PROPOS–16
OPERÁCIÓS RENDSZER
(3.0 VÁLTOZAT)

Kezelési útmutató

SZÁMÍTÁSTECHNIKAI KUTATÓ INTÉZET
ÉS INNOVÁCIÓS KÖZPONT

PROPOS–16 operációs rendszer (3.0 változat)

Kezelési útmutató

2. kiadás

Budapest, 1986

© SZÁMÍTÁSTECHNIKAI KUTATÓ INTÉZET ÉS INNOVÁCIÓS KÖZPONT, 1986

Jelen kiadvány az SZKI előzetes írásbeli engedélye nélkül
sem teljes terjedelmében, sem részleteiben semmilyen módon,
illetve eszközzel nem másolható, információ-visszakereső
rendszerben nem tárolható, nem mikrofilmezhető.

Hitvallásom: nem lenni kevesebb, mint látszom;
híven szolgálni azt, ki bizalmába fogad;
szeretni, aki becsületes; társalogni avval,
aki bölcs és keveset beszél. . .

(W. SHAKESPEARE: LEAR KIRÁLY)

Tartalomjegyzék

1. Bevezetés 1–1

2. A PROPOS–16 operációs rendszer jellemzői 2–1

2.1. A PROPER–16 számítógép 2–1
2.2. Mágneslemezes tárolás 2–3
2.3. File-azonosítók 2–4
2.4. A PROPOS–16 betöltése és elindítása 2–5

3. Az operációs rendszer parancsai 3–1

3.1. A parancsok megadásának módja 3–1
3.2. Jelölések 3–2
3.3. Lemezkezelés 3–3
3.4. Könyvtárkezelés 3–6
3.5. File-kezelés 3–7
3.6. Perifériakezelés, konfigurálás 3–10
3.7. Batch parancsok 3–12
3.8. Bemenet/kimenet átirányítása 3–13
3.9. Parancsláncolás 3–15

4. A parancsok hasznalata 4–1

ASSIGN (lemez-hozzárendelés) 4–3
BACKUP (mentés) 4–4
Batch parancsok 4–7
ECHO (parancskiírás letiltása) 4–9
FOR (ismételt végrehajtás) 4–11
GOTO (feltétel nélküli elágazás) 4–12
IF (feltételes parancs) 4–13
PAUSE (futás felfüggesztése) 4–15
REM (üzenet kiírása) 4–16
SHIFT (paraméterek átsorszámozása) 4–17
CHDIR (aktuális könyvtár kijelölése) 4–18
CHMOD (file-attributum módosítása) 4–19
CMP (file-összehasonlítás) 4–21
COMM (szövegfile-összehasonlítás) 4–23
CONFIG (konfiguráció lekérdezése, paraméterek módosítása) 4–25
CONVEXE (programformátum átalakítása) 4–29
COPY (file-másolás, egyesítés) 4–30
COUNT (sor-, szó- és karakterszámlálás) 4–36
CTTY (konzol kijelölése) 4–38
DEL (file-törlése) 4–39
DIFF (file-ok eltérésének kiíratása) 4–41
DIR (könyvtár katalógusának kiíratása) 4–43
DISKCOPY (hajlékony lemez másolása) 4–46
FIND (szövegrész megkeresése szövegfile-ban) 4–48
FORMAT (lemezformálás) 4–50
FSYS (könyvtárfa kiíratása) 4–56

HD (hexadecimális kiíratás) 4–57
IMP (PROPOS–8/PROPOS–16 file-konverzió) 4–60
MENUSYS (menü/help-file módosítása) 4–62
MKDIR (alkönyvtár létrehozása) 4–65
MODE (perifériák üzemmódjának módosítása) 4–67
MORE (kiíratás laponként) 4–70
PATH (keresési utak kijelölése) 4–71
PRINT (szövegfile kinyomtatása) 4–72
RECOVER (hibás lemez javítása) 4–75
RENAME (file átnevezése) 4–76
RESTORE (elmentett file-ok visszatöltése) 4–77
RMDIR (alkönyvtár megszüntetése) 4–80
SET (rendszerparaméterek kiíratása, módosítása) 4–81
SORT (szövegfile rendezése) 4–84
STAT (lemez ellenőrzése, méretadatok kiíratása) 4–87
TYPE (szövegfile kiíratása képernyőre) 4–91
VOL (lemezazonosító címke kiíratása) 4–92

5. Az EDIT sorszerkesztő program 5–1

5.1. Az EDIT program elindítása 5–1
5.2. Az EDIT parancsok paraméterezése 5–3
Sor javítása 5–6
A (sor beolvasása) 5–9
C (sor másolása) 5–10
D (sor törlése) 5–11
E (EDIT befejezése és mentés) 5–12
I (sor beszúrása) 5–13
L (kiírás) 5–14
M (sor áthelyezése) 5–15
P (kiírás laponként) 5–16
Q (EDIT befejezése) 5–17
R (szöveg helyettesítése) 5–18
S (szöveg keresése) 5–20
T (file-ok egyesítése) 5–21
W (sor visszaírása) 5–22

6. Üzenetek és hibajelzések 6–1

Függelékek F–1

A. Foglalt file-nevek, szabványos file-név-kiterjesztések F–3
B. Billentyűzet F–5
C. ASCII kódtáblázat F–10
D. A PROPOS–16 parancskészlete F–13

Tárgymutató T–1

1. Bevezetés

A PROPOS–16 (*PRO*fessional *P*ersonal Computer *O*perating *S*ystem) a PROPER–16 személyi számítógépek operációs rendszere. Az új, 3.0. változat felülről kompatibilis a PROPOS–16 korábbi változataival és az MS–DOS 2.0. operációs rendszerrel. Ez azt jelenti, hogy minden olyan program, amely ezen operációs rendszerek mellett működött, változtatás nélkül használható a PROPOS–16 3.0. változattal is.

A PROPOS–16 alapüzemmódban egyfelhasználós alkalmazást tesz lehetővé, de számos eszközzel támogatja azt, hogy egy számítógépen – nem egy időben – több, egymástól független alkalmazási rendszer működhessen.

Az operációs rendszer legfontosabb új szolgáltatásai:

- a különböző PROPER–16 modellek egységes kiszolgálása ,
- egyszerűbb kezelést biztosító, menüvezérelt üzemmód,
- tájékoztató információk kérésének lehetősége (HELP),
- fastruktúrájú könyvtárak kezelése,
- újabb szövegkezelési szolgáltatások,
- parancsláncolás, be- és kimenet átirányíthatósága,
- szélesebb körű grafikus lehetőségek,
- más operációs rendszerek (CP/M, PROPOS–8, ...) alatt készített file-ok konvertálása PROPOS–16 formátumra, vagy fordítva,
- fix lemezes háttértár kezelése,
- központi tárban szimulált (virtuális) lemez,
- speciális perifériák kezelésének egyszerű megoldási lehetősége (ún. device driver koncepció),
- bővítési lehetőségek többfelhasználós (PROMOS) és lokális hálózati (PROPNET) alkalmazásokhoz.

Végül, de talán nem utolsósorban a PROPOS–16 3.0. változat újdonságának tekinthető az eddiginél részletesebb – az operációs rendszer megismerését jobban segitő – kézikönyvsorozat is.

Az operációs rendszert három kötetben írjuk le, a géppel különböző szinten foglalkozó, ill. különböző számítástechnikai felkészültségű szakemberek számára.

Az első kötet, a **PROPOS–16 operációs rendszer – Alapismeretek**, elsősorban azokhoz szól, akik először kerülnek közvetlen kapcsolatba számítógéppel.

Ezen második kötet, a **PROPOS–16 operációs rendszer – Kezelési útmutató** olvasóiról feltételezünk némi számítástechnikai alapismeretet. Ez azonban nemcsak első kötetünkből sajátítható el, más számítógép mellett szerzett gyakorlat alapján is megérthető e **Kezelési útmutató**. A PROPER–16 számítógép és a PROPOS–16 operációs rendszer alapfogalmainak ismertetése megtalálható a 2. fejezetben.

A 3. fejezet célja a PROPOS–16-tal megoldható feladatok körének összefoglalása. Itt a parancsokat *funkciójuk szerinti csoportosításban*, röviden mutatjuk be.

A *rutinszerű munkavégzés* során már inkább a 4. fejezet forgatását ajánljuk, ahol a parancsok részletes magyarázata és számos példa található. E fejezetben *betűrendben* ismertetjük a PROPOS–16 parancsokat.

Az 5. fejezetben az EDIT szövegszerkesztő program használatát találhatjuk.

A 6. fejezetben a PROPOS–16 rendszer üzeneteit és hibajelzéseit foglaltuk össze betűrendben, magyarázatokkal, és az üzenet megjelenítését követő teendők leírásával.

A függelékekben összefoglaló táblázatok találhatók.

A harmadik kötet a **PROPOS–16 operációs rendszer – Programozóknak**. Annak ajánljuk, aki az operációs rendszer működését részleteiben is szeretné megismerni, és saját programokat is fejleszt a gépen. Megismerhetők a kötetből az operációs rendszer programból hívható moduljai, a legfontosabb rendszertáblázatok felépítése, a periféria- és a file-kezelés apparátusa stb. Az operációs rendszer parancskészletét többféle csoportosításban foglaljuk össze, és számos táblázattal tesszük áttekinthetővé.

Reméljük, hogy örömét leli majd a PROPER–16 számítógép megismerésében, és munkáját eredményesebben tudja segítségével elvégezni. Kérjük, hogy ha a gép szolgáltatásaival vagy a kézikönyvek tartalmával kapcsolatban észrevétele van, jelezze a kötet végére befűzött válaszlapokon.

A PROPOS_16

2. A PROPOS–16 operációs rendszer jellemzői

Az operációs rendszer olyan programok együttese, amelyek lehetővé teszik a számítógép részegységeinek (perifériáinak) és más erőforrásainak kezelését anélkül, hogy fizikai működésük részleteit meg kellene ismernünk. A felhasználó csak az operációs rendszeren keresztül működtetheti a számítógépet, és nem kell tudnia, hogy hol és milyen feladatmegosztás van hardware és software között.

A PROPOS–16 operációs rendszer azonos módon kezeli a részét képező külső programmodulokat és a felhasználói (feldolgozó) programokat. A lemezen tárolt programok bármelyikét egyszerűen a nevének a megadásával indíthatjuk el. Ezt úgy is értelmezhetjük, hogy az operációs rendszer parancsainak köre minden új programmal bővül.

A számítógéppel végzett munka általában a gépnek adott parancsok megfelelő sorozatát jelenti. A számítógép minden egyes parancs hatására elvégez valamilyen feladatot, erről jelzést ad, majd az alapállapotba visszatérve készen áll újabb parancs fogadására.

A gyakran használt parancssorozatokat célszerű ún. BATCH file-okban tárolni, így megtakarítható a parancsok beírásával járó fáradság és elkerülhetők az elütések. Elegendő a BATCH file nevét megadni, és a parancssorozat értelmezése, a parancsok végrehajtása már az operációs rendszer feladata.

Mielőtt az egyes parancsok megadási módját részletesen tárgyalnánk, vegyük sorra azokat a hardware és software fogalmakat, amelyek ismerete nélkül a gép kezelése nem sajátítható el.

2.1. A PROPER–16 számítógép

A számítógép a következő nagyobb egységekből áll:

- központi egység,
- billentyűzet,
- képernyő,
- fix mágneslemezes tároló (opcionális),
- hajlékony (floppy) lemezes tároló.

A PROPER–16 személyi számítógép konfigurációja szükség esetén további berendezésekkel bővíthető:

- központi tár bővítése,
- aritmetikai modul,
- nyomtató,
- rajzgép (plotter),
- gyors kazettás háttértár (streamer),

- adatátviteli csatoló, hálózati adapter,
- terminálok.

A következőkben röviden ismertetjük ezen berendezések feladatát, használatukat és helyes kezelésüket. További részletek az üzemeltetői kézikönyvben találhatók.

A 16-bites, 8088 típusú mikroprocesszorra épülő *központi egység* a számítógép „lelke", amely a munka zömét végzi. A PROPER–16 központi egysége tartalmazza a *központi tárat* (memóriát) is, amelynek kapacitása 128–704 Kbyte lehet.

A számítógép és a felhasználó kapcsolatában a *billentyűzet* a parancsok és adatok bevitelének eszköze, a *standard input* periféria. A szokásos alfanumerikus billentyűk mellett külön numerikus billentyűcsoport (egyben kurzormozgató billentyűk) és tíz funkcionális billentyű is található. A billentyűk ismertetése a B. függelékben található. A beírt parancs- vagy adatsorokat mindig az [ENTER] lenyomásával kell lezárni.

A színes vagy egyszínű (monokróm) *képernyő*, a *standard output* periféria, az eredmények megjelenítésének eszköze, 25 sorban, soronként 80 keskeny vagy 40 széles karaktert képes megjeleníteni (l. MODE parancs). Grafikus üzemmódban 200 pontsorban, soronként 320 színes vagy 640 egyszínű képpont ábrázolható.

A PROPER–16 számítógépbe beépített *fix (merev) mágneslemez* (Winchester disk) nem cserélhető, kapacitása típustól függően kb. 10 Mbyte vagy 27 Mbyte lehet. A fix lemez – ha van a konfigurációban – a programok és adatok tárolásának legfontosabb eszköze. A lemez „elején" találhatók a PROPOS–16 operációs rendszer 3.0. változatának programmoduljai, amelyek nélkül a számítógép csak igen korlátozottan használható.

A *hajlékony (floppy) lemez*es meghajtók a kompatibilis számítógépek közötti program- és adatcserét, tartalék másolatok készítését (l. BACKUP parancs), ritkábban használt programok és adatok tárolását teszik lehetővé. Az 5,25 inch (13,3 cm) átmérőjű lemezeken tárolható adatmennyiség (Kbyte-ban) a meghajtó típusától függően:

	egyoldalas lemez	kétoldalas lemez
40 sávos meghajtó	184	368
80 sávos meghajtó	–	736

A számítógéphez szükség esetén a beépített két meghajtón kívül további két hajlékony lemezes meghajtó csatlakoztatható. Ezek vagy a fenti típusúak, vagy 8 inch (20,5 cm) átmérőjű lemezekkel működő egységek lehetnek.

A lemezt mindig cimkével fölfelé, a cimkés oldalánál fogva tegyük a meghajtóba, majd nyomjuk be a rögzítőgombot.

A PROPOS–16 operációs rendszer új szolgáltatása, hogy elegendően nagy (256–512 Kbyte) központitár-kapacitás esetén egy vagy két, a központi tárban szimulált lemezt is használhatunk. Ennek az ún. *virtuális (RAM)* lemeznek a kapacitása paraméterezhető, és igen gyors file-műveleteket tesz lehetővé (l. CONFIG parancs).

A mágneslemez-meghajtókat és a virtuális lemezt az operációs rendszer az A, B, C,... betűkkel azonosítja. Az első kettő mindig a beépített két floppy meghajtó, amelyek közül a felsőt az A, az alsót a B betű jelöli.

A további betűk kiosztásánál először az esetleges kiegészítő floppy meghajtók, majd a fix lemezes egységek következnek. Ezután kapják betűjelüket a virtuális lemezek.

Floppy lemezes meghajtók				Fix lemezes meghajtók		Virtuális lemezek*	
1.	2.	3.	4.	1.	2.	1.	2.
A,B						C	D
A	B					C	D
A,B				C		D	E
A	B			C		D	E
A	B	C	D			E	F
A	B			C	D	E	F
A	B	C	D	E		F	G
A	B	C	D	E	F	G	H

* Csak ha definiáljuk!

A PROPER–16 számítógéphez többféle típusú *nyomtató* is csatlakoztatható, amelyeken lista vagy hard copy készíthető. Ékezetes kis- és nagybetűk nyomtatására alkalmas készülékkel levélminőségű dokumentumok készíthetők. A mátrixnyomtatók általában a grafikus jelek kiíratására is alkalmasak.

2.2. Mágneslemezes tárolás

A PROPER–16 számítógép programjait és adatállományait a fix és a hajlékony lemezeken, rendezetten tároljuk. Az összetartozó adatokat, ill. egy program utasításait egy-egy *file* tartalmazza. Az egymással logikai kapcsolatban levő file-okat *könyvtárak*ban tároljuk, és több könyvtár együttese adja egy *lemez* teljes tartalmát. A file-okon belüli adat- vagy programsorokat *rekord*nak nevezzük. Egy rekord több *mező*ből állhat, amelyek tartalma egy karaktersorozat.

A PROPOS–16 rendszer egy meghajtót mindig kitüntetetten kezel. Ha ehhez fordulunk, nem kell meghajtót specifikálnunk. Az operációs rendszer betöltése után a kitüntetett, a *feltételezett meghajtó* az, ahonnan a rendszert betöltöttük (ez többnyire az első fix lemez, de lehet az első floppy meghajtó is).

Új feltételezett meghajtó a meghajtó betűjelének és egy kettőspontnak a beütésével jelölhető ki, pl.:

B: [ENTER]

A program- és adatfile-okat a PROPOS–16 rendszer könyvtárakba szervezve tárolja, illetve tartja nyilván. Minden fizikailag különálló lemezen (a fix lemezeken, illetve a számos cserélhető floppy lemezen) egy vagy több, *névvel ellátott könyvtár* lehet. Az egy lemezen levő könyvtárak fastruktúra szerint kapcsolódnak egymáshoz. Minden lemez „elején" található egy ún. *gyökérkönyvtár*, ez alá rendelhetők az *alkönyvtárak*.

A gyökérkönyvtárat a lemez megformálásakor (l. a FORMAT parancsot), fix mérettel hozzuk létre, ezért ebbe csak bizonyos számú bejegyzés írható (egyoldalas hajlékony lemezen 64, kétoldalason 112, fix lemezen 512). Az alkönyvtárak bejegyzéseinek száma elvben nem korlátozott, gyakorlatban a lemez kapacitása jelenti a felső határt.

A könyvtárak bejegyzései vonatkozhatnak adat- vagy programfile-okra és további alkönyvtárakra.

A gyökérkönyvtárnak nincs neve; az alkönyvtáraknak ugyanolyan szabályok szerint kell nevet adni, mint bármely file-nak (l. 2.3. fejezet).

A könyvtárak ilyen rendszerét legcélszerűbben arra használhatjuk, hogy a különböző alkalmazói területek program- és adatfile-jait egymástól elkülönítsük. A könyvtárak neve is utalhat tartalmukra. Ezzel a szervezéssel egy-egy file kikeresése gyorsabbá válik, és némi védelmet is biztosítunk számukra.

A feltételezett meghajtóhoz hasonlóan a PROPOS–16 operációs rendszer *minden meghajtón* egy-egy könyvtárat is alapértelmezésként tart nyilván, ezt nevezzük az *aktuális könyvtár*nak. Ha ezen belül keresünk vagy hozunk létre egy file-t, a könyvtár nevét nem kell előtte megadnunk.

Az operációs rendszer betöltése után az aktuális könyvtár minden meghajtón a gyökérkönyvtár. A hatékony munkavégzés érdekében minden meghajtón kijelölhetünk egy aktuális könyvtárat (l. a CHDIR parancs leírásánál).

Összefoglalva: egy teljes file-hivatkozás mindig a következő részekből áll:

- meghajtó azonosítója (nem kell kiírni, ha a feltételezett meghajtó),
- könyvtárkeresési út (nem kell kiírni, ha az aktuális könyvtár; l. a 3.2. fejezetben),
- file-azonosító (file-név és opcionális kiterjesztés; l. 2.3. fejezet).

A file-ok között az operációs rendszer nem tesz semmiféle megkülönböztetést. Az operációs rendszerhez tartozó programok, a nyelvi rendszerek, valamint egyes alkalmazói programcsomagok általában a file-név kiterjesztésében különböztetik meg speciálisan kezelt file-jaikat (l. A. függelék).

2.3. File-azonosítók

A PROPOS–16 rendszerben minden file-t és alkönyvtárat azonosítóval kell ellátni. Egy könyvtáron belül egy azonosító csak egyszer szerepelhet, ennek alapján tudja a rendszer a file-t vagy alkönyvtárat megkeresni (*egyértelmű file-azonosító*). A jól megválasztott név utalhat a file vagy alkönyvtár tartalmára.

A PROPOS–16 rendszerben az azonosítók felépítése a következő:

file-név.kiterjesztés

ahol az elsődleges *file-név* 1–8 karakter hosszú, a ponttal elválasztott (opcionális) *kiterjesztés* pedig legfeljebb 3 karakteres lehet, az alábbi karakterkészletből:

- betűk (a kis- és nagybetűk felcserélhetők, mivel a rendszer minden kisbetűt nagybetűre alakít át),
- számok (0–9),
- a következő speciális karakterek:
  $ # & @ - _ ! % ( ) '

Az azonosítón belül betűköz (space) nem írható. Ha 8 karakternél rövidebb elsődleges nevet, illetve 3 karakternél rövidebb kiterjesztést adunk meg, az operációs rendszer jobbról betűközzel egészíti ki azt.

A pont (.) a file-név és a kiterjesztés elválasztására szolgál.
Az alkönyvtárak elnevezése is ugyanilyen felépítésű lehet.

Példák:

BEVETEL.FEB
FILE12
EXPORT.85
PROG5.ASM
KONYVTAR

A PROPOS–16 operációs rendszer a mágneslemezes file-okat és a karakteres perifériákat hasonló módon kezeli. E célból a standard logikai *periférianeveket* nyilvántartja (l. az A. függeléket), és ezeket a karaktersorozatokat nem engedi mágneslemezes file-névként használni.

Pl.
COPY CON SZOVEG

file-másoló parancs segítségével a billentyűzetről a SZOVEG nevű file-ba írhatunk, míg a

COPY SZOVEG LPT1

parancs a file tartalmát a nyomtatóra „másolja".

Olyan esetekben, amikor egy file pontos nevét nem tudjuk, vagy egy PROPOS–16 paranccsal több file-on akarunk valamilyen műveletet elvégezni, a rendszer lehetőséget ad (nem egyértelmű) *file-csoport*-azonosítók használatára is. Ezek két speciális karaktert tartalmazhatnak:

? : az adott pozíción tetszőleges karakter állhat,
* : az adott pozíciótól kezdődően a file-név (illetve kiterjesztés) végéig tetszőleges karakterek állhatnak.

Néhány példa segíthet a megértésben:

file-csoport-azonosítók	*megfelelő egyértelmű file-azonosítók*
FILE*.DAT	FILE1.DAT FILE37.DAT FILEX.DAT
EXPORT.*	EXPORT EXPORT.BAS EXPORT.84
??PORT.84	EXPORT.84 IMPORT.84
*.DAT	összes .DAT kiterjesztésű file
.	összes file

2.4. A PROPOS–16 betöltése és elindítása

A *bekapcsolás*t mindig a perifériákkal kezdjük (nyomtató, képernyő stb.), és utolsónak kapcsoljuk be a központi egységet, a vele egy dobozban levő lemezmeghajtókkal együtt. A *kikapcsolás* fordított sorrendben történik.

Bekapcsolás után a számítógép automatikusan leteszteli a hozzákapcsolt perifériákat és felméri a központi tár méretét. Ez kb. 1 percet igényel, amely idő alatt a gép még nem használható.

Ezután automatikusan betöltődik a PROPOS–16 operációs rendszer. A töltést a gép először az *A* (felső) floppy meghajtóról kíséreli meg. Ha ide nincs lemez betéve, akkor az első fix lemezhez fordul, és onnan tölti be a rendszert. A jelzőlámpák felvillanása, illetve a lemezmeghajtó halk zümmögése jelzi ezt a folyamatot.

Az operációs rendszer a számítógép kikapcsolása nélkül is újból betölthető és elindítható az [ALT] [CTRL] és [DEL] billentyűk egyidejű lenyomásával.

Az operációs rendszer üzemkész állapotát a parancsválasztékot ismertető menű megjelenése jelzi.

```
SZKI        PROPOS-16 OPERACIOS RENDSZER 3.0 valtozat        1985.03.30

                    VALASZTHATO PARANCSCSOPORTOK

        [KONYVTARKEZELES]          FILE-MUVELETEK
         LEMEZKEZELES              SZOVEGFILE KEZELES
         RENDSZERPARAMETEREK       NYELVI RENDSZEREK

LEMEZ: A KONYVTAR: \                                        8:00:21
 PARANCS:

 ^H : Help          ENTER : Menu/Program
 ↓ : Kurzor elore        ↑ : Kurzor vissza        ← : Visszaleptetes/parancs megsz.
```

3. Az operációs rendszer parancsai

A PROPOS–16 parancsainak nagy része ún. külső parancs (utility), ami azt jelenti, hogy a parancsokat végrehajtó programok mágneslemezes file-okban találhatók, és csak hívásukkor töltődnek be a központi tárba.

A rendszer számára minden .COM és .EXE kiterjesztésű file végrehajtható programot jelent. Ezeket az elsődleges névvel (a kiterjesztés megadása nélkül) hívhatjuk. Ugyanígy indíthatók a parancssorozatot tartalmazó, ún. batch file-ok is, amelyeket a .BAT kiterjesztésről ismer fel a rendszer. Azonos elsődleges név (pl. PROGRAM) esetén a PROPOS–16 a következő módon dönt: megvizsgálja, hogy van-e az aktuális könyvtárban PROGRAM.COM nevű file, és ha van, betölti és elindítja azt. Ha ilyet nem talál, PROGRAM.EXE azonosítót keres, és ha azt sem találja, a PROGRAM.BAT nevű file-t keresi. Csak ha ilyen sincs, akkor jelez hibát.

A PROPOS–16 parancsok általában két részből állnak:

parancsnév, paraméterek

ahol a parancs neve megegyezik az őt végrehajtó program nevével. A paraméterek a program működését befolyásolják, pl.: milyen file-on (vagy file-csoporton) kívánunk műveletet végezni, hol kívánjuk az eredményt megjeleníteni, milyen opcionális műveletekre van szükségünk stb. *A parancsokat végrehajtó programok általában úgy működnek, hogy ha hívásukkor nem adjuk meg a szükséges paramétereket, azokat megkérdezik*, illetve kijelzik, hogy milyen alapértelmezést feltételeztek.

3.1. A parancsok megadásának módja

A képernyőn megjelenő menüből a [↓] és a [↑] billentyűk segítségével választhatjuk ki a kívánt bejegyzést. Ez lehet egy parancs neve vagy egy részletesebb menü azonosítója. A kijelölt tétel negatív írással látszik.

Az előző menüoldalra a [←] kurzormozgató billentyű lenyomásával térhetünk vissza.

Az [ENTER] billentyű lenyomására a kiválasztott részletesebb menü jelenik meg a képernyőn, vagy a kiválasztott parancs a képernyő alján levő parancssorba íródik. Itt megadhatjuk a parancs működéséhez szükséges paramétereket is.

A felkínált menütől függetlenül is adhatunk parancsot a PROPOS–16 rendszernek. Egy betűköz lenyomására is a parancssorba jutunk, ahol tetszőleges parancs megadható.

A bebillentyűzött parancssorokat, valamint a programok által bekért paraméter- és adatsorokat mindig az [ENTER] billentyű lenyomásával zárjuk le és adjuk át a számítógépnek.

Az elütéseket a [←] (backspace) billentyűvel javíthatjuk.
A kijelölt menübejegyzésről a [CTRL] [H] lenyomásával kérhetünk magyarázatot.

A menü használatához szükséges információk a képernyő alsó részén mindig láthatók.

A parancsok bebillentyűzésekor akár kis-, akár nagybetűket használhatunk.

Eldöntendő kérdésekre az I (igen) vagy N (nem) billentyűk és az ENTER lenyomásával kell válaszolni.

Előfordul, hogy egy parancs végrehajtása során felszólítást kapunk valamilyen akcióra (pl. floppy lemezcseréje), vagy operátori jóváhagyástól függ a program továbbfutásának engedélyezése. Ilyenkor tetszőleges billentyű (betűk, számok, ENTER stb.) lenyomásával folytathatjuk a feldolgozást.

A [CTRL] [C] ([CTRL] [BREAK]) lenyomásával a feldolgozást megszakíthatjuk.

Ez a keretrendszer igény szerint bővíthető, illetve átalakítható. A felhasználó saját fejlesztésű vagy készen vásárolt alkalmazói programjait beillesztheti a menübe, míg a ritkábban használt bejegyzéseket (esetleg teljes menüoldalakat is) kihagyhatja abból. Ily módon minden konfiguráción a helyi igényekhez legjobban alkalmazkodó, a leghatékonyabb munkavégzést lehetővé tevő menürendszer hozható létre.

3.2. Jelölések

A következő fejezetekben a parancsok leírásánál az alábbi jelöléseket használjuk:

- Nagybetűkkel írjuk az alapszavakat (pl. DIR, IF, /C stb.), és dőlt kisbetűkkel jelöljük a paramétereket (pl. *m* = meghajtó, amit a konkrét parancsban az A, B, C, ... betűk egyikével kell helyettesíteni).
- A szögletes zárójelek között írt paraméterek opcionálisak, megadásuk nem kötelező.
- Három pont [...] jelzi, hogy az előtte álló paraméter többször is megismételhető – értelemszerűen különböző értékekkel.
- A paraméterek sorrendje általában tetszőleges lehet, kivéve ahol két, azonos típusú paraméter értelmezését a sorrendjük határozza meg. Pl. két file-hivatkozás közül az első mindig a bemenő, a második a kimenő file-t azonosítja.
- A / jellel jelzett, általában egykarakteres paraméterek összevonhatók egyetlen / jel mögé, a következő paraméterleírások tehát egyenértékűek:
  /A/B /AB /BA /B/A
- A parancsok paramétereiként általában teljes file-hivatkozások adhatók meg az alábbi alakban:

  [*m*:] [*út*] *file-név* [.*kit*]

  ahol: *m* a meghajtó azonosítója (utána a kettőspont kötelező),
  út az alkönyvtár kijelölése (l. lejjebb ...),
  kit a file-név kiterjesztése (előtte a pont kötelező).
  A file-hivatkozáson belül betűköz nem fordulhat elő!
- A könyvtárat meghatározó keresési út megadásának módja:

  [\] *könyvtár*1 [*könyvtár*2 [\...]]

  ahol • a felsorolt (egymás alá rendelt) könyvtárneveket \ jel („backslash") választja el;

- a kezdő \ azt jelzi, hogy az út a gyökérkönyvtárnál kezdődik;
- a kezdő \ hiánya esetén az út az aktuális könyvtártól kezdődik;
- ha az út leírása után file-név is következik, azt is \ jellel kell elválasztani az utolsó könyvtárnévtől;
- az út leírása (a gyökértől a legalsó szintig) nem lehet hosszabb 63 karakternél.

■ A meghajtó azonosítójának elhagyása esetén az alapértelmezés a feltételezett meghajtó; az út alapértelmezése az aktuális könyvtár; a kiterjesztés alapértelmezése három betűköz.

A parancsszó és a paraméter(ek) között az elválasztó betűköz kötelező, míg a file-hivatkozás közben betűköz nem szerepelhet (itt az elválasztó jelek a :, \ és .).

Végül, néhány gyakorlati jó tanács a PROPOS–16 parancsok használatához:

A beírt parancssort –különösen a géppel való ismerkedés időszakában – nézzük át még egyszer, mielőtt az [ENTER] billentyűt lenyomjuk.

Ellenőrizzük, hogy a megfelelő lemezt tettük-e a meghajtóba (esetleg a tartalomjegyzékének kiíratásával is.)

Ügyeljünk rá, hogy ha nem a feltételezett meghajtót akarjuk használni, adjuk meg a meghajtó jelét (kettősponttal lezárva).

A rendszer nem találja meg a kért program- vagy adatfile-t, ha nem adjuk meg pontosan az azonosítóját. Ne feledkezzünk meg a paraméterként megadott file-név kiterjesztéséről sem!

Törlésnél (DEL parancs) mindig meggondoltan járjunk el, különösen akkor, ha file-csoportnév megadásával egyszerre több file-t akarunk törölni.

Program- és adatfile-jainkról rendszeresen készítsünk másolatot, hogy a lemezek fizikai sérüléséből eredő kellemetlenségeket elkerüljük (l. COPY, BACKUP parancsok).

Mágneslemez írás/olvasás közben a programot ne állítsuk le, a gépet ne kapcsoljuk ki, a floppy lemezt ne vegyük ki.

Lemezformálás (FORMAT parancs) előtt győződjünk meg arról, hogy a lemez nem tartalmaz hasznos információt.

A következő alpontokban részletesen ismertetjük a PROPOS–16 operációs rendszer standard parancskészletét. Ne feledjük, hogy ez a készlet minden újabb – saját fejlesztésű vagy készen vásárolt – programmal tovább bővül!

3.3. Lemezkezelés

A PROPOS–16 parancsok ezen csoportjába soroljuk azokat a programokat, amelyek a hajlékony vagy a merev lemez egészén végeznek műveleteket. Használatuk kellő óvatosságot igényel, mivel helytelen (de nem okvetlenül hibás) paraméterezés esetén könnyen tönkretehetjük előző munkánk eredményét.

Az itt ismertetésre kerülő parancsok feladata:

- ■ új hajlékony lemez kezdeti megformálása (FORMAT),
- ■ lemezazonosító cimke lekérdezése (VOL),
- ■ hajlékony lemez teljes tartalmának átmásolása (DISKCOPY),
- ■ file-ok, file-csoportok, könyvtárágak mentése lemezről lemezre (BACKUP),
- ■ az elmentett file-ok, file-csoportok, könyvtárágak visszatöltése (RESTORE),
- ■ a lemez paramétereinek kiíratása, „elvesztett" területek feltárása (STAT),
- ■ hibás lemez vagy file-ok javítása (RECOVER).

FORMAT

Az új lemezeken még semmilyen információ nem található, sem sáv-, sem szektorbeosztás nincs a felületükön. Így ugyanaz a lemez felhasználható bármely – ilyen méretű lemezt kezelő – számítógéphez.

A FORMAT parancs segítségével a hajlékony lemezen a PROPOS–16 operációs rendszernek megfelelő beosztást hozhatjuk létre. Ugyanazon a meghajtón általában többféle felépítésű lemezt is formálhatunk, majd a továbbiakban írhatjuk, illetve olvashatjuk azokat. A lehetséges változatokat x jellel jelöljük.

			Meghajtó		
			Egyoldalas	Kétoldalas	
			40 sáv	40 sáv	80 sáv
Lemez	Kétoldalas	80 sáv			x
		40 sáv		x	x
	Egyoldalas	40 sáv	x	x	x

Ennek elsősorban a különböző PROPER–16 számítógépek közötti kompatibilitás miatt van jelentősége. A FORMAT parancs a lemez teljes felületét ellenőrzi. A hibás, használhatatlan szektorokat, illetve sávokat kiiktatja (az ún. file-allokációs táblában megjelöli). Az így kialakított file-allokációs táblát, a rendszer betöltéséhez szükséges ún. „boot" rekordot, valamint az üres gyökérkönyvtárat is felírja a lemezre. A lemezre azonosító cimke is írható.

4–50. oldal

A paranccsal PROPOS–16 rendszerlemezt is létrehozhatunk, amely betölthető formában tartalmazza az operációs rendszer moduljait is.

