kódok",CHR\$(45);CHR\$(20)
 utasítást használjuk, akkor az eddig kiírtak mögé (innen
 ered a szekvenciális=soros elnevezés) a lemezeze-
 ren a "két szám" szöveg (8 byte), egy vessző (1 byte), szövege-
 sen a 45 (" 45 " 4 byte), újabb vessző (1 byte),
 szövegesen a 20 (" 20_" 4 byte), újabb vessző (1 byte), a
 "kódok" szöveg (5 byte), újabb vessző (1 byte), majd a
 két CHR\$ közötti pontosvessző miatt közvetlenül egymás
 után a 45 és a 20 (ASCII) kód (2-szer 1 byte), végül egy
 sorvéjjel (1 byte, kódja 13).

Ha az utasítás végére egy pontosvesszőt is írtunk
 volna, a sorvéjjel nem került volna be a file-ba. Ha
 pedig az utasítás végére vesszőt teszünk, a file-ba csak
 a vessző fog bekerülni sorvéjjel NÉLKÜL. Fontos különbség
 a képernyőre írással szemben, hogy míg ott a vessző
 hatására annyi szóköz íródik ki, hogy a kiírt számok
 szépen oszlopokban legyenek, file-ba íráskor a vessző
 hatására vessző íródik a ki a csatornára. (Figyelem! A

Commodore BASIC esetében nem így van, ott file-ba írskor is szöközők írónak ki, így több hely vész kárba.)

Az utasítás speciális esete pl. a
CMD PRINT #12
 utasítás, printlista nélkül. Ennek hatására egyetlen sorvéjjel írónik a file-ba.

A numerikus értékek (számok, változók tartalma, kifejezések értéke) szöveggé alakítva írónak be a file-ba olyan formában, amit az STR\$(kif) függvény állít elő: vagyis pozitív számok előtt és mögött is egy-egy szököző lesz, negatív számok mínuszjellel kezdődnek, mögöttük pedig szintén szököző áll.

A CHR\$(kif) függvénnyel lehet a lestimörebb számkii-rást megvalósítani: a kif értékű számot (0-tól 255-ig terjedhet) 1 byte-ban (1 karakternyi helyen) írja bele a file-ba.

A kiirt formátum olyan, hogy a CMD INPUT #kif utasítással minden nehézség nélkül vissza lehessen olvasni.

Több lemezegység használatakor fontos az, hogy a csatornaszámba beletartozik az az egységyszám is, ami a megnyitáskor aktív volt (tehát ahol a file keletkezett).
 Pl. a

CMD SET 9: CMD CREATE 2,"A": CMD SET 8: CMD CONT 3,"B"
CMD PRINT #2,"Ez lesz az A nevű file elején"
CMD PRINT #3,"Ez lesz a B nevű file folytatása"
 utasítássorozat hatására a 2-es csatorna a 9-es egység "A" nevű file-ját jelenti, a 3-as csatorna pedig a 8-as lemezegység "B" nevű file-ját; az egyes PRINT utasítások pedig a megfelelő file-okba írnak.

- **CMD INPUT #kif,varlista**

Az utasítás csak programfutás közben használható, közvetlen parancsként nem!

A kif sorszámú csatornából (amit pl. **CMD OPEN kif,"név" utasítással** nyitottunk meg) beolvas egy teljes sort (vagyis sorvéjjelis), s az abban található adatokat az egyes változóknak adja értékül. Fontos az, hogy ahová stringet írunk ki, ott stringet is olvassunk vissza! kiirt számot visszaolvashatunk számként (ekkor numerikus változót kell a varlista megfelelő pozíciójába tenni), vagy szövekként is (egy stringváltozóba) teljesen hasonlóan ahhoz, ahogy az INPUT utasítás működik a billentyűzetről.

