

**VT-DOS KOMPATIBILIS
FLOPPY DISZK**



VIDEOTON

TV-Computer

**VT-DOS KOMPATIBILIS FLOPPY DISZKES RENDSZER
VIDEOTON TV-COMPUTERHEZ
FELHASZNÁLÓI KÉZIKÖNYV
210-06198-11**

- 49122 (V6198) Floppy csatoló TVC-hez
- 49118 (V7583) Nagykapacitású duál minifloppy egység
- 49123 (V7632) Nagykapacitású duál minifloppy egység
- 49119 (V7607) Nagykapacitású minifloppy egység
- 49124 (V7682) Nagykapacitású minifloppy egység
- 49121 (V7608) Nagykapacitású bővítő minifloppy

Készült a Forma-Art Kiadó gondozásában
KÖNYOMAT Sokszorosító és Szolgáltató Kiszövetkezet, 88/19
Felelős vezető: ifj. Kasza Ferenc

TARTALOMJEGYZÉK

| | | |
|------|----------------------------------|----|
| 1. | BEVEZETÉS | 9 |
| 2. | ÜZEMBE HELYEZÉS ÉS KEZELÉS | 12 |
| 2.1. | Üzembe helyezés | 12 |
| 2.2. | Kezelés | 12 |
| 3. | HASZNÁLAT A BASIC-BŐL | 14 |
| 4. | A BASIC CLI | 18 |
| 4.1. | Bevezetés | 18 |
| 4.2. | Jelölések | 18 |
| 4.3. | A parancsok leírása | 22 |
| | CD | 22 |
| | CHDIR | 22 |
| | CLS | 24 |
| | COPY | 24 |
| | DATE | 26 |
| | DEL | 27 |
| | DIR | 27 |
| | DOS | 29 |
| | ERA | 29 |
| | ERASE | 29 |
| | FORMAT | 30 |
| | HELP | 31 |
| | LDIR | 31 |
| | LTYPE | 32 |
| | MD | 32 |
| | MKDIR | 32 |
| | MOVE | 33 |

| | |
|--------------|----|
| RD | 34 |
| REN | 34 |
| RENAME | 34 |
| RMDIR | 35 |
| RNDIR | 35 |
| TIME | 36 |
| TYPE | 37 |
| VAR | 38 |
| VOL | 39 |

| | | |
|-----------|--|----|
| 5. | MÉLYEBB ISMERETEK A FLOPPY-KEZELŐ PROGRAMOKRÓL | 41 |
| 5.1. | A diszk-ROM szervezése | 41 |
| 5.2. | A MOPS DISK DEVICE (MDD) | 42 |
| 5.2. 1. | Bevezetés | 42 |
| 5.2. 2. | Fájl létrehozása: fájl megnyitása | 43 |
| 5.2. 3. | Fájl írása és olvasása | 44 |
| 5.2. 4. | Fájl ellenőrzése | 45 |
| 5.2. 5. | Fájlmutató és fájl méret beolvasása ill. beál- lítása | 45 |
| 5.2. 6. | Fájl lezárása | 46 |
| 5.2. 7. | CLI puffer definiálása | 46 |
| 5.2. 8. | CLI parancs végrehajtása | 47 |
| 5.2. 9. | FISH-funkció végrehajtása | 48 |
| 5.2.10. | Hibák | 49 |
| 5.2.11. | Kazetták használata | 50 |
| 5.3. | A VT-DOS fájlkezelője (FISH) | 51 |
| 5.3.1. | Bevezetés | 51 |
| 5.3.2. | Inicializálás | 51 |
| 5.3.3. | A diszk szervezése | 52 |
| 5.3.3. 1. | A BOOT-szektor | 53 |
| 5.3.3. 2. | Fájl elhelyezési tábla (FAT) ... | 54 |

| | | |
|--------|--|----|
| 5.3.3. | 3. A könyvtár-bejegyzések formátuma | 55 |
| 5.3.3. | 4. Kötetazonosító | 56 |
| 5.3.3. | 5. „Piszkos diszk”-jelző, törölt fájlok visszaállítása | 56 |
| 5.3.4. | FISH funkcióhívások | 57 |
| 5.3.4. | 1. Általános FISH-hívási konvenciók | 57 |
| 5.3.4. | 2. Rendszerváltozók | 59 |
| 5.3.4. | 3. File Control Block-ok (FCB-k) | 59 |
| | 5.3.4.3.1. FCB típusok. | 60 |
| | 5.3.4.3.2. Az FCB-k formátuma | 61 |
| | 5.3.4.3.3. Útvonalnév-stringek. | 63 |
| | 5.3.4.3.5. Kötetazonosítók az FCB-ben | 66 |
| 5.3.4. | 4. Útvonalnév-string elemzése | 67 |
| 5.3.4. | 5. Könyvtár-változtatás | 69 |
| 5.3.4. | 6. Aktuális könyvtár beolvasása | 70 |
| 5.3.4. | 7. Első keresése | 71 |
| 5.3.4. | 8. Következő keresése | 72 |
| 5.3.4. | 9. Fájl létrehozása | 73 |
| 5.3.4. | 10. Könyvtár létrehozása | 74 |
| 5.3.4. | 11. Törlés | 75 |
| 5.3.4. | 12. Átnevezés | 76 |
| 5.3.4. | 13. Áthelyezés | 76 |
| 5.3.4. | 14. Attributumok változtatása | 77 |
| 5.3.4. | 15. Fájlmegnyitás | 78 |
| 5.3.4. | 16. Fájllezárás | 78 |
| 5.3.4. | 17. Pufferek törlése | 79 |
| 5.3.4. | 18. Olvasás fájlból | 80 |
| 5.3.4. | 19. Írás fájlba | 80 |
| 5.3.4. | 20. Írás O-töltéssel | 81 |
| 5.3.4. | 21. Abszolút szektor olvasás | 82 |

| | | |
|-----------|---|-----|
| 5.3.4.22. | Abszolút szektor írás | 82 |
| 5.3.4.23. | Login vektor beolvasása | 83 |
| 5.3.4.24. | Lemez-fogl. információk beolvasása | 83 |
| 5.3.4.25. | Logikai meghajtó konvertálása fizikai meghajtóvá | 85 |
| 5.3.4.26. | Lemez formattálása | 86 |
| 5.3.4.27. | MAPDISK | 87 |
| 5.3.4.28. | ASSIGN | 88 |
| 5.3.4.29. | Rendszerváltozók beolvasása/be- állítás | 88 |
| 5.3.4.30. | Aktuális dátum beolvasása | 89 |
| 5.3.4.31. | Aktuális dátum beállítás | 89 |
| 5.3.4.32. | Aktuális idő beolvasása | 90 |
| 5.3.4.33. | Aktuális idő beállítás | 90 |
| 5.3.4.34. | A diszk-ROM verziószámának be- olvasása | 90 |
| 5.3.5. | Pufferelés diszk-műveleteknél | 91 |
| 5.3.6. | Hibakezelés | 91 |
| | 5.3.6. 1. A beépített diszkhiba-kezelés | 92 |
| | 5.3.6. 2. A felhasználói hibavektor | 92 |
| 5.3.7. | Változók a diszk-RAM területén | 94 |
| 5.4. | A közvetlen diszk-kezelő modul (DISKIO) | 95 |
| 5.4.1. | Bevezetés | 95 |
| 5.4.2. | Parancsok és paraméterek | 96 |
| 5.4.3. | A diszk-átviteli cím | 98 |
| 5.4.4. | DISZK RENDSZER RESET parancs | 98 |
| 5.4.5. | Szektorok olvasása, írása és ellenőrzése | 99 |
| 5.4.6. | Formattálás és pályaolvasás | 100 |
| 5.4.7. | Cím-olvasás | 101 |
| 5.4.8. | A meghajtó vizsgálata | 102 |
| 5.4.9. | Hibák és javasolt újra-próbálkozási stratégiák | 103 |

| | | |
|----------|--|-----|
| 6. | SPECIFIKÁCIÓ | 105 |
| 6.1. | Floppy csatoló | 105 |
| 6.2. | Mini-floppy egység | 106 |
| 7. | HARDVER-SZOFTVER INTERFÉSZ | 108 |
| 7.1. | A memória felépítése | 109 |
| 7.2. | Memória „paging” regiszter | 109 |
| 7.3. | Paraméter regiszter | 109 |
| 7.4. | Hardver állapot regiszter | 111 |
| 7.5. | Összefoglaló táblázat az I/O eszközökről | 111 |
| 7.6. | Floppy Disc Controller (FDC) | 112 |
| Függelék | | |
| I. | Változók | 115 |
| II. | Hibajelzések | 120 |
| III. | A parancsok áttekintése | 130 |

JELLEN LEÍRÁS ELOLVASÁSA AJÁNLTATOS A BERENDEZÉS
ÜZEMBE HELYEZÉSE ELŐTT

1. BEVEZETÉS

A Videoton TV-Computer alapkiépítése esetén az adatok, programok tárolására magnókazetta szolgál. Ez a tárolási mód nem elégíti ki az összes alkalmazási területet. Példaként lehet említeni a kis üzleti alkalmazásokat, ahol nagyobb mennyiségű adatot illetve programot kell tárolnunk azok gyors elérhetősége mellett.

Ezt biztosítja a TV-Computer floppy diszkes kiegészítő berendezés, mely a következő egységekből épül fel:

- floppy csatoló kártya
- nagykapacitású mini floppy vagy duál mini floppy periféria

A minifloppy periféria 1 db 5 1/4 inch-es hajlékonylemezes meghajtót tartalmaz, egy-egy hajlékony lemezre 720 kb-ot (737280 bájt) információ írható. A duál minifloppy 2 db meghajtót tartalmaz.

Általános tudnivalók

A TVC-hez kétféle floppy csatoló létezik:

- UPM kompatibilis verzió
- VT-DOS kompatibilis verzió

A VT-DOS kompatibilis verzió a diszk szervezésének és fájlkezelésének az azonosságát jelenti a TVC VT-DOS rendszerben alkalmazottal.

A VT-DOS kompatibilis floppy csatoló önmagában nem jelent VT-DOS operációs rendszert, csupán azt biztosítja, hogy a TVC a BASIC rendszer alatt használhat diszkes háttértárat.

A floppy csatolós TVC-hez a VIDEOTON kívánságra VT-DOS operációs rendszert is tud szállítani. A VT-DOS operációs rendszer indítása program-modul csatlakoztatással történik.

A VT-DOS kompatibilis floppy-csatolós TVC az MS-DOS operációs rendszerrel kompatibilis lemezformátummal dolgozik.

A VT-DOS operációs rendszer alatt futtathatók a CP/M 2.2. rendszer alá írt programok, ehhez csupán a fájl-szerkezet konvertálására van szükség, amit a CONVERT paranccsal elvégezhetünk.

Az MS-DOS elnevezés a Microsoft, a CP/M 2.2. elnevezés a Digital Research cég védjegyzett elnevezése.

Ez a leírás a BASIC rendszer alatti VT-DOS kompatibilis lemezkezelést ismerteti, de nem foglalkozik a VT-DOS operációs rendszerrel.

Minden floppy csatoló kártya bármely bővítő kártyahelyre tehető. Egy TVC-ben egyszerre csak egy floppy csatoló képes helyesen működni.

Érintésvédelmi előírás

Mivel a floppy periféria egység működéséhez szükséges tápfeszültségeket előállító hálózati tápegység a meghajtókkal közös burkolatban helyezkedik el, a burkolat eltávolítása **életveszélyes és tilos.**

A floppy-s periféria és a TVC csatlakoztatásához a rendszerhez tartozó összes berendezést, azaz a TV-Computert, a floppy perifériát, a TV készüléket, a magnókészüléket, nyomtatót, stb. ki kell kapcsolni!

Esetleges meghibásodás esetén a szervízhálózat szakembereit kell értesíteni.

Érintésvédelmi osztály I.

A készüléket csak védőérintkezős hálózati csatlakozó aljzatról lehet biztonságosan üzemeltetni!

! Biztosítékot csak a készülék hátlapján feltüntetett értékű biztosítékra szabad cserélni!

Néhány szó a floppy-lemez szervezéséről és a csatolóról

A floppy-lemez mágneses anyagára koncentrikus körök, az ún. pályák, vagy track-ek mentén írjuk fel az információt. A pályákat szektorokra osztjuk.

Az egymáshoz tartozó adatok (pl. egy program, egy levél szövege, stb.) fájlokat alkotnak. A fájlokat a VT-DOS egy adott lemezen egy, vagy több könyvtárba rendezheti. Minden lemezen van egy kiindulási (root) könyvtár, amelynek bejegyzései fájlokra és alkönyvtárakra is vonatkozhatnak. Az alkönyvtárakon belül újabb alkönyvtárakat is létrehozhatunk – határt csak a lemez mérete szab.

A fájlokhoz (amelyeknek neve egy főrészből és egy kiegészítőből áll) így egy ún. útvonal (path) is tartozik, ami azon könyvtárak nevéből áll, amelyeken keresztül a fájlhoz el lehet jutni.

A csatolón ROM és RAM memória is található, ezeket a könyvben gyakran „diszk-ROM” ill. „diszk-RAM” néven emlegetjük.

A diszk-ROM tartalmazza a közvetlen hardverkezelést végző programot, ezt DISKIO-nak fogjuk hívni, tartalmazza az előbb említett fájl- ill. könyvtárkezelést végző programokat (FISH), egy olyan programot, amely a TV-Computer hagyományos operációs rendszerében (a MOPS-ban) való floppy kezelést valósítja meg a FISH-en és a DISKIO-n keresztül (MDD), végül egy negyedik modult, amely a VT-DOS néhány fontos, a floppy-k használatához szükséges parancsát teszi hozzáférhetővé a BASIC-ből. Ezt a programot „BASIC CLI”-nek nevezzük.

2. ÜZEMBE HELYEZÉS ÉS KEZELÉS

2.1. Üzembe helyezés

Kicsomagolás után győződjünk meg arról, hogy a szállítás során nem szenvedett-e fizikai sérülést a berendezés, majd a következő műveleteket végezzük el:

- a TVC kikapcsolt állapotában az interfész kártyát helyezzük a TVC-be
- a floppy egység logikai (szalag) kábelét csatlakoztassuk az interfész kártyához
- kapcsoljuk be a TVC-t
- kapcsoljuk be a floppy egységet
- helyezzük a lemezt (lemezeket) a meghajtóba (meghajtókba) olyan módon, hogy a lemezen lévő címke a jobb oldalra nézzen
- rögzítsük a lemezt a zárszerkezettel.

2.2. Kezelés

A tároló működés közbeni kezelése a lemezek ki- és behelyezésén kívül más tennivalót nem igényel, mivel a tároló működését a HBF kártyán lévő elektronika vezérli, az adott kezelő programnak megfelelően.

Két meghajtós (duál) egység esetén a meghajtók az alábbiak szerint különböztethetők meg:

- 0-ás egység: a bal oldalon lévő meghajtó
- 1-es egység: a jobb oldalon lévő meghajtó

A működtető programok vagy a fizikai sorszámra (0, 1), vagy egy logikai elnevezésre hivatkoznak.

A szokásos logikai elnevezés:

0-ás egység: "A" meghajtó

1-es egység: "B" meghajtó

A tároló megfelelő, üzembiztos működéséhez elengedhetetlen az adathordozó megfelelő, szakszerű tárolása, melyre vonatkozó szabályok általában az adathordozó védőtasakján megtalálhatók, vagyis:

- A floppylemezt használat után helyezzük vissza a védőtasakba.
- A hordozót tartjuk távol mindenféle mágneses anyagtól.
- A védőborítékra ne írjunk semmit, erre a célra felragasztható címkék találhatók a lemezcsomagban.
- Kézzel ne érintsük meg a hordozó felületét.
- Védjük a hőtől, napsugárzástól, portól.
- Ne gyúrjuk, ne hajlítsuk meg a lemezt, ne tűzzünk hozzá iratokat, stb.

3. HASZNÁLAT A BASIC-BŐL

A floppy automatikusan helyettesíti a kazettát, így a kazettára vonatkozó BASIC utasításokat és parancsokat minden további nélkül használhatjuk.

A BASIC CLI parancsaihoz az EXT2 parancs kiadása után férhetünk hozzá. Ekkor lehet aktuális meghajtót, ill. aktuális könyvtárat is kijelölni.

Példák a floppy diszkre vonatkozó BASIC utasításokra és parancsokra:

Programbetöltés:

| | |
|---------------------|--|
| LOAD"CY*" | Az aktuális meghajtón az aktuális könyvtárban lévő első CY névkezdetű fájl betöltése |
| LOAD"B:CYRUS" | A B: meghajtón lévő CYRUS program betöltése |
| LOAD"B:CYRUS.CAS" | A B: meghajtón lévő CYRUS program betöltése |
| LOAD"B:\KONY\CYRUS" | A B: meghajtó KONY könyvtárában lévő CYRUS program betöltése. |

Program kimentés

| | |
|-------------|---|
| SAVE"PROG1" | Az aktuális meghajtón lévő PROG1.CAS fájlba kerül a BASIC program kimentésre. |
|-------------|---|

SAVE "B:\KONY\PROG1"

A B: meghajtón a KONY könyvtárban lévő PROG1.CAS fájlba kerül a BASIC program kimentésre.

Programkimentés ellenőrzése

VERIFY "B:\KONY\PROG1.CAS"

A B: meghajtón a KONY könyvtárban lévő PROG1 fájlal hasonlítja össze a memóriatartalmat a rendszer.

Fájl megnyitás olvasásra

OPEN "NEV"

A "NEV" megadásánál ugyanazok a lehetőségek vannak, mint programbetöltésnél.

Fájl megnyitás írásra

OPEN OUTPUT "NEV"

A "NEV" megadásánál ugyanazok a lehetőségek vannak, mint a program kimentésnél.

Példa az adatkivitelre

A következő programrészlet a 20 elemű B tömb elemeit írja fel az ADATOK.CAS nevű fájlba az aktuális meghajtón.

```
100 OPEN OUTPUT "ADATOK"  
110 FOR I=0 TO 19  
120 PRINT #5:B(I)  
130 NEXT  
140 CLOSE OUTPUT
```


A következő programrészlet a 20 elemű C tömb elemeit tölti fel az aktuális meghajtón lévő ADATOK.CAS fájlból:

```
200 DIM C(19)
210 OPEN "ADATOK"
220 FOR I=0 TO 19
230 INPUT #5: C(I)
240 NEXT
250 CLOSE
```

A BASIC parancsok végrehajtásakor, ha hiba van a rendszer kiírja:

***System error XXX.

Ahol XXX a visszaadott hibakód decimális értéke.

A visszaadott hibakódok:

| Hibakód | Jelentés |
|---------|---|
| 128 | Nemlétező fájl, hiba opennél |
| 129 | Fájl létrehozási hiba |
| 131 | Close hiba |
| 132 | írási hiba |
| 133 | Olvasási hiba |
| 230 | Védett fájl másolási kísérlet |
| 231 | Belső hiba. Helytelen fájl-típus (A fájl-típus pufferezt vagy nem pufferezt lehet: 1 v. 11H) |
| 232 | Ellenőrzési hiba |
| 233 | Nincs nyitott fájl |
| 235 | Túl sok nyitott fájl |
| 236 | Fájl vége |
| 239 | Fájlnév hiba |
| 245 | Stop karakter konzolról |

Kazetta és floppy diszk egyidejű használata

Ha szükséges, a #5 input és output átrendelhető kazettához ill. visszarendelhető floppy-hoz.

Az input átrendelése kazettához:

PO KE 2821, 5

Az output átrendelése kazettához:

PO KE 2829, 5

Az input visszarendelése diszkhez:

PO KE 2821, Z

ahol Z értéke:

| Diszk csatoló kártyahely száma | Z értéke |
|--------------------------------|----------|
| 0 | 128 |
| 1 | 129 |
| 2 | 130 |
| 3 | 131 |

Az output visszarendelése diszkhez:

PO KE 2829, Z

ahol Z értéke fenti táblázat szerinti.

A bővítő kártyahelyek közül a 0-ás kártyahely a jobbszélső.

4. A BASIC CLI

4.1. Bevezetés

A BASIC CLI az

EXT 2

paranccsal indítható el. Ekkor megjelenik egy kijelentkező üzenet (prompt) és használhatók lesznek a 4.3. fejezetben leírt parancsok.

A BASIC-be visszatérni az ESC billentyű megnyomásával lehet. A BASIC programunk és a változók érintetlenek maradnak a BASIC CLI használata során.

A parancssorokat a MOPS-nál megszokott módon lehet begépelni.

Hiba esetén az

***Unrecognised command (ismeretlen parancs)

vagy az

***Error XXX

üzenetet írja ki a gép.

4.2. Jelölések

A 4.3. fejezetben a parancsok leírásánál a következő jelöléseket fogjuk alkalmazni:

Nagybetűs szavak

Ezek kulcsszavak, a megadott módon kell őket begépelni, de a kis- és nagybetűk tetszőlegesen keverhetők.

Kisbetűs elemek

Ezek paraméterek, amelyeket a parancssor megjelölt helyén meg kell adni.

Elemek szögletes zárójelben ([és])

Ezek opcionális tételek (nem feltétlenül kell megadni őket).
A zárójeleket nem kell begépelni.

Függőleges vonallal (|) elválasztott elemek

Ez azt jelenti, hogy a két elem közül az egyik (de nem mindkettő) szükséges.
A függőleges vonalat nem kell begépelni.
A továbbiakban a parancssorban előforduló elemeket soroljuk fel:

d:

Ez azt jelenti, hogy meghajtó-nevet kell beírni. Ha nem tesszük, az az implicit meghajtót jelenti. Érvényes meghajtó-nevek: A:-tól D:-ig.

path

Azt jelzi, hogy könyvtár-útvonalat kell megadni, minden könyvtárnevet egy „\” karakterrel elválasztva. Ha az útvonal elején „\” van, az azt jelenti, hogy az útvonal a kiindulási könyvtárban kezdődik; ha nincs az elején „\”, akkor pedig az aktuális könyvtárban.

Két egymás utáni pont „. . .” az útvonalban a közvetlen „szülő” könyvtárat jelzi. Az egyetlen pont „.” az útvonalban az aktuális könyvtárat jelöli, így nincs értéke az útvonal-specifikációban.

Mindkét esetben használni kell a „\”-t, jelezve, hogy a pontok könyvtárra vonatkoznak.

A „\” karakter helyett használható a „!” és a „' ” is.

A könyvtárnevek szintaktikája megegyezik a fájlnevek alább közölt szintaktikájával.

filename

Azt jelzi, hogy fájlnevet kell megadni. Többértelműnek nevezzük az olyan fájlnevet, amelyben „*”, vagy „?” karakter van, így a lemezen több fájlnek is megfelel. Az ilyen karaktereket nem tartalmazó fájlneveket egyértelműnek nevezzük.

A fájlnevek szintaxisa a következő: mainname.suf

ahol a „mainname” egy maximum 8-elemű karaktersorozat, a „suf” pedig egy maximum 3-karakteres kiegészítés.

Ha megadunk kiegészítést, akkor ponttal kell elválasztani a „mainname”-től.

A fájlnev egyik részében sem használhatók a következő karakterek:

vezérlő kódok és a szóköz (00H ... 20H)

: ; . , = + \ < > | / " [] # !

A fájlnevben előforduló betűket a VT-DOS nagybetűkké konvertálja, így a kis- és nagybetűk jelentése megegyezik.

A fájlnev gyakran egy útvonalnevet követ, ilyenkor a kettőt „\” karakterrel lehet elválasztani.

filespec

Gyakran használt jelölés a fájl megadására, amelyre a parancs vonatkozik. Szintaxisa: [d:] [path] [filename].

A három opcionális elem közül legalább egyet meg kell adni. Könyvtárra vonatkozó parancs esetén a fájlnevnél egy, vagy több létező könyvtárat kell specifikálnia. Fájlra vonatkozó parancs esetén a fájlnevnél egy, vagy több létező fájlt, vagy egy egyértelmű könyvtárnevet kell jelentenie, amikor is egy „.” fájlnevet feltételezünk a megadott könyvtárban.

volname

Azt jelzi, hogy egy kötetnevet kell megadni. A kötetnév egy maximum 11-elemű karaktersorozat, amely tartalmazhat szóközöket és a fájl-nevekben nem megengedett karaktereket is, kivéve a vezérlő kódokat és a „\” karaktert. Az elején esetleg előforduló szóközöket a VT-DOS kiszűri.

device

Készülék megadását jelenti, ami a következők egyike lehet:

- CON:** A konzol. A karaktereket a billentyűzetről olvassa és a képernyőre írja.
- PRN:** A nyomtató. A karaktereket a nyomtatóra írja. Az olvasás nem érvényes a PRN készülékről.
- AUX:** Az RS-232 soros interfész. A karaktereket a soros kártyáról olvassa és oda írja. A VT-DOS csak egy soros kártyát támogat, azt, amelyik a legkisebb sorszámú bővítő kártyahelyen van.
- NUL:** Egy „üres”, „semmit sem csináló” periféria. Az írt karaktereket egyszerűen figyelmen kívül hagyja, olvasási kísérlet esetén „fájl vége” hibát ad vissza.

number Azt jelzi, hogy egy 0 és 255 közé eső előjel nélküli számot kell megadni.

Egy parancs paramétereit általában szóközökkel, vagy TAB-okkal kell a parancsnévtől és egymástól elválasztani (ez a két karakter a CLI számára egyenértékű). Kivétel ez alól a „/” karakter, amely parancs-opció kezdőkaraktere lehet.

4.3. A parancsok leírása

Ez a fejezet részletesen ismerteti a BASIC CLI valamennyi parancsát, felhasználva a 4.2. fejezetben leírt jelöléseket.

A „bejegyzett” meghajtó megváltoztatására kiadható a

d:

parancs, amely a d: meghajtót jelöli ki. A CLI prompt-ja ezután megváltozik erre a meghajtó-azonosítóra.

CD

Lásd a CHDIR parancsot.

