

M08X

PROFESSIONÁLIS SZEMÉLYI SZÁMÍTÓGÉP

operációs rendszer

Szki  Sci-L

M08X PROFESSZIONÁLIS SZEMÉLYI SZÁMITÓGÉP

OPERÁCIÓS RENDSZER KÉZIKÖNYV

Budapest, 1983.

M-08.S0017-16. BD-17

OPERÁCIÓS RENDSZER

PROPOS - 8

V.1.0.

C SZÁMITASTECHNIKAI KOORDINÁCIÓS INTÉZET, 1983.

Az OPERÁCIÓS RENDSZER kézikönyvnek ez a kiadása érvényteleníti a korábbi M08X professzionális személyi számítógép Üzemeltetési kézikönyv című SZKI 008/82/B jelű kiadványt.

Jelen kiadvány az SZKI előzetes írásbeli engedélye nélkül sem teljes terjedelmében, sem részleteiben semmilyen módon, illetve eszközzel nem másolható, információvisszakereső rendszerben nem tárolható, nem mikrofilmlezhető.

TARTALOM

| | |
|---|----|
| Bevezető | 5 |
| 1. Az M08X SZÁMITÓGÉP KOMPONENSEI | 7 |
| 2. Az M08X SZEMÉLYI SZÁMITÓGÉP RENDSZER- KEZELÉSÉRE VONATKOZO PARANCSONK | 11 |
| 2.1. Rendszerlemez, indítási módok | 11 |
| 2.1.1. Hidegindítás (Bekapcsolás) | 11 |
| 2.1.2. Melegindítás | 12 |
| 2.2. Lemezcseré | 12 |
| 2.3. Parancs formátum, programindítás | 12 |
| 2.4. File hivatkozások | 13 |
| 2.4.1. Egyértelmű file-név (efn) | 13 |
| 2.4.2. Nem egyértelmű file-név (nfn) | 13 |
| 2.4.3. Lemez hivatkozás a file névben | 14 |
| 2.4.4. File attributumok | 15 |
| 2.5. Parancs beadása, másolás vezérlése | 15 |
| 2.6. Logikai és fizikai beviteli/kiviteli eszközök | 16 |
| 2.7. STAT parancs | 16 |
| 2.8. Parancslánc előállítás és indítása (SUBMIT) | 19 |
| 3. FILE-KEZELÉS | 21 |
| 3.1. A hajlékony lemez tartalomjegyzékének megjelenítése (DIR) | 21 |
| 3.2. Szövegszerkesztő program | 21 |
| 3.2.1. Egyszerű parancsok | 22 |
| 3.2.2. Kurzor mozgató parancsok | 22 |
| 3.2.3. Törlést kiváltó parancsok | 23 |
| 3.2.4. Nyomtatást vezérlő parancs | 23 |
| 3.2.5. Szövegrész másolását vezérlő parancs | 23 |
| 3.2.6. Makro parancsok definiálása és hívása | 23 |
| 3.2.7. Karakter csoportot kereső parancs | 24 |
| 3.2.8. Tabulátor parancsok | 25 |
| 3.2.9. A memória és a lemez közötti adatforgalomra vonatkozó parancsok | 25 |
| 3.3. File-ok nevének megváltoztatása (REN) | 26 |
| 3.4. File-ok törlése (ERA) | 26 |
| 3.5. File-ok megjelenítése (TYPE) | 27 |
| 3.6. Tárterület lemezre való írása (SAVE) | 27 |
| 3.7. Általános másolóprogram (PIP) | 27 |
| 3.7.1. A PIP program indításának módjai | 28 |
| 3.7.2. File-ok másolása | 28 |
| 3.7.3. File-ok összekapcsolása | 29 |
| 3.7.4. A beviteli/kiviteli eszközök logikai nevének megadása | 29 |
| 3.7.5. Speciális beviteli/kiviteli eszköznevek | 30 |
| 3.7.6. Másolást vezérlő paraméterek | 31 |

| | | |
|------|--|----|
| 4. | A PROPOS–8 MŰKÖDÉSÉNEK ÁLTALÁNOS ISMERTETÉSE | 35 |
| 4.1. | Az operációs rendszer elhelyezkedése az M08X személyi számítógép operatív tárjában | 35 |
| 4.2. | A PROPOS–8 programindítás közben biztosított funkciói | 36 |
| 4.3. | A file-kezelés általános jellemzői | 38 |
| 5. | A PROPOS–8 HÍVÁSAINAK VÁZLATOS ISMERTETÉSE | 41 |
| | F Ü G G E L É K E K | 45 |
| | „A” FÜGGELEK | 47 |
| | „B” FÜGGELEK | 51 |
| | „C” FÜGGELEK | 53 |
| | „D” FÜGGELEK (Hibaüzenetek) | 56 |
| | T A R G Y M U T A T Ó | 57 |

BEVEZETŐ

Ez a kézikönyv az M08X professzionális személyi számítógép üzemeltetését és operációs rendszerét, a PROPOS–8-at írja le. (PROfessional Personal Operating System).

A könyv felépítése a következő:

Az 1. fejezet az M08X hardwarejét ismerteti röviden. Ezt a fejezetet minden M08X felhasználónak ajánlatos elolvasni. A részletes hardware ismertetőt az M08X Üzemeltetési Kézikönyv tartalmazza.

A 2. fejezetben az M08X rendszer kezelését ismertetjük. A bekapcsolásra, az indítási módokra és a parancsokra vonatkozó fejezeteket minden felhasználónak ismernie kell. Speciális programok már ezek alapján is használhatók (természetesen, az adott programok részletes leírása alapján).

A 3. fejezet az M08X file-kezelését tárgyalja. Ez a fejezet nagyon fontos, amikor az M08X-en adatokat akarunk előkészíteni, vagy lemezfile-okat kezelni.

A 2. és 3. fejezet elegendő ismeretet tartalmaz tetszőleges program kezeléséhez az adott program kezelési leírásának birtokában, illetve bonyolultabb adatelőkészítő és rendszerkezelő tevékenységek elvégzéséhez.

A 4. fejezet a PROPOS–8 működésének vázlatos ismertetése, amit a PROPOS 8-al dolgozó, számítástechnikában jártas szakembernek szántunk.

Az 5. fejezet a PROPOS–8-nak a felhasználói programokhoz nyújtott szolgáltatásait ismerteti. Ismerete azok számára szükséges, akik az M08X számítógépre programokat írnak.

A könyvhöz függelékeket csatolunk, ezek a programozók számára tartalmaznak fontos ismereteket.

A könyvet a hibaüzenetek gyűjteménye és a tárgymutató egészíti ki.

1. AZ M08X SZÁMITÓGÉP KOMPONENSEI

A számítógéprendszert hardware és software részre szokás bontani, ahol az előbbi a ténylegesen „megfogható” tartozékokat és alkatrészeket, az utóbbi pedig a rendszer működését biztosító programok összességét jelenti. Ez a fejezet az M08X hardwarejét ismerteti vázlatos formában és néhány gyakorlati tanácsot is ad a használatra vonatkozóan, míg a következő négy fejezetet együttesen „M08X software”-nek is nevezhetnénk.

Az M08X mind hardware, mind software szempontból tipikus felépítésű mikroszámítógép rendszer. Alkotóelemei a számítógép, a billentyűzet, a képernyő, 1–4 lemez meghajtó és a sornyomtató.

A számítógép

A számítógépet a képernyővel azonos dobozban helyezték el. Ez a doboz tartalmazza a rendszer egyes elemeit vezérlő áramköröket, a központi egységet és a memóriát. A számítógép memória tárolja azokat a programokat amelyek futtatni akarunk, ugyanis a gép csak a memóriájában lévő programokat képes végrehajtani.

Az M08X memóriakiépítése 64 kbyte (65526 byte), amiből 62 kbyte használható a rendszer és felhasználói programok tárolására. Az M08X operációs rendszere 7 kbyte méretű, így a felhasználói programok tárolására 55 kbyte memóriaterület jut.

Megjegyzés: A memória tartalma a gép kikapcsolásakor elvész, tehát minden bekapcsoláskor a szükséges programokat – így az M08X software-t is – be kell tölteni a hajlékony lemezről. Ugyancsak elveszhet a memóriatartalom a RESET gomb megnyomásakor is.

A billentyűzet

Az M08X billentyűzete hasonlít az elektromos írógéphez. A gép működése közben lenyomott (nem vezérlő) billentyű tartalma megjelenik a képernyőn. Ha egy billentyűt hosszabb ideig lenyomva tartunk, a tartalma, illetve funkciója addig ismétlődik, amíg azt el nem engedjük.

A „CR” gomb leütésével az előzőleg beadott parancsot küldjük el a számítógépnek, amire az valamilyen működésbe kezd.

A CTRL „control” gomb működése statikus, tehát ha a leírásban „CTRL P” vagy „CTRL C” szerepel, akkor a CTRL és a P vagy a C gombot egyidejűleg kell lenyomni, hogy a kívánt funkciót aktivizáljuk. Ez úgy történik, hogy a control gombot lenyomva tartjuk és ez alatt az idő alatt ütjük le az előbbi példában szereplő P vagy C billentyűt.

Ugyanígy működik a SHIFT gomb is. Ezzel a billentyűvel kapcsolhatunk kis betűről nagy betűre, illetve vissza. Ha a CTRL D gombokat lenyomjuk, akkor nagy betűk helyett kis betűket, illetve fordítva, kis betűk helyett nagy betűket kapunk.

A képernyő

A képernyőn többféle megjelenítési mód létezik, villogó, fordított – inverz –, aláhúzott stb. Az M08X alapsoftware nem használja ki a normáltól eltérő megjelenítési módokat, de

ezek a felhasználói programból természetesen kihasználhatók. A képernyőn megjelenített szöveg rosszul olvasható ha a képernyőt közvetlenül fény éri; ha a nap sűti a képernyőt, a szöveg teljesen láthatatlan lesz.

A programozó számára a „B” függelékben ismertetjük a képernyő kezelésének módját.

Hajlékony lemez

A hajlékony lemez az MOBX rendszer háttértárolója. A szállított egységtől függően szimpla, vagy dupla sűrűségű, 8”-os lemez használható. A lemez lehet 1 vagy 2 oldalas, attól függően, hogy az egyik vagy mindkét oldalát bevonták mágneses anyaggal.

A lemezen az információ koncentrikus körökön helyezkedik el, ezeket sávoknak nevezzük. A sávokat egyenlő nagyságú körcikkekre osztották fel, ezek a szektorok. A szektor az a legkisebb információ egység, amit az MOBX rendszer direkt módon címezni és írni/olvasni tud. A lemezen általában két címke van, egy négyzet alakú ún. „állandó” és egy téglalap alakú „azonosító” címke. Ez utóbbit a lemez tulajdonosa belátása szerint cseréli. A lemez lehet írásvédett; ez azt jelenti, hogy a rendszer nem ír a lemezre akkor, ha a tasakon az írásvédelmi „kivágás” megvan.

Megjegyzés: A lemez akkor védett, ha annak – a címke felől nézve – jobb élén egy rendszerint négyzet alakú „kivágás” van.

Az MOBX rendszer írásvédett állapotba helyezi a lemezt, ha azt a meghajtóból kivették és más lemezt visszahelyeztek a rendszer újraindítása nélkül. Írásvédett állapotba lehet helyezni a lemezt a STATparancs felhasználásával is.

A hajlékony lemez kezelésével kapcsolatban szem előtt tartandó szabályok:

- A hajlékony lemez egy négyzet alakú tasakban van, abból kivenni nem lehet, azzal együtt kell a meghajtóba helyezni,
- a lemezt összehajtani nem szabad,
- ne tegyük ki hőnek, erős mágneses térnek,
- csak puha végű tollal szabad a címkére írni,
- a négyzetes tasakba vágott ovális nyíláshoz nyúlni nem szabad,
- a lemezt mindig úgy kell a meghajtóba helyezni, hogy a címkével átellenben lévő él kerüljön először a meghajtó nyílásába és ütközésig kell becsusztatni. A címke felül, illetve a lemez meghajtó gombjának háttal van. Helyes betétel esetén a lemez a helyén marad és a gomb benyomásával rögzíthetjük. A lemez kivételénél az eljárás fordított; helyes végrehajtás esetén a lemez néhány centivel kiugrik a helyéről, amikor könnyedén kiemeljük és a borítékba helyezzük,
- A készülék teljes kikapcsolása előtt győződjünk meg arról, hogy nem maradt-e lemez valamelyik meghajtóban,
- a lemez meghajtó működés közben kattogó hangot hallat. A kattogó hang azt jelenti, hogy a meghajtó éppen dolgozik, tehát nem szabad kinyitni.

A fenti szabályok betartása ellenére is előfordul, hogy a lemez olvashatatlanná válik. Ezért nagyon fontos, hogy minden lemezről legyen másolat. Minden hosszabb adatbevitel végén – néha még közben is – ajánlatos a fontosabb állományokról másolatot készíteni.

Az MOBX a hajlékony lemez meghajtóit 0, 1, 2, vagy 3 számmal, a felhasználó – a különböző felhasználói programok futtatása során – A, B, C, vagy D betűkkel azonosítja. C, D jelzésű meghajtó csak a 4 meghajtóval szállított konfigurációkon van. Fontos tudnunk, hogy melyik meghajtó milyen azonosítóval szerepel a rendszerben. Jó gyakorlat, hogy az egyes meghajtókra címkéket ragasztanak a nekik megfelelő jelzésekkel.

A továbbiak során feltételezzük – ha csak explicite nem írjuk az ellenkezőjét –, hogy az MOBX két meghajtóval működik.

Sornyomató

A nyomtató írásban jeleníti meg az adatokat, programokat, futási eredményeket. Az M08X alapvetően két fajta, vagy DZM, vagy CENTRONICS interface-ű nyomtatóval kerül forgalomba. Ha az M08X nyomtatóját rendszeresen cserélik a két típus között, a felhasználóknak ajánlatos a STAT program (lásd a rendszerkezelés parancsait tárgyaló fejezetet) működésével tisztában lenni.