A lemezek felépítéséről, a file-allokációs tábla és a könyvtárak tartalmáról a **PROPOS–16 – Programozóknak** című kézikönyvből tudhatunk meg többet.

VOL

Célszerű a FORMAT paranccsal azonosító cimkét is íratni a lemezre. Ily módon a lemez neve nemcsak tokjára írva látható, hanem a számítógép számára is hozzáférhető. A VOL parancs segítségével bármely lemezazonosító cimkéjét kiírathatjuk a standard output perifériára.

4–92. oldal

DISKCOPY

Köztudott, hogy a hajlékony lemez sérülékeny, előfordulhat, hogy a rajta tárolt adatok egy része elvész, olvashatatlanná válik. Ennek oka lehet mechanikus vagy mágneses sérülés, sőt véletlen törlés vagy felülírás is. Ezért elterjedt gyakorlat, hogy fontosabb lemezes állományainkról időközönként másolatot készítünk.
A DISKCOPY paranccsal egy- vagy kétoldalas, 40, illetve

80 sávos floppy lemezekről készíthetünk másolatot. A másolat pontosan megegyező tartalmú az eredeti lemezzel, amit a másolást követő összehasonlítással is ellenőrizhetünk.

4–46. oldal

Új lemezre is másolhatunk, mert a parancs lemezformálást és ellenőrzést is végez.

BACKUP – – RESTORE

4–4. oldal

A biztonsági lemezmásolatok készítésének másik eszköze a BACKUP parancs. Segítségével bármilyen típusú lemezről bármilyen másik lemezre átvihetünk (menthetünk) file-okat, file-csoportokat vagy teljes könyvtárakat. A másolatot azonban önmagában nem használhatjuk, mert ezen a file-ok formátuma eltér az eredetitől. A mentés során minden file előtt a hozzá vezető keresési út is rögzítésre kerül.

A másolatot a RESTORE paranccsal tölthetjük vissza. A visszatöltés mindig az eredeti helyre, a megfelelő (al)könyvtárba történik.

4–77. oldal

A mentés és visszatöltés szelektíven, a file keletkezésének, illetve utolsó módosításának dátumától függően is végezhető.

STAT

A STAT parancs lehetővé teszi, hogy számszerű adatokat kaphassunk bármely (fix, hajlékony vagy virtuális) lemez kapacitásáról, a rajta tárolt file-ok számáról és összes helyfoglalásáról, továbbá az elérhetetlen terület nagyságáról. Ez utóbbi adat a file-módosítások eredményeként felszabadult, de pillanatnyilag nem elérhető szektorok összes területére utal. A STAT parancs ezen lemezterületek újbóli használatbavételét is elősegíti oly módon, hogy a kiesett szektorokat file-okká fűzi össze. Az így keletkezett file-okat törölhetjük (felszabadítva helyüket).

A STAT parancs külön kijelzi a vizsgált lemezen talált, ún. rejtett file-ok számát és összterületét is. Ebből általában következtethetünk a lemez jellegére, illetve tartalmára is:

Rejtett file-ok száma	Lemezcimke	PROPOS–16 rendszer
0	nincs	nincs
1	van	nincs
2	nincs	van
3	van	van

4–87. oldal

A rejtett file-ok fogalomkörével a CHMOD parancs ismertetésénél is találkozunk.

RECOVER

Bármilyen óvatosak vagyunk, és gondosan másolatokat készítünk file-jainkról, előfordulhat, hogy egy olyan lemez vagy file sérül meg, amelyről éppen nincs másola-

tunk. Ekkor jogos óhajunk, hogy a sérült file-ból vagy lemezről legalább az ép részeket visszanyerhessük.

A RECOVER parancs megvizsgálja a kijelölt file-(oka)t, és ha hibás, fizikailag sérült szektort talál, azt kiiktatja, majd az így megcsonkított file-t ismét használhatóvá teszi.

4–75. oldal

3.4. Könyvtárkezelés

Ebben a fejezetben a PROPOS–16 operációs rendszer azon parancsait ismertetjük, amelyek hatásköre egy (al)könyvtárra terjed ki. Segítségükkel lehetséges:

- egy könyvtár tartalmának kiíratása (DIR),
- a könyvtárak kapcsolatrendszerének megjelenítése (FSYS),
- alkönyvtár létrehozása (MKDIR vagy MD),
- alkönyvtár megszüntetése (RMDIR vagy RD),
- aktuális könyvtár kijelölése (CHDIR vagy CD),
- keresési utak kijelölése (PATH).

DIR

A lemezen való tájékozódás legfontosabb segédeszköze a DIR parancs. Lehetővé teszi egy-egy könyvtár vagy file-csoport tartalmának kiíratását (a file-azonosítók kilistázását). A listán szerepel a lemezazonosító cimkéje, a felsorolt file-ok egyenkénti és összes helyfoglalása, valamint a lemezen lévő szabad terület mérete.

A parancs a listán ⟨DIR⟩ megjelöléssel emeli ki a kiíratott könyvtár alá rendelt alkönyvtárak azonosítóját. Rejtett file-ok ezzel a paranccsal nem fedhetők fel.

4–43. oldal

FSYS

Míg a DIR parancs csak egy könyvtár tartalmába engedett bepillantani, és így csak a közvetlenül alá tartozó alkönyvtárak azonosítóit ismerhettük meg, az FSYS parancs segítségével egy lemez összes könyvtárának kapcsolatrendszerét kiírathatjuk. A könyvtárak fastruktúrájának végigkövetése során az alkönyvtárakban található file-ok azonosítóiról is készíttethetünk listát.

4–56. oldal

MKDIR

Az MKDIR (vagy MD) parancs kiadásával tudunk egy lemezen új alkönyvtárat létrehozni. Az új alkönyvtár természetesen még üres, ezután kell file-okkal feltölteni.

A fix lemezen az alkönyvtárak rendszerének kialakítása fontos, az egész számítógépes rendszer hatékonyságát befolyásoló feladat. Célszerű alkönyvtárakba elkülöníteni az egyes alkalmazási területeket, illetve felhasználói csoportokat, nyelvi rendszereket, fejlesztői tevékenységet stb.

30–40 tételnél több bejegyzést (file-t vagy további alkönyvtárt) általában nem érdemes egy alkönyvtárban tárolni. A könyvtárak rendszerét ne tervezzük túl bonyolultra, mert háromnál több szintű fastruktúra már elég nehezen tekinthető át.

4–65. oldal

RMDIR

4–80. oldal

Egy alkönyvtár megszüntetésére használhatjuk az RMDIR (vagy RD) parancsot. Csak üres, file-okat és további alkönyvtárakat nem tartalmazó alkönyvtár szüntethető meg.

CHDIR

4–18. oldal

Az előző fejezetben már megismerkedtünk az aktuális könyvtár fogalmával. Tudjuk, hogy az ebben tárolt program- és adatfile-okhoz egyszerűbben, gyorsabban lehet hozzáférni, ezért célszerű mindig azt a könyvtárat tenni aktuálissá, amelyben pillanatnyilag dolgozunk. Az aktuális könyvtár kijelölésének eszköze a CHDIR (vagy CD) parancs.

PATH

4–71. oldal

Munkánkat általában nem lehet úgy szervezni, hogy csak egyetlen (az aktuális) könyvtárban levő programokat használjunk. Az operációs rendszer külső parancsait, közös programokat egy lemezen csak egy példányban tárolunk, és azokhoz többen szeretnénk hozzáférni. A programok hívásakor viszont nem kívánunk azzal foglalkozni, hogy a betöltést melyik könyvtárból kell elvégezni.
A PATH parancs lehetővé teszi, hogy kijelöljük azokat a könyvtárakat, amelyekben az általunk használt parancsokat végrehajtó programok keresendők.

3.5. File-kezelés

Az ebbe a csoportba sorolt PROPOS–16 parancsok egy file-on vagy file-csoporton fejtik ki hatásukat. A file-kezelő parancsok egy részének működése csak a szövegfile-okra korlátozódik – ezt részletes leírásuknál jelezzük. Ugyancsak külön megadjuk, hogy mely parancsok engedik meg file-csoport-azonosítók megadását paraméterként. Az itt ismertetésre kerülő parancsok feladata:

- szövegfile tartalmának kiíratása (TYPE),
- szövegfile kinyomtatása (PRINT),
- a standard output lapokra bontása (MORE),
- file-ok másolása és/vagy egyesítése (COPY),
- file nevének megváltoztatása (RENAME vagy REN),
- file törlése (DEL),
- file-ok összehasonlítása byte-onként (CMP),
- szövegfile-ok összehasonlítása soronként (COMM),
- szövegfile-ok közti eltérések kiíratása (DIFF),
- szövegfile sorainak, szavainak, illetve karaktereinek leszámlálása (COUNT),
- adott karaktersorozat előfordulásainak megkeresése egy szövegfile-ban (FIND),
- szövegfile rendezése (SORT),
- hexadecimális file-kiíratás (HD),
- programfile átalakítása EXE formátumról COM formára (CONVEXE),
- más operációs rendszer alatt felírt file-ok átalakítása PROPOS–16 formátumra és viszont (IMP),
- egy file attributumainak megváltoztatása (CHMOD).

TYPE

4–91. oldal

Szövegfile-ok tartalmát egyszerűen kiírathatjuk a képernyőre (standard output perifériára) a TYPE parancs segítségével. Paraméterezéssel a file elejétől, illetve végétől adott távolságra is kezdhetjük a listázást.

PRINT

Szövegfile-ok tartalmának kinyomtatására többféle lehetőség is van. A TYPE parancs kimenetét átirányítva, vagy azzal egy időben hard copy-t kérve is csinálhatunk nyomtatott listát – ezalatt azonban a számítógép teljes kapacitását lekötjük.

A PRINT parancs a neki megadott file-okat sorba állítja, és kinyomtatásukról a „háttérben" gondoskodik, miközben a számítógépen más feladatot oldhatunk meg. Természetesen a nyomtatót közben más célra nem használhatjuk, és a nyomtatás alatt álló file-okat sem módosíthatjuk.

4–72. oldal

MORE

A programok üzeneteiket általában a standard output perifériára, azaz a képernyőre írják ki. Ha ez egy olyan lista, amely 25 normál sornál hosszabb, a képernyőn az üzenetek „elfutnak", és a program lefutása után csak az utolsó 25 sor marad látható. Ha a programot összeláncoljuk (l. 3.7. fejezet) a MORE paranccsal, vagyis a standard outputot a MORE standard inputjára irányítjuk, a listát 24 soros lapokra tördelve kapjuk, és csak egy billentyű lenyomása után folytatódik a program üzeneteinek kiírása.

4–70. oldal

COPY

Több helyen is utaltunk már rá, hogy a hajlékony mágneslemezek sérülékenyek, ezért ajánlatos róluk másolatot készíteni. Ennek egyik eszköze a COPY parancs, amely tetszőleges (fix, hajlékony és virtuális) lemezek, sőt egy lemezen belül könyvtárak közötti file-másolás elvégzésére is alkalmas. Segítségével a másolatnak új azonosító is adható – ily módon akár egy könyvtáron belül is duplikálhatunk egy file-t. A file-másolás file-egyesítéssel is összeköthető, azaz több bemenő file-t egyetlen eredményfile-ba másolhatunk át.

4–30. oldal

RENAME

4–76. oldal

Egy file vagy file-csoport nevét, illetve nevének kiterjesztését változtathatjuk meg a RENAME (vagy REN) parancs kiadásával.

DEL

Feleslegessé vált file-jainkat a DEL parancs segítségével törölhetjük a lemezről. File-csoport törlésekor különösen fontoljuk meg a DEL kiadását, nehogy olyan file-t is töröljünk, amelyre még szükségünk van. A törölt file többé nem található meg a lemezen.
Alkönyvtár megszüntetésére (törlésére) a DEL nem használható (l. RMDIR parancs).

4–39. oldal

CMP

Két, tetszőleges formátumú file összehasonlítását a CMP paranccsal végezhetjük el. Az összehasonlítás byte-onként történik, az eltérő byte-ok helyéről és tartalmáról listát

kapunk. Az összehasonlítandó file-ok különböző meghajtókon (akár a virtuális lemezen is) lehetnek, így ez a parancs a másolatok ellenőrzésére használható.

4–21. oldal

COMM

Csak szövegfile-ok összehasonlítására szolgál a COMM parancs. A file-ok összevetését soronként végzi, és az eltérő sorokat jelzi. A két file-ban a sorok különböző sorrendben is szerepelhetnek, ám ha különben azonos sorokból állnak, a COMM parancs egyezést jelez. Az egyező, illetve csak az egyik vagy csak a másik file-ban előforduló sorokról listát kérhetünk.

4–23. oldal

DIFF

Egymástól kismértékben eltérő, de nagy helyigényű szövegfile-ok kisebb lemezterületeken tárolhatók, ha csak az egyik file-t őrizzük teljes terjedelmében, és létrehozunk egy olyan file-t, amely a másik szövegfile előállításához szükséges szövegszerkesztő parancsokat tartalmazza (sortörlések és -beszúrások). A második file-t szükség esetén az EDIT programmal (l. az 5. fejezetet) állíthatjuk elő úgy, hogy az első file-t módosítjuk a DIFF paranccsal létrehozott javítófile alapján.

4–41. oldal

COUNT

Szintén csak szövegfile-ok kezelésére használható a COUNT parancs. Lehetővé teszi, hogy egy file sorainak, szavainak és karaktereinek számáról információt kapjunk.

4–36. oldal

FIND

Gyakori feladat, hogy egy szövegfile-ban meg szeretnénk keresni egy adott karaktersorozat előfordulásait. A FIND parancs felhasználásával listát kaphatunk azokról a sorokról, amelyekben a megadott szövegrész szerepel. A parancs paraméterezésével a keresés menete, illetve az eredmény kijelzése sokféleképpen befolyásolható.

4–48. oldal

SORT

Szövegfile-ok sorainak rendezéséhez használható a SORT parancs. A rendezés több file egybeolvasztásával kombinálva is végeztethető. A rendezés betűrendben történik, alapértelmezés szerint a sor (rekord) egésze alapján. Paraméterek megadásával kijelölhető, ha a rendezést a sor bizonyos pozíciói közé eső karakterek alapján kívánjuk végezni, ha fordított sorrendet akarunk előállítani, ha a kis- és nagybetűket nem kívánjuk megkülönböztetni stb.

Lehetőség van egy file rendezettségének ellenőrzésére is, a file változtatása nélkül.

4–84. oldal

HD

Tetszőleges formátumú file tartalmát kiírathatjuk a standard output perifériára a HD (hexadecimális dump) parancs segítségével. A kiíratás hexadecimális és/vagy karakteres formában történik. A parancs paraméterezésével kijelölhető egy távolság is, ha a kiíratást nem a file elejétől kérjük.

A HD parancs főleg ismeretlen tartalmú, nem szövegrekordokból álló file-ok kiíratására használható.

4–57. oldal

CONVEXE

A szerkesztő (LINK) program által előállított futtatható programok .EXE kiterjesztésűek, felépítésüket a szerkesztő meghatározza. Ha összeszerkesztett programunk bizonyos előírásoknak megfelel, a CONVEXE parancs segítségével .COM formátumra alakítható át. Ennek a helyfoglalása a lemezen kisebb, betöltése gyorsabb.

4–29. oldal

IMP

Előfordulhat, hogy más operációs rendszer (CP/M, PROPOS–8, ...) alatt felírt lemezeinkről szeretnénk legfontosabb file-jainkat a PROPOS–16 operációs rendszer alá átültetni, mert feldolgozásukat ezen a gépen akarjuk folytatni. A file-ok kétirányú konverzióját végzi az IMP parancs.

4–60. oldal

CHMOD

Az operációs rendszer minden file-hoz hozzárendel egy ún. attributum byte-ot, amely a file fő jellemzőire utal. Egy-egy bit jelzi, hogy a file csak olvasható, rejtett, rendszerfile, lemezazonosító cimke vagy alkönyvtár, és változott-e az utolsó mentés (BACKUP) óta. A CHMOD parancs segítségével a file attributumában megváltoztatható a csak olvasható, a rejtett és a rendszerfile jelzés, illetve megtudható ezen jelzőbitek pillanatnyi állása. A többi bit állítására nincs mód, mert megváltoztatásukkal könnyen tönkretehetnénk lemezünk szervezettségét.

4–19. oldal

3.6. Perifériakezelés, konfigurálás

Néhány parancs segítségével az operációs rendszer működését is befolyásolhatjuk. Ezek az ún. konfiguráló parancsok részben az egyes perifériák üzemmódját állítják át, részben pedig az operációs rendszer paraméterezését, néhány software-kapcsoló átállítását teszik lehetővé:

- a képernyő és nyomtató üzemmódjának beállítása (MODE),
- mágneslemez-műveletek átirányítása másik meghajtóra (ASSIGN),
- standard input/output átirányítása egy terminálra (CTTY),
- dátum, pontos idő és más PROPOS–16 változók lekérdezése, illetve beállítása (SET),
- a CONFIG.SYS konfigurációs file lekérdezése, illetve módosítása (CONFIG),
- a menü/help információs file módosítása (MENUSYS).

MODE

A MODE parancs segítségével a képernyő és a nyomtató üzemmódját változtathatjuk meg. A képernyőn színes vagy fekete-fehér üzemmódban, soronként 40 vagy 80 karakteres írásmódot állíthatunk be. A nyomtatót soronként 80 vagy 132 karakter kiírására és 1/6 vagy 1/8 inches (4,3, illetve 3,2 mm) soremelésre utasíthatjuk. Ezzel a paranccsal állíthatók be az aszinkron adatátviteli csatolók paraméterei is.

4–67. oldal

ASSIGN

Egyes programok adatfile-jaikat egy-egy, a program belsejében rögzített meghajtón keresik. Ha valamilyen okból ehelyett más meghajtót szeretnénk használni, az ASSIGN parancs segítségével egy adott meghajtóra vonatkozó be- és kiviteli kérelmeket egy másik meghajtóra irányíthatunk át.

4–3. oldal

CTTY

Kivételes esetekben lehet szükségünk arra, hogy standard input/output perifériaként egy, a géphez kapcsolt aszinkron terminált jelöljünk ki.

4–38. oldal

SET

A SET nevű, interaktív kezelésű parancs segítségével lekérdezhetjük és szükség esetén módosíthatjuk az operációs rendszer néhány belső paraméterét:

- napi dátum,
- pontos idő,
- CTRL–BREAK figyelés,
- ellenőrzés lemezírás után,
- a parancsértelmező paraméterei (environment).

4–81. oldal

CONFIG

A rendszerlemez gyökérkönyvtárában hozható létre a CONFIG.SYS azonosítójú file, amely az operációs rendszer betöltésekor és elindításakor meghatározza a PROPOS–16 néhány fontos jellemzőjét:

- rendszerpufferek száma,
- egyidejűleg használható file-ok maximális száma,
- speciális perifériakezelő rutinok,
- virtuális lemez(ek) definiciója.

Ezeket a jellemzőket (szöveges magyarázattal) írja ki a CONFIG program, és lehetővé teszi megváltoztatásukat is. Ezenkívül kiírja néhány nem módosítható hardware jellemző értékét is:

- központi tár mérete,
- lemezmeghajtók száma,
- aszinkron csatolók száma,
- párhuzamos illesztésű nyomtatók száma.

4–25. oldal

MENUSYS

Az operációs rendszer felhasználói interface-ét elsősorban a menü/help rendszer határozza meg. A menü/help információs file tartalma – mint minden szövegfile – az EDIT szövegszerkesztő segítségével megváltoztatható. A változtatást a MENUSYS.BAT batch-file segítségével végezhetjük el.

4–62. oldal

3.7. Batch parancsok

A PROPOS–16 rendszer lehetővé teszi, hogy gyakran szükséges parancssorozatokat speciális, ún. batch file-okban tároljunk. A parancsok egyenkénti bebillentyűzése helyett elegendő a kívánt batch file nevét megadnunk – a PROPOS–16 parancsértelmező egyenként kiolvassa és végrehajtja a file-ban talált parancsokat. Ezek lehetnek az operációs rendszer parancsai vagy bármilyen más program hívásai.

A batch file paraméterezhető is, azaz indításakor megadhatunk file-azonosítókat vagy más paramétereket, amelyeket az operációs rendszer behelyettesít a végrehajtásra kerülő parancsok kijelölt paraméterei helyére. Ily módon ugyanazzal a batch file-lal különböző file-okon, file-csoportokon végezhetjük el ugyanazt a feladatot. A batch file-ok néhány tipikus alkalmazása:

- működési környezet kialakítása,
- programmodulok fordítása, szerkesztése,
- ismétlődő, összetett feldolgozások,
- könyvtárak tartalmának kiírása,
- file-ok másolása, hasonlítása stb.

A batch file nevét felvehetjük a PROPOS–16 menürendszerbe is, így elindítása leegyszerűsödik.
A batch file-ok létrehozására bármilyen szövegszerkesztő program (pl. EDIT, l. e. könyv 5. fejezetében), vagy a COPY parancs (l. 4-30. old.) használható. A file neve tetszőleges lehet. A file-név kiterjesztése kötelezően .BAT. Természetesen egy könyvtárban minden file-azonosító csak egyszer fordulhat elő.

A batch file-okban néhány speciális batch alparancs is használható, amelyeknek a feladata:

- a batch parancsok kiírásának letiltása vagy engedélyezése (ECHO),
- üzenet kiírása a képernyőre (REM),
- üzenet kiírása és várakozás egy billentyű lenyomására (PAUSE),
- eltérés a parancsok soros feldolgozásától (GOTO),
- egy parancs feltételtől függő végrehajtása (IF),
- egy parancs ismételt végrehajtása (FOR),
- paraméterek eltolása (SHIFT).

Az AUTOEXEC.BAT azonosítójú file tartalma a PROPOS–16 betöltése után automatikusan végrehajtódik.

ECHO

4–9. oldal

A batch file-ok végrehajtásakor a sorra kerülő parancs kiíródik a képernyőre (a standard output perifériára). Az ECHO paranccsal ezt a kiírást tilthatjuk le, illetve engedélyezhetjük újra. Az egyes programok által kiírt üzeneteket az ECHO nem tiltja le, azok mindenképpen megjelennek a képernyőn.

REM

4–16. oldal

A REM parancs a sorában megadott szöveget (üzenetet) kiírja a képernyőre. Használatával a gépkezelőt tájékoztathatjuk a batch munka futásának pillanatnyi fázisáról. Az ECHO paranccsal ezek az üzenetek is letilthatók.

PAUSE

A PAUSE paranccsal is írathatunk ki üzeneteket, a REM-től eltérően azonban a PAUSE hatására felfüggesztődik a batch file végrehajtása, és csak valamelyik billentyű le-

nyomására folytatódik a munka. Ezáltal lehetőséget adhatunk a gépkezelőnek a floppy lemez kicserélésére, a nyomtatópapír beállítására vagy éppen a batch munkafolyamat befejezésére (CTRL–C).

4–15. oldal

GOTO

A programozási nyelvekben ismert módon a batch file-ban is elhelyezhetünk címkéket, és a GOTO paranccsal átadhatjuk ezekre a vezérlést. Ez azt eredményezi, hogy eltérünk a parancsok soros végrehajtásától, sőt ciklusokat is szervezhetünk a batch file-ban. Vigyázzunk, mert végtelen ciklust is könnyen létrehozhatunk!

4–12. oldal

IF, IF NOT

A ciklusok, elágazások megvalósításához nélkülözhetetlen az IF parancs, amellyel bizonyos feltételeket vizsgálhatunk, és teljesülésüktől függően hajthatunk végre egy parancsot, vagy ágazhatunk el a parancssorozat más részére. Az IF NOT parancs ellenkező értelmű vizsgálatot is lehetővé tesz.

4–13. oldal

FOR

Egy programot ismételve, változó paraméterekkel is végrehajtathatunk. A parancssorban megadható egy értékkészlet, amelynek elemeit a FOR parancs rendre behelyettesíti az adott programot hívó parancsba. Így egyetlen sorral írhatunk le több, hasonló műveletet (pl. több könyvtár tartalmának kiíratása, többpéldányos nyomtatás stb.).

4–11. oldal

SHIFT

Ez a parancs lehetővé teszi, hogy a batch file-nak 10-nél több paramétert is megadhassunk. A megadott paramétereket eggyel balra lépteti, vagyis az első paraméter kapja a második értéket, a második paraméter a harmadik értéket, és így tovább. A tizedik paraméter felveszi az eddig nem használt 11. értéket.

4–17. oldal

3.8. Bemenet/kimenet átirányítása

A PROPOS–16 operációs rendszer parancsait végrehajtó programok, illetve minden olyan program, amely az operációs rendszeren keresztül végez be- vagy kiviteli műveletet, használhatja az ún. *standard input* és *standard output* perifériákat.
A PROPOS–16 elindításakor a standard input a billentyűzet, a standard output pedig a képernyő.

Az operációs rendszer segítségével átirányíthatjuk a standard inputról érkező, illetve a standard output perifériára küldött adatainkat.

Egyik lehetőség a CTTY parancs alkalmazása, amellyel egy, a géphez kapcsolt terminált jelölhetünk ki standard input/output perifériának.

A másik lehetőség alapja az, hogy a PROPOS–16 operációs rendszer egységes módon kezeli a perifériákat és a mágneslemezes file-okat. Így az is megvalósítható, hogy a billentyűzet bemenet és a képernyő kimenet helyére más perifériát vagy mágneslemez file-t iktassunk be. Általában ezt a lehetőséget nevezzük *bemenet/kimenet átirányítás*nak.

A program megváltoztatása nélkül, a parancssorban jelölhetjük ki a logikai be- és kiviteli műveletek végrehajtásához igénybe vett fizikai berendezéseket, illetve file-okat.

A **kimenet átirányításának** leírási módjai:

...〉[*m*:] [*út*] *file-név* [.*kit*]

Az ilyen módon kiegészített paranccsal a standard output perifériára kiküldött rekordok a megadott file-ba íródnak. Ha korábban már volt a kijelölt könyvtárban ilyen file, az törlődik, és az elejétől kezdődik feltöltése. File-névként bármely, kivitelre alkalmas periféria neve (PRN, LPTn, NUL, ...) is megadható.

...〉〉[*m*:] [*út*] *file-név* [.*kit*]

Ha a meghajtó kijelölt könyvtárában nem található a megadott azonosítójú file, a rendszer létrehozza, ha viszont volt már ilyen file, nem törli, hanem folytatásként írja hozzá a standard outputra küldött valamennyi üzenetet.

A bemenet átirányítása:

...〈[*m*:] [*út*] *file-név* [.*kit*]

A fordított nyíl (kisebb jel) jelzi a standard input átirányítását. A program a billentyűzetről várt adatait, paramétereit a kijelölt file-ból veszi.A file-név helyén inputra használható perifériánév (AUX, COMn, NUL, ...) is megadható.

A standard input átirányítása különös figyelmet igényel, mert előre kell gondoskodnunk *minden input paraméter megfelelő sorrendben* történő tárolásáról.

Hangsúlyozzuk, hogy csak azon programoknál lehet a standard input/output perifériákat átirányítani, amelyek az operációs rendszer közvetítésével végzik adatforgalmazásukat.

Példák

DIR〉PRN

Ezzel a paranccsal az aktuális könyvtár tartalmát a képernyő helyett a nyomtatóra írathatjuk ki.

DIR A:〉TARTALOM
DIR B:〉〉TARTALOM

Az A és B meghajtókon levő lemezek tartalmát egyetlen file-ba íratjuk ki. Az első parancs új file-t hoz létre, a második azt folytatja.

SZAMOL〈PARAM

A program bemenő paramétereit a billentyűzet helyett a PARAM file-ból olvassa.

SZAMOL〈PARAM〉EREDM

A program bemenő paramétereit a PARAM file-ból veszi, eredményeit az EREDM file-ba írja.

3.9. Parancsláncolás

Gyakori igény, hogy egy program eredményeit egy másik programmal értékelhessük ki. Az előző fejezetben tárgyaltuk a programok által leggyakrabban használt standard input és output perifériák használatának módjait. A PROPOS–16 *parancsláncolási* funkciója egy program standard outputra küldött üzeneteit egy másik program standard inputjaként használja fel.

A parancsláncolás használatához tudnunk kell, miként hajtja végre az operációs rendszer ezt a műveletet. A standard outputot egy ideiglenes file-ba irányítja, amelyet a feltételezett meghajtó gyökérkönyvtárában hoz létre. A következő program ezt a file-t használja bemenetként. Ha a gyökérkönyvtárban nincs elég hely az ideiglenes file számára, vagy a feltételezett meghajtón írásvédett lemez van, nem végezhetünk parancsláncolást. Felhasználása után a rendszer automatikusan törli az ideiglenes file-t.

A láncolt parancsokat egy sorban írjuk, és a ¦ jellel választjuk el.

Példák

DIR ¦MORE

A feltételezett meghajtó aktuális könyvtárának tartalomjegyzékét laponként íratjuk ki.

DIR ¦SORT – /D ¦MORE

A könyvtár tartalmát betűrendbe szedve íratjuk ki, ismét lapokra tördelve.

DIR A:〉TARTALOM

⋮

DIR B: ¦COMM – TARTALOM /3 〉LPT1

Az A meghajtó aktuális könyvtárát összehasonlítjuk a B meghajtó aktuális könyvtárával. Az elsőt először egy file-ba írjuk, majd összehasonlítjuk a másodikkal. Csak az eltérő sorokat íratjuk ki, a nyomtatóra átirányítva.

A parancsok

4. A parancsok használata

Könyvünk e fejezetében betűrendben, teljes részletességgel ismertetjük a PROPOS–16 operációs rendszer parancskészletét. A parancsok használatának lehetőségeit példákkal is szemléltetjük. Az EDIT program használatát az 5. fejezetben mutatjuk be.

A PROPOS–16 parancsok közül az operációs rendszer rezidens része hajtja végre az alábbiakat (belső parancsok):

Batch	parancsfile feldolgozása
CHDIR	aktuális könyvtár kijelölése
CTTY	standard input/output átirányítása terminálra
DEL	file törlése
DIR	könyvtártartalom kiírása
MKDIR	alkönyvtár létrehozása
PATH	alternatív könyvtárak kijelölése
RENAME	file-azonosító megváltoztatása
RMDIR	alkönyvtár megszüntetése
TYPE	szövegfile kiírása képernyőre
VOL	lemezazonosító kiírása

Ezek a legegyszerűbb, leggyakrabban szükséges parancsok, amelyeket fix lemez nélküli konfigurációban a rendszerlemez behelyezése nélkül is hívhatunk.

A rendszerlemezről csak hívásukkor töltődnek be az alábbi parancsokat végrehajtó programok (külső parancsok, utilityk):

ASSIGN	meghajtókijelölés átirányítása
BACKUP	mágneslemezes file-ok mentése
CHMOD	file-attributumok lekérdezése, módosítása
CMP	file-ok összehasonlítása byte-onként
COMM	szövegfile-ok összehasonlítása soronként
CONFIG	konfiguráció lekérdezése, módosítása
CONVEXE	.EXE file átalakítása .COM formátumra
COPY	file másolása, egyesítése, összehasonlítása
COUNT	szövegfile sorainak, szavainak, karaktereinek számlálása
DIFF	szövegfile-ok közötti eltérések kiíratása
DISKCOPY	hajlékony lemez másolása
EDIT	szövegfile javítása, kiíratása
FIND	karaktersorozat keresése szövegfile-ban
FORMAT	lemezformálás
FSYS	könyvtárak kapcsolatrendszerének kiíratása
HD	file tartalmának hexadecimális kiíratása
IMP	PROPOS–8/PROPOS–16 file-konverzió

MENUSYS	menü és help szövegek módosítása
MODE	képernyő, nyomtatók és aszinkron vonalak üzemmódjának beállítása
MORE	standard output lapokra osztása
PRINT	szövegfile-ok kinyomtatása
RECOVER	sérült file-ok helyreállítása
RESTORE	mentési file-ok visszatöltése
SET	rendszerváltozók lekérdezése, átállítása
SORT	szövegfile-ok rendezése
STAT	könyvtárak, file-ok elemzése

A PROPOS–16 parancsok paraméterezését összefoglaló táblázatok a D. függelékben találhatók.

Feladata

Egy meghajtóra vonatkozó be- és kiviteli kérelmeket másik meghajtóra irányítja át.

Paraméterezése

ASSIGN [*x* =*y* [...]]

Használata

A parancs használatára akkor van szükség, ha egy program a kódban rögzített meghajtó(ka)t használ működése során, de ehelyett ezúttal másik meghajtón levő fileokkal szeretnénk dolgozni.

Paraméterek nélküli alakban az ASSIGN parancs visszaállítja az eredeti meghajtóhozzárendeléseket.

Az ASSIGN paramétereiben *x* jelenti a programban eredetileg használatos meghajtó azonosítóját, *y* pedig a használni kívánt másik meghajtót.

A parancs kiadása után a PROPOS–16 rendszer minden, az *x* meghajtóra vonatkozó be- és kiviteli kérelmet az *y* meghajtóra irányít át.

x és *y* az A, B, C, ... betűk lehetnek, utánuk nem kell kettőspontot írni.

Figyelem! Azok a rendszerparancsok, illetve programok, amelyek a meghajtó típusától függően másként működnek, nem használhatók ASSIGN paranccsal módosított környezetben (BACKUP, FORMAT, DISKCOPY).

Saját fejlesztésű programjainkba ne építsük be egy bizonyos meghajtó azonosítóját, így az ASSIGN parancs használatára nem lesz szükségünk! Ha mégis szükséges, használata után ne feledkezzünk meg az eredeti hozzárendelések visszaállításáról!

Példák

ASSIGN A=E B=E

Valamennyi, az A és B meghajtóra vonatkozó igényt az E meghajtóra irányítunk át. Ilyen parancs kiadása után pl. a DIR A:, DIR B: és a DIR E: parancsok hatása megegyezik.

ASSIGN

Az eredeti meghajtó-hozzárendeléseket állítjuk vissza, tehát minden betűvel a megfelelő meghajtóra hívatkozhatunk.

BACKUP

Feladata

Egyik mágneslemezen levő file-okat másolja át másik lemez(ek)re, archiválási célra. (Visszaolvasás csak a RESTORE paranccsal lehetséges.)

Paraméterezése

BACKUP [*m*:] [*út*] [*file-név* [.kit]] [*m*:]
[/S] [/M] [/A] [/D:*éé-hh-nn*] [/C]

Használata

A másolás (mentés) csak a PROPOS–16 rendszer alatt megformált lemezekre történhet. A parancssorban nem kötelező paramétert megadni, azokat a program megkérdezi.

Ha fix lemezről floppyra másolunk, a kimentett file-ok összmérete nagyobb is lehet egy floppy lemez kapacitásánál. A BACKUP program a céllemez megteltekor kéri a következő (megformált!) lemezt mindaddig, amíg van még kimentendő file. Ajánlatos a lemezre ráírni a sorszámukat, mert a sorrendet a visszatöltéskor be kell tartani!