Ha kevesebb változót tartalmaz a varlista, mint ahány adat a file beolvasott sorában található volt, csak a beolvasott adatok eleje adódik értékül a megfelelő változóknak, a maradék elvész! Ezért nagyon fontos az, hogy a

beolvasás ugyanolyan struktúrában történjen, mint ahogy a kiírás volt. Példának nézzük a következő programot:

```

10 REM ide kell tenni azt a programrészt,
20 REM ami bekérdezi a neveket és az
30 REM életkorokat.
40 REM ...

1000 REM N tartalmazza az emberek számát
1010 REM NAME$(I) az I-edik ember nevét
1020 REM AGE(I) az I-edik ember életkorát.
1030 REM
1040 CMD CREATE 5,"ADATOK"
1050 CMD PRINT #5,N,"ember neve és életkora:"
1060 FOR I=1 TO N
1070 CMD PRINT #5,NAME$(I),AGE(I)
1080 NEXT I
1090 CMD CLOSE 5
1100 END
(Az 1090-es sor magyarázatát lásd később, a CMD CLOSE
utasításnál).
```

Az a program pedig, ami az előző által kiírt adatokat beolvassa és kiírja a képernyőre, a következő lehet:

```

10 CMD OPEN 7,"ADATOK"
20 CMD INPUT #7,N,S$
30 PRINT N;S$
40 PRINT
50 FOR I=1 TO N
60 CMD INPUT #7,NAM$,A
70 PRINT "Név: ";NAM$,"Kor: ";A;"év"
80 NEXT I
90 CMD CLOSE 5
100 END
```

A harmadik program pedig egy év elteltével minden ember életkorát megnöveli, hogy az adatok újra jók legyenek:

```

10 CMD OPEN 2,"ADATOK"
20 CMD CREATE 12,"új adatok"
30 CMD INPUT #2,N,S$
40 CMD PRINT #12,N,S$
50 FOR I=1 TO N
60 CMD INPUT #2,NAM$,A
70 CMD PRINT #12,NAM$,A+1
80 NEXT I
90 CMD CLOSE 2: CMD CLOSE 12
100 CMD KILL "ADATOK"
110 END
```

Érdeemes megfigyelni, hogy az olvasó programok ugyanannyi adatot olvastak a file-ből, mint ahányat a lezelső

beleirt. Ha több adatot próbálnánk kiolvasni, hibaüzenetet kapnánk! (FD error)

- CMD GET #kif, strlista
Ez a fajta beolvasó utasítás lehetővé teszi, hogy ismeretlen struktúrájú file-ból is olvasni tudjunk, illetve, hogy a tömörített adatábrázolást is feldolgozhassuk. CMD GET #kif esetén a printlistán szereplő vesszők, illetve a sorvégsjel elvesztik elhatároló szerepüket, és beolvasásuk semmilyen mellékhatást nem fejt ki. Pl. a következő programrészlet lefuttatása után:

```
10 DIM A$(13)
20 CMD CREATE 2,"ADATOK"
30 CMD PRINT #2,12,8;CHR$(128)
40 CMD PRINT #2,-3 : CMD CLOSE 2
50 CMD OPEN 3,"ADATOK"
60 FOR I=0 TO 13 : CMD GET #3,A$(I) : NEXT I
70 CMD CLOSE 3
```

az A\$ tömb egyes elemei a következők lesznek:
" ", "1", "2", " ", ",", " ", "8", " ", CHR\$(128),
CHR\$(13), "-", "3", " " és CHR\$(13)

A strlistán több stringváltozó is szerepelhet, ekkor több byte (karakter) beolvasása történik: minden stringváltozó egykarakteres (egy hosszú) stringet kap értékül.