Az eddigi tapasztalatok szerint a számítógép használatával kapcsolatban a legtöbb gyakorlati nehézséget a nyomtató okozza. Nem ritka, hogy a fáradt és ideges felhasználó a nyomtatót hibásnak tartja és „megjavítja”, vagy durván kezeli (ütögeti, rázza). Az ilyen esetek többségében a nyomtató hibátlanul működik, de például a papírt, vagy a festék-szalagot rosszul helyezték be. Minden felhasználónak ismernie kell a papír leszakításának helyes módját. Gyakran megesik ugyanis, hogy a nyomtató éppen a helytelen kezelés miatt továbbítja szabálytalanul a papírt.

Általános gyakorlat, hogy váltogatva, több papírszélesség használatos. Ezért fontos, hogy minden felhasználó ismerje a papírszélesség beállításának módját is.

Ennyi az a minimális hardware ismeret, ami az M08X számítógép kezeléséhez szükséges. Hangsúlyozzuk, hogy az aktuális nyomtatóhoz külön kezelési utasítás szükséges. A fenti tanácsok betartása a gyakorlatban nem okoz nehézséget, néhány órai munka után alkalmazásuk természetessé válik.

2. AZ M08X SZEMÉLYI SZÁMÍTÓGÉP RENDSZERKEZELÉSÉRE VONATKOZÓ PARANCSOK

Rendszerlemez, indítási módok

2.1.

A rendszer indításához szükség van legalább egy ún. „rendszer” lemezre, amin a gép forgalmazója az M08X software-t elhelyezte. Ajánlatos, hogy több rendszerlemez legyen a gép közelében, hiszen ezek is megsérülhetnek a korábban említett okok miatt.

A rendszerlemez első két sávján helyezkedik el a PROPOS-8 rendszer, ami a – továbbiakban részletes ismertetésre kerülő – funkciók mellett a „DIR”, a „REN”, az „ERA”, a „TYPE” és a „SAVE” beépített parancsokat is végrehajtja (lásd 3. fejezet). Ez azt jelenti, hogy ilyen nevű programokat nem találunk a „rendszer” lemezen. A „STAT”, a „SUBMIT”, az „EDIT” és a „PIP” (lásd 2. és 3. fejezet) funkcióit az azonos nevű programokkal valósfitták meg (tehát a STAT funkciót a STAT.COM programmal stb.), amiket a lemezen meg kell találnunk.

Hidegindítás (Bekapcsolás)

2.1.1.

Hidegindítást akkor végzünk, ha a rendszert kikapcsolt állapotból kiindulva helyezzük üzembe (lásd az üzemeltetési kézikönyv megfelelő fejezetét). Az M08X kulcsos hálózati kapcsolóját fordítsuk el. A hálózati feszültséget jelző lámpa kigyullad, és a képernyőn a következő szöveg jelenik meg inverz karakterek (világos alapon sötét betűk) formájában:

```
TERM. MOD 1      FULL dpx...
CTRL ESC         (felhasználó adja be CTRL és ESC egyidejű lenyomásával)
USERS MODE?     (inverz, villogó kijelzés)
F               (felhasználó adja be)
EGYSEG? 0 VAGY 1
                Ekkor kell elhelyezni a rendszert tartalmazó lemezt a „0” jelölésű
                meghajtóba
0               (felhasználó adja be)
A >            A rendszer válasza, ami azt jelzi, hogy az M08X software parancsot
                vár.
```

Az M08X rendszert az „1” egységről is indíthatjuk, ebben az esetben az eljárás az előzőkben leírtakhoz teljesen hasonló, de a rendszert tartalmazó lemezt az „1” egységbe kell helyezni, és a „0” egységbe is egy hajlékony lemezt kell tenni.

Az utolsó kérdésre a válasz nem 0, hanem 1. Tehát az eljárás:

```
                bekapcsolás
TERM. MOD 1      ...
CTRL ESC         felhasználó
USERS MODE?     felhasználó
F               felhasználó
EGYSEG? 0 VAGY 1
                rendszerlemez az „1” egységbe és a „0” egységbe egy hajlékony
                lemez.
1               felhasználó
B >            a rendszer parancsra vár.
```

2.1.2. Melegindítás

Ha a rendszer bizonytalanná válna, vagy a megszokottól teljesen eltérő módon működik, – esetleg nem jelentkezik ki A > vagy B > –, ajánlatos ún. „melegindítást” végezni. A melegindítást a CTRL C (CTRL és C együtt) leütéssel válthatjuk ki. Ekkor megjelenik az A > vagy B > jel attól függően, hogy melyik meghajtót választottuk ki utoljára (lásd 2.2.), ami a rendszer parancsra várakozás állapotát jelzi. A melegindítás alkalmával az MOBX rendszer újra töltődik, ami által az esetleg felülírt részek ismét hibátlanok lesznek.

Minden esetben melegindítást kell alkalmazni, ha lemezt cserélünk, és erre a lemezre írni szándékozunk! A rendszer ugyanis az újonnan behelyezett lemezt írásvédett állapotba hozza, és csak a melegindítás után enged írást az új lemezre. Ajánlatos melegindítást végezni ha egy program futását meg akarjuk szakítani. Egy véletlenül elindított programot a RESET gomb megnyomásával is megszakíthatunk. Ha egy futó program a lemezt használja (a lemez kattogó hangot ad), nem szabad azt a gép kikapcsolásával vagy a lemez meghajtóból való kivételével megállítani! Abban az esetben, ha a melegindítás nem sikerül – nem jön az A > jel –, vegyük ki a meghajtóból a lemezeket és végezzünk hidegindítást a 2.1.1. pont szerint.

2.2. Lemezcseré

A lemezcseré elnevezés két egymástól független tevékenységet is jelöl: egy hajlékony lemez tényleges cseréjét valamely meghajtóban, vagy a „feltételezett” meghajtó cseréjét. Az elsőt illetően lásd 2.1.2. pontot.

A feltételezett meghajtó az, amelynek az azonosítója hideg-, vagy melegindítás után a > jel előtt megjelenik. Az MOBX rendszer minden parancsot vagy filet a feltételezett meghajtón keres, ha a parancsban vagy a file-névben nem adtuk meg a meghajtó nevét.

Példa: A > az „A” a feltételezett meghajtó jele.

A > xyz az xyz programot az „A” meghajtón keresi a rendszer.

B > xyz az xyz programot a „B” meghajtón keresi a rendszer.

Ha a feltételezett meghajtót meg akarjuk változtatni, az új meghajtó azonosítóját gépeljük be egy kettősponttal (:) lezárva. A parancsot a CR gomb lenyomásával küldjük el.

A > b: Az „A”-ról „B”-re váltás.

B > a: „B”-ről „A”-ra váltás

A >

2.3. Parancs formátum, programindítás

Az MOBX rendszernek szóló parancs formátuma három féle lehet:

```
parancs
parancs file1
parancs file1 file2
```

Minden parancsot a CR billentyű lenyomásával kell lezárni!

A „parancs” a futtatandó program neve. A rendszer egy parancs .COM filet keres a feltételezett meghajtóban lévő lemezen, vagy azon a lemezen, aminek az azonosítóját a parancs előtt beadtuk.

Példa: A > xyz hatására, a rendszer az xyz.com programot az „A”, tehát a feltételezett meghajtóban lévő lemezen keresi, de az

A > b: xyz hatására az xyz.COM programot már a „B” lemezen keresi a rendszer.

A > b: }
B > } lemezcseré, (lásd 2.2. pont)
B > xyz a hatás azonos a második példával,
B > a: xyz a parancs hatása már az első parancs hatásával azonos.

Kezdőknek ajánlatos gyakorlásul elküldeni néhány parancsot.

Ne feledkezzenek meg a CR-rel való lezárásról!

Példa: A > xyz
NOT FOUND
A > huhu
NOT FOUND jelenti, hogy xyz.com és huhu.com file nincs az „A” meghajtóban lévő lemezen.

Bátran gyakorolhatjuk a gép üzemeltetését, a billentyűzetet használva nem tehetünk tönkre semmit, de azért óvakodjunk az ERA és REN-nel kezdődő parancsok beadásától.

A file1 és file2 hivatkozásokról a 2.4. pontban lesz szó.

File hivatkozások

2.4.

A 2.3. pontban láttuk, hogy egy M08X parancsban maximum két file hivatkozás lehet. A további példákban a file-nevet rendszerint nagy betűvel írjuk, hogy a példákat meg lehessen különböztetni az esetleges magyarázatoktól. Valójában a file-nevet megadhatjuk kisbetűvel, nagybetűvel vagy azok tetszőleges kombinációjával.

Egyértelmű file-név (efn)

2.4.1.

Az M08X rendszerben – mint más mikrogépes rendszerekben is – a file-név két részből áll: egy nyolc betűs elsődleges file-névből és egy három betűs másodlagos file-névből. A másodlagos file-nevet szokás kiterjesztésnek, vagy típusnak is nevezni. Leírva, a két részt egy pont választja el egymástól, tehát az

nnnnnnn.kkk
egy file egyértelmű neve, ahol n és k tetszőleges karakter lehet a
„,: = ? *

speciális karakterek kivételével. A ki nem töltött betűk a helyköz (blank) karakterrel azonosak.

Példa: SZAMLA azonos a SZAMLA.bbb
FIZETESE.K azonos a FIZETESE.Kbb
file névvel, ahol a „b” karakter helyközt jelöl.

Nem egyértelmű file-név (nfn)

2.4.2.

A nem egyértelmű file-név formája hasonló az egyértelmű file-névhez, de a névben ? , vagy * karakter is szerepel.

A ? azt jelenti, hogy azon és csak azon a helyen bármilyen érvényes karakter állhat a file nevében. Ily módon egyetlen nem egyértelmű file névvel több file-t is jelölhetünk, pontosan azokat, amelyek megfelelnek a kijelölésnek.

Speciális esetben, amikor az elsődleges file-név, vagy kiterjesztés ? karakterekkel végződik, a ? karaktersorozat helyett a * karaktert is használhatjuk. Abban az esetben is használhatjuk a * karaktert, ha a teljes (elsődleges) file-név, vagy a teljes kiterjesztés ? karakterekből áll.

Ez azt jelenti, hogy tetszőleges (elsődleges) file-névű, vagy tetszőleges típusú file-a név által meghatározott csoportba tartozik.

| | | | |
|--------|---------------|------------------|-----------------|
| Példa: | ez | megfelel ezeknek | nem felel meg |
| | k?c?i | KICSI kocsi | kacsa kicsi.tmk |
| | A?I.* | kiczi KYCZI | KACSIN |
| | | ani.ani api.huh | ara.asm |
| | | ari.ani | |
| | SYS*.* | bármely SYS-sel | SISTEM.ASM |
| | SYS?????.??? | kezdődő file név | |
| | | SYSTEM.COM | |
| | | SYSIPHOS.ITT | |
| | ?????????.??? | bármely file név | |
| | *.* | bármely file név | |

Ez a kézikönyv az MO8X egyes parancsainak tárgyalásánál megadja, hogy a felhasználó az adott esetben egyértelmű file-nevet (efn) kell hogy használjon, vagy nem egyértelmű file-nevet (nfn) is használhat. Természetesen, minden olyan parancsban, ahol nem egyértelmű file-név megadása megengedett, egyértelmű file-nevet is használhatunk. A nem egyértelmű file-névnek az az előnye, hogy ott egy parancs beütésével egész file csoportot is kezelhetünk, ha valamilyen elnevezési konvenciót használunk. Ez a célja a file-név kiterjesztésének, vagy típusának. Az MO8X esetében – más rendszerhez hasonlóan – bevezettünk elnevezési konvenciókat, ezeket adjuk meg a következő táblázatban:

| kiterjesztés | jelentés | példa |
|-----------------|--|------------------------------------|
| COM | parancs vagy futtatható program | STAT.COM |
| ASM | assembly nyelvű forrásprogram | KERES.COM KERES.ASM IHOL.ASM |
| PRN | assembly nyelvű forrásprogram lista file-ja | KERES.PRN IHOL.PRN |
| BAS | BASIC forrásprogram | PÉLDA.BAS |
| FOR | FORTRAN nyelvű forrásprogram | SZAMOL.FOR |
| SRC vagy PAS | PASCAL nyelvű forrásprogram | SORT.SRC, SORT.PAS |
| C | C nyelvű forrásprogram | BEIR.C |
| CRL | C nyelvű relokálható forrásprogram | BEIR.CRL |
| BAK | editor program mentés file-ja | KERES.BAK |
| \$\$\$ | editor munkafájl-ja, editor törli | KERES. \$\$\$ |
| SUB | köteget futás parancs file-ja | FRDFUT.SUB |

2.4.3 Lemez hivatkozás a file névben

A file név megadásában a felhasználó jelezheti, hogy az általa keresett file melyik meghajtóban lévő lemezen található. A meghajtó jelét közvetlenül a file név előtt kell beütni. A

meghajtó jele és a file név közé : karaktert kell írni.

Példa: a:xyz.abc az „A” meghajtóban lévő hajlékony lemez xyz.abc file-ját jelenti.
b: pelda.asm a „B” lemez pelda.asm file-ját jelenti.

File attributumok

2.4.4.

Az MOBX rendszerben minden file-nak két alapvető tulajdonsága lehet: „rendszer” (SYS) vagy „nem rendszer” (DIR) file, illetve „írásvédett” (R/O) vagy „nem írásvédett” (R/W) file. „DIR” attributumú file neve megjelenik a képernyőn a „DIR” parancs hatására (lásd 3. fejezet, DIR parancs). A file-ok túlnyomó többsége „DIR” file, mivel a felhasználó nem akarja a fileokat elrejteni. A „DIR”-rel ellentétes tulajdonság a „rendszer”, vagy „SYS” file. A „SYS” attributumú file neve nem jelenik meg a „DIR” parancs hatására, tehát „rejtve marad”. A PIP parancs (lásd 3. fejezet) sem találja meg normál esetben, tehát ez a tulajdonság bizonyos védelmet is nyújt másolás ellen. Az ERA (a file-ok törlésére szolgáló) parancs megtalálja a rendszer file-t is, tehát a rendszer file törölhető az adott lemezről.