CHDIR

Formátuma: CHDIR [d:] [path]
vagy
CD [d:] [path]

Feladata: Megjeleníti vagy megváltoztatja az aktuális könyvtárt.

Használata: Ha nincs megadva útvonal [path], akkor a kijelölt, vagy az implicit meghajtó aktuális könyvtár-útvonalát írja ki: ez a kiindulási könyvtárból az aktuális könyvtárba vezető útvonal.

Ha adtunk meg útvonalat, akkor a kijelölt vagy az implicit meghajtó aktuális könyvtára a megadott útvonallal meghatározott könyvtár lesz.

A CD parancs a CHDIR rövidített formája, amely kényelmi és MS-DOS kompatibilitási célokat szolgál.

Példák: CHDIR \BOOT\RAMDISK

Az aktuális meghajtó aktuális könyvtára megváltozik a BOOT\RAMDISK könyvtárra (a kiindulási könyvtárhoz képest).

CHDIR A:UTIL

Az A: meghajtó aktuális könyvtára megváltozik – az aktuális könyvtárból kiágazó UTIL könyvtárra.

CD

Nem adtunk meg sem útvonalat, sem meghajtót, így az aktuális meghajtó aktuális könyvtárát kiírja, pl:

E:\BOOT\RAMDISK

CHDIR A:

Csak a meghajtót specifikáltuk: a meghajtó aktuális könyvtárát írja ki, pl:

A:\UTIL

CLS

Formátuma: CLS

Feladata: Törli a képernyőt.

Használata: Egyszerűen törli a képernyőt, a cursort pedig a HOME pozícióba viszi.

Példa: CLS

COPY

Formátuma: COPY forrás [/A] [/H] [cél[/A]][/T]]

Feladata: Adatokat másol egy fájlból vagy egy készülékről egy másik fájlba, vagy készülékre.

Használata: A „forrás” felépítése: filespec | device

Ha „filespec”-et adunk meg, akkor a kijelölt fájlokat másolja, amelyek a /H opció megadása esetén rejtett fájlok is lehetnek.

Ha „device”-ot adunk meg, akkor az adatokat a megfelelő I/O-készülékről olvassa.

A „cél” felépítés: filespec | device

Ha a „filespec” többértelmű karaktereket („?” vagy „*”) is tartalmaz, akkor a többértelműnek jelzett pozíciókra a „forrás” fájlnev megfelelő karakterei kerülnek – ez lehetővé teszi a fájlok átnevezését a másolás alkalmával. Ha a forrás I/O-készülék és a cél neve többértelmű, akkor a COPY hibát jelez.

Az /A opció szabja meg, hogy a másolt adatokat ASCII-ként vagy binárisként kell kezelni az alábbiak szerint.

A forrásnál a /A azt jelenti, hogy az olvasás az első CTRL-Z karakterig (vagy a fájl végéig) tart. A CTRL-Z nem másolódik át. A /A hiányában minden adatot beolvas a COPY.

A cél-oldalon a /A azt jelzi, hogy a másolás az első CTRL-Z karakterig tart. E karaktert a COPY nem írja föl, viszont a fölírt adatsor végére egyetlen CTRL-Z karaktert ír. A /A hiányában minden adatot fölír és semmilyen karaktert sem ír a végére.

Ha forrásként többértelmű fájlnevet adtunk meg, akkor a fájlnevek egymás után megjelennek az átmásolásakor. Ha hiba fordul elő a másolás közben, vagy megnyomják a CTRL-ESC (stop) billentyűt, akkor minden szükséges fájllezárást elvégez a VT-DOS és jelzi a hibát. Bizonyos körülmények között a CTRL-ESC az egyetlen mód a COPY leállítására, pl. ha a soros vonalról másolunk a /A opció nélkül.

Ha több fájlt másolunk egy I/O-készülékre, akkor a másolás fájlanként történik; az adatok végére szükség esetén CTRL-Z kerül.

Ha fájlból fájlba másolunk, akkor a jellemzők nem változnak: a cél-fájl attribútumai meg fognak egyezni a forrás-fájl jellemzőivel. Hasonlóképpen, a cél-fájl a forrás dátum- és időadatait kapja, kivéve ha megadjuk a /T opciót, amikor az aktuális dátum és idő kerül bele. Az attribútumok az ATTR paranccsal változtathatók meg.

Példák: COPY FRED B:

A FRED fájlt átmásolja a B: meghajtó aktuális könyvtárába.

COPY A: \BOOT\AUTOEXEC.BAT B:\

Az A: meghajtó megadott könyvtárából az AUTOEXEC.BAT fájlt átmásolja a B: meghajtó kiindulási könyvtárába.

COPY A:\BOOT B:\BOOT

Az A: meghajtó BOOT könyvtárában levő valamennyi fájlt átmásolja a B: egy hasonló könyvtárába.

COPY B:

A B: meghajtó aktuális könyvtárából minden fájlt átmásol az aktuális meghajtó aktuális könyvtárába.

COPY *.TXT PRN:

Az aktuális meghajtó aktuális könyvtárából minden TXT névkiegészítésű fájlt átmásol a nyomtatóra.

DATE

Formátum: DATE [dátum]

Feladata: Megjeleníti vagy beállítja az aktuális dátumot.

Használata: Ha a parancs után megadjuk a dátumot, akkor a dátumot beállítja erre az értékre (a formátumot l. később). Ha nem adunk meg dátumot, akkor az aktuális dátumot megjeleníti a hét napjával együtt, s megkérdezi, hogy új dátumot akarunk-e beírni. Ha nem írunk be új dátumot (azaz rögtön RETURN-t ütünk le), akkor az aktuális dátum nem változik. Egyébként a begépett sort új dátumként, az alább leírt formátum szerint értelmezi. Ha érvénytelen dátumot gépeltünk be, akkor hibajelzést ad és újra kéri a dátumot.

A dátum formátuma rugalmas, a VT-DOS DTFORM változója (l. l. függelék) szabja meg. Ez a változó lehetővé teszi, hogy a dátumot az angol formátum (nap-hó-év), az amerikai formátum (hó-nap-év), a nemzetközi formátum (év-hó-nap), vagy bármilyen más formátum szerint írja ki, ill. várja a DATE.

Begépeléskor az elválasztó karakterek lehetnek; „,” „-” „.” „/” „:”, vagy szóköz. A szeparátor mindkét oldalán lehet szóköz. A dátumbeli számok bevezető 0-i nem kötelezőek. A nem megadott mezők megtartják az eredeti értéküket.

Példák: DATE 12-7-85

Az aktuális dátumot beállítja az adott értékre.

DATE

Nincs paraméter, így kiírja az aktuális dátumot, és megkérdezi, hogy akarunk-e újat megadni:

**Current date is Fri 12-7-85
Enter new date:**

DATE 85/2/1

A dátum formátumát előzőleg az ISO szerintire változtatták, a most megadott új dátumot így állítja be.

DEL

lásd ERASE

DIR

Formátuma: DIR [d:] [path] [filename] [/H] [/W] [/T] [/S]

Feladata: A lemezen levő fájlok nevét megjeleníti.

Használata: A meghajtó, az útvonal és a (többértelmű) fájlnev határozza meg, hogy mely fájlokat kell kilistázni. Ha megadjuk a /H opciót, akkor a rejtett fájlokat is kilistázza.

A listázás kétféle formátumban történhet. Ha megadjuk a /W opciót, akkor „széles” lista készül, amely soronként több

fájlnevet tartalmaz. Az alkönyvtárneveket, a fájl jellemzőit, a fájl létrehozásának dátumát és idejét ilyenkor nem jeleníti meg.

Ha nem adjuk meg a /W opciót, akkor 64-karakteres képernyő esetén minden sorban csak egy fájl neve jelenik meg a fájl jellemzőivel, méretével, létrehozásának dátumával és idejével együtt. A lista végén megjelenik a listázott fájlok száma; a bennük foglalt bájtok száma és a lemezen levő szabad bájtok száma. 32-karakteres képernyő esetén csak a fájl nevét és méretét írja ki.

A /T opció megadása esetén a fájl méret helyett a dátumot és az időt írja.

Ha a /T mellett a /S opciót is megadjuk, akkor minden mezőt kiír, és egy fájl adatai a képernyőn két sorba kerülnek.

Példák: DIR

Az aktuális meghajtó aktuális könyvtárában levő fájlok nevei megjelennek – méretükkel, valamint létrehozásuk dátumával és idejével együtt.

DIR B: /W

A B: meghajtóban levő lemez aktuális könyvtárában levő fájlnevek listázása úgy, hogy soronként több fájl név jelenik meg.

DIR A:\BOOT

A megadott könyvtárban levő fájlok és könyvtárak listázása.

DIR E:*.COM

Az E: meghajtó aktuális könyvtárában levő minden COM névkiegészítésű fájl és könyvtár listázása.

DOS

Formátuma: DOS

Feladata: Áttérés a BASIC-ből VT-DOS rendszerbe.

Használata: Ha bent van a gépben a VT-DOS dugaszolható programmodul, akkor felteszi a kérdést:

Enter VT-DOS (Y/N)? (Belépjek a VT-DOS-ba?)

„Y” válasz esetén elindul a VT-DOS.

Ha nincs bent a gépben a VT-DOS programmodul, akkor

*** No VT-DOS cartridge

üzenetet ír ki.

ERA

Lásd ERASE

ERASE

Formátuma: ERASE filespec [/H]

Feladata: Fájlok törlése a lemezeről.

Használata: A „filespec” meghatározza, hogy mely fájlokat kell törölni: a /H opció a rejtett fájlok törlését is lehetővé teszi.

Ha a megadott fájlok között csak olvasható is van, akkor ezt nem törli az ERASE, hanem tovább megy a következő fájlra.

Ha egyetlen fájlt sem tudott törölni, akkor hibajelzést ad.

Ha a fájlnev * . * (vagy annak tekintendő), mert

Erase all files (Y/N)?

kérdést, és választ vár. Ha a válasz nem „Y”, vagy „y”, akkor a fájl-törést nem hajtja végre. Ez az eljárás biztonsági célokat szolgál, hogy véletlenül ne lehessen egy könyvtár minden fájlját megsemmisíteni.

Példák: ERASE TEST.BAK

Törli a TEST.BAK fájlt.

```
DEL *.COM /H
```

Az aktuális meghajtó aktuális könyvtárában minden – akár rejtett, akár nem rejtett – COM névkiegészítésű fájlt töröl.

```
DEL B:\BOOT\AUTOEXEC.BAT
```

A megadott könyvtárban levő AUTOEXEC.BAT fájlt törli.

```
DEL B:\BOOT
```

Felteszi az

```
Erase all files (Y/N)?
```

Kérdést, és „Y”, vagy „y” válasz esetén a BOOT könyvtár valamennyi fájlját törli.

FORMAT

Formátuma: `FORMAT [d:] [volname] [/1] [/H] [/8]`

Feladata: Lemez formattálása.

Használata: Formattálja a megadott (vagy az implicit) meghajtóban levő lemezt. A „volname” lesz (ha megadták) az újonnan formattált lemez kötetneve.

Ha nem adunk meg opciókat, akkor a lemezt a meghajtón elérhető legnagyobb kapacitásra formattálja. A /1 opció hatására a lemezt a kétoldalas meghajtóban is csak egyoldalásra, a /H opció hatására a 80-pályás meghajtóban 40-pályásra, a /8 opció hatására a lemezt pályánként 8-szektorosra (egyébként 9-szektorosra) formálja.

A formattálás megkezdése előtt biztonsági okokból egy billentyű lenyomását kéri.

Példák: `FORMAT B:`

A B: meghajtóban levő lemezt a meghajtón elérhető maximális kapacitásra formálja, ha a

`Destroy data on drive B: (Y/N)?`

(megsemmisítem az adatokat a B: meghajtón?)

Kérdésre a válasz „Y”.

`FORMAT B:SOURCE/1/H/8`

A B: meghajtóban levő lemezt egyoldalás, 40-pályás, pályánként 8-szektoros lemezzé formattálja, amelynek kötetneve SOURCE.

HELP

Formátuma: `HELP`

Feladata: Felsorolja a BASIC CLI parancsait.

LDIR

Formátuma: `LDIR [d:] [path] [filename] [/H] [/W] [/T] [/S]`

Feladata: A lemezen lévő fájlok nevét nyomtatóra írja.

Használata: Ez a parancs ugyanúgy működik, amint a DIR parancs, de a listát a nyomtatóra készíti. A nyomtatási szélesség a képernyő üzemmódjától függ, ezért a /T és /S opciók hasznosak lehetnek.

LTYPE

Formátuma: LTYPE filespec [/H]

Használata: Ez a parancs ugyanúgy működik, mint a TYPE parancs, de a listát a nyomtatóra készíti.

MD

Lásd MKDIR

MKDIR

Formátuma: MKDIR [d:] path
vagy
MD [d:] path

Feladata: Új alkönyvtár létrehozása.

Használata: Az útvonal utolsó eleme az új alkönyvtár neve, amelyet létre kell hozni a megadott, vagy az implicit meghajtón. Ebből következik, hogy ha egyetlen eleme van az útvonalnak, akkor ezen a néven az aktuális könyvtárban hoz létre az MKDIR új könyvtárt. Ha az új könyvtárt rejtetté kívánjuk tenni, ezt az ATDIR paranccsal külön kell elvégezni.

Az MD parancs az MKDIR rövidített formája, amely kényelmi és MS-DOS kompatibilitási célokat szolgál.

Példák: MKDIR UTIL

Egy UTIL nevű új könyvtárt hoz létre az aktuális meghajtó aktuális könyvtárában.

MKDIR A:\UTIL\COM

Az A:\UTIL könyvtárban létrehoz egy COM nevű új könyvtárt.

MOVE

Formátuma: MOVE filespec [/H] [path]

Feladata: Fájlok áthelyezése egyik könyvtárból a másikba.

Használata: A "filespec" meghatározza, hogy mely fájlokat kell áthelyezni /H lehetővé teszi rejtett fájlok áthelyezését is.

A "path" azt a könyvtárt jelöli ki, amelybe a fájlokat át kell helyezni.

Ha egy fájlt olyan könyvtárba próbálunk áthelyezni, amelyben már van ugyanolyan nevű fájl, akkor az áthelyezési művelet megáll, és hibajelzés történik.

Példák: MOVE FRED \

Az aktuális meghajtó aktuális könyvtárából a FRED nevű fájlt áthelyezi az aktuális meghajtó kiindulási könyvtárába.

MOVE A: *.BAT/H \BOOT

Az A: meghajtó aktuális könyvtárában levő minden – akár rejtett, akár nem rejtett – BAT névkiegészítésű fájlt áthelyez a \BOOT könyvtárba.

MOVE \UTIL

Az aktuális meghajtó UTIL könyvtárában szereplő minden fájlt átmásol az aktuális könyvtárba.

RD

Lásd RMDIR

REN

Lásd RENAME

RENAME

Formátuma: RENAME filespec [/H] filename

Feladata: Egy vagy több fájl átnevezése.

Használata: A „filespec” meghatározza az átnevezhető fájlt: /H lehetővé teszi rejtett fájlok átnevezését is.

A „filename” megadja a fájlnak adandó új nevet. Az új névben szereplő „?” azt jelzi, hogy ezen a karakterpozíción a régi név megfelelő karaktere kerül az új névbe (a „*” kérdőjelek sorozatának felel meg).

Ha valamelyik átnevezendő fájl új neve már létezik, vagy érvénytelen, akkor a RENAME működése megszakad és hiba-jelzés történik.

Példák: RENAME FRED WOMBAT

A FRED fájl új neve WOMBAT lesz.

```
REN B:\SOURCE\* .MAC/H *.OLD
```

A B: meghajtó \SOURCE könyvtárában minden MAC névkiegészítésű fájl (beleértve a rejtett fájlokat is) OLD kiterjesztésű lesz.

RMDIR

Formátuma: RMDIR [d:] path [/H]

Feladata: Egy vagy több alkönyvtár törlése.

Használata: Ha a /H opciót megadjuk, a rejtett könyvtárak is törölhetők.

A „path” utolsó eleme a törlendő alkönyvtár(ak) neve (többértelmű is lehet). Ha valamelyik törlendő könyvtár nem üres, a parancs végrehajtása megszakad és hibajelzés történik.

Példák: RMDIR UTIL

Az UTIL könyvtár törlődik az aktuális meghajtó aktuális könyvtárából.

```
RMDIR A:\BOOT\FRED? /H
```

Minden, a FRED? névnek megfelelő könyvtár (pl. FRED1, FRED2) törlődik az A: meghajtó BOOT könyvtárából.

RNDIR

Formátum: RNDIR filespec [/H] filename

Feladata: Egy vagy több alkönyvtár átnevezése.

Használata: A „filespec” kijelöli az átnevezendő könyvtárakat: a /H opció lehetővé teszi rejtett könyvtárak átnevezését is.

Az útvonalnév utolsó eleme meghatározza egy vagy több átnevezendő könyvtárt a kijelölt, vagy implicit meghajtón. A „filename” megadja a könyvtárak új nevét (többértelmű is lehet). Az új névben a „?” helyére a régi név megfelelő pozíciójú karaktere kerül (a „*” kérdőjelek sorozatának felel meg).

Ha valamelyik átnevezendő könyvtár új neve már létezik, vagy érvénytelen, akkor a művelet megszakad és hibajelzés történik.

Példák: RNDIR UTIL COM

Az aktuális meghajtó aktuális könyvtárában levő UTIL könyvtár új neve COM lesz.

RNDIR A:\SOURCE\FRED? /H BILL?

Az a: meghajtó \SOURCE könyvtárában minden, a FRED? -nek megfelelő nevű könyvtár (pl. FRED1, FRED2) átneveződik úgy, hogy a FRED-ből BILL lesz (pl. BILL1, BILL2).

TIME

Formátuma: TIME [idő]

Feladata: Megjeleníti, vagy beállítja az aktuális időt.

Használata: Ha a parancs után időt is adunk meg, akkor az aktuális időt erre az értékre állítja be (a formátumot l. később). Ha nem adunk meg időt, akkor kiírja az aktuális időt, és kéri az új időt. Ha nem kap (azaz rögtön a RETURN billentyűt ütjük le), akkor az aktuális időt nem változtatja, egyébként a begépelte sort az alábbiakban leírt formátumú időadatnak tekinti. Ha ez érvénytelen, akkor hibajelzést ad és újból kéri az időt.

Az idő megjelenítésének formátuma HH:MM (óra:perc). A VT-DOS DTFORM változója határozza meg, hogy 12 órás, vagy 24 órás legyen az idő (l. l. függelék). Az idő begépelésekor az óra és a perc elválasztására használható karakterek:

, - . / : és szóköz

A szeparátor mindkét oldalán lehetnek szóközök. A számok elején levő nullák nem kötelezőek. A hiányzó mezők megtartják eredeti értéküket. Az idő begépelhető 12-órás formátumban is. Bár a másodpercek és századmásodpercek sohasem jelennek meg, be lehet őket gépelni.

Példák: TIME 16:45

Beállítja az aktuális időt 16:45-re, 24-órás formátumban.

TIME

Nincs paraméter, így kiírja az aktuális időt és kéri az újat, pl.:

Current time is 10:45a

Enter new time:

(12-órás formátum, délelőtt 10:45 van.)

TIME 2:30p

Beállítja az aktuális időt délután 1/2 3-ra.

TYPE

Formátuma: TYPE device | filespec [/H]

Feladata: Adatokat jelenít meg egy fájlból, vagy egy I/O-készülekről.

Használata: Beolvassa az ASCII-kódú adatokat a fájlból, vagy az I/O-készülekről és megjeleníti. A működés a fájl végéig, vagy egy CTRL-Z olvasásig tart. A nem megjeleníthető karaktereket (így a vezérlő karaktereket is) megjeleníthetőkké konvertálja a nemkívánatos cursor- és egyéb vezérlések elkerülése érdekében.

A /H opcióval rejtett fájl is megjeleníthető.

Példák: TYPE MYFILE

A MYFILE nevű fájl tartalmát kiírja.

TYPE AUX:

Adatokat olvas a soros vonalról és megjeleníti a képernyőn.

VAR

Formátuma: VAR number [[number]I[on]I[off]]

Feladata: Megjeleníti, vagy beállítja egy VT-DOS változó értékét.

Használata: Az első „number” annak a változónak a sorszámát jelenti, amelyre a parancs vonatkozik. A paranccsal változtatható és megjeleníthető változók sorszámait az I. függelék tartalmazza.

Ha nem adunk meg második paramétert, akkor a változó aktuális értéke megjelenik a képernyőn.

Ha a második paraméter szám, akkor a változó ezt az értéket kapja. Ha a második paraméter ON, vagy OFF, akkor a változó értéke 0, ill. 255 lesz.

Példák: VAR 0

Nincs második paraméter, így a 0. VT-DOS változó értékét kiírja.

VAR 0, 42

A 0. változó értéke 42 lesz.

VAR 0 OFF

A 0. változó értéke 255 lesz, ami gyakran „OFF”-ot (kikapcsolt állapotot) jelent.

VOL

Formátuma: VOL [d:] [filename]

Feladata: Megjeleníti, vagy megváltoztatja a lemez kötetnevét.

Használata: Ha nem adunk meg paramétert, vagy csak meghajtónevet adunk, akkor az aktuális, ill. a megadott meghajtóban levő lemez kötetnevét kiírja. Ha adunk meg fájlnevet, akkor ez lesz a lemez új kötetneve.

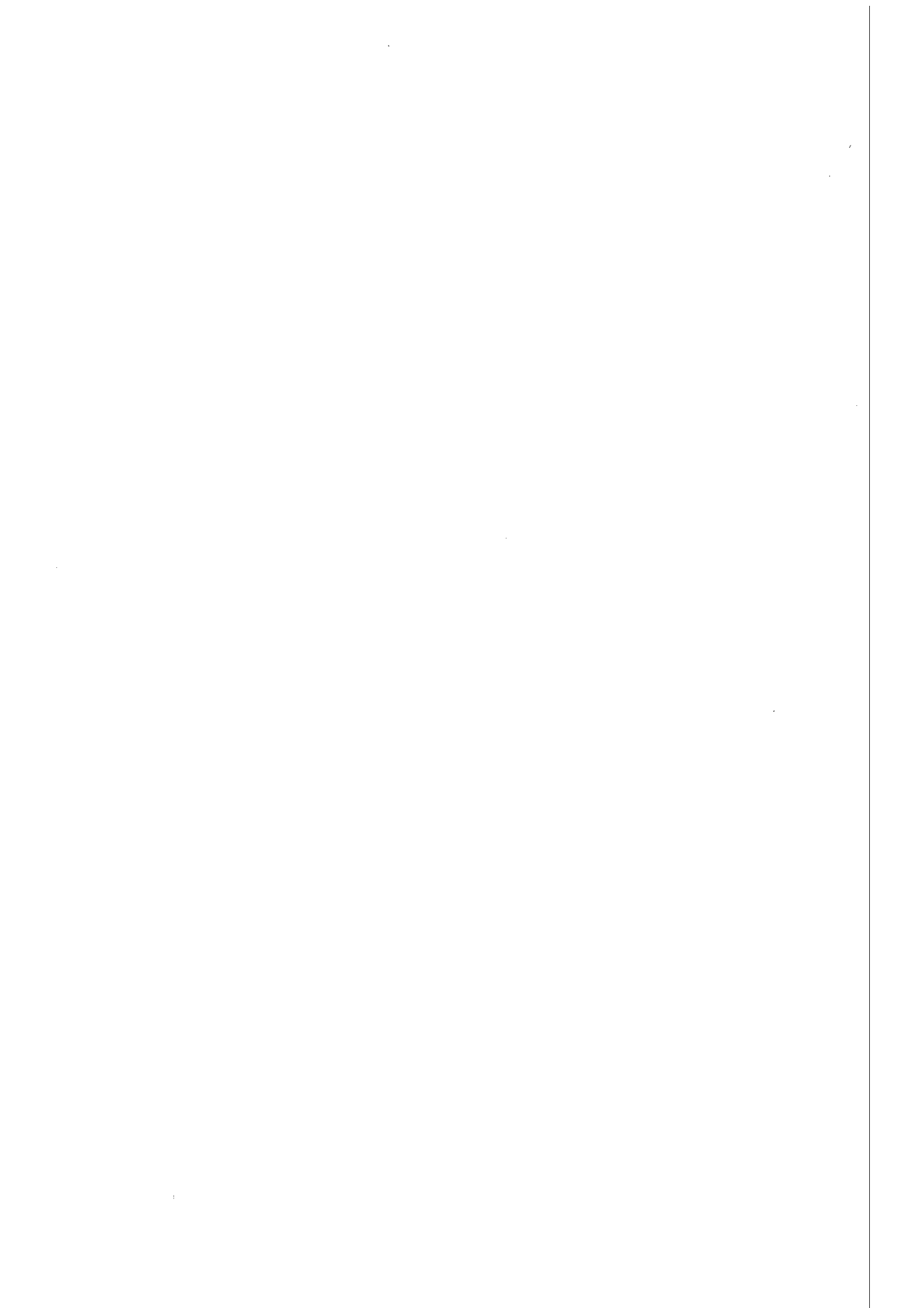
Példák: VOL B:

A B: meghajtóban levő lemez kötetnevét kiírja, pl. ha nincs név:

Volume in drive B: has no name

VOL BACKUP

Az aktuális meghajtóban levő lemez a BACKUP kötetnevet kapja.



5. MÉLYEBB ISMERETEK A FLOPPY–KEZELŐ PROGRAMOKRÓL

5.1. A diszk–ROM szervezése

A diszk–ROM négy modulból áll. Közülük a legfőbb a fájlkezelő (FISH – Filing System Handler), amely a fájlhoz való hozzáférés, és a rajtuk való manipuláció valamennyi eszközét biztosítja. Ezt a modult külső programok és a diszk–ROM más moduljai is használják. Részletesen l. az 5.3. fejezetben.