Amennyiben a file végén túl próbálunk olvasni (tehát pl. az előbbi példában a 60-as sorban 13 helyett 20 szerepelne), 'FD error' hibaüzenetet kapnánk. Mivel ismeretlen szerkezetű file feldolgozásakor nincs információnk annak pontos hosszáról, szükség van egy olyan eszközre, amellyel eldönthetjük, hogy végre értünk-e már a file-nak. Ez a segédeszköz az EOF(kif)

függvény. kif a lekérdezendő csatorna számát szolgáltatja, a függvény értékéül a logikai igaz (-1) vagy hamis (0) értéket kapjuk. Jelentése: "a kif sorszámú csatornából már beolvastuk a file utolsó karakterét is". Ennek segítségével az előbbi 60-as sor a következő "elegánsabb" alakba írható:

```
60 IF NOT EOF(3) THEN CMD GET #3,A$(I) : GOTO 60
```

Miután a file-ba minden szükségeset beleirtunk, illetve kiolvas-tunk onnan, le kell azt zárunk. Erre használhatók a következő utasítások:

- CMD CLOSE kif
Lezárja a kif-ben megadott számú csatornát. Ennek eredményeként a csatornához tartozó file lezárt lesz, ez a csatorna pedig újra felhasználható egy állománymegnyitási

parancsban. Ha egy létrehozandó (pl.:
 CMD CREATE kif,stri kif parancssal megnyitott) állományt
 elfelejtünk lezárni, a benne lévő adatok elvesznek, nem
 lesznek elérhetők!

Ezzel az utasítással a lehetséges 4 megnyitási mód
 bármelyikével (CREATE, CONT, OPEN, DEF) megnyitott csa-
 tornát le tudunk zárni.

- CMD CLEAR

Az aktuális lemezesységen lezárja az összes megnyitott
 file-t, mintha valamennyire kiadtuk volna a megfelelő
 CMD CLOSE utasítást. A parancs hibátlan végrehajtása
 esetén az aktuális lemezesység piros lámpájának el kell
 aludnia. Figyelem! Ha több lemezesyséssel akarjuk ezt a
 parancsot használni, mindegyik esységre ki kell ezt
 adnunk, mivel CSAK az AKTUALIS lemezesység csatornát
 zárja le! (A CMD CLOSE_kif utasítás ezzel ellentétben nem
 az aktuális lemezesység csatornáját zárja le, hanem amely
 esységen a kif csatornát megnyitottuk.)

A fenti állomány- (file-) kezelő utasítások használatát részlete-
 sebben ld.: a VIC-1541 felhasználói kézikönyvében.

Bár a legfontosabb diszkrparancsoknak létezik megfelelőjük a
 CDOS-ban, a fennmaradók használatához szükség van a parancs- ill.
 hibacsatorna kezelésére. Ezt a következő alakban tehetjük meg:

- CMD PRINT printlista

Ez az utasítás a lemezesység parancs-csatornájára küldi a
 printlistának megfelelő értékeket. A megfelelő
 Commodore-alak:

OPEN 15,8,15,printlista : CLOSE 15

A kiadható parancsokat és formájukat ld.: a VIC-1541
 felhasználói kézikönyvében.

- CMD GET strlista

Ezt az utasítást kizárólag egy esetben érdemes használni:
 amikor a lemezesység memóriáját az "M-R:" parancssal
 olvassuk: Az strlista stringváltozói értékül kapják a
 beállított memóriacímről kezdődő byte-okat. Bővebben lásd
 a VIC-1541 felhasználói kézikönyvében.

4. Nyomtatókezelő parancsok

A CDOS programrendszer lehetővé teszi a Commodore típusú
 nyomtatók egyszerű használatát is. A megfelelő parancsok:

- CMD LPRINT printlista

Hatása teljesen megegyezik a PRINT utasításéval, de az
 írás az aktuális nyomtatóra történik (ld. CMD SET ,kif

utasítás). Mivel a nyomtatók a kiírt szöveget egy pufferba gyűjtik, ezért a nyomtatás eredménye csak akkor látható, amikor egy sor teljes egészében elkészült. Pl. a következő programrészletben:

```
10 PRINT "Most kezdek el írni"
20 FOR I=32 TO 90 : CMD LPRINT CHR$(I); : NEXT I
30 PRINT "Most fogja csak kinyomtatni!"
40 CMD LPRINT
```

a tényleges nyomtatás csak a 40-es sor végrehajtásakor fog bekövetkezni, mert a cikluson belül a soremelés a CHR\$(I) utáni ";" karakter segítségével le volt tiltva.