A BACKUP végrehajtása során listát kapunk az átmásolt file-okról. Ne feledjük, hogy a kimentett file-ok formátuma eltér az eredetitől, ezért az így kapott lemezek nem alkalmasak közvetlen használatra. Tartalmukat a RESTORE paranccsal kell visszatölteni az eredeti (fix) lemezre.

A BACKUP által készített lemezen az elmentett file-ok floppy lemez esetén a gyökérkönyvtárba, fix lemez esetén a \BACKUP alkönyvtárba kerülnek. Alapértelmezés az A meghajtó.
A kimentendő információ megadására több lehetőségünk is van. A meghajtó kijelölése csak akkor hagyható el, ha a feltételezett meghajtó éppen a másolandó lemezegység. (Forrásmeghajtóként leggyakrabban a fix lemezes meghajtót adjuk meg.) Alapértelmezés a feltételezett meghajtó aktuális könyvtára.

A keresési *út* megadásával az aktuális könyvtártól eltérő (al)könyvtárat jelölhetünk ki bemenetként. Ezen belül megadhatunk egy file-t vagy file-csoportot is, különben az egész (al)könyvtár mentése végbemegy.

A bemeneti és a kimeneti meghajtó nem lehet azonos.

A /S paraméter lehetővé teszi, hogy a kijelölt (al)könyvtárral együtt valamennyi alárendelt (alacsonyabb szintű) alkönyvtár tartalmát is egyetlen paranccsal mentsük ki.

A /M paraméter szelektív mentést tesz lehetővé. Megértéséhez tudni kell, hogy a BACKUP program minden átmásolt file eredeti katalógusbejegyzésében jelzi a mentés megtörténtét. Ezt a jelzőt a file módosítását előidéző műveletek során a PROPOS–16 rendszer ismét átállítja, ezzel jelezve az utolsó mentés óta bekövetkezett változást. A /M megadása esetén a BACKUP parancs csak ezeket a módosított file-okat másolja a kimeneti lemezre.

A /A paraméter megadása szükséges akkor, ha egy (korábban már mentésre használt) lemezen levő file-ok mellé kívánunk további file-okat másolni. A /A elhagyása esetén a BACKUP parancs felülírja a céllemez, illetve a \BACKUP alkönyvtár korábbi tartalmát!

A /D paraméterrel is a kimentett file-ok köre szűkíthető. Minden file katalógusbejegyzése tartalmazza a file létrehozásának, illetve legutolsó módosításának dátumát. A /D után megadott dátummal az azóta létrehozott, illetve módosított file-ok kimásolását kezdeményezhetjük. A dátum kötött formában írandó, pl. 84-07-12 vagy 85-12-17 (l. a SET parancs leírásánál).

A /C paraméterrel a másolás utáni ellenőrzésre adhatunk utasítást.

A BACKUP parancs befejezési kódjai (batch alkalmazásban):

0 : normál befejeződés,
1 : az elmentendő file-ok nem találhatók a kijelölt lemezen vagy könyvtárban,
3 : hiba miatt félbeszakadt a mentés.

Példák

BACKUP

A feltételezett meghajtó a fix lemez; az aktuális könyvtár teljes tartalma mentésre kerül. A megformált, üres floppy lemez(eke)t az A meghajtóba kell helyezni (alapértelmezés). Ha a program kérdéseire válaszolva csak az ENTER -t ütjük le, kiírja az alapértelmezéseket:

```
C:\BACKUP

A forras filespec.: *.*
A cel meghajto [A:]: A:
Opciok (S A M D:ev-ho-nap C):
Helyezze a(z) 1. lemezt a(z) A meghajtoba!
Usson le egy billentyut!

\COMMAND.COM
\ANSI.SYS
   :
\SS.EXE
Helyezze a(z) 2. lemezt a(z) A meghajtoba,
es usson le egy billentyut!
\SS.EXE
\GRAFOS.COM
   :
```

BACKUP

BACKUP B:*.C /M E:

A B (hajlékony lemezes) meghajtó aktuális könyvtárából azok a .C kiterjesztésű file-ok (C nyelvű programok) kerülnek mentésre, amelyeket a legutóbbi BACKUP óta hoztunk létre vagy módosítottunk. A mentés az E fix lemezes meghajtóra történik, a \BACKUP alkönyvtárba.

BACKUP E: A: /SD:85-01-01

Az E meghajtó gyökérkönyvtárából és valamennyi alkönyvtárából minden olyan file-t kimentünk, amely 1985. január 1-én vagy utána keletkezett, illetve módosult.

A példákból látható, hogy a paraméterek felcserélhetők (kivéve a két meghajtóazonosítót), és egy / jel után több paraméter is felsorolható.

Ajánlatos elmenteni fontosabb file-jainkat minden nagyobb munkafázis (pl. negyedéves zárás) után, illetve a számítógép új helyre telepítése előtt!

Feladata

A batch file-ban tárolt parancssorozatot az operációs rendszer ugyanúgy hajtja végre, mintha a benne foglalt parancsokat egyenként, egymás után adnánk ki.

Paraméterezése

file-név [. BAT] [*paraméterek*]

Használata

A batch file-okat gyakran ismétlődő parancssorozatok tárolására célszerű létrehozni. A file tetszőleges programhívásokat (parancsokat) tartalmazhat. Az önállóan is használható parancsokon kívül néhány speciális batch alparanccsal is befolyásolhatjuk a file feldolgozásának menetét. A feldolgozás elindításakor paraméterek is megadhatók.

Batch file hívásánál sem a BATCH szót, sem a (kötelező) . BAT kiterjesztést nem kell megadni.

A file végrehajtása során az aktuális könyvtár kijelölése egy tárolt paranccsal meg is változtatható, ez a végrehajtást nem befolyásolja. Ha a batch file cserélhető lemezen van, a lemezt a parancssorozat teljes lefutásáig nem vehetjük ki a meghajtóból. A batch file végrehajtását lényegesen meggyorsíthatjuk, ha a file-t előzőleg a virtuális lemezre átmásoljuk.

A batch file utolsó parancsa tartalmazhatja egy másik batch file nevét, ezzel a parancsfile-ok egymás utáni feldolgozását válthatjuk ki.

A batch file végrehajtása a CTRL C billentyűk lenyomásával szakítható meg. Az éppen futó parancs végrehajtása megszakad, és a képernyőn megjelenik a kérdés:

Batch vége (i/n)? –

Ha válaszunk I [ENTER], a batch file éppen futó és még hátralévő parancsait a rendszer nem dolgozza fel. Ha a válasz N [ENTER], a feldolgozás a következő parancstól folytatódik.

A batch file-ban szereplő parancsok paramétereit a parancssorozat elindításakor, a file-név után adhatjuk meg. A paramétereket a file parancssoraiban a %0, %1, ... , %9 szimbólumokkal jelölhetjük, helyükre a végrehajtáskor helyettesítődik be a batch hívás parancssorában felsorolt paraméterek közül a megfelelő. A %0 paraméter értéke mindig a batch file-t tartalmazó meghajtó azonosítója és a batch file neve. A többi paraméternek a felsorolt paraméterek (legfeljebb 9) adnak értéket.

Kitüntetett jelentősége van a PROPOS–16 rendszerben az *AUTOEXEC.BAT* nevű batch file-nak. A gép bekapcsolásakor, illetve az operációs rendszer újbóli betöltésekor és elindításakor a PROPOS–16 mindig megnézi a rendszerlemez gyökérkönyvtárát, és ha talál benne AUTOEXEC.BAT nevű file-t, végrehajtja azt. Ezt a lehetőséget felhasználhatjuk állandó kezdeti feladataink elvégzésére, pl. aktuális alkönyvtár és keresési utak kijelölése (CD és PATH parancsok).

Példák

Hozzunk létre a COPY parancs segítségével egy batch file-t, amely paraméterezhető. A parancssorozat egy file-t másol, összehasonlítja a másolatot az eredetivel, és kinyomtatja.

```
COPY CON MASOLO.BAT
COPY %1 %2
CMP %1 %2 /L
PRINT %2
CTRL-Z (file vége)
```

Batch file-unk a feltételezett meghajtó aktuális könyvtárában található. Hívásakor két file-azonosítót kell megadnunk:

```
MASOLO FILE1 FILE2
```

A paraméterek értéke:

```
%0 = MASOLO
%1 = FILE1
%2 = FILE2
```

Ilyen paraméterezéssel a következő parancssorozat hajtódik végre:

```
COPY FILE1 FILE2
CMP FILE1 FILE2 /L
PRINT FILE2
```

A következő, PARANCS.BAT nevű batch file lehetővé teszi, hogy az általunk használt – és paraméterként felsorolt – könyvtárakból hívható parancsokat összefoglalva kiírassuk:

```
FOR %%D IN(%1 %2 %3) DO DIR %%D\*.BAT>>LISTA
FOR %%D IN(%1 %2 %3) DO DIR %%D\*.COM>>LISTA
FOR %%D IN(%1 %2 %3) DO DIR %%D\*.EXE >>LISTA
PRINT LISTA
```

Ennek a file-nak egy lehetséges lefuttatása a következő hívás:

```
PARANCS A: E: E:\KULKER,
```

amely az A és E meghajtó gyökérkönyvtárából és az E meghajtó KULKER alkönyvtárából íratja ki a parancsokat.

A fenti batch file működését a COPY, DIR és PRINT parancsok, a FOR batch alparancs, és az output átirányítás ismeretében érthetjük meg. A batch alparancsok ismertetésére a következőkben kerül sor.

Feladata

A batch file-ban tárolt parancssorok képernyőn történő megjelenítését vezérli. A parancsok végrehajtása során kiadott üzenetekre nincs hatással.

Paraméterezése

ECHO ON
vagy
ECHO OFF
vagy
ECHO [*üzenet*]

Használata

A rendszer indításakor (alapállapotban) minden végrehajtott batch parancs kiíródik a standard output perifériára. Az ECHO parancs **OFF** paraméterével a parancsok kiírása letiltható, az **ON** paraméterrel ismét engedélyezhető.

OFF állapotban a REM alparancsban szereplő üzenetek sem íródnak ki.

Ha a parancssorba tetszőleges *üzenet*et írunk, ez az ON vagy OFF állapottól függetlenül megjelenik a standard output periférián.

Ha az ECHO parancs paraméter nélkül szerepel a batch file-ban, végrehajtásakor az aktuális ECHO állapot (ON vagy OFF) íródik ki.

Példa

Legyen egy batch file tartalma az alábbi:

(1) ECHO OFF
(2) VOL
(3) ECHO ON
(4) VOL

A megszámozott parancsok hatására a batch file végrehajtásakor a képernyőn a következők jelennek meg:

```
(1)  echo off
(2)    "A" meghajtoban levo lemez : cimke nelkuli
(4)  vol
       "A" meghajtoban levo lemez : cimke nelkuli
```

Feladata

Lehetővé teszi egy PROPOS–16 parancs ismételt végrehajtását, változó paraméterekkel.

Paraméterezése

FOR *%%változó* **IN** (*értékkészlet*) **DO** *parancs*

Használata

A DO alapszó után megadott parancs többször kerül végrehajtásra. A parancsban szerepel egy változó is, két % jellel jelezve, amely az IN alapszó után zárójelben felsorolt és betűközzel elválasztott értékeket rendre felveszi. Az értékek karaktersorozatok és batch paraméterek (%1, %2 stb.) lehetnek, betűközzel elválasztva. Az értékkészlet file(csoport)-azonosítókat is tartalmazhat, és azokat útkijelölés is megelőzheti (teljes file-hívatkozás).

A FOR parancsok nem ágyazhatók egymásba – egy sorban csak egyszer szerepelhet a FOR szó.

A FOR parancsot önmagában (batch file-on kívül, közvetlen módon) is használhatjuk. Ekkor a ciklusváltozót csak egy % jellel kell azonosítani.

Példák

Az alábbi példákban csak egy-egy kiragadott batch parancssort vizsgálunk.

FOR %%N IN (1 2 3) DO PRINT LISTA

A batch parancs hatására a LISTA file háromszor kinyomtatódik. Az N változónak csak az ismételtetésben van szerepe.

FOR %%F IN (\KULKER \KULKER\EXPORT) DO DIR %%F

Ez a batch parancs a felsorolt alkönyvtárakat listázza ki, vagyis a következő parancsokat hajtja végre:

DIR \KULKER
DIR \KULKER\EXPORT

FOR %%LIB IN (%1 %2 %3) DO DIR %%LIB

A parancs a batch file hívásakor paraméterként megadott könyvtárak tartalmát írja ki.

Batch
GOTO alparancs

Feladata

Átadja a vezérlést a parancsban megadott címkével jelölt másik sorra.

Paraméterezése

GOTO *cimke*

Használata

Lehetőségünk van arra, hogy egy batch file sorait ne a leírás sorrendjében hajtassuk végre, illetve magunk szervezzünk ismétlődő parancsszekvenciákat. Ennek érdekében a batch file-ban címkéket helyezhetünk el, amelyek legfeljebb 8 karakter hosszúak, és kötelezően kettőspont (:) előzi meg őket. A GOTO paranccsal a címkével jelzett sortól folytathatjuk a batch file végrehajtását.

A cimke sorában az első kilenc karakter utáni részt a PROPOS–16 rendszer nem veszi figyelembe, és nem is írja ki a végrehajtás során, ezért itt tetszőleges magyarázó megjegyzéseket (kommentárokat) helyezhetünk el.

Példa

```
IF %1==%2 GOTO HIBA
COPY %1 %2
GOTO VEGE
:HIBA        Azonos névvel nem lehet másolni
REM  file-név hiba
:VEGE
```

A fenti példában egy file-másoló batch parancssorozatban megvizsgáljuk, hogy a kapott két paraméter egyenlő-e. Ha igen, a másolás nem végezhető el. A :HIBA cimke sorában (a 10. pozíciótól) magyarázó szöveget helyeztünk el. Ez a képernyőn nem jelenik meg, a hibajelzést a REM parancs hatására kapjuk.

Feladata

Lehetővé teszi, hogy a batch file végrehajtása során valamilyen feltételtől függően hajtsunk végre parancsokat.

Paraméterezése

IF [NOT] *feltétel parancs*

Használata

A **NOT** szóval a kijelölt feltétel nem teljesülése, különben pedig teljesülése esetén hajtódik végre a parancs. A parancs tetszőleges programhívás, PROPOS–16 parancs vagy batch alparancs lehet.

A *feltétel* három alakú lehet:

1. **szöveg1 == szöveg2**

A feltétel akkor teljesül (igaz), ha a két szöveg azonos (a kis- és nagybetűk itt különbözőnek számítanak). A feltételben mindig két egyenlőségjelet kell írni! A szövegek lehetnek karaktersorozatok vagy batch paraméterek (%1, %2 stb.).

2. **ERRORLEVEL szám**

A feltétel akkor teljesül, ha az utoljára lefutott program befejezési kódja a megadott számmal egyenlő vagy annál nagyobb. A számot decimálisan kell megadni.

A befejezési kódot minden program beállítja. A PROPOS–16 parancsokat végrehajtó, e kötetben ismertetett programok a befejezési kódot 0-ra állítják, ha futásuk eredményes volt, és 1-re, ha hiba volt a végrehajtásban. Néhány parancs (CMP, BACKUP, RESTORE) ennél differenciáltabban jelez vissza – hibajelzésük módja a parancsok leírásánál megtalálható.

3. **EXIST** [*m*:] [*út*] *file-név* [.*kit*]

A feltétel akkor teljesül, ha létezik ilyen azonosítójú file a kijelölt (vagy feltételezett) meghajtó megadott (vagy aktuális) könyvtárában. A feltételben file-csoport-hivatkozás is megadható.

Példák

```
IF %1==P PRINT %2
IF %1==T TYPE %2
```

A batch parancs első paramétere P vagy T lehet, és ettől függően a második paraméterben kijelölt file-t a nyomtatón vagy a képernyőn jelenítjük meg.

```
:
CMP %1 %2 /S
IF NOT ERRORLEVEL 1 DEL %1
:
```

A fenti batch file-részlet törli az első paraméterrel specifikált file-t, ha az azonos a második file-lal (a CMP parancs befejezési kódja 0).

```
:
IF NOT EXIST A:%1 GOTO CIMKE
:

:CIMKE
:
```

A parancssorozat egy részét kihagyjuk a feldolgozásból, ha az első paraméterrel kijelölt file nem található az A meghajtón.

Feladata

Felfüggeszti a feldolgozást, kiír egy üzenetet a standard outputra, és egy billentyű lenyomására vár.

Paraméterezése

PAUSE [*üzenet*]

Használata

A PAUSE parancs a batch feldolgozás felfüggesztésére, operátori teendők elvégeztetésére használható. A folytatást tetszőleges billentyű (kivéve a vezérlőbillentyűk, pl. [CTRL] [C]) lenyomásával kérhetjük. A végrehajtást abbahagyathatjuk a [CTRL] [C] billentyűk lenyomásával.

A paranccsal üzenet is kiíratható, amely legfeljebb 121 karakter hosszú lehet.

Példa

```
:
PAUSE Tegye az A meghajtóba az EXPORT lemezt!
:
```

Amikor a batch file végrehajtása ehhez a sorhoz ér, futása megszakad, és a képernyőn az alábbiak jelennek meg:

```
PAUSE Tegye az A meghajtoba az EXPORT lemezt !
Usson le egy billentyut, ha folytathatjuk . . .
```

Batch
REM alparancs

Feladata

A batch file végrehajtása közben üzenetet ír ki.

Paraméterezése

REM *üzenet*

Használata

Amikor a file végrehajtása a REM parancshoz ér, a sorban szereplő üzenet megjelenik a képernyőn. Az üzenet legfeljebb 123 karakter hosszúságú lehet. A REM parancs egyaránt használható a batch file tartalmának magyarázatára (kommentálás), és a gépkezelő tájékoztatására az éppen folyó műveletekről.

A REM üzenetek kiírása az ECHO OFF paranccsal letiltható.

Példa

REM File összehasonlítás kezdődik

Amikor a batch file végrehajtása ehhez a sorhoz ér, a képernyőn megjelenik a megadott üzenet.

Feladata

Lehetővé teszi, hogy egy batch parancssor paramétereit átcsoportosítsuk. Segítségével tíznél több paraméter is használható egy parancsban.

Paraméterezése

SHIFT

Használata

A batch parancssorban megadható paramétereket a batch file-ban a %0, %1, ... , %9 szimbólumokkal jelöljük. A SHIFT parancs segítségével a paramétereket eggyel balra „tolhatjuk", azaz a %0 felveszi a %1 értékét, a %1 felveszi a %2 értékét stb. Ha a parancssorban tíznél több paraméter van, a %9 felveszi a 11. paraméter értékét. További SHIFT parancsokkal további paraméterek is felhasználhatók.

Példák

```
FOR %%I IN (%2 %3 %4 %5 %6 %7 %8 %9) DO COPY %1+%%I
SHIFT
COPY %0+%9
:
```

A batch hívásakor tíz file-azonosítót sorolhatunk fel, ezeket az első azonosító alatt egyesítve állítja elő ez a parancssorozat.

```
:CIKLUS
CMP %1 %2
SHIFT
IF NOT ERRORLEVEL 1 GOTO CIKLUS
```

A paraméterként felsorolt azonosítójú file-okat páronként összehasonlítjuk mindaddig, amíg eltérést nem találunk. A SHIFT parancs hatására először az első file-t a másodikkal, majd a másodikat a harmadikkal stb. hasonlítjuk össze.

CHDIR

Feladata

Kijelöli (megváltoztatja) az aktuális könyvtárat, illetve tájékoztat az aktuális könyvtárhoz vezető útról.

Paraméterezése

CHDIR [*m*:] [*út*]

vagy

CD [*m*:] [*út*]

Használata

Ha a parancsot paraméterek nélkül vagy csak a meghajtó kijelölésével adjuk ki, a standard output perifférián megjelenik a feltételezett, illetve kijelölt meghajtóhoz rendelt aktuális könyvtárhoz vezető út teljes leírása.

Ha a parancsban megadunk útleírást, ezzel az adott meghajtóhoz tartozó új aktuális könyvtárat határozzuk meg. Az út leírása akár a gyökérkönyvtárból, akár az aktuális könyvtártól kiindulva megadható.

Példák

CD

A feltételezett meghajtó aktuális könyvtárához vezető utat íratjuk ki.

CHDIR A:

Az A meghajtóhoz rendelt aktuális könyvtár leírását kérjük.

CD \

A feltételezett meghajtón a gyökérkönyvtárat tesszük aktuális könyvtárrá.

CD B:\TEST

A B meghajtón a gyökér alatti TEST alkönyvtárat tesszük aktuálissá.

CD ..\ADAT

A feltételezett meghajtó aktuális könyvtárával azonos szintű (ugyanazon felsőbb szintű könyvtár alá rendelt) ADAT alkönyvtár lesz az új aktuális könyvtár.

Feladata

Kijelzi egy file vagy file-csoport módosítható attributum bitjeit (csak olvasható, rejtett, illetve rendszerfile), valamint lehetővé teszi ezen bitek módosítását.

Paraméterezése

CHMOD [*m*:] [*út*] *file-név* [.*kit*] [/I] [/S] [/H] [/R]

Használata

Minden file könyvtári bejegyzésében található egy ún. attributum byte, amelynek alsó hat bitje a file jellemzőit írja le:

bit pozició	bit jelentése	
	0	1
0	írható/olvasható	csak olvasható
1	nem rejtett file	rejtett file
2	nem rendszerfile	rendszerfile
3	nem lemezazonosító	lemezazonosító
4	file	alkönyvtár
5	nem változott	megváltozott
6	–	–
7	–	–

Az attributum bitek közül csak a 0–2. pozición állók esetében van értelme annak, hogy a felhasználó megváltoztassa értéküket. A 3–4. bit értéke értelemszerűen nem változtatható meg, az 5. bitet pedig a file írásakor a PROPOS–16 rutinjai állítják 1-re, és a BACKUP parancs törli. A CHMOD parancs segítségével a 0–2. bitek kérdezhetők le vagy állíthatók be egy file vagy file-csoport könyvtári bejegyzésében.

A parancsot az opcionális paraméterek nélkül kiadva a kijelölt file(ok) attributum bitjeiről kapunk tájékoztatást. File-csoport megadása esetén a rejtett és a rendszerfile-okról nem kapunk információt, ezek csak egyedi azonosítójuk megadásával kérdezhetők le.

CHMOD

Filespec	Attributum
A:PRODOS.COM	read-only hidden system

A /S paraméterrel a megadott file-t vagy file-csoportot rendszerfile-lá, illetve file-okká tehetjük.
A /H paraméter az adott file(ok)ból rejtett file-t csinál.
A /R paraméter segítségével a kijelölt file(ok) csak olvashatóvá, azaz írásvédetté tehető(k).
A /I paramétert a fenti három paraméter valamelyike előtt adhatjuk ki, ekkor annak, és az azt követő paramétereknek a hatását ellenkezőjére változtatja (a megfelelő attributum bitet nullázza).

Példák

CHMOD ADAT.84

A megadott file attributum bitjeit kérdezzük.

CHMOD A:BERLISTA.* /SHIR

A BÉRLISTA file-okat rejtett rendszerfile-okká tesszük, így ezek a továbbiakban nem látszanak majd a könyvtár tartalomjegyzékében. Írásvédettségüket feloldjuk.

CHMOD E:\BER\ALAP /IR

Az E meghajtó BER alkönyvtárában tárolt ALAP nevű file írásvédettségét feloldjuk, mert a file-ban módosításokat akarunk végezni.

Feladata

Két file-t hasonlít össze byte-onként, és kijelzi az eltéréseket.

Paraméterezése

CMP [*m*:] [*út*] *file-név* [. *kit*] [*m*:] [*út*] *file-név* [. *kit*] [/**L**] [/**S**]

Használata

Az összehasonlítandó két file különböző meghajtókon és különböző könyvtárakban is lehet. Leírásukból elhagyható a meghajtó és az út specifikálása, ha a feltételezett meghajtó, illetve az aktuális könyvtár tartalmazza a file(oka)t.

Az egyik file-név helyén " – " jelet írva, az olvasás a standard inputról történik.

File-csoport-azonosító nem adható meg.

Ha valamelyik file nem található a könyvtárban, vagy a keresési út megadása helytelen, hibajelzést kapunk.

Az összehasonlítás eredményeképpen a file-párból csak az első eltérő byte-ot kapjuk meg. Ha a két file közötti valamennyi eltérést ki akarjuk íratni, a /**L** opciót kell megadnunk a parancssorban. (Vigyázat, a lista így hosszúra nyúlhat!)

Az eltérések helyét a file elejéhez képest (offset) decimálisan, valamint a byte tartalmát hexadecimálisan kapjuk meg.

A /**S** paraméterrel a program output listáját letilthatjuk (pl. batch alkalmazásnál).

A CMP parancs befejezési kódjai (batch alkalmazásokhoz):

0: a két file egyező
1: a két file különböző, vagy hiba volt.

CMP

Példák

CMP A:FILE1.BIN B:MENTÉS.JAN

A paranccsal két, különböző meghajtó aktuális könyvtárában levő, eltérő azonosítójú file-t hasonlítunk össze.

```
CMP E:\BASIC\JATEK.BAS  B:JATEK.BAS  /L
        3:        E8  62
        5:        28  26
       15:        CF  87
       19:        19  68
```

A két azonos nevű, különböző meghajtókon levő file-t összehasonlítva minden eltérő byte-ot kiíratunk.

Feladata

Két szövegfile-t hasonlít össze, és kijelzi a megegyező és eltérő sorokat. A két file-ban a sorok tetszőleges sorrendben lehetnek, ez az egyezőség szempontjából közömbös.

Paraméterezése

COMM [*m*:][*út*]*file-név* [.*kit*] [*m*:][*út*]*file-név* [.*kit*] [/1] [/2] [/3]

Használata

A COMM parancs paramétereként két szövegfile-hivatkozás megadása kötelező. File-csoport-azonosítók nem megengedettek. A "–" jellel a standard inputra hivatkozhatunk. A parancs a két file-t soronként hasonlítja össze úgy, hogy az első file minden sorát összehasonlítja a második file minden sorával.

A program outputja egy lista a standard output berendezésen (természetesen átirányítható máshova is).

A lista 1. oszlopában azok a sorok szerepelnek, amelyek csak az első file-ban szerepelnek, 2. oszlopában azok, amelyek csak a második file-ban, és a 3. oszlopában azok, amelyek mindkét file-ban előfordulnak.

A /1, /2 és /3 paraméterekkel az output-lista megfelelő oszlopai letilthatók. Mindhárom oszlop is letiltható, ekkor output-lista nem készül (pl. batch alkalmazásnál), csak a visszatérési kódot állítja be a program.

A COMM parancs által készített lista sorainak hossza a vizsgált file-ok rekordhosszának függvénye – ezért előfordulhat, hogy a képernyőn vagy a nyomtatón nem fér el egy sorban!

COMM

Példák

COMM NEVSOR.EGY NEVSOR.KET

Két névsort hasonlítunk össze. Listát kérünk az egyező és a csak egyik vagy másik file-ban szereplő nevekről is.

```
NEVSOR.EGY egyedi sorai        NEVSOR.KET egyedi sorai        Kozos sorok

Antal Imre
                                                              Bartok Bela
Csortos Gyula
                                                              Devecseri Gabor
                                                              Eszterhazy Peter
Fekete Gyula
Gabor Andor
Hernadi Gyula
                               Fekete Istvan
                               Kodaly Zoltan
                               Liszt Ferenc

PARANCS VEGREHAJTVA !  USSON LE EGY BILLENTYUT !
```

COMM A:CIMLISTA E:\POSTA\CIMLISTA /3

Két CIMLISTA nevű szövegfile-unk van, egyik az A meghajtó aktuális könyvtárában, másik az E meghajtó POSTA alkönyvtárában. Az összehasonlítás eredményeként csak azon címek listáját kérjük, amelyek nem közösek a két file-ban.

COMM A:CIMLISTA E:\POSTA\CIMLISTA /12

A két file összehasonlítása során csak az egyező címeket íratjuk ki.

Feladata

Tájékoztatást ad a konfiguráció hardware jellemzőiről, értelmezi a CONFIG.SYS file tartalmát, és lehetővé teszi annak módosítását.

Paraméterezése

CONFIG

Használata

A paraméterek nélkül kiadandó parancs interaktív üzemmódban működik. Először a hardware konfigurációról ad tájékoztatást:

```
A HARDWARE KONFIGURACIO

Floppy illesztesek szama : 4
Nyomtatok szama          : 1
RS232 portok szama       : 1
Kozponti tar merete (Kbyte) : 256
```

A felsorolásban a központi egység hardware *lehetőségei* – nem pedig a tényleges konfiguráció – szerepelnek. A 4 lemezegység tehát a kezelhető floppy meghajtók maximális száma az adott gépen.

Függetlenül a ténylegesen csatlakoztatott meghajtók számától, ebben a konfigurációban a floppy meghajtók az A, B, C és D azonosító betűket kötik le.

Hasonlóképpen a nyomtatók és az RS232 portok (aszinkron csatolók) száma is a maximális lehetőséget jelenti.

Ezeket a jellemzőket nem módosíthatjuk software eszközökkel.

CONFIG

A következőkben a CONFIG program az operációs rendszer változtatható paramétereiről ad információt. A PROPOS–16 paraméterezését a rendszerlemez gyökérkönyvtárában elhelyezett *CONFIG.SYS* azonosítójú file-ban tárolt speciális parancsok segítségével végezhetjük. A CONFIG parancs ezt a file-t olvassa, szövegesen értelmezi, és felkínálja a módosítás lehetőségét.

A CONFIG.SYS file-t az operációs rendszer betöltésekor, a PROBIO modul olvassa és interpretálja. Módosításának hatása tehát csak az operációs rendszer legközelebbi indításakor érvényesül.

A CONFIG.SYS file-t a CONFIG parancstól függetlenül, bármely szövegszerkesztő programmal (l. EDIT, 5. fejezet) is kiírathatjuk és módosíthatjuk.

A CONFIG paranccsal módosítható opciók:

BREAK

A CONFIG.SYS file-ban a BREAK=OFF vagy BREAK=ON parancsok egyike szerepelhet. OFF esetén a [CTRL] [BREAK] ([CTRL] [C]) billentyűk lenyomását csak konzol, nyomtató és aszinkron kommunikációs tevékenység alatt figyeli az operációs rendszer. Az ON esetén a rendszer minden tevékenysége során figyeli a [CTRL] [BREAK] lenyomását. Az alapértelmezés az OFF állapot. A CONFIG parancs kiírja a pillanatnyi állapotot. Ha nem kívánjuk módosítani, csak az [ENTER] -t nyomjuk le.

A BREAK állapot a SET paranccsal is módosítható.

BUFFERS

A CONFIG.SYS file-ban BUFFERS=x alakú paranccsal írhatjuk elő, hogy az operációs rendszer betöltésekor hány puffert allokáljon a blokkos input-output műveletek számára.
A CONFIG parancs kiírja a pufferek aktuális számát, és módot ad új szám megadására. A pufferek száma alapértelmezésben 2, és legfeljebb 99 adható meg. Ha nem akarunk változtatni, csak az ENTER -t kell lenyomni. A központi tárban minden puffer 528 byte területet foglal le. A helyfoglaláson kívül figyelembe vehető szempontok még a pufferek számának meghatározásánál:

- olyan alkalmazásoknál, amelyek közvetlen (random) file-hozzáféréssel dolgoznak, a programok hatékonysága javítható a pufferek számának növelésével;
- szekvenciális file-kezelésnél a több puffer nem jelent előnyt;
- sok file egyidejű használata esetén a pufferek nagyobb számának lehet sebességnövelő hatása.

A pufferek optimális száma általában csak kísérletezéssel határozható meg. A legtöbb alkalmazás 10–20 közötti számú pufferrel hatékonyan működik.

A rendszerpufferekről a **Programozóknak** címet viselő kézikönyvből tudhatunk meg többet.

Az operációs rendszer saját, belső file-leíró blokkjainak számát FILES=*y* alakú paranccsal adhatjuk meg.

y értéke azt határozza meg, hogy a PROPOS–16 egyidejűleg hány nyitott file-t tud kezelni. A file-ok száma alapértelmezésben 8, és legfeljebb 99 adható meg. Ha nem akarunk változtatni, csak az ENTER -t kell lenyomni. Az egyidejűleg megnyitandó file-ok száma általában nem több 20-nál.

A file-leíró blokkokról a **Programozóknak** című kézikönyvből tudhatunk meg többet.

DEVICE

A PROPOS–16 lehetővé teszi, hogy speciális berendezéseink kezeléséhez a szükséges programot (device driver) betöltésekor beillesszük az operációs rendszerbe. Ehhez a CONFIG.SYS file-ban.

DEVICE=[*m*:] [*út*] *file-név* [.*kit*]

alakban kell specifikálni a perifériakezelő programokat.

A perifériakezelő programok írásának szabályait a **PROPOS–16 Programozóknak** című kézikönyv ismerteti. A PROPOS–16 rendszerhez két szabványos perifériakezelő program tartozik, amelyek a rendszerlemez gyökérkönyvtárában találhatók:

ANSI.SYS: kiterjesztett képernyő- és billentyűzetkezelést lehetővé tevő program. Használata a PROPOS–16 V.3.0. vagy ennél nagyobb verziószámú rendszerekben kötelező.

VDISK.SYS: virtuális lemez(ek) kezeléséhez szükséges program. A virtuális lemez installálásához szükséges paramétereket a CONFIG program kérdéseire válaszolva adhatjuk meg:

- a virtuális lemez kapacitása (Kbyte), alapértelmezése 64;
- könyvtárbejegyzések maximális száma (alapértelmezés: 64);
- a virtuális lemez szektormérete – a valódi lemezekhez hasonlóan – mindig 512 byte.

Végül a szabványos perifériakezelő modulokon kívül továbbiak megadására is lehetőséget nyújt a CONFIG program. Ha ilyenek nincsenek, az [ENTER] lenyomásával befejezhetjük a parancs végrehajtását.

Példa

A következő két párbeszéd a CONFIG program egy-egy futását illusztrálja.

CONFIG

```
A HARDWARE KONFIGURACIO

Floppy illesztok szama : 4
Nyomtatok szama        : 1
RS232 portok szama     : 1
Központi tar merete (Kbyte) : 256

AZ OPERACIOS RENDSZER JELLEMZOI

A programvegrahajtas barmikor megszakithato.
BREAK=ON
Uj ertek: OFF
A lemezmuveletekhez hasznalt pufferek szama :
BUFFERS=22
Uj ertek:
Az egyidejuleg megnyithato file-ok max. szama :
FILES=16
Uj ertek:
Kiterjesztett kepernyokezeles letezik.
DEVICE=ANSI.SYS
Virtualis lemez nincs.
Szukseges? (i/n) : i
A virtualis lemez merete= 64
File-ok szama (max.)= 12
Uj driver =
Az uj konfiguracio a kovetkezo rendszerinditaskor lep eletbe!
```

```
A HARDWARE KONFIGURACIO

Floppy illesztok szama : 4
Nyomtatok szama        : 1
RS232 portok szama     : 1
Kozponti tar merete (Kbyte) : 256

AZ OPERACIOS RENDSZER JELLEMZOI
A programvegrehajtas csak konzol- es kepernyomuveleteknel szakithato meg.
BREAK=OFF
Uj ertek:
A lemezmuveletekhez hasznalt pufferek szama :
BUFFERS=22
Uj ertek. 8
Az egyidejuleg megnyithato file-ok max. szama :
FILES=16
Uj ertek: 24
Kiterjesztett kepernyokezeles letezik.
DEVICE=ANSI.SYS
Virtualis lemez van.
A virtualis lemez azonositoja : F
Kapacitasa                    : 64512 byte
Fizikai szektor merete        : 512
File-ok szama (max.)          : 16
Ervenyes? (i/n) : i
Uj driver =
Az uj konfiguracio a kovetkezo rendszerinditaskor lep eletbe!
```

Feladata

.EXE kiterjesztésű és ennek megfelelő formátumú, futtatható programfile-t alakít át .COM kiterjesztésű és ennek megfelelő formátumú file-lá.