A közvetlen hardver-kezelést a DISKIO modul végzi. A FISH is ezen keresztül használja a diszk-csatolót. A DISKIO meglehetősen alacsony szintű interfészt biztosít, amely az adott meghajtótól független, így egyetlen más program számára sem indokolt a csatolót közvetlenül kezelni. A DISKIO hozzáférhető VT–DOS-ra írt külső programok számára is pl. nem-szabványos diszkek kezelésére, vagy olyan tevékenységekhez, mint a lemezek formattálása. A DISKIO részletes ismertetése az 5.4. fejezetben található.

A harmadik modul egy „DISK DEVICE”-t valósít meg a TVC eredeti operációs rendszerében, amely kompatibilis a beépített kazetta-kezeléssel, és automatikusan felváltja azt a készülék bekapcsolásakor, ha a diszket csak a csatoló-kártyával, a BDOS-t tartalmazó opcionális programmodul nélkül használjuk. (Természetesen a kazettát továbbra is lehet használni.) A DISK DEVICE modult részletesen l. az 5.2. fejezetben.

A diszk–ROM negyedik modulja a TVC eredeti BASIC-jéből kiadható fájl- és könyvtár-kezelő parancsokat biztosít.

Elnevezése: BASIC CLI (Command Line Interpreter), részletesebben l. a 4. fejezetben.

5.2. A MOPS DISK DEVICE (MDD)

5.2.1. Bevezetés

Az 5.2. fejezet azt a diszk-kezelő I/O-készüléket írja le, amely a MOPS-ból, a TVC eredeti operációs rendszeréből az ottani konvenciók szerint használható (a kazetta-kezelőt emulálja).

Az MDD a diszk-ROM-ban helyezkedik el és a FISH-en keresztül használja a diszket.

A lehető legpontosabban emulálja a kazetta-kezelőt, így a létező kazettás programok változtatás nélkül használhatják. Bekapcsoláskor automatikusan a kazetta helyére lép. Biztosít egy egyszerű véletlen hozzáférést is a fájlukhoz azáltal, hogy a fájlmutató elmozdítható.

Az MDD-t ugyanúgy lehet elérni, mint a kazetta-kezelőt. Biztosítja a fájlmegnyitás, fájlírás/olvasás, fájlellenőrzés és fájllezárás funkciókat. Ezen kívül beolvasható és beállítható a fájlmutató, beolvasható a fájl mérete, parancs adható át a BASIC CLI-nek, valamint közvetlen FISH-hívások is végrehajthatók.

A MOPS funkciókódok:

| | |
|---|-----------|
| Karakter olvasás | 10000001B |
| Karakter írás | 00000001B |
| Blokk olvasás | 10000010B |
| Blokk írás | 00000010B |
| Fájlmegnyitás | 10000011B |
| Fájlletréhozás | 00000011B |
| Fájllezárás | X0000100B |
| Blokk ellenőrzés | 10000101B |
| Fájlmutató beállítása a fájl elejéhez képest | X0000110B |
| Fájlmutató beállítása az akt. állapothoz képest | X0000111B |
| Fájlmutató beállítása a fájl végéhez képest | X0001000B |
| CLI puffer definiálása | X0001001B |
| CLI parancs végrehajtása | X0001010B |
| FISH-funkció végrehajtása | X0001011B |

Az MDD-nek, a kazetta-készülékhez hasonlóan, egy olvasási és egy írási csatornája lehet megnyitva maximálisan egyidejűleg; viszont – a kazettától eltérően – ehelyett lehet egy megnyitott írási/olvasási csatornája. Amikor két megnyitott csatorna van, a fenti funkciókódokban az X az olvasási csatornára „1”, az írási csatornára „0” értékű.

A MOPS kazetta-készülék számos változót használ, amelyek legnagyobb részének nincs értelme diszkes környezetben. Az MDD a \$BUFFER változót nem használja, de létrehozza a kompatibilitás érdekében, mint a \$EOF változót is. A többi kazetta-változót (REMRD, \$PROTECT, \$MUDDLE), egyáltalán nem használja, s értéküket nem is változtatja meg.

5.2.2. Fájl létrehozása; fájl megnyitása

Paraméterek: DE = a meghajtó/útvonal/fájlnev stringre mutat

Kimeneti interfész: A = hibakód
DE = a fájlnévre mutat (az útvonalat nem adja vissza)

Legföljebb egy olvasási és egy írási csatorna lehet megnyitva egyidejűleg, vagy pedig egyetlen írási/olvasási csatorna.

Egy olvasási csatorna megnyitása úgy történik, hogy a fájlnevet átadjuk egy pufferben, amelyre a DE mutat (hasonlóan, mint a kazetta-kezelőnél); a funkcióhívás az „A” regiszterben hibakódot ad vissza. Ha a DE által megcímezett karakterláncban nincs fájlnev, akkor a „START” fájlnevet használja. Ha nincs megadva névkiegészítés, akkor a „.CAS” kiegészítést használja. Meghajtó és útvonal is lehet a karakterláncban.

Egy írási csatorna megnyitása úgy történik, hogy a fájlnevet átadjuk egy pufferben, amelyre a DE-mutat (hasonlóan, mint a kazetta-kezelőnél); a funkcióhívás az A regiszterben hibakódot ad vissza. Ha nincs megadva kiegészítés, akkor a „.CAS” kiegészítést használja. Az olvasási csatorna megnyitásától eltérően a fájlnev hiánya esetén hibajelzés történik. Ez megakadályozza a START.CAS fájl véletlen felülírását. Meghajtó és útvonal is lehet a karakterláncban.

Egy írási/olvasási csatorna megnyitásához csupán egy olvasási, vagy írási csatornát kell megnyitni. Ezután minden írás és olvasás erre az egy csatornára fog vonatkozni, mivel nincs másik nyitott csatorna.

Ha olvasásra megnyitunk egy csatornát olyankor, amikor már létezik megnyitott olvasási csatorna, akkor az eredetileg nyitott csatorna automatikusan lezáródik. Hasonlóképpen, ha nyitott írási csatorna mellett új írási csatornát nyitunk meg, az eredeti lezáródik.

A diszkes rendszerben nem létezik a „pufferelt” és „nem pufferelt” fájl megkülönböztetése, amit a kazettakezelő alkalmaz. A \$ BUFFER MOPS-változót figyelmen kívül hagyja az MDÜ írási csatorna megnyitásakor, és nem nulla értéket ír bele olvasási csatorna megnyitásakor; ez pufferelt fájlt jelent, s mivel ez az általánosabb, így lesz maximális a kompatibilitás a kazettás programokkal.

Mindkét csatornamegnyitási funkció esetén a fájlmutató a fájl kezdetére áll be. (Ez .CAS fájl esetén 128-at jelent, mert a .CAS fájlok mindig tartalmazzanak egy 128 bájtos fejblokkot az UPM-mel való kompatibilitás érdekében.

5.2.3. Fájl írása és olvasása

| | |
|---------------------|---|
| Paraméterek: | Karakter-olvasásnál: — |
| | Karakter-írásnál: C = karakter |
| | Blokk-olvasásnál: BC = a karakterek száma |
| | DE = a blokk elejére mutat |
| | Blokk-írásnál: BC = a karakterek száma |
| | DE = a blokk elejére mutat |
| Kimenetí interfész: | Karakter-olvasásnál: A = hibakód |
| | C = karakter |

Az írás ill. olvasás az aktuális fájlmutató által kijelölt karakterrel kezdődik a fájlban. A fájlmutató kezdetben a fájl elejére mutat, majd mindig aktualizálódik, s a fájl következő karakterére mutat. Nincs különbség pufferezt és nem-pufferezt fájlok között, és bármelyik fajta írás és olvasás végrehajtható bármelyik nyitott csatornán, a fájl bármelyik pontján.

Ha egy írási, vagy olvasási művelet után a fájlmutató a fájl végére, vagy azon túlra mutat (a fájlmutató nagyobb, vagy egyenlő, mint a fájl méret) a \$EOF MOPS-változót az MDD OFFH-ra állítja, így biztosítva a kompatibilitást a kazettás programokkal.

A karakterek írása és olvasása belső puffer közbeiktatásával történik. A blokk-olvasások és -írások hatékonyan, közvetlenül a memóriából történnek, így kb. 20 kbájt/sec fölötti sebességek érhetők el a szabványos floppy-diszkeken.

5.2.4. Fájl ellenőrzése

Paraméterek: BC = a karakterek száma
DE = a blokk kezdetére mutat

Kimeneti interfész: A = hibakód

A kazetta-kezelőhöz hasonlóan csak a blokk-ellenőrzést támogatjuk.

A karaktereket egyenként kiolvassa az írási/olvasási karakter-pufferből és összehasonlítja a memóriában levő karakterrel.

5.2.5. Fájlmutató beállítása

Paraméterek: BCDE = fájlmutató eltolás

Kimeneti interfész: A = hibakód
BCDE = aktuális fájlmutató

Bármelyik csatornán a fájlmutató a fájl tetszőleges pozíciójára beállítható; ezután az írási és olvasási műveletek ettől a pozíciótól kezdve hajtódnak végre.

A fájlmutató egy 32 bites mutató. Mindhárom fájlmutató-beállítási funkció esetén egy eltolási értéket kell átadni a BCDE regiszterekben történik. B a legnagyobb, az E pedig a legkisebb helyiértékű bájt. Bármilyen értékre beállítható 0 és 0FFFFFFFH között – még akkor is, ha a fájl végén túlra esik. A mutató új értékét visszaadja a BCDE regiszterekben.

Ha a „fájlmutató beállítása az aktuális állapothoz képest” funkciót 0 eltolással hívjuk meg, akkor egyszerűen megkapjuk a mutató aktuális értékét.

Ha a „fájlmutató beállítása a fájl végéhez képest” funkciót hívjuk meg 0 eltolással, akkor megkapjuk a fájl méretét.

Ha a fájlmutató nagyobb, vagy egyenlő, mint a fájl méret egy ilyen funkció végrehajtása után, akkor a SEOF MOPS-változó OFFH értéket vesz fel.

5.2.6. Fájl lezárása

Paraméterek: —
Kimeneti interfész: A = hibakód

Ez a funkció egyszerűen lezárja az írási, vagy olvasási csatornát, ha nyitva volt, és töröl minden diszk-puffert.

5.2.7. CLI puffer definiálása

Paraméterek: DE a puffer elejére mutat
BC a méret bájtokban

Kimeneti interfész: A = hibakód (rendszerint 0)

Számos CLI parancs működése közben memóriát használ, s ezzel a paranccsal lehet puffert definiálni számukra.

Az 5.2.8. pontban leírt parancshívás előtt kötelező a puffer definiálása, de több egymás utáni parancs között nem kell újra definiálni. A puffer a lehető legnagyobb legyen, különösen a COPY parancsnál. Ha a puffer átlépi a 2. lap tetejét, akkor hibajelzés történik.

5.2.8. CLI parancs végrehajtása

Paraméterek: DE egy parancssorra mutat
C = I/O választás

Kimeneti interfész: DE a puffer elejére mutat
BC = a pufferben levő karakterek száma
A = hibakód

E funkció hívása előtt az előző pontban leírt definiálást kötelező végrehajtani.

A DE által megjelölt parancssor egy karakterlánc, melynek az elején a hossz-bájt található, és a végrehajtandó parancsot tartalmazza.

Ha a hossz-bájt 0, akkor a CLI interaktív módban indul el, ilyenkor az ESC billentyűvel lehet megszakítani. Ha nem 0, akkor a parancs végrehajtódik (ugyanúgy, mintha interaktív módban begépeték volna).

A C regiszter 0. és 1. bitje vezérli az I/O művelet típusát. Csak akkor van hatásuk, ha nem interaktív módban indult el a CLI.

Ha a 0. bit értéke „1”, akkor az output nem a képernyőre, hanem a definiált pufferbe kerül.

Néhány parancs másképp működik a 0. bit „1” értéke esetén:

| | |
|--------|--|
| CLS | – nincs különbség |
| COPY | – nincs képernyő/puffer output |
| DATE | – nem kéri az új dátumot |
| DOS | – nem kérdezi meg, hogy tényleg belépjen-e |
| FORMAT | – nem kérdezi meg, hogy kezdheti-e a formálást |
| LDIR | – nem nyomtat |
| LTYPE | – nem nyomtat |
| TIME | – nem kéri az új időt |

Ha a C regiszter 1. bitje „1” értékű, akkor minden output a párhuzamos nyomtatóra megy a képernyő helyett. (Soros nyomtató esetén a MOPS-ban a nyomtatást át kell irányítani.)

5.2.9. FISH–funkció végrehajtása

Paraméterek: DE a FISH paraméter-blokkra (FPB) mutat
C a FISH funkcióhívási kód

Kimeneti interfész: A = hibakód
DE megmarad
az FPB aktualizálódik

Ez a funkció a FISH-funkciók végrehajtását teszi lehetővé MOPS környezetben.

Az FPB elemei a FISH-funkciók hívásánál a regiszterekben átadott értékeket tartalmazzák a következőképpen:

DE+0,1 : IX
DE+2,3 : HL
DE+4,5 : DE
DE+6,7 : BC

5.2.10. Hibák

Az MDD minden funkció esetén visszaad egy hibakódot az A regiszterben. A hibakódokat a függelékben közöljük.

Ha egy alacsony szintű diszk-hiba előfordul, akkor a normál VT-DOS „Abort/Retry” hibakezelés működik. Ez kiírja a hiba kódszámát, a meghajtó betűjelét és egy kérdést. Pl.:

Error 140 – Drive B:
Abort, Retry or Ignore (A/R/I)?

ABORT válasz esetén a diszk-művelet félbeszakad és a hibát az MDD átadja a hívó programnak, amely valószínűleg hibaüzenetet fog kiírni.

RETRY válasz esetén a műveletet az MDD egyszerűen megpróbálja megismételni, s a hibát ismét jelenti, ha még fennáll. Az operátor gyakran valamilyen beavatkozás után adja a RETRY választ, pl. lemezt tesz a meghajtóba.

Az IGNORE válasz csak végső megoldás lehet, amikor megkíséreljük helyreállítani az adatokat egy sérült lemezen; azt jelenti ugyanis, hogy az adatok egy része elvész. Ahol ez a lemez súlyos sérülését okozhatja (pl. könyvtár-olvasáskor) az IGNORE válaszlehetőség nem jelenik meg, de azért az MDD (ill. a FISH) általában elfogadja).

5.2.11. Kazetták használata

Amikor a diszkcsatoló kártya a normál inicializálás során inicializálódik, a standard kazettás I/O-funkciók átirányítódnak az MDD-re, így a kazettás MOPS programok mindig a diszket használják.

Ha mégis használni kívánjuk a kazettát, akkor a MOPS redirekciós (átírányítási) táblák visszaállíthatók úgy, hogy ismét a kazetta-kezelőre hivatkozzanak. Ezt a BASIC POKE utasításával lehet megtenni.

Az input és output átirányítási táblák közül bármelyiket, vagy mindkettőt is átírhatjuk. Ha pl. csak az input táblát változtatjuk meg, akkor a BASIC LOAD parancsra a kazettát, a SAVE parancsra pedig a diszket fogja használni.

Ha a kazetták használatát befejezve vissza akarunk térni a diszkhez, akkor a táblák megfelelő bejegyzését az eredeti értékre kell módosítani. Ezt az értéket nem lehet általános érvénnyel megadni, mert függ attól, hogy a diszkcsatoló kártyát melyik bővítő kártyahelyre csatlakoztatták.

Fontos tudni, hogy a kazetta-kezelés és a CLI funkciók ugyanezt a RAM-területet használják, ezért a kettő közötti váltásnál meleg reset végrehajtása ajánlatos.

5.3. A VT-DOS fájlkezelője (FISH)

5.3.1. Bevezetés

A diszk-ROM tartalmának legnagyobb terjedelmű része a fájlkezelő modul (Filing System Handler – FISH). Minden olyan program, amely fájlokat és könyvtárakat használ, a FISH-en keresztül éri el a diszket, mint pl. a MOPS DISK DEVICE, a VT-BDOS és a standard VT-DOS utility programok. Ez a fejezet leírja, hogyan működik a FISH, mit csinálnak a különböző funkcióhívások, és milyen paraméterek tartoznak hozzájuk.

A FISH a VT-DOS-ból a VT-DOS kézikönyv 4.2.6. fejezetében leírt módon hívható. A MOPS rendszerben jelen kézikönyv 5.2.9. fejezetének megfelelően használható.

5.3.2. Inicializálás

A FISH-t és a diszk-ROM többi részét a MOPS DISK DEVICE inicializáló belépési pontja inicializálja. A változók, stb. kezdeti beállítása után a MOPS visszakapja a vezérlést és folytatja a rendszer inicializálását. Ha a VT-DOS dugaszolható programmodul jelen van, akkor az inicializálás végeztével a MOPS átadja a vezérlést a VT-BDOS-nak, amely ezután a MOPS mellőzésével a saját I/O handlereit használja. Ha nincs jelen a VT-BDOS, akkor a MOPS normál módon viselkedik.

A hideg inicializálás során a diszk-ROM megvizsgálja az A: és B: meghajtókat, s megállapítja, hogy egy- vagy kétmeghajtós rendszerről van-e szó. Ha csak egy meghajtó van, akkor a B: automatikusan az A: meghajtóhoz rendelődik (v.ö. az operációs rendszer MAPDISK CLI-parancsával és a MAPDISK FISH-funkcióval).

MOPS-környezetben a FISH a MOPS igénye szerint meleg, vagy hideg inicializálást végez. A meleg inicializálás hasonló a hideg inicializáláshoz, kivéve, hogy a változókat nem inicializálja újra.

A VT–BDOS környezetben a MOPS nincs használatban, így a reset gomb megnyomása hideg indítást jelent a MOPS számára. A FISH figyelembe veszi ezt a körülményt, és meleg indítást végez, annak ellenére, hogy a MOPS és a VT–BDOS hidegen indul.

5.3.3. A diszk szervezése

A FISH által támogatott diszk-szervezés kompatibilis az MS–DOS minden verziójával, legalább a 2.00 verzióig és az MSX–DOS-sal, feltéve, hogy 512 bájtos szektorokkal dolgozik.

Minden lemeznek van egy boot-szektora (a 0. szektor), egy kiindulási könyvtára és egy, vagy több (általában 2) példánya a fájl allokációs (elhelyezési) táblából (FAT). A lemez fennmaradó része logikailag ún. alapegységekre oszlik, amelyek számozása $2 \dots M+1$, ahol M az összes alapegység száma.

Az alapegység a helyfoglalás legkisebb egysége a diszken; mérete függ a lemezformátumtól, rendszerint 1 vagy 2 szektor. Egy alapegység szektorai mindig szekvenciálisan érhetők el.

A kiindulási könyvtár fix számú (rendszerint 64, vagy 112, a lemezformátumtól függően) könyvtárbejegyzést tartalmaz, amelyek mindegyike 32 bájtos. Minden könyvtárbejegyzés egy fájlra, alkönyvtárra, vagy a kötetnévre vonatkozik. Egy alkönyvtár szintén 32 bájtos könyvtárbejegyzések sorozatából áll, de változó méretű; ha betelik, újabb lemezterületet foglal le. A könyvtárbejegyzések formátuma pontosan olyan, mint az MS–DOS 2.00-nál (l. 5.3.3.3.).

A fájlok és az alkönyvtárak is alapegységek láncolatából állnak. Az első alapegység számát a könyvtárbejegyzés tartalmazza, a további alapegységek pedig az alapegységsszámok láncolatának a FAT-ban való követésével érhetők el. A FAT azt is jelzi, hogy a lemezen mely alapegységek szabadok, ill. nem használhatók.

5.3.3.1. A boot-szektor

A boot-szektor mindig a lemez első szektora, amelyet a FORMAT és a DISKCOPY programok hoznak létre. Ez a formátum az MS-DOS-sal való kompatibilitást szolgálja. Bár boot-szektornak hívják, valójában nem tartalmaz boot-programot a VT-DOS számára, hiszen a teljes program ROM-ban van.

- 00H – 0EBH \ A VT-DOS-ban nem használt 8086-os ugrási utasítás az
- 01H – 0FEH – MS-DOS kompatibilitás érdekében.
- 02H – 090H /

- 03H ... 0AH – „VTDOS1.0” Rendszer-azonosító karakterlánc

- 0BH ... 1DH – UPB (Unit Parameter Block); egység-paraméter blokk, a korai MS-DOS lemezeken hiányozhat

- 1EH – Z80 „RET” utasítás az MSX-kompatibilitás érdekében. Az MSX egy boot-program belépési pontjaként használja, amely behozza az MSX-DOS-t, ha jelen van. A VT-DOS nem használja.

- 1FH ... 3FH – Nem használt. Ez a 33 bájt az MS-DOS, vagy MSX-DOS boot-szektorának esetleges későbbi kibővítésével való kompatibilitás érdekében van fenntartva.

- 40H ... 45H – „VOL-ID”. Karakterlánc annak jelzésére, hogy a kötetazonosító és a „piszkos diszk” jelző létezik.

- 46H – „Piszkos diszk” jelző (l. 5.3.3.5.)
0 = > „tisztá” a lemez
1 = > „piszkos” a lemez

47H...4AH — 4 bájtos egyedi kötetazonosító. Ez egy véletlen szám, amelyet a FORMAT és a DISKCOPY helyez el a lemezcserek ellenőrizhetősége érdekében (l. 5.3.3.4.). Mind a 4 bájtnak 0...7FH közé kell esnie.

4BH...1FFH — Nem használt.

5.3.3.2. Fájlelhelyezési tábla (File Allocation Table, FAT)

A FAT 12 bites bejegyzéseket tartalmaz, a lemez minden alapegységéhez egyet. Az első két bejegyzés a 0. és a 1. alapegységhez tartozik, amelyek nem léteznek, így az első alapegységhez (a 2. alapegységhez) tartozó FAT bejegyzés a FAT negyedik bájtnál kezdődik. A FAT első három bájtnál a következő:

0. bájtnál: FATID bájtnál, mindig 0F8H...0FFH

1. bájtnál: mindig 0FFH

2. bájtnál: mindig 0FFH

A FATID bájtnál a lemezformátum meghatározására szolgál, ha nincs UPB a boot-szektorban, egyébként nem használatos.

Minden alapegységhez tartozó bejegyzés nulla, ha az alapegység szabad; egyébként az alapegység része egy fájl-nak, és a bejegyzés értéke a fájl következő alapegységének számát adja meg. A fájl utolsó alapegységét a 0FF8H...0FFFH értékek valamelyike jelzi. Ezáltal a fájl-t alkotó alapegységek láncba fűzötten jelennek meg a FAT-ban.

A 0FF0...0FF7H értékek a fenntartott alapegységeket jelölik, ha azok nem alkotják egy lánc elemét sem; a 0FF7H jelzi a hibás alapegységeket, amelyeket nem szabad használni.

Egy lemez a FAT-ot akárhány példányban tartalmazhatja (rendszerint 2 példányban). A másodpéldányokat akkor használják, ha az első lemezhiba mi-

att olvashatatlaná válik. Rendszerint az összes példány egyforma, de az 5.3.3.5. pontban olvasható egy kivétel ez alól (a fájl visszaállítás lehetősége érdekében).

5.3.3.3. A könyvtárbejegyzések formátuma

Minden könyvtárbejegyzés, akár a kiindulási könyvtárban, akár egy alkönyvtárban 32 bájttal hosszú, formátuma pedig a következő:

00H...07H fájlnev. 8 karakteres, balra igazított, szóközzel kitöltve. Első bájta jelzi a bejegyzés állapotát:

00H: sohasem volt használatban

0E5H: törölve

05H: a fájlnev első karaktere 0E5H

Minden más karakter a fájlnev első karakterét jelenti. A két speciális név, a „.” és a „...” fenntartott és a saját könyvtárra, ill. a „szülő” könyvtárra mutatnak.

08H...0AH A fájlnev kiegészítése. 3 karakter, balra igazítva és szóközzel feltöltve.

0BH Attributum-bájt (l. 5.3.4.3.2.)

0CH...15H Fenntartva

16H – 17H Időpont \ A pontos formátum olyan, mint az

18H – 19H Dátum / FCB-ben (l. 5.3.4.3.2.)

1AH – 1BH A fájl első alapegysége (0002H...0FF7H)

1CH – 1FH A fájl mérete bájtokban, az első bájta a legkisebb helyiértékű

5.3.3.4. Kötetazonosító

A boot-szektor formátuma megengedi egy 4 bájtos egyedi kötetazonosító tárolását (l. 5.3.3.1.). Ez egy véletlen szám, amelyet a FORMAT és a DISKCOPY programok helyeznek oda, hogy a lemez megkülönböztethető legyen a többtől. Van egy „VOL-ID” karakterlánc is, amelynek vizsgálatából megállapítható, hogy létezik-e kötetazonosító. Ez biztosítja a kompatibilitást az MS-DOS és az MSX-DOS rendszerekkel.

Amikor a FISH beolvassa az egység paraméter blokkot (UPB), akkor beolvassa a kötetazonosítót is a lemezről (ha létezik), ezáltal ellenőrizheti, hogy a lemezt időközben nem cserélték-e ki. Ezzel elkerülhető az olvasás, ill. írás a téves lemezről, vagy lemezre, ha a felhasználó nyitott fájlok mellett lemezt cserélt.

5.3.3.5. „Piszkos diszk”-jelző, törölt fájlok visszaállítása

Általában az MS-DOS kompatibilis rendszerekben, ha egy fájlt töröltek, akkor a hozzá tartozó alapegységeket szabadnak nyilvánítják a FAT valamennyi példányában. Ez azt jelenti, hogy nincs mód a véletlenül törölt fájlok visszaállítására, mivel a lánc nem építhető fel újból. A VT-DOS azonban lehetővé teszi a fájlok visszaállítását mindaddig, amíg nem került sor helyfoglalással járó műveletekre (tehát nem hoztak létre fájlt és nem írtak adatot a diszkre), mivel a törlési funkció nem aktualizálja a FAT minden példányát. Ez a lehetőség csak azokon a lemezeken biztosított, amelyeken a boot-szektor tartalmazza a kötetazonosítót. Így az MS-DOS, vagy MSX-DOS alatt készített lemezeken a fájlokat nem lehet visszaállítani.

