

**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 számítógépről          *
*
*****

```

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

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 kapcsolatosan:

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 buszára 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. Leggyakrabban egy szám, de lehet tetszőleges bonyolult számértéket eredményező kifejezés is.
Pl.: 12, 5*I%-3, LEN(A\$), stb. |
| strkif | String értékű kifejezés. Leggyakrabban idézőjelek között álló szöveg, de lehet tetszőleges stringváltozókból, 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 jellemzősége, 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, amig meg nem szokjuk. A kulcsszavak között a szögzők - 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 egységszámát 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 lemezesységet (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 lemezesységgel 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 még 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 '0' 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 különbözőek a lemezesység számára, tehát a "név" valamint "Név" nevű programok különbözőek!

- 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

as

névére vonatkozóan lásd az előző bekezdést.

- CMD TEST strkif

13

A strkif-nek megfelelő nevű PRG típusú file-t hasonlítja össze a memóriában levő 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, érdekesebb 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 katalógustételt, 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 lemezről, 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.)

MDM 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 mestelt 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 paranccsal kétféle műveletet is végezhetünk. Egyrészt letörölhetjük a lemez teljes katalógusát pl. a

CMD NEW "ÜRES LEMEZ"

paranccsal. 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, olvasására; illetőleg 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 elvész 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 parancs-csal 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,"betők" ≡ CMD DEF 8,"betők,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, illetve 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 lemezre íródik 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
 sorvégjel (1 byte, kódja 13).

Ha az utasítás végére egy pontosvesszőt is írtunk
 volna, a sorvégjel nem került volna be a file-ba. Ha
 pedig az utasítás végére vesszőt teszünk, a file-ba csak
 a vessző fog bekerülni sorvégjel 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 íráskor is szöközők írónak ki, így több hely vesz kárba.)

Az utasítás speciális esete pl. a
 CMD PRINT #12
 utasítás, printlista nélkül. Ennek hatására egyetlen sorvégsjel íródik 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 minuszjellel kezdődnek, mögöttük pedig szintén szököző áll.

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

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

Több lemezegeység használatakor fontos az, hogy a csatornaszámba beletartozik az az egységszám 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 lemezegeység "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) beolvassunk egy teljes sort (vagyis sorvégsjelig), s az abban található adatokat az egyes változóknak adja értékül. Fontos az, hogy ahová stringet írunk ki, ott stringet is olvassunk vissza! kiírt 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 strukturában történjen, mint ahogy a kiírás volt. Példának nézzük a következő programot:

```
10 REM ide kell tenni azt a program részt,
20 REM ami bekérdezi a neveket és a:
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
```

Érdemes megfigyelni, hogy az olvasó programok ugyanannyi adatot olvastak a file-ből, mint ahányat a legelső

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

- CMD GET #kif, strlista

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 értékekül.

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 vége-e már a file-nak. Ez a segédeszköz az EOF(kif)

függvény. kif a lekérdezendő csatorna számát szolgáltatja, a függvény értékekül 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 beleírtunk, illetve kiolvastunk 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, strkif paranccsal 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 csat-
 ornát le tudunk zárni.

- CMD CLEAR

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

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

Bár a legfontosabb diszkparancsoknak 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 lemezegegyesség 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 lemezegegyesség memóriáját az "M-R:" paranccsal
 olvassuk. Az strlista stringváltozói értékül kapják a
 beállított memóriacíműtől 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 pufferben 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 soromelés 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);"E"
```

hatására egy "E" 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 végig 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 LIST sorszámtartomány

Megszerezzük 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 lemezegység 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 csak 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 következő:

- 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 FO 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. Okja valószínűleg az, hogy elfeledtük 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 (lemezegység 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). Sülyosabb hiba után előfordulhat, hogy a lemezegység "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 fölírt file-t akarunk betölteni. Elképzelhető még a hiba okaként az is, hogy a program a lemezen megsérült, vagy valami miatt hibásan lett fölírva.

FÜGGELÉK A CDOS LEÍRÁSHOZ

1. AUTOMATIKUSAN INDULÓ BASIC PROGRAMOK LEMEZRE VITELE (MENTÉSE) A KÖVETKEZŐ FORMÁBAN TÖRTÉNIK:
CMD SAVE AUTO strinskif
ENNEK HATÁSÁVAL A strinskif-NEK MEGFELELŐ NÉVEN ELMENTETT PROGRAM BETÖLTÉSEKOR (CMD LOAD) A PROGRAM AZ ELSŐ UTASÍTÁSÁVAL AUTOMATIKUSAN ELINDUL.
2. A LEMEZKATALÓGUS LISTÁZÁSOKOR (CMD \$) LEHETŐSÉG VAN ARRA, HOGY A TELJES KATALÓGUS HELYETT ANNAK CSAK EGY, MINŐT ÉRDEKLŐ RÉSZLETÉT ÍRASSUK KI. EZT A CMD \$ UTÁN ÍRT STRINGKIFEJEZÉSEL ADHATJUK MEG, FORMÁJA PEDIG MEGEGYEZIK AZ EGYÉB, PL. CMD KILL UTASÍTÁSSAL TÖRTÉNŐ NÉVMEGADÁSSAL. A
CMD \$ "?A?"
UTASÍTÁSSAL TEHÁT A KATALÓGUS AZON TÉTELEIT ÍRATHATJUK KI, AMELYEK OLYAN HÁROMBETŰS FILE-NEVEKET TARTALMAZNAK, MELYEKNEK KÖZELSŐ BETŰJE "A".
3. 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.