Az „írásvédett” – R/O – file-ra nem enged írást a rendszer. Az ilyen állományt nem törölhetjük az ERA parancs, illetve nem nevezhetjük át a REN parancs segítségével sem. A „nem írásvédett” – R/W – attributum az előbbi tulajdonságok hiányát jelenti.

A file attributumokkal kapcsolatban a STAT és a DIR parancsokat taglaló fejezeteket javasoljuk elolvasni.

Parancs beadása, másolás vezérlése

2.5.

A MOBX rendszer bizonyos „szerkesztési” lehetőségeket nyújt a felhasználó számára a parancs beadása közben. Ezen lehetőségek felhasználásával tudunk például hibásan beütött, de el nem küldött parancsot javítani. Az adott funkciót a megfelelő gombok lenyomásával válthatjuk ki; ne feledjük, hogy a CTRL „billentyű” a CTRL és billentyű egyidejű lenyomását jelenti.

A következő szerkesztési funkciók állnak rendelkezésre:

- EL Az utolsó leütött karaktert törli és a kurzor egy pozícióval visszaugrik.
- CTRL U Törli a parancsot és új sort kezd a képernyőn.
- CTRL X Törli az egész begépelte parancsot.
- CTRL R A begépelte parancsot újra megjeleníti.
- HT Tabulátor billentyű. A tabulátor értékek minden nyolcadik pozícióban vannak.
- CTRL E A begépelte parancs sor fizikai vége. A kurzor visszatér a következő sor elejére, ahol a parancs folytatható. Az egész parancsot a CR gomb lenyomásával küldjük el a rendszernek.
- CTRL Z A billentyűzetről való begépelés vége (lásd PIP parancs).
- CR Parancs lezárása.

A CTRL U, a CTRL X és a CTRL R az egész begépelte (tehát a CTRL E-vel megszakított) parancsra vonatkoznak.

A következő másolást vezérlő lehetőségeket nyújtja a rendszer:

- CTRL P Minden utána következő képernyő kiírást a sornyomtatóra másol. A kiírás az újabb CTRL P-ig tart.
- CTRL S Megállítja a képernyőre való kiírást.
Akkor szükséges, ha a képernyőre való írás olyan gyors, hogy azt a gép kezelője nem tudja elolvasni. A következő CTRL S hatására a kiírás folytatódik.

2.6. Logikai és fizikai beviteli/kiviteli (I/O) eszközök

Az M08X-en négy ún. logikai I/O eszközt különböztetünk meg. Az egyes programok mindig a logikai néven szólítják meg az I/O eszközöket, és nem a fizikai néven. Ennek az az előnye, hogy a fizikai eszközök cseréje esetén nem kell átírni a programokat, hanem a logikai és fizikai nevek összerendelésének megváltoztatásával (például STAT parancs) figyelembe vehetjük a változást. A négy logikai I/O eszköznév:

CON: a rendszer billentyűzete
RDR: a rendszer olvasó eszköze
PUN: a rendszer kiíró eszköze
LST: a rendszer listázó eszköze

A fizikai eszközök nevei a következők lehetnek:

TTY: kis sebességű írógép
CRT: billentyűzet és képernyő fizikai neve
BAT: ún. kötegelt feldolgozás
UC1: felhasználó által megadott beíró eszköz
PTR: kommunikációs beviteli eszköz
UR1: felhasználó által megadott első olvasó eszköz
UR2: felhasználó által megadott második olvasó eszköz
PTP: kommunikációs kiviteli eszköz
UP1: felhasználó által megadott első kiíró eszköz
UP2: felhasználó által megadott második kiíró eszköz
LPT: különböző típusú sornyomtatók
UL1:

A logikai és fizikai eszköznevek összerendeléséről lásd a 2.7. pontot.

2.7. STAT parancs: a hajlékony lemez, az egyes file-ok és a rendszer állapotának megjelenítése illetve megváltoztatása

A STAT parancs formátuma eltér a 2.3. pontban taglalt általános parancs-formátumtól, mivel a paraméterek között nemcsak file hivatkozás lehetséges. Ezért a STAT formátuma:

STAT paraméterek

ahol a paraméterek rész el is maradhat.

STAT Ha a paraméter terület üres, a STAT program minden működő meghajtóban lévő hajlékony lemezre megadja a rajta rendelkezésre álló üres terület nagyságát, valamint a meghajtó állapotát. A válasz tehát a következő:

x: R/W, SPACE: nnnK vagy
x: R/O, SPACE: nnnK, ahol

x a meghajtó jele, R/W „nem írásvédett”, az R/O „írásvédett” állapotot jelent, nnn az adott lemezen található szabad hely mérete kbyte-ban.
(1 kbyte = 1024 byte)

STAT x: A paraméter egy meghajtó neve; ekkor csak a megadott meghajtóban található lemez adatait kapjuk meg.

STAT nfn A paraméter egy nem egyértelmű file-név, ekkor a csoporthoz tartozó file-ok neveit kapjuk ABC sorrendben méreteikkel együtt, valamint az egyes file-ok attribútumait a következő fejléc alatt:

| Recs | Bytes | Ext | Acc | |
|------|-------|-----|-----|-----------------|
| rrrr | bbbK | ee | | d: pppppppp.sss |

rrr jelzi a file-ban található 128 byte-os rekordok számát, bbb a file mérete kbyte-ban. (bbb = rrrx128/1024), ee a 16 kbyte-os területek száma (ee=bbb/16), d a meghajtó jele, p . . . p a file-név, sss a típus. Acc alatt R/O írásvédett, R/W nem írásvédett állapotot jelent. Ha a file-név zárójelben van, az illető file „SYS” file, ellenkező esetben „DIR” file.

Nem egyértelmű file-név előtt meghajtó azonosítót is beadhatunk, ekkor az adott lemez file-jainak adatait kapjuk meg.

Emlékezzünk arra, hogy minden esetben, amikor nem egyértelmű file-név megadása megengedett, egyértelmű file nevet is megadhatunk. Ez egy bizonyos file keresésére ad kényelmes lehetőséget.

STAT x:= R/O Az X meghajtót írásvédett állapotba helyezi. Fontos, hogy minden lemezcsere is írásvédett állapotba helyezi a lemezt, és ezt melegendítással lehet feloldani.

STAT VAL: A logikai és fizikai I/O eszközök (lásd 2.6. pont) lehetséges összerendeléseit jeleníti meg.

| | | | | | |
|------|---|------|------|------|------|
| CON: | = | TTY: | CRT: | BAT: | UC1: |
| RDR: | = | TTY: | PTR: | URI: | UR2: |
| PUN: | = | TTY: | PTP: | UPI: | UP2: |
| LST: | = | TTY: | CRT: | LPT: | UL1: |

Megjegyzés: ez az összes megengedett összerendelést jelenti, de nem biztos, hogy minden összerendelés értelmes is az MO8X esetében.

STAT DEV: A logikai és fizikai I/O eszközök (lásd 2.6. pont) aktuális összerendelését jeleníti meg. Az MO8X esetében a következő összerendelések lehetségesek:

| | | |
|------|---|-----------------------|
| CON: | = | CRT: |
| RDR: | = | PTR: |
| PUN: | = | PTR: |
| LST: | = | CRT:, LPT:, vagy UL1: |

Az egyes logikai és fizikai nevek konkrét jelentését illetően lásd 2.6. pontot. Nagyon lényeges, hogy az LPT: és UL1: eszközök különböző típusú nyomtatókat jelölnek. Ha egy MO8X-et váltakozva különböző típusú nyomtatókkal használunk, fontos, hogy mindig a megfelelő összerendelés legyen érvényben, különben a nyomtató nem működik!

STAT len1=fen1, len2 = fen2, . . . , lenn = fenn

A parancs a logikai és a fizikai I/O eszközök aktuális összerendelésére szolgál. Az MO8X jelenlegi kiépítésében ez a következő lehet:

| | |
|--|-----------|
| STAT CON:=CRT:,RDR:=PTR:~,PUN:=PTP:~,LST:= | UL1: |
| | vagy LPT: |
| | vagy CRT: |

A lista eszköz fizikai nevének megválasztásához lásd STAT DEV: parancs leírását.

STAT efn \$attr A parancs az efn nevű file attributumának megváltoztatására szolgál. Az „attr” lehet SYS, DIR vagy R/O, R/W. A parancs sikeres végrehajtását a „SET TO attr” üzenettel jelzi a rendszer. Az egyes attributumok jelentését lásd 2.4.4. pontban.

STAT x:DSK: A parancs az x nevű lemezre vonatkozó adatokat adja meg. Az „x:” el is maradhat, ekkor az összes működő (az előző „melegindítás”-tól eltelt időben megszólított) lemez adatait adja meg. A kiírt adatok a következők:

- a lemez azonosítója (a: vagy b:),
- a lemez tárolókapacitása 128 byte-os rekordokban és kbyte-ban (1024 byte) mérve,
- a lemez elhelyezhető file-leíró blokkok maximális és ellenőrzött száma,
- egy „extent”-ben található 128 byte-os rekordok maximális száma,
- egy „blokk”-ban található rekordok száma,
- egy sávon található szektorok száma és a fenntartott sávok száma.

A gép kezeléséhez nem szükséges minden adat jelentését pontosan ismerni.

Megjegyzés: a „blokk” és az „extent” egy file helyfoglalásának egységei. A „blokk” a legkisebb lefoglalható egység, az „extent” az a legnagyobb terület, amelyet egy file leíró blokk megadhat.

Példa:
A > STAT DSK:
A > STAT A:DSK:
A > STAT B:DSK:.

Mint láttuk, a STAT parancs ismerete nagyon lényeges a rendszer, az egyes file-ok és lemezek állapotának beállításához, illetve kijelzéséhez. A legegyszerűbb forma gyakori használata javasolt, mivel ajánlatos tudni, hogy mennyi üres hely van az egyes lemezekben. Különösen fontos egy file szerkesztése előtt megnézni, hogy az adott lemezen van-e annyi üres hely, amennyit az új állomány várhatóan lefoglal (lásd szövegszerkesztés 3.2. pont).

Az M08X rendszer használata során rendszerint kiderül, hogy a felhasználók gyakran kénytelenek ugyanazt a parancs-sorozatot begépelni. Programfejlesztés esetén például a forráskód lefordítása, a lefordított program futtatható alakra hozása és futtatása, vagy felhasználói rendszer esetén az adatok gyűjtése, azok ellenőrzése egy újabb program segítségével, a felvitt adatok másolása egy harmadik program felhasználásával azonos parancssorozat ismételt begépelésével történik. Azért, hogy ne kelljen ugyanazt a parancs-sorozatot állandóan megismételni – ami újabb hibák elkövetésére ad alkalmat – lehetőség van az ilyen sorozatok egy file-ban való összegyűjtésére. Az így felsorolt parancs-sorozatot azután egyetlen parancs begépelésével indíthatjuk, így a kezelés kényelmesebb és egyben biztonságosabb lesz. A parancs-sorozat indítást kötegelt (batch) feldolgozásnak is nevezik. Az elnevezésnek történeti oka van; nagygépes környezetben így nevezik azt a feldolgozást, amikor a gép számára az egyes programokat – és adatokat – egy csomagban adják meg, és az így összeállított „kötegt” a felhasználó közreműködése nélkül lefut.

Mint látni fogjuk, az M08X esetében a tárolt parancs-sorozatokat nem keverhetjük az elindított programoknak szánt adatokkal, viszont figyelemmel kísérelhetjük a parancs-sorozat lefutását és megszakíthatjuk azt.

Ha a következő magyarázat első olvasásra nehéznek tűnik a számítástechnikában kezdő felhasználóinknak, haladjanak tovább és néhány heti M08X gyakorlat után térjenek vissza a témára. A gép ugyanis korlátozások nélkül használható a SUBMIT parancs alkalmazása nélkül is.

A parancs formája a következő:

```
SUBMIT efn v1 v2 . . . vn
```

A file-név az ún. parancs file, a parancs-sorozatot tartalmazó file egyértelmű neve. A file típusa kötelezően SUB, amit a SUBMIT parancsban nem kell megadni. A v1 v2 . . . vn paraméterek megadása nem kötelező, de megadásuk azt jelenti, hogy a parancs-fileban megadott változókat rendre v1 v2 . . . vn konkrét értékekkel kell helyettesíteni.

A parancs-file-ban M08X parancsok vannak, a változók \$k alakúak, ahol k 1 . . . n értéket vehet fel. A SUBMIT parancs elindítása után a \$1 változó a v1, a \$2 a v2 stb. konkrét értékekkel helyettesítődik, és az így kapott parancs-sorozat a \$\$\$\$.SUB segéd-fileben tárolódik.

Ezek után a SUBMIT parancs megleindítást hajt végre és ezzel a \$\$\$\$.SUB segéd-file-ban lévő konkrét parancs-sorozat az adott sorrendben végrehajtott.

A parancs-file-t szövegszerkesztéssel állíthatjuk elő, pontosan úgy, mint bármely más szöveg-file-t. A file előállításánál a következő szabályokat kell betartani:

- ha \$ jelet akarunk beírni a parancs-file-ba, és az nem változót jelöl, megkülönböztetésül két \$ karaktert viszünk be,
- ha a CTRL + betű kombinációt akarunk bevinni, akkor a ^ + betű kombinációt kell beadni, mivel a CTRL gombra a szövegszerkesztő program valamilyen műveletet hajt végre, és azt nem mint adatot kezeli.

A parancs-sorozatot az EL gomb lenyomásával szakíthatjuk meg, ez után új parancsot adhatunk be.

Megjegyzés: a parancs-file csak az „A” lemezen lehet.

Egy parancs-file tartalmazhat SUBMIT parancsot, tehát a parancs-file-ok egymáshoz láncolhatók. Ez a láncolás azonban nem jelent egymásba ágyazást, mivel az újonnan indított parancs-láncból nem lehet az azt indító parancs-láncba visszatérni. Ez azt jelenti, hogy egy parancs-láncban az esetleges SUBMIT parancs utáni parancsok nem hajtódnak végre.

A következő példa megértéséhez tudni kell, hogy az általános másolóprogram neve PIP (lásd 3.6. pont).