Amikor a FISH kitöröl egy fájlt egy érvényes kötetazonosítójú lemezről, a hozzá tartozó alapegységeket felszabadítja az utolsó kivételével valamennyi FAT-példányban, és a boot-szektorban levő „piszkos diszk” jelzőt 1-be állítja. Így az alapegység-lánc a FAT utolsó példányában még rendelkezésre áll, és egy külső program újra felépítheti az eredeti fájlt (vagy fájlokat) ebből az információból. Az UNDEL utility program szolgál erre a célra.

Ha bármilyen alapegység-foglalással járó műveletet hajt végre a FISH, megvizsgálja a „piszkos diszk” jelzőt. Ha értéke nulla, akkor a művelet normálisan folytatódik; ha azonban nem nulla, akkor a FAT utolsó példányát is aktualizálja, így azonosá teszi a többi példánnyal, mielőtt szabad alapegységet keresne. Erre azért van szükség, hogy az utolsó példány ne kitörölt fájlok és a volt területre került új fájlok keverékét tartalmazza.

5.3.4. FISH funkcióhívások

5.3.4.1. Általános FISH-hívási konvenciók

Minden írási és olvasási funkcióhívás átadja egy diszkátviteli terület címét. Ez a terület 64k méretű is lehet, tehát 4 lapot is elfoglalhat. Ha a diszkátviteli terület a 3. lapra is kiterjed, akkor normál esetben a 3. felhasználói RAM-ba íródik, ill. onnan olvasódik az adat. Egy FISH-változó beállításával elérhető, hogy a FISH a 3. felhasználói RAM helyett a diszk-csatoló RAM-ját használja, de erre a megoldásra a közönséges programoknak nincs szükségük – valójában csak a VT-BDOS céljait szolgálja. Ez a változó minden FISH-hívás után törlődik, így egyetlen programnak sem kell tőle félnie.

A FISH-nek átadott egyéb adatterületeknek, főként a File Control Block-oknak (FCB) és a pathname (útvonalnév) stringeknek rendszeresen a 0, 1 vagy 2. lapon kell lenniük. Átadhatók a diszkcsatoló RAM-jában is, de ezt a lehetőséget csak a VT-BDOS használja.

A funkció-kód átadására az A regiszter használatos; a többi CPU-regiszterbe funkciófüggő paraméterek kerülnek. Számos funkció esetében az IX a diszkátviteli címet tartalmazza, a DE pedig az FCB-re mutat. Egyes funkciók a HL-t és a BC-t is használják.

Egy FISH funkcióhívásból való visszatéréskor a memória lapozás állapota és az IY, AF', BC', DE', és HL' regiszterek tartalma megegyezik a hívás előttivel. A többi CPU regiszter (AF, BC, DE, HL, IX) tartalma megváltozik; egyes esetekben eredményt is adnak vissza. Minden funkcióhívás visszaad egy hibakódot az A regiszteren, amely sikeres lefutás esetén 0 lesz. A jelzőbitek beáll-

nak az A regiszter értékének megfelelően (visszatérés előtt egy OR A utasítás hajtódik végre).

A FISH funkciók általában nem törlik ki a használt adatterületeket. Ez azt jelenti, hogy fájlok törlése, átszervezése, stb. után egy „puffer-törlési” funkcióhívásra van szükség. A használt adatterületek egy bizonyos drive-ra kitörölődnek, ha egy használatlan FCB-t adunk át a FISH-nek egy „első keresése” (search for first) funkcióhíváskor és még néhány más funkcióhíváskor (pl. FORMAT). A VT-DOS parancsértelmező minden parancs végrehajtásának befejeztével kitöröl és érvénytelenít minden adatterületet.

Az alábbiakban közöljük a FISH funkcióhívások és funkciókódok teljes listáját:

| Kód | Funkció |
|-----|----------------------------------|
| 01H | Útvonal-string elemzése |
| 02H | Könyvtár-váltás |
| 03H | Aktuális könyvtár bekérése |
| 04H | Első keresése |
| 05H | Következő keresése |
| 06H | Fájl létrehozása |
| 07H | Könyvtár létrehozása |
| 08H | Törlés |
| 09H | Átnevezés |
| 0AH | Áthelyezés másik könyvtárba |
| 0BH | Attributumok változtatása |
| 0CH | Fájl megnyitás |
| 0DH | Fájl lezárás |
| 0EH | Pufferek törlése |
| 0FH | Olvasás fájlból |
| 10H | Írás fájlba |
| 11H | Írás 0-töltéssel |
| 12H | Abszolút szektor-olvasás |
| 13H | Abszolút szektor-írás |
| 14H | Login vektor beolvasása |
| 15H | Allokációs információ beolvasása |

| | |
|-----|--|
| 16H | Logikai meghajtó konvertálás fizikai meghajtóvá |
| 17H | Lemez formattálása |
| 18H | MAPDISK (fizikai meghajtók egymáshoz rendelése) |
| 19H | ASSIGN (logikai és fizikai meghajtók összerendelése) |
| 1AH | Rendszerváltozók beolvasása/beállítása |
| 1BH | Aktuális dátum beolvasása |
| 1CH | Aktuális dátum beállítása |
| 1DH | Aktuális idő beolvasása |
| 1EH | Aktuális idő beállítása |
| 1FH | A diszk-ROM verziószámának beolvasása |

5.3.4.2. Rendszerváltozók

A FISH a diszk-csatoló RAM-jában tartja az úgynevezett rendszerváltozókat. Ezek általában a felhasználó által módosítható változók, és a VT-DOS parancsértelmezője rendelkezik egy, a módosításukra szolgáló paranccsal. A rendszerváltozók a diszk-rendszer számos jellemzőjét vezérlik, gyakran egy-egy funkciót ki-, illetve bekapcsolnak. Olvashatók és írhatók a „rendszerváltozók beolvasása/beállítása” FISH-hívással. A rendszerváltozókat az I. függelékben soroljuk fel.

5.3.4.3. File Control Block-ok (FCB-k)

Minden fájlhozzáférést a FISH FCB-i vezérelnek. Az FCB-k hasonlóak, de nem azonosak az MS-DOS kiterjesztett FCB-ivel, amelyek pedig a CP/M FCB-k kiterjesztett változatai. Ha átadunk a FISH-nek egy FCB-t, annak mindig a 0, 1 vagy 2. lapon kell lennie. Ez azt jelenti, hogy a VT-DOS-nak minden UPM FCB-t át kell másolnia egy belső pufferbe (ha a felhasználói FCB a 3. lapon van), mielőtt átadná a FISH-nek. Ez lehetővé teszi azt is, hogy elvégezze a konverziót az UPM és a FISH formátum között.

5.3.4.3.1. FCB típusok

„Használatlan FCB”. Ebben csak a meghajtó, az attributumok, a fájlnev és a fájlnev-kiegészítés mezők érvényesek, az első bájtt pedig OFFH-ra van állítva. Ilyen FCB-t csak az „első keresése” a „fájl létrehozása” és a „könyvtár létrehozása” funkcióknak lehet átadni, és mindig meg kell adni egy útvonalnev-stringet, amely meghatározza a használható könyvtárat (az útvonalnev-stringek részleteit (l. az 5.3.4.3.3. pontban).

„Keresési FCB”. Egy sikeres „első keresése”, vagy „létrehozás” funkcióhívásokból való visszatéréskor az eredetileg használatlan FCB-ben új információt fogunk találni, amely meghatározza az aktuális helyet a könyvtár-keresésben. Ezután az FCB neve már „keresési FCB”, amelyet át lehet adni további „következő keresése” hívásoknak.

„Meghatározott FCB”. Ha egy keresési funkcióhívás talál egy megfelelő könyvtár-bejegyzést, akkor amellet, hogy visszaad egy módosított keresési FCB-t, létrehoz egy új úgynevezett „meghatározott FCB”-t is. Ennek az első bájttja OFEH, és tartalmazza a 32 bájtos könyvtár bejegyzést.

Minden olyan FISH-funkció számára, amely fájlnevre, vagy alkönyvtárra hivatkozik, át kell adni egy „meghatározott FCB”-t és a keresési funkció által visszaadott keresési FCB-t is. A keresési FCB megadja, hogy hol található a könyvtár-bejegyzés, a meghatározott FCB pedig az összes többi információt, amelyre a funkciónak szüksége van, vagy megváltoztatható a funkcióhívás által.

A következő parancsok igénylik mind a keresési, mind a meghatározott FCB-t:

| | |
|------------|---------------------------|
| törlés | attributumok változtatása |
| átnevezés | első keresése |
| áthelyezés | fájl megnyitás |
| létrehozás | |

Különösen fontos megjegyezni, hogy ha egy „első keresése”, vagy „létrehozás” funkcióhívásoknak átadunk egy meghatározott FCB-t, akkor az egy korábbi keresésben megtalált alkönyvtárban való keresést indít el. Valahányszor meghatározott FCB-t adnak át bármelyik fenti funkciónak, a könyvtár-bejegyzésben levő attributum bájtot a FISH ellenőrzi, hogy megfelelő-e az adott művelethez.

„Nyitott FCB”. Ha egy „fájl megnyitás” funkciónak átadunk egy meghatározott FCB-t, akkor az FCB számos más mezője kitöltődik és „nyitott FCB” keletkezik. A nyitott FCB-k az írási és olvasási, valamint a fájl lezárás funkciókban használhatók. A nyitott FCB-nek nincs speciális jelzése; a programnak tudnia kell, hogy megnyitotta-e már, vagy nem.

5.3.4.3.2. Az FCB-k formátuma

Minden, a FISH-nek átadott FCB hossza 48 bájttal. Az alábbi formátum nyitott FCB-re vonatkozik. A használatlan FCB-nek csak az első 19 bájttal van kitöltve; a meghatározott FCB-k első 19 bájttal az alábbiak szerinti, s ezt követi a könyvtár-bejegyzés maradék 21 bájttal (l. 5.3.3.3. pont). A keresési FCB-k első 13 bájttal megegyeznek a használatlan FCB-k megfelelő bájttalival, a többi pedig a FISH számára van fenntartva a könyvtár-információ eltárolásához.

| Hely az FCB-ben | Tartalom |
|-----------------|---|
| 00H | 0FFH – > használatlan FCB 0FEH – > meghatározott FCB |
| 01H ... 05H | kötelezően 0 (későbbi felhasználásra) |
| 06H | keresési attributum-bájt (l. 5.3.4.3.4.) |
| 07H | meghajtó-szám |
| | 0 – > default meghajtó |
| | 01H ... 1FH – > A: ... Z: meghajtó |
| | 20H ... FFH – > Nem megengedett |

| | |
|---------------|--|
| 08H . . . 0FH | balra igazított fájlnev, a végén blank-ek |
| 10H . . . 12H | fájlnev kiterjesztés |
| 13H – 14H | nem használt (blokkszám a VT–BDOS-ban, bővítési szám és S1 az UPM-ben) |
| 15H – 16H | nem használt (rekordméret a VT–BDOS-ban, S2 és rekordszám az UPM-ben) |
| 17H . . . 1AH | fájl méret bájtokban, az első helyen a legkisebb helyiértékű bájt |
| 1BH – 1CH | az utolsó módosítás dátuma 15. . . 9 bitek: év 0 . . . 119 (1980–2099) 8. . . 5 bitek: hónap 1 . . . 12 4. . . 0 bitek: nap 1 . . . 31 |
| 1DH – 1EH | az utolsó módosítás ideje 15. . . 11 bitek: óra 0 . . . 23 10. . . 5 bitek: perc 0 . . . 59 4. . . 0 bitek: másodperc 0 . . . 29 (2 sec-ként) |
| 1FH . . . 26H | fenntartva a FISH használatára |
| 27H | nem használt (aktuális rekord az UPM-ben) |
| 28H . . . 2BH | fájlmutató, az első bájt legkisebb helyiértékű. A megnyitás nullázza, minden írás és olvasás aktualizálja. (Az UPM-ben random record) |
| 2CH . . . 2FH | Kötet-azonosító (l. 5.3.4.3.5.) |

5.3.4.3.3. Útvonalnév-stringek

Használatlan FCB átadásakor útvonalnév-stringet is kell átadni a FISH számára. A karakterlánc egy 00 bájttal végződik és null-string is lehet.

Az útvonalnév-string meghatározza, hogy melyik meghajtót és melyik könyvtárat kell használni a művelethez. Null-string esetén az aktuális könyvtár (ha ilyen nincs, akkor az ún. kiindulási könyvtár) az érvényes az FCB által definiált meghajtón. Ha az FCB-ben a meghajtó-szám 0, akkor a DEF_UNIT rendszerváltozó által meghatározott meghajtót használja a FISH.

Ha a karakterlánc nem null-string, akkor tartalmaz egy opcionális meghajtó-azonosítót (pl. „A:”, „Z:”), amelyet alkönyvtár-nevek sorozata követ. A sorozat elemeit „\”, „!”, vagy „” választja el egymástól. Ha a karakterlánc formátuma érvénytelen, pl. hamis karakterek vannak a végén, vagy többértelmű könyvtár-neveket tartalmaz, akkor a FISH „.IPATH” hibajelzést ad vissza a hívónak.

Ha van meghajtó-azonosító, akkor a megfelelő meghajtó-szám bekerül az FCB-be, felülírva az eredetileg ottlevőt. Ha nincs megadva meghajtó-azonosító, akkor az FCB-ben levő meghajtó-számot használja a FISH. Ez többnyire 0 szokott lenni, tehát a DEF_UNIT-ban megadott meghajtót választja ki, és ez a meghajtó-szám bekerül az FCB-be a 0 helyére.

Ha a „:” utáni első karakter „\”, akkor az alkönyvtár lista a kiindulási könyvtárral kezdődik, egyébként pedig a meghajtó aktuális könyvtárával. Mindegyik alkönyvtár-nevet sorra megkeresi a FISH a diszken, egészen a karakterlánc végéig. Ha valamelyik nem létezik, akkor .NODIR hibát jelez a FISH. Az alkönyvtár nevek nem tartalmazhatnak többértelmű karaktereket („?” „*”).

A könyvtár-nevek pontosan olyanok, mint a fájlnevek, tehát egy 8 karakteres névből és egy opcionális 3 karakteres kiegészítésből állnak. A név és a kiegészítés bármilyen ASCII karaktert tartalmazhat, kivéve a definiált fájlnev végkaraktereket, valamint a „?” és „*” karaktereket.

Ha a név 8-nál több, vagy a kiegészítés 3-nál több karaktert tartalmaz, akkor a fölösleges karakterek figyelmen kívül maradnak – feltéve, hogy érvényes karakterek. A fájlnev végkarakterek jelzik az egyes fájlnev komponensek végét. A végkarakterek a következők lehetnek:

Minden vezérlő kód és a szóköz (00H . . . 20 H)

: ; . , = + # \ ! < > | / " ' []

Van egy speciális FISH funkcióhívás, amely egy parancssor-karakterláncból útvonalnév-stringet és opcionálisan fájlnevet készít. Ezt a funkciót használja a parancsértelmező, a MOPS DISK DEVICE és a BASIC CLI is. Ezt a funkcióhívást részletesen az 5.3.4.4. pontban ismertetjük.

Valahányszor útvonalnév-stringet adunk át a FISH-nek egy könyvtár meghatározásához, a felhasználó visszakap egy mutatót, amely az úgynevezett „teljes út string”-re mutat. Ez leírja a teljes utat a kiindulási könyvtártól az útvonalnév-string által definiált könyvtárig. Ezt a stringet a FISH a diszk-csatoló RAM-ban egy belső pufferben állítja össze, így az csak a következő, a puffert használó FISH hívásig érvényes.

A puffert használó FISH hívások a következők:

első keresése (csak, ha útvonalnév-átadás történt)

fájl létrehozása (" ")

könyvtár " (" ")

áthelyezés

könyvtárváltoztatás

aktuális könyvtár beolvasása

Ez a lehetőség főként a parancsértelmező DIR parancsának céljait szolgálja.

5.3.4.3.4. Attributum-bájt

A diszken minden könyvtár-bejegyzés tartalmaz egy „fájlattributum bájt”-ot, amelynek bizonyos bitjei jelzik a könyvtár-bejegyzés típusát és a fájl különböző jellemzőit.

Minden FCB tartalmaz egy „keresési attributum bájt”-ot, de ezt csak a keresési, létrehozási és útvonalnév-elemzési funkciók használják. Ennek a bájtnek csak a „kötetnév” bitje használatos. Hatásának részletei a különböző funkciók leírásában találhatóak.

A fájlattributum bájtok bitkiosztása a következő (a keresési attributum bájtban csak a „kötetnév” bit használt, amely ugyanazon a pozíción található, mint itt):

0. bit CSAK OLVASHATÓ. Ha ez a bit 1-be van állítva, akkor a fájlt nem lehet írni, vagy törölni, csak olvasni, átnevezni, vagy áthelyezni.

1. bit REJTETT FÁJL. Ha ez a bit „1” értékű, akkor ezt a fájlt csak úgy lehet megtalálni, ha a „rejtett fájl” bitet 1-be állítjuk a keresési attributum bájtban. A rejtett fájlokat minden olyan módon el lehet érni, ahogy a normál fájlokat. Alkönyvtárakat is el lehet rejtetni.

2. bit RENDSZER FÁJL. A FISH szempontjából ez a bit pontosan olyan funkciójú, mint a REJTETT fájl bit, kivéve, hogy a „fajllétrehozás” funkció nem törli automatikusan a rendszer fájlokat.

3. bit KÖTETNÉV. Ha a bit értéke „1”, akkor az adott bejegyzés a kötet nevét definiálja. Csak a kiindulási könyvtárban fordulhat elő és csak egyszer. Ez a jellemző nem változtatható meg a könyvtár-bejegyzésben, bár a kötetnév megváltoztatható a létrehozási, törlési és átnevezési funkciókkal.

4. bit ALKÖNYVTÁR. Ha ez a bit „1”-es, akkor ez a belépési pont egy alkönyvtár, nem pedig egy fájl, így nem lehet megnyitni olvasásra, vagy írásra. Ezt a bitet nem lehet megváltoztatni.
5. bit ARCHIVÁLÁSI BIT. Ez a bit 1-be állítódik, ha a fájlt írás után lezárták. Megvizsgálhatja egy ARCHIV program, amely megállapítja, hogy a fájl megváltozott és archiválni kell.
6. bit fenntartva (mindig 0)
7. bit fenntartva (mindig 0)

5.3.4.3.5. Kötetazonosítók az FCB-ben

Valahányszor egy használatlan FCB-t adunk át a FISH-nek keresési, vagy létrehozási funkcióhíváskor, az aktuális diszk kötetazonosítója átmásolódik az FCB-be. Ha ezt az FCB-t visszaadjuk a FISH-nek egy „következő keresése” funkcióhívással, a FISH ellenőrizni fogja a kötetazonosítót, hogy még mindig az a lemez van-e a meghajtóban.

Amikor egy keresési funkció talál egy könyvtár-bejegyzést, vagy amikor egy „létrehozás” funkció létrehoz egy újat, a „meghatározott FCB”-ben nem jön létre azonnal a kötetazonosító. Amikor a meghatározott FCB-t visszaadjuk a FISH-nek (pl. fájl megnyitáshoz), a keresési FCB-ben levő kötetazonosítót ellenőrzi a FISH, majd átmásolja a meghatározott FCB-be. Mindig ellenőrzi a kötetazonosítót, ha nyitott FCB-t használ fájlírásra, vagy olvasásra.

Megjegyzendő, hogy ez a kötetazonosító-ellenőrzés nem feltétlenül jár külön diszkhez fordulással a FISH által használt időzítések, stb. miatt. A felhasználó akár el is felejtheti a kötetazonosítókat az FCB-ben mindaddig, amíg biztos benne, hogy a hely mindig megvan számukra, és nem „rontja el” őket.

5.3.4.4. Útvonalnév-string elemzése – funkcióhívási kód: 01H

Paraméterek:

- DE → az inicializálandó FCB
- HL → a string első karaktere
- C = fájlnev flag
- 0 – > fájlnev másolása az FCB-be
- 1 – > fájlnev az útvonalban

Kimeneti interfész:

- A = állapotkód
- B = visszatérési jelzőbitek
- C = záró karakter
- DE → záró karakter
- HL = változatlan (útvonalnév-mutató)

Ez a funkció megkönnyíti a parancssorok elemzését, mivel közvetlenül a FISH-nek átadható útvonalnév-stringet és használatlan FCB-t készít. Tisztán karakterlánc manipulációs funkció, egyáltalán nem fordul a diszkhez. Fő működése az, hogy kialakít egy érvényes útvonal-stringet abból a karakterláncból, amelyre a HL mutat, úgy, hogy az átadható egy keresési, vagy létrehozási FISH funkcióhívásnak.

Ha az útvonalnév-string magában foglal egy fájlnevet, akkor azt át lehet másolni a DE által kijelölt FCB-be, amit aztán át lehet adni a FISH-nek, mint használatlan FCB-t.

A DE által kijelölt FCB-nek érvényes keresési attributumokkal és meghajtószámmal kell rendelkeznie; az FCB további részei kitöltetlenek is lehetnek. A keresési attributum-bájt kötetnév-bitje dönti el, hogy „meghajtó/útvonalnév/fájlnev”, vagy „meghajtó/kötetnév” karakterláncot kell-e elemezni.

Ha a kötetnévbit „0”, akkor a karakterlánc szintaxisa az 5.3.4.3.3. pontban leírtak szerinti: egy opcionális meghajtónév és az ezt követő könyvtárnév sorozat, melynek elemeit „\”, „!”, vagy „” karakterek választják el egymástól. Ha egy elem nem „\” karakterrel van lezárva, akkor ezt az utolsó elemzendő egységnek tekinti a FISH. Ezt az utolsó elemet vagy bemásolja az FCB-be, ha a C regiszter „0” volt, vagy belekerül az útvonal-stringbe, ha a C regiszter „1”

volt. A fájlnev, ha bemásolódik az FCB-be, végig nagybetűs lesz, szóközzel kiegészítve és minden „*” karakter a megfelelő számú „?”-lel kifejtve.

Ha a kötetnév-bit „1”, akkor a karakterlánc ugyanazt az opcionális meghajtónevet tartalmazza, de utána egy kötetnév következik, amely átmásolódik az FCB-be. A kötetnév bármilyen karaktert tartalmazhat az első „/” karakterig. Minden más fájlnev végkarakter (a szóköz is) megengedett a kötetnévben, és a kötetnév nem csak nagybetűket tartalmazhat, mint a fájlnev. A visszaadott útvonalnév-string csak a meghajtónevet foglalja magában.

Ha a karakterlánc meghajtónévvel kezdődik, akkor ez a meghajtószám beíródik az FCB megfelelő bájtvárára. Ha nincs meghajtónév a karakterláncban, akkor az FCB-ben levő meghajtószám nem változik, csak ha 0. Ez utóbbi esetben a DEF_UNIT rendszerváltozó értékét veszi fel.

Visszatéréskor a HL regiszterpár tartalma nem változik, s így az érvényes útvonalnév-string kezdőpontjára mutat, amely készen áll arra, hogy a FISH-nek átadják. A string magában foglalja a meghajtót (ha van) és a könyvtárelemeket, a korábban leírtak szerint. A string egy null-karakterrel végződik a megfelelő helyen, de ettől eltekintve változatlan marad.

A DE regiszterpár az eredeti karakterláncban az első nem feldolgozott karakterre fog mutatni, amely lehet a karakterlánc vége is, ha az egész string feldolgozásra került. A HL által kijelölt karakterlánc végére kerülő null-karakter felülírhatja a DE által megcímzett pozícióban levő eredeti karaktert, ezért ezt a C regiszterben, külön visszaadja a FISH. Ne tekintsük ezt a karaktert érvényesnek a stringben, de az útvonalnév-string használata után visszaírhatjuk.

A visszatérési jelzőbitek, amelyeket a B regiszterben kapunk vissza, bizonyos, potenciálisan hasznos tényeket közölnek az elemzési műveletről az alábbiak szerint:

- 0. bit „1”, ha a meghajtónéven kívül más kielemezett karakterek is vannak
- 1. bit „1”, ha volt specifikált könyvtár-útvonal
- 2. bit „1”, ha van specifikált meghajtónév
- 3. bit „1”, ha fájlnev van az utolsó elemben
- 4. bit „1”, ha fájlnev kiegészítés van az utolsó elemben

6. bit „1”, ha az utolsó elem többértelmű („?” vagy „*“)
6. bit „1”, ha az utolsó elem „.” vagy „.”
7. bit „1”, ha az utolsó elem „.” (csak akkor érvényes, ha a 6. bit „1”)

A 0. bit hasznos lehet olyan parancsokhoz, mint pl. a CD, amelyek különböző funkciójúak, attól függően, van-e útvonalnév a meghajtó után.

Az 1. bit főleg belső használatra szolgál, és a 2. bittel együtt hasznos lehet null-paraméter tesztelésére.

A 3. és 4. bit lehetővé teszi implicit fájlnevek használatát, és hasznosak lehetnek „\” karakterrel zárandó fájlok felismerésére (az utolsó elem teljesen üres, tehát a 3. és 4. bit is „0”).

Az 5. bit a többértelmű könyvtárnevek visszautasítására, a 6. és 7. bitek pedig a „.” és „.” speciális nevek visszautasítására használható, amelyek csak könyvtárnevek lehetnek (és nem törölhetők, nem átnevezhetők, nem áthelyezhetők) ám bár ezeket a FISH általában mindenképpen visszautasítja.

Ha a kötetnév-bit értéke „1”, akkor a visszatérési jelzőbitek közül a 0, 2 és 3. a fenti értelmezés szerint „1”, vagy „0”. Az 1. bit mindig „1” (bár természetesen ekkor nincs könyvtárútvonal megengedve), a 4, 5, 6 és 7. bitek értéke mindig „0”.