Paraméterezése

CONVEXE [*m*:] [*út*] *file-név* [.**EXE**] [*m*:] [*út*] *file-név* [.**COM**]

Használata

Az elsőként megadott file-hivatkozás a parancs bemeneteként szolgáló .EXE file. A kiterjesztés leírása nem kötelező, a feltételezés .EXE, más kiterjesztés vagy nem megfelelő file-formátum esetén hibajelzést kapunk.

A második file-hivatkozás a kimenő file-t specifikálja. Elhagyása esetén a parancs az eredetivel megegyező nevű, .COM kiterjesztésű file-t hoz létre a feltételezett meghajtó aktuális könyvtárában. A kimeneti file-hivatkozásban csak .COM kiterjesztés adható meg.

File-csoport-nevek nem használhatók.

.EXE kiterjesztésű és ennek megfelelő formátumú a LINK programszerkesztővel előállított minden futtatható program. Az átalakíthatóság egyik előfeltétele, hogy a program mérete 64 Kbyte-nál kisebb legyen és STACK (verem) szegmenst ne tartalmazzon.

A program kezdőpontja (a CS:IP regiszterek kezdőértéke) 0:0 vagy hexadecimális 0:100 lehet.

Az így létrehozott .COM file mérete kisebb, betöltése gyorsabban megy végbe.

A parancs használatára ritkán, főleg assembler nyelvű rendszerprogramok készítésénél lehet szükség.

Példák

CONVEXE TESTPROG

A feltételezett meghajtó aktuális könyvtárában tárolt, TESTPROG.EXE azonosítójú file-ból ugyanebben a könyvtárban létrehozzuk a TESTPROG.COM azonosítójú programfile-t.

CONVEXE TESTPROG E:\RAKTAR

A TESTPROG.COM file-t az E meghajtó RAKTAR alkönyvtárában helyezzük el.

CONVEXE A:TESTPROG.EXE E:\RAKTAR\SZAMLA.COM

Az A meghajtó aktuális könyvtárából véve az input file-t, az E meghajtó RAKTAR alkönyvtárában új néven hozzuk létre az átalakított, .COM kiterjesztésű file-t.

COPY

Feladata

Egy vagy több file másolását, illetve egyesítését teszi lehetővé. Az új file azonosítója az eredetitől eltérő is lehet.

Paraméterezése

COPY [*m*:] [*út*] *file-név* [.*kit*] [/A] [/B]
[+ [*m*:] [*út*] *file-név* [.*kit*] [/A] [/B] ...]
[*m*:] [*út*] [*file-név* [.*kit*]] [/A] [/B] [/C] [/M] [/P]

Használata

A COPY parancs igen sokféleképpen használható file-ok vagy file-csoportok mozgatására. A másolást vagy egyesítést végezhetjük egy másik file-ba (amely másik meghajtón és/vagy másik könyvtárban lehet) vagy valamelyik standard perifériára (l. A. függelék). A másolás után az eredeti file(oka)-t megtarthatjuk vagy törölhetjük.

A BACKUP parancs működésétől eltérően a COPY nem teszi lehetővé könyvtárkapcsolatok átvitelét, viszont a másolat az eredetivel teljesen megegyező formátumú lesz.

A parancsban az *első file-hivatkozás* a bemenő file-t vagy file-csoportot határozza meg. Standardperiféria-név is megadható. Ha több file vagy file-csoport egyesítését (konkatenálását) kérjük, a felsorolt forrásokat a + jellel kell összekapcsolni.

A *második file-hivatkozás* a kimenő file-t, file-csoportot vagy standard perifériát jelöli ki. Teljes elhagyása esetén a feltételezett meghajtó aktuális könyvtárába, az eredetivel megegyező névvel és kiterjesztéssel másolódnak a file-ok. Meghajtó és/vagy út megadásával lemezek, illetve könyvtárak közötti file-átvitelt valósíthatunk meg. A file-ok önmagukra nem másolhatók át.

Az átmásolt file-ok nevét és kiterjesztését a program folyamatosan kiírja a standard output perifériára – kivéve, ha valamelyik file a CON berendezés. A listán azt is jelzi, hogyha a másolattal felülírunk egy azonos nevű és kiterjesztésű file-t.

Csak olvasható (read-only) file-ok másolata már írható/olvasható jelzéssel lesz ellátva.

Rejtett file-ok és alkönyvtárak a COPY-val nem másolhatók.

A **/A** és **/B** paraméterekkel a file-ok típusát határozzuk meg, és egyben a másolás mennyiségét is befolyásolhatjuk. A két paraméter használata egymást kizárja egy file-ra vonatkozóan. Hatásuk a *megelőző* file-azonosítóra, és *minden további* file-azonosítóra kiterjed, amíg valamelyik után ismét nem specifikáljuk a file típusát. Jelentésük, illetve hatásuk:

/A (alfanumerikus): *bemenő* file specifikálásánál a file-t szövegfile-nak tekinti a program, a másolás végét a file-ban talált első file-végjel (CTRL–Z, azaz hexadecimális 1A) jelzi; *kimenő* file esetében az új file végére CTRL–Z végjelet tesz. /A az alapértelmezés, ha file-egyesítést végzünk vagy ha a file-nevek bármelyike foglalt név (standardperiféria-azonosító).

/B (bináris): az így minősített *bemenő* file teljes egészében, a könyvtár bejegyzésében megadott hosszúságban kerül másolásra; a bináris *kimenő* file végére nem kerül

CTRL–Z karakter. /B az alapértelmezés, ha egyesítés nélkül végzünk másolást mágneslemezes file-ból mágneslemezes file-ba.

Az egész COPY parancsra kiterjedő hatállyal adhatók meg az alábbi paraméterek:

/C hatására az átmásolt file(oka)-t a program visszaolvassa, és összehasonlítja az eredeti file-lal vagy file-okkal, vagyis automatikusan végrehajt egy CMP parancsot is. A paraméter megadása értelmetlen, ha akármelyik file-név helyén foglalt nevet (standard perifériát) jelöltünk ki.

/M paraméterrel kérhetjük az eredeti file törlését. Úgy is felfoghatjuk, hogy az adott file-t vagy file-okat új helyre mozgattuk ahelyett, hogy másolatot készítettünk volna róluk.

/P megadásával kérhetjük, hogy a parancs végrehajtása során ne csak listát kapjunk az átmásolt file-okról, hanem egyenként engedélyezhessük a file-ok másolását. A paraméter használatának csak file-csoport másolásakor van értelme. Segítségével megakadályozhatjuk a file-ok felülírását.

Mágneslemezes file-ok átmásolásakor az új file a bemenő file dátumát kapja meg. File-egyesítéskor az eredményfile keletkezési dátuma a napi dátum.

Példák

COPY A : PROGRAM

Az A meghajtóban levő PROGRAM file-t a feltételezett meghajtó aktuális könyvtárába másoljuk PROGRAM néven, bináris (/B) feltételezéssel.

COPY *.* A: /P

A feltételezett meghajtó aktuális könyvtárából minden file átmásolását kérjük az A meghajtó aktuális könyvtárába. A parancs végrehajtása során a képernyőn (standard output periférián) megjelennek a másolásra kerülő file-nevek, az esetleges felülírásra figyelmeztető üzenettel. A /P paraméter megadása következtében lehetőségünk van a másolás file-onkénti engedélyezésére (I ENTER: másolható, N ENTER: nem másolható a file):

```
Masolando file          Masolat file
SET5.C          A :SET5.C        Masolhato (i/n)? i
CONIO.C         A :CONIO.C       Masolhato (i/n)? i
CONVEXE.C       A :CONVEXE.C     Masolhato (i/n)? n
DCOPY.C         A :DCOPY.C       Masolhato (i/n)? i
TYPE.C          A :TYPE.C        Masolhato (i/n)? n
MORE.C          A :MORE.C        Masolhato (i/n)? n
  :
```

COPY \LIB1\PROGRAM \LIB2 /C

A feltételezett meghajtón a LIB1 alkönyvtárból a PROGRAM file-t átmásoljuk a LIB2 alkönyvtárba, és kérjük az átmásolt file összehasonlítását az eredetivel. A file a lemez mindkét alkönyvtárában azonos néven fog szerepelni! (Egy könyvtáron belül azonos névvel nem másolhatunk!)

A fenti példa új értelmezést kap, ha a lemezen nincs LIB2 nevű alkönyvtár. Ekkor a COPY program a második paramétert úgy értelmezi, hogy a másolatot a gyökérkönyvtárban kell létrehozni, LIB2 néven!

COPY ADAT?.84 *.85

Ha az átmásolt file(ok)-nak új nevet adunk, a másolást egy könyvtáron belül is elvégezhetjük. A fenti paranccsal több file-t másolunk (pl. ADAT1, ADAT2, ...), és a másolatok azonosítójában azonos file-név mellett új kiterjesztést adunk meg. A könyvtárban lesz tehát ADAT1.84 és ADAT1.85 azonosítójú file is, azonos tartalommal.

COPY TEST.COM PROG.* /M

Ezzel a paranccsal új nevet adunk a másolt file-nak, és az eredeti file-t nem tartjuk meg (M paraméter!). Mivel egy (az aktuális) könyvtáron belül végeztük a műveletet, hatása azonos az átnevezést végrehajtó RENAME parancséval.

COPY \TEST.COM \LIB1\PROG.* /M

Most az átnevezéssel egyidejűleg másik könyvtárba is került a file: a gyökérkönyvtárból a LIB1 alkönyvtárba.

COPY B:\TEST.COM A:\LIB1\PROG.* /M

Az előzőhöz hasonló parancs hatására most tényleges másolás megy végbe a B meghajtóról az A meghajtóra, és az eredeti file törlődik a B meghajtó gyökérkönyvtárából.

Formailag a mágneslemezes file-másoláshoz hasonlóan írhatók fel azok a COPY parancsok, amelyekben egyik vagy mindkét file-hivatkozás helyén foglalt file-név, tehát standardperiféria-azonosító áll:

COPY CON TEXTFILE

hatására a TEXTFILE-ba (a feltételezett meghajtó aktuális könyvtárába) írhatunk közvetlenül a billentyűzetről;

COPY LISTPROG.PAS LPT1

paranccsal PASCAL programunk listáját kinyomtathatjuk;

COPY EREDMENY CON

paranccsal a file tartalmát képernyőre írathatjuk;

COPY CON PRN

kiadásával a bebillentyűzött sorokat a nyomtatóra küldhetjük. Ne feledjük, hogy a billentyű inputot a [CTRL] [Z] (vagy az [F6] funkcionális billentyű) és az [ENTER] lenyomásával zárhatjuk le!

Természetesen a NUL azonosítót is használhatjuk, ennek azonban – tesztelésen kívül – nincs különösebb értelme.

COPY X.LST+Y.LST E:AB.LST

A feltételezett meghajtó aktuális könyvtárában tárolt, X és Y nevű listafile-okat egyetlen file-ba másoljuk az E meghajtón, AB.LST azonosítóval.

COPY A:PROG1.BAS + B:PROG2.BAS + A:PROG3.BAS

Ha második (output) file-azonosítót nem adunk meg a parancsban, az egyesített file az első input file helyén, azzal megegyező azonosítóval jön létre. A PROG1.BAS file a három eredeti file egyesítését fogja tartalmazni, tehát megváltozik, míg a másik két file eredeti tartalommal marad meg.

Az *egyesítés alapértelmezésben mindig alfanumerikus* módban (/A paraméter feltételezésével) megy végbe, vagyis minden inputfile-t az első CTRL–Z karakterig olvas a program, és az eredményfile-t CTRL–Z karakterrel zárja le. Ha ettől eltérő módon akarunk file-okat egyesíteni, meg kell adnunk a /B paramétert. Természetesen kombinálhatjuk is ezeket a paramétereket:

COPY PR1.BAT + TEXT.ABC/B + A:LISTA/A EGYFILE

ahol a TEXT.ABC file-t binárisnak, az összes többit alfanumerikusnak értelmezi a program. PR1.BAT esetében a /A az alapértelmezés, mivel egyesítést végzünk, és eddig még nem szerepelt a sorban /B opció. A LISTA file-nál specifikáltuk a /A értelmezést, és ennek hatása a következő file-hivatkozásra, az EGYFILE nevű eredményfile-ra is érvényes, mivel ott nem adtunk semmilyen paramétert.

COPY *.LST LISTA.12

Ezzel a paranccsal az aktuális könyvtárban levő összes .LST kiterjesztésű file-t egyesíthetjük egyetlen, LISTA.12 azonosítójú file-ba.

COPY *.LST + *.CRF LISTA.12A

Az előző példa bővített változatával az eredményfile-ba minden .CRF kiterjesztésű file is átmásolódik.

COPY *.LST + *.CRF *.PRN

Ebben az esetben nem egy, hanem több új file-t hozunk létre, az azonos file-névvel rendelkező, .LST és .CRF kiterjesztésű file-párokat egyesítjük ugyanezen file-név és .PRN kiterjesztés alatt.
A képernyőn megjelenő üzenetek:

```
A letrehozando file: SYF01A.PRN
A kovetkezo file-okbol fog allni:
        SYF01A.LST
        SYF01A.CRF
A letrehozando file: SYF02A.PRN
A kovetkezo file-okbol fog allni:
        SYF02A.LST
        SYF02A.CRF
```

A file-csoport-hivatkozásokkal (* és ? karaktereket tartalmazó file-hivatkozások) mindig óvatosan bánjunk!

COPY /B LIST.COM+,, vagy COPY LIST.COM /B+NUL /A

A file-t egyesítjük egy üres file-lal, és az eredményt ugyanabban a könyvtárban, ugyanazzal az azonosítóval helyezzük el. A parancs hatására csak a file keletkezési dátuma változik az aktuális napi dátumra.

COUNT

Feladata

A megadott szövegfile(ok) sorait, szavait, illetve karaktereit számlálja meg és jelzi ki.

Paraméterezése

COUNT [*m*:] [*út*] *file-név* [.*kit*] [...] [/L] [/W] [/C]

Használata

A parancsban több file-, illetve file-csoport-hivatkozás adható meg. Ha a "–" jelet adjuk meg mint file-hivatkozást, bemenő adatait a standard input perifériáról várja.

A /L opcióval a kijelölt szövegfile(ok) sorait számláltathatjuk meg. A parancs sornak tekint minden olyan karaktersorozatot, amelyet kocsi-vissza és soremelés (CR–LF, hexadecimális kódja: 0D 0A) karakter zár le. A sor lehet üres is, vagyis csak a kocsi-vissza és soremelés karakterekből álló.

A /W paraméter a file(ok) szavainak leszámlálását váltja ki. Szónak tekintünk minden olyan karaktersorozatot, amelyet mindkét oldalon space (betűköz), TAB (tabulátor) vagy új-sor (NL = CR–LF) karakter határol, és ezek a karakterek a szó belsejében nem fordulnak elő.

A /C opció megadásával a file(ok) karaktereinek számát kaphatjuk meg. Minthogy minden karakter egy byte-ot foglal el, így a file(ok) méretét kapjuk meg byte-okban.

A /L, /W és /C paraméterek tetszés szerinti kombinációban is megadhatók. Hatásuk minden kijelölt file-ra kiterjed. A COUNT parancs a standard output periférián közli a számlálások eredményét.

Példák

COUNT E:*.BAS /L

Az E meghajtó gyökérkönyvtárában levő BASIC programok sorainak számáról kérünk összefoglaló táblázatot.

COUNT A:NEVSOR /CLW

Az A meghajtó aktuális könyvtárában található NEVSOR file karaktereit, szavait és sorait is megszámoltatjuk.

Filespec	Sor	Szo	Karakter
A:NEVSOR	8	16	117

CTTY

Feladata

A standard input/output perifériát (konzolt) jelöli ki.

Paraméterezése

CTTY *perifériánév*

Használata

Standard input/output perifériaként a következők jelölhetők ki:

CON	a központi gép billentyűzete és képernyője
AUX COM1 COM2	az aszinkron kommunikációs adapterek egyikére kapcsolt billentyűzet és képernyő

A fenti szabványos periférianeveken kívül olyan karakteres üzemmódú berendezések is kijelölhetők, amelyekhez a PROPOS–16-ba beépített saját fejlesztésű perifériakezelő programot használunk. A kijelölt berendezésnek be- és kiviteli funkciókat is kell tudnia végrehajtani.

A standard input/output átirányításának lehetőségét csak akkor használjuk, ha a programjaink az operációs rendszer rutinjai segítségével végzik input/output műveleteiket.

Példák

CTTY AUX

A standard input/output műveleteket ettől kezdve az AUX berendezésre irányítja a PROPOS–16.

CTTY CON

Visszaállítottuk az eredeti állapotot, amikor a központi képernyőt és billentyűzetet használhatjuk.

Feladata

Törli a megadott file-t vagy file-csoportot.

Paraméterezése

DEL [*m*:] [*út*] *file-név* [. *kit*] [/P]

Használata

Az adott meghajtó kijelölt (al)könyvtárából törölhetjük a parancsban szereplő file(oka)-t. Használhatjuk az alapértelmezésként feltételezett meghajtót és aktuális könyvtárát is.

A törölt file-ok véglegesen elvesznek!

Csak olvasható, rejtett és rendszerfile-ok nem törölhetők. File-csoport törlésekor listát kapunk a törölt file-ok azonosítóiról. Ha az összes file-t (*.*) akarjuk törölni, a program visszakérdez:

Biztos (I/N)? _

és csak az I (igen) betű és az [ENTER] lenyomásával kezdhetjük a törlést. Az N (nem) válasszal a parancs végrehajtását megakadályozhatjuk.

A DEL paranccsal alkönyvtár nem szüntethető meg (l. RMDIR parancs).

A /P opció megadásával kérhetjük, hogy a törlésre kerülő file-ok megszüntetését egyenként engedélyezhessük. Használata file-csoport törlésekor (ha a megadott file-névben vagy kiterjesztésben ?, illetve * szerepel) ajánlott:

```
C:\DEL  \FILE*.*   /P
FILE0000 REC      Torolheto  (I/N)?  I
FILE0001 REC      Torolheto  (I/N)?  I
FILE0002 REC      Torolheto  (I/N)?  I
FILE0003 REC      Torolheto  (I/N)?  I
FILE0004 REC      Torolheto  (I/N)?  I
FILE0005 REC      Torolheto  (I/N)?  I
FILE0006 REC      Torolheto  (I/N)?  I
         7 file torolve
```

DEL

Példák

DEL B:\SZAMLA.COM

A kijelölt file-t a B meghajtó gyökérkönyvtárából töröljük.

DEL *.BAK

Minden .BAK kiterjesztésű file-t törlünk a feltételezett meghajtó aktuális könyvtárából.

DEL MACRO*.* /P

Az aktuális könyvtár alá rendelt MACRO alkönyvtárból az összes file-t töröljük, de a törlés végrehajtását file-onként kívánjuk engedélyezni, hogy az esetleg mégis szükséges file-okat megmenthessük.
A parancs végrehajtása után (ha minden file törlését engedélyeztük) a MACRO alkönyvtárban csak a "." és ".." bejegyzések maradnak, így az alkönyvtár az RMDIR paranccsal megszüntethető.

Feladata

Két szövegfile összehasonlítása után kiírja azokat a sortörlő és -beszúró parancsokat, amelyekkel az első file-ból a második előállítható.

Paraméterezése

DIFF [*m*:] [*út*]*file-név* [.*kit*] [*m* :] [*út*] *file-név* [.*kit*] [/**E**] [/**B**]

Használata

A DIFF parancs paramétereként két szövegfile-hivatkozás megadása kötelező. File-csoport-azonosítók nem használhatók. A "–" jellel a standard inputra hivatkozhatunk.

A parancs a két file-t soronként hasonlítja össze úgy, hogy a sorok sorrendjét is figyelembe veszi. Eltérések esetén a standard outputon jelzi, hogy az elsőnek kijelölt file-ból mely sorokat kell törölni, illetve melyek elé és milyen új sort kell beszúrni ahhoz, hogy abból a második file-t állíthassuk elő.

A DIFF parancsot nagyméretű, de egymástól csak alig különböző szövegfile-ok összehasonlítására célszerű használni, mert így a kapott módosítási lista még kezelhető terjedelmű lesz. Lemezterületet is megtakaríthatunk, ha nem mindkét nagy file-t tároljuk, hanem csak az egyiket, valamint a másik előállításához szükséges módosító parancsokat őrizzük meg.

A standard output periférián megjelenő módosítási lista **I** betűvel a beszúrandó, **D** betűvel a törlendő sorokat jelzi. A betűt a megfelelő sorszám követi (l. az EDIT program leírását az 5. fejezetben).

A /**E** paraméter megadásával az EDIT sorszerkesztő program inputjának megfelelő formátumban áll elő a módosítási lista.

A /**B** opció megadása esetén a parancs végrehajtása során a program nem tekinti különbözőnek a space és a TAB karaktereket.

DIFF

Példák

DIFF NEVSOR1 NEVSOR2

Az összehasonlított két file közötti eltérésekről a standard output periférián kapunk listát. Mindkét file a feltételezett meghajtó aktuális könyvtárában található.

```
C:\DIFF NEVSOR1 NEVSOR2
Opcio (k) :

3d
6i
Kodaly Zoltan

PARANCS VEGREHAJTVA !      USSON LE EGY BILLENTYUT !
```

Tehát a 3. sor törlendő, és a 6. sor elé egy új sort kell beszúrni ahhoz, hogy a NEVSOR1 file-ból a NEVSOR2 file-t előállítsuk.

DIFF NEVSOR1 NEVSOR2 /E>MODFILE

Az EDIT programnak megfelelő formátumú módosító parancsokat a MODFILE nevű file-ba irányítjuk át. Ezt az

EDIT NEVSOR1<MODFILE

parancsban felhasználva a NEVSOR1 file-t ugyanolyanra módosíthatjuk, amilyen a NEVSOR2 file.

DIFF A:BERLISTA.JAN E:\BER\BERLISTA.FEB /B

A két, különböző meghajtón levő file-t úgy hasonlítjuk össze, hogy a space (betűköz) és a TAB (tabulátor) karaktereket nem tekintjük különbözőnek.

Feladata

Egy (al)könyvtár katalógusának teljes tartalmát vagy egy file-csoport elemeit listázza ki. Közli a lemez kötetazonosítóját, és a lemezen levő szabad terület méretét is.

Paraméterezése

DIR [*m*:] [*út*] [*file-név* [. *kit*]] [/P] [/W]

Használata

Az egyik leggyakrabban használt PROPOS–16 parancs, amely a paraméterként megadott (vagy a feltételezett) meghajtó kijelölt (vagy aktuális) könyvtárából kiírja az ott tárolt file-ok nevét, kiterjesztését, méretét byte-okban, valamint létrehozásának, illetve legutolsó módosításának dátumát és időpontját. Kiíródnak az adott könyvtár alá tartozó alkönyvtárak azonosítói is, ⟨DIR⟩ megjelöléssel. Ha a parancssorban file- vagy file-csoport-azonosítót is megadunk, csak a megfelelő file-okról kapunk listát.

A felsorolásban a „rejtett" és a „rendszer" megjelölésű file-ok nem szerepelnek. A parancs tájékoztat a felsorolt file-ok összes helyfoglalásáról és az adott lemezen még rendelkezésre álló szabad terület méretéről is:

```
"C" meghajtoban levo lemez  : cimke nelkuli
Konyvtar  C:\propos.16

.                  <DIR>        84- 1-01     0:13
..                 <DIR>        84- 1-01     0:13
COMMAND    COM      18321       85- 3-02     0:28
HELP       IND       1536       84- 1-01     0:05
INXGEN     COM       5504       85- 3-15     9:20
PRINT      COM       4608       85- 3-15     9:21
MORE       COM        384       85- 3-15     9:21
DIFF       EXE      19840       84- 1-01     2:17
MENU       COM      21026       85- 3-15     0:21
MENUSYS    BAT         63       85- 3-01     0:04
CONFIG     SYS         39       85- 2-01    10:14
ANSI       SYS       1664       85- 3-01    12:00
EDIT       COM       4608       85- 3-15     9:21
SORT       EXE      21120       84- 1-01     0:13
COUNT      EXE      15744       84- 1-01     2:11
FIND       EXE      15616       84- 1-01     3:24
COMM       EXE      15488       84- 1-01     3:24
  :
NEVSOR                117       84- 1-01     0:32
RESTORE    EXE      19072       85- 3-15    10:01
        40 file, osszesen:    432160 byte.   5255168 byte szabad
```

A /P paraméter használata hosszabb könyvtárlisták kiírásánál szükséges. Hatására a lista laponként jelenik meg, és a képernyő megtelte után csak egy tetszőleges billentyű lenyomása után jelenik meg a következő lapnyi lista.

A /W paraméterrel tömörített formátumú listát kérhetünk. Ezen csak a file-nevek és kiterjesztések szerepelnek, minden sorban öt file-é.

```
"A" meghajtoban levo lemez : cimke nelkuli
Konyvtar A:\

COMMAND COM HELP     IND MEDIACHG COM FSYS   COM PRINT COM
FORMAT  EXE MENU     COM BACKUP   EXE CONFIG SYS ANSI  SYS
HD      EXE CMP      EXE CONVEXE  EXE IMP    EXE CHMOD EXE
FDISK   COM RESTORE  EXE DISKCOPY EXE CONFIG EXE MODE  COM
MENUSYS BAT EDIT     COM VDISK    SYS STAT   COM MORE  COM
ASSIGN  COM RECOVER  COM INXGEN   COM HELP   HLP DEBUG COM
AUTOEXEC BAT
        31 file, osszesen:     0 byte.     7168 byte szabad
```

Alkönyvtárak listájának elején két különleges bejegyzés is szerepel:

. ⟨DIR⟩
.. ⟨DIR⟩

Ezen bejegyzések tartalmát az operációs rendszer használja az adott alkönyvtár azonosítása (.), valamint a fölérendelt könyvtár meghatározása (..) során.

Példák

DIR vagy DIR .

A feltételezett meghajtó aktuális könyvtárának teljes listáját kérjük.

DIR A:

Az A meghajtó aktuális könyvtárának teljes listáját kérjük.

DIR E:\SZAMLA vagy DIR E:SZAMLA

Az E meghajtó SZAMLA alkönyvtárának teljes listáját kérjük. (Aktuális könyvtár a gyökérkönyvtár.) A SZAMLA lehet egy file-név is, ebben az esetben ennek a file-nak a jellemzői íródnak ki.

DIR \ /W

Az aktuális meghajtó gyökérkönyvtárának teljes listáját tömörített formában kérjük.

DIR *.BAS vagy DIR .BAS

A feltételezett meghajtó aktuális könyvtárában található BASIC programok felsorolását kérjük.

DIR FILE.* vagy DIR FILE

A feltételezett meghajtó aktuális könyvtárából a FILE nevű, tetszőleges kiterjesztésű file-ok és alkönyvtárak listáját kérjük.

DIR *.

A kiterjesztés nélküli (3 space kiterjesztésű), tetszőleges nevű file-ok és alkönyvtárak listáját kérjük.

DIR FILE*.CHK

A STAT program által a gyökérkönyvtárban létrehozott file-ok listáját kérjük (l. 4–87. oldal)

DIR ..

A feltételezett meghajtó aktuális könyvtára fölötti könyvtár tartalmát listázzuk ki. (A gyökérkönyvtárban nem adható ki.)

DISKCOPY

Feladata

Hajlékony lemez(ek) teljes tartalmának átmásolását és a másolatnak az eredetivel való összehasonlítását teszi lehetővé. Szükség esetén az új lemez megformálását is elvégzi.

Paraméterezése

DISKCOPY [*m*:] [*m*:] [/C [O]]

Használata

A parancs első paramétere a forrás (input)-, második paramétere a cél (output)-meghajtót jelzi. Ha csak egy meghajtót jelölünk ki, azt forrásként értelmezi, és a másolatot a feltételezett meghajtóra készíti. Ha nem adunk meg meghajtó azonosítót. a feltételezett meghajtón történik a másolás, a forrás- és céllemez cserélgetésével. Ugyanez történik akkor is, ha egyazon meghajtót jelöljük ki mindkét paraméterben.

A parancsot végrehajtó program mindig felszólít a megfelelő lemez behelyezésére.

```
A forras meghajto azonositoja: a
A cel meghajto azonositoja: b
Opcio(k):

Helyezze a masolando lemezt a(z) A meghajtoba
rogzitse es usson le egy billentyut!

Helyezze a masodpeldanyt a(z) B meghajtoba
rogzitse es usson le egy billentyut!

A masolas 2 oldalas, 40 savos, 9 szektoros lemezek kozott tortenik
```

A másolat számára felhasznált lemez eredeti tartalma elvész!

A másolat mindig a forráslemezzel azonos felépítésű lesz. Egyoldalas lemez másolata egyoldalas lesz, kétoldalasé kétoldalas, és a sávok száma sem változik.

A /C opció lehetővé teszi, hogy a másolatot ellenőrizzük is. A program visszaolvassa

és az eredetivel összehasonlítja a lemezre írt file-okat. Ez a parancs végrehajtási idejét természetesen megduplázza, de cserébe biztosít minket a hibátlan másolat létrejöttéről.

/CO paraméterrel csak összehasonlítást kérünk, másolás nélkül.

A parancs akkor is hibátlanul végrehajtódik, ha az outputként kijelölt lemez megformálatlan, „gyári új". A formálást a másolás során elvégzi a DISKCOPY, és jelzi a hibás szektorokat is.

Ha a forráslemezen a PROPOS–16 operációs rendszer programfile-jai is szerepeltek (PROBIO.COM, PRODOS.COM és COMMAND.COM), természetesen ezek is átmásolásra kerülnek – jóllehet „rejtett" file-ok (a COPY paranccsal ezek nem másolhatók!). A másolat azonosító cimkéje is megegyezik az eredeti lemezével.

Olyan lemezeken, amelyeken sok file-kreálás, -törlés és -javítás történt, a file-ok egy része már nem folyamatosan helyezkedik el. Ezeknek a széttöredezett file-oknak a feldolgozása is lassabb, mivel az iró/olvasó fej többszöri mozgatását igényli. A DISKCOPY paranccsal átmásolva az új kópia is ugyanilyen lesz. Ezért ezeket a lemezeket nem így célszerű másolni, hanem a FORMAT paranccsal megformált új lemezre, a COPY program segítségével.

A DISKCOPY parancs *nem használható:*

- a fix lemez másolására,
- különböző típusú meghajtók közötti átvitelre,
- a virtuális (RAM) lemezre, illetve lemezről másolásra,
- az ASSIGN paranccsal átirányított meghajtók esetén.

Példák

DISKCOPY

A másolandó és az üres lemezt felváltva kell a feltételezett meghajtóba helyezni (egymeghajtós másolás).

DISKCOPY B: A: /C

A B meghajtóban levő lemez tartalmát átmásoljuk az A meghajtóban levő lemezre, majd összehasonlítjuk a két lemez tartalmát.

FIND

Feladata

A parancssorban megadott szöveg (karaktersorozat) előfordulásait keresi meg a felsorolt szövegfile-okban.

Paraméterezése

FIND [*m*:][*út*]*file-név* [.*kit*] [...] [/V] [/C] [/L] [/N] [/H] [/Y] [/E]

Használata

A FIND parancsnak több szövegfile-, illetve file-csoport-hivatkozással adhatók meg azok a file-ok, amelyekben a keresést végzi.

A keresett szövegrészt (stringet) a program kérdésére válaszolva, a billentyűzetről kell megadni, ENTER-rel lezárva (l. példák).

A parancs végrehajtása során a standard output perifférián megjelennek a bemenő file(ok) azon sorai, amelyekben a megadott szöveg előfordul.

A /V paraméter megadásával a parancs működése megfordítható, és a program a file-ból csak azokat a sorokat írja ki, amelyekben az adott szöveg *nem* fordul elő.

A /C opció lehetővé teszi, hogy csak a kiválasztott sorok számát írassuk ki (a /V egyidejű megadásától függően a szöveget tartalmazó, illetve nem tartalmazó sorok a kiválasztottak).

A /L paraméter megadásával csak azon file-ok azonosítójáról kapunk listát – a parancssorban kijelöltek közül –, amelyekben a megadott szövegrész előfordul.

A /N megadásával kérhetjük, hogy a parancs végrehajtása során a kiválasztott sorok sorszámát is írja ki a program. Ez megkönnyíti ezen sorok további kezelését, feldolgozását, javítását.

A /H paraméterrel a kiválasztott file-ok nevének kiíratását letilthatjuk. A /L opcióval ellentétes hatása miatt a két paraméter egy parancsban nem adható meg.

A /Y paramétert is meg kell adnunk, ha a keresés során nem kívánjuk megkülönböztetni az adott szövegrész kis- és nagybetűvel írt változatát.

Csak a /E opció egyidejű megadásával kereshetünk a file-okban "/ " karakterrel kezdődő szövegrészeket.

Példák

FIND PROGRAM.LST /Y
String: error

A feltételezett meghajtó aktuális könyvtárában levő listafile-ból az „error" szót tartalmazó sorokat íratjuk ki, függetlenül attól, hogy a szó kis- vagy nagybetűkkel írva fordul elő az adott sorban.

FIND B:CIMFILE /C
String: XVI.

A megadott nevű és a B meghajtón található file azon rekordjainak (sorainak) számát kérdezzük, amelyekben a „XVI." karaktersorozat előfordul – vagyis a XVI. kerületi címek számát kérjük.

FIND RAJZ*.BAS /N
String: G1

RAJZ1, RAJZ2 stb. nevű BASIC programjainkból azokat a sorokat íratjuk ki, amelyekben a G1 változót használtuk. Az illető sorok előtt a file-on belüli sorszámukat is kérjük kiírni.

FIND RAJZ*.BAS /LV
String: G1

A kijelölt file-csoportból csak azoknak a file-oknak az azonosítóját kérdezzük, amelyekben a G1 szövegrész nem fordul elő.

FORMAT

Feladata

Hajlékony lemez(ek) megformálását (inicializálását) végzi, a PROPOS–16 operációs rendszer tárolási módjának megfelelően.