Példa: File-t akarunk menteni egy parancs-lánc segítségével. A parancs-file neve. MENTEES.SUB

tartalma: PIP B: = \$1 [V]
STAT B: \$1\$\$R/O

Ha ezt a parancs-file-t a

SUBMIT MENTES FONTOS.TXT

paranccsal hívjuk, a FONTOS.TXTfile a „B” lemezre másolódik változatlan névvel, ellenőrzéssel, és írásvédett attributumú lesz.

3. FILE-KEZELÉS

Ebben a fejezetben az M08X file-kezelő parancsait tárgyaljuk.

A hajlékony lemez tartalomjegyzékének megjelenítése (DIR)

3.1.

A parancs formája:

DIR nfn < n

A DIR paramétere egy nem egyértelmű file-név. Ha a paraméter elmarad, az éppen aktuális lemezen található file-ok nevei jelennek meg. A < jel mögött álló szám megadása nem kötelező, megadása azt jelzi, hogy hány file-név kerül egy kírásai sorba.

Példa:

A > DIR hatására az „A” tartalmát kapjuk
A > DIR B: hatására a „B” lemez tartalmát kapjuk
A > DIR ALMA.COM az ALMA.COM file-nevet kapjuk, ha van ilyen nevű file az „A” lemezen.

További érvényes parancsok:

DIR B:A?M?*
DIR *.* ugyanaz mint a DIR parancs
DIR *.BAK < 2 a BAK kiterjesztésű file-ok jelennek meg, egy kírásai sorba 2 file-név kerül.

Megjegyzés: ha nem létezik a paraméternek megfelelő file a hajlékony lemezen, NOT FOUND üzenetet kapunk. A DIR parancs nem jeleníti meg a SYS attributumú file-okat (2.4.4. pont).

A CTRL P aktivizálásával listát is kapunk a file-ok neveiről az LST: listázó eszközön.

Szövegszerkesztő program

3.2.

A program az

A > EDIT efn

parancs hatására indul el: a parancsban az egyértelmű nevet kötelező megadni. Ha már van ilyen nevű file az adott lemezen, akkor a file módosításáról, ha még nincs, új file létesítéséről beszélünk. Új file létesítésének tényére a szövegszerkesztő program a „NEW FILE” üzenettel hívja fel a felhasználó figyelmét. Ebben az esetben a szövegszerkesztés megkezdődhet. Ha már meglévő file-t akarunk módosítani vagy bővíteni, akkor a módosítandó file-t a CTRL SHIFT R parancs segítségével olvashatjuk be a tárba.

Kézikönyvünk a 3.2.1. ponttól kezdődően a szövegszerkesztő program parancsait írja le.

Megjegyzés: minden parancsot szükség szerint automatikusan ismételtünk. Az ismétléshez a következőket kell beadni:

CTRL n azol n 1...9999 közötti szám.
A leütött számnak megfelelően ismétli az utána aktivizált parancsot. Például

CTRL 25
A

hatására 25 db A betű szűrődik be (lásd a 3.2.1. pontot) az

aktuális kurzor pozíció helyén, azaz a gép az egyszerű A betű beszúrását jelentő parancsot 25-ször megismétli.

A szövegbevitel alatt az első sor tartalma a bevitt adatokról, a szabad tár nagyságáról ad visszajelzést.

A képernyő első sorában megjelenik:

- az editált file neve,
- az aktuális oszlop száma (kurzor pozíció),
- az utoljára leütött 20 billentyű,
- a még üres betűhelyek száma,
- a teljes szövegen belüli aktuális sorszám,
- esetleges hibaüzenet.

A mindenkori aktuális betű (karakter) pozíciót egy villogó aláhúzás, a továbbiakban „kurzor” jelzi. A szövegszerkesztés folyamán aktivizált parancsok általában ezen kurzorpozícióban hajtódnak végre.

Bármely billentyű, vagy billentyűcsoport lenyomása egy-egy parancsot vált ki. Ezen parancsok közül az ABC betűinek, a számoknak és az írásjeleknek a leütése egy egyszerű parancsot jelent, míg a vezérlési funkciókat, az ún. vezérlő billentyűkkel aktivizálhatjuk. Az összetett parancsok a fenti két billentyűtípus megfelelő kombinációjával állíthatók elő.

Hosszú művelet végrehajtása vagy összetett parancs definiálása a CTRL CR parancssal mindig megszakítható.

3.2.1. Egyszerű parancsok

Az ABC betűi, a számok és az írásjelek közül bármelyiket leütve az illető karakter kurzorpozícióba kerül úgy, hogy az adott pozícióban levő karakter, továbbá a tőle jobbra lévő karakterek előzőleg egy hellyel jobbra tolódnak, majd a leütött karakter a felszabadult üres pozícióban jelenik meg. Egyidejűleg a kurzor egy pozícióval jobbra lép.

Megjegyzés

- A 3.2. pontban a ↑, ↓, →, ← és ↘ nyilakkal jelölt billentyűket rendre CFEL, CLE, CJOB, CBAL és HOME betűcsoportokkal jelöljük.
- A betűváltó szerepét a SHIFT billentyű veszi át. A billentyűzet alapértelmezésben nagybetűs, azaz a kisbetűket a SHIFT egyidejű lenyomása esetén generálja. Ez az alapértelmezés megváltoztatható a CTRL D parancssal. Ekkor az alapértelmezés a kisbetű, míg a SHIFT-tel írhatók a nagybetűk. Minden CTRL D parancs ellentételre változtatja az alapértelmezést.
- A továbbiakban a szövegszerkesztő parancsok ismertetésében az „alapvető” megjegyzéssel arra utalunk, hogy a szóbanforgó parancs ismerete feltétlenül szükséges a szövegszerkesztő használatához. A megjegyzés azonban nem része a parancsoknak.

3.2.2. Kurzor mozgató parancsok

CFEL „alapvető”

A kurzor az előző sor elejére ugrik. Ha a kurzor a képernyő első sorában van, de a szöveg tartalmaz ezen sor előtti, nem látható sorokat, akkor a képernyő sorai egy sorral lejjebb lépnek, és a felszabaduló első sorban jelenik meg a szöveg megelőző sora. A kurzor automatikusan ennek a sornak az elejére ugrik.

| | |
|-----------------|---|
| CLE „alapvető” | A kurzor a következő sor elejére ugrik. Ha a kurzor a képernyő utolsó sorában van, de maga a szöveg tartalmaz ezen sort követő, nem látható sorokat is, akkor a képernyő sorai egy sorral feljebb lépnek, és a felszabaduló utolsó sorban megjelenik a szöveg előbb még nem látott sora. A kurzor ezen sor elejére ugrik. |
| CJOB „alapvető” | A kurzor egy pozícióval jobbra lép. Ha a sor végén volt, akkor a következő sor elejére ugrik. |
| CBAL „alapvető” | A kurzor egy pozícióval balra lép. Ha a sor elején volt, akkor az előző sor végére ugrik. |
| HOME | A kurzor a képernyő első sorának első pozíciójába ugrik. |
| CTRL B | A kurzor a képernyő legutolsó karakterének pozíciójába ugrik. |
| CTRL HOME | A kurzor a teljes szöveg legelső karakterére ugrik úgy, hogy a szöveg eleje kerül a képernyőre. |
| CTRL SHIFT B | A kurzor a teljes szöveg legutolsó karakterére ugrik úgy, hogy a szöveg utolsó 24 sora jelenik meg a képernyőn. |
| CTRL CJOB | A kurzor a kurzort tartalmazó sor végére ugrik. |
| CTRL CBAL | A kurzor a kurzort tartalmazó sor elejére ugrik. |
| CR „alapvető” | A kurzor pozíciójában végrehajt egy soremelést, és a kurzor a következő sor elejére ugrik. A sor vége tehát ott van, ahol a CR parancsot kiadták. Ennek helyéről úgy győződhetünk meg, hogy a kurzort a sor végére pozicionáljuk, például a CTRL CJOB parancssal. |

Törlést kiváltó parancsok

3.2.3.

| | |
|---------------------|--|
| DEL „alapvető” | Törli a kurzor pozíciójában lévő karaktert. |
| EL „alapvető” | Törli a kurzor pozícióját megelőző karaktert. A kurzor egy pozícióval visszalép. |
| CTRL DEL „alapvető” | Törli a kurzor sorába a karaktereket a kurzortól a sor végéig. Ha a kurzort a sor elejére állítjuk, az egész sort törli. |
| CTRL EL | Törli a kurzort tartalmazó sorban a kurzor pozícióját megelőző karaktereket a sor elejéig. A kurzor a sor elejére kerül. |

Nyomtatást vezérlő parancs

3.2.4.

| | |
|--------|--|
| CTRL L | lapdobást előidéző vezérlő karaktert szúr be a szövegbe. |
|--------|--|

Szövegrész másolását vezérlő parancs

3.2.5.

| | |
|--------------|---|
| CTRL ONLINE | Megjelöli a másolandó szövegrész elejét. |
| CTRL OFFLINE | Megjelöli a másolandó szövegrész végét. |
| CTRL PRINT | Beszúrja a megjelölt szövegrészt az aktuális kurzorpozíció helyére. (A szövegrész nem lehet hosszabb 1000 karakternél.) |

Makró parancsok definiálása és hívása

3.2.6.

Nyolc, egyenként 32 karakter hosszú makró parancs definiálására van lehetőség. Ezek a makrók az S, T, U, V, W, X, Y, Z betűk valamelyikével azonosíthatók, amelyeket a továbbiakban „name-ként” jelölünk.

A makró definiálás általános alakja:

CTRL SHIFT „name”
makró törzs (max. 32 karakter)

CTRL SHIFT „name”

A makró hívás általános alakja:

CTRL „name”

Megjegyzés

A makró definíció törzse maximálisan 32 karakter lehet.
A makró törzsében már definiált makró hívása is szerepelhet. Makró definíálás közben le-
ütött billentyűk csak a rendszersorban jelennek meg.
Ha hibázunk makró definíálás közben, akkor a CTRL CR paranccsal hagyjuk abba a de-
finíciót, amelyet azután előírni kell kezdeni.

Példák egyszerűbb makrókra:

| | | |
|--|---|--|
| CTRL SHIFT S CTRL 5 SPACE CLE CTRL SHIFT S | } | Ez a makró beszúr 5 szóközt az aktuális kurzorpozíciótól kezdve, majd a kurzort a következő sor elejére állítja. Előnyösen használható sorok beljebb tolására. |
| CTRL CBAL CTRL 25 CTRL S | } | Ez a négy parancs a fenti makró hívásával 25 sort 5 szóközzel beljebb tol. |

3.2.7. Karakter csoportot kereső parancs

Formája: CTRL F keresett karaktersorozat CTRL F
A második CTRL F beadásakor a parancs végrehajtódik, a kurzor a keresett karaktersorozat elejére áll. A parancsot a program nem tárolja, ismételt meghívása makró felhasználással célszerű.

| | | |
|--|---|--|
| Példák. CTRL F A 2 CTRL F | } | Ennek a parancsnak a hatására a kurzor az aktuális kurzor pozíciót követő A2 karaktercsoport A betűjére ugrik. |
| CTRL SHIFT V CTRL F A L F A CTRL F CTRL 4 DEL B E T | } | Ez a makró megkeresi a kurzor pozícióját követő első ALFA betűcsoportot, azt kitörli, majd beszúrja a BETA betűcsoportot. Hatása tehát az, hogy az ALFA-t BETA-val helyettesíti. |

A
CTRL SHIFT V
CTRL 43
CTRL V

Ez a parancs a szövegben 43 helyen BETA-ra írja át az ALFA-t.

Tabulátor parancsok

3.2.8.

HT A kurzort a következő, már definiált tabulátor pozícióba lépteti.
CTRL HT Az aktuális kurzorpozícióban tabulátor pozíciót definiál.
CTRL _ (aláhúzás) Az aktuális kurzorpozícióban törli a tabulátor definiálását.
CTRL SHIFT _ (aláhúzás) Az összes tabulátor pozíció törlése.

A memória és a lemez közötti adatforgalomra vonatkozó parancsok:

3.2.9.

– olvasás a lemezről:

CTRL R
CTRL SHIFT R

egy sort olvas a lemezről a központi memóriába a teljes file-t beolvassa a lemezről

– írás a lemezre:

CTRL P
CTRL SHIFT P
CTRL ESC

a tárban elhelyezkedő első sort írja fel lemezre a tárban lévő file-t teljes egészében felírja a lemezre, efn a tárban lévő file-t felírja a lemezre efn file-névvel (ugyanaz, mint a CTRL SHIFT P), és az M08X rendszer parancsra várakozás állapotba kerül. (Valójában a szövegszerkesztő program „melegindítást” hajt végre) nagyobb szövegrész törlésére szolgál;

CTRL SHIFT *

ehhez az
ERASE TEXT
ABOVE = A, BELOW = B, SCREEN = S, ALL = E,
RETURN TO M08X = R
üzenet jelenik meg.

„A” válasz hatására a képernyő „fölötti”, tehát a file-ban a képernyőn látható rész előtti adatok törlődnek a tárból.
„B” válasz hatására a képernyő „alatti”, tehát a file-ban a képernyőn látható rész mögötti adatok törlődnek a tárból.
„S” hatására a képernyőn látható adatok törlődnek, és az „R” hatására az M08X parancsra várakozás állapotba jut (melegindítással), és a szerkesztett file nem tárolódik vissza a lemezre. Ezt a tényt a szövegszerkesztő a „NO SAVE RETURN” üzenettel hozza a felhasználó tudomására. „R” választ olyankor kell adni, ha egy file módosítása közben kiderül, hogy mégis az eredeti, tehát a lemezen lévő változatot kell megtartani.

3.3. File-ok nevének megváltoztatása (REN)

A névváltoztató parancs

```
REN efn1 = efn2
```

alakú, aminek hatására az efn2 egyértelmű file-neve efn1 egyértelmű file névre változik. Meghajtó azonosítót mindkettő, vagy az első file név elé tehetünk, ekkor az adott meghajtóban lévő lemezen történik meg a névváltoztatás. Ha mindkét file-név elé meghajtó azonosítót írunk, azoknak meg kell egyezniük. Ha az adott lemezen efn1 nevű file már létezik, FILE EXISTS, ha efn2 nem létezik, NOT FOUND üzenetet kapunk. Írásvédett file esetén a FILE R/OON x: üzenet jelenik meg, ahol x a meghajtó azonosítója. Bármely gomb lenyomásának hatására a rendszer parancsra vár, és a névváltoztatás nem történik meg.