Mivel a Commodore és a PRIMO gépek karakterkészlete eltérő, a CDOS program minden karakter kiküldése előtt egy konvertálást végez. Sajnos ékezetes betűket így sem képes nyomtatni, helyükbe a megfelelő, ékezet nélküli karakter kerül. Amennyiben a fent leírt karakterkonvertálást le akarjuk tiltani, a printlista valamelyik eleme CHR\$(0) legyen; a CDOS ezt a kódot nem küldi el a nyomtatónak, de megjegyzi, hogy ezután nem kell konvertálást végeznie. Pl.:

```
CMD LPRINT CHR$(0);"é"
```

hatására egy "é" karakter íródik ki a nyomtatóra. A konvertálás újra bekapcsolható szintén CHR\$(0) kiküldésével. Szintén visszakapcsolódik a konvertálás a következő CMD LPRINT utasítás hatására is, mivel a tiltás hatásköre egyetlen printlista terjed csak ki. Ha tehát mégis konvertálás nélkül akarunk dolgozni (pl. grafikus üzemmódban), minden egyes CMD LPRINT utasítást CHR\$(0)-val kell kezdeni.
 Figyelem: Az LPRINT kulcsszó nem írható L?-nek!

- CMD LLIST sorszámtartomány
 Megeszezik a LIST paranccsal, de a programot (vagy egy lemez katalógusát, ld. CMD LOAD "\$") a képernyő helyett a nyomtatóra listázza. A sorszámtartomány értelmezése is azonos, és ugyanúgy el is hagyható.

A nyomtató kezelésére vonatkozóan részletesebb információval szolgált a konkrét készülék felhasználói kézikönyve.

5. Hibakezelés

A CDOS használata közben a hibakezelés szempontjából figyelembe kell venni, hogy egyrészt új hibaüzenetek keletkeztek (pl. NO Error), másrészt a BASIC hibakezelési rendszerébe a lemezesség által generált hibaüzeneteket is be kell illeszteni. Általánosságban elmondható, hogy a hibakezelés lényegében azonos maradt

a megszokottal, és annak felhasználói átvételét az ON ERROR GOTO utasítással a rendszer messzemenően támogatja.

a diszkkezelő parancsok többségének jellegzetessége, hogy diszk hiba esetén automatikusan végrehajtják a CMD ERROR parancsot, vagyis a diszk hibaüzenete a képernyőre kerül (kivéve, ha ON ERROR GOTO van érvényben, ekkor programból is lekezelhetjük a hibát.) A CMD KILL parancs esetében csak akkor kerül üzenet a képernyőre, ha ténylegesen hiba történt: a lemezegység által küldött 1-es üzenet (a törölt file-ok száma) nem számít hibának, így nem íródik ki. Ha érdekel bennünket, hajtsuk végre a CMD ERROR parancsot. A diszkkezelő parancsok közül csupán a CMD PRINT+kif, a CMD PRINT, a CMD INPUT+kif, a CMD GET+kif valamint a CMD GET nem kérdezi le a hibacsatornát azért, hogy ne lassítsák a programfutást. Ha lemezhiba fordul elő ezen utasítások közben, akkor a diszk nem válaszol a PRIMO írási vagy olvasási kérésére, ezért a CDOS ezt úgy értékeli, hogy nincs jelen az eszköz (DEVICE NOT PRESENT hiba). Természetesen ez is lekezelhető programozottan, ON ERROR GOTO használatával. (Megjegyzés: A Commodore BASIC esetében ezen szolgáltatások többsége nem volt megtalálható, ezért hiba esetén bonyolult műveletekkel kellett lekérdezni annak fajtáját, ha csak nem használtunk valamilyen BASIC bővítést, pl. a DOS SUPPORT-ot.)