Megvizsgálja a lemez teljes felületének használhatóságát, létrehozza a gyökérkönyvtárat és a file-allokációs táblát a lemez elején, és felírja a rendszert betöltő („boot") programot.

Opcionálisan a PROPOS–16 rendszerfile-okat is a lemezre másolja.

Paraméterezése

FORMAT [*m*:] [/S] [/1] [/8] [/L]

Használata

Minden mágneslemezt meg kell formálni, mielőtt először írnánk rá. A FORMAT parancsot olyan lemezre is kiadhatjuk, amelyet előzőleg már használtunk, ilyenkor azonban a lemez teljes korábbi tartalma elvész. Az újraformálást akkor célszerű elvégezni, ha egy régebbi lemez tartalmára már nincs szükségünk, és az új használatbavétel előtt szeretnénk meggyőződni a lemez felületének használhatóságáról. A FORMAT parancs ugyanis a lemez minden szektorába ír, majd visszaolvassa és ellenőrzi a szektorok tartalmát. A hibásnak talált szektorokat megjelöli a lemezen, és ezeket később minden program „átlépi". Ily módon csökken ugyan a lemez hasznos területe, de számtalan kellemetlenségtől menekülünk meg.

A parancsot paraméterek nélkül megadva a feltételezett meghajtóban levő lemez formálását kísérli meg a rendszer, a meghajtó típusától függetlenül 40 sávos, 2 oldalas alakban, lemezazonosító címke és a rendszerprogramok felírása nélkül.

Figyelem! A fix lemez megformálását akár feltételezett meghajtóként, akár explicit módon megadva kérjük, hibajelzést kapunk! A parancs csak a hajlékony lemezekhez használható.

A program lefutását az alábbi operátori párbeszéd kíséri:

```
A:\FORMAT
Meghajtoazonosito: A
Opcio(k):

A formalas 2 oldalas, 40 savos, 9 szektoros lemezen tortenik

Helyezze a formalando lemezt a(z) A meghajtoba,
rogzitse es usson le egy billentyut!

FORMALAS...
```

Futása közben a FORMAT program a képernyőre írja az éppen vizsgált sáv sorszámát, így követhetjük működését.

A FORMAT parancs /S paraméterével kérhetjük, hogy az új lemezre a PROPOS–16 operációs rendszer fő moduljai automatikusan felíródjanak. Ezek:

PROBIO.COM
PRODOS.COM
COMMAND.COM

Tekintettel arra, hogy az első két file-nak kötelezően a lemez elején kell lennie, és ezek ún. „rejtett" file-ok (a könyvtárlistán nem szerepelnek), más eszközökkel utólag már nem másolhatjuk őket új lemezünkre.

A /1 paraméter megadásával bármilyen meghajtón egyoldalas lemez megformálására adhatunk utasítást. A lemezen 40 információs sávot hoz létre a program. Ne feledjük, hogy egyoldalas meghajtó csak egyoldalas lemezt tud olvasni, míg a kétoldalas meghajtó az egy- és kétoldalas lemezeket egyaránt képes kezelni!

A /8 paraméter használatával 80 sávos meghajtón 80 sávos lemezeket is formálhatunk. A 80 sávos meghajtó minden korábbi típusú floppy lemez kezelését lehetővé teszi (35 és 40 sávos, egy-, illetve kétoldalas). A FORMAT parancs alapértelmezés szerint ezen is 40 sávos lemezt formál.

A /L paraméter megadása esetén a FORMAT program megkérdezi és a lemezre ír egy azonosító címkét:

```
Adja meg a lemez cimkejet (max. 11 karakter): PROPOS-16
```

(l. DIR parancs) a lista fejlécében láthatjuk viszont. A lemezazonosító használata nem kötelező, de ajánlott, mivel megkönnyíti a lemezeink nyilvántartását. A lemez neve a file-nevekhez hasonlóan 1–11 karakterből állhat, ám a 8. és 9. karakter között nem tartalmaz elválasztó pontot. A lemezazonosító címkét a VOL parancscsal írathatjuk ki.

Ha a FORMAT paranccsal rendszerlemezt hozunk létre, ne feledkezzünk meg a PROPOS–16 működéséhez szükséges többi file

HELP.HLP	menü/help szövegfile
HELP.IND	menü/help indexfile
ANSI.SYS	kiterjesztett képernyőkezelés
CONFIG.SYS	konfigurációs file
AUTOEXEC.BAT	inicializáló batch-file

valamint a munkánkhoz szükséges parancsokat végrehajtó programok, perifériakezelő programok (pl. a virtuális lemezt kezelő VDISK.SYS) stb. átmásolásáról sem.

Példák

FORMAT

A feltételezett meghajtóban levő hajlékony lemezt kétoldalas, 40 sávos alakra formáljuk meg. A lemezen nem lesz rajta a PROPOS–16 rendszer, és azonosító címkét sem kap. A program lefutása után a képernyőn látható üzenetek:

```
A:\FORMAT
Meghajtoazonosito:
Opciok(k):

A formalas  2 oldalas, 40 savos, 9 szektoros  lemezen tortenik

Helyezze a formalando lemezt a(z) A meghajtoba,
rogzitse es usson le egy billentyut!

FORMALAS...
```

FORMAT

FORMAT B: /S /1

A B meghajtóban egyoldalas, 40 sávos floppyt formálunk, és az operációs rendszert is átmásoljuk rá. (Egyoldalas, 80 sávos lemez létrehozása értelmetlen.)

```
A:\FORMAT B: /S /1

A formalas  1 oldalas, 40 savos, 9 szektoros  lemezen tortenik

Helyezze a formalando lemezt a(z) B meghajtoba,
rogzitse es usson le egy billentyut!

FORMALAS...

Sav: 39

FORMALAS VEGE

A lemez osszterulete (byte):  184320
A lemez szabad terulete (byte):  139264
A rendszer altal lefoglalt terulet (byte):  40448

PARANCS VEGREHAJTVA!     USSON LE EGY BILLENTYUT!
```

FORMAT A: /8 S L

Az A meghajtóban 80 sávos, kétoldalas, lemezazonosító címkével ellátott lemezt formálunk, amely a PROPOS–16 rendszert is tartalmazza.

```
C:\FORMAT A: /8SL

A formalas  2 oldalas, 80 savos, 9 szektoros  lemezen tortenik

Helyezze a formalando lemezt a(z) A meghajtoba,
rogzitse es usson le egy billentyut!

FORMALAS...

Sav: 79

FORMALAS VEGE

Adja meg a lemez cimkejet (max. 11 karakter): PROPOS–16

A lemez osszterulete (byte): 737280
A lemez szabad terulete (byte): 689152
A rendszer altal lefoglalt terulet (byte): 40960

PARANCS VEGREHAJTVA!    USSON LE EGY BILLENTYUT!
```

FSYS

Feladata

Végigköveti a lemezen található könyvtárak kapcsolatrendszerét (fastruktúráját), és kiírja a standard output perifériára.

Paraméterezése

FSYS [*m*:] [/F]

Használata

A parancssorban megadott (illetve a feltételezett) meghajtón levő lemez azonosítóját és a lemezen tárolt könyvtárak kapcsolatrendszerét írathatjuk ki. Ez a művelet sok alkönyvtárat tartalmazó fix lemez esetén néhány percet is igénybe vehet. A listán az alkönyvtárakhoz vezető keresési út teljes leírása szerepel.

A parancs végrehajtása független attól, hogy melyik az aktuális könyvtár.

A **/F** paraméter megadásával kérhetjük, hogy az alkönyvtárakban tárolt file-okról is készítsen listát a program. (A gyökérkönyvtárról nem készít tartalomjegyzéket!)

Példa

FSYS A: /F

Az A meghajtóban levő lemez könyvtárainak kapcsolatáról, valamint az alkönyvtárak tartalmáról kapunk listát.

```
Konyvtari eleresek listaja.
Lemezazonosito: FORDITOK

Konyvtar: A:\C

File-ok:
            LC1.EXE
            LC2.EXE

Alkonyvtarak: nincsenek

Konyvtar: A:\BASIC

File-ok:
            BASICS.COM

Alkonyvtarak: nincsenek

Konyvtar: A:\PASCAL

File-ok:
            PAS1.EXE
            PASKEY
            PAS2.EXE
```

Feladata

A kijelölt file tartalmát hexadecimális és/vagy karakteres formában jeleníti meg a standard output perifférián.

Paraméterezése

HD [*m*:] [*út*] *file-név* [. *kit*] [[±] *offset* [.] [/**B**]] [/**H**] [/**C**]

Használata

A HD parancs segítségével tetszőleges típusú file tartalmát megjeleníthetjük. Ha a parancsban file-név helyett " - " jelet adunk meg, bemenetét a standard input perifériáról várja. File-csoport-azonosító megadása nem megengedett.

Alapértelmezés szerint (opcionális paraméterek nélkül) a file kiíratása hexadecimális formában történik, a file elejétől a végéig. A byte pozició (offset) a sor elején, szintén hexadecimális kódban látható.

```
0000000  63 6F 70 79 20 68 65 6C 70 2E 68 6C 70 20 68 65
0000010  6C 70 2E 73 61 76 0D 0A 65 64 69 74 20 68 65 6C
0000020  70 2E 68 6C 70 0D 0A 69 6E 78 67 65 6E 20 68 65
0000030  6C 70 2E 68 6C 70 0D 0A 72 65 73 74 61 72 74 0D
0000040  0A 1A
```

HD

A /C opció megadása esetén a kiírás karakteresen történik. A nem nyomtatható karakterek helyén betűköz jelenik meg, kivéve néhány speciális kódot (00 = \0, backspace = \b, LF = \n, CR = \r, tab = \t):

```
0000000    c  o  p  y     h  e  l  p  .  h  l  p     h  e
0000010    l  p  .  s  a  v \r \n e  d  l  t     h  e  l
0000020    p  .  h  l  p \r \n i  n  x  g  e  n     h  e
0000030    l  p  .  h  l  p \r \n r  e  s  t  a  r  t \r
0000040   \n
```

A /H opciót önmagában használva az alapértelmezéssel megegyező, hexadecimális kiírási formátumot állíthatunk elő. A H és C paraméter együttes megadása a két formátum kombinációját eredményezi:

```
0000000    c o p y   h e l p . h l p   h e
0000000   63 6F 70 79 20 68 65 6C 70 2E 68 6C 70 20 68 65

0000016    l p . s a v \r \n e d i t   h e l
0000010   6C 70 2E 73 61 76 0D 0A 65 64 69 74 20 68 65 6C

0000032    p . h l p \r \n i n x g e n   h e
0000020   70 2E 68 6C 70 0D 0A 69 6E 78 67 65 6E 20 68 65

0000048    l p . h l p \r \n r e s t a r t \r
0000030   6C 70 2E 68 6C 70 0D 0A 72 65 73 74 61 72 74 0D

0000064    \n
0000040   0A 1A
```

Az előjeles *offset* paraméter segítségével lehetőségünk van a file részeinek kiíratására is. A "+" előjellel megadott offset a file elejéhez képest, a "–" előjellel megadott a végéhez képest adja meg a kiíratás kezdőcímét. Az offset alapértelmezés szerint egy hexadecimális szám, de ha utána ". " áll, decimálisnak tekinti a program.

A file elejétől, illetve végétől való távolság egysége alapértelmezés szerint byte, a /B opció megadása esetén blokk (512 byte).

Példák

HD TESZT.COM /HC

Futtatható formátumú programunkat íratjuk ki a feltételezett meghajtó aktuális könyvtárából. Hexadecimális és karakteres listát is kérünk – így a programban levő szövegek könnyen felismerhetők.

HD A:TESZT.DAT -10. /B

Az A meghajtóról adatfile-unk utolsó (decimális) 10 blokkját íratjuk ki, hexadecimális alakban.

HD A:TESZT.DAT +200

Fenti file-unkat az elejétől 512 byte (hexadecimális 200) távolságra kezdjük kiíratni.

IMP

Feladata

Lehetővé teszi, hogy a PROPOS–8 és a PROPOS–16 operációs rendszerek között file-okat cserélhessünk.

Paraméterezése

IMP [*m*:] [*út*] *file-név* [. *kit*] [*m*:] [*út*] [*file-név* [. *kit*]] [/C] [/M]

vagy

IMP [*m*:] [*út*] [*file-név* [. *kit*]] **DIR** [/C] [/M]

Használata

Az IMP parancs segítségével a PROPOS–8 operációs rendszer alatt felírt hajlékony lemezes file-okat a PROPOS–16 alatt is olvashatjuk, illetve PROPOS–16 file-okba átmásolhatjuk. A konverzió ellenkező irányban is elvégezhető, a parancs paraméterezésétől függően.

A /C paraméter PROPOS–8, a /M paraméter PROPOS–16 formátumú bemenő file-okra utal. A két paraméter közül egynek a megadása kötelező.

A parancsban megadott első file-hivatkozás a bemenő file-t vagy file-csoportot azonosítja. Ha ezek PROPOS–8 formátumúak, nem lehetnek a feltételezett meghajtón, tehát ekkor kötelező specifikálni a meghajtót is. PROPOS–8 lemezen út (alkönyvtár) nem lehet.

Az IMP parancs első alakjában megadható második file-hivatkozás az átalakított file azonosítója. Ha file-csoportot konvertálunk, kimenő file-név nem adható meg. Kimenő file-név híján az átalakított file-ok az eredetivel megegyező azonosítót kapnak.

Az IMP parancs második alakjának megadásával a kijelölt meghajtó, könyvtár, illetve file-csoport listáját írathatjuk ki, a DIR parancshoz hasonló formátumban.

Az IMP parancs használhatóságának előfeltétele, hogy a konfiguráció tartalmazzon olyan meghajtót, amely a PROPOS –8 lemezek olvasását fizikailag lehetővé teszi. Az alábbi két PROPOS–8 lemezformátum olvasása és írása megengedett:

Lemezátmérő (inch)	5,25	8
Sávok száma	40	77
Szektor/sáv	8	26
Byte/szektor	512	256
Logikai szektor (byte)	128	128
Szektoreltolás	–	7
File-ok száma (max.)	64	128
Logikai szektor/blokk	8	16
Kapacitás (Kbyte)	156	486
Katalógusszektorok	2	2
Foglalt sávok	1	2

Példák

IMP C: *.* /C

A C meghajtón levő PROPOS–8 formátumú lemez minden file-ját PROPOS–16 formátumúra alakítjuk át, és változatlan azonosítóval a feltételezett meghajtó aktuális könyvtárába írjuk.

IMP C: DIR /C

A C meghajtón levő PROPOS–8 formátumú lemez katalógusának tartalmát íratjuk ki a képernyőre.

IMP \BER \ALAP.FEB C:ALAPBER /M

A feltételezett meghajtó BER alkönyvtárában tárolt, ALAP.FEB azonosítójú, PROPOS–16 formátumú file-t a C meghajtóra írjuk, ALAPBER file-névvel és PROPOS–8 formátumban.

MENUSYS

Feladata

A menü/help információs file módosítását lehetővé tevő batch parancsfile.

Paraméterezése

MENUSYS

Használata

A parancs a MENUSYS.BAT batch file-t aktivizálja, paramétereket nem kell a parancssorban megadni. A batch file tartalma:

```
copy help.hlp help.sav
edit help.hlp
inxgen help.hlp
restart
```

Az első parancs (**COPY**) biztonsági célból másolatot készít a HELP.HLP szövegfileról.

A második sor az **EDIT** szövegszerkesztő programot hívja, amellyel a menü/help információs file tartalmát – az alábbiakban ismertetésre kerülő szabályok betartásával – módosíthatjuk (l. az 5. fejezetet).

A batch-file harmadik sora az **INXGEN** program segítségével elemzi és kiértékeli a javított szövegfile-t, és új indexfile-t (HELP.IND) állít elő.

Végül újraindul a menürendszer, az új file-ok felhasználásával (**RESTART** program).

A menü/help file felépítése

A szövegfile utasításokat és menü/help szövegeket tartalmazhat.

Az utasítások alakja:

/// név P (parancs definiálása)
/// név H (helpszöveg definiálása)
/// név M (menüoldal definiálása)

A menüt és a helpszöveget definiáló utasítás után következő sorok alkotják a megjelenítendő menüt, illetve helpszöveget, egészen a legközelebbi ///... sorig.

A menü definiciókban # ... ¦ jelek között kell megadni azokat a karaktersorozatokat, amelyekkel parancs és help bejegyzéseket is definiálunk. Az így „zárójelezett" karaktersorozat a képernyőn negatív karakterekkel jelenik meg.

A menü/help file végét egy @ karakter jelzi.

Példa

Az alábbi lista a szabványos menü/help file elejét és egy másik részletét mutatja be a fentiek illusztrálására.

```
///VALASZTHATO PARANCSCSOPORTOKM
                    VALASZTHATO PARANCSCSOPORTOK

          #KONYVTARKEZELES¦          #FILE-MUVELETEK¦

          #LEMEZKEZELES¦             #SZOVEGFILE-KEZELES¦

          #RENDSZERPARAMETEREK¦      #NYELVI RENDSZEREK¦

///RENDSZERPARAMTEREKM
                    RENDSZERPARAMETEREK

                    #CONFIG¦ Rendszerkonfiguracio
                         ismertetese, modositasa

                    #UZEMELTETESI KORNYEZET MEGADASA¦
///UZEMELTETESI KORNYEZET MEGADASAM
                    UZEMELTETESI KORNYEZET MEGADASA

          Parameterek    -#SET¦      Uzemmod        -#MODE¦

          Hozzarendeles  -#ASSIGN¦   Konzol         -#CTTY¦

                    Menumodositas -#MENUSYS¦
///RENDSZERPARAMTEREKH
                    RENDSZERPARAMETEREK MODOSITASA
```

A CONFIG programmal kiirathatjuk, illetve modosithatjuk a rendszerinditaskor eletbe lepo rendszerkonfiguraciot (CONFIG.SYS file).

Az UZEMELTETESI KORNYEZET MEGADASA menuben talalhato parancsokkal a rendszer nehany jellemzojet valtoztathatjuk meg (ASSIGN, CTTY, MENUSYS, MODE, SET parancsok).

```
///CONFIGP
///CONFIGH
                    CONFIG
```

Rendszerinditaskor a PROPOS betolti a CONFIG.SYS konfiguracios file-t, es vegrehajtja a benne levo parancsokat. A CONFIG.SYS file tartalma a CONFIG programmal kiirathato es modosithato. A modositasok a kovetkezo rendszerinditaskor lepnek eletbe.

MENUSYS

Konfiguracios parancsok:

:
:

///UZEMELTETESI KORNYEZET MEGADASAH

A menuben levo parancsokkal megvaltoztathatjuk a rendszer uzemeltetesi kornyezetet.

SET	Datum, idopont, rendszerparameterek modositasa
MODE	Nyomtato, kepernyo es aszinkron csatolo uzemmodjanak beallitasa
ASSIGN	Lemezmuveletek atiranyitasa masik meghajtora
CTTY	Konzol kijelolese
MENUSYS	A menu/help file modositasa

///ASSIGNP
///ASSIGNH
ASSIGN [x=y[...]]

Egy adott meghajtora kiadott kereseket egy masik meghajtora iranyitja.

Parameterek nelkul megadott ASSIGN visszaallitja a normal meghajto-hozzarendeleseket.

x: atiranyitando meghajto
y: kijelolt masik meghajto

Peldak:

ASSIGN A=C B=C

A rendszer az A es B meghajtora kiadott kereseket a C meghajtora iranyitja.

ASSIGN

Visszaallitja az eredeti hozzarendeleseket.

///CTTYP
///CTTYH
CTTY periferia nev

Ezzel a paranccsal megadhatjuk, hogy melyik karakteres berendezes legyen a standard input-output periferia.

Peldak:
CTTY AUX
:
:

Feladata

A megadott meghajtón üres alkönyvtárat hoz létre.

Paraméterezése

MKDIR [*m*:]*út*

vagy

MD [*m*:]*út*

Használata

Az alkönyvtárat a parancssorban megadott, illetve a feltételezett meghajtón hozzuk létre. Az új alkönyvtárhoz vezető keresési utat vagy a gyökérkönyvtárból, vagy az aktuális könyvtárból kiindulva adhatjuk meg. Az út utolsó elemeként értelemszerűen az új alkönyvtár nevét kell megadni.

Egy lemezen tetszőleges számú alkönyvtár létrehozható – a rendelkezésre álló terület korlátain belül. Másik korlátozó feltétel, hogy a gyökérkönyvtártól a legalacsonyabb szintű alkönyvtárhoz vezető út teljes leírása nem lehet 63 karakternél hosszabb.

Az alkönyvtárak használatának egyik célja az, hogy file-jainkat könnyebben áttekinthessük. Ezért egy-egy alkönyvtárban nem célszerű néhány tucatnál több file-t tárolni, ugyanakkor egy-két file kedvéért sem érdemes új alkönyvtárat létrehozni. A két-háromnál több, egymás alá rendelt könyvtárszint is az áttekinthetőséget rontja.

MKDIR

Példák

MKDIR TEST

Az aktuális könyvtár alá rendelve létrehozzuk a TEST nevű alkönyvtárat, a feltételezett meghajtón. Az új, üres könyvtár katalógusa:

```
"C" meghajtoban levo lemez : cimke nelkuli
Konyvtar C:\propos.16 \test

.          <DIR>      85-3-30   13:00
..         <DIR>      85-3-30   13:00
      2 file, osszesen:          0 byte.     5251072 byte szabad
```

A minden alkönyvtár katalógusának elején látható két különleges bejegyzés jelentése:

. – magára az alkönyvtárra mutató bejegyzés;

.. – a felette levő (apa) könyvtárra mutató bejegyzés.

Ezt a két tételt törölni vagy átnevezni nem lehet, lehetséges alkalmazásukra az utolsó példa utal.

MD \TEST

A TEST alkönyvtárat a feltételezett meghajtó gyökérkönyvtára alatti szinten hozzuk létre.

MD E:\SZAMLA\TEST.15

Az E meghajtó gyökérkönyvtára alatt már meglevő SZAMLA alkönyvtár alá rendeljük a TEST.15 azonosítójú új alkönyvtárat.

MD ..\ADAT

A feltételezett meghajtó aktuális könyvtárával azonos szinten, azaz ugyanazon felsőbb szintű könyvtár alá hozzuk létre az ADAT alkönyvtárat. Ilyen parancs természetesen csak alkönyvtárban adható ki.

Feladata

Beállítja a nyomtató, a színes grafikus képernyő vagy az aszinkron adatátviteli csatoló üzemmódját. A nyomtató kimenetet átirányíthatja az aszinkron adatátviteli csatolóra is.

Paraméterezése

(1) **MODE LPT***p*: [*n*] [, [*m*] [,**P**]]

(2) **MODE** *n*

vagy

MODE [*n*], *m*[,**T**]

(3) **MODE COM***n*: *b* [, *p*[, *d* [, *s* [,**P**]]]]

(4) **MODE LPT***p* := **COM***n*

Használata

A MODE parancs felhasználási lehetőségeit a négy formátumopciónak megfelelően ismertetjük.

(1) *Nyomtató*

A paranccsal beállíthatjuk a *p*-edik nyomtató nyomtatási paramétereit: *p*=1, 2 vagy 3 lehet, ennek megfelelően a kijelölt periféria az LPT1 (=PRN), LPT2 vagy LPT3.

Az *n* paraméterrel az egy sorba nyomtatható karakterek számát állíthatjuk be. Értéke 80 vagy 132 lehet, a számítógép bekapcsolásakor az alapértelmezés 80 karakter/sor. A 132 karakteres sorhossz beállítása keskenyebb formátumú karakterek kiírását eredményezi.

Az *m* értéke a nyomtatott sorok sűrűségét határozza meg. Értéke 6 vagy 8 lehet, ettől függően 1/6 vagy 1/8 inch (kb. 4,2, illetve 3,2 mm) soremeléssel dolgozik a nyomtató. Bekapcsolás után az alapértelmezés 6 sor/inch.

Ha nem a fenti *n*, illetve *m* értékek valamelyikét adjuk meg, a korábbi érték marad.

A **P** paraméter hatására a nyomtató „time-out" hibája esetén újra próbálkozik a rendszer mindaddig, amíg a hiba meg nem szűnik, vagy a [CTRL] [C] lenyomásával meg nem szakítjuk a ciklust. Ennek megvalósításához a MODE program egy része rezidenssé válik, ezáltal a szabad tárterület csökken.

A P paraméter nélkül a time-out hatására hibaüzenetet kapunk.

(2) *Display-csatoló*

A MODE parancs második változatával a képernyő üzemmódját állíthatjuk be. Az *n* paraméter a következő értékek egyikével használható:

40 : a képernyő minden sorában 40 (széles formátumú) karakter írható ki,
80 : a képernyő minden sorában 80 (keskeny formátumú) karakter írható ki,
BW40 : a színes grafikus képernyőcsatolóval fekete-fehér képernyőt akarunk vezérelni, soronként 40 karaktert kiírva,

BW80 : a színes grafikus képernyőcsatolóval fekete-fehér képernyőt akarunk vezérelni, soronként 80 karaktert kiírva,
CO40 : a színes grafikus képernyőcsatolóval színes képernyőt akarunk vezérelni, soronként 40 karaktert kiírva,
CO80 : a színes grafikus képernyőcsatolóval színes képernyőt akarunk vezérelni, soronként 80 karaktert kiírva.

Az *m* paraméter helyén az R vagy L betűk egyike állhat. Hatására a képernyőn megjelenő sorok jobbra (R) vagy balra (L) tolódnak el, egy széles, illetve két keskeny karakter szélességének megfelelő távolsággal. Ezzel a képernyő torzítását, eltérítési hibáját korrigálhatjuk.

Ha a T paramétert is megadjuk, a képernyőn megjelenik egy ellenőrző karaktersorozat, így az *m* opcióval kiváltott elcsúsztatás hatását ellenőrizhetjük, és interaktív módon további képeltolást végezhetünk.

(3) *Aszinkron adatátviteli csatoló*

A MODE paranccsal állíthatjuk be az adatátviteli protokoll paramétereit. Az *n* paraméter értéke 1 vagy 2 lehet, ennek megfelelően a COM1 vagy COM2 aszinkron csatolót állítjuk be.

Legalább a *b* paramétert, az adatátviteli sebességet kötelező megadni. A megengedett értékek: 110, 150, 300, 600, 1200, 2400, 4800 vagy 9600 baud. A számnak elég az első két számjegyét megadni, ez egyértelműen azonosítja a sebességet.

A *p* paraméter a paritásbit jelentését írja elő. A paritás lehet O (páratlan), E (páros – ez az alapértelmezés) vagy N (nincs paritásbit).

Ugyancsak specifikálható az adatbitek *d* száma (7 vagy 8) és a stopbitek *s* száma (1 vagy 2). Az adatbitek száma alapértelmezésben 7. A stopbitek számának alapértelmezése 110 baud sebességnél 2, egyébként 1 bit.

A **P** paraméterrel jelezhetjük, ha az adatátviteli vonalon soros interface-szel rendelkező nyomtatót működtetünk. Time-out hiba esetén a rendszer ismételten próbálkozik a kiírással. Ha a hiba nem szüntethető meg, a [CTRL] [C] lenyomásával szakíthatjuk meg az output műveletet.

Ha a parancsot P opció nélkül ismét kiadjuk, a time-out hibák esetén nem próbálkozik újra a rendszer, hanem hibajelzést ad. A P paraméter első megadásának hatására a MODE parancsot végrehajtó program egy része a tárban marad (rezidens lesz), tehát a legközelebbi rendszerhívásig valamelyest csökken a felhasználható központi tár területe.

(4) *Nyomtatásátirányítás*

A MODE parancs e formájában a *p*-edik nyomtatóra küldött sorokat az *n* sorszámú adatátviteli csatolóra irányíthatjuk. *p* értéke 1, 2 vagy 3 (LPT1, LPT2, illetve LPT3), *n* értéke 1 vagy 2 (COM1, illetve COM2) lehet.

A (4) formátumú MODE parancs kiadása előtt kötelező egy (3) formátumú MODE parancsot is kiadni, P paraméterrel.

A nyomtató átirányítását az (1) alakú MODE paranccsal szüntethetjük meg.

Példák

(1) *Nyomtató*

MODE LPT1:132,8

Sűrű nyomtatást állítunk be: 132 karakter/sor, 8 sor/inch.

MODE LPT1: ,6,P

A sorhossz változtatása nélkül, 6 sor/inch sorközt állítunk be. Kérjük az ismételt próbálkozást time-out hiba esetén.

(2) *Display-csatoló*

MODE CO80

A színes képernyőn soronként 80 karaktert akarunk megjeleníteni.

MODE 40

A képernyőt 40 karakteres üzemmódra állítjuk át.

MODE ,R,T

A képet jobbra elcsúsztatjuk, mert a sorok eleje nem olvasható a képernyőn. Kérjük az ellenőrző sor megjelenítését.

(3) *Aszinkron adatátviteli csatoló*

MODE COM2:60,O,,,P

A 2. aszinkron csatolón keresztül soros interface-szel rendelkező nyomtatót kívánunk működtetni. Az átviteli sebesség 600 baud, páratlan paritással, 7 adatbit és 1 stopbit alapértelmezéssel.

MODE COM1:12

Az 1. aszinkron csatolón 1200 baud sebességgel akarunk adatokat forgalmazni.

(4) *Nyomtatásátirányítás*

MODE LPT3:=COM2

Az LPT3 nyomtatóra menő outputot a COM2 csatolóra irányítjuk át. Ez a parancs csak a fenti (3) formátumú paranccsal előkészített környezetben adható ki!

MORE

Feladata

A standard input perifériáról beolvasott adatokból egyszerre egy képernyőnyit küld a standard output perifériára, majd egy tetszőleges billentyű lenyomására adja a következő lapot.

Paraméterezése

MORE

Használata

A MORE parancsot általában más programok eredményeinek megjelenítésénél és a szövegfile-ok kiíratásánál használjuk. Ehhez a standard input/output átirányításának lehetőségét vagy a parancsláncolás operátorát vehetjük igénybe.

Példák

MORE < RAJZ.BAS

A RAJZ.BAS nevű szövegfile-t (BASIC programot) íratjuk ki a standard output perifériára. Minden 24. sor után a képernyő alján megjelenik a

– Tovább –

üzenet, és a kiíratást csak egy (tetszőleges) billentyű lenyomásával folytathatjuk. A [CTRL] [C] lenyomásával a kiíratást megszakíthatjuk (befejezhetjük).

DIR | MORE

A könyvtárkiírató DIR parancs outputját laponként nézhetjük meg a képernyőn.

Feladata

A paranccsal kijelölhetők azok a könyvtárak, ahol az általunk kiadott parancsokat végrehajtó programok keresendők, ha az aktuális könyvtárban nem találta meg őket a rendszer.

Paraméterezése

PATH [*m*:] *út* [; [*m*:] *út*...]

vagy

PATH

Használata

A parancsban több, pontosveeszővel elválasztott meghajtó- és útspecifikáció sorolható fel. Az operációs rendszernek adott parancsokat végrehajtó programokat, illetve batch file-okat a PROPOS–16 először mindig a feltételezett vagy a parancs neve előtt kijelölt meghajtó aktuális könyvtárában keresi. Ha itt nem találja, sorra veszi a legutolsó PATH parancsban megadott könyvtárakat, és csak akkor jelez hibát, ha egyikben sem találja a keresett azonosítót. Az aktuális könyvtár kijelölése nem változik meg.

A PATH parancsok közül mindig az utoljára megadott parancsban rögzített könyvtárnevek jelentik a lehetséges keresési utakat. Az éppen érvényes lista a PATH parancs paraméterek nélküli kiadásával tudható meg.

Ha több könyvtárban szerepelnek azonos nevű programok, a keresés eredménye függ a PATH parancsbeli felsorolás sorrendjétől. *A parancs csak a .COM, .EXE és a .BAT kiterjesztésű file-ok keresésénél ad alternatív utakat!*

Példák

PATH

Az alternatív keresési utak éppen érvényes listáját kérdezzük.

PATH ;

Nem adunk meg további utat, mindig csak az aktuális könyvtárban kell keresni.

PATH \

Az aktuális könyvtáron kívül a mindenkori feltételezett meghajtó gyökérkönyvtárában is kell keresni. Ez a parancs leggyakoribb felhasználási módja.

PATH E:\; E:\EXPORT; E:\IMPORT

Az aktuális könyvtáron kívül az E meghajtó gyökérkönyvtárában, valamint EXPORT és IMPORT alkönyvtáraiban keresendők a hívott programok.

PRINT

Feladata

Egy vagy több szövegfile-t sorba állít, majd tartalmát kinyomtatja. A nyomtatás ideje alatt a számítógépen más feladatokat végezhetünk.

Paraméterezése

PRINT [/T]

vagy

PRINT [*m*:] [*út*] *file-név* [. *kit*] [...] [/C]

Használata

A program a kinyomtatandó file-ok azonosítóját egy ún. várakozási sorban tárolja. Ha az első file tartalmát kinyomtatta, veszi a következőt, mindaddig, míg a sor ki nem ürül. Újabb PRINT parancsokkal a várakozási sor végéhez fűzhetünk további file-azonosítókat.

Paraméterek nélküli formájában a parancs a várakozási sorában levő file-ok listáját adja ki.

A /T paraméter megadásával (ekkor más paraméter nem szerepelhet a parancssorban) a teljes várakozási sort töröljük, az éppen folyamatban levő nyomtatás is megszakad, és a nyomtató a következő lap tetejére pozícionál. Listát kapunk a törölt várakozási sorról.

A parancs másik formájában egy vagy több file- vagy file-csoport-hivatkozás áll. A /C paraméter egyidejű kiadásával a felsorolt file-ok törlődnek a várakozási sorból. Az éppen nyomtatás alatt álló file kiíratását is abbahagyhatjuk így. A várakozási sorban megmaradó file-ok kiíratása folytatódik.

A /C paraméter *nélküli* parancssorok a várakozó file-ok listáját bővítik. File-csoport-azonosítók is megadhatók, de egyidejűleg legfeljebb 10 file várakozhat nyomtatásra. A nyomtatás független a feltételezett meghajtótól és az aktuális könyvtártól.

A parancs első elindításakor megkérdezi a listázó periféria nevét:

```
Nyomtato azonosito [PRN]:
PRINT rezidens resze betoltve
```

Ha a PRN perifériára kívánunk nyomtatni, elég az [ENTER] billentyűt lenyomni. Kijelölhetjük még az LPT1, LPT2, LPT3, COM1, COM2, AUX, ... berendezéseket is.

A nyomtatás során a program a TAB karaktereket kiterjeszti a nyolccal osztható sorszámú (8, 16, 24, ...) pozíciókig.

A PRINT parancs első kiadásakor a végrehajtó program betöltődik – így a nyomtatás alatt csak kevesebb szabad tárterület áll rendelkezésünkre. A nyomtatás ideje alatt a számítógépet más feladatok elvégzésére is használhatjuk, a következő megszorításokkal:

- a kijelölt nyomtatóeszközt nem vehetjük igénybe,
- a nyomtatásra várakozó file-okhoz nem fordulhatunk sem módosítás-, sem törlésigénnyel,
- ha a nyomtatandó file-ok hajlékony lemezen vannak, a lemezt nem cserélhetjük ki.