A. melléklet

PRIMO B-XX és az FDU 1109-es meghajtó parancsszavai (részletezésük a „CDOS” programkazetta mellékleteként forgalmazott leírásban.)

Adminisztrációs parancsok

- CMDSET
az aktuális lemezegység vagy nyomtató azonosítójának beállítására
- CMDERROR
a lemezegység hibaüzenetének kiolvasására
- CMDRESET
a lemezegység alaphelyzetbe állítására

Programkezelő parancsok

- CMDSAVE
a memóriában (PRIMO) lévő program mentésére
- CMDSAVESCREEN
a képernyő aktuális tartalmának mentése
- CMDTEST
a megjelölt file összehasonlítása a memóriatartalommal
- CMDLOAD
a megjelölt file betöltésére
- CMD\$
a lemezkatalógus olvasása a memóriában lévő program rontása nélkül
- CMDKILL
file-ok törlése
- CMDNEW
új lemez formálására, régi törlésére, újraformázására

Állománykezelő parancsok

- CMDCREATE
szekvenciális file létrehozására
- CMDOPEN
szekvenciális file megnyitására (olvasásra)
- CMDDEF
tetszőleges file megnyitására
- CMDPRINT#
adatok file-ba írására
- CMDINPUT#
adatok beolvasására
- CMDGET#
adatok beolvasása karakterenként
- CMDCLOSE
a megnyitott csatorna lezárására
- CMDCLEAR
az összes megnyitott file lezárására
- CMDPRINT
a VIC 1541-es Commodore floppy egység parancsait küldi a parancs-csatornára
- CMDGET
a lemezegység memóriájának olvasására

Nyomtatókezelő parancsnok

- CMDLPRINT
írás a nyomtatóra
- CMDLLIST
a memóriában lévő program nyomtatóra listázása

B. melléklet

Commodore VC 64 és az FDU 1109-es meghajtó kapcsolata (a Commodore 1541-es floppy egység leírása alapján kezelhető a PRIMO floppy meghajtó is, a következő eltéréseket figyelembe véve):

Az FDU 1109-es a SCRATCH paranccsal törölt file-okat nem semmisíti meg, csak a lemezkatalógusban jelöli meg őket. A LOAD „\$”, 8 parancs kiadásakor beolvassa ezen file-ok azonosítóit is. Ha ilyen törölt file-t akarunk beolvasni, ez a LOAD „file-név, ”, 8 paranccsal lehetséges. A törölt file-ok által foglalt lemezterületeket a PRINT # 15, „V” utasítással tudjuk felszabotítani.

SAVE felülírással: az FDU 1109 esetében a parancsból a „@” utáni Ø-át el kell hagyni, tehát a formátum:

SAVE,@: file-név”, periféria szám

Szekvenciális file-ok: a „file-típus” azonosító „O”-tól „Z”-ig bármi lehet. A file megnyitásakor a Ø-át el kell hagyni, a megnyitás formátuma:

OPEN file-szám, periféria-szám, csatorna szám, „file-név, file-típus, irány”.

A 1541-es kézikönyvében említett „DOS segédprogram” az FDU 1109-re nem érvényes, mivel ez a Commodore-floppy demonstrációs lemezen található.

A GET# parancsnál közölt katalógus-olvasó mintaprogram az FDU 1109-el nem működik.

Véletlen elérésű (Random) file-ok: a PRIMO-floppy blokk-kiosztása nem egyezik meg a 1541-esével, minden sáv egységesen 10 blokkot tartalmaz.

Relatív file-ok: ezt a file-típust az FDU 1109-es nem tudja.

A lemezegység azonosítójának megváltoztatása: formátum a PRIMO-floppy esetében:

PRINT # file-szám, „M-W:” CHR\$ (18) (CHR\$ (56) CHR\$ (2) CHR\$ (cím+32)
CHR\$ (cím+64)

Példa a periféria-szám 9-re váltására:

10 OPEN 15,8,15

20 PRINT # 15, „M-W” CHR\$ (18) CHR\$ (56) CHR\$ (2) CHR\$ (9+32) CHR\$ (9+64)

Az FDU 1109-es hibaüzeneteit ld. a C. mellékletben, különböznek a VC 1541-esétől.

A 1541-es assemblenből való működtetéséről a kézikönyvben leírtak a PRIMO floppy-ra nem alkalmazhatók.