Az új hibakódok, belső megjelenítési formájuk (értékük) és jelentésük a következők:

- 12 OM Error
A hiba kis szabad memóriahely miatt fordulhat elő esetünkben a CMD LOAD utasításkor, vagy CMD\$ esetében.
- 42 FD Error
Ha a hiba file-ból való olvasáskor jelentkezik, akkor azt jelenti, hogy túl akartunk olvasni a file végén. Ezt elkerülhetjük az EOF(kif) használatával.
A hiba a CMD TEST használatánál azt jelenti, hogy a megnevezett program nem tesztelhető. Oka valószínűleg hibásan megadott file-név.
CMD TEST használatakor szintén ez a hibakód jelzi azt, hogy a hasonlítás különbségeket talált. Ekkor a kiírás formája pl.
17 FD Error
ahol a 17 jelenti a hibás byte-ok (karakterek) számát. Ha ez a szám 255, az legalább 255 hibát jelent (lehet több is).
- 90 BO Error
A hibajelzés akkor keletkezik, ha egyszerre 10-nél több csatornát akarunk megnyitni.
- 92 FD Error
Már nyitva van a csatorna. A hiba oka az, hogy

elfeledkeztünk a csatorna lezárásáról, és így próbálkozunk újbóli megnyitásával.

94 NO Error
Nincs nyitva a csatorna. Akkor kapjuk az üzenetet, ha olyan csatornából akarunk olvasni vagy írni, amit még nem nyitottunk meg. Okai valószínűleg az, hogy elgérfeltük a számot.

96 BD Error
Adatirányhiba. Abban az esetben fordul elő, ha olvasásra megnyitott file-ba írni akarunk, vagy írásra megnyitottból akarunk olvasni. Valószínűleg CMD OPEN utasítással próbálkoztunk file-t létrehozni CMD CREATE helyett.

100 DEVICE NOT PRESENT
A használni kívánt eszköz (lemezesység vagy nyomtató) a CDOS rendszer hívására nem felel. Próbáljuk kiadni ismét az utasítást, miután meggyőződünk róla, hogy a készülék üzemképes és a megfelelő egységszámot használjuk (CMD SET parancs). Súlyosabb hiba után előfordulhat, hogy a lemezesység "meghal", és nem válaszol; ilyenkor adjuk ki a CMD RESET utasítást.

100+derr derr,HIBASZÖVEG,sáv,szektor
A diszk által szolgáltatott hibaüzenetek változtatás nélkül a képernyőre kerülnek, itt 'derr' jelentette a hiba számát. ON ERROR GOTO használatkor hibaszámként ennek a számnak 100-zal növelt értékét (100+derr) kapjuk, tehát pl. 21-es hiba (olvasási hiba) esetén 121-et. Az egyes számértékek jelentését a VIC-1541 felhasználói kézikönyvében találhatjuk meg. A 100-zal megnövelt hibaszámok nem párosával követik egymást, mint a BASIC eredeti hibaszámai!

199 99,END OF FILE,00,00
A hibaüzenet a LOAD parancs használatakor fordulhat elő, ha nem a PRIMO-n fölirt file-t akarunk betölteni. Elkérdezhető még a hiba okaként az is, hogy a program a lemezen megsérült, vagy valami miatt hibásan lett fölírva.