Ha a program valamelyik file olvasása közben lemezhibát észlel, a file nyomtatása hibaüzenet kiírásával megszakad. A következő várakozó file kiírása új lapon kezdődik.

PRINT

Példák

PRINT PROG*.BAS

A feltételezett meghajtó aktuális könyvtárából legfeljebb 10, a file-csoport leírásának megfelelő BASIC forrásprogram a nyomtató várakozási sorába kerül, majd megkezdődik kinyomtatásuk. A parancs kiadása után a feltételezett meghajtó és/vagy az aktuális könyvtár kijelölését megváltoztathatjuk, ez a nyomtatást már nem befolyásolja.

PRINT PROG5.BAS /C

A feltételezett meghajtó aktuális könyvtárában tárolt file-t kivesszük a nyomtató várakozási sorából, illetve megszakítjuk kinyomtatását, ha már elkezdődött. (Itt is megadhatnánk file-csoport-azonosítót, vagy felsorolhatnánk több file-nevet is.)

PRINT

A várakozási sorban álló file-ok azonosítóinak listáját íratjuk ki a standard output perifériára.

```
C:MENUSYS .BAT nyomtatasa folyamatban
C:AUTOEXEC.BAT varakozik
```

PRINT /T

A nyomtatásra várakozó file-ok listáját töröljük, a folyamatban levő nyomtatást megszakítjuk.

Feladata

Fizikailag sérült lemezfile-ok, illetve könyvtárak helyreállítását végzi, a hibás szektor(ok) kiküszöbölésével.

Paraméterezése

RECOVER *m*:

vagy

RECOVER [*m*:] [*út*] *file-név* [.*kit*]

Használata

A parancs első alakjában egy teljes lemez összes file-ját, második alakjában pedig egyetlen file-t vizsgál meg. Ha hibás szektort talál, azt kihagyja a file-ból, és így egy rövidebb (hiányos), de fizikailag ép, feldolgozható file-t hoz létre. A második alaknál élhetünk a feltételezett meghajtó és az aktuális könyvtár használatával is. File-csoport-név megadása esetén (? vagy *) csak az első illeszkedő file javítását végzi el a program. Minthogy az ellenőrzés kizárólag a fizikai olvashatóságra vonatkozik, és a parancs teljes allokációs egységek vizsgálatát végzi, előfordulhat, hogy az új file byte-okban kifejezett mérete nagyobb lesz az eredetinél.

Szövegfile-ok esetében a file végén levő felesleges részt a szövegszerkesztő programmal vághatjuk le. A hibásnak talált lemezterületeket a program megjelöli, így ezek a későbbiekben sem kerülhetnek felhasználásra.

Egyetlen file helyreállítása esetén a file neve nem változik meg. A teljes lemez javítása esetén – amire általában a könyvtárak sérülésekor van szükség – a felismert és helyreállított file-ok és alkönyvtárak rendre a FILE0000.REC, FILE0001.REC stb. elnevezést kapják. Ezek egyenkénti megvizsgálásával, kiíratásával állapíthatjuk meg, hogy melyik korábbi file-unknak felelnek meg.

A parancs alkalmazása csak akkor szükséges, ha fontos információt tartalmazó lemezünk sérült meg. A helyreállítás fáradságos művelete általában elkerülhető, ha file-jainkról rendszeresen másolatot készítünk (l. BACKUP, DISKCOPY és COPY parancsok).

Példák

RECOVER A:

Az A meghajtóba helyezett hajlékony lemez megsérült, a teljes lemez helyreállítása szükséges. Új file-nevek jönnek létre!

RECOVER \SZAMLA\EXPORT.84

A feltételezett meghajtó SZAMLA alkönyvtárában megsérült EXPORT.84 nevű file helyreállítását kérjük. A file neve nem változik meg!

RENAME

Feladata

A megadott file(ok) nevét és/vagy kiterjesztését módosítja.

Paraméterezése

REN [**AME**] [*m* :] [*út*] *file-név* [. *kit*] *file-név* [. *kit*]

Használata

A parancs akár RENAME, akár REN névvel kiadható. Az első file-hivatkozás az átnevezendő file-t vagy file-csoportot határozza meg. A második file-azonosító lesz az új név, illetve kiterjesztés. Ez értelemszerűen nem tartalmazhat meghajtó- és útkijelölést, mivel az azonos kell legyen az eredetivel.

File-csoport-átnevezéskor listát kapunk az érintett file-ok régi és új nevéről.

Rejtett file-ok és alkönyvtárak neve nem változtatható meg.

Példák

RENAME A:PROG LISTAZO

Az A meghajtó aktuális könyvtárában a PROG nevű file-t átnevezzük LISTAZO-nak.

REN *.PRN *.LST

Minden .PRN kiterjesztésű file azonosítóját .LST kiterjesztésűre változtatjuk.

REN E:\LIB1\PROGRAM *.PAS

Az E meghajtó LIB1 alkönyvtárában található PROGRAM file-névhez .PAS kiterjesztést fűzünk hozzá. Itt a második file-azonosítóban a * jel nem file-csoportra utal, hanem az elsődleges file-név változatlanul hagyására ad utasítást.

Feladata

A BACKUP paranccsal létrehozott lemez(ek)ről visszatölti a file-ok tartalmát egy másik lemezre.

Paraméterezése

RESTORE *m*: [*m*:] [*út*] [*file-név* [.*kit*]] [/S] [/P] [/C]

Használata

A BACKUP paranccsal kimentett file-okat csak a RESTORE parancs segítségével tölthetjük vissza. Ez leggyakrabban hajlékony lemez(ek)ről a fix lemezen levő könyvtár(ak)ba történik. A parancs első (egyben egyetlen kötelező) paramétere az input meghajtó azonosítója.

A file-ok elhelyezésére vonatkozó paramétereket csak akkor kell megadni, ha nem a feltételezett meghajtón, illetve az aktuális könyvtárban kívánjuk tárolni azokat.

A mentéshez hasonlóan a visszatöltés során is megszabhatjuk, hogy melyik file-t vagy file-csoportot másoljuk át. Az *út* megadásával a visszatöltött file-ok alkönyvtárakban is elhelyezhetők. A BACKUP program az elmentett file-ok mellé feljegyzi, hogy milyen (al)könyvtárból másolta át azokat. A visszatöltés csak ugyanide megengedett!

A /S paraméter megadásával a kijelölt (al)könyvtár alá tartozó valamennyi alacsonyabb szintű alkönyvtár tartalma is visszatölthető.

Előfordulhat, hogy egyes file-ok tartalma megváltozott a legutolsó mentés óta, így a visszatöltéssel egy régebbi, esetleg már elavult változatot állítanánk vissza. A /P paraméterrel kérhetjük, hogy a RESTORE program jelezze a megváltozott file-ok azonosítóját, és csak a gépkezelő jóváhagyása esetén végezze el a visszatöltést. Hasonló módon a csak olvasható (read-only) file-ok felülírása is csak operátori engedéllyel történhet meg (l. CHMOD parancs).

A /C paraméterrel a visszatöltés utáni ellenőrzést kérhetünk.

Ha a BACKUP paranccsal több lemezre mentettük file-jainkat, visszatöltésnél a mentéssel azonos sorrendben kell a lemezeket beolvasni, függetlenül attól, hogy a teljes állományt másoljuk-e vissza vagy csak egyes részeit.

A RESTORE parancs befejezési kódjai (batch alkalmazásokhoz):

0 : normál befejeződés
1 : a visszatöltendő file-ok nem találhatók a kijelölt lemezen vagy könyvtárban
3 : hiba miatt félbeszakadt a visszatöltés

RESTORE

Példák

RESTORE A:

A feltételezett meghajtó a fix lemez, ennek aktuális könyvtárába töltjük vissza az A meghajtóba helyezett floppy lemez(ek) tartalmát. Csak azok a file-ok kerülnek átmásolásra, amelyeket ebből a könyvtárból mentettünk ki!
A képernyőn látható párbeszéd:

```
A forras meghajto [A:]: A:
A cel filespec.: *.*
Opciok ( S, P, C ):
Helyezze a(z) 1. lemezt a(z) A meghajtoba!
Usson le egy billentyut!
```

RESTORE A: C:\SZAMLA\EX*.*

Az A meghajtóról a C meghajtó gyökérkönyvtára alá rendelt SZAMLA alkönyvtárba töltjük vissza az onnan korábban kimentett, EX kezdetű azonosítóval rendelkező file-okat.

RESTORE B: E: /SP

A floppy lemezeket a B meghajtón olvasva, az E fix lemez teljes tartalmát szeretnénk visszatölteni. A mentés óta megváltozott file-ok felülírása előtt figyelmeztetést kérünk, pl.

```
A(z) PROBA.C file az utolso elmentes ota megvaltozott.
Visszatoltse? (i/n) IGEN
```

RMDIR

Feladata

Megszüntet (töröl) egy üres alkönyvtárat.

Paraméterezése

RMDIR [*m*:] [*út*]

vagy

RD [*m*:] [*út*]

Használata

A parancs lehetővé teszi a megadott út végén levő alkönyvtár megszüntetését. Csak olyan alkönyvtár szüntethető meg, amely nem tartalmaz bejegyzést (kivéve az azonosítási célokat szolgáló "." és ".." tételeket). A gyökérkönyvtár és az aktuális könyvtár nem szüntethető meg.

Példa

RD A:\BER.84

Az A meghajtó gyökérkönyvtára alatti, BER.84 nevű alkönyvtárat megszüntetjük.

Feladata

Tájékoztatást ad a konfiguráció software jellemzőiről, és lehetővé teszi a paraméterek beállítását.

Paraméterezése

SET

Használata

A paraméterek nélkül kiadandó parancs interaktív üzemmódban működik. Először kiírja az operációs rendszer nevét és verziószámát, majd lehetővé teszi a dátum és idő beállítását.

A **dátum** megadásnak kötelező formája: az év utolsó két jegye vagy valamennyi jegye, továbbá a hónap és a nap egy- vagy kétjegyű számmal megadva, pl. 85/1/15, vagy 85/01/15, vagy 1985/1/15.

Elválasztójelként kötőjel is használható, pl. 85-10-6, vagy 85-10-06, vagy 1985-10-6.

Az **időpont** megadásának módja: az óra, perc és másodperc adatokat kettősponttal elválasztva adhatjuk meg (a másodperc 1/100 pontossággal, tizedes számként írható), pl. 8:30:20, vagy 08:30:20, vagy 8:30:20.17.
Más formában a rendszer nem fogadja el ezeket az adatokat. Ha nem kívánunk új dátumot, illetve időpontot megadni (munkánk szempontjából közömbös ezek tartalma), egyszerűen az [ENTER] billentyű lenyomásával léphetünk tovább. Ne feledjük azonban, hogy a PROPOS–16 rendszer minden lemezfile és alkönyvtár létrehozásának, illetve utolsó módosításának dátumát és időpontját feljegyzi, amit később felhasználhatunk ellenőrzési, mentési stb. célokra. A dátum és időpont megadása tehát fontos lehet.

BREAK

A BREAK paraméter állításával (BREAK=OFF vagy BREAK=ON) a [CTRL] [BREAK], illetve [CTRL] [C] billentyűk lenyomásának figyelését vezérelhetjük. OFF megadása esetén a programok működése csak konzol, nyomtató és aszinkron kommunikációs tevékenység végrehajtásakor szakítható meg. ON opcióval minden operációs rendszer híváskor megszakíthatjuk a programok futását. Ha a jelenlegi állapotot nem kívánjuk módosítani, csak az [ENTER]-t kell lenyomni. A BREAK állapot a CONFIG paranccsal is lekérdezhető, illetve módosítható.

VERIFY

A VERIFY paraméter állításával a lemezírás műveletek utáni ellenőrzés kérhető (VERIFY=ON), illetve letiltható (VERIFY=OFF). Az ON állapotban természetesen lassabban hajtódnak végre a lemezírás műveletek, mert a rendszer vissza is olvassa a kiírt adatokat.

Ha nem akarjuk megváltoztatni a jelenlegi állapotot, csak az [ENTER]-t kell lenyomni.

SET

Paraméterdefiniálás

Az operációs rendszer segítségével tárolhatunk tetszőleges paramétereket is, amelyek azután minden futó programból elérhetők. Ily módon több programból álló rendszereket is paraméterezhetünk, működési környezetüket módosíthatjuk.

Ezeket az ún. „environment", azaz környezeti paramétereket nevükkel és értékükkel definiálhatjuk:

név1 = érték1
név2 = érték2
.
.
.

Már meglevő paramétereknek új értéket is adhatunk, és azokat törölhetjük is (l. példa).

A kisbetűvel írt paraméterneveket az operációs rendszer nagybetűsre alakítja át, a paraméterértékeket viszont a bebillentyűzött formában tárolja.

Az éppen érvényes paraméterek programból való elérésének módja a **PROPOS–16 – Programozóknak** című kézikönyvben található meg.

A **PATH** parancs is a fenti módon tárolja a parancsokat végrehajtó programok keresésénél használandó alternatív meghajtók és utak leírását.

A **COMSPEC=...** paraméter az operációs rendszer parancsértelmező (COMMAND.COM) moduljának elérési útját rögzíti, megadva azt a meghajtót, ahonnan szükség esetén újratölthető.

Az „environment" paraméterterület mérete 1 Kbyte, ez korlátozza a paraméterek számát és hosszát.

Példa

A következő párbeszéd a SET program működését illusztrálja.

```
SZKI PROPOS-16 OPERACIOS RENDSZER 3.0. valtozat
Mai datum:  1984/1/1 V
 Uj datum:  85/3/28
Pontos ido:  3:43:18.79
 Uj ido:     12:30
Iras/olvasas ellenorzes: VERIFY=OFF
            Uj ertek: (ON/OFF) = ON
   CTRL-BREAK figyeles : BREAK=OFF
            Uj ertek: (ON/OFF)=ON
Az aktualis kornyezet bejegyzesei:
COMSPEC=E:\COMMAND.COM
PATH=\
```

```
Most modosithatok, illetve torolhetok a kornyezet bejegyzesei.
Ha nem akarja modositani, usson ENTER-t!
PATH=\
PATH=C:\;A:\
Akar tovabbi bejegyzeseket beepiteni a kornyezetbe? (I/N)
I
Kerem a szoveget 'nev= parameter' formaban
PAR1=PERDATIN
Akar tovabbi bejegyzeseket beepiteni a kornyezetbe? (I/N)
I
Kerem a szoveget 'nev=parameter' formaban
PAR2=GRAFOS
Akar tovabbi bejegyzeseket beepiteni a kornyezetbe? (I/N)
N
COMSPEC=E:\COMMAND.COM
PATH=C:\;A:\
PAR1=PERDATIN
PAR2=GRAFOS
```

SORT

Feladata

A parancs a megadott szövegfile-ok sorait a paraméterek által specifikált módon rendezi. Lehetőséget nyújt több bemenő file egyetlen, rendezett eredményfile-ba történő összeolvasztására is.

Paraméterezése

SORT [*m*:] [*út*]*file-név* [.*kit*] [:..] [+*pos*1] [-*pos*2] [/C] [/M] [/U] [/B]
[/D] [/F] [/I] [/N] [/R] [/Tx]

Használata

A rendezés bemenő adataként egy vagy több file-, illetve file-csoport-hivatkozás adható meg. A "-" jellel a standard inputra hivatkozhatunk, bármelyik file-név helyén. A rendezés soronként (rekordonként) történik, a rendezett lista a standard output perifériára kerül, amely átirányítható file-ba is. A rendezett file az egész sor alapján, betűrend (ASCII kód) szerint növekvő sorrendben áll össze.

Ha a rendezést csak a sorok egy része alapján kívánjuk végezni, kijelölhetünk egy rendezési kulcsot is. A kulcs kezdő- és végpozicióját a +*pos*1 -*pos*2 paraméterekkel írhatjuk elő. Bármelyik elhagyható, ekkor alapértelmezés az első, illetve utolsó karakter a sorban. Mindkét pozíciót *m*·*n* alakban adhatjuk meg, ahol *m* és *n* egész számok: *m* a soron belüli mező sorszáma, *n* a mezőn belüli karakter sorszám. A sorszámozás nullától kezdődik. Mezőnek tekintünk minden olyan karaktercsoportot, amely előtt és után space (betűköz), TAB (tabulátor) vagy kocsi-vissza karakter áll, és közbül ezek az elválasztó karakterek nem fordulnak elő. Ily módon tehát rendezési kulcsként egyetlen, összefüggő karaktersorozat határozható meg, amely több szót is átfoghat, és a rekord elejétől számított kezdőpozíciója rekordonként máshol is lehet (ha az előtte álló mezők hossza változó).

A /C paraméter hatására tényleges rendezés nem történik, a SORT parancs csak kijelzi, hogy az adott file(ok) rendezettsége megfelel-e a parancs paramétereivel előírtaknak.

A /M opció megadásának több, rendezett bemenő file(-csoport) esetén van jelentősége. Az eredmény mindig egyetlen, összeolvasztva rendezett lista – illetve átirányítás esetén file – lesz. Ha azonban a /M paramétert elhagyjuk, a bemenő file-ok rendezettségét nem feltételezi, ezért a rendezés lassabb lesz.

A /U paramétert akkor kell megadnunk, ha a rendezés eredményeként létrejött listában, illetve file-ban az azonos sorokból csak egyet akarunk megtartani.

A /B opció lehetővé teszi, hogy a bemenő sorok elején levő space és TAB karaktereket a rendezésnél figyelmen kívül hagyhassuk.

A rendezés szempontjait írhatjuk elő a következő paraméterekkel:

A /D megadásával a rendezésből kizárunk minden karaktert a betűkön, számokon és a betűközön kívül.

A /F hatására a SORT parancs nem tesz különbséget a kis- és nagybetűk között. A /D-vel együtt is megadható.

A /I opció megadásakor a rendezésnél csak a 40–176 (hexadecimális 28–B0) közötti ASCII karaktereket vesszük figyelembe.

A /N megadásával a rendezést csak számok alapján végezzük.

A /R opcióval a fordított sorrend előállítását kérhetjük (csökkenő ASCII kód szerint). Ez a paraméter megadható a /D, /F, /I vagy /N opcióval együttesen is.

A sorok mezőkre bontásának módját változtathatjuk meg a /Tx paraméterrel. A T után közvetlenül álló karaktert tekinti a parancs mezőelválasztónak a betűköz és tabulátor helyett.

Példák

SORT NEVSOR

A feltételezett meghajtó aktuális könyvtárában tárolt NEVSOR file-t rendezzük betűrendbe, és a rendezés eredményét a standard output periférián jelenítjük meg.

```
C:\ SORT NEVSOR
Opciok(k):
Kulcs:
Antal Imre
Bartok Bela
Csortos Gyula
Devecseri Gabor
Eszterhazy Peter
Fekete Gyula
Gabor Andor
Hernadi Gyula
```

Rendezhetjük a névsort a második mező, a keresztnevek szerint is:

```
C:\ SORT NEVSOR +1.0 -1.5
Opciok(k):
Gabor Andor
Bartok Bela
Devecseri Gabor
Csortos Gyula
Fekete Gyula
Hernadi Gyula
Antal Imre
Eszterhazy Peter
```

SORT CIM*.*>LPT1

Minden CIM kezdetű azonosítóval rendelkező file-t egyesítünk és rendezünk. A rendezett listát a standard outputról az LPT1 nyomtatóra irányítjuk át.

SORT A:NEVSOR1 B:NEVSOR2 /MUF>NEVSOR.UJ

Az A és B meghajtókon található két rendezett file-t egyesítve rendezzük, és az eredményfile-ban az azonos sorokból csak egyet tartunk meg. Kis- és nagybetűk között nem teszünk különbséget. Az eredményt a feltételezett meghajtó aktuális könyvtárában, NEVSOR.UJ file-azonosító alatt tároljuk.

SORT NEVSOR.UJ /CF

A file rendezettségét ellenőrizzük.

SORT BERLISTA.FEB +6.0 -6.7 /NR

A feltételezett meghajtó aktuális könyvtárában levő file-t a 7. mező 1–8. (numerikus) karakterei alapján, fordított sorrendbe rendezzük, és az eredményt a standard output perifférián jelenítjük meg (pl. bérlista növekvő fizetés szerint rendezve).

Feladata

A könyvtárak és a file-allokációs tábla (FAT) tartalmát elemzi, és kiírja a lemez főbb jellemzőit.

Paraméterezése

STAT [*m*:] [*út*] [*file-név* [.*kit*]] [/F] [/V]

Használata

A parancs paraméterként megadott, illetve feltételezett meghajtó fix vagy hajlékony lemezes egység is lehet. Leggyakoribb – paraméter nélküli – használatakor a STAT parancs elemzi a lemezen levő könyvtárakat, és táblázatot készít a lemez kapacitásának és kitöltöttségének adatairól. Azt is közli, hogy a lemezen mekkora terület az, amely hibák, módosítások következtében elérhetetlenné vált, pl.:

- ha file módosításának eredményeként a file rövidebb lett, a mögötte levő területet a rendszer nem szabadítja fel;
- ha egy eredmény (kimeneti) file-t nem sikerül lezárni – programhiba miatt, a lemez megtelt vagy operátori megszakítás következtében –, a felírt terület használhatatlan.

```
2 darab lancban 2 elveszett allokacios egyseg
Az F parameter hianyzik, a talalt hibakat nem javitom
   A lemezen 7966720 byte-bol 86016 byte szabad
   20 alkonyvtarban 86016 byte
   2 rejtett file-ban 28672 byte
   593 normal file-ban 3825664 byte
   A kozponti tarban 262144 byte-bol 231664 byte szabad
```

A lemezen azonban ekkor semmi sem változik. Mint láthatjuk, a táblázat a rejtett file-ok méretét (operációsrendszer-modulok és kötetazonosító), valamint a központi tár teljes és szabad kapacitását is tartalmazza.

A /F paraméter megadásával kérhetjük, hogy az elveszett területeket file-okká fűzze össze a program. A létrehozott file-ok azonosítója

FILE*nnnn*.CHK

alakú lesz, ahol *nnnn* = 0000, 0001, ...

A STAT által létrehozott file-ok szükség szerint törölhetők (így az elveszett terület ismét használhatóvá válik) vagy kiírathatók, átnevezhetők, újra használhatók. Ezek a file-ok mindig a gyökérkönyvtárban találhatók!

A /V opció hatására a program részletesebb információt szolgáltat működésének menetéről, a talált hibákról.

A STAT program alkalmazásának másik módja, ha a parancsnak file- vagy file-csoport-nevet is megadunk. A program ekkor a fenti táblázat elkészítése után kijelzi ezen file(ok) nem összefüggően elhelyezkedő részeinek számát is. A művelet az aktuális vagy az adott úttal kijelölt könyvtárban levő file-okra vonatkoztatva kérhető. Ha a vizsgált file erősen széttöredezett, sok részből áll, feldolgozásának hatékonysága igen leromolhat. Ilyenkor segíthet a file átmásolása (l. COPY parancs), és a régi példány törlése.

Célszerű a STAT programot rendszeresen lefuttatni fix és hajlékony lemezeinkre!

Példák

STAT

A feltételezett meghajtó főbb kapacitásadatairól kapunk tájékoztatást.

```
A filespec: :
Opciok ( F, V ):
   A lemezen 362496 byte-bol 344064 byte szabad
   0 alkonyvtarban 0 byte
   0 rejtett file-ban 0 byte
   1 normal file-ban 18432 byte
   A kozponti tarban 262144 byte-bol 232304 byte szabad
```

STAT

STAT A:*.EXE

Az A meghajtóban levő floppy lemezen található, .EXE kiterjesztésű file-ok szét-töredezettségéről kérünk statisztikát.

```
A(z) A:\O.EXE file 2 nem osszefuggo blokkbol all
A(z) A:\O.EXE file altal elfoglalt clusterek szama: 41 ( =20992 byte )
     ez nincs osszhangban a konyvtarbejegyzesben szereplo hosszal (48000 byte)!
Kettos allokalasu egyseg: A:\ST.EXE file 102. cluster
Kettos allokalasu egyseg: A:\O.EXE file 102. cluster
     A lemezen 179712 byte-bol 109568 byte szabad

     A kozponti tarban 262144 byte-bol 231664 byte szabad
```

STAT E: /F

Az E meghajtón található elveszett területeket összefűzzük a FILE0000.CHK, a FILE0001.CHK stb. azonosítójú file-okba.

Feladata

Egy szövegfile tartalmát megjeleníti a standard output periférián.

Paraméterezése

TYPE [[*m*:][*út*]*file-név* [.*kit*]]

Használata

Paraméter nélküli formájában a bemenő adatsorokat a standard input perifériáról várja.

Bemenő file-ként file-csoport nem adható meg.

A parancs a listán a TAB karaktereket kiterjeszti a nyolccal osztható sorszámú pozíciókig (8, 16, 24, ...).

Példák

TYPE CON > PRN

A parancs bemenete a standard input periféria (billentyűzet), outputját átirányítjuk a nyomtatóra, ily módon közvetlenül a nyomtatóra írhatunk.

TYPE B:PROGRAM.LST

A B meghajtóról az adott listafile-t írathatjuk ki.

TYPE ADAT \FORGALOM.84

Az aktuális könyvtár alatti ADAT alkönyvtárból a FORGALOM.84 azonosítójú file-t listázzuk ki a standard outputra.

VOL

Feladata

A lemezazonosító címke kiíratását teszi lehetővé.

Paraméterezése

VOL [*m*:]

Használata

A parancs a megadott, illetve a feltételezett meghajtóban levő lemez azonosítóját megkeresi (ha van rajta), és kiíratja a standard output perifériára.

Példák

```
A:\VOL C:
"C" meghajtoban levo lemez : cimke nelkuli
```

```
A:\VOL
"A" meghajtoban levo lemez : HORVATH1
```

EDIT

5. Az EDIT sorszerkesztő program

Ebben a fejezetben a programok, szövegfile-ok létrehozására, kiíratására és javítására szolgáló EDIT programot ismertetjük.

Az EDIT program minden olyan file kezelésére használható, amely kiíratható karaktereket tartalmazó, olvasható formátumú sorokból áll. Ezek a sorok lehetnek bármelyik programnyelven írt utasítások, adatrekordok vagy szövegfile sorai, mondatai. Az EDIT programmal a következő műveleteket végezhetjük:

- új file létrehozása és lemezre írása,
- a file egészének vagy egyes sorainak kiíratása a képernyőre,
- sorok törlése és beszúrása,
- soron belüli módosítások,
- adott szövegrészek kikeresése, helyettesítése,
- a módosított file lemezre írása.

A szövegfile-ok sorai különböző hosszúságúak lehetnek (max. 253 karakter). A bevitelnél a sor végét az [ENTER] lenyomásával jelezzük.

A file-ok kiíratásakor az EDIT program minden sorhoz automatikusan egy sorszámot rendel, ezt azonban a lemezen nem tárolja. A számozás 1-től egyesével megy a legutolsó sorig. Új sor beszúrásakor vagy törlésekor a program azonnal automatikusan átsorszámozza a mögötte levő sorokat.

5.1. Az EDIT program elindítása

A programot a PROPOS–16 parancsokhoz hasonló módon kell betölteni és elindítani.

EDIT [*m*:] [*út*]*file-név* [.*kit*] [/**B**]

Ha a megadott file-hivatkozással nincs még file a lemezen, az EDIT feltételezi, hogy ilyen azonosítójú *új file-t kívánunk létrehozni.* Megjelenik a képernyőn az

Uj file

*_

üzenet, jelezve, hogy új file létrehozására kerül sor. A * jel, és a mögötte villogó kurzor mindig azt mutatja, hogy az EDIT program parancs megadására vár. Új file esetében csak az **I** (új sor bevitele) parancsnak van értelme.

Az **I** parancs hatására az EDIT program kiírja az első sor sorszámát, a sorba tetszőleges karaktersorozat írható. A sort az [ENTER] lenyomásával zárjuk le, és máris kapjuk a következő sorszámot.

```
Uj file
*I [ENTER]
    1:*Ez az elso sor. [ENTER]
    2:*Ez a masodik sor [ENTER]
    3:*^C [ENTER]

*E [ENTER]
```

Mint a példában is látható, a sorok bevitelét a [CTRL] [C] billentyűk lenyomásával fejezhetjük be.
Az E parancs segítségével a létrehozott új szövegfile-t lemezre írhatjuk.

Ha az EDIT hívásakor megadott file-hivatkozás létező file-ra utal, a program megkísérli a *javítandó file betöltését* a központi tárba. Ha a szövegfile elfér az EDIT pufferterülete 3/4 részében (kb. 45 Kbyte méret alatt), az egész file-t betölti. Ennek megtörténtét az

Input file vege

*_

üzenet megjelenése, és a parancsra várakozó állapot jelzi.

A betöltött file tetszőleges sorait javíthatjuk, módosíthatjuk vagy kiírathatjuk a képernyőre. Az ehhez használható parancsokat a továbbiakban részletesen ismertetjük.

A file módosítása után a megváltozott file-t is az EDIT E parancsával írathatjuk vissza a lemezre. A file felírása előtt az EDIT program az eredeti file-t .BAK kiterjesztésűre változtatja, és a módosított file-t az eredeti azonosítóval, de új fileként írja a lemezre. Ügyeljünk tehát, hogy a lemezen legyen elegendő hely a módosított file felírására!
Ha a lemez adott könyvtárában volt már ilyen nevű, .BAK kiterjesztésű file, az törlődik, és helyébe az új .BAK file lép.

Előfordulhat, hogy a lemezen levő szabad hely csak a javított file egy részének visszaírására elegendő. Ekkor ez a töredék file .$$$ kiterjesztéssel kerül felírásra, az eredeti file és a régi .BAK file (ha volt ilyen) pedig megmaradnak. A fel nem írt javítások, sajnos, elvesznek.

Példa

Tegyük fel, hogy a kijelölt (vagy feltételezett) meghajtó megadott (vagy aktuális) könyvtárában van PROG1.C és PROG1.BAK nevű file, amelyek programunk utolsó és utolsó előtti állapotát tartalmazzák. A PROG1.C javításával új PROG1.C file-t hozunk létre. A régi PROG1.C-ből PROG1.BAK lesz, a régi PROG1.BAK pedig elvész.

Ha az előző .BAK file-t is meg akarjuk őrizni, nevezzük át, vagy másoljuk át másik könyvtárba vagy másik lemezre.

Az EDIT programmal nem fordulhatunk .BAK kiterjesztésű file-hoz. Ezeket a rendszer mentési (backup) file-nak tekinti, ezért módosításukat nem engedélyezi. Ha mégis ilyen file-t kell módosítanunk, az EDIT hívása előtt változtassuk meg azonosítóját (l. RENAME parancs).

Ritkán fordul elő, hogy olyan *nagy szövegfile-t* akarunk kezelni az EDIT programmal, amely már nem fér el a pufferterület 3/4 részében. Ilyenkor csak a file első része töltődik be, és ezt az EDIT program parancsra várakozó állapotát jelző * kiírása mutatja.

A file betöltött részét a szokásos eszközökkel javíthatjuk vagy kiírathatjuk a képernyőre. Ha a file első részével nincs több dolgunk, a **W** paranccsal írjuk vissza a lemezre, és az **A** parancs kiadásával olvashatjuk be a file következő részét.

A javítás befejezését, az utolsó rész visszaírását most is az **E** parancs segítségével végezhetjük el.

A szövegfile-ok végét a lemezen általában a CTRL–Z (hexa 1A) karakter jelzi. Előfordulhat azonban, hogy egy file belsejében is található CTRL–Z kód, ekkor az EDIT hívásakor megadott **/B** paraméterrel kell utasítanunk a programot a CTRL–Z kódok figyelmen kívül hagyására.

5.2. Az EDIT parancsok paraméterezése

Az EDIT program sorszerkesztő funkcióit egybetűs parancsokkal válthatjuk ki. A parancsokat a paraméterek egy része megelőzi, más része a parancs betűkódja mögött áll. A parancssor végét az [ENTER] lenyomásával jelezzük.

Az EDIT parancsok a PROPOS–16 többi parancsához hasonlóan a [CTRL] [C] billentyűvel állíthatók le. Hosszú kimenő lista készítését a [CTRL] [S] lenyomásával függeszthetjük fel, amíg megnézzük a képernyőn megjelent sorokat. A folytatást bármelyik billentyű lenyomásával kiválthatjuk.

A szövegfile sorainak bevitelét és javítását igen megkönnyítik a sorszerkesztő billentyűk (l. B. függelék).

Az EDIT sorszerkesztő programnak a következő parancsok adhatók:

Feladata	Paraméterezése
EDIT befejezése és mentés	E
EDIT befejezése	Q
File-ok egyesítése	[*sor*] T [*m* :] *file-név* [.*kit*]
Kiírás	[*sor*] [,*sor*] L
Kiírás laponként	[*sor*] [,*sor*] P
Sor áthelyezése	[*sor*] , [*sor*] , *sor* M
Sor beolvasása	[*n*] A
Sor beszúrása	[*sor*] I
Sor javítása	[*sor*]
Sor másolása	[*sor*] , [*sor*] , *sor* [,*i*] C
Sor törlése	[*sor*] [,*sor*] D
Sor visszaírása	[*n*] W
Szöveg helyettesítése	[*sor*] [,*sor*] [?] R [*szöveg*] [⟨F6⟩ *szöveg*]
Szöveg keresése	[*sor*] [,*sor*] [?] S [*szöveg*]

A parancsokat a következőkben részletesen is ismertetjük, betűjelük szerinti sorrendben (elsőnek kiemelve a betűjel nélküli sorjavítást). Előtte még néhány általános tudnivalót foglalunk össze.

Egy sorban több EDIT parancs is megadható, amelyek balról jobbra kerülnek végrehajtásra. A sorjavítás parancsot (amely egyetlen sorszámból áll) pontosvesszővel kell elválasztani az őt követő parancstól. A szöveg keresése, illetve szöveg helyettesítése parancsokat a CTRL–Z (F6) karakterrel kell lezárni a szöveg után. A többi parancsot nem kell külön jellel elválasztani az őt követő parancstól, mivel a parancsok határa egyértelműen megállapítható.

A szöveg keresését (S) és a szöveg helyettesítését végző (R) parancsok szövegparaméterében a vezérlő (CTRL) karakterek csak közvetett módon helyezhetők el. Vezérlő karakter megadásához először a CTRL–V karaktereket, majd a kívánt karaktert nagybetű állásban írjuk le. Pl. CTRL–Z helyett CTRL–V, majd Z írandó.

Sorszám megadása

A sorszerkesztés folyamán az EDIT program mindig nyilvántart egy *aktuális sort*. Ezt a képernyőre kiírt listán a sorszám mögé írt * jelzi.
Pl.

```
C:\EDIT CONFIG.SYS
Input file vege
*L
        1: *files=20
        2: buffers=16
        3: device=ansi.sys
```

Az EDIT elindításakor az aktuális sor az első sor. A továbbiakban általában az utoljára kiírt, illetve módosított sort *követő* sort jelzi a * jel.