5.3.4.5. Könyvtár-változtatás – funkcióhívási kód: 02H

Paraméterek: HL = útvonalnév-string

Kimeneti interfész: A = állapotkód
HL = új aktuális könyvtárstring

Ez a funkció megváltoztatja az aktuális könyvtárt az adott meghajtón. Az útvonalnév-string megadja a meghajtót (ha nem, akkor az implicit meghajtót használjuk) és egy alkönyvtár listát. Ha „.” vagy „.” elemek találhatóak

benne, akkor ezeket a FISH funkció megfelelően kifejti. Ha az útvonalnév-string „\”, „!” vagy „' ” karakterrel kezdődik, akkor a régi aktuális útvonal figyelmen kívül marad, egyébként pedig az új útvonalnév hozzáadódik az aktuális útvonalhoz. Ha az útvonalon bármelyik könyvtár nem létezik, akkor a FISH „NODIR” hibát jelez, és az aktuális útvonal nem változik.

Az új aktuális könyvtárstring azonos azzal a karakterlánccal, amelyet egy „aktuális könyvtár beolvasása” funkcióhívás adna vissza és az ezen funkció leírásában olvasható megjegyzések ide is vonatkoznak.

5.3.4.6. Aktuális könyvtár beolvasása – funkcióhívási kód: 03H

Paraméterek: B = meghajtó-szám (1 . . . 26)

Kimeneti interfész: A = állapotkód
 HL = aktuális útvonal-string

Ez a funkció visszaad egy pointert, amely az aktuális útvonalnév-stringre mutat az adott meghajtón. A karakterláncnak sem az elején, sem a végén nincsenek „\” karakterek, és 00-val végződik. Ha az aktuális könyvtár azonos a kiindulási könyvtárral, akkor a visszaadott karakterlánc null-string.

A FISH belső „aktuális útvonal-string” példányát átmásolja egy külön pufferbe, s ennek a címét adja át a felhasználónak. Ez lehetővé teszi a felhasználónak, hogy módosítsa a stringet az aktuális könyvtár-útvonal zavarása nélkül. Ez a puffer azonos azzal, amelyben a „teljes útvonal-string” visszaadása történik, így a karakterlánc felülíródhat, ha ezt a puffert használó bármelyik funkciót meghívjuk (l. 5.3.4.3.3.).

Megjegyezzük, hogy a karakterláncot tartalmazó puffer a diszk-csatoló RAM-jában van, s a HL regiszterpár ide mutat.

5.3.4.7. Első keresése – funkcióhívási kód: 04H

Paraméterek:

- BC – > hely az FCB számára (48 bájtt)
- DE – > használatlan (vagy meghatározott) FCB
- HL – > útvonalnév-string (vagy keresési) FCB

Kimeneti interfész:

- A = állapotkód
- BC – > ua. a cím meghatározott FCB-vel
- DE – > ua. az FCB (most keresési FCB)
- HL – > teljes útvonal-string

Az adott könyvtárban olyan bejegyzést keres, amelyben a név megegyezik a DE által kijelölt FCB-ben levővel. az FCB-ben levő fájlnev többértelmű is lehet (tartalmazhat „?” karaktereket), mely esetben az első egyező nevet tartalmazó bejegyzést találja meg. Ha nem talál egyező nevet, akkor „.NOFIL” hibával tér vissza.

Az útvonalnév-stringek és az FCB-k együttműködésének magyarázatát l. 5.3.4.3.1. és 5.3.4.3.3. pontokban. Megjegyzendő, hogy a HL-ben kijelölt „teljes útvonal-string” nem érvényes, ha indulásnál a DE „meghatározott FCB”-re mutat.

A BC és DE regiszterpárok tartalma változatlan marad, ha nincs hiba, sőt bizonyos hibák esetén is. Különösen érdemes megjegyezni, hogy „.NOFIL” hiba esetén e regiszterek tartalma megmarad.

Ha a keresési FCB-ben az attributum-bájt „kötetnév” bitje „1”, akkor csak a kötetnév-bejegyzést adja vissza a FISH-funkció és ezt mindig visszaadja (ha van), tekintet nélkül az FCB-ben levő útvonalnévre, vagy fájlnévre. Ha a „kötetnév” bit „0”, akkor a kötetcímke-bejegyzést sohasem adja vissza, és másféle egyezéseket figyel. Ebben az esetben a hívónak kell vizsgálnia a „rendszerfájl”, a „rejtett fájl” és „alkönyvtár” biteket az attributum-bájttban, ha korlátozni akarja a keresést. Az attributum-bájtok részletezése az 5.3.4.3.4. pontban található.

Ha a DE meghatározott FCB-re mutat, akkor annak egy meghatározott alkönyvtárnak kell lennie (a könyvtár-bejegyzésnél az attributum-bájt ellenőrzésre kerül), a meghatározott FCB név-mezőjét pedig előzetesen felül kellett írni az ebben az alkönyvtárban keresendő új fájl névvel (rendszerint csupa „?” karakter).

Ha a FISH funkció talált egy megfelelő könyvtári bejegyzést, akkor egy új meghatározott FCB-t hoz létre a BC által kijelölt címen. Ez egy OFEH bájtal fog kezdődni (jelelvén, hogy meghatározott FCB-ről van szó), amely után következő 7 bájt a keresési FCB-ből másolódik át (ebben benne van a keresési attributum-bájt és a meghajtó), majd a könyvtár-bejegyzés 32 bájtja. Az új FCB utolsó 8 bájtja nem töltődik ki. Jegyezzük meg, hogy az új FCB 6. bájtja (az attributum-bájt) a keresési FCB megfelelő bájtjával lesz azonos, nem pedig a meghatározott könyvtár-bejegyzés attributumával, amely a fájl név utáni bájton lesz.

A meghatározott FCB átadható bármelyik olyan FISH funkciónak, amely meghatározott FCB-t igényel (felsorolásukat l. az 5.3.4.3.1. pontban). E funkciók jelentős része ellenőrzi a könyvtárbejegyzés jellemzőit az illegális műveletek megelőzése céljából. Például a csak olvasható fájlok nem törölhetők és alkönyvtárak nem nyithatók meg.

5.3.4.8. Következő keresése – funkcióhívási kód: 05H

Paraméterek: BC → hely az FCB számára (48 bájt)
DE → keresési FCB („első keresése” hívásból)

Kimeneti interfész: A = állapotkód
BC → ua. a cím meghatározott FCB-vel
DE → ua. az FCB (aktualizálva)

Egy „első keresése” funkcióhívás (l. 5.3.4.7.) után ezt a funkciót ismétlődően sokszor meg lehet hívni a további egyező nevek meghatározása céljából, ha a név többértelmű volt. A BC-ben megadott cím különbözhet az előző keresésben használttól, ha úgy célszerű. A BC és DE regiszterpárok tartalma megmarad azokban az esetekben, mint az „első keresése” funkciónál (5.3.4.7.).

A keresési FCB fenntartott része tartalmazza azokat az információkat, amelyekre a FISH-nek a keresés folytatásához szüksége van, így nem szabad módosítani az „első keresése” és a jelen funkció bármely hívása között. Szükség esetén elmentés céljából át lehet másolni, de ez általában elkerülhető.

5.3.4.9. Fájl létrehozása – funkcióhívási kód: 06H

Paraméterek:

- BC – > hely az FCB számára (48 bájtt)
- DE – > használatlan (vagy meghatározott) FCB
- HL – > útvonalnév-string (vagy keresési FCB)

Kimeneti interfész:

- A = állapotkód
- BC – > ua. a cím meghatározott FCB-vel
- DE – > ua. az FCB (aktualizálva)
- HL – > teljes útvonal-string

Ez a funkció nagyon hasonlít az „első keresése” funkcióhoz, kivéve, hogy többértelmű fájlnev nincs megengedve. Ha egyező névvel könyvtárbejegyzést talál, akkor azt – ha lehetséges – kitörli és a hozzá allokált területet felszabadítja. Ha a bejegyzés nem törölhető, akkor a következők szerint egy megfelelő hibajelzéssel tér vissza.

„.DIRX” – A bejegyzés egy könyvtár. A könyvtárakat explicit módon kell törölni.

„.SYSX” – A bejegyzés egy rendszer-fájl. Ezeket explicit módon kell törölni.

„.FILRO” – A bejegyzés csak olvasható fájl. Ezeket nem lehet törölni, csak ha előzetesen megváltoztatjuk a jellemzőjüket.

Ha nem talál egyező névvel bejegyzést, akkor a könyvtárban az első üres bejegyzést veszi használatba. Ha a könyvtár tele van, akkor kibővíti egy alapegységgel (kivéve, ha a kiindulási könyvtárban vagyunk, amikor „.DRFUL” hibát ad vissza, vagy ha nincs szabad alapegység a lemezen, amikor „.DKFUL” hibát jelez).

Ha a „fájl létrehozás” funkció talál egy megfelelő könyvtár-bejegyzést, akkor beleírja a fájlnevet az FCB-ből („.IFNM” hibával tér vissza, ha a fájlnev illegális), kitölti az attributum bájtot úgy, hogy csak az „archiválás” bit „1” beírja a „0” fájl méretet, valamint az aktuális dátumot és időpontot, amelyet a FISH rendszer-órája tart nyilván.

A BC-ben megadott címen egy erre az új bejegyzésre vonatkozó meghatározott FCB jön létre – pontosan olyan, amelyet az „első keresése” funkció ad vissza. Az új fájl nem nyitódik meg automatikusan; egy külön „fájl megnyitás” funkcióhívásra van szükség, hogy írni, vagy olvasni lehessen. Ha azt akarjuk, hogy a fájl rendszer-fájl, vagy rejtett, vagy csak olvasható legyen, ezt a létrehozás után egy „fájl attributumok változtatása” funkcióhívással érhetjük el.

Ha a használatlan FCB-ben a keresési attributum-bájt „kötetnév” bitje „1”, akkor ez a funkció a lemez kötetnevét fogja létrehozni, vagy megváltoztatni. Ez esetben a kötetnevet keresi, s ha megtalálja kicseréli az FCB-ben levő névre. Ha nincs kötetnév-bejegyzés, akkor a kiindulási könyvtár első üres bejegyzésén létrehozza a kötetnevet.

A funkcióból való visszatéréskor a DE által kijelölt FCB keresési FCB lesz. Ez a BC-ben megcímzett meghatározott FCB-vel együtt ugyanúgy használható, mintha az „első keresése” funkció adta volna vissza.

Ez gyakran hasznos, ha megkíséreltünk egy létrehozást, és „.DIRX” hibajelzést kaptunk, tehát a név egy könyvtár neve volt. A program tovább mehet, és létrehozhatja a fájlt ebben a könyvtárban, A BC és DE regiszterpárok tartalma megmarad az „első keresése” funkciónál ismertetett feltételek mellett (l. 5.3.4.7.) és akkor is, ha „.SYSX”, „.DIRX”, „.DRFUL”, vagy „.DKFUL” hiba fordul elő.

5.3.4.10. Könyvtár létrehozása – funkcióhívási kód: 07H

Paraméterek: BC – > hely az FCB számára (48 bájt)
 DE – > használatlan (vagy meghatározott) FCB
 HL – > útvonalnév-string (vagy keresési) FCB

Kimeneti interfész: A = állapotkód
BC → ua. a cím meghatározott FCB-vel
DE → ua. az FCB (aktualizálva)
HL → teljes útvonal-string

Ez a funkció nagyon hasonló a „fájlétrehozása” funkcióhoz (l. 5.3.4.9.) kivéve, hogy nem fájl számára készít könyvtár-bejegyzést, hanem egy alkönyvtár számára. Megvizsgálja, hogy van-e az adott névvel könyvtár-bejegyzés, de ha talál egy ilyen fájlt, nem törli ki, még ha törölhető is lenne, hanem „FILEX” hibajelzéssel visszatér.

Az új alkönyvtár egy alapegységnyi diszk-területet foglal el és üres lesz — tekintve a „.” és „. .” jelzésű két bejegyzéstől, amelyek a könyvtárra magára ill. a „szülő” könyvtárra mutatnak. E két alkönyvtár jellemzője „rejtett” lesz, így normál könyvtár-listázáskor nem jelennek meg.

5.3.4.11. Törlés — funkcióhívási kód: 08H

Paraméterek: DE → meghatározott FCB
HL → keresési FCB

Kimeneti interfész: A = állapotkód

A meghatározott FCB-hez tartozó könyvtár-bejegyzés törlődik, kivéve ha csak olvasható fájl („FILOR” hibajelzés), vagy nem üres alkönyvtár („DIRNE” hibajelzés). A „.” és „. .” alkönyvtárakat nem lehet törölni („DOT” hibajelzés). Ha nincs hiba, akkor minden a fájlhoz vagy a könyvtárhoz rendelt lemezterület felszabadul. Ha a lemeznek van érvényes kötetazonosítója, akkor a „plázkos diszk” jelző „1”-be áll, és a FAT utolsó példányában az alapegységek nem szabadulnak fel. Ez lehetővé teszi a fájlok visszaállítását az UNDEL program segítségével (a részleteket l. 5.3.3.5. pontban).

5.3.4.12. Átnevezés – funkcióhívási kód: 09H

Paraméterek: DE – > meghatározott FCB
HL – > keresési FCB
BC – > használatlan FCB, az új névvel

Kimeneti interfész: A = állapotkód

A meghatározott könyvtár-bejegyzés neve megváltozik a BC által megcímzett használatlan FCB-ben levő névre. Ez az új név többértelmű is lehet, ahol „?” fordul elő benne, azon a pozícion megmarad az eredeti karakter. Visszatérés-kor a meghatározott FCB-ben az új név szerepel, a „?”-ek megfelelően helyettesítve.

Ellenőrzés történik, hogy ne keletkezessen duplikált, vagy érvénytelen fájlnev; ha a funkcióhívás végrehajtása erre vezetne, akkor „.DUPF”, ill. „.IFNM” hibajelzést ad vissza. A „.” és „. .” alkönyvtárak nem nevezhetők át, ennek megkísérlése „.DOT” hibajelzéshez vezet. A használatlan FCB-ben, a meghatározott és a keresési FCB-ben a meghajtónak azonosnak kell lennie; ha nem így van, „IDRV” hibajelzés történik.

Ha olyan alkönyvtár nevét változtatjuk meg, amely az aktuális könyvtár-útvonalnak eleme, akkor az aktuális könyvtár-útvonal a következő hozzáférés-kor törlődik, mivel a FISH nem tudja relokálni a könyvtár-útvonalat.

5.3.4.13. Áthelyezés – funkcióhívási kód: 0AH

Paraméterek: DE – > meghatározott FCB
HL – > keresési FCB
BC – > új útvonalnév-string

Kimeneti interfész: A = állapotkód

A meghatározott könyvtárbejegyzést kiveszi abból a könyvtárból, ahol jelenleg van, és áthelyezi abba a könyvtárba, amelyet a BC-ben megcímzett útvonalnév-string specifikál. „.IATTR” hibajelzés történik, ha a bejegyzés kötet-

név. A funkció ellenőrzést végez, hogy ne keletkezzék duplikált fájlnev (,,.DUPF" hibajelzés). Ha az új útvonalnévben meghajtó is szerepel, akkor ez meg kell, hogy egyezzen azzal a meghajtóval, ahol a könyvtár-bejegyzés található, különben „.IDVR" hibajelzés történik.

Ha alkönyvtár áthelyezésről van szó, akkor további ellenőrzéseket végez a funkció, hogy nem a „.", vagy „. ." bejegyzéseket akarjuk áthelyezni („.DOT" hibajelzés) és, hogy a könyvtárt nem valamelyik saját „leszármazottjába" akarjuk áthelyezni („.DIRE" hibajelzés), mivel ez hurkot hozna létre. Egy alkönyvtár sikeres áthelyezése után a „. ." bejegyzés aktualizálódik, az új „szülő" könyvtárra fog mutatni.

5.3.4.14. Attributumok változtatása – funkcióhívási kód: OBN

Paraméterek: DE → meghatározott FCB
HL → keresési FCB

Kimeneti interfész: A = állapotkód

Ez a funkció kicseréli a meghatározott könyvtár-bejegyzés attributum bájttját, továbbá a dátum- és időadatát a meghatározott FCB-ben szereplő adatokra. A könyvtár-bejegyzés többi mezőjét változatlanul hagyja. Az attributum bájtot ellenőrzi, hogy a változtatás megengedett-e, és ha nem, akkor sem a dátumot és az időpontot, sem a jellemzőket nem változtatja meg, hanem „.IATTR" hibajelzéssel tér vissza. Az új dátumot és időpontot nem ellenőrzi.

Ha a könyvtár-bejegyzés kötetnév, akkor egyik attributum bit sem változtatható meg, ha pedig alkönyvtár, akkor csak a „rejtett fájl" bit változhat.

Ha a bejegyzés fájlhoz tartozik, akkor csak a „rejtett fájl", „rendszer-fájl", „csak olvasható" és az „archiválás" bitek változtathatók. A bitkiosztást l. az 5.3.4.3.4. pontban.

Jegyezzük meg, hogy az új jellemzők a meghatározott FCB könyvtárbejegyzés példányában vannak definiálva, és nem a keresési attributum bájttban. Az új dátumnak és időpontnak a meghatározott FCB-ben a normál pozíciójukban kell lenniük, ami nem azonos azzal a pozícióval, amelyet megnyitás után foglálnak el az FCB-ben (v.ö. 5.3.3.3.).

5.3.4.15. Fájl megnyitás – funkcióhívási kód: OCH

Paraméterek: DE → meghatározott FCB
HL → keresési FCB

Kimeneti interfész: A = állapotkód
DE → ua. az FCB (megnyitva, ha nincs hiba)

A meghatározott könyvtárbejegyzéshez tartozó fájlt megnyitja, így az készen áll írásra, vagy olvasásra. A bejegyzés nem lehet alkönyvtár, vagy kötetnév („.IATTR” hibajelzés), és ha „csak olvasható”, akkor írni nem lehet bele.

A fájl mérete, a dátum, az időpont és a meghajtó aktuális kötetazonosítója bemásolódik az FCB-be, a fájlmutató nullázódik. Az FCB azonnal használható a fájl elejétől induló szekvenciális írásra, vagy olvasásra.

Megjegyezzük, hogy a fájl méret, a dátum és az időpont a megnyitott FCB-ben nem ugyanazon a helyen van, s nem is ugyanabban a sorrendben, mint a meghatározott FCB-ven volt a megnyitás előtt. A megnyitott FCB felépítéséről az 5.3.4.3.2. pont közöl részleteket.

5.3.4.16. Fájl lezárás – funkcióhívási kód: ODH

Paraméterek: DE → meghatározott FCB

Kimeneti interfész: A = állapotkód
DE → ua. az FCB

Lezárja az FCB-hez tartozó fájlt. Azokat a fájlokat, amelyeket csak olvastunk, nem szükséges lezárni; ha mégis lezárjuk, a diszk-könyvtár nem változik. Ha írtunk a fájlba, akkor a könyvtárbejegyzést megfelelően aktualizálja az új dátum-, idő-, méret- és helyfoglalási információkkal és az archíválási bitet „1”-be állítja.

Jegyezzük meg, hogy ha az FCB-ben a fájl név megváltozik, akkor a lezárás nem történik meg helyesen. Az attributum-bájt sem változhat meg a

könyvtárban; ezt csak az „attributumok változtatása” funkcióhívás teheti meg (l. 5.3.4.14.).

Az FCB a fájl lezárása után is használható a fájlhoz való hozzáférésre, így a fájllezárás funkció arra is alkalmas, hogy hosszabb fájlírás közben időnként a fájl változásait teljes egészében rögzítsük a lemezen biztonsági okokból. A fájl-mutató nem változik.

Egy FCB-t csak akkor szabad lezárni, ha előzőleg sikeresen megnyitottuk. A FISH-nek nincs módja ezt ellenőrizni; ha megkíséreljük, a hatása nem definiált és katasztrófális lehet.

5.3.4.17. Pufferek törlése – funkcióhívási kód: OEH

Paraméterek:

B = meghajtó száma 0 => implicit
meghajtó, OFFH => mindegyik

C = 00H => csak puffertörlesztés
OFFH => pufferek törlése és érvénytelenítése

Kimeneti interfész: A = állapotkód

Ez a funkció kitörli az összes „piszkos” szektorpuffert, adatterületet az adott meghajtóra (vagy mindegyikre, ha B=OFFH). A nyitott fájlok nem záródnak le (mivel a FISH nem tartja nyilván őket). Mindig 0 állapotkódot ad vissza, kivéve, ha a meghajtószám érvénytelen, vagy megnyomták a CTRL-ESC billen-

Ha a C regiszter 0, akkor a piszkos pufferek törlődnek, de bennük maradó adatok nem kapnak „érvénytelen” jelzést. Ha a C regiszter tartalma OFFH, akkor a pufferek törlése mellett az adatok is érvénytelenek lesznek, így többé nem használhatók fel. Ebben az esetben is mindegyik meghajtó olyan jelzést kap, hogy a benne levő lemezt ellenőrizni kell.

Ezt a lehetőséget olyan külső programok alkalmazzák, mint a FORMAT, vagy a DISKCOPY, amelyek közvetlenül „összemaszatozzák” a diszket a DISKIO használatával, így biztosítva, hogy a FISH felismerje a módosított lemezt.

5.3.4.18. Olvasás fájlból – funkcióhívási kód: 0FH

Paraméterek: IX => diszk átviteli cím
DE => megnyitott FCB
BC => olvasandó bájtok száma

Kimeneti interfész: A = állapotkód
IX => aktuálizált memóriacím
DE => ua. az FCB
BC => olvasott bájtok száma

Beolvassa a megadott számú bájtot a fájlból a „diszk átviteli cím”-mel kezdődő memóriaterületre. Maximum 64k olvasható; hibajelzés („OV64K”) történik, ha az IX és a BC regiszterpárok tartalmának összege túlszordul a memórialap tetején. Az FCB-t előzőleg meg kell nyitni.

A fájlon belüli pozíciót az FCB fájlmutató mezője határozza meg, amely minden olvasás után aktualizálódik, s így a következő bájtra mutat. Ha a funkció hibát jelez, a sikeresen olvasott bájtok száma kisebb lehet, mint a két bájtszám. Ez a visszaadott szám minden hibajelzés esetén helyes, kivéve a „STOP” és „ABORT” hibákat. Ha a fájl végén túl próbálunk meg olvasni, „EOF” hibajelzést kapunk.

5.3.4.19. Írás fájlba – funkcióhívási kód: 10H

Paraméterek: IX => diszk átviteli cím
DE => megnyitott FCB
BC => írandó bájtok száma

Kimeneti interfész: A = állapotkód
IX => aktuálizált memóriacím
DE => ua. az FCB
BC => írt bájtok száma

A diszk átviteli területről a megadott számú bájtot beírja a fájlba. Maximum 64k írható, s „OV64K” hibajelzést ad, ha az IX és a BC regiszterpárok tartal-

amának összege túlcserül a 3. memórialap tetején. Ha a VERIFY rendszerváltozó értéke „0”, akkor a fölirt adatokat automatikusan ellenőrzi. Az FCB-t előzőleg meg kell nyitni.

A fájl belüli pozíciót az FCB fájlmutató mezője határozza meg, amely minden írás után aktualizálódik, s így a következő bájtra mutat. Ha hibajelzéssel tér vissza a funkció, akkor a sikeresen fölirt bájtok száma kisebb lehet, mint az írni kívánt bájtszám. A visszaadott szám minden hibakód esetén érvényes, kivéve a „STOP” és az „ABORT” hibajelzéseket.

Ha az írás túlhalad a fájl végén, akkor a fájl számára szükség szerint további lemezterületeket foglal le, a fájl méretet pedig aktualizálja. Ha az írás a fájl végén túl kezdődik, akkor a „hézag” betöltésére a lemezterületet lefoglalja, de nem inicializálja.

Ha nincs elég hely a lemezen a teljes írási művelet számára, akkor a „DKFUL” hibajelzéssel tér vissza és egyáltalán nem ír fel adatot, még akkor sem, ha az adatok egy része számára volna hely.

Előzetes problémák miatt a 3. felhasználói memórialapról olvasott adatok a lemeze írás előtt pufferbe kerülnek, így ebben az esetben az írási funkció gyorsabban működik a szokottnál.

3.4.20. Írás 0-töltéssel – funkcióhívási kód: 11H

Paraméterek:
IX => diszk átviteli cím
DE => megnyitott FCB
BC => írandó bájtok száma

Információs interfész:
A = állapotkód
IX => aktualizált memórialcím
DE => ua. az FCB
BC => írt bájtok száma

Az a funkció azonos a fent ismertetett „Írás fájlba” funkcióval. Az egyetlen különbség, hogy ha az írás a fájl végén túl kezdődik, akkor a „hézag” kitöltésére lefoglalt lemezterületet 0-val írja végig.

5.3.4.21. Abszolút szektor olvasás – funkcióhívási kód: 12H

Paraméterek: IX => diszk átviteli cím
DE = logikai sektorszám
H = olvasandó szektorok száma
L = meghajtó száma (1 . . . 26)

Kimeneti interfész: A = állapotkód
IX => aktualizált memóriacím
DE = következő logikai sektorszám
H = olvasott szektorok száma
L = u.a. a meghajtó-szám

A megadott szektorokat beolvassa a diszkről az IX-ben adott címtől kezdődő memóriaterületre. Max. 64k olvasható, s „OV64K” hibajelzést ad, ha az IX tartalmának és a beolvasandó szektorok összesített bájt számának összege túlszordul a 3. memórialap tetején.

Hiba előfordulásakor a beolvasott szektorok száma kisebb lehet, mint a kért szektorok száma. A visszaadott szám minden hibakód esetén érvényes, kivéve a „STOP” és „ABORT” hibajelzéseket. Hiba esetén normál ismételt próbálkozások történnek, s a diszk-hibák keresztülmennek a normál Abort/Retry/Ignore hibakezelő szubrutinon (l. 3.6. fejezet).