JEGYZETEK

FÜGGELÉK A CDOS LEÍRÁSHOZ

- AUTOMATIKUSAN INDULÓ BASIC PROGRAMOK LEMEZRE VITELE (MENTÉSE)
A KÖVETKEZŐ FORMÁBAN TÖRTÉNIK:
CMD SAVE AUTO strinskif
ENNEK HATÁSÁRA A strinskif-NEK MEGFELELŐ NÉVEN ELMENTETT PROGRAM
BETÖLTÉSEKOR (CMD LOAD) A PROGRAM AZ ELSŐ UTASÍTÁSÁVAL AUTOMA-
TIKUSAN ELINDUL.
- A LEMEZKATALÓGUS LISTÁZÁSOKOR (CMD \$) LEHETŐSÉG VAN ARRA, HOGY
A TELJES KATALÓGUS HELYETT ANNAK CSAK EGY, MINKET ÉRDEKLŐ RÉSZ-
LETÉT ÍRASSUK KI. EZT A CMD \$ UTÁN ÍRT STRINGKIFEJEZÉSEL ADHATJUK
MEG, FORMÁJA PEDIG MEGEGYEZIK AZ EGYÉB, PL. CMD KILL UTASÍTÁSNÁL
TÖRTÉNŐ NÉVMEGADÁSSAL. A
CMD \$ "?A?"
UTASÍTÁSSAL TEHÁT A KATALÓGUS AZON TÉTELEIT ÍRATHATJUK KI, AMELYEK
OLYAN HÁROMBETŰS FILE-NEVEKET TARTALMAZNAK, MELYEKNEK KÖZÉPSŐ
BETŰJE "A".
- RELATÍV FILE-OK KEZELÉSÉNÉL A KÖVETKEZŐKET KELL FIGYELEMBE VENNI:
AMIKOR A RELATÍV FILE-T LÉTREHOZZUK, A
CMD DEF filno, strinskif+"",L,""+CHR\$(kif)
UTASÍTÁST KELL KIADNI, AHOL strinskif A LÉTREHOZANDÓ FILE NEVET ADJA
(PL. "FILE 1"), filno A FILE SORSZÁMA, kif PEDIG A REKORDHOSSZ (PL. 23),
AMELYNEK 254-NÉL KISEBBNEK KELL LENNI.

LÉTEZŐ RELATÍV FILE MEGNYITÁSA:
CMD DEF filno, strinskif+"",L"
TEHÁT A "",L" RÉSZT ITT IS KI KELL ÍRNI!

RELATÍV FILE-BAN POZÍCIONÁLÁSHOZ A KÖVETKEZŐ UTASÍTÁST HASZNÁLJUK:
CMD PRINT "P" CHR\$(filno)CHR\$(lo)CHR\$(hi)
AHOL 16 * hi + lo A KÍVÁNT REKORD SORSZÁMA,

ILLETVE:
CMD PRINT "P" CHR\$(filno) CHR\$(lo) CHR\$(hi) CHR\$(kif),
HA A REKORDON BELÜL A kif-EDIK KARATERRE AKARUNK POZÍCIONÁLNI.

RELATÍV FILE LEZÁRÁSA, AZ ABBA TÖRTÉNŐ ÍRÁS, VALAMINT AZ ABBÓL VALÓ
OLVASÁS UGYANÚGY TÖRTÉNIK, MINT SOROS FILE-OK ESETÉN.
AZ EGYETLEN KÜLÖNBSÉGET AZ EDF(filno) FÜGGVÉNY OKOZZA: RELATÍV
FILE ESETÉBEN A FÜGGVÉNY A REKORD VÉGÉT JELENTI. ITT UGYANIS A FILE
VÉG JELZÉSRE NINCS KIMONDOTTAN SZÜKSÉG, A REKORDVÉG INFORMÁCIÓJA
FONTOSABB.



**Forgalmazó:
ELEKTROMODUL**

**1132 Budapest, Victor Hugo u. 13-15.
Telefon: 495-340 ● Telex: 22-5154**

**Menedzser:
MTA-SZTAKI COSY MŰSZAKI FEJLESZTŐ LEÁNYVÁLLALAT
BUDAPEST 5.**

**Pf.: 690
1365**