Példa:

- A > REN HELO = JONAPOT parancs hatására az „A” lemezen a JONAPOT file új neve HELO lesz.
- A > REN B:HELO = JONAPOT ugyanaz de a JONAPOT file a „B” lemezen található.
- A > REN HELO.* = JONAPOT.* hibás parancs, a file név nem egyértelmű. A válasz HELO.* = JONAPOT.*?
- A > REN A:ERRE = B:AZT hibás parancs, mert a két file különböző lemezen van. A B:AZT? választ kapjuk.

3.4. File-ok törlése (ERA)

A file-ok törlésére szolgáló parancs

```
ERA nfn < Q
```

alakú.

A parancs az adott lemezről minden olyan file-t töröl, amely a nem egyértelmű file névnek megfelel. Ha bizonyos file-okat a véletlen törlés lehetősége ellen védeni akarunk, a következőket kell tennünk: készítsünk másolatot a file-ról és helyezzük azt írásvédett állapotba. A file SYS attribútuma korlátozott védelmet nyújt, mivel a file törölhető marad (lásd 2.4.4. pont).

Gyakorlatlan felhasználóknak ajánlatos, hogy az ERA parancsot megfontoltan kezeljék, mivel értékes információ veszhet el.

Ha egy lemez megtelt, jó gyakorlat az ERA *.*BAK vagy az ERA *.*PRN parancs kiadása, ami az összes mentési, illetve lista-file-t törli.

A Q paraméter megadásának hatására ellenőrző kérdéseket tesz fel a rendszer. Y válaszra a törlés végbemegy, N válaszra nem törlődik az adott file.

Példa:

- A > ERA NEMKELL.ASM törli az „A” lemezen lévő, adott nevű file-t.
- A > ERA B:NEMKELL.ASM ugyan az „B” lemezre.
- A > ERA B*.* törli az összes file-t a „B” lemezen. Ez veszélyes parancs, ezért az ALL(Y/N)? kérdés jelenik meg, és csak Y válasz esetén történik meg a kívánt művelet.

File-ok megjelenítése (TYPE)

3.5.

A megjelenítésre a

TYPE efn

alakú parancs szolgál. A parancs hatására az efn egyértelmű file névvel megadott file megjelenik a képernyőn, mint listázó eszközön. A listázás gyors művelet, ezért ha szükséges, meg lehet állítani a CTRL S gomb lenyomásával. Egy újabb CTRL S hatására a megjelenítés folytatódik. A CTRL P lenyomásával listát is kérhetünk a file-ról.

Példa:

```
A > TYPE ITTVAGY.ASM a file megjelenik a képernyőn.  
B > TYPE ITTVAGY.ASM ugyanaz, de a file a „B” lemezen van.  
A > TYPE A?B.HUH hibás, mert a file-név nem egyértelmű.  
Válaszul a A?B.HUH? üzenetet kapjuk.
```

A TYPE parancs nagyon jól használható a file-ok nyomtatására. A TYPE felismeri a tabulátor karaktereket, és a tabulátor pozíciókat minden nyolcadik oszlopba helyezi. Ha bonyolult formátumozási lehetőségekre van szükség, az általános másolóprogram (PIP, 3.7. pont) használata ajánlatos.

Tárterület lemezre való írása (SAVE)

3.6.

Tárterület mentésére a

SAVE n efn

alakú parancs szolgál. Az „n” egy decimális egész szám, azt jelenti, hogy annyszor 256 byte íródik a tár 256-os címétől kezdve a lemeze „efn” egyértelmű file-név alatt. Ha a lemezen van már „efn” nevű file, és az nem írásvédett (R/O) attribútumuakkor az felülíródik.

Pl. az

```
A > SAVE 4 MENTETT.COM
```

parancs hatására a tár tartalma a 256-os című az 1279-es címig (256-tól 4x256 hossz) a lemezre íródik „MENTETT.COM” név alatt.

A SAVE beépített parancs, kiadása nem jár programtöltéssel, tehát nem rontja el a 256-os címnél nagyobb című tárterületet.

Általános másolóprogram (PIP)

3.7.

A PIP program alapvető szolgáltatásai:

- lemez-file másolása másik lemez-file-ra és egyes (logikai) I/O eszközökre
- lemez-file-ok összefűzése.

A PIP program ritkábban használt szolgáltatásait paraméterek megadásával válthatjuk ki, amelyeket a parancs végén kell megadni.

- másolás közben a hosszú sorok végeinek levágása (D paraméter)
- listázó eszközre nyomtatás közben sorszámozás (N)
- lapok formátumozása (P)
- megadott karaktorsorozattól kezdődő másolás (O paraméter)
- megadott karaktorsorozatig tartó másolás (S paraméter)

A PIP program különleges szolgáltatásai:

- speciális I/O eszköz nevek, amik az átvitt adatokra, illetve az átvitel módjára vannak hatással. (NUL.; EOF.; INP; stb.)

3.7.1. A PIP program indításának módjai

Az indításnak két módja van, az ún. „parancs” és a „program” módok. A Parancs mód amikor a PIP neve mellett paraméterek is állnak, a program módban csak a PIP nevet kell beadni. Ez utóbbi esetben a PIP program egy * karaktert küld a képernyőre, ami azt jelenti, hogy a felhasználó közvetlenül a PIP programnak adhat parancsot. A két mód között az a különbség, hogy az első esetben – mivel a PIP paraméterei az MOBX rendszer közvetítésével jutnak el a programhoz – minden kis betű nagy betűre változik, a második esetben viszont a parancs változatlanul jut el a PIP programhoz. Ez abban az esetben nem jelent gyakorlati különbséget, ha a paraméterek csak file-névből állnak, ellenben lényeges különbséget jelent a Q és P paraméterek használata esetén. Ha a PIP programot csak a nevével indítjuk, akkor a program indítása és a tényleges másolás megkezdése között lemezcserét hajthatunk végre, ami előnyös lehet.

3.7.2. File-ok másolása

Egyetlen file-t a következő parancs beadásával másolhatunk:

```
PIP efn1 = efn2
```

ahol az efn2 egyértelmű névvel jelölt file másolódik, és az új file neve efn1 egyértelmű név lesz.

Példa:

- A > PIP MENTES.TXT = B:FONTOS.TXT hatására a FONTOS.TXT file a „B” lemezről az „A” lemezre másolódik, és ott MENTES.TXT lesz a neve.
- A > PIP MENTES.TXT = FONTOS.TXT a FONTOS.TXT file is az „A” lemezen van.
- A > PIP A: = B:FONTOS.TXT a „B” lemezről másolódik a file az „A” lemezre, az új név is FONTOS.TXT lesz.
- A > PIP B: = FONTOS.TXT az előző példával azonos, de a másolás iránya fordított.

Ha a cél-lemezen – amelyikre a másolás történik – már van efn1 nevű file, az törlődik a lemezről.

Ha a PIP-et mint programot indítjuk, a fentieket a következő képpen hajtjuk végre:

```
A > PIP
* MENTES.TXT = B:FONTOS.TXT
* MENTES.TXT = FONTOS.TXT
* A: = B:FONTOS.TXT
* B: = FONTOS.TXT
*
A >
```

Megjegyzés: minden PIP-nek szóló parancsot CR-rel kell elküldeni. Ha csak a CR-t nyomjuk le, az MOBX rendszer parancsra várakozik.

Egy lemezen nem lehet két azonos nevű file, tehát

```
A > PIP A: = JAJDEFON.TOS
parancs hibás.
```

Ekkor az INVALID FORMAT:JAJDEFON.TOS üzenetet kapjuk.

Több másolás megadása esetén célszerű a PIP-et mint programot indítani, mert így a másoló program csak egy alkalommal töltődik.

Egy elnevezési konvenció felhasználásával több file-t másolhatunk.

Példa:

A > PIP B: = KIADASOK. * parancs a KIADASOK.JAN KIADASOK.FEB stb. file-okat másolja a „B” lemezre, változatlan file-nevekkel.

A > PIP B: = *.*

és az A > PIP A: = B*.* parancsok az „A” illetve a „B” lemez összes file-jét másolják a másik lemezre, változatlan file-nevekkel.

A file-nevet kicsi vagy nagybetűvel, vagy azok tetszőleges kombinációjával gépelhetjük be.

File-ok összekapcsolása

3.7.3.

Az

A > PIP efn1 = efn2, efn3, . . . , efnk alakú parancs az efn2, . . . efnk nevű file-okat egy file-ba fűzi össze. Az így létrejövő file neve efn1 lesz.

Példa:

A > PIP B:EGYUTT.ASM = ELSO.ASM,B:MASODIK.ASM parancs lefutása után az „A” lemezen lévő ELSO.ASM és a „B” lemezen elhelyezkedő MASODIK.ASM összemásolva EGYUTT.ASM néven a „B” lemezen helyezkedik el.

Tetszőleges számú file-t másolhatunk így össze, de a parancs hossza nem lehet több 255 karakternél. (A 80 karakternél hosszabb parancs beadásához lásd 2.5. pont CTRL E gomb működését). A fenti általános parancs megadásban efn1 megegyezhet efn2-vel, ekkor a PIP az efn1 nevű file mögé fűzi a parancsban felsorolt file-okat.

Példa:

A > PIP B:SZAMLA.EV = B:SZAMLA.EV, SZAMLA.JUN, SZAMLA.JUL parancs a „B” lemez SZAMLA.EV file-át megnöveli az „A” lemez SZAMLA.JUN és SZAMLA.JUL file-jaival.

A fenti másolásokkal nem írhatunk felül írásvédett file-okat, illetve nem másolhatunk SYS attributumú file-okat (lásd 2.4.4. pont). Ha írásvédett file-t akarunk felülírni, DESTINATION IS R/O, DELETE (Y/N)? üzenet jelenik meg. Y hatására felülíródik a file, N hatására NOT DELETED választ kapjuk, és a file nem íródik át. Van azonban olyan paraméter, amivel ezek a megszorítások áthághatóak: a „W” hatására a PIP parancs az írásvédett file-okat is felülírja, az „R” paraméter hatására pedig a SYS file-okat is másolja.

Példa:

Az

A > PIP B: = TARTOZAS.TXT [W] parancs a „B” lemez TARTOZAS.TXT file-ját még akkor is felülírja, ha az írásvédett (R/O) attributumú.

Az

A > PIP A: = B:RENDSZER.COM [R] parancs megtalálja a „B” lemezen a RENDSZER.COM file-t abban az esetben is, ha SYS attributumú.

a W és az R paraméterek együtt is használhatók, pl.

az A > PIP MENTES.TXT = B:FONTOS.TXT [RW] parancs hatására MENTES.TXT minden esetben felülíródik, a FONTOS.TXT átmásolódik akkor is, ha az SYS file.

A beviteli/kiviteli eszközök logikai nevének megadása

3.7.4.

Az MOBX rendszerben négy „logikai” I/O eszköz van, amiket a STAT parancs felhasználásával különböző „fizikai” eszközökhöz rendelhetünk (például 2.6., 2.7. pontok). A logikai

nevek a CON, RDR, PUN és az LST, amelyek sorrendben billentyűzetet és képernyőt, rendszer olvasót, rendszer kírót és rendszer listázó eszközt jelentenek. Az M08X rendszerben az RDR és a PUN soros adatátviteli eszközt jelent, PTR illetve PTP fizikai eszköz hozzárendeléssel.

Példa:

- az A > PIP CON: = B:SZAMLA.TXT parancs a B:SZAMLA.TXT file-t másolja a képernyőre.
- az A > PIP LST: = PROGRAM.ASM [T8] parancs a PROGRAM.ASM file-t nyomtatja ki. A [T8] paraméter a tabulátor pozíciók megadása, később szó lesz róla.
- az A > PIP A:TAVADAT = RDR: beadására a PIP az RDR-ről jövő adatokat az „A” TAVADAT nevű file-jába helyezi.
- az A > PIP PUN: = B:VONALRA.TXT parancs a B:VONALRA.TXT file-t a PUN logikai eszközhöz rendelt fizikai eszközre másolja.

Gyakorlati jelentősége leginkább a listázó eszközre (LST) való másolásnak van, mivel az RDR, illetve PUN eszközökkel speciális programcsomagok végeznek adatforgalmat, a CON eszközre pedig a TYPE paranccsal másolhatunk. Az LST eszközre is lehet a TYPE felhasználásával másolni (TYPE és CTRL P bekapcsolásával), de a PIP parancs még sok olyan tevékenységet is elvégez, amire a TYPE nem képes. Ezzel részletesen foglalkozik a 3.7.6. pont.

3.7.5. Speciális beviteli/kiviteli eszköznevek

Az előző pontban felsorolt logikai I/O neveken kívül a PIP program még néhány különleges I/O eszköz nevet is felismer. Ezeknek a neveknek a megadása többnyire csak a másolás működését befolyásolja.

A továbbiak megértéséhez szükséges még az egyes file-típusokról szólni. A „szöveg-file” olyan adathalmaz, amiben csak karakterekből álló információ fordulhat elő. Ez azt jelenti, hogy a file-ban előforduló byte-ok értékei megadott határok között vannak; az ezen a tartományon kívül eső byte-ok nem adatok, hanem speciális működést váltanak ki a file-t olvasó programból. Az M08X rendszerre a „file vége” karakter van hatással, ez a CTRL Z karakter, ami a szöveg-file végét jelzi. Egy szöveg-file-ben csak egy ilyen byte fordul elő. A 3.7.3. pontban leírt file összekapcsolás úgy működik, hogy a PIP program a „cél” file-ról levásztja a „file vége” byte-ot, és azt a másolandó file-ok hozzáfűzése után így előálló file végére helyezi.