Az EDIT parancsok legtöbbjének paraméterként megadható a javítandó sor vagy sorok sorszáma. Erre a következő lehetőségek vannak:

(1) 1–65529 közötti decimális egész szám megadása. Ha a tárban levő utolsó sor sorszámánál nagyobb értéket adunk meg, akkor az utolsó sor mögé írhatunk új sorokat.

(2) A sorszám helyett megadott # jel szintén az utolsó sor utáni file-bővítést tesz lehetővé.

(3) A sorszám helyett megadott . (pont) az aktuális sorra való hivatkozást jelzi.

(4) Az aktuális sorhoz képest adott távolságra levő sorokra előjeles számokkal hivatkozhatunk. Negatív számokkal az aktuális sor előtti, pozitívokkal a mögötte következő sorokra utalhatunk. A pozitív előjelet is kötelező kiírni!

Az EDIT parancsok egy részében sorszámintervallum is megadható. Ez két, a fenti szabályok szerint megadott számmal specifikálható. A sorszámokat vesszővel vagy betűközzel kell elválasztani.

Sor javítása

Feladata

Egy sor módosítását teszi lehetővé.

Paraméterezése

[*sor*]

Használata

A sorszámával kijelölt sor megjelenik a képernyőn, és azt módosíthatjuk.

A sorszám helyett . (pont) beírásával az aktuális sort választhatjuk ki.

Ha csak az [ENTER] billentyűt nyomjuk le, az aktuális sort követő sort kapjuk. Minthogy az aktuális sor általában az, amelyet utoljára módosítottunk, az [ENTER] lenyomásával sorba vehetjük a szövegfile egymást követő sorait.

Megjelenik tehát a kiválasztott sor sorszáma és jelenlegi tartalma, és alatta egy üres sorban a kurzor, előtte ismét a sor számával. A sorba új szöveget billentyűzhetünk, vagy a szövegkezelő billentyűk felhasználásával „előcsalogathatjuk" a régi tartalmat, és azt módosíthatjuk. Az [ENTER] lenyomásáig ugyanis az operációs rendszer *mintaként* a sor régi tartalmát őrzi, és csak az [ENTER] lenyomásakor írja azt felül az új sorral.

Ha az üres sorba nem írunk, csak az [ENTER]-t nyomjuk le rögtön a sor elején, a sor tartalma változatlan marad. Ugyanezt a hatást érhetjük el akkor is, ha a sor javítása közben meggondolva magunkat, a [CTRL] [C] vagy [ESC] billentyűket nyomjuk le.

Példa

Tegyük fel, hogy egy szövegfile 10. sorát kívánjuk módosítani. A sorszám beírása (és az [ENTER] lenyomása) után a képernyőn a következők láthatók:

```
*10
 10: Ez a  szovegfile egy sora.
 10: _
```

Az [F1] billentyűt lenyomva tartjuk, amíg az új sorban megjelennek a következő karakterek:

```
*10
 10: Ez a szovegfile egy sora.
 10: Ez a szovegfile egy _
```

Sor javítása

Nyomjuk meg az [INS] billentyűt, amely lehetővé teszi karakterek beszúrását a minta két karaktere közé, majd írjuk be a „javított" szót és egy betűközt. Az [F3] billentyű lenyomásával hívjuk elő a minta maradék részét:

```
*:10
  10: Ez a szovegfile egy sora.
  10: Ez a szovegfile egy javitott sora.
```

Ha elégedettek vagyunk a javítással, az [ENTER] billentyűvel tárolhatjuk az új sort.

Ha a mondaton belül még akarunk javítani, az [F5] billentyű lenyomásával az új mondat íródik a mintába, és ezen további javításokat végezhetünk.

Ha a mondat végéhez akarunk még írni, minden további nélkül megtehetjük.

Ha a javítást semmisnek akarjuk nyilvánítani, az [ESC] vagy a [CTRL] [C] lenyomása után a sor tartalma a régi állapotában marad meg.

A sorszerkesztő billentyűk használata részletesen megtalálható a B. függelékben.

Feladata

Adott számú sort betölt a lemezről az éppen javítás alatt álló file-ból, a memóriában levő sorok mögé.

Paraméterezése

[*n*] A

Használata

Ennek a parancsnak csak olyan file-ok javításánál van értelme, amelyek egyszerre nem tölthetők be a központi tárba.

A file további részét csak úgy javíthatjuk, illetve írathatjuk ki, ha a file elejét előbb visszaírjuk a lemezre (l. W parancs).

Az A parancs *n* paramétere a szövegfile-ból beolvasandó sorok számát adja meg. Ha nem adunk meg számot, a rendelkezésre álló pufferterület 3/4 részéig tölt be sorokat a program.

Ha ez a terület már tele van, a parancs hatástalan, nem olvas be többet a file-ból.

Ha az A parancs végrehajtása közben az EDIT program a file végéhez ér, az

Input file vége

üzenet jelenik meg a képernyőn.

Példák

25A

A szövegfile hátralevő részéből még 25 sort olvasunk be.

A

A szövegfile hátralevő részéből annyi sort olvasunk be, hogy a pufferterület ismét 3/4 részéig fel legyen töltve. Mindkét példában előzőleg a W (sorvisszaírás) parancscsal kellett helyet felszabadítanunk a tárban.

C

Feladata

A sorszámintervallummal meghatározott sorokat átmásolja a szövegfile-on belül egy adott sor elé.

Paraméterezése

[*sor*] [*sor*] , *sor* [, *i*] C

Használata

A C parancs első két paramétere az első és utolsó átmásolandó sort határozza meg. Elhagyásuk esetén az alapértelmezés az aktuális sor sorszáma, amely lehet akár az első, akár az utolsó átmásolandó sor. Ha mindkét paramétert elhagyjuk, csak az aktuális sor átmásolására kerül sor. Az elhagyott paramétereket követő vesszőt mindenképpen ki kell írni, hogy a program a következő paramétereket helyesen tudja értelmezni.

A harmadik paraméter megadása kötelező. Ez azt a sort jelöli ki, amely elé az adott sorokat be akarjuk illeszteni. Ezek a sorok azonban változatlanul megmaradnak eredeti helyükön is. A harmadik sorszám nem eshet az első két sorszám közé.

Az *i* paraméterrel az ismétlések számát adhatjuk meg, alapértelmezése 1. Egész szám, értékétől függően a kijelölt sorok többször kerülnek átmásolásra a megadott sor elé.

A másolás után az EDIT újrasorszámozza a file-t, és az első átmásolt sor lesz az aktuális sor.

Példák

, , 5 C

Az aktuális sort az 5. sor elé másoljuk át.

25,30,112 C

A 25–30. sorokat átmásoljuk a 112. sor elé.

25,30,112,3 C

A 25–30. sorokat háromszor másoljuk a 112. sor elé.

Feladata

Egy vagy több sort töröl a szövegfile-ból.

Paraméterezése

[*sor*] [,*sor*] **D**

Használata

A D parancs paraméterek nélkül az aktuális (a képernyőn *-gal jelzett) sort törli a file-ból.

Csak az első paramétert megadva az adott sorszámú, egyetlen sor törlődik. Mindkét paraméter megadásával a két sorszámmal meghatározott rész törlődik a file-ból.

Ha csak a második paramétert írjuk a parancsba (előtte vesszővel!), az aktuális sortól a megadott sorszámig törölhetünk.

A törlés után az EDIT program minden esetben átsorszámozza a kitörölt sor(ok) mögött megmaradt sorokat úgy, hogy a számozás továbbra is egyesével menjen végig a file-on.

A törlés után a törölt sor(ok) mögött megmaradt első sor lesz az aktuális sor.

Példák

D

A szövegfile aktuális sorát törli.

17D

A file 17. sora törlődik, a mögötte levő sorok eggyel kisebb sorszámot kapnak (a régi 18. sor lesz a 17., és így tovább).

17,20D

A 17., 18., 19. és 20. sorok törlődnek. A régi 21. sor kapja a 17-es sorszámot és lesz az aktuális sor.

,34D

Az aktuális sortól kezdődően a 34. sorig minden sort törlünk. A korábbi 35. sor lesz az aktuális.

E

Feladata

Befejeződik az EDIT program működése, és a javított file visszaíródik a lemezre.

Paraméterezése

E

Használata

A szövegfile ugyanolyan file-hivatkozással (lemezmeghajtó, könyvtár, file-név és kiterjesztés) íródik fel a lemezre, mint amit az EDIT elindításakor adtunk meg. Az eredeti file azonosítója megváltozik, .BAK kiterjesztést kap, függetlenül az eredeti file-név kiterjesztésétől.

Ha az EDIT programmal új szövegfile-t hoztunk létre, természetesen nem generálódik .BAK file (l. még az EDIT program általános ismertetésénél).

Az EDIT minden esetben tesz a file végére egy kocsi-vissza és egy soremelés karaktert, továbbá egy CTRL–Z karaktert, amely a file végét jelzi. A szövegfile utolsó három byte-jának hexadecimális értéke tehát mindig: 0D, 0A, 1A.

Feladata

Új sorokkal bővíti a szövegfile-t.

Paraméterezése

[*sor*] I

Használata

Ha nem adunk meg sorszámot, vagy ha a sorszám helyén a . (pont) karaktert használjuk, az aktuális sor elé szúrhatunk be új sorokat. Új szövegfile létrehozásakor is paraméter nélkül kell az I parancsot megadni.

Ha a megadott sorszám a file utolsó sora mögé mutat, vagy ha a sorszám helyén # karaktert írunk, a file végéhez írhatunk további sorokat.

A billentyűzetről beírt sorokat mindig az [ENTER] billentyűvel kell lezárni. Ezután rögtön megjelenik a kurzor a következő, üres sorban, előtte a következő sorszámmal. Minden újabb sor beírásával a mögötte levő sorok sorszáma eggyel megnő, így a folytonos sorszámozás mindig fennáll.

Az új sorok bevitelének befejezését a [CTRL] [C] lenyomásával jelezzük. Ekkor az a sor, amelyben a [CTRL] [C]-t lenyomtuk, már nem kerül a file-ba, és ismét megjelenik a * karakter, jelezve, hogy az EDIT program új parancs fogadására készen áll.

Új sorok bevitele után az utolsó új sort követő sor lesz az aktuális sor.

Példák

I

Az aktuális sor elé akarunk új sorokat beilleszteni. Új file létrehozásakor e parancs hatására az 1. sortól kezdve vihetünk be szöveget.

I

A meglevő szövegfile végéhez új sorokat szeretnénk hozzáírni.

23I

A file 23. sora elé visszük be az új sorokat. A 23. sortól kezdődően a sorszámok annyival nőnek meg, ahány új sort bevittünk.

L

Feladata

A file megadott sorait kiírja a standard outputra.

Paraméterezése

[*sor*] [,*sor*] L

Használata

Az L paranccsal a megadott két sorszám közé eső sorokat írathatjuk ki a képernyőre.

Paraméterek nélküli alakban használva az aktuális sor előtti 11. sortól az utána következő 11. sorig (összesen 23 sor) történik a kiírás. Ha az aktuális sor sorszáma 12-nél kisebb, a file első 23 sora jelenik meg.

Csak az első paramétert megadva a kijelölt sortól kezdődően írathatunk 23 sort a képernyőre.

Csak a második paramétert megadva (előtte vesszővel!) az aktuális sor előtti 11. sortól kezdődik a kiírás, és a megadott sorszámig tart. Ha azonban a megadott sor 11-nél több sorral van az aktuális sor mögött, csak 23 sor jelenik meg – ugyanúgy, mintha nem adtunk volna meg paramétert az L parancs előtt.

A megadott intervallum 23 sornál hosszabb is lehet, ám ekkor a kiírás eleje elfut a képernyőn, és csak az utolsónak kiírt 23 sor marad látható.

Az L parancs az aktuális sor kijelölését nem változtatja meg, ellentétben a hasonló működésű P paranccsal.

Példák

L

-11,+11L

Mindkét paranccsal az aktuális sor 11 soros környezete jeleníthető meg a képernyőn.

17L

A 17–39. sorokat íratjuk ki.

,30L

Az aktuális sor előtt 11 sorral kezdve listázunk.

Ha az aktuális a 23. sor, akkor a 12–30. sorok látszanak.

Ha az aktuális a 14. sor, akkor a 3–25. sorok látszanak.
Ha az aktuális a 6. sor, akkor az 1–23. sorok látszanak.

10,25L

A 10–25. sorok jelennek meg (16 sor).

8,8L

Csak a 8. sort íratjuk ki.

Feladata

A szövegfile megadott sorait másik helyre teszi a file-on belül.

Paraméterezése

[*sor*], [*sor*], *sor* M

Használata

Az M parancs működése hasonlít a C parancséra, azzal a különbséggel, hogy az átmásolt sorok eredeti helyükről törlődnek. Az átmásolandó sorokat a parancs első két paramétereként megadható kezdő és záró sorszám határozza meg. Elhagyásuk esetén az alapértelmezés az aktuális sor.

A sorokat a harmadik, kötelező paraméterrel kijelölt sor elé helyezi el a parancs. A művelet után az első áthelyezett sor lesz aktuális.

A file sorait a parancs befejeztével az EDIT újra sorszámozza.

Példák

8,10,32M

A parancs a 8–10. sorokat a 32. sor elé másolja, és eredeti helyéről törli.

Az átsorszámozás az eredeti file 8–31. sorait érinti.

,,25M

Az aktuális sort a 25. sor elé helyezzük.

-2,+4,12M

Az aktuális sor előtti két sort, az aktuális sort és az őt követő négy sort mozgatjuk a 12. sor elé.

P

Feladata

A szövegfile-ból a megadott sorokat vagy egy lapot (23 sort) ír ki.

Paraméterezése

[*sor*] [,*sor*] **P**

Használata

A parancs működése hasonlít az L parancséhoz, de a P parancs megváltoztatja az aktuális sor kijelölését. A parancs végrehajtása után az utoljára kiírt sor lesz az aktuális.

Az első paraméter elhagyása esetén az alapértelmezés az aktuális sor mögött álló első sor. A második paraméter alapértelmezése az első sorszám plusz 22.

Paraméterek nélkül, ismételten használva tehát laponként (23 soronként) nézhetjük végig a szövegfile-t.

Példa

```
100P
:
P
:
P
```

A 100. sortól kezdődően laponként kiírhatjuk a file-t.

Feladata

Befejeződik az EDIT program működése, a lemezen levő file nem változik meg.

Paraméterezése

Q

Használata

A Q parancs kiadásával úgy fejezhetjük be a szövegfile javítását, hogy a központi tárból nem írjuk vissza a file-t a lemezre, és nem hozunk létre .BAK file-t. Erre két okunk lehet:

- mégsem kívánjuk módosítani a file-t, a beírt (és eddig csak a központi tárban végrehajtott) módosításokat nem akarjuk megőrizni,
- a file-t nem módosítottuk, csak megnéztük részeit, ezért mellőzni szeretnénk a visszaírással együtt járó .BAK file létrehozását.

A biztonság kedvéért az EDIT megkérdezi:

EDIT vége (i/n)?_

AZ I (vagy i) beütésével kilépünk az EDIT programból, és ismét PROPOS–16 parancsokat adhatunk meg. N, n vagy bármely más karakter beütésével folytathatjuk a file javítását.

R

Feladata

Egy adott szöveg (karaktersorozat) minden előfordulását kicseréli egy másik karaktersorozatra a file kijelölt sorai között.

Paraméterezése

[*sor*] [,*sor*] [?] **R** [*szöveg*] [⟨**F6**⟩ *szöveg*]

Használata

A parancs a megadott sorszámok között keresi az R után közvetlenül (betűköz nélkül!) kezdődő, és az [F6] (vagy [CTRL] [Z], leütésekor a képernyőn [^Z] látszik) karakter előtt végződő karaktersorozat előfordulásait. Ezeket az F6 és a sor végét jelző [ENTER] közé eső karaktersorozattal helyettesíti. A megváltozott sorokról a képernyőn listát kapunk.

Az első sorszám elhagyása esetén a keresés az aktuális sor utáni első sorban kezdődik. A második sorszám alapértelmezése a file utolsó sora.

A ? paraméter megadásával minden módosított sor kiírása után megjelenik az

O.K.?_

kérdés. A módosítás csak az [I] (igen) vagy az [ENTER] lenyomása esetén valósul meg. Bármely más karakterre az adott sor változatlan marad, de a keresés tovább folytatódik a megadott szöveg előfordulása után.

Az első szöveg elhagyása esetén ismét az R parancsot megelőzően utoljára kiadott R vagy S parancsban használt szöveget keresi a parancs.

A második szöveg elhagyása esetén az első szöveget az [ENTER] billentyű határolja, és előfordulásai törlődnek (üres karaktersorozattal helyettesítődnek).

Mindkét szöveg hiánya az utolsó R parancsban használt szövegek újbóli felhasználását eredményezi.

A szövegekben a kis- és nagybetűk nem egyenértékűek!

Az utolsóként módosított sor lesz az aktuális sor.

Példák

1,120RNagy Peter ^ZKiss Pal

Egy szövegfile első 120 sorában a Nagy Peter nevet mindenhol Kiss Palra cseréljük ki.

Rpropos ^ZPROPOS

Az aktuális sortól a file végéig végezzük el a cserét, kisbetűs írásról nagybetűsre.

1R 00 ^Z 0

A szövegfile-ban számok is vannak, ezekben az esetleges vezető nullákból csak egyet akarunk meghagyni. (A két szöveg betűközzel kezdődik!) Minden 00 egy 0-ra cserélődik, mert az első csere után tovább pásztázza a sort a program. Ha pl. a 38. sor tartalma:

38:...00007...

a fenti parancs kétszeri végrehajtásakor készült listában az alábbi sorokat találjuk:

⋮
38:...0007...
38:...007...
1R
38:...07...
⋮

A cserét a file minden érintett sorában elvégezzük.

20,150?Rfile

A 20–150. sorokból töröljük a „file" szót – de minden törlés előtt szeretnénk meggyőződni a törlés hatásáról, és egyenként engedélyezni azokat.

S

Feladata

Megkeresi egy szöveg (karaktersorozat) előfordulásait a file kijelölt sorai között.

Paraméterezése

[*sor*] [,*sor*] [?] S [*szöveg*]

Használata

A parancs a megadott sorszámok között keresi az S után közvetlenül (betűköz nélkül) kezdődő, és az [ENTER] előtt végződő karaktersorozatot tartalmazó sorokat.

Az első sorszám elhagyása esetén a keresés az aktuális sor utáni első sorban kezdődik. A második sorszám alapértelmezése a file utolsó sora.

A ? paraméter megadásával minden talált sor kiírása után megjelenik az

O.K.?_

kérdés. Az [I] (igen) vagy az [ENTER] lenyomása esetén az utoljára kiírt sor lesz az aktuális sor, és a keresés nem folytatódik (ezt a sort kerestük). Bármely más karakter hatására a keresés tovább folytatódik a szöveg következő előfordulásáig, illetve a file végéig.

A ? paraméter nélkül a keresés csak az első talált sor kiírásáig tart, és az lesz az aktuális sor.

Ha szöveget nem adunk meg a parancsban, ismét az S parancsot megelőző utolsó R vagy S parancsban használt karaktersorozatot keressük.

A szövegben a kis- és nagybetűk nem egyenértékűek!

Példák

1,100 S EDIT

A file első 100 sorában az EDIT szó (előtte és utána betűköz) első előfordulását keressük. Az illető sor lesz az aktuális sor.

S

Az aktuális sor utáni sortól a file végéig, az utoljára megadott szöveg (EDIT) első előfordulását keressük. Ezt a parancsot ismételgetve a file végéig valamennyi ilyen szót megkereshetjük.

1 ? S program

A „program" szó egy bizonyos előfordulását keressük. A teljes file-on végigmegyünk, minden sor után megállva, míg a keresett sorhoz nem érünk. Az lesz utána az aktuális sor.

Feladata

Egy megadott szövegfile-t bemásol a javítás alatt álló file egy sora elé.

Paraméterezése

[*sor*] **T** [*m*:]*file-név* [.*kit*]

Használata

A kijelölt (vagy a feltételezett) meghajtó aktuális könyvtárából a parancssorban megadott azonosítójú szövegfile-t beolvassuk, és tartalmát az adott sorszámú sor elé beillesztjük. Könyvtár (keresési út) nem adható meg.

A sorszám elhagyása esetén alapértelmezése az aktuális sor.

A beillesztés után a file sorait újrasorszámozza az EDIT program.

Példa

1T FEJLEC

A feltételezett meghajtó aktuális könyvtárában tárolt FEJLEC nevű file tartalmát a javítandó file első sora elé másoljuk.

W

Feladata

A javított file elejéről adott számú sort visszaír a lemezre.

Paraméterezése

[*n*] W

Használata

A W parancs használatára csak akkor van szükség, ha a file mérete nagyobb a rendelkezésre álló tárterület 3/4 részénél. Ha a file elejének javítását befejeztük, a W paranccsal az első *n* sort visszaírhatjuk a lemezre. Az így felszabaduló területre az A paranccsal tölthetünk be további sorokat a file folytatásából. Az eljárást addig ismételgethetjük, míg a file végére nem érünk.

Ha *n* értékét nem adjuk meg, a kiírás addig tart, míg a pufferterület 3/4 része szabad nem lesz. Ha 1/4-nél alacsonyabb e terület kitöltöttsége, a W parancs hatástalan. (A szövegfile végét az E parancs írja ki!)

A központi tárban maradó sorok újra sorszámot kapnak, ismét 1-től kezdődően!

Példák

100W

A file első 100 sorát visszaírjuk a lemezre.

W

Annyi sort írunk vissza a lemezre, hogy csak a pufferterület 1/4 részében maradjanak sorok.

Üzenetek

6. Üzenetek és hibajelzések

Ebben a fejezetben a PROPOS–16 operációs rendszer üzeneteit és hibajelzéseit ismertetjük, betűrendben. Az üzenetekkel együtt adjuk meg a kiíróprogram nevét, az üzenet kiírásának okát, és az esetleges teendőket is. Külön csoportban előreveszszük a perifériahibákat.

Perifériahibák jelzése

A bemeneti és a kimeneti műveletek során a perifériák által adott hardware hibajelzéseket az operációs rendszer az alábbi egységes formában szövegezi meg:

⟨*hiba típusa*⟩⟨*periféria*⟩⟨*művelet*⟩

ahol a periféria neve az A. függelékben ismertetett szabványos nevek egyike, vagy egy lemezmeghajtó azonosítója; a művelet írás vagy olvasás lehet; a hiba típusa pedig a következők egyike:

Adathiba

Egy adatrekord írása vagy olvasása nem sikerült. Ez a hibajelzés általában a fizikailag sérült lemezre utal.

A papír kifogyott

A nyomtató által jelzett hiba új papír befűzésével szüntethető meg.

Hibás perifériakezelő program

A rendszerbe a PROPOS–16 operációs rendszerhez nem illeszkedő perifériakezelő programot töltöttünk be.

Íráshiba

Perifériára írás nem sikerült.

Írásvédelem megsértése

Írásvédett lemezre próbáltunk írni. Felírás nem történik!

Lemezhiba

A fentiektől eltérő minden más lemezhiba jelzése.

Nem PROPOS lemez

A file-allokációs tábla felismerhetetlen információt tartalmaz.

Nem üzemkész

A jelzett berendezés nem fogad, illetve nem küld adatokat.

Nincs olvasható szektor

A megcímzett lemezterületről nem lehet adatot olvasni. Lehet fizikai hiba vagy hibás címzés is.

Olvasási hiba

A kijelölt perifériáról vagy lemezről nem sikerült olvasni. Ellenőrizzük a lemez behelyezését!

Sávkeresési hiba

A meghajtó nem tud a keresett sávra pozicionálni. Ellenőrizzük a lemez behelyezését, vagy próbálkozzunk másik meghajtóval, esetleg futtassuk le a STAT programot! A fenti üzenetek után mindig megjelenik a folytatásra vonatkozó döntést kérő üzenet:

Befejeződjön, Újra, Tovább?

amire a B, U vagy T betűk beírásával válaszolhatunk. Hatásuk a következő:

B : a hardware hibajelzést észlelő program működése *b*efejeződik,
U : a program *ú*jra próbálkozik a művelet végrehajtásával,
T : a hibajelzést figyelmen kívül hagyva *t*ovábbfut a program.

Az üzenetek betűrendes ismertetése

A céllemez olvasása + összehasonlítás
Sáv: ...

DISKCOPY: az újonnan létrehozott lemezt olvassa, és összehasonlítja az eredetivel. A kiírt szám az éppen vizsgált sáv sorszáma.

Adja meg a lemez cimkéjét (max. 11 karakter)

FORMAT: az új lemezre írandó cimkét kéri, ha a parancsban /L opciót adtunk meg. A legfeljebb 11 karakteres cimkét egybeírva (elválasztó betűköz vagy pont nélkül) kell bebillentyűzni.

A file nem másolható önmagára

COPY: a kimenő file-t másik könyvtárba, illetve másik lemezre kell írni (ekkor lehet azonos a neve és kiterjesztése), vagy új nevet, illetve kiterjesztést kell megadnunk.

A file tartalma elveszett a másolás előtt

COPY: file-egyesítésnél az összevonandó file-ok egyikének (de nem az elsőnek) az azonosítója megegyezik az eredményfile-éval, így tartalma elveszett, mire az olvasására sor került.

A formálás ... oldalas, ... sávos, ... szektoros lemezen történik

FORMAT: a parancsnak megadott opciók, illetve az alapértelmezés szerint, a kiírt jellemzőkkel végzi a formálást.

A forrás- és a célmeghajtó ugyanaz!

BACKUP, RESTORE: a bemenő és kimenő lemeznek különböző meghajtón kell lennie.

A forráslemez olvasása
Sáv: ...

DISKCOPY: a kiírt sorszámú sáv olvasásánál tart.

A gyökérkönyvtárban kevés a hely

STAT: az elveszett szektorok összefűzéséből keletkező file-ok katalógusbejegyzése nem fér már el a gyökérkönyvtárban, ezért a program működése befejeződik.

A két lemez különböző típusú, nem összehasonlítható

DISKCOPY: a /CO opció megadásával különböző sávszámú lemezeket akartunk összehasonlítani.

A lemez alkalmatlan rendszerfile-ok tárolására

FORMAT: a lemez első sávjai közül valamelyik fizikailag sérült, ezért rendszerlemezként nem használható.

A lemezek összehasonlítása

DISKCOPY: a lemez másolása befejeződött, a /C opció hatására az összehasonlítás kezdődik.

A lemez hibás, nem formálható

FORMAT: a lemez nem formálható a kívánt módon.

A lemez írásvédett, cserélje ki!

FORMAT: cseréljük ki a lemezt, vagy távolítsuk el az írásvédelmet biztosító öntapadó cimkét. A megfelelő lemez behelyezése után egy billentyű leütésével indíthatjuk a formálást.

A lemez összterülete (byte): ...
A lemez szabad területe (byte): ...
A rendszer által lefoglalt terület (byte): ...

FORMAT: a lemezformálás után kapott táblázat a létrehozott lemez kapacitásadatait tartalmazza.

Alkönyvtár nem hozható létre

MKDIR: a megadott alkönyvtár már létezik, vagy a hozzá vezető út leírása hibás, vagy a gyökérkönyvtár betelt, és ezért nem hozható létre.

A másodpéldány felírása
Sáv: ...

DISKCOPY: a kiírt sorszámú sáv felírásánál tart.

A másodpéldány formálása
Sáv: ...

DISKCOPY: a kiírt sorszámú sáv formálását végzi az új lemezen.

A másodpéldány írásvédett, cserélje ki!

DISKCOPY: az írásvédett lemez helyett másikat kell betenni, vagy az írásvédő cimkét el kell távolítani. Egy billentyű lenyomásával folytathatjuk a másolást.

A másolandó lemez olvasása
Sáv: ...

DISKCOPY: a kiírt sorszámú sáv olvasásánál tart.

A sor hosszú!

EDIT: a szöveghelyettesítő (R) parancs hatására 253 karakternél hosszabb sor keletkezne, ezért a módosítást nem hajtja végre a program.

Az első allokációs egysége hibás, a(z) ... file-t törölni kell

STAT: a jelzett file első allokációs egysége érvénytelen sorszámú, ezért a vizsgálata nem lehetséges. /F opció esetén 0 byte hosszúságú file keletkezik.

A(z) ... file csak olvasható, vagy az utolsó elmentés óta megváltozott.
Visszatölti (I/N)?

RESTORE: a jelzett file a kimenő lemezen írásvédett, vagy az attributumbyte változás bitje 1. Visszatöltését csak I válasz esetén végzi el a program.

A(z) ... lemezen a(z) ... file-t nem találta.
A visszatöltött file valószínűleg használhatatlan.

RESTORE: egy mentési lemez végén egy file-nak csak az első része volt, a következő lemezen viszont nem található ennek a folytatása. Megsérült a lemez, vagy különböző mentések lemezei összekeveredtek.

A(z) ... meghajtóban levő lemezen nincs elég hely!
Tegyen be egy másik lemezt, és üssön be egy billentyűt!

BACKUP: a mentéshez folytató lemezre van szükség. Csak előre megformált lemezek használhatók!

A(z) ... meghajtó nem formálható

FORMAT: a fix lemez megformálása nem megengedett.

A(z) ... meghajtó nem másolható

DISKCOPY: a fix lemez tartalma ezzel a programmal nem másolható át.

A(z) ... meghajtó nem összehasonlítható

DISKCOPY: /CO opció megadásával a fix lemez összehasonlítását kértük egy másik lemezzel – ez nem végezhető el.

Azonos meghajtó kiválasztása!

IMP: a bemeneti és kimeneti meghajtó nem lehet ugyanaz.

Az összehasonlítás ... oldalas, ... sávos, ... szektoros lemezek között történik

DISKCOPY: az összehasonlított lemezek (/CO opcióval) a felsorolt paraméterekkel rendelkeznek.

.BAK file nem módosítható

EDIT: .BAK kiterjesztésű file módosításához előzőleg meg kell változtatnunk a file-név kiterjesztését.

BATCH file hiányzik

PROPOS: a batch file feldolgozása közben a file elérhetetlenné vált. Vagy kicseréltük a lemezt, vagy egy batch lépés törölte a file-t, ezért a batch-futás megszakad.

BATCH vége (I/N)?

PROPOS: [CTRL] [BREAK] ([CTRL] [C]) billentyűt nyomtunk le batch-futás közben. Az [I] (igen) lenyomására a file feldolgozása befejeződik; N (nem) hatására csak az éppen végrehajtott program futása szakad meg, és a batch file következő sorának végrehajtása következik.

Befejeződjön, Újra, Tovább?

PROPOS: hardware-hibát jelez a gép. [B] lenyomására a program futása befejeződik, [U] hatására újra próbálkozik, [T] hatására pedig a hiba ellenére továbbmegy.

. bejegyzés hibás
.. bejegyzés hibás

STAT: az alkönyvtár elején levő bejegyzés megsérült. /F opció megadása esetén a program megkísérli a javítást.

... berendezésről nem lehet binárisan olvasni

COPY: a /B paramétert olyan file-névhez rendeltük, amely karakteres perifériát azonosít, és bináris olvasásra nem használható.

Biztos (I/N)?

DEL: egy teljes könyvtár (*.*) törlésére adtunk parancsot. Csak az I (igen) megerősítés után hajtja végre a program.

Célfile neve túl hosszú

IMP: legfeljebb 8 karakteres file-név és 3 karakteres kiterjesztés adható meg.

COM*n*: *bbbb, p, d, s,* P

MODE: az *n*-edik aszinkron vonal jellemzői:

bbbb – adatátviteli sebesség (baud)
p – paritásbit: e=páros, o=páratlan, n=nincs
a – adatbitek száma (7 vagy 8)
s – stopbitek száma (1 vagy 2)
P – a vonalon nyomtatót működtetünk

Csak olvasható file

EDIT: a file csak olvasható attribútumú, nem módosítható.

Duplikált file-név vagy nem létező file

RENAME: a megadott új file-név már előfordul a könyvtárban, vagy az átnevezendő file nem található, ezért a parancs nem hajtható végre.

EDIT vége (I/N)?

EDIT: a Q parancs hatására a program a javítások elmentése nélkül fejezné be működését, ezért megerősítést kér (I).

Eltérés

DISKCOPY: a jelzett sorszámú sáv tartalma nem egyforma a két lemezen.

„Environment" terület megtelt

SET: a rendelkezésre álló terület megtelt a *név=érték* alakú paraméterekkel. Újabb paraméter csak egy meglévő törlése után vihető be.

Érvénytelen a címke, adja meg újra:

FORMAT: a megadott címke hibás, adjunk meg újat, vagy az [ENTER] leütésével jelezzük, hogy címke nélküli lemez létrehozását kérjük.

Érvénytelen berendezés

CTTY: a megadott perifériaazonosító ismeretlen, vagy nem használható konzolként.

Érvénytelen meghajtó

PROPOS parancsok: a parancsnév előtt vagy valamelyik paraméterében nem létező meghajtóra hivatkoztunk.

Érvénytelen meghajtó: ...
Új meghajtó azonosító: ...

DISKCOPY, FORMAT: a megadott azonosító nem létező meghajtóra utal, másikat kell megadnunk.

Érvénytelen opció

PROPOS parancsok: a megadott opcionális paramétert a program nem tudja értelmezni.

Érvénytelen paraméter

PROPOS parancsok: a megadott paraméter nem értelmezhető.

Érvénytelen út, nem könyvtár, vagy a könyvtár nem üres

RMDIR: a kijelölt könyvtár a felsorolt okok valamelyike miatt nem szüntethető meg. Csak üres alkönyvtár távolítható el a lemezről!

Érvénytelen út vagy file-név

COPY: nem létező vagy nem a kijelölt könyvtárban található file másolását kértük.

EXEC hiba

PROPOS: a parancsot végrehajtó program betöltése közben történt hiba. Lehet, hogy a CONFIG.SYS file FILES paraméterével túl kevés file-t adtunk meg.

EXE file hiba

PROPOS: egy EXE típusú file betöltésekor a relokációs adatokat hibásnak találta a rendszer. Valószínűleg megsérült a file.

.EXE file-típus átalakítva .COM file-típusra

CONVEXE: sikeresen befejezte működését.

Ezen ág feldolgozása tovább nem folytatható

STAT: a könyvtárak fastruktúrájában az adott ágakon nem tudja folytatni a file-ok ellenőrzését.

File-allokációs tábla hibája: ...

STAT: a jelzett file elemzése során hibás bejegyzést talált a program a file-allokációs táblában, ezért a file-t az utolsó érvényes allokációs egységnél csonkolja.

... file hibás vagy hiányzik

PROBIO: nem tudja betölteni a COMMAND.COM parancsértelmező programot, vagy a CONFIG.SYS file-ban felsorolt perifériakezelő programok valamelyikét.

File-kreálási hiba

PROPOS parancsok: új file létrehozása nem sikerült, mert vagy írásvédett file-t kellett volna felülírni, vagy megtelt a könyvtár, illetve a lemez.

... file másolása kész

COPY: a kiírt számú file másolása sikeresen befejeződött.