5.3.4.22. Abszolút szektor írás – funkcióhívási kód: 13H

Paraméterek: IX => diszk átviteli cím
DE = logikai sektorszám
H = írandó szektorok száma
L = meghajtó száma (1 . . . 26)

Kimeneti interfész: A = állapotkód
IX => aktualizált memóriacím
DE = következő logikai sektorszám
H = írt szektorok száma
L = u.a. a meghajtó-szám

A megadott szektorokra fölírja a diszk átviteli területen levő adatokat. Max 64k írható, s „OV64K” hibajelzést ad, ha az IX tartalmának és a fölírando szektorok összesített bájt számának összege túlcso rdul a 3. memó rialap tetején.

Hiba előfordulásakor a felírt szektorok száma kisebb lehet, mint a felírni kívánt szektorszám. A visszaadott szám minden hibakód esetén érvényes, kivéve a „STOP” és „ABORT” hibajelzéseket. Ha a „VERIFY” rendszerváltozó értéke 0, akkor automatikusan szektor-ellenő rzés történik.

5.3.4.23. Login vektor beolvasása — funkció hívási kód: 14H

Paraméterek: —

Kimeneti interfész: A = állapotkód
DE = A login vektor nagyobb helyiértékű része
HL = A login vektor kisebb helyiértékű része

A login vektor egy 26 bites szám, amely azt adja meg, hogy mely logikai meghajtók használhatók. Az UPM-től eltérően a rendszer valamennyi meghajtója állandóan „él” (bejelentkezett állapotban van), így a login vektor egyszerűen azt mutatja, hogy mely meghajtók vannak jelen a rendszerben. A 0. bit az A: meghajtónak felel meg, és így tovább; az „1”-be állított bitek jelzik a meglévő meghajtókat.

5.3.4.24. Lemez-foglaltsági információk beolvasása — funkció hívási kód: 15H

Paraméterek: B = logikai egység szám (1. . . 26)

Kimeneti interfész: A = állapotkód
B = szektorok száma alapegységenként
DE = alapegységek száma a lemezen
HL = szabad alapegységek száma (HL < DE)
IX = az egység-leíró terület címe

E funkció fő felhasználása információ szerzése a diszk méretéről és a szabad terület nagyságáról. Az alapegységenkénti sektorszám, amit a B regiszterben kapunk meg, mindig 2 egész kitevőjű hatványa, a sektorméret pedig mindig 512 bájt. Így az alapegység mérete 512 bájtól 64k-ig terjedhet. A szabad hely fölmérése néhány másodpercet igénybe vesz, mert a diszken az egész fájlallokációs táblát meg kell vizsgálni.

Az IX-ben visszaadott mutató felhasználásával részletesebb információt kaphatunk a lemezformátumról. Ez a mutató a FISH saját belső egység-leíró területét címzi meg, amely a diszk-csatoló RAM-jában van, így a felhasználónak egyáltalán nem szabad módosítania. Az egyes mezőket az alábbiakban ismeretjük. Természetesen beolvasásuk előtt a diszk RAM-ot be kell lapozni.

| | |
|----------|---|
| IX-1 | aktuális egységszám (1. . .26) |
| IX+0 | fizikai egységszám (1. . .26) |
| IX+1 | belső használatra |
| IX+2 | alapegység-maszk (alapegységenkénti sektorszám - 1) |
| IX+3 | alapegység-eltolás (az alapegységenkénti sektorszám 2-es alapú logaritmus) |
| IX+4,5 | az első FAT szektor sektorszáma |
| IX+6 | a FAT példányok száma |
| IX+7 | a szabad bejegyzések száma az utolsó szektorban |
| IX+8 | a kiindulási könyvtár első szektorának sektorszáma |
| IX+9 | az egy FAT-ban levő szektorok száma |
| IX+10,11 | a kiindulási könyvtár első szektorának sektorszáma |
| IX+12,13 | az első adatszektor sektorszáma |
| IX+14,15 | az adat-alapegységek száma + 2 |
| IX+16 | „piszkos diszk” jelző 0 => tiszta; 1 => piszkos |
| IX+17-20 | kötet-azonosító. Ha (IX+15) negatív, akkor nincs kötetazonosító, és a „piszkos diszk” jelző nem definiált |
| IX+21 | hordozó bájt |

5.3.4.25. Logikai meghajtó konvertálása fizikai meghajtóvá – – funkcióhívási kód: 16 H

Paraméterek: B = b0...b6 = egységszám (1...26)
 b7 „0” =>logikai egység
 „1” =>fizikai egység

Kimeneti interfész: A = állapotkód
 B = fizikai meghajtószám
 C = aktuális meghajtószám
 IX = az egységleíró terület címe

Ez a funkció logikai, vagy fizikai egységszám konvertálására szolgál fizikai és aktuális meghajtószámmá. A logikai egységszámok azok, amelyeket a felhasználó begépel a parancsokban, és amelyeket az FCB-be teszünk.

Ezeket a FISH fizikai egységszámokká konvertálja az ASSIGN műveletekben megadott összerendelések szerint (l. 5.3.4.28.). Ez egy egyszerű konverzió és, ha nem volt ASSIGN akkor a logikai és fizikai meghajtószámok megegyeznek.

A fizikai egységszámokat aktuális egységszámokká konvertálja a FISH a MAPDISK műveletekre megadott összerendelések szerint (l. 5.3.4.27.). Egy meghajtós rendszerben a „B:” fizikai meghajtó automatikusan az „A:” aktuális meghajtóhoz rendelődik, így két diszk érhető el az egy meghajtón. Az aktuális egységszámok közvetlenül meghajtókat jelentenek, így az 1...4 aktuális egységszámok mindig azt a négy meghajtót jelentik, amelyeket a DISKIO kezelni tud.

Ez a funkció olyan parancsok számára készült, amelyek a DISKIO-t közvetlenül használják – különösen, ha emellett FISH funkcióhívásokkal is fordulnak a diszkhez. A felhivatástól kapott meghajtószám logikai egységszámnak tekintendő, és közvetlenül átadandó a FISH-hívásoknak. A DISKIO viszont meghajtó-maszkot vár, amely az 1...4 aktuális meghajtónak felel meg. Ezért a felhivatástól kapott logikai meghajtószámot aktuális meghajtószámmá kell konvertálni, mielőtt meghajtómaszkot készítenénk belőle, s ezt

átadnánk a DISKIO-nak. Ez biztosítja, hogy minden érvényes ASSIGN és MAPDISK művelet érvényre jut.

Az egységeíró terület mutatója, amelyet a HL-ben kapunk meg azonos a „lemez-foglaltsági információ beolvasása” funkciótól visszakpott mutatóval (l. 5.3.4.24.). Megjegyzendő azonban, hogy jelen funkció nem vizsgálja meg a lemezt, mielőtt visszaadná a mutatót, így lehet, hogy számos mező nem lesz kitöltve. Ha a funkció hívása előtt közvetlenül a diszkhez fordulunk, akkor az egységeíró mezők érvényesek lesznek.

5.3.4.26. Lemez formattálása – funkcióhívási kód: 17H

Paraméterek: B = „hordozó” bájt a kívánt formátumhoz
 C = logikai egységszám (1 . . . 26)
 DE = pufferverület (6500 bájt) címe

Kimeneti interfész: A = állapotkód

Ez a funkció formattálja a megadott meghajtóban levő lemezt, a „hordozó” bájt által definiált módon (l. alább). Nincs ellenőrzés arra vonatkozóan, hogy a lemez már formattált-e, a hívó programnak kell megbizonyosodni róla, hogy a felhasználó valóban formattálni akarja a lemezét.

A funkció ellenőrzi, hogy az adott meghajtó alkalmas-e a kívánt formátumban való működésre, és szükség esetén 40-ben korlátozza a pályák számát, ill. 1-ben az oldalak számát.

A hordozó-leíró bájt olyan, mint a FATID bájt a FAT elején és a boot-szektorban az MS-DOS2 utáni változatú lemezekon:

| hordozó-leíró | oldal | szektor/pálya | pálya/oldal |
|---------------|-------|---------------|-------------|
| 0F8H | 1 | 9 | 80 |
| 0F9H | 2 | 9 | 80 |
| 0FAH | 1 | 8 | 80 |
| 0FBH | 2 | 8 | 80 |
| 0FCH | 1 | 9 | 40 |
| 0FDH | 2 | 9 | 40 |
| 0FEH | 1 | 8 | 40 |
| 0FFH | 2 | 8 | 40 |

A megformált lemez boot-szektora egy véletlenszerű kötetazonosítóval jön létre. A FAT-ok és a kiindulási könyvtárak inicializálódnak és a lemeznek nem lesz kötet-azonosítója.

5.3.4.27. MAPDISK (fizikai meghajtók egymáshoz rendelése) – – funkcióhívási kód: 18H

Paraméterek: B = első fizikai egység szám
 C = második fizikai egység szám

Kimeneti interfész: A = állapot kód

Az első fizikai egység szám ugyanazon az aktuális egység számon fogja elérni a diszket, mint a második fizikai egység – nem pedig a saját aktuális egység számán. Csak létező egységek adhatók meg mindkét paraméterben. Az összerendelés nem törölhető.

Ez a funkció legfőképpen az egymeghajtós rendszerek támogatására szolgál. Egymeghajtós rendszerben a FISH az inicializáláskor a „B:” meghajtót automatikusan az „A:”-hoz rendeli.

5.3.4.28. ASSIGN (logikai és fizikai meghajtók összerendelése) – – funkcióhívási kód: 19H

Paraméterek: B = első (logikai) egységszám
 C = második (fizikai) egységszám

Kimeneti interfész: A = állapotkód

Ez a funkció a parancsértelmező számára készült az ASSIGN parancs végrehajtásához. Ha egyik egységszám sem 0, akkor minden az első (logikai) egységszámra vonatkozó jövőbeni hivatkozás konvertálódik és a második (fizikai) egységszámra fog vonatkozni. Ha a második egységszám 0, akkor az első egységszám visszarendelődik önmagához (C=0 egyenértékű a C=B-vel). Ha az első egységszám (a B regiszter) 0, akkor minden előző ASSIGN művelet törlődik.

Megjegyezzük, hogy e funkció használata után a megnyitott FCB-k hirtelen másik lemezre fognak vonatkozni (mivel az FCB-ben levő egységszámok logikai egységszámok). Ennélfogva ajánlatos, hogy a funkció meghívásakor ne legyenek megnyitott FCB-k.

5.3.4.29. Rendszerváltozók beolvasása/beállítása – funkcióhívási kód: 1AH

Paraméterek: B = 0 ==> olvasás, egyébként írás
 D = új adat, ha B $\neq 0$
 E = a változó száma

Kimeneti interfész: A = állapotkód
 D = a változó aktuális értéke
 HL = a változóra mutat a diszk RAM-jában

Ez a funkció a B regiszter értékétől függően beolvassa, vagy beállítja a megfelelő változó értékét. Visszatéréskor a D mindkét esetben tartalmazza az aktuális értéket. A HL tartalma használható a változó közvetlen megcímezésére, ehhez azonban a diszk RAM-ját be kell lapozni a 3. memórialapra. A változók konkrét memóriacímeinek azonossága nem garantált a VT-DOS különböző verzióiban.

5.3.4.30. Aktuális dátum beolvasása – funkcióhívási kód: 1BH

Paraméterek: —

Kimeneti interfész: A = állapotkód
HL = év (1980. . .2079)
D = hónap (1. . .12)
E = nap (1. . .31)
C = nap a héten belül (0. . .6, 0=> vasárnap)

A VT–DOS nyilvántartja a dátumot és az időt, ha a készülék bekapcsolt állapotban van. Ez a funkció az aktuális dátumot adja meg, mely csak akkor érvényes, ha a legutóbbi hideg reset után beállítottuk. A VT–DOS a megfelelő módon kezeli a szökőéveket.

A dátumot és az időt a VT–DOS az 1/50 másodpercenként beérkező programmegszakítások alkalmával növeli. Így, ha az IT-t ennél hosszabb időre letiltják, „elveszhet” egy IT, és az óra késhet. A diszk-hozzáférések, a „meleg” reset, az áttérés a BASIC-be, vagy vissza a VT–DOS rendszerbe hosszabb időre letilthatják az IT-t. Normál használat esetén tehát az óra néhány másodpercet késhet.

5.3.4.31. Aktuális dátum beállítása – funkcióhívási kód: 1CH

Paraméterek: HL = év (1980. . .2079)
D = hónap (1. . .12)
E = nap (1. . .31)

Kimeneti interfész: A = állapotkód

Ez a funkció beállítja az aktuális dátumot. Érvénytelen dátum megadása esetén „.IDATE” hibát jelez.

5.3.4.32. Aktuális idő beolvasása – funkcióhívási kód: 1DH

Paraméterek:

Kimeneti interfész: A = állapotkód
H = óra (0. . .23)
L = perc (0. . .59)
D = másodperc (0. . .59)
E = századmásodperc (0. . .99)

Megadja a VT–DOS által nyilvántartott aktuális időt.

5.3.4.33. Aktuális idő beállítása – funkcióhívási kód: 1EH

Paraméterek: H = óra (0. . .23)
L = perc (0. . .59)
D = másodperc (0. . .59)
E = századmásodperc (0. . .99)

Kimeneti interfész: A = állapotkód

Ez a funkció beállítja a VT–DOS által nyilvántartott időt. Érvénytelen idő megadása esetén „.ITIME” hibát jelez.

5.3.4.34. A diszk–ROM verziószámának beolvasása – funkcióhívási kód: 1FH

Paraméterek: –

Kimeneti interfész: A = állapotkód
B = fő verziószám
C = aktualizálási szám

Megadja a diszk–ROM verziószámát. Pl. az 1.0 verzió esetén B=1 és C=0.

5.3.5. Pufferelés diszk-műveleteknél

Ahol lehetséges, a VT-DOS közvetlenül a felhasználói memóriába írja az adatokat. Amikor viszont nem teljes szektorok átviteléről van szó, vagy a 3. lap felhasználói RAM-jába kell írni, szükség van pufferelésre. Puffereket igényel a FISH ahhoz is, hogy belsőleg manipulációkat végezzen a könyvtárban és a FAT-okon. A VT-DOS ezért 512 bájtos szektorpufferek összefűzött listáját kezeli.

A pufferek sorrendjét a láncon belül a FISH állandóan változtatja, hogy egy olyan prioritási sémát alakítson ki, ahol a leggyakrabban szükséges adatok, mint pl. a FAT szektorok lehetőleg bent tartózkodjanak a szektor-pufferekben ha szükség van rájuk. Ezzel a szükséges diszk-hozzáférések száma csökken.

5.3.6. Hibakezelés

Sokféle hiba fordulhat elő a FISH használata során. Nagy részüket, mint pl. a „file not found” (a fájl nem található), vagy a „disk full” (lemez megtelt) hibajelzéseket a FISH egyszerűen visszaadja a hívó programnak (pl. a parancsértelmezőnek), amely kiírja a megfelelő hibaüzenetet.

Néhány hibát azonban, amelyek az alacsony szintű diszk-kezeléssel függenek össze, külön kezel. Ezeket a hibákat „diszk-hibák” néven fogjuk említeni. Diszk-hiba esetén, ha van definiált felhasználói hibavektor (l. 5.3.6.2.), akkor ezt a szubrutint a FISH meghívja, hogy megkapja a hibára adandó választ: kilépés a funkcióból (abort), újrapróbálkozás (retry), vagy figyelmen kívül hagyás (ignore). Ha nincs felhasználói hiba-szubrutin definiálva, vagy nem ad választ, akkor az 5.3.6.1. pontban leírt beépített hibakezelő lép működésbe.

5.3.6.1. A beépített diszkhiba-kezelés

A beépített diszkhiba-kezelés megjelenít a képernyőn egy hibaüzenetet és egy kérdést. A hibaüzenet közli, hogy melyik meghajtón (fizikai egység szám) és milyen hiba történt, pl. „not ready” (nem üzemkész), vagy „write protected” (írásvédett). A kérdés általában: „Abort, Retry, or Ignore” (kilépni, újra próbálni, vagy figyelmen kívül hagyni). Amikor az „Ignore” válasz nem ajánlatos, ez a lehetőség kimarad a kérdésből, de általában ilyenkor is elfogadja ezt a választ a hibakezelő. A kérdés kiírása után a billentyűzetről várja a választ, amelyeket az operátor a következő karakterekkel adhat meg:

| | |
|--------|-------------------------------------|
| IGNORE | – „I”, „i” |
| RETRY | – „R”, „r”, ENTER |
| ABORT | – „A”, „a”, ESC, CTRL-„C”, CTRL-ESC |

A RETRY válasz esetén a FISH újra megpróbálkozik a hibásan végződött művelettel; ha ismét hiba van, akkor jelenti a hibát.

IGNORE válasz esetén a hibával végződött műveletet úgy tekinti, mintha eredményes lett volna, s a feldolgozás tovább folyik – valószínűleg rossz adatokkal. Olyan esetekben, amikor ez veszélyes (FAT-, vagy könyvtár -hozzáférés), az „Ignore” válasz hiányzik a kérdésből. Az IGNORE opció választása azonban más esetekben is általában adatvesztéssel jár, így rendkívül óvatosan és csak végső megoldásként szabad használni.

ABORT válasz esetén a folyó FISH funkcióhívás megszakad és „.ABORT” hibajelzést ad a hívónak. A diszkhiba-kód, amely a kilépést okozta, beolvasható az „AB-ERR” rendszerváltozóból, bár erre az információra általában nincs szükség.

5.3.6.2. A felhasználói hibavektor

A felhasználó létrehozhat egy szubrutint, amely „feltartóztatja” a diszk-hibákat, még mielőtt a FISH kiírná a hibaüzenetét. Ez hasznos lehet olyan progra-

moknak, amelyek bizonyos hibakódokat értelmezni kívánnak, illetve alkalmazható a teljes hibakezelési rendszer kicserélésére.

Annak részletei, hogy a program miként hozhatja létre a hibavektort, a VT-DOS kézikönyv 4. fejezetében olvashatók. Az itt következő magyarázat csak a FISH-t közvetlenül hívó programok, mint pl. a VT-BDOS és a MOPS DISK DEVICE esetére vonatkozik.

A felhasználói hiba-szubrutin definiálása egyszerűen úgy történik, hogy kezdőcímét az „ERV_ADDR” változóba írjuk. Ha ez nulla, akkor nincs definiált felhasználói hibarutin. A diszk-RAM-ban levő FISH-változók címeit az 5.3.7. fejezet ismerteti.

A felhasználói hibarutin indulásakor a lapozási állapot olyan, mintha a FISH futna; de egy belső FISH stack-kel, amely a diszk-RAM területén van. A szubrutinnak átadott paramétereket és a várt válaszokat alább ismertetjük. Az összes többi fő-regiszter (BC, DE, HL, IX) tartalma megváltozhat, de a másod-regiszterkészlet (AF', BC', DE', HL') tartalmát meg kell őrizni. Ha a szubrutin az „A” regiszterben 1-et, 2-t, vagy 3-at ad vissza, a normál FISH hibakezelést felülbírálja.

Paraméterek: B = hibakód
 C = fizikai egység szám, ahol a hiba jelentkezett
 A = 00H, ha az IGNORE válasz ajánlható
 = 0FFH, ha az IGNORE válasz nem ajánlott

Kimeneti interfész: A = 0 => használjuk a beépített hibakezelést
 1 => ABORT
 2 => RETRY
 3 => IGNORE

A szubrutin íráskor ügyelni kell arra, hogy ne hívja meg a FISH-t sem közvetlenül, sem közvetve készülék átirányítással, mert a FISH nem újrarahívható.

Bár valójában nem diszk-hiba, a fenti diszkhiba-kezelő rendszer végzi az operátor felszólítását a megfelelő lemez meghajtóba helyezésére.

Ez csak akkor fordul elő, amikor a MAPDISK funkciót használjuk (l. 5.3.4.27.), tehát általában egymeghajtós rendszerekben. Az ilyenkor a hibavektor szubrutinnak átadott hibakód a „.WDRV”, a szokásos válasz pedig a megfelelő lemez behelyezése után a RETRY (A=2); bár ABORT-ot (A=1 is meg lehet adni).

5.3.7. Változók a diszk–RAM területén

A FISH RAM-területe a diszkcsatoló RAM-jában van; főként belső változókból és pufferekből áll. Néhány változó azonban – az ún. „definiált változók” – jól ismert helyen található, így más programok (különösen a VT–BDOS) is használhatják. Megjegyezzük, hogy ezek nem ugyanolyanok, mint a „rendszerváltozók”, amelyekhez FISH funkcióhívással hozzá lehet férni. A különbség az, hogy a „definiált változók” csupán a FISH-t közvetlenül hívó programok céljait szolgálják, míg a rendszer-változók nevezhetők „felhasználói változók”-nak is; maga a parancsértelmező is rendelkezik egy VAR paranccsal, amely e változók használatára szolgál.

A kompatibilitás érdekében a definiált változók címe a diszk–RAM esetleges későbbi változataiban sem fog megváltozni.

| | |
|----------|---|
| SEL_TIME | A DISKIO-ig hátralevő idő ha lejár, deszelektálja az összes meghajtót. 110-től 0-ig számlál vissza 50Hz frekvenciával és nullázódik, ha a DISKIO-t meghívjuk. |
| ERV_ADDR | A felhasználói hibarutin címe (l. 5.3.6.2.) |
| FSH_PG | A 3. memórialap lapozása = 0 => felhasználói RAM kiválasztása nem = 0 => diszk-RAM kiválasztása. |

5.4. A közvetlen diszk-kezelő modul (DISKIO)

5.4.1. Bevezetés

A DISKIO a legalsó szintű diszk-kezelést valósítja meg, így közvetlenül használja a diszk-csatolót.

Az olvasási és írási funkciók lehetővé teszik több egymásra következő, egy pályán található szektor olvasását, írását és ellenőrzését, egy pálya formattálását megadott minta szerint, teljes pálya beolvasását, és szektor-fejek beolvasását. Hiba esetén nem próbálkozik újra, ez a hívó program feladata. Lehetőséget biztosít a „diszk kicserélve” jel vizsgálatára (ha van ilyen a hardverben), és a meghajtó típusának meghatározására.

A FISH minden diszk-műveletet a DISKIO-n keresztül végez. A DISKIO a VT-DOS felhasználó számára is biztosít belépési pontot (l. VT-DOS kézikönyv 4.2.6. fejezet, így a felhasználó is felhasználhatja közvetlenül a diszket; pl. DISKCOPY nevű tranziens program használja a DISKIO „szektorok olvasása” funkcióját. A DISKIO-n kívül egyetlen programnak sincs szüksége arra, hogy a diszk-csatoló hardvert közvetlenül használja.

A DISKIO és a diszk-csatoló hardver sok különféle típusú meghajtóval tud dolgozni: 5.25”, 3.5” és 3.0” méretűekkel, ill. ezek kombinációjával. Használhatók a READY jellel rendelkező és nem rendelkező meghajtók egyaránt. A felírási formátum lehet szimpla sűrűségű (FM) 125 kbit/sec, vagy dupla sűrűségű (MFM) 250 kbit/sec. Egy- és kétoldalas meghajtók is használhatók.

A DISKIO bármilyen szoftver-es lemezformátumot képes kezelni, ami kompatibilis a fenti hardver feltételekkel, továbbá a WD1793 diszkverzáló IC-vel és a TVC diszk-csatoló hardverrel – de a normál VT-DOS lemezek dupla sűrűségűek, szektoronként 512 bájjal. A VT-DOS által használt normál formátumok: egy- vagy kétoldalas, 40 vagy 80 pályás, 8 vagy 9 szektor/pálya. Más formátumokat is el lehet érni, ha a lemez tartalmazza a megfelelő paraméterblokkot. Minden program, amely a DISKIO-t használja (mint pl. a CONVERT utility), bármely, a hardver által támogatott formátumban dolgozhat.

A 5.4. fejezet leírja a DISKIO működését, azt, hogy milyen parancsok adhatók, és milyen paramétereket kell átadni.

5.4.2. Parancsok

A DISKIO híváskor az „A” regiszterben át kell adni egy „parancskód”-ot, a többi regiszterben pedig más paramétereket. A visszatéréskor a regiszterekben ad vissza információkat; az „A” regiszterben hibakódot találunk. Ezek nem a MOPS-ban és a VT-DOS többi részében használt általános hibakódok, így közvetlenül nem adhatók vissza a felhasználónak. A DISKIO visszatérése előtt egy „OR A” utasítással beállítja a jelzőbiteket.

Paraméterek:

IX = diszk átviteli cím

A = parancskód: 0 = diszk-rendszer reset
1 = szektorok olvasása
2 = szektorok írása
3 = szektorok ellenőrzése
4 = pálya formattálása
5 = pálya olvasása
6 = cím olvasása
7 = fenntartva későbbi felhasználásra
8 = meghajtó vizsgálata

B = meghajtó kiválasztása (1,2,4,8) (1...8 parancsoknál)
= meghajtó maszk (0...15) (0. ")
0.bit: 0. meghajtó
1.bit: 1. meghajtó
2.bit: 2. meghajtó
3.bit: 3. meghajtó

C = oldal száma (0,1) (1...6 ")

D = pálya száma (0...73) (1...6 ")
 0FFH => aktuális pálya

E = szektor száma (1...9) (1...3 ")

H = 0-1. bit: lépéssebesség
 0 => 6 ms 1 => 12 ms
 2 => 20 ms 3 => 30 ms
 2. bit: előkompenzálás engedélyezése (2, 4. ")
 3. bit: dupla pálya jelző engedélyezése (1...6 ")
 4 bit: 0
 5. bit: felírási mód (1...6 ")
 0 => MFM, 1 => FM
 6-7. bit: 0

L = szektorok száma (1...9) (1...3 és 6. ")

Kimeneti interfész:

A = hibajelző bitek
 0. bit: nem üzemkés (1...6 és 8. ")
 1. bit: hiba az adat-ellenőrzésnél (3. ")
 2. bit: adatvesztés (1...6 ")
 3. bit: CRC-hiba (1...3 és 6. ")
 4. bit: a rekord nem található (1...3 és 6. ")
 5. bit: fenntartva későbbi felhasználásra
 6. bit: írásvédett (2., 4. ")
 7. bit: — ismeretlen parancs (9...255 ")
 — nincs megadva meghajtó (1...6 ")

F = a jelzőbitek „OR A” utasítással beállítva

B = változatlan (0...6 és 8. ")

L = változatlan (0. ")
 = átvitt szektorok száma (1...6 ")
 = meghajtó típusa (8. ")

A DISKIO hívása és visszatérése után a fent jelzett eseteket kivéve az A, F, B és L regiszterek tartalma megváltozik, hiszen információkat adnak vissza.