Az M08X rendszerben – mint minden számítógép rendszerben – előfordulnak a fentől eltérő tartalmú file-ok is, olyanok, amelyekben tetszőleges értékű byte is előfordulhat. Az ilyen file végét az jelzi, hogy a file adatai elfogynak. Ilyen pl. a COM típusu file, mivel egy futtatható programban tetszőleges értékű byte előfordulhat. A PIP program – mint látni fogjuk – az egyes vezérlő paramétereire hatására a szöveg file-ba betesz, illetve abból kiszűr vezérlő karaktereket, illetve nem szöveg-file másolása esetén a file vége karaktereket nem veszi figyelembe.

A speciális I/O eszközök tehát a következők:

NUL: hatására a PIP program 40 hexadecimális 0-t küld az output eszköznek.

Példa:

A > PIP PUN: = PGR.HEX,NUL:

EOF: hatására file vége (CTRL Z) karakter kerül az output eszközre.

Példa:

A > PIP B: = A:FILEKELL.HEX,EOF: parancs a „B” lemezre másolja a file-t, és a végére file vége karaktert ír.

PRN: megadására a másolás az LST: logikai eszközre megy azzal a különbséggel, hogy a tabulátor értékek minden nyolcadik oszlopba kerülnek, a lista sorszámozott lesz, és minden hatvanadik sor után új lap kezdődik. A működés azonos a T8NP60 paraméterek hatásával.

Példa:

A > PIP PRN: = FORRAS.ASM parancs sorszámozott listát készít az A:FORRAS.ASM fileről.

INP: különleges adatbeviteli lehetőség, amit assembler programozók használhatnak ki. Hatására a PIP program egy karakter beviteléhez a 103h címre ad hívást, és a 109h címen várja az adatot akkor, amikor a szubrutinból visszatér a vezérlés a PIP programba.

OUT: különleges adatkiviteli eszköz, amit assembler programozók használhatnak ki. Hatására a PIP program egy karakter „kiírásánál” a 106h címet hívja és az átvitelre szánt byte-ot a C regiszterbe helyezi. Egy alkalmas programrész azután tetszőleges műveletet végezhet, és egy szubrutin visszatérési utasítással a PIP-nek adhatja a vezérlést.

Az INP: és az OUT: által nyújtott lehetőségek kihasználásához fontos tudni, hogy a 109h-től az 1ffh címekig a PIP program nem tárol adatot, tehát ez a terület alkalmas a fenti funkciókat kiszolgáló szubrutinok tárolására.

Másolást vezérlő paraméterek

3.7.6.

A másolást vezérlő paramétereket mindig szögletes zárójelbe, a parancs-sor végére kell helyezni.

Példa:

A > B:PIP A: = B:MASOLAND.O [D60FLP40V]

ahol D60, F, L, P40, V az egyes paraméterek. Mint látjuk, az egyes paraméterek közé nem kell tenni elválasztó jelet. A PIP programot a „B” lemeztől indítottuk. Az egyes paraméterek:

„Dn” az n-ik oszlop utáni karaktereket elhagyja az átvitelre kerülő sorból. Előnye, hogy hosszú sorokból álló – pl. lista – file-t is áttekinthetőbb formában jeleníthetjük meg a képernyőn vagy nyomtatón.

Példa:

A > PIP CON:=FORRAS.PRN [D60]

„E” hatására minden átvitt file kiíródik a képernyőre mintha azt a TYPE paranccsal jelenítenénk meg.

Példa:

A > PIP A:=B*.PRN [E]

„F” hatására minden új nyomtatási lap kezdetét kiváltó (form feed) karakter kiszűrődik a másolt file-ból. A „P” paraméter megadásával helyezhetünk új lapdobást vezérlő karaktereket a file-ba (lásd ott).

„H” hexadecimális átvitelt jelez. A PIP program ellenőrzi a helyes formátumot, hiba esetén a billentyűzetről javítások adhatók be.

„I” automatikusan beállítja a „H” paramétert és az átvitel során kihagyja a „00” rekordokat.

„L” hatására a másolás során nagybetűkből kisbetűk lesznek.

Példa:

A > PIP B:KISBETUS.TXT=A:NAGYBETU.TXT [L]

„N” az átvitt file sorait 1-gyel kezdődően egyesével sorszámozza. Ha N2-t adunk meg, a sorszámokban a vezető 0-k is megjelennek. A sorszám után tabulátor karaktert ír, de a tényleges tabulálás – pl. nyomtatás esetén – a „T” paraméter beírásával váltható ki. Az N2 paraméter megadásával 6 jegyen kapjuk a sorszámot.

„O” „nem szöveg-file” átvitel. Hatására a file vége karakterek pontosan úgy kezelődnek, mint a többi karakter (lásd 3.7.5. pont).

„Pn” megadása esetén minden n-ik sor után lapdobást vezérlő karakterek kerülnek a file-ba. Ha az „F” paramétert is használjuk, először szűrődnek ki a lapdobást vezérlő karakterek, azután kerülnek be a „P” paraméter által megadottak.

Példa:

A > PIP LST=>B:JELENTES.TXT [P50] parancs végrehajtásakor egy lapra 50 sor kerül.

„Q. CTRL Z” azt jelenti, hogy a másolás a „Q” és a „CTRL Z” közt beütött karaktersorozat megtalálásáig tart.

Példa:

A > PIP LST=>B:HOSSZU.ASM [Q SUB100:CTRL ZT8] parancs a „B” lemezen lévő HOSSZU.ASM file-t a SUB100: karaktersorozat megtalálásáig nyomtatja. Tabulátor értékek minden nyolcadik pozíción vannak (T8).

Megjegyzés: ha az összehasonlítható sorozatban kisbetűk is előfordulnak, a PIP-et mint programot kell indítani, és a keresendő sorozatot pontosan úgy kell megadni, mint amit a szövegben keresünk.

„S CTRL Z” azt jelenti, hogy a másolás az „S” és a „CTRL Z” közt beütött karaktersorozat megtalálásakor kezdődik.

Példa:

A > PIP B:KIVONAT.TXT = A:JEGYZOKO.NYV [S2.PONTCTRL ZQ5. PONT-CTRL Z] parancs az „A” lemez JEGYZOKO.NYV file-ból a „B” lemez KIVONAT.TXT file-ba másolja a 2. PONT és az 5. PONT közötti szövegrészt. Kisbetűs keresésre lásd Q . . . CTRL Z megjegyzést.

„Tn” minden n-ik oszlopba helyezi a tabulátor pozíciókat. Erre példát a D és Q paraméterek tárgyalásánál láthatunk.

„U” a másolás során kisbetűkből nagybetűk lesznek.

Példa:

A > PIP A:NAGYBETU.TXT = B:KISBETUS.TXT [U]

„V” a másolás helyességét ellenőrzi. Az ellenőrzés időigényes művelet, de értékes forrás- és szövegfile-ok átvitelénél nagyon hasznos.

Itt a „cél-file” (ahová másolunk) lemezfile kell, legyen, nem lehet pl. LST:.

Példa:

A > PIP B: = A:*. * [V] az egész „A” lemez másolódik „B”-re, ellenőrzéssel.

„Z” hatására a PIP program minden input karakterben nullázza a paritás bitet.

Az előbbi paraméterek a H, I, O és a Z kivételével „szöveg” típusú file-ok másolásánál használhatók, a V paraméter használata tetszőleges file esetén előnyös.

A felsorolt lehetőségek ismerete és helyes használata nagyban megkönnyíti a nagyméretű file-ok kezelését.

4. A PROPOS-8 MŰKÖDÉSÉNEK ÁLTALÁNOS ISMERTETÉSE

A 2. és 3. pontban részletesen ismertettük az M08X egyes parancsait és azok működését. A leírás célja volt, hogy az M08X személyi számítógépet számítástechnikai ismeretekkel nem rendelkező személy is kezelni tudja. A 4. és 5. fejezetekben az M08X már eddig megismert működését lehetővé tevő programmal, a PROPOS-8-al foglalkozunk. A PROPOS-8 ismertetésénél feltételezzük, hogy az olvasó járatos a számítástechnikában, és bizonyos programozási ismeretekkel is rendelkezik.

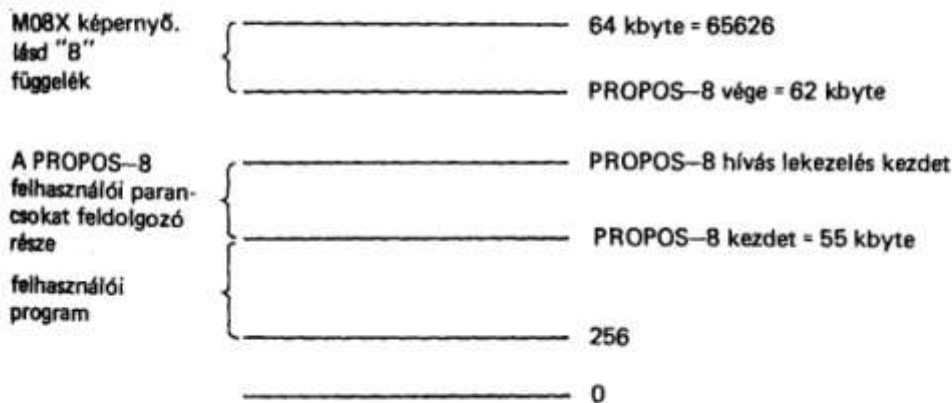
A 4. fejezetben a PROPOS-8 operatív tárban való elhelyezkedését, a rendszer programindítás közben végrehajtott funkcióit és a lemez-file kezelését tárgyaljuk.

A PROPOS-8 kis méretű operációs rendszer, tárfoglalása 7 kbyte (1 kbyte = 1024 byte). A rendszer a „hidegindítás” (lásd 2.1. pont) alkalmával töltődik az operatív tárba. Fontos, hogy ezen alkalommal a teljes operációs rendszer beolvasódik az operatív tárba.

Az operációs rendszer elhelyezkedése az M08X személyi számítógép operatív tárában.

4.1.

A következő ábrán az M08X személyi számítógép operatív tárát ábrázoljuk, alul a „nulla” cím van, felül a 64 kbyte cím.



A 0–255-ig (decimális) tartó címtartományban a rendszer és az egyes felhasználói programok által közösen használt adatterület helyezkedik el. Egy felhasználói parancs által indított program ezen a területen találja meg azokat a paramétereket, amiket a felhasználó megadott; ezt a 4.2. pontban részletesen kifejtjük.

A 256-tól a „PROPOS-8 kezdet”-ig tartó tárterületre egy felhasználó által indított program kerül. A program indítási módját lásd 2.3. pontban.

A program (vagy „parancs”) nevének megadása egy parancs „COM” típusú file betöltését jelenti, amely file pontosan azt az adatsorozatot tartalmazza, ami a 256-tól maximálisan „PROPOS-8 kezdet”-ig tartó tárterületre kerül a betöltés során. Ez a file tehát nem tartalmaz semmilyen elhelyezési vagy más vezérlő információt, amit a töltés során figyelembe kell venni.

A „PROPOS–8 kezdet” és a PROPOS–8 vége között helyezkedik el az operációs rendszer. A PROPOS–8-at a felhasználó és a felhasználói program szempontjából két részre lehet osztani: az egyik a felhasználó parancsait feldolgozó rész, a másik a felhasználó programjai által kiadott hívásokat lekezelő rész. A lényeges különbség kettő között az, hogy a felhasználó parancsait lekezelő részre semmilyen program nem adhatja a vezérlést, így arra programfutas során nincs szükség, normál esetben egy program felül is írja ezt a tárterületet. A felhasználói programok hívásait lekezelő programrészre viszont mindig szükség van, azt semmilyen program nem írhatja felül.

A PROPOS–8 felhasználói parancsait lekezelő programrész a „PROPOS–8 kezdet” és a PROPOS –8 hívás lekezelés kezdet” közötti tárterületen helyezkedik el, míg a felhasználói programok hívásainak lekezelése a „PROPOS–8 hívás lekezelés kezdet” címtől kezdődik. Fontos tudni, hogy ez utóbbi cím a 6-os tárcímen elhelyezkedő szóban van, ennél nagyobb címre egy felhasználói program nem írhat. Természetesen, ha a PROPOS–8-ban a felhasználó parancsait feldolgozó programrészt egy program felülírja, akkor a program befejeződésekor az adott programrészt ismét be kell tölteni a tárbá. Ez a művelet az ún. „melegindítás”, amit a felhasználó is kiválthat a 2.1.2-ben leírt módon; egy program melegindítást úgy válthat ki, hogy a 0 címre adja a vezérlést (pl. egy JMP 0 utasítással), ami által a PROPOS–8 nagy része – így a felhasználói parancsokat feldolgozó programrész is – újra töltődik, és a rendszer „parancsra várakozás” állapotba jut, tehát megjelenik a 2. és a 3. fejezetekben részletesen ismertetett A > vagy B > jel. A melegindítást kiváltó program tehát nem kapja vissza a vezérlést, ugyanis a melegindítás egy program befejeződésének normális módja.

4.2. A PROPOS–8 programindítás közben biztosított funkciói

A program indítási módját a kézikönyv 2.3. pontjában tárgyaljuk. A PROPOS–8 a felhasználói parancs hatására a „parancs.COM” file-t keresi a lemezen, és a 2.3. pontban leírtaknak megfelelően a felhasználó az általa indított programnak file1, illetve file1 file2 paramétert adhat. Ennek a parancs felépítésnek az az értelme, hogy egy program az esetek nagy részében legfeljebb két file-t kezel. Ha több file egyidejű kezelésére van szükség, akkor az általános megoldás szerint egyetlen file név megadása elegendő és az egyes file-okat a rájuk jellemző kiterjesztésekkel azonosítják.

Pl. egy fordító program három file-t kezel, a forrás kód file teljes neve fnév.MAC, a lefordított kód file teljes neve fnév.REL és a lista file teljes neve fnév.PRN lehet.

Ekkor a program elindításához elegendő egy file név megadása kiterjesztés nélkül, ami az előbbi példa szerint három file-t jelent.