File-másolás kész

IMP: a file-átalakítás befejeződött.

File neve túl hosszú

CONVEXE: az átalakítandó file azonosítója nem felel meg a PROPOS–16 szabályoknak.

File-név megadása kötelező

EDIT: a javítandó file azonosítóját kötelező a parancssorban megadni.

File=spec:

PROPOS parancsok: a parancs paramétereként teljes file-hivatkozás adható meg. Ha a feltételezett meghajtón, és/vagy az aktuális könyvtárban jelölünk ki file-t, ezek azonosítóját nem kötelező kiírni.

Formai hiba a célfile nevében

IMP: a kimenő file azonosítójának megadása hibás vagy felesleges (pl. file-csoport átalakításkor nem adható meg az eredetitől eltérő file-azonosító).

FORMÁLÁS
Sáv ...

FORMAT: az új lemez formálása a jelzett sorszámú sávnál tart.

Formálási hiba

DISKCOPY: a másolat számára kijelölt lemezen hibás szektor van, ezért a másolás nem hajtható végre.

Formálási hiba: ...

FORMAT: a jelzett sáv fizikai hibás a lemezen. A formálás folytatódik, a program a file-allokációs táblába bejegyzi a hibát, a lemez hasznos kapacitása ennek megfelelően kevesebb lesz.

FOR nem ágyazható egymásba

PROPOS: batch file-ban a FOR parancsok nem tartalmazhatnak másik FOR parancsot.

F paraméter hiányzik
A talált hibákat nem javítom

STAT: a parancsnak nem adtunk meg /F paramétert, ezért csak jelzi a lemezen talált hibákat, de nem javítja azokat.

Helyezze a céllemezt az ... meghajtóba, rögzítse, és üssön le egy billentyűt!

DISKCOPY: az üres lemez behelyezése után tetszőleges billentyű leütésével indíthatjuk a másolást. A lemez korábbi tartalma elvész!

Helyezze a formálandó lemezt az ... meghajtóba, rögzítse, és üssön le egy billentyűt!

FORMAT: tetszőleges billentyű leütésével indíthatjuk a lemezformálást. A lemez korábbi tartalma elvész!

Helyezze a forráslemezt az ... meghajtóba, rögzítse, és üssön le egy billentyűt!

DISKCOPY: a másolandó lemez behelyezését tetszőleges billentyű lenyomásával jelezzük.

Helyezze az ... lemezt az ... meghajtóba, és üssön le egy billentyűt!

BACKUP, RESTORE: a megfelelő sorszámú mentési lemezt kell betenni a jelzett meghajtóba. Tévesztés esetén mindaddig ismétlődik ez az üzenet, amíg a kért lemezt nem tettük be.

Helyezzen rendszerlemezt az ... meghajtóba, rögzítse, és üssön le egy billentyűt!

FORMAT: rendszerlemezt akarunk előállítani (/S opció), ezért a parancs kér egy olyan lemezt a jelzett meghajtóba, amelyről az operációs rendszer moduljait (PROBIO.COM, PRODOS.COM, COMMAND.COM) leolvashatja.

Hibás alkönyvtár

FSYS: valószínűleg megsérült a lemez, mert egy alkönyvtárat nem lehet leolvasni. Futtassuk le a STAT programot!

Hibás alkönyvtár: ...

STAT: hibás információ van a lemezen egy alkönyvtárra vonatkozóan. A program megkísérli a hiba javítását. Részletesebb információt a /V opció megadásával kaphatunk.

Hibás allokációs egység

STAT: a lemez file-allokációs táblájában hibás bejegyzés található. Az érintett file-t csonkolja a program az utolsó helyes allokációs egység mögött.

Hibás az átviteli sebesség

MODE: átviteli sebességként 110, 150, 300, 600, 1200, 2400, 4800 vagy 9600 baud adható meg, a szám teljes kiírásával vagy első két számjegyével.

Hibás BATCH parancs

PROPOS: a batch file feldolgozása során nem értelmezhető parancsot talált az operációs rendszer.

Hibás dátum

SET: a dátum megadásának kötelező formája: év (négy számjeggyel vagy utolsó két számjegyével), hónap (egy vagy két számjeggyel) és nap (egy vagy két számjeggyel) – törtvonallal vagy kötőjellel elválasztva.

Hibás file-allokációs tábla

PROPOS: a lemez valószínűleg megsérült, a még használható file-okat mentsük el, majd formáljuk újra a lemezt.

Hibás file-név-megadás

BACKUP, RESTORE: a megadott file-vagy file-csoport-azonosító hibás.

Hibás időpont

SET: a pontos időt az óra, perc és másodperc egy- vagy kétjegyű számértékével adhatjuk meg, kettősponttal elválasztva az értékeket.

Hibás meghajtó/file-név

EDIT: a megadott meghajtóazonosító hibás, vagy a javítandó file nem található a kijelölt könyvtárban.

Hibás paraméter

ASSIGN: a megadott meghajtóazonosítók valamelyike érvénytelen.
EDIT: a programnak adott opcionális paraméter nem értelmezhető.

Hibás parancs vagy file-név

PROPOS: az adott parancsot végrehajtó COM, EXE vagy BAT kiterjesztésű file nem található sem az aktuális könyvtárban, sem a PATH paranccsal kijelölt könyvtárakban.

Hibás periférianév

MODE: nem létező periféria paramétereit akartuk beállítani.

Hibás COMMAND.COM
Tegye be a COMMAND.COM lemezt!

PROPOS: a COMMAND.COM parancsértelmező tranziens részének újratöltésekor a rendszerlemezen nem megfelelő programot talált a rendszer. A lemez kicserélése után a munkát egy billentyű leütésével folytathatjuk.

Hibás sor

EDIT: a programnak adott parancssor nem értelmezhető.

Indexfile kész

MENUSYS: az új menü/help file-hoz elkészült a megfelelő indexfile.

Input file vége

EDIT: a javítandó file-t a végéig betöltötte a központi tárba.

Íráshiba

CONVEXE: a kimeneti lemez sérült vagy megtelt.
DIKSCOPY: a másolat írásakor a meghajtó hibát jelzett, a másolat valószínűleg nem hibátlan.

Ismeretlen könyvtár

PROPOS parancsok: a megadott keresési útban valamelyik könyvtár nem létezik vagy hibás.

Ismeretlen meghajtó a keresési útban

PROPOS parancsok: a PATH parancsban kijelölt alternatív keresési utak egyike hibás meghajtóazonosítót tartalmaz. Ez csak a PATH-ban megadott utak első felhasználásakor derül ki!

Ismételt próbálkozás nyomtató time-outnál

MODE: a nyomtató kezelésénél előforduló time-out-hiba esetén a rendszer ismét próbálkozik.

Kettős allokálású egység: ... file ... cluster

STAT: a lemezen levő két file lánca azonos allokációs egységet tartalmaz. Ez az üzenet mindig páros számban jelentkezik, megadva mindkét file közös pontját.

A STAT program nem javítja a hibát. A javasolt javítási mód:

- másoljuk le mindkét file-t (COPY)
- töröljük az eredeti példányokat (DEL)
- szükség szerint javítsuk a másolatokat.

... kiírás leállítva

PRINT: a jelzett file kiíratását a gépkezelő megszakította a parancs /C opciójával.

... könyvtárból képezhető file (I/N)?

STAT: a kiírt azonosítójú könyvtár olyan sérült, hogy könyvtárként nem helyreállítható. I (igen) válaszra a program egyetlen file-ba írja a könyvtár tartalmát, N (nem) válaszra nem módosítja.

Könyvtárkapcsolatok listája
Lemezazonosító cimke: ...

FSYS: a program kimenő listájának fejléce.

Könyvtárkatalógus megtelt

EDIT: a gyökérkönyvtárban több bejegyzés számára nincs hely, ezért a javított file nem írható vissza, elvész.

... láncban ... elveszett allokációs egység.
Átalakíthatók file-okká (I/N)?

STAT: az első adat az összefüggő elveszett területek (láncok) számát, a második az ezekben összesen található allokációs egységek számát adja. A program csak akkor fűzi össze ezeket file-okká, ha a parancsnak /F paramétert is megadtunk, és a kérdésre I (igen) választ adunk.

Látja a bal szélső 0-ást (I/N)?

MODE: a képernyősorok interaktív eltolása (2. parancsforma, T opció) során két pozicióval jobbra tolhatjuk a képet, ha az első pozíció nem látszik. I (igen) válasz esetén a képernyő nem változik.

Látja a jobb szélső 9-est (I/N)?

MODE: a képernyősorok interaktív eltolása során két pozícióval balra tolhatjuk el a képet, ha az utolsó pozíció nem látszik.

Lemezhiba a FAT*n* írásakor

STAT: file-allokációs tábla *n*. (1. vagy 2.) példányának felírása nem sikerült. Ha mindkét példány hibás, a lemez használhatatlan.

Lemezhiba miatt a végrehajtás megszakadt!

BACKUP, RESTORE: fizikai lemezhiba miatt a mentés, illetve visszatöltés nem hajtható végre.

Lemez megtelt, felírás hiányos

EDIT: a lemez megtelt, ezért a javított file nem írható vissza a lemezre teljes egészében. A file vége elvész. Az eredeti file .BAK kiterjesztéssel megmaradt, a javítást ebből kiindulva lehet megismételni.

LPT*n* : átirányítva 〉 COM*n*

MODE: az LPT*n* jelű párhuzamos nyomtatót a COM*n* soros vonalra irányítottuk át.

LPT*n* : nem módosított

MODE: az LPT*n* párhuzamos nyomtató ismét eredeti hozzárendelésben.

LPT*n*: 80 kar/sor
LPT*n* : 132 kar/sor

MODE: az *n*-edik nyomtató sorhosszúságát a kiírt értékre módosítottuk.

... meghajtóban levő lemez: ....

DIR: a parancs kimenő listájának fejlécében a lemezazonosító címkéjét írja ki, vagy azt, hogy címke nélküli.

Megnyitáshiba

CONVEXE: a kimenő .COM file-t nem sikerült létrehozni (a gyökérkönyvtár vagy a lemez megtelt).

Megtelt a célmeghajtó!

RESTORE: a visszatöltést nem sikerült befejezni, mert a célmeghajtón nincs több szabad hely.

Memória túl kevés

EDIT: a beszúrások következtében a rendelkezésre álló központitár-terület megtelt, a W paranccsal kell helyet csinálni a további növeléshez.

Minden file kiírása leállítva

PRINT: a gépkezelő /T opcióval leállította a nyomtatást, a várakozási sorban nincs több nyomtatandó file.

Minden megadott file összefüggő

STAT: a vizsgált file-csoport minden tagja összefüggő területet foglal el a lemezen.

MODE rezidens része betöltve

MODE: a parancsot végrehajtó program egy része a következő rendszertöltésig a központi tárban rezidens marad.

Nem javítható hiba

STAT: a megelőző üzenet jelzi, hogy miért nem tudja a program folytatni működését.

Nem konvertálható

CONVEXE: a bemenő .EXE file formátuma nem felel meg a .COM file-okra vonatkozó előírásoknak, ezért az átalakítás nem végezhető el.

Nem lehet megnyitni a ... file-t

PROPOS parancsok: a paraméterként megadott file nincs a könyvtárban, rejtett, vagy más ok miatt nem nyitható meg.

Nem létező BATCH cimke

PROPOS: egy GOTO parancs olyan címkére hivatkozik, amely nem szerepel a batch file-ban.

Nem létező file

PROPOS parancsok: a paraméterként megadott file-azonosító nem található a kijelölt (vagy az aktuális) könyvtárban.

Nem létező PROPOS–8 file(-ok)
Nem létező PROPOS–16 file(-ok)

IMP: a paraméterként megadott file-azonosítók nem találhatók a lemezen.

Nem összefüggő blokkok száma: ...

STAT: a vizsgált file a kiírt számú, egyenként összefüggő, de a lemezen nem egymás mellett elhelyezett részből áll.

Nincs a lemezen alkönyvtár

FSYS: a vizsgált lemezen csak a gyökérkönyvtár található.

Nincs elegendő lemezterület

PROPOS parancsok: a kimenő file nem fér el a lemezen. Ha nincs törölhető file, a STAT paranccsal próbáljunk helyet felszabadítani (elveszett területekből).

Nincs elegendő memória

DISKCOPY, FORMAT: a központi tárban kevés a szabad hely, a program nem működik. Próbálkozzunk a CONFIG.SYS file-ban megadott pufferek számának csökkentésével, az operációs rendszer újraindításával és a rezidensen maradó programok betöltésének elhagyásával.

Nincs elég hely file-egyesítéshez

EDIT: a T parancs végrehajtása során a kijelölt file-nak csak egy részét sikerült betölteni.

Nincs eltérés

DISKCOPY: az összehasonlított két lemez tartalma megegyezik.

Nincs file a PROPOS–8 lemezen!

IMP: a kijelölt bemenő lemezen nincs felismerhető PROPOS–8 formátumú file.

Nincs file-név

CONVEXE: a parancsnak kötelező legalább bemenő file-nevet megadni.

Nincs ilyen file

CHMOD: a módosítani kívánt file nem található a kijelölt (vagy az aktuális) könyvtárban.

Nincs kijelölt nyomtató

PRINT: a kijelölt nyomtató azonosítója nem létező berendezésre utal.

Nincs más

EDIT: az S (szövegkereső) parancs nem talált több olyan sort, amelyben a keresett karaktersorozat előfordul.

Nincs menteni való file

BACKUP: a megadott file-csoport-hivatkozás egyetlen file-ra sem utal.

Nincs sorrendben

SORT: az ellenőrzött file (/C opció) nincs a megadott szempontok szerint rendezve.

Nincs szabad PROPOS–8 directory hely

IMP: a PROPOS–8 lemezre több file nem írható, mert a lemezkatalógus megtelt.

Nincs több hely a PROPOS–16 lemezen

IMP: a PROPOS–16 lemezre több file nem írható, mert megtelt.

Nincs törölhető file

DEL: a törölni kívánt file vagy file-csoport nem található a kijelölt (vagy az aktuális) könyvtárban.

Nincs út

PATH: a paraméterek nélkül kiadott paranccsal az alternatív keresési utakat kérdeztük – ilyenek jelenleg nincsenek specifikálva.

Nincs szabad file-nyél

PROPOS: az operációs rendszer nem tud több file-t megnyitni. A CONFIG.SYS file-ban adjunk meg több file-t (FILES paraméter), majd indítsuk újra a rendszert.

Nincs visszatölteni való file

RESTORE: a bemenő lemezen nincs olyan azonosítójú mentési file, amelyet visszatölteni akarunk.

... nyomtatás folyamatban

PRINT: a kiírt azonosítójú file kinyomtatását végzi a program.

Nyomtatóazonosító [PRN]:

PRINT: a program első betöltésekor erre az üzenetre válaszolva jelölhetjük ki a nyomtatóperiféríát a rezidensen maradó program számára.
Alapértelmezés: PRN (LPT1).

Nyomtatóhiba

MODE: a nyomtató üzemmódját beállító parancs kiadásakor a periféria hibát jelzett. Az üzenet oka lehet kikapcsolt vagy off-line (nem üzemkész) nyomtató, papírhiány vagy hardware-hiba.

Nyomtatóhiba!
Ellenőrizze az on-line állapotot!

PRINT: a nyomtató hibát jelzett, oka lehet a kikapcsolt vagy off-line berendezés, papírhiány vagy hardware-hiba.

O.K.?

EDIT: a szövegkereső (S), illetve -helyettesítő parancsok használatakor ezzel az üzenettel kérdezi a program, hogy az előzőleg kiírt sort kerestük-e. A válasz I (igen) vagy N (nem) lehet.

Olvasási hiba

CONVEXE: az átalakítandó .EXE típusú file-t nem sikerült elolvasni.
DISKCOPY: a bemenő lemez jelzett sávjának olvasásakor a periféria hibát talált.

SET: a BREAK és a VERIFY paraméterek értéke csak ON vagy OFF lehet (kis- és nagybetűvel írva is megadható).

Osztástúlcsordulás

PROPOS: a 0. megszakítási szint által kiírt üzenet. Oka általában programhiba.

Output-sorszám megadása kötelező

EDIT: a másoló (M és C) parancsoknak meg kell adni az átmásolt sorok helyét.

Összeférhetetlen paraméterezés

DISKCOPY, FORMAT: a parancsnak megadott paraméterek ellentmondásosak, nem értelmezhetők.

Papír!

PRINT: nyomtatás közben kifogyott a papír. Új papír befűzése után a kiírás folytatódik.

Parancsláncoláshoz kevés a lemezterület

PROPOS: a parancsláncoláshoz szükséges átmeneti file-t az operációs rendszer nem tudja létrehozni.

PRINT rezidens része betöltve

PRINT: a program első hívása után jelzi, hogy ettől kezdve rezidens marad a legközelebbi rendszerbetöltésig.

PRINT sor megtelt

PRINT: a nyomtatásra várakozó file-ok sora megtelt, további file kiíratására később adhatunk csak parancsot. Egyidejűleg legfeljebb 10 file várakozhat nyomtatásra.

PRINT sor üres

PRINT: a paraméterek nélkül kiadott parancsra válaszolva a program közli, hogy jelenleg nincs nyomtatásra várakozó file.

PROPOS betöltése nem sikerült. PROPOS vége

PROPOS: a központi tár adminisztrálását nyilvántartó adatok elvesztek vagy megsérültek, illetve a COMMAND.COM modul nem található a feltételezés szerinti rendszerlemezen. Újra kell indítani az operációs rendszert!

PROPOS nem indítható

PROPOS: a parancsértelmező elindítása nem sikerült. Vagy nincs elég szabad hely a központi tárban, vagy a CONFIG.SYS file-ban levő FILES paraméterben túl kevés file-t adtunk meg.

PROPOS–8 directory
PROPOS–16 directory

IMP: a DIR paraméter megadásával kiíratott könyvtár tartalom fejléce.

PROPOS–8 directory hiba!

IMP: a kijelölt PROPOS–8 formátumú lemezen a könyvtár nem felismerhető vagy hibás.

PROPOS–8 directory íráshiba

IMP: a kijelölt PROPOS–8 formátumú lemezen a könyvtár megtelt vagy hibás, ezért új tétel nem írható bele.

PROPOS–8 file íráshiba
PROPOS–16 file íráshiba

IMP: a jelzett lemez írásvédett vagy hibás, ezért a kimenő file-ok nem írhatók fel rá.

PROPOS–8 file-másolás ...
PROPOS–16 file-másolás ...

IMP: a kiírt típusú bemenő file(-ok) átalakítása folyamatban van.

PROPOS–16 file-megnyitás hiba

IMP: nem létező file-ra hivatkoztunk a parancsban.

PROPOS–8 lemez az alapegységben

IMP: a parancs szerint a feltételezett meghajtóban kellene a PROPOS–8 típusú lemezt keresni, ez nem megengedett.

PROPOS–8 lemezhiba az ... meghajtóban

IMP: a PROPOS–8 formátumú lemezt kezelő egység hibát jelzett.

PROPOS–8 lemez tele van
PROPOS–16 lemez tele van

IMP: a jelzett kimenő lemez megtelt, több file nem írható rá.

PROPOS–8 olvasási hiba
PROPOS–16 olvasási hiba

IMP: a jelzett bemenő lemez olvasásakor az egység hibát jelzett.

Sorsűrűség átállítva

MODE: a nyomtató sorsűrűségének beállítására adtunk parancsot. Ha a végrehajtás során hibát észlelt a program, hibaüzenetet is kapunk.

Szektorméret túl nagy a file-ban: ...

PROBIO: az üzenetben jelzett perifériakezelő program az előzőleg specifikált perifériáknál nagyobb, a PROPOS–16 rendszerrel összeférhetetlen szektorméretet határoz meg.

Tárallokációs hiba

PROPOS parancsok: a program számára kevés a rendelkezésre álló központitár-terület. Csökkentsük a rendszerpufferek számát a CONFIG.SYS file-ban (BUFFERS paraméter), és töltsük újra az operációs rendszert.

Tegye be a BATCH lemezt az ... meghajtóba, és üssön le egy billentyűt, ha kész!

PROPOS: a feldolgozás alatt álló batch file-t tartalmazó lemezt kivettük a meghajtóból. A következő lépés elindításához vissza kell tennünk a jelzett meghajtóba!

– Tovább –

MORE: minden képernyőnyi (24 sor) kimenő lista után ezzel az üzenettel várakozik a program egy tetszőleges billentyű leütésére.

Törölhető (I/N)?

DEL: file-csoport törlésekor, a /P opció hatására egyenként engedélyezhetjük a file-ok törlését. I (igen) hatására a kiírt azonosítójú file törlődik, N (nem) hatására megmarad.

Túl nagy a bemeneti adatmennyiség

COMM, DIFF, SORT: a feldolgozandó file-nak a központi tárban el kell férnie, különben a program nem tudja feladatát elvégezni.

Túl nagy program, nem tölthető be

PROPOS: a külső parancsot végrehajtó program nem fér el a rendelkezésre álló központitár-területen. Csökkentsük a CONFIG.SYS file-ban megadott rendszerpufferek számát (BUFFERS paraméter), töltsük újra a rendszert, és ismét próbálkozzunk a program elindításával.

Túl sok opció

BACKUP, RESTORE: a parancsnak adott opcionális paramétereket a program nem tudja értelmezni.

Túl sok paramétert adott meg

DISKCOPY, FORMAT: a parancsnak adott opcionális paramétereket a program nem tudja értelmezni.

Új file

EDIT: a programnak adott file-hivatkozás nem létező file-ra utal, amelyet most hozhatunk létre a szövegszerkesztővel.

Út \könyvtárnév:

MKDIR, RMDIR: a programnak meg kell adni a – létrehozni, illetve megszüntetni kívánt – könyvtárhoz vezető keresési út leírását és a könyvtár nevét.

Üssön le egy billentyűt, ha folytathatjuk ...

DIR: a parancsot /P opcióval indítottuk, ezért minden képernyőnyi kiírás után egy tetszőleges billentyű leütéséig vár.

Valószínűleg nem PROPOS–16 lemez
Folytassam (I/N)?

STAT: a file-allokációs táblában levő azonosító byte nem feleltethető meg egyik PROPOS–16 formátumú lemeztípusnak sem. Vagy nem megformált lemez, vagy megsérült, vagy más operációs rendszer alatt készült. Célszerű először a /F opció nélkül futtatni a STAT programot ezen a lemezen.

... várakozik

PRINT: a jelzett azonosítójú file nyomtatásra várakozik.

Verziószámhiba

PROPOS parancsok: nem a megfelelő verziószámú operációs rendszer alatt próbáltuk elindítani a programot.

Függelék

Foglalt file-nevek

A PROPOS–16 operációs rendszer az alábbi elsődleges file-neveket a standard perifériák kezelésénél használja, ezért ezek felhasználói file-névként semmilyen kiterjesztéssel nem használhatók. A foglalt file-neveket egy kettősponttal megtoldva (pl. CON helyett CON:) is ugyanarra a perifériára utalunk. A korlátozás az ilyen kezdetű file-nevekre nem vonatkozik (pl. CON1 vagy CON:ST megengedett mágneslemezes file-nevek)!

Foglalt file-név	Hozzárendelt periféria
CON	Billentyűzet és képernyő. Inputként használva a bevitelt az F6 vagy CTRL–Z karakterekkel zárhatjuk le.
AUX vagy COM1	Első aszinkron kommunikációs adapter.
COM2	Első párhuzamos illesztésű nyomtató.
LPT2, LPT3, ...	További párhuzamos illesztésű nyomtatók.
NUL	Nem létező berendezés (pl. programteszteléshez használható). Inputként azonnal file végét jelez, outputként látszólag elvégzi a műveleteket, de tényleges kivitelt nem hajt végre.

Ezeket a neveket mindenhol használhatjuk, ahol a PROPOS–16 file-név megadását várja. Fizikailag nem létező perifériára hivatkozás (kivéve a NUL nevet) hibát eredményez. A foglalt név előtt megadott meghajtóazonosítót, illetve az utána írt kiterjesztést az operációs rendszer figyelmen kívül hagyja.

Szabványos file-név-kiterjesztések

Az alábbi file-név-kiterjesztésekkel jelölt file-okat a PROPOS–16 operációs rendszer, illetve a hozzákapcsolódó rendszerprogramok (fordítók, szövegszerkesztő stb.) a felsorolt jelentésük szerint értelmezik, ezért ezen kiterjesztések használata más típusú file-okhoz nem javasolt.

A. Függelék

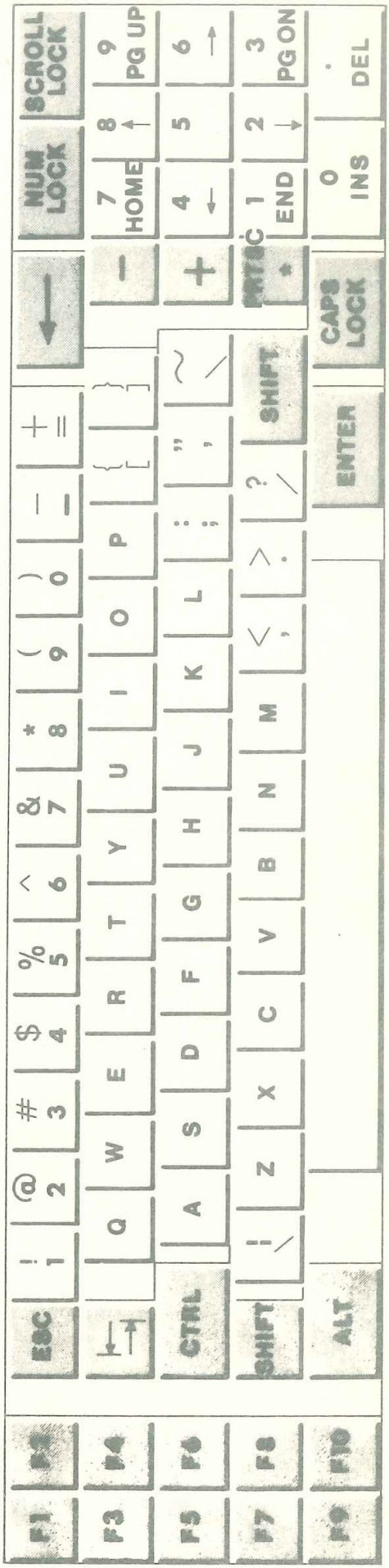
Kiterjesztés	File típusa
.ASM	assembler nyelvű forrásprogram
.BAK	az EDIT program mentési file-ja
.BAS	BASIC nyelvű forrásprogram
.BAT	parancs (BATCH) file
.BIN	bináris file
.C	C nyelvű forrásprogram
.CHK	a STAT parancs által létrehozott file
.COM	futtatható program
.CRF	fordítók által készített keresztreferencia-file
.EXE	összeszerkesztett, futtatható program
.FOR	FORTRAN nyelvű forrásprogram
.LIB	könyvtárfile
.LST	fordítók listafile-ja
.MAP	a szerkesztőprogram által készített tárkiosztás
.OBJ	lefordított (tárgykódú) program
.OVR	program overlay-ág
.PAS	PASCAL nyelvű forrásprogram
.PLI	PL1 nyelvű forrásprogram
.Q	C nyelvű programok fordítási közbülső állománya
.REC	a RECOVER parancs álta ellenőrzött/generált file
.SCR	FORTH nyelvű screen-file
.TMP	ideiglenes file
.$$$	ideiglenes file
.@@@	BACKUP lemezazonosító file

A billentyűzet

A PROPER–16 számítógép billentyűzetének négy váltója van:

CAPS LOCK	Kisbetű/nagybetű állás váltása. A billentyű egyszeri lenyomása után a betűbillentyűk üzemmódja megváltozik: ha előzőleg nagybetűket írtak, ezután kisbetűket fognak generálni és fordítva. A számítógép bekapcsolásakor a kiindulóállapot a kisbetűs üzemmód.
NUM LOCK	A billentyűzet jobb oldalán található numerikus billentyűk üzemmódkapcsolója: egyszeri megnyomásának hatására ezen billentyűk numerikus karaktereket (számokat) generálnak, újbóli megnyomására a kurzor vezérlését végzik. Bekapcsoláskor a kiindulóállapot a kurzorvezérlő üzemmód.
SHIFT	Alsó/felső állás váltása. ■ Az ezzel egyidejűleg lenyomott billentyű felső karaktere generálódik, ha van ilyen (például ; helyett :, 3 helyett #). ■ A CAPS LOCK állásától függően kisbetű állásban nagybetűk írását teszi lehetővé, és fordítva. ■ A NUM LOCK állásától függően a numerikus billentyűzeten szám állásban kurzormozgást tesz lehetővé, és fordítva.
CTRL	Control (vezérlő) billentyű. Az ezzel egyidejűleg lenyomott billentyűk vezérlő karaktert állítanak elő.

B. Függelék


F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
ESC
CTRL
SHIFT
ALT
SHIFT
ENTER
CAPS LOCK
NUM LOCK
SCROLL LOCK
7 HOME
8
9 PG UP
4
5
6
1 END
2
3 PG ON
0 INS
. DEL

A legfontosabb, egy vagy több billentyű lenyomásával kiváltható funkciók:

[ENTER] A beírt sort lezárja, átadja az operációs rendszernek vagy más programnak kiértékelésre. A sort az operációs rendszer megőrzi (l. a funkcionális billentyűk használatának leírásánál).
Menübejegyzés kiválasztására is szolgál.

[ ] (space, betűköz) Egy pozícióval jobbra lépteti a kurzort, egy üres helyet hagyva a képernyőn.

[⇤ ⇥] (TAB=tabulátor) a kurzort jobbra lépteti a következő nyolccal osztható sorszámú pozícióra, a közbenső helyeket üresen hagyva.

[CTRL] [ALT] [DEL]

Az operációs rendszer betöltése lemezről.

[CTRK] [BREAK] vagy [CTRL] [SCROLL LOCK] vagy

[CTRL] [C] Egy program (operációs rendszer parancs vagy felhasználói program) működésének megszakítása.

[CTRL] [NUM LOCK] vagy [CTRL] [S]

Egy program futásának (például folyamatos kiírás) leállítása, illetve továbbengedése.

[SHIFT] [PRT SC]

A képernyő pillanatnyi teljes tartalmát kinyomtatja a gép (ha van nyomtató hozzákapcsolva)

CTRL PRT SC vagy CTRL P

Ennek a két billentyűnek az egyidejű lenyomása után a képernyőn megjelenő minden szöveg folyamatosan a nyomtatóra is kikerül, függetlenül attól, hogy billentyűzetről vagy programból került a képernyőre; újbóli lenyomással a nyomtatást lekapcsolhatjuk.

CTRL ENTER

A beírt sor a következő sorban folytatható, és csak a legközelebbi ENTER beütésekor kerül feldolgozásra.

CTRL H

Tájékoztató információ (HELP) kérése a menüvezérelt üzemmódban.

CTRL Z vagy F6

Bemenő adatok vége, file végjel.

7 HOME	8 ↑	9 PG UP
4 ←	5	6 →
1 END	2 ↑	3 PG DOWN

Numerikus billentyűk, egyben kurzormozgató gombok. Menükezelésnél:

↓ : következő bejegyzés kijelölése
↑ : előző bejegyzés kijelölése
← : visszatérés az előző menülapra

Sorszerkesztő billentyűk

A billentyűzet bal szélén, F1–F10 jelzéssel találhatók az ún. *funkcionális billentyűk*. Ezek hatása programban is definiálható, ám néhánynak már bekapcsoláskor is van funkciója. Ezen billentyűk segítségével lehet az utoljára elküldött parancs-, illetve adatsort „előcsalogatni" és újból felhasználni. Ezt az „előző sort" röviden *mintá*nak nevezzük. A minta csak a következő sor elküldésekor, vagyis az ENTER lenyomásakor változik meg!

F1 — A soron következő karaktert előveszi a mintából, és megjeleníti a képernyőn. A billentyűt lenyomva tartva folyamatosan „előjön" a minta.

F2 — és egy tetszőleges karakter
A megadott karakter első (illetve következő) előfordulása előtti karaktereket veszi elő a mintából.

F3 — A minta hátralevő karaktereit veszi elő.

F4 — és egy tetszőleges karakter
A megadott karakter első (illetve következő) előfordulása előtti karaktereket kihagyja a mintából (átlépi azokat).

F5 — A jelenleg látható utolsó sort elteszi mintának, de nem adja át a programnak (további szerkesztést végezhetünk rajta).

Ha bármely – nem szerkesztésre szolgáló – billentyűt lenyomjuk, az illető karakter megjelenik a képernyőn, és egyidejűleg a minta egy karakterét átlépjük (kihagyjuk).

A sorszerkesztést segíti még néhány billentyű:

← — Egy pozícióval visszalépteti a kurzort, és az utolsó karaktert letörli a képernyőről (a minta nem változik).

DEL — Egy karaktert átlép a mintában.

INS — Karakterek beszúrását teszi lehetővé a minta karakterei közé (a beszúrás végét a billentyű ismételt lenyomása vagy valamelyik funkcionális billentyű alkalmazása jelenti).

ESC — A beírt sort „eldobja" (sem a programnak nem adja át, sem a mintát nem írja felül), új sor adható meg helyette.

C. Függelék

ASCII kódtáblázat

Kód		Képernyő-kimenet	Billentyű-bemenet
Hexa-decimális	Decimális		
00	0	Blank (Null)	Ctrl-2
01	1	☺	Ctrl-A
02	2	☻	Ctrl-B
03	3	♥	Ctrl-C
04	4	♦	Ctrl-D
05	5	♣	Ctrl-E
06	6	♠	Ctrl-F
07	7	●	Ctrl-G
08	8	■	Ctrl-H Backspace, Shift Backspace
09	9	○	Ctrl-I
0A	10	■	Ctrl-J, Ctrl-ENTER
0B	11	♂	Ctrl-K
0C	12	♀	Ctrl-L
0D	13	♪	Ctrl-M, ENTER Shift-ENTER
0E	14	♫	Ctrl-N
0F	15	☼	Ctrl-O
10	16	►	Ctrl-P
11	17	◄	Ctrl-Q
12	18	↕	Ctrl-R
13	19	!!	Ctrl-S
14	20	π	Ctrl-T
15	21	§	Ctrl-U
16	22	▬	Ctrl-V
17	23	↨	Ctrl-W

Kód		Képernyő-kimenet	Billentyű-bemenet
Hexa-decimális	Decimális		
18	24	↑	Ctrl-X
19	25	↓	Ctrl-Y
1A	26	→	Ctrl-Z
1B	27	←	Ctrl [Esc. Shift-Esc. Ctrl-Esc.
1C	28	∟	Ctrl-\
1D	29	↔	Ctrl-]
1E	30	▲	Ctrl-6
1F	31	▼	Ctrl –
20	32	betűköz (space)	Space, Shift, Space, Ctrl Space Alt Space
21	33	!	!
22	34	”	”
23	35	#	#
24	36	$	$
25	37	%	%
26	38	&	&
27	39	’	’
28	40	(	(
29	41	)	)
2A	42	*	*
2B	43	+	+
2C	44	,	,
2D	45	—	—
2E	46	.	. (pont)