Az összes többi regiszter (C, DE, H, IX, IY, AF', BC', DE', HL') tartalma megmarad.

A nem szabványos lemezformátumok kezelhetősége érdekében a DISKIO nem ellenőrzi a kapott paramétereket, azonban a meghajtó-maszk (B), a fej-szám (C) és a lépésszélesség (H) paramétereket használható értékére maszkolja alkalmazásba vétel előtt, mivel más értékek értelmetlenek a hardver számára.

A DISKIO nem lapozza a memóriát, így visszatéréskor változatlan a lapozási állapot. Megőrzi az IT-rendszer engedélyezett, vagy tiltott állapotát is, így mindkét esetben hívható.

5.4.3. A diszk-átviteli cím

A DISZK-átviteli cím minden adatolvasás, -írás és ellenőrzés parancshoz szükséges. IX-nek tartalmaznia kell egy olyan, folytonos memóriaterület kezdőcímét, amely elég nagy az átvinni kívánt adatmennyiség számára. Ez a memóriaterület a 0, 1 vagy 2. memórialapon lehet.

5.4.4. DISZK-RENDSZER RESET parancs

A RESET parancsból való visszatéréskor az „A” regiszter mindig „0”. A parancs a „B” regiszterben kap egy négybites meghajtó-maszkot, amellyel a négy meghajtó tetszőleges kombinációját meg lehet adni. Minden meghajtóra, amelyet így kiválasztottunk, az „aktuális pálya” változó OFFH értéket kap. Ennek az a hatása, hogy amikor legközelebb a meghajtóhoz fordulnak, „restore” fog történni, ami biztosítja, hogy a fej a megfelelő pályán legyen.

Rendszerint, ha egy olvasási vagy írási művelet hibával végződik egy adott meghajtón, akkor az újabb próbálkozás előtt ki szokták adni egy, csak erre a

meghajtóra vonatkozó RESET parancsot. Ezt a kérdést később, az 5.4.9. fejezetben részletesebben is tárgyaljuk.

Minden RESET parancs, a specifikált meghajtótól függetlenül megvizsgálja, hogy a WD1793 diszkvezérlő IC foglalt-e, s ha igen, akkor egy „kényszerített megszakítás” parancssal leállítja. Ez biztosítja, hogy ha az IC kétes állapotban maradt, akkor a DISKIO helyrehozza.

Ha nem specifikálunk meghajtót (B=0), akkor a parancs az összes meghajtót deszelektálja. Ez elsősorban a FISH céljait szolgálja, amely egy 2 másodperces időzítés után deszelektálja a meghajtókat, természetesen a felhasználó is igénybe veheti ezt a funkciót.

A meghajtók deszelektálása előtt a DISKIO ellenőrzi, hogy a motor megy-e még. Ha már megállt, akkor minden meghajtót deszelektál oly módon, hogy a meghajtó-kiválasztó címre 0-t ír. Ha a motor még megy, akkor az „A:” meghajtót kiválasztja, és az időzítést visszaállítja 10-re, hogy a deszelektálási kísérlet 1/5 másodperc múlva megismétlődjék. Ez az eljárás csökkenti annak lehetőségét, hogy a motor folyamatosan engedélyezve maradjon egy lemez behelyezése után, ami előfordulhat, ha működő motornál deszelektálják a meghajtót.

5.4.5. Szektorok olvasása, írása és ellenőrzése

E három parancs paraméterei (a parancsszám kivételével) azonosak, s a kimeneti interfészük is azonos. Az egyetlen különbség az adatmozgatás irányában van. Az „ellenőrzés” parancs összehasonlítja az átviteli területen levő és a diszkre felírt adatokat. Ha eltérést talál, akkor ellenőrzési hibát jelez.

A felhasználó az 5.4.2. alfejezetben leírt módon megad egy átviteli címet, meghajtó-jelzőt, oldalszámot, pályaszámot, az első szektor számát, és a szektorok darabszámát.

A DISKIO először kiválasztja a specifikált meghajtót, a fejet és a felírási módot (FM, vagy MFM), majd megkeresi a specifikált pályát („seek”-et hajt végre), ha a fej nem ott van (ha a meghajtót előzőleg resetelték, akkor előbb

restore-t is végez). Ha a megadott pályaszám OFFH, akkor az aktuális pályát fogja használni, így nem végez seek-et.

A megfelelő pálya beállítása után megkeresi a specifikált sektorszámot. A sektorszám az első olyan sektort jelöli ki a pályán, amelyre, vagy amelyről adatot kell mozgatni. A szektorok darabszámát adja. Ha a kért darabszám túllép a pálya végén, akkor csak a pályán levő szektorok adatait viszi át a DISKIO, és egy „rekord nem található” hibajelzéssel tér vissza (bizonyos késéssel, mert a vezérlő IC megpróbálja megtalálni a nemlétező sektort). Megjegyzendő, hogy bár az „egymásra következő szektorok” sektorszáma egymásra következik, ez nem jelenti feltétlenül azt, hogy a lemezen fizikailag egymás után helyezkednek el.

Rendszerint minden szektor mérete 512 bájt, és ekkora területnek minden kért szektor számára rendelkezésre kell állnia az átviteli memóriacímen. Ha nem-szabványos lemezeket olvasunk, a sektorméret lehet 128, 256, 512 vagy 1024 bájt is, vagy akár ezek keveréke. A hívó programnak kell biztosítania, hogy a diszk-átviteli terület elég nagy legyen.

Ha a „dupla pálya” jelző „1”-be van állítva, akkor a pálya elérésekor kiadott léptetőimpulzusok száma megkettőződik. Ez arra alkalmazható, hogy a 40-pályás lemezt 80-pályás meghajtóban használjuk.

Ha egy „szektorok írása” parancsnál az „előkompenzálás” flag 1-be van állítva, akkor a WD1793 IC „előkompenzálás” funkcióját a DISKIO engedélyezi. Ezt rendszerint a belső pályáknál (nagy pályaszámok) engedélyezni, a külső pályáknál pedig tiltani kell.

5.4.6. Formattálás és pályaolvasás

A „pálya formattálása” parancs azonos a „szektorok írása” parancssal – azzal a különbséggel, hogy nem különálló szektorokat ír, hanem egy egész pályát; a „szektorok száma” paramétert nem veszi figyelembe. Ehhez hasonló a kapcsolat a „pályaolvasás” és a „szektorok olvasása” parancs között.

A „pályaolvasás” parancssal olvasott adatok a szektorok közti hézagokat (gap), a szektorfejeket és címjelzéseket is tartalmazzák a szektorokban levő

adatokon kívül. Hasonlóképpen, a „pálya formattálása” parancs által írt adatok magukban foglalják a hézagokat és szektorfejeket, valamint speciális bájtokat, amelyek címjelzések és CRC-értékek fölírását vezérlik, a WD1793 IC specifikációjának megfelelően. E speciális bájtok miatt a „pályaolvasás” által olvasott adatok nem azonosak a „pálya formattálása” parancs által írt adatokkal.

Megjegyezzük, hogy a „pályaolvasás” parancs majdnem mindig jelez egy „adatvesztés” hibát, de ezt általában figyelmen kívül lehet hagyni. Ez a hiba abból származik, hogy a WD1793 IC -nek két szektor között mindig újra kell szinkronoznia magát az adatokra, és csupán azt jelenti, hogy a hézagban levő bájtok közül néhány elveszett – az adatbájtok bizonyára jól olvashatók be. Ez a WD1793-nak egy elkerülhetetlen tulajdonsága.

A pályatartalom névleges mérete mind a „pályaolvasás”, mind a „pálya formattálása” parancs esetén 6250 bájt a dupla sűrűségű (MFM) és 3125 bájt a szimpla sűrűségű (FM) lemezeken normál percenkénti fordulatszámnál (300). A diszk-sebesség változása miatt azonban meg kell engedni egy biztonsági többletet, így 6500 (ill. 3250) a javasolt érték.

5.4.7 Cím-olvasás

Ezt a parancsot általában hiba előfordulása után szokták meghívni, hogy segítsen a hívó programnak (különösen a FISH-nek) a hiba meghatározásában. Használható szimpla és dupla sűrűségű lemezek megkülönböztetésére és idegen lemezformátumok meghatározására is (pl. másolás ellen védett lemezeknél).

Működésében hasonló a „szektorok olvasása” parancshoz, de a kiválasztott szektoroknak csak a 6-karakteres „szektorfej” részét adja át, több szektor esetén egymás után, a megtalálás sorrendjében.

Egy szektorfej felépítése:

| | | | |
|------|------------------|------------|---|
| IX+0 | a pálya száma | (0...255) | |
| IX+1 | az oldal száma | (0 vagy 1) | |
| IX+2 | a szektor száma | (0...255) | |
| IX+3 | a szektor hossza | (0...3) | --> 0 => 128 bájt 1 => 256 bájt 2 => 512 bájt 3 => 1024 bájt |

IX+4,5 CRC bájtok

5.4.8. A meghajtó vizsgálata

Ez a funkció két különböző műveletet hajthat végre, amelyek közül a B regiszterben átadott paraméter választ. Ha B=0, akkor a funkció eldönti, hogy egymeghajtós-e a rendszer (az „A:” meghajtó létezik, de a „B:” nem). Ha a B regiszter tartalma nem 0, akkor az kijelöl egy meghajtót, amelyről a funkció eldönti, hogy 40 vagy 80 pályás.

Az első funkciót (B=0) a FISH használja hideg reset esetén. Ha azt állapítja meg, hogy egymeghajtós a rendszer, akkor a „B:” meghajtót automatikusan az „A:”-hoz rendeli a FISH. Ez a DISKIO funkció az A regiszterben 0-t ad vissza, ha a rendszer egymeghajtós; egyébként $A \diamond 0$. Ilyenkor az A regiszter tartalma nem a normál DISKIO hibajelzőket tartalmazza.

A vizsgálatot a DISKIO úgy végzi, hogy kiad egy „restore” parancsot az „A:” meghajtónak, és ellenőrzi, hogy a fej beállt-e a 0. pályára. Ha nem, akkor az „A:” meghajtó nem létezik, tehát a rendszer nem egymeghajtós. Ha beállt a 0. pályára, akkor a művelet megismétlődik a „B:” meghajtóra. Ha ez is beállt a 0. pályára, akkor megintcsak nem egymeghajtós a rendszer, ha viszont nem áll be, akkor a rendszer egymeghajtósnak bizonyult, így visszatéréskor az A regiszter tartalma 0 lesz.

Ha a B regiszter a parancs hívásakor nem 0, akkor kijelöl egy meghajtót, amelyet vizsgálni kell.

A vizsgálat menete a következő:

A meghajtó kap egy „restore” parancsot, majd 50 db befelé léptető parancsot. Ezután a DISKIO visszalépteti a 0. pályára és megszámlálja, hogy ehhez hányszor kellett visszafelé léptetni. Ebből meghatározható, hogy a meghajtó 40, vagy 80 pályás, mert ha 40 pályás, akkor a befelé léptetéseknél elérte a végállást, így kevesebb kifelé léptetéssel visszajut a 0. pályára. A hívónak visszaadott eredmény az L regiszterben van; az A regiszter mindig 0:

L = 00H — 40 pályás

= 0FFH — 80 pályás

Megjegyezzük, hogy a FISH ezt a funkciót csak formattálás előtt használja, mivel más körülmények között a szektorfejek beolvasását alkalmazva egy megfelelő újra-próbálkozási stratégiával meg tudja állapítani a meghajtó típusát. A fent leírt léptetési módszer, bár teljesen ártalmatlan, egyes meghajtókon eléggé zajos lehet.

5.4.9. Hibák és javasolt újra-próbálkozási stratégiák

Ha a DISKIO HW-hibát észlel, visszajelzi azt a hívónak az A regiszterben. Ilyenkor általában ajánlatos, hogy a hívó program többször is újra próbálkozzon a művelet végrehajtásával, mielőtt feladná és jelentené a hibát a felhasználónak. A részletek természetesen eltérnek a különböző hibák esetén. Az operátornak mindig meg kell adni a lehetőséget az újbóli próbálkozásra, hiszen lehet valami, amivel segíthet (pl. lemezt tehet a meghajtóba).

Ha „nem üzemkész”, vagy „írásvédett” hiba fordul elő, akkor egyszeri sikertelen ismétlés után jelenteni kell az operátornak. A DISKIO biztosítja a megfelelő időzítést a motor indulásakor, stb., így a többszöri próbálkozás nem fog segíteni az operátor beavatkozása nélkül.

„Adatvesztés” hiba nem fordulhat elő (kivéve a „pályaalvasás” parancsot), mivel ez azt jelentené, hogy a belső DISKIO program nem volt elég gyors. Ha mégis előfordulna, akkor a CRC-hibához hasonlóan kellene kezelni.

A CRC-hibák, vagy a „rekord nem található” hibák gyakran olyan adathibából származnak, amelyet ismétléssel helyre lehet hozni. Ajánlatos először azonnal megismételni a műveletet; ha ismét eredménytelen, akkor még kétszer-háromszor úgy, hogy közben mindig végrehajtunk egy „reset” hívást is. Ha ez sem segít, akkor a hibát jelezni kell az operátornak.

Megjegyezzük, hogy a szektor-olvasás és szektor-ellenőrzés műveleteknél a DISKIO nem várja meg a motor felpörgését a szektorolvasás megkísérlése előtt. Ez gyakran CRC-hibához vezet, így ezeket mindig meg kell ismételni többször, mielőtt jelentenénk az operátornak. Más műveleteknél, mint pl. a szektor-írásnál, a DISKIO a kellő ideig vár, így a hasonló hibák kevésbé valószínűek, de előfordulhatnak.

A „rekord nem található” hiba esetén érdemes végrehajtani egy cím-olvasási parancsot, hogy láthassuk, hol van a fej a lemezen. Ha a beolvasott szektorfej szerint a 0. oldalon vagyunk az 1. oldal helyett, akkor a kétoldalas lemezt egyoldalas meghajtóban próbáljuk használni (bár bizonyos egyoldalas meghajtók ilyen esetben nem is olvasnak címet). Ha hibás a pályacím, vagy a cím-olvasás parancs hibát jelez, akkor lehetséges, hogy a „dupla pálya” jelző rosszul volt beállítva. Ezt ellenőrizhetjük úgy, hogy reseteljük a meghajtót és megpróbálunk címet olvasni pl. a 8. pályáról úgy, hogy a „dupla pálya” jelző 0 értékű. Ha a visszakapott címben „8” a pályaszám, akkor a „dupla pálya” jelzőt „0” értékkel kell használni, ha pedig „4”, akkor „1” értékkel (a lemez 40 pályás).

„Ellenőrzési” hiba csak a „szektorok ellenőrzése” parancsban történhet, és nem a HW, hanem a DISKIO SW detektálja. Ha előfordul, akkor ismételni kell, de kívánatos lehet sikeres ismétlés esetén is értesíteni az operátort, mert azt jelentheti, hogy a lemez gyanús és hamarosan tönkremehet.

6. SPECIFIKÁCIÓ

6.1. Floppy csatoló

A floppy csatoló HBF típusjelű kártyát tartalmaz.

A kártyán felépített vezérlőegység lehetővé teszi mini hajlékony mágneslemez tároló mechanizmusok csatlakoztatását a TV Computerhez, valamint ezen meghajtók helyes működését koordinálja a megfelelő programok segítségével.

Az információ mágneslemezre történő rögzítése és visszaolvasása az IBM SYSTEM 34 formátumnak megfelelően történik.

A meghajtók működtetése során a következő interfész jeleket kezeli a HBF kártya:

| Jelnév | Jelcsatlakozó száma | Földcsatlakozó száma |
|------------------|---------------------|----------------------|
| IN USE | 4 | 3 |
| DRIVE SELECT 3 | 6 | 5 |
| INDEX | 8 | 7 |
| DRIVE SELECT 0 | 10 | 9 |
| DRIVE SELECT 1 | 12 | 11 |
| DRIVE SELECT 2 | 14 | 13 |
| MOTOR ON | 16 | 15 |
| DIRECTION SELECT | 18 | 17 |
| STEP | 20 | 19 |
| WRITE DATA | 22 | 21 |
| WRITE GATE | 24 | 23 |
| TRACK 00 | 26 | 25 |
| WRITE PROTECT | 28 | 27 |
| READ DATA | 30 | 29 |
| SIDE ONE SELECT | 32 | 31 |
| READY | 34 | 33 |

6.2. Mini–floppy egység

| | |
|---------------------------|--|
| Tárolókapacitás | dupla sűrű rögzítés esetén |
| Formátatlan: | |
| Diszk | 1000 kbájt |
| Felületenként | 500 kbájt |
| Sávonként | 6,25 kbájt |
| Formált: | 512 bájt/szektor esetén |
| Diszk | 720 kbájt |
| Felületenként | 360 kbájt |
| Sávonként | 4608 bájt (512 bájt x 9 szektor) |
| Adatátviteli sebesség: | 250 kbit/s |
| Átlagos késleltetési idő: | 100 ms |
| Hozzáférési idő: | |
| Sávról sávra | 3 ms |
| Átlagos | 94 ms (beleértve a 3 ms léptetési időt és a fej megnyugvási időt is) |
| Fejmegnyugvási idő | 15 ms |
| Fejleengedési idő | 50 ms |
| Motor felpörgési idő | 250 ms |
| | dupla sűrű rögzítés esetén |
| Jelrögzítési sűrűség | 5922 bit/inch |
| Kódolási eljárás | MFM |
| Sávsűrűség | 96 track/inch |
| Cilinderek száma | 80 |
| Sávok száma | 160 |

| | |
|-------------------|--|
| Fejek száma | 2 |
| Forgási sebesség | 300 ford/perc |
| Index | 1 |
| Információhordozó | |
| Jellemző | 96 TPI-s kétoldalas 5,25 inch-es diskette |

Interfész jelek

A jelek megnevezését és a hozzárendelt csatlakozópont számát lásd a 6.1. pontban.

Hiba adatok

| | | |
|----------------------|----------|--------------|
| „Soft” olvasási hiba | $10E-9$ | bit |
| „Hard” olvasási hiba | $10E-12$ | bit |
| Pozícionálási hiba | $10E-6$ | pozícionálás |

Elektromos táplálás

- Bemenő feszültség 220V +10, –15%
- A bemenő frekvencia tartománya 47 – 63 Hz
- Bemenő teljesítmény: max. 50VA
- Érintésvédelmi osztály: I.

7. HARDVER–SZOFTVER INTERFÉSZ

Bár a floppy használatához szükséges szoftver tudnivalók a 3. fejezetekben megtalálhatók, a teljesség kedvéért ismertetjük az egység HW/SW interfészét.

A kártyán helyet foglaló floppy vezérlő elektronika szoftver szempontból a következő funkcionális egységekre bontható:

- memória
- memória „paging” (lapozási) regiszter
- paraméter regiszter
- HW állapot regiszter
- Floppy Disk Controller (FDC)

7.1. A memória felépítése

A kártyán a C000–DFFF tartományban 8 kb-ot memória érhető el a következőképpen:

| | | | | |
|------|--------------------|----------------------|----------------------|----------------------|
| C000 | PAGE0 (0 – FFF) | PAGE1 (1000–1FFF) | PAGE2 (2000–2FFF) | PAGE3 (3000–3FFF) |
| CFFF | | | | |
| D000 | RAM | | | |
| DFFF | | | | |

A PAGE0–PAGE3 memória területek egy darab 16k x 8 bites ROM-ban foglalnak helyet és egy page regiszter segítségével választható ki a kívánt lap (l. a következő pontban).

A gép hardver felépítése nem teszi lehetővé, hogy a bővítőkártyán 16 kbyte ROM foglaljon helyet, viszont bizonyos programok ezt megkövetelik. Ezt az ellentmondást áthidalandó a hardver 4 kbájtos részekben be tudja „lapozni” a 16 kbájtos memóriát.

7.2. Memória „paging” regiszter

A megfelelő lap a ROM-ból a következő táblázat alapján választható ki. (Az „X” értéke a bővítőkártyahelytől függően 1,2,3 vagy 4 lehet).

| | Page 0 | Page 1 | Page 2 | Page 3 |
|--------------|--------|--------|--------|--------|
| OUT X8H – ra | 00H | 10H | 20H | 30H |

Figyelem!

A HBF kártyán lévő memória eléréséhez az alapgépben biztosítani kell a megfelelő memória „page” beállítását.

7.3. Paraméter regiszter

A DS (Drive Select) jelek, MON (motor on), Side Select stb. vezérlőjelek értékét eltároló regiszter. Az OUT X4 utasítással érhető el. A paraméter regiszter felépítése a következő:

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----|-----|------|-----|-----|-----|-----|-----|
| SS | MON | DDEN | HLD | DS3 | DS2 | DS1 | DS0 |

D0–D3 bit

DS0–DS3 meghajtót kiválasztó jel

DS0=0 0. meghajtó inaktív

DS=1 0. meghajtó aktív

DS1=0 1. meghajtó inaktív

DS1=1 1. meghajtó aktív

DS2=0 2. meghajtó inaktív

DS2=1 2. meghajtó aktív

DS3=0 3. meghajtó inaktív

DS3=1 3. meghajtó aktív

D4 bit

HLD A floppy fejleengedését vezérlő jel

HLD=0 fej nincs a felületen

HLD=1 fej leengedése a felületre

(Jelen kiépítésben a periférián belül a HLD jel a MON jellel van kapuzva, tehát az író-olvasó fej mindaddig a felületen marad, amíg a Motor on (MON) jel aktív.)

D5 bit

DDEN ezt a bitet 0-ba kell állítani a duplasűrű működésmódhoz

D6 bit

MON (Motor on) a lemezt forgató motort kapcsolja be

MON=0 motor kikapcsolt állapotban van

MON=1 motor bekapcsolt állapotban van

D7 bit

SS (Side Select) lemezoldal kiválasztás

SS=0 0-ás oldal van kiválasztva

SS=1 1-es oldal van kiválasztva

7.4. Hardver állapot regiszter

E regiszter segítségével az FDC INTRQ és DRQ jelei olvashatók be. (Ezen jelek értéke az FDC státuszából is kiolvasható, de hozzáférésük hosszabb időt igényel). Ez a regiszter az IN X4 utasítással érhető el. A regiszter felépítése a következő:

| | | | | | | | |
|-------|----|----|----|----|----|----|-----|
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| INTRQ | 0 | 0 | 0 | 0 | 0 | 0 | DRQ |

7.5. Összefoglaló táblázat az I/O eszközökről

| Cím | I/O eszköz | Read | Write |
|-----|------------------------------------|---------------------------|---------------------------|
| X0H | FDC | Állapot regiszter | Command regiszter |
| X1H | FDC | Track regiszter | Track regiszter |
| X2H | FDC | Szektor regiszter | Szektor regiszter |
| X3H | FDC | Adat regiszter | Adat regiszter |
| X4H | <u>Parancs reg.</u> Kont.status | Controller * HW status | Paraméter ** regiszter |
| X8H | Page reg. | — | Page regiszter *** |

* Controller HW status l. az 5.4. pontban

** Paraméter regiszter l. az 5.3. pontban

*** Page regiszter l. az 5.2. pontban

Megjegyzés: Az I/O címeknél alkalmazott „X” helyére a kártyahelynek megfelelő értéket kell beírni.

Fontos!

„Reset után minden esetben aktiválni kell a „Page” regisztert, ellenkező esetben az FDC inaktív állapotban marad.

7.6. Floppy Disc Controller (FDC)

A HBF kártyán egy FD1793 típusú vezérlő koordinálja az összes mágneslemezrel kapcsolatos műveletet.

Ezen műveleteknek négy csoportja létezik az alábbi táblázat szerint:

| Parancs típus | Parancs összefoglaló bitek | | | | | | | |
|---------------------|----------------------------|---|---|---|----|----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I. Restore | 0 | 0 | 0 | 0 | h | V | r1 | r0 |
| I. Seek | 0 | 0 | 0 | 1 | h | V | r1 | r0 |
| I. Step | 0 | 0 | 1 | T | h | V | r1 | r0 |
| I. Step—In | 0 | 1 | 0 | T | h | V | r1 | r0 |
| I. Step—Out | 0 | 1 | 1 | T | h | V | r1 | r0 |
| II. Read Sector | 1 | 0 | 0 | m | S | E | C | 0 |
| II. Write Sector | 1 | 0 | 1 | m | S | E | C | a0 |
| III. Read Address | 1 | 1 | 0 | 0 | 0 | E | 0 | 0 |
| III. Read Track | 1 | 1 | 1 | 0 | 0 | E | 0 | 0 |
| III. Write Track | 1 | 1 | 1 | 1 | 0 | E | 0 | 0 |
| IV. Force Interrupt | 1 | 1 | 0 | 1 | I3 | I2 | I1 | I0 |

(Az FDC parancs regiszter az OUT X0H címen érhető el).