Tehát a PROPOS–8 2.3. pontban ismertetett parancs formátuma azt jelenti, hogy ha a PROPOS–8 az indítandó programnév mögött egy karaktersorozatot talál – az előzőtől szóközzel elválasztva –, azt egy file-névnek tekinti. A file-név egy, legfeljebb nyolc karakteres „elsődleges” névből és egy, legfeljebb három karakteres „másodlagos” névből vagy típusból áll (lásd. 2.4. pont). A PROPOS–8 az indítandó programnév mögött álló karaktersorozatokat (legfeljebb kettőt) az előbbi alakra hozza, – tehát, ha pontot talál a karaktersorozatban, az előtte lévő karaktersorozatot kiegészíti nyolc byte hosszúra, a pont mögötti részt három byte-ra, a * karaktert megfelelő számú ?-vel helyettesíti – és az így kapott file-nevet vagy neveket letárolja a 93-as, illetve a 109-es címtől kezdődően, pontosan 11 byte hosszon. Ha a file-név előtt lemezhivatkozás is van (lásd 2.4.3. pont), akkor az „A” lemez esetén 1, a „B” lemez esetén 2, . . . , „D” lemez esetén 4 kerül a 92-es, illetve a 108-as tárcímre az 1., illetve a 2. file-név esetén. Ha a file-név előtt nincs lemezhivatkozás (tehát a feltételezett meghajtó egy file-járól van szó), a 92-es, illetve a 108-as címekre egy 0 byte kerül. A file-nevek másolása közben a PROPOS–8 minden byte legnagyobb, és a 2⁵ helyiértéknek megfelelő biteket 0-ra állítja. Az utóbbi azt jelenti, hogy a PROPOS–8 a kisbetű-

vel megadott nevet nagybetűre alakítja, a nagybetűvel megadott neveket változtatlanul hagyja. Az előbbi műveletek csak akkor jelentenek könnyítést a felhasználói program számára, ha a paraméterek ténylegesen file-nevek.

Természetesen, a felhasználó más paramétereket is adhat az általa indított programnak, mint file-neveket. Ezért a PROPOS-8 a programnév után beadott karaktersorozatot a továbbiakban megadott módon elhelyezi a 128-as címtől kezdődően: a 128-as címre kerül a programnév után beadott karaktersorozat hossza, 1 byte-on. A 0 érték azt jelzi, hogy nem adtak paramétert az indítandó programnak. Ha a 128-as címen lévő byte $\neq 0$ értékű, a 129-es címtől kezdődően a 128-as byte tartalma által megadott hosszon található az a karaktersorozat, amit a felhasználó a programnév után beütött.

Ebben az esetben is végrehajtódik az egyes byte-koⁿ az a konverzió, amit a file-név esetében leírtunk, tehát minden byte legnagyobb, és a 2⁵ helyiértéknek megfelelő bitjei 0-ra állítódnak.

Példa:

```
A > MASOL abc.xb:lmnopqrs.prn
```

parancs hatására a MASOL.COM file töltődik az „A” (feltételezett) lemezről a 256-os címtől kezdődően. A 93-tól 103-ig tartó címekre a következő tartalom kerül: ABCbbbbXbb. A kisbetűkből nagybetűk lettek, a b helyköz karaktert jelöl. A 92-es címre egy 00 byte íródik, mivel az első file név nem tartalmazott lemez azonosítót. A 109-től a 119-ig tartó címekre a következő karaktersorozat töltődik: LMNOPQRSPRN. A kisbetűkből nagybetűk lettek, mivel minden byte 2⁵ helyiértékű bitje nullázódott. A 108-as címre egy 02-t tartalmazó byte kerül, mivel a második file-név a „b” lemezazonosítót tartalmazza.

A 128-as címen lévő tartalma 21 (= 15 h) lesz, mivel ez a MASOL programnév után következő karaktersorozat hossza. A 129-es címtől kezdődően 149-es címig a következő karaktersorozat található a MASOL program indítása után:

```
bABC.XbB:LMNOPORS.PRN.
```

A kisbetűkből nagybetűk lettek, a „b” a helyköz karaktert jelöli. Látható, hogy ha a 128-as byte értéke $\neq 0$ (tehát a program kapott paramétereket), a 129-es címre mindig helyköz karakter kerül.

Tekintsük a következő parancsot:

```
A > K11R *.prn
```

A PROPOS-8 a 93-tól a 103-ig tartó címekre a következő karakter sorozatot helyezi el: ???????PRN. Ez a parancsban megadott nem egyértelmű file-név (lásd 2.4.2. pont). A 92-es címre egy 00 byte kerül, mert a file-névben nem volt lemez hivatkozás. A 128-as címre 06 érték tárolódik, mivel az a K11R program utáni karaktersorozat hossza. A 129-es címtől a 134-es címig a következő karaktersorozatot helyezi el a rendszer: b*.PRN.

Látható, hogy az operációs rendszer a felhasználói program számára megkönnyíti a kapott paraméterek file-névként való értelmezését, de az elindított program elérheti a paramétereket olyan formában is, ahogy azt a felhasználó megadta.

A PROPOS-8 egy betöltött programot mindig úgy indít el, hogy a 256-os címre ad ki egy „call” utasítást. Az író/olvasó terület — az a terület, ami file írás vagy olvasás esetén az frott/olvasott adatokat tárolja (lásd 5. pont) — kezdőcíme 128.

4.3. A file-kezelés általános jellemzői

A 2.4-es pontban részletesen leírtuk, hogy egy parancsban milyen módon lehet az egyes file-okra hivatkozni. A file-név megadásában szembevetendő jellegzetesség, hogy a file-név nem egyértelmű is lehet, továbbá, hogy a file-név megadásánál lemez-hivatkozást is meg lehet adni. A lemez-hivatkozás hiánya azt jelenti, hogy az adott file a feltételezett meghajtóban lévő lemezre vonatkozik. Ebben a pontban azt írjuk le, hogy milyen módon lehet egy felhasználói programban file-okra hivatkozni.

Egy programban minden használandó file-t egy file leíró tábla (FLT) reprezentál. A file leíró tábla létrehozását és kezelését a magasszintű nyelvek fordítóprogramja vagy interpretere eltakarja a felhasználói program írója elől. A kezelés a szokásos file műveleteken keresztül valósul meg, leírásuk az egyes nyelvek leírásának része. Az FLT 33 vagy 36 byte hosszú lehet, attól függően, hogy „soros” vagy „direkt” file-kezelést végez a program (lásd később).

A következőkben az FLT beosztását írjuk le. A táblázat bal oldalán megadott számok a tárcímeket jelentik, amiket a file leíró tábla kezdőcíme és a számok összege ad; tehát a 9-es szám az FLT kezdőcím + 9 tárcímet jelzi.

FLT

| 0 | 1 | 11 | 12 | 31 | 32 | 33 | 34 | 35 |
|--------------------|----------|----------|---------------------------------|----------|----|---|----|--------------|
| meghajtó azonosító | file név | | rendszer által használt terület | | | rekord szám | | 00 kezdetben |
| | | | | | | csak direkt file művelet esetén szükséges | | |

- 0 meghajtó azonosító, értéke 0 és 4 között lehet. 0 jelenti, hogy a FLT-ben megadott file a feltételezett meghajtón van.
- 1 jelenti, hogy a file az „A” meghajtón van,
- .
- .
- 4 jelenti, hogy a file a „D” meghajtón van.

Megjegyzés: Az MOBX hardware konfigurátora max. 4 meghajtót enged meg.

- 1–8 A file elsődleges neve; minden byte 2^5 helyiértékének megfelelő bit-je = 0. Ez azt jelenti, hogy a file-névben csak nagybetűk szerepelnek. A file-nevet kiadó byte-ok legnagyobb helyiértékű bit-jei nem kötelezően nullák az FLT-ben, de ezek a bitek nem részei a file-névnek; tehát a rendszer összehasonlítási műveleteiben – amikor a rendszer egy file-t megnyitás, lezárás, névváltoztatás, törlés stb. miatt a lemezen név szerint keres – ezek a bitek nem vesznek részt az összehasonlításban.
- 9–11 A file-név kiterjesztése, vagy típusa, minden byte 2^5 helyiértékű bit-je = 0. Ez azt jelenti, hogy a kiterjesztésben csak nagybetűk szerepelnek. A byte-ok legnagyobb helyiértékű bit-je nem kötelezően nulla az FLT-ben, de ezek a bitek nem részei a file-név kiterjesztésének (lásd a file-név alatt tett megjegyzést). A PROPOS–8 használja a 9. és a 10. byte legnagyobb helyiértékű bit-jeit.

A 9. byte legnagyobb helyiértékű bitje

- = 0, az adott file R/W
- = 1, az adott file R/O attributumú (lásd 2.4.4. pont)

- A 10. byte legnagyobb helyi értékű bitje
= 0, az adott file DIR
= 1, az adott file SYS attributumú (lásd 2.4.4. pont).

- 12 „Extent” szám: file megnyitás előtt 0-ra kell állítani.
- 13–31 Az FLT azon része, amit a felhasználói program nem ír vagy olvas a file műveletek közben, mert azt a PROPOS–8 használja.
- 32 Rekordszám; a file megnyitás után 0-ra kell állítani, ha soros feldolgozást végez a program (lásd 5. pont).
- 33–34 Ez a terület csak akkor tartozik az FLT-hez, ha direkt írást vagy olvasást végzünk (lásd később). Tartalma az olvasandó vagy írandó rekord száma úgy elhelyezve, hogy a kisebb helyi értéket adó byte a 33-as, a nagyobb helyiértéket adó byte a 34-es címre kerül. A rekordszám 0 és 65535 közé eshet.
- 35 Ez a byte csak akkor tartozik a FLT-hez, ha direkt írást vagy olvasást végzünk (lásd később). A felhasználói program ezt a byte-ot nem használja, de a tartalmát az első file művelet előtt 00-ra kell állítani.

A PROPOS–8 – 4.2. pontban tárgyalt – programindítás közben végrehajtott funkcióinak egy része abból áll, hogy az 1. file-név alapján kitölti a 92-es címen kezdődő FLT első 12 byte-ját, a 2. file-név alapján pedig a 108-as címen kezdődő FLT első 12 byte-ját. Ha egy program mindkét file-t használja, fontos, hogy az első file-művelet előtt a 108-as címen kezdődő 12 karaktert egy szabad helyre másolja. Erre azért kell ügyelni, mert az első file-ra vonatkozó FLT 92-n kezdődik, és a legalább 33 byte-os FLT felülírja a 108-cal kezdődő területet.

Egy felhasználói program minden file-műveletéhez szükséges az adott file-t leíró FLT. A file létrehozására, megnyitására és lezárására vonatkozó műveletek mellett az adatforgalomra vonatkozó műveletek ismerete nélkülözhetetlen a programozó számára, ezért ezt tárgyaljuk a továbbiakban.

Egy file rekordokból épül fel. Minden rekord pontosan 128 byte hosszú. A file-t felépítő rekordokat egymás után lehet írni vagy olvasni (soros feldolgozás) vagy közvetlen módon is lehet írni vagy olvasni (direkt elérés). A közvetlen elérés azt jelenti, hogy olvasás esetén a file-ből a megadott sorszámú rekordot kapjuk (ha van ilyen), írás esetén a 128 byte-os rekord a file-on belül a megadott sorszámú rekord helyére kerül, ha van hely a lemezen. Közvetlen elérés esetén a sorszámot a FLT 33–34 byte-jában kell elhelyezni olvasás illetve írás kérelem előtt.

Megjegyzés: A file legelső rekordjának sorszáma 0. Egy PROPOS–8 file felépítése nem függ attól, hogy azt szekvenciális vagy direkt módon dolgozzák fel. A file létrehozás és megnyitás műveletben nem kell a feldolgozás módjára vagy a file méretére vonatkozóan információkat adni.

Ha egy felhasználói program egy file művelet végrehajtását kéri a PROPOS–8-tól, a következő információkat kell szolgáltatnia:

- FLT címe
- az elvégzendő műveletet azonosító műveleti kód
- file írás vagy olvasás esetén egy 128 byte hosszú input/output terület címe.

Az input/output terület megadására speciális PROPOS–8 hívás áll rendelkezésre. Az egyszerű megadott I/O terület címe a következő hívásig változatlan, tehát az I/O terület címét nem kell minden írás/olvasás művelet alkalmával megadni.

5. A PROPOS-8 HIVÁSAINAK VÁZLATOS ISMERTETÉSE

PROPOS-8 hívások alatt azt értjük, hogy egy felhasználói program valamilyen kérelemmel fordul az operációs rendszerhez, és a kérelem kiszolgálása után a hívó program visszakapja a vezérlést. A PROPOS-8-at egységes hívási konvenció alapján lehet elérni. Minden hívásra jellemző, hogy

- az 5-ös tárcímre kell egy szubrutin hívás (call) utasítást kiadni,
- a kérelmet a „C” regiszterben tárolt értékkel kell azonosítani,
- ha a hívás tárterületre hivatkozik, annak címe a „D, E” regiszter párban van,
- ha a hívás eredménye egy byte-ban tárolható, akkor azt az „A” regiszterben adja vissza a rendszer. A funkció sikeres vagy sikertelen végrehajtását is az „A” regiszterben jelzi vissza a rendszer,
- ha a hívás eredménye két byte-ban tárolható, akkor azt a „H, L” regiszter párban adja vissza a rendszer.

A PROPOS-8 hívások túlnyomó többsége írás/olvasás kérelmet valósít meg. Közös jellemzőjük, hogy a hívó felhasználói program a kérelem kiszolgálásának befejeződéséig nem kapja vissza a vezérlést, tehát a program mindig vár a PROPOS-8-ra.

A PROPOS-8 hívásokat három kategóriába sorolhatjuk.

Az első csoportba helyezzük azokat a hívásokat, amelyek a billentyűzetre és a képernyőre vonatkozó funkciókat valósítják meg (a bal oldalon lévő számok a hívás azonosító számai):

- 1: karakter olvasás billentyűzetről
- 2: karakter írása a képernyőn
- 5: karakter írása a lista eszközre
- 6: karakter olvasás közvetlenül a billentyűzetről és/vagy írás képernyőre
- 9: karakter sorozat írása a lista eszközre
- 10: parancs olvasása billentyűzetről

A fenti hívások részletes leírását lásd az „A” függelékben.