Diszk-parancsok: – az FDU 1109-nél nem használhatók: DUPLICATE; POSITION; USER Commands

– csak az FDU 1109-nél meglévő: UNERASE, vagy rövidítve: UE vagy UN; formátuma.

„UE: file név” a törölt file-t újra visszateszi a directory-ba.

C. melléklet

Az FDU 1109 hibaüzenetei

Kód	Üzenet	Jelentés
0	NO ERROR	nincs hiba
1	READ ERROR	olvasási hiba, íráshiba javításakor
2	READ ERROR	olvasási hiba a lemezen
3	WRITE ERROR	katalógus felírási hiba
4	WRITE ERROR	katalógus íráshiba utolsó elemnél
5	WRITE ERROR	íráshiba utolsó elemnél (CLOSE FILE)
6	DISK FULL	megtelt a lemez
7	CANNOT READ HEADER	nem olvasható a katalógus
8	FORMATTING ERROR	sikertelen formázási parancs
9	DISK NAME/ID MISMATCH	nem egyezik a diszk név a szektoron
11	WILDCHAR(S) IN FILE NAME	? vagy * az új file-névben (RENAME)
12	WILDCHAR(S) IN FILE NAME	? vagy * az új file-típus (RENAME)
13	FILE ALREADY EXIST	létező file-név (RENAME)
14	FILE NOT FOUND	nem létező file (RENAME)
15	FILE ALREADY OPEN	a file nyitva van írásra (RENAME)
21	CHANNEL ALREADY OPEN	foglalt logikai csatorna (OPEN READ)
22	WILDCHAR(S) IN FILE NAME	? vagy * file-névben
23	NO MORE BUFFER	file nyitáskor 7-nél több buffer foglalva
24	FILE NOT FOUND	nem létező file-ra való hivatkozás
25	NO MORE BUFFER	random file nyitáskor nincs több buffer
26	ALREADY ASSIGNED BUFFER	foglalt buffer (random file-nál)
27	NO MORE BUFFER	foglalt az összes buffer a katalógus kiíratáskor
28	FILE ALREADY OPEN	a file már meg van nyitva
29	NO MORE BUFFER	foglalt a 7. buffer a katalógus kiíratáskor
31	CHANNEL NOT OPENED	hozzáfordulási kísérlet nem nyitott csatornára
32	READ ERROR	olvasási hiba adatolvasáskor
33	READ ERROR	olvasási hiba katalógus kiíratáskor
41	CHANNEL ALREADY OPEN	foglalt logikai csatorna (OPEN WRITE)
43	FILE EXIST	létező file-ra írás-megnyitási kísérlet
44	DIRECTORY FULL	megtelt a katalógus
46	WRITE ERROR	íráshiba a katalógusban file nyitáskor
47	WILDCHAR(S) IN FILE NAME	? vagy * file típusban
51	CHANNEL NOT OPENED	írási kísérlet nem nyitott csatornára
52	WRITE ERROR	írási hiba adatíráskor
59	DISK FULL	megtelt a lemez adatíráskor
62	READ ERROR	katalógus olvasási hiba file lezárásakor
63	WRITE ERROR	katalógus írási hiba file lezárásakor
65	FREE BLOCK	szabad blokk címe (nem hibát)
72	READ ERROR	katalógus olvasási hiba (SCRATCH FILE)
73	WRITE ERROR	katalógus írási hiba (SCRATCH FILE)
82	READ ERROR	katalógus olvasási hiba (VALIDATE)
83	MULTIUSED BLOCK	többszörösen hivatkozott szektor (VALIDATE)
84	READ ERROR	olvasási hiba VALIDATE parancs alatt
98	ILLEGAL DOS FORMAT	alacsonyabb DOS verziószám a lemezen
99	ILLEGAL DOS FORMAT	nem egyezik a DOS kód a katalógusban
101	CHANNEL NOT OPENED	buffer pointer parancs nem nyitott csatornára
102	READ ERROR	olvasási hiba BLOCK READ paranccsal
103	WRITE ERROR	íráshiba BLOCK WRITE-nál
111	SYNTAX ERROR	szintaktikus hiba a parancsban
112	NO IMPLEMENTED COMMAND	nem létező parancs
127	NO MORE BLOCK	nincs több szabad blokk (GET BLOCK)
130	DRIVE NOT READY	a floppy nem kész
131	ILL. TRACK	nem létező track-re való hivatkozás
132	SECTOR=0.	0-ás szektorcímre való hivatkozás
133	SECTOR>13.	nem létező szektorra való hivatkozás
136	WRITE CHECK ERROR	az írásellenőrzés sikertelen
137	TRACK00 NOT FOUND	a 0-ás track nem olvasható
138	SEEK ERROR	a keresett track nem olvasható
200	DISK WRITE PROTECTED	írási kísérlet írásvédett lemezre
201	RESTORE TIME OUT	sikertelen 0-ás track keresés
202	WRITE FAULT IN DRIVE	drive íráshiba jelzés