A parancs bitjeinek jelentése

(CT = Command típus)

| | | |
|-----------|---------|--|
| CT=I. | Bit 0,1 | r1r0=léptető motor sebessége |
| CT=I. | Bit 2 | V= Track szám ellenőrzés V=0, nincs cél track ellenőrzés V=1, track ellenőrzés a cél track-en |
| CT=I. | Bit 3 | h= fejleengedés vezérlő h=1, fejleengedés parancs kezdésnél h=0, fejfelemelés parancs kezdésnél |
| CT=I. | Bit 4 | T= Track módosítás T=0, nem módosul T=1, track regiszter tartalma módosul |
| CT=II. | Bit 0 | a0= Data Address Mark a0=0, → FB (DAM) a0=1, → FB (törölt DAM) |
| CT=II. | Bit 1 | C= Lemezoldal összehasonlítás C=0, oldal összehasonlítás tiltása C=1, oldal összehasonlítás engedélyez. |
| CT=II&III | Bit 2 | E= 15 ms-os késleltetés E=0, nincs 15 ms-os késleltetés E=1, van 15 ms-os késleltetés |
| CT=II. | Bit 3 | S= Lemezoldal összehasonlítás S=0, összehasonlítás 0-ra S=1, összehasonlítás 1-re |
| CT=II. | Bit 4 | m= többszörös rekord olvasás m=0, szimpla rekord olvasás m=1, többszörös rekord olvasás |
| CT=IV. | Bit 0–3 | Interrupt feltétel flag-ek I0=1 nem READY – READY átmenet I1=1 READY – nem READY átmenet I2=1 INDEX impulzus I3=1 Immediate Interrupt I3–I0=0 befejezés interrupt (INTRQ) nélk. |

Az átvitel és a tároló állapotáról kaphatunk információt, ha az FDC status-regiszterét kiolvassuk. Ez az IN X0H utasítással lehetséges. A kapott állapotbájt jelentése az előzőleg kiadott parancs függvényében a következő:

Minden I. típusú parancsnál

S7 Nem üzemkész
 S6 írás védett
 S5 Fej leengedve
 S4 Keresési hiba
 S3 CRC hiba
 S2 0. sáv
 S1 Index impulzus
 S0 BUSY

Read Address parancsnál

S7 Nem üzemkész
 S6 0
 S5 0
 S4 nemlétező rekord
 S3 CRC hiba
 S2 Adatvesztés
 S1 DRQ
 S0 BUSY

Read Sector parancsnál

S7 Nem üzemkész
 S6 0
 S5 Rekord típus
 S4 Nemlétező rekord
 S3 CRC hiba
 S2 Adatvesztés
 S1 DRQ
 S0 BUSY

Read Track parancsnál

S7 Nem üzemkész
 S6 0
 S5 0
 S4 0
 S3 0
 S2 Adatvesztés
 S1 DRQ
 S0 BUSY

Write Sector parancsnál

S7 Nem üzemkész
 S6 írásvédett
 S5 írási hiba
 S4 Nemlétező rekord
 S3 CRC hiba
 S2 Adatvesztés
 S1 DRQ
 S0 BUSY

Write Track parancsnál

S7 Nem üzemkész
 S6 írásvédett
 S5 írási hiba
 S4 0
 S3 0
 S2 Adatvesztés
 S1 DRQ
 S0 BUSY

(Amennyiben a felhasználó önálló handlert kíván írni, úgy szükséges az FD1793 részletes ismerete.)

FÜGGELÉK

I. VT-DOS rendszerváltozók

A VT-DOS-ban számos funkció működését változók vezérik. Ezek értékeit a felhasználó a VAR paranccsal, a programozó a FISH 1AH kódú funkciójával olvashatja ki, ill. módosíthatja. Több változó ki-be kapcsol egy funkciót; ilyenkor a „0” érték a bekapcsolt állapotot, a „255” a kikapcsolt állapotot jelenti.

A változók a következők.

| Sorszám dec. | Sorszám hexadec. | Név | Funkció |
|-----------------|---------------------|---------------|---|
| 0. . . 10 | 0. . . 0A | | VT-DOS tranzien programok számára hozzáférhető, pl. eredmények átadására. |
| 11. . 20 21 | 0B. . 14 15 | SER_OK | Fenntartva a VT-DOS számára A soros vonal paramétereinek változását jelzi. 0: nem változott 255: változott |
| 22 | 16 | FORM | Soros vonal adatátviteli formátum (alapesetben 238): 2. bit = 0 : 7 adatbit 1 : 8 adatbit 4. bit = 0 : paritásbit nincs 1 : paritásbit van 5. bit = 0 : páratlan paritás 1 : páros paritás 7. bit = 0 : 1 stop bit 1 : 2 stop bit |

| | | | |
|----|----|-----------------|---|
| 23 | 17 | BAUD | <p>Soros adatátviteli sebesség (alap- esetben 4).</p> <p>0 : 110 bit/s 1 : 150 bit/s 2 : 300 bit/s 3 : 600 bit/s 4 : 1200 bit/s 5 : 2400 bit/s 6 : 4800 bit/s 7 : 9600 bit/s 8 : 19200 bit/s</p> |
| 25 | 19 | RATE_KEY | <p>A lenyomva tartott billentyű au- tomatikus ismétlésének periód- ussideje 1/50 másodpercben (a- lapesetben értéke 3).</p> |
| 26 | 1A | LOCK_KEY | <p>A billentyűzet LOCK-állapota:</p> <p>1 : nagybetű LOCK 2 : SHIFT LOCK 8 : ALT LOCK</p> |
| 27 | 1B | DLY_KEY | <p>A lenyomva tartott billentyű au- tomatikus ismétlésének kezdete a lenyomáshoz képest (1/50 másodpercben; alapértéke 30)</p> |
| 28 | 1C | DSK_ERR | <p>A legutoljára végrehajtott pa- rancs által visszaadott hibakód. A VAR parancs 0-ra állítja (nem jelez hibát).</p> |
| 29 | 1D | ECHO | <p>Ha a VT-DOS parancsot olvas be egy batch-fájlból, alaphelyzet- ben nem írja ki a képernyőre. Ha az ECHO változót bekapcsoljuk, akkor a parancssorokat a beolva-</p> |

sás és a paraméterek behelyettesítése után kiírja. (Az ECHO alapértéke tehát KI.)

30

1E

CD_PROMPT Ez a változó a parancsértelmező "prompt"-ját (kijelentkező jelzést) vezérli. Induláskor kikapcsolt állapotban van, s ilyenkor a prompt az aktuális meghajtó betűjele, amelyet egy „nagyobb” jel (>) követ (pl. B>). Bekapcsolt állapotában a prompt az aktuális meghajtó és az aktuális könyvtár nevét tartalmazza (pl. B:\UTIL>). Mivel az aktuális könyvtár kiírása a diszkhez is fordul, a prompt kiírása egy kis sé tovább tarthat, ha a PROMPT változót bekapcsoljuk.

31

1F

DTFORM

A dátum és az idő kiírásának és begépelésének formátumát a DTFORM változó határozza meg.

A változó alsó 6 bitje a dátum formátumát adja meg. Három db 2-karakteres mező helyezkedik el itt, melyek mindegyike az 1, 2 vagy 3 értéket veheti fel. A mezők sorrendje megfelel a kiírási és begépelési pozícióknak; a legkisebb két bit a jobb szélső számpárhoz tartozik.

A mező „1” értéke azt jelenti, hogy az adott számpár a nap; a „2” érték a hónapot, a „3” érték az évet jelenti. Így tehát bármilyen formátumot megadhatunk. Kényelmi célokból a három leggyakoribb formátumot táblázatba is foglaltuk (l. később). Ha mindhárom mező 00, akkor az angol formátum (nap-hó-év) az érvényes. Ha nem

mindhárom mező 00, akkor a 00-ás mezőkben a képernyőn is 00 jelenik meg, begépelésnél pedig ez a mező figyelmen kívül marad.

A változó legfelső bitje vezérli az idő formátumát (12-órás vagy 24-órás). Induláskor a 12-órás formátum az érvényes; ekkor a perc után az „a” betű délelőttöt, a „p” betű délutánt jelent. Ezek a betűk a 24-órás formátumban nem szerepelnek.

Az időt bármelyik formátumban egyértelműen be lehet gépelni.

| Dátum f. \ Idp f. | 12-órás | 24 -órás |
|-------------------|---------|----------|
| Angol (nap-hó-év) | 27 | 155 |
| USA (hó-nap-év) | 39 | 167 |
| ISO (év-hó-nap) | 57 | 185 |

- 32 20 . **BOOT_DRV** Azt a meghajtót azonosítja, amelyről a COMMAND.COM betöltődött.
(1 > A.; 2 > B.; stb.)
- 33 21 **DEF_UNIT** Az implicit (default) meghajtót azonosítja (ez érvényes, ha nincs megadva meghajtó).
(1 > A.; 2 > B.; stb.)
- 34 22 **STEP_RATE** Ez a változó a diszk-meghajtó lépéssebességét vezérli. Induló értéke 6 ms, amely a legtöbb korszerű meghajtóhoz megfelel, de kisebb sebességre is állítható. Mindegyik a rendszerben levő meghajtó ugyanazt az értéket használja.

A megadható érték: 0 => 6 ms
 1 => 12 ms
 2 => 20 ms
 3 => 30 ms

| | | | |
|----|----|----------------|---|
| 35 | 23 | DSK_CHK | Ez a változó „BE” állapotban van, és azt vezérli, hogy a VT-DOS ellenőrizze-e a lemez azonosítóját, amikor nyitott fájlhoz fordul. Általában célszerű bekapcsolt állapotban hagyni, de olyan programok, amelyek csak ritkán fordulnak korábban megnyitott fájlhoz (2 másodpercenként, vagy még ritkábban), felgyorsíthatók a CHECK kikapcsolásával. |
| 37 | 25 | VERIFY | Bekapcsolt állapotában minden diszk-írás után visszaolvassa a VT-DOS az adatokat, és ellenőrzi, hogy megegyeznek-e az eredetivel. Ez lassítja a diszk-írást, ezért a rendszer indulásakor kikapcsolt állapotban van. |
| 38 | 26 | AB_ERR | A legutolsó .ABORT hibát okozó diszkhiba kódja. |
| 39 | 27 | RND_0 | A 28-bites véletlen szám első 7 bitje. (A 7. bit mindig 0.) |
| 40 | 28 | RND_1 | A 28-bites véletlen szám második 7 bitje. (A 7. bit mindig 0.) |
| 41 | 29 | RND_2 | A 28-bites véletlen szám harmadik 7 bitje. (A 7. bit mindig 0.) |

42 2A RND_3 A 28-bites véletlen szám negyedik 7 bitje. (A 7. bit mindig 0.)

II. Hibakódok

Az alábbiakban a hibakódok és (ahol van) a hozzájuk tartozó hibaüzenetek teljes listáját közöljük. A felsorolásban jelezzük, hogy mely hibák jelennek meg normál esetben abort/retry/ignore kérdéssel.

A hibaüzeneteket idézőjelbe tettük, utána zárójelben olvasható a magyar jelentésük.

| dec. | Kód hexadec. | Név | Hibaüzenet és leírás |
|-----------------------|-----------------|----------|--|
| MOPS Hibakódok | | | |
| 255 | FF | .NFUNC | Illegális funkcióhívási kód a FISH-ben. |
| 252 | FC | .PAGE3 | A BASIC CLI puffere a 3. lapra nyúlik. |
| 245 | F5 | .STOP | CTRL-ESC-t nyomtak. |
| 244 | F4 | .PARITY | Paritás-hiba a soros vonalon. |
| 243 | F3 | .FRAME | Forátumhiba a soros vonalon. |
| 242 | F2 | .OVERRUN | Karaktervesztés a soros vonalon. |
| 241 | F1 | .NCARD | Nincs soros vonali kártya a gépben. |
| 236 | EC | .EOF | „End of file” (fájl vége). Olvasásnál váratlanul elértük a fájl végét. |

| | | | |
|--------------------------|----|---------|---|
| 233 | E9 | .NOTOP | Nincs megnyitva a fájl az MDD-n (ugyanaz, mint a MOPS kazettakezelő 233-as hibája). |
| 232 | E8 | .VERIFY | „Verify error” (hiba az ellenőrzésnél). Kazetta/diszk ellenőrzési hiba. |
| 231 | E7 | .INTERR | Érvénytelen .CAS fájl-fejléc az MDD-n. |
| 229 | E5 | .IPTR | Érvénytelen fájl-mutató váltás az MDD-n (ugyanaz, mint a MOPS kazettakezelő „hibás blokk” hibajelzése). |
| Diszkes hibakódok | | | |
| 191 | BF | .ICMD | Belső hiba, rendes körülmények között nem fordulhat elő. (Érvénytelen parancsot jelent az egység-kezelőnek). |
| 190 | BE | .IUNUM | Belső hiba, rendes körülmények között nem fordulhat elő. (Érvénytelen egység az egységkezelőnek.) |
| 189 | BD | .ISECT | Belső hiba, rendes körülmények között nem fordulhat elő. (Érvénytelen szektor.) |
| 188 | BC | .NRDY | „Not ready” (nem üzemkész) A diszkmeghajtó nem válaszol. Általában azt jelenti, hogy nincs lemez a meghajtóban (ABORT/-RETRY kezelésű hiba). |

| | | | |
|-----|----|----------------|--|
| 186 | BA | .VERIFY | „Verify error” (hiba az ellenőrzésnél). Csak akkor fordul elő, ha a VERIFY jelző bekapcsolt állapotban van. Azt jelenti, hogy a lemeztől visszaolvasott adat nem egyezik meg a felírttal. (ABORT/RETRY hiba) |
| 185 | B9 | .DATA | „Data error” (adathiba). A kívánt szektor nem olvasható be a lemeztől. Ez általában azt jelenti, hogy a lemez megsérült (ABORT/RETRY hiba). |
| 184 | B8 | .RNF | „Sector not found” (a szektor nem található). A kívánt szektor nem található a diszken. Ez általában azt jelenti, hogy a lemez megsérült. (ABORT/RETRY hiba) |
| 183 | B7 | .WPROT | „Write protected disk” (írásvédett lemez). Olyan lemezre próbáltunk írni, amelyen rajta van az írásvédő címke. (ABORT/RETRY hiba) |
| 182 | B6 | .UFORM | „Unformatted disk” (formátlan lemez). Teljesen formátlan lemezre próbáltunk írni. (ABORT/RETRY hiba) |
| 181 | B5 | .NDOS | „Not an DOS disk” (nem DOS lemez). A meghajtóban levő lemez formattált, de nem kompatibilis a VT-DOS-sal, azaz nem MS-DOS típusú lemez. (ABORT/RETRY hiba) |

| | | | |
|-----|----|--------|--|
| 180 | B4 | .WDISK | „Wrong disk” (rossz lemez). A meghajtóban kicserélték a lemezt, miközben a FISH diszk-művelet közben volt. (ABORT/RETRY hiba) |
| 179 | B3 | .WFILE | „Wrong disk for file” (rossz lemez a fájl számára). Nagyon hasonló a .WDISK-hez, de azt jelenti, hogy kicserélték a lemezt, miközben a fájl nyitva volt (ABORT/RETRY hiba) |
| 178 | B2 | .WDRV | Ezt használjuk az „insert diszk for drive X:” (helyezzen be lemezt az X: meghajtóba) üzenet generálásra, ha a MAPDISK használatban van; a felhasználó sohasem kapja meg. |
| 177 | B1 | .NCOMP | „Incompatible disk” (nem kompatibilis lemez). Érvényes DOS-lemezt olyan meghajtóval akarunk használni, amely nem képes azt olvasni. Ez előfordulhat, ha pl. kétoldalas lemezt teszünk az egyoldalas meghajtóba, vagy 80 pályás lemezt 40 pályás meghajtóba. (ABORT/RETRY hiba) |
| 176 | B0 | .IFAT | Egy, a fájl-elhelyezési táblából beolvasott érték kívül esik a lemez címtartományán. Rendszerint sérült lemez esetén fordul elő különösen, ha a FAT felülíródott. (ABORT/RETRY kezelési hiba, bár a RETRY sohasem fog működni). |

FISH hibakódok

| | | | |
|-----|----|--------|---|
| 175 | AF | .NOUPB | Belső hiba, rendes körülmények között nem fordulhat elő (nincs UPB). |
| 174 | AE | .ICALL | Érvénytelen FISH funkciószám. |
| 173 | AD | .ABORT | „Disk operation aborted” (a diszkművelet félbeszakadt). Az ABORT/RETRY, vagy ABORT/RETRY/IGNORE kérdésre az operátor ABORT választ adott |
| 172 | AC | .IFCB | A FISH-nek átadott FCB érvénytelen volt. Az FCB első bájtja adja meg az FCB típusát, a FISH ezt a bájtot ellenőrzi. |
| 171 | AB | .IDVR | „Invalid drive” (érvénytelen meghajtó). Nem létező meghajtóhoz próbáltunk fordulni. |
| 170 | AA | .IPATH | „Invalid pathname string” (érvénytelen útvonalnév-string). Az útvonalnév valamilyen hibájából ered, pl. többértelmű elem, túl hosszú útvonalnév, vagy hamis karakterek az útvonalnév végén. |
| 169 | A9 | .IFNM | „Invalid filename” (érvénytelen fájlnev). Érvénytelen fájlnevet kíséreltünk meg létrehozni. Aból származhat, hogy valamilyen érvénytelen karakter került a FISH-nek átadott FCB-ben levő |

fájlnévbe, vagy abból, hogy az „átszervezés” funkció szóközöket ágyazna be a fájlnevbe.

| | | | |
|-----|----|---------------|--|
| 168 | A8 | .IATTR | „Invalid file attributes” (érvénytelen fájl-attribútumok). Fájlként hivatkozhatunk al-könyvtárra, vagy fordítva. |
| 167 | A7 | .DOT | „Invalid . or . . operation” (érvénytelen . vagy . . művelet). Valami illegális műveletet próbáltunk végezni az al-könyvtár speciális . vagy . . bejegyzésén pl. átnevezést, vagy áthelyezést. |
| 166 | A6 | .OV64K | Megkíséreltünk a 64k-s területen túlnyúló mennyiségű adatot írni, vagy olvasni. |
| 165 | A5 | .FILE | A fájl alapegység-lánca túl rövid a könyvtár-bejegyzésben megadott fájl-mérethez. Általában azt jelentí, hogy a FAT megsérült. |
| 164 | A4 | .DRFUL | „Root directory full” (a kiindulási könyvtár megtelt.) Csak a kiindulási könyvtár képes megtelni, az alkönyvtárak szükség szerint bővülnek. |
| 163 | A3 | .DKFUL | „Disk full” (a lemez megtelt). |
| 162 | A2 | .DUPF | „Duplicate filename” (kétszeres fájlnev). Átnevezés, vagy áthelyezés funkcióban fordulhat elő, |

ha két azonos nevű fájlnev keletkezne egy könyvtárban.

| | | | |
|-----|----|---------------|---|
| 161 | A1 | .NOFIL | „file not found” (a fájl nem található. A megadott fájl nem létezik a könyvtárban. Ebben az összefüggésben a „fájl” könyvtárat is jelenthet. |
| 160 | A0 | .NODIR | „Directory not found” (a könyvtár nem található). Az útvonalnév-stringben definiált egyik alkönyvtár nem létezik. |
| 159 | 9F | .DIRE | „Invalid directory move” (érvénytelen könyvtár-áthelyezés). Egy alkönyvtárat megkíséreltünk áthelyezni egy saját „leszármazott” alkönyvtárjába. |
| 158 | 9E | .DIRNE | „Directory not empty” (a könyvtár nem üres). Meg akartunk szüntetni egy alkönyvtárat, amely nem üres. |
| 157 | 9D | .SYSX | „System file exists” (rendszer-fájl létezik). Fájl vagy könyvtárt kíséreltünk meg létrehozni egy létező rendszer-fájl nevével. |
| 156 | 9C | .DIRX | „Directory exists” (könyvtár létezik). Létező könyvtár nevével próbáltunk meg alkönyvtárat létrehozni. |
| 155 | 9B | .FILEX | „File exists” (fájl létezik). Létező fájl nevével próbáltunk fájl vagy könyvtárat létrehozni. |

| | | | |
|----------------------|----|----------------|--|
| 154 | 9A | .FILRO | „Read only file” (csak olvasható fájl). Olyan fájlt próbáltunk törölni, vagy írni, amelynek a „csak olvasható” attributum-bit-je „1”-be van állítva. |
| 153 | 99 | .VAR | „Invalid variable number” (érvénytelen változószám). Nemlétező változóra hivatkoztunk. |
| 152 | 98 | .IMAP | „Invalid MAPDISK” (érvénytelen MAPDISK). Nemlétező fizikai egység számra hivatkoztunk a MAPDISK funkcióban. |
| 151 | 97 | .IDATE | „Invalid date” (érvénytelen dátum). Hibás dátumot adtunk meg. |
| 150 | 96 | .ITIME | „Invalid time” (érvénytelen idő). Hibás idő adatot adtunk meg. |
| CLI hibakódok | | | |
| 149 | 95 | .OKCMD | A paráncs végrehajtása rendben befejeződött (belső jelzés, a felhasználó nem kapja meg). |
| 148 | 94 | .BATEND | A batch-fájl véget ért (belső jelzés, a felhasználó nem kapja meg). |
| 147 | 93 | .BADCOM | Érvénytelen COMMAND.COM (belső hiba, a felhasználó nem kapja meg). |

| | | | |
|-----|----|----------------|--|
| 146 | 92 | .CTRLC | „CTRL–C pressed” (CTRL–C-t nyomtak). A parancsot CTRL–C lenyomásával megszakították. |
| 145 | 91 | .BADCMD | „Unrecognised command” (ismeretlen parancs). Az adott parancs nem tartozik a beépített parancsok közé, és .COM, vagy .BAT fájlként sem található meg. |
| 144 | 90 | .BUFUL | „Command too long” (túl hosszú parancs). A parancs hosszabb volt, mint a maximálisan megengedett 127 karakter. |
| 143 | 8F | .BADNO | „Invalid number” (érvénytelen szám). Az adott szám nagyobb, mint 255, vagy nem számjegy karakter található benne. |
| 142 | 8E | .IPARM | „Invalid parameter” (érvénytelen paraméter). A paraméternek nincs értelme. |
| 141 | 8D | .INP | „Invalid number of parameters” (érvénytelen számú paraméter). Vagy túl kevés paramétert adtunk át, vagy az összes paraméter dekódolása után még maradtak karakterek. |
| 140 | 8C | .NORAM | „Not enough memory” (nincs elég memória). Nincs elegendő szabad hely a memóriában a parancs számára |

| | | | |
|-----|----|--------|--|
| 139 | 8B | .NCART | „No VT-DOS cartridge” (nincs VT-DOS programmodul). A BASIC CLI-ből kiadtuk a DOS parancsot, de nincs a gépben a VT-DOS programmodul. |
| 138 | 8A | .ICOPY | „File cannot be copied onto itself” (a fájl nem másolható önmagára). Olyan COPY parancsot adtunk ki, amellyel a cél-fájl felülírná a forrás-fájlt. |
| 137 | 89 | .IPSTR | „Invalid PATH string” (érvénytelen útvonal-string). A PATH parancsnak átadott valamelyik útvonalnév érvénytelen |
| 136 | 88 | .NOHLP | „File for HELP not found” (a HELP fájl nem található. A HELP parancs paraméterének megfelelő .HLP fájl nem található annak a lemeznek a \HELP könyvtárában, amelyről a rendszer betöltődött. |
| 135 | 87 | .IDEV | „Invalid device use” (érvénytelen eszközhasználat). A COPY parancsban megadott I/O eszköz nem használható az adott módon (pl. a nyomtató inputra), vagy a MODE parancsban megadott eszköz nem átirányítható. |

III. A parancsok áttekintése

CD [d:] [path]

Megjeleníti vagy megváltoztatja az aktuális könyvtárt.

CHDIR [d:] [path]

Megjeleníti vagy megváltoztatja az aktuális könyvtárt.

CLS

Törli a képernyőt.

COPY forrás [/A] [/H] [cél [/A] [/T]]

Adatokat másol egy fájlból vagy egy készülékről egy másik fájlba vagy készülékre.

DATE [dátum]

Megjeleníti vagy beállítja az aktuális dátumot.

DEL filespec [/H]

Fájlok törlése a lemezről.

DIR [d:] [path] [filename] [/H] [/W] [/T] [/S]

A lemezen levő fájlok nevét megjeleníti.

DOS

Áttérés a BASIC-ből a VT-DOS rendszerbe.

ERA filespec [/H]

Fájlok törlése a lemezeiről.

ERASE filespec [/H]

Fájlok törlése a lemezeiről.

FORMAT [d:] [volname] [/1] [/H] [/8]

Lemez formattálása

HELP [parancs]

Felsorolja a BASIC CLI parancsait.

LDIR [d:] [path] [filename] [/H] [/W] [/T] [/S]

A lemezen levő file-ok nevét kinyomtatja.

LTYPE filespec [/H]

A file tartalmát kinyomtatja.

MD [d:] path

Új alkönyvtár létrehozása

MKDR [d:] path

Új alkönyvtár létrehozása.

MOVE filespec [/H] [path]

Fájlok áthelyezése egyik könyvtárból a másikba.

RD [d:] path [/H]

Egy vagy több alkönyvtár törlése.

REN filespec [/H] filename

Egy vagy több fájl átnevezése.

RENAME filespec [/H] filename

Egy vagy több fájl átnevezése.

RMDIR [d:] path [/H]

Egy vagy több alkönyvtár törlése.

RNDIR filespec [/H] filename

Egy vagy több alkönyvtár átnevezése.

TIME [idő]

Megjeleníti, vagy beállítja az aktuális időt.

TYPE device | filespec [/H]

Adatokat jelenít meg egy fájlból, vagy egy I/O-készülekről.

VAR number [number | ON | OFF]

Megjeleníti vagy beállítja egy VT-DOS változó értékét.

VOL [d:] [filename]

Megjeleníti, vagy megváltoztatja a lemez kötetnevét.

VIDEOTON

ELEKTRONIKAI VÁLLALAT
SZÁMÍTÁSTECHNIKAI GYÁRA