A második csoportba helyezzük azokat a funkciókat, amelyek alapvető fontosságú lemezre vonatkozó műveleteket hajtanak végre. Természetesen, az mindig az adott feladattól függ, hogy melyik funkció „alapvető” fontosságú. A következő felsorolás a leggyakrabban használt funkciókat adja meg (a bal oldalon használt számok itt is a hívás azonosító számai):

- 22: file létrehozás
- 15: file megnyitása
- 16: file lezárása
- 33, 20: közvetlen vagy soros olvasás
- 34, 21: közvetlen vagy soros írás
- 26: írási/olvasási cím beállítás

(A részletes leírást lásd az „A” függelékben.)

Az utolsó hívás nem tartalmaz tényleges írást vagy olvasást a lemezre vonatkozóan, jelenleg mégis a lemez írás/olvasás esetén alapvető, mert az ezen funkció során beállított címtől kezdődően helyezkedik el az írási/olvasási munkaterület 128 byte hosszon.

Fontos tulajdonság, hogy a „file létrehozás” esetében nem kell megadni a file várható feldolgozási módját, vagy a file maximális méretét. A „file megnyitása” hívásnál nem kell megadni, hogy a file-t írni vagy olvasni akarjuk-e stb.

Az előbbi lemez-file hívások felhasználásával már komplikáltabb file-kezelést végző programokat is lehet írni.

A harmadik csoportba az olyan, lemezre vonatkozó hívásokat soroltuk, amelyeket az egyszerűbb programok várhatóan nem használnak.

- 13: rendszer alapállapotba hozás
- 14: feltételezett meghajtó beállítása
- 17, 18: adott file keresése
- 19: file törlés
- 23: file névváltoztatás
- 24: rendelkezésre álló lemezek lekérdezése
- 25: feltételezett meghajtó lekérdezése
- 30: file attributumok beállítása
- 35: file méret lekérdezés
- 36: direkt rekord beállítás.

A továbbiakban vázlatosan ismertetjük a fenti hívások működését. Az ismertetés célja, hogy az olvasó képet kapjon a file-kezelés áttekinthető és használható voltáról.

A hívások elérhetők assembler, Pascal vagy C nyelvekből.

A FORTRAN és a BASIC nyelvekből nem érhetőek el a hívások, mivel ezek a nyelvek olyan PROPOS-8-tól független szolgáltatásokat nyújtanak, amelyek a nyelvre, nem pedig az operációs rendszerre jellemzőek. A PROPOS-8 szolgáltatások közvetlenül is elérhetőek egy, a magas szintű nyelvből hívott assembler szintű programból.

A „rendszer alapállapotba hozás”

nak parancs szinten megfelelője a „melegindítás” (lásd 2.1. pont), mivel a „rendszer alapállapotba hozás” felhívás után minden lemez írható lesz, még akkor is, ha azt az előző melegindítást követően cserélték.

A „feltételezett meghajtó beállítása,”

hívás a 2.2. pontban ismertetett lemezcserére vonatkozó parancsnak (a: illetve b:) programból kiadott megfelelője. Ezzel a hívással egy meghajtó nevet közölhetünk a rendszerrel. Az így beállított meghajtóra vonatkoznak a továbbiakban azon lemez- file kezelő hívások, amikhez tartozó FLT-ben (lásd 4.3. pont) a lemez azonosító byte (byte 0) értéke 00.

Az „adott file keresése” hívás bemenő adata egy file leíró bla. A hívás segítségével az FLT-ben megadott file-t lehet keresni. Ha az FLT-ben megadott file-név nem egyértelmű, több hívás segítségével (egy 17-es, több 18-as) az összes adott nevű file-t megkereshetjük. A PROPOS-8 visszajelzi, ha egyáltalán nincs, vagy nincs több adott nevű file.

A „file törlés” felhívás segítségével programból törölhetünk adott nevű file-t egy adott lemezen. Ennek a hívásnak a használata azért nem gyakori, mert a file-törlést általában az „ERA” parancs felhasználásával szokás végezni (lásd 3.4. pont). Azonban egy szövegkezelő program az új file előállítás után a régit törölheti, és az új file-t a törölt file nevére nevezheti át (lásd a következő funkciót).

A „file névváltoztatás”

hívás működése a nevében benne van. Az egyszerűbb programok mégsem használják ezt a funkciót, mert erre parancs-szinten is adott kényelmes lehetőség (lásd „REN” parancs 3.3. pont).

A „rendelkezésre álló lemezek lekérdezése”

hívásra mindazon meghajtókat kapjuk válaszul, amelyekkel az előző „melegindítás” parancs vagy „rendszer alapállapotba hozás” hívástól számítva a rendszer valamilyen módon adatforgalmat hajtott végre. A válasz ilyenkor egy két byte-os érték (a hívási konvencióknak megfelelően a H,L regiszter párban), amelyben az egyes bitek az egyes meghajtókhoz vannak rendelve, és az „1” érték azt jelenti, hogy az illető meghajtót már használta a rendszer. Itt a legmagasabb helyiértékű bit az „A” meghajtót jelöli.

A „feltételezett meghajtó lekérdezés”

hívásra a feltételezett meghajtó azonosítóját kapjuk válaszul. Ez az a meghajtó, amire akkor történik hivatkozás, amikor az FLT-ben a 0. byte értéke 00. (lásd 4.3. pont).

A „file attributumok beállítása”

hívással egy adott file attributumait (lásd 2.4.4. pont) programból állíthatjuk be. Egy file „DIR” vagy „SYS” illetve „R/O” vagy „R/W” attributumú lehet.

A „file méret lekérdezése”

hívás segítségével egy adott file méretét kérdezhetjük le. Egy file méretét a file-ba írt legnagyobb rekordszám határozza meg. Soros feldolgozás esetén ez a szám a ténylegesen felírt rekordszámmal egyenlő. Direkt felírás esetén „lyukakat” is lehet hagyni, ekkor a file méretét nem a file-ba írt rekordok száma, hanem a beleírt legnagyobb sorszámú rekord határozza meg.

A „közvetlen rekord beállítás”

hívást egy file soros feldolgozása közben lehet hívni. A hívás arra ad választ, hogy direkt feldolgozás esetén mi az utoljára feldolgozott rekord sorszáma. A hívás akkor hasznos, ha soros feldolgozás közben meg akarunk jelölni egy rekordot, amire később közvetlen feldolgozás alatt vissza akarunk térni. Ez a részfeladat gyakran fordul elő információ-visszakereső rendszerek létrehozása közben.

Megjegyzés: a PROPOS—8 hívások előbbi csoportosítása önkényes és nem teljes. A rövid ismertetéssel be akartuk mutatni, hogy a PROPOS—8-nak egy felhasználói program számára is kevés számú és jól meghatározott szolgáltatásai vannak. Látjuk, hogy a PROPOS—8-ban a felhasználói program is el tud végezni egy sor olyan feladatot, amit a gép kezelője egy adott parancs segítségével végezhet el. Ilyen a feltételezett meghajtó beállítása, a file névváltoztatása, törlése, vagy egy file attributumainak beállítási lehetőségére vonatkozó parancsok, illetve hívások.

Egy, a PROPOS—8 alatt megvalósított magasszintű programozási nyelvnek lehetnek olyan jellegű szolgáltatásai, amelyek egyébként általános operációs rendszer szolgáltatások, a PROPOS—8 azonban nem nyújtja ezeket. Ilyen szolgáltatásokra példa az overlay struktúra lehetősége, egyfajta korlátozott multiprogramozási lehetőség, az egyes programok közötti szinkronizáció vagy kommunikáció lehetősége.

A PROPOS—8 jelentősége abban áll, hogy egy egyszerű parancskészlet és funkcióhalmaz segítségével lehetőséget nyújt a legnehezebb kezelési és programozási feladatok megoldására is.

FÜGGELEK

„A” függelék

A PROPOS–8 1, 2, 5, 6, 9–10, 15, 16, 20–22, 26, 33 és 34-es hívásainak ismertetése.

A PROPOS–8 hívási konvenciókat vázlatosan az 5. pontban ismertettük.

Ismertetésünkben a bal oldalon a hívás számát adjuk meg, „be:” után a hívás bemenő adatai következnek, „ki:” után a visszatérési információt ismertetjük.

Az A, C, D, E betűk a nekik megfelelő regisztereket jelentik.

1. Karakter olvasása billentyűzetről
be: C = 1
ki: A = ASCII karakter
A 2.5.-ben megadott speciális karakterek egy részének beolvasásánál a rendszer a nekik megfelelő funkciókat végrehajtja.
Ezek a speciális karakterek az EL, CR, HT, CTRL P és CTRL S.
2. Karakter írása képernyőre
Be: C = 2
E = ASCII karakter
ki: –
Speciális karakterek esetén lásd az 1. hívás magyarázatot. A kiírás az aktuális kurzor pozícióba történik, a kurzor tovább lép.
5. Karakter írása lista eszközre.
be: C = 5
E = ASCII karakter
ki: –
A kiírás az LST: logikai eszközre megy végbe (lásd 2.6. pont).
6. Karakter olvasás közvetlenül a billentyűzetről, és/vagy annak a képernyőre való írása
be: C = 6
E = 255 esetén beolvasás, illetve
E = ASCII karakter esetén kiírás
ki: A = ASCII karakter, illetve
A = 0.
A visszatérő érték csak beolvasás esetén adott, karakter kiírás esetén ilyen nincs.
Ha E = 255, akkor karakter olvasás megy végbe; az 1-es híváshoz viszonyítva a következő különbségek állnak fenn:
ha a felhasználó már beütött egy karaktert, ezt visszaadja a hívás az „A”-ban.
Ha nincs beadott karakter, az A = 0 értéket kapja a hívó program;
az 1-es hívásban említett speciális karakterek is a többi karakterrel azonosan kezelődnek, tehát a rendszer nem hajtja végre a nekik megfelelő funkciókat.
Ha E = 255, akkor a karakter képernyőre való írása következik be, kimenő érték nincs.
9. Karakter sorozat írása képernyőre
be: C = 9
D, E = kinyomtatandó karakter sorozat címe
ki: –
A kiírás a karakter sorozatban talált „\$” jelig tart.
A 2.5.-ben megadott CTRL P, CTRL S és HT speciális karaktereknek megfelelő funkciók végrehajtódnak a kiírás közben.

10. Parancs olvasása billentyűzetről

be: C = 10

D, E = input terület címe, ahova a beolvasott parancsot kéri a program

ki: input terület töltődik.

Az input terület felépítése:

DE: +0+1+2 +n
h sz k₁ k_{sz}

ahol h tartalmazza a puffer hosszát,
sz a beadott parancs hossza,
k₁ k_{sz} a tényleges beadott parancs.

A parancs beadása akkor ér véget, ha a felhasználó CR-et ad be, vagy a parancs hossza nagyobb mint h. A parancs beadása során a beütött EL, CTRL U, CTRL X, CTRL R, CTRL E, CTRL P, CTRL S és a HT a nekik megfelelő funkciókat váltják ki (ld. 2.5. pont). A CTRL C (melegindítás) csak akkor hajtódik végre, ha elsőként adták be, egyéb esetekben az input területbe kerül. Figyeljük meg, hogy az olvasó terület felépítése hasonló a 4.2. pontban tárgyalt paraméter-terület (128-tól 255-ig tartó címtartomány) felépítéséhez.

15. File megnyitása

be: C = 15

D, E = FLT cím

ki: A = hibakód

Az FLT-t a 4.3.-ban leírtaknak megfelelően kell kitölteni. Felhívjuk a figyelmet az FLT 12-es és 32-es byte-jának nullázására. Az FLT „file-név” részében szerepelhet nem egyértelmű file-név is, ekkor az első, a file névnek megfelelő file-t nyitja meg a rendszer. A = 255 esetén sikertelen a file megnyitása, ha nem létezik az adott nevű file, vagy esetleg a 12-es byte ≠ 0, különben a file megnyitása sikeresen végbement.

16. File lezárása

be: C = 16

D, E = FLT cím

ki: A = hibakód

Megnyitott vagy létrehozott (ld. 15. illetve 22. hívást) file lezárása. Ha a programban írtunk a file-ra, ajánlatos a file-t lezárni, mielőtt a futás befejeződik, mert a felírt adatok a lezárás alkalmával kapcsolódnak a file-hoz. Ha csak olvasunk a file-ból, nem fontos azt lezárni.

A = 255 esetén a file zárása sikertelen (ha nem létezik az adott nevű file), ellenkező esetben a hívás sikeresen végrehajtódott.

20. Soros olvasás

be: C = 20

D, E = FLT cím

ki: A = hibakód

A következő 128 byte-os rekord olvasása az input területre.

Az input terület elhelyezkedését illetően ld. 4.3. pontot és a 26-os hívást.

Ha egy file-t az elejétől kezdve akarunk sorosan olvasni, akkor az FLT 12. byte-ját = 0-ra kell állítanunk a file nyitása előtt, és a 32. byte-ját 0-ra kell állítani vagy a file

nyitás előtt, vagy utána, de az első olvasás előtt.

A = 0, ha az olvasás sikeres volt, különben az olvasás sikertelen ha az adott file nincs megnyitva.

21. Soros írás

be: C = 21

D, E = FLT cím

ki: A = hibakód

A következő 128 byte-os rekord írása az output területről.

A továbbiakban azonos a 20-as hívás működésével.

22. File létrehozás

be: C = 22

D, E = FLT cím

ki: A = hibakód

A működés a „file megnyitása” (15-ös hívás) működéséhez hasonló, de itt az FLT-ben megadott nevű file jön létre a lemezen. Fontos tudni, hogy a hívás nem vizsgálja van-e az adott nevű file a lemezen. Ha van ilyen file, egy újabb azonos nevű file lesz az adott lemezen. Ha a létrehozott file-ba írni akarunk, nem kell újra megnyitni, mivel a file létrehozás hívás a file megnyitása funkcióit magában foglalja.

A = 255 hibás működés esetén: ellenkező esetben a hívás hibátlanul lefutott.

26. Írás/olvasás cím beállítás

be: C = 26

D, E = írási/olvasási terület kezdőcíme

ki: –

A file írási/olvasási műveletek 128 byte-os blokkok írását, illetve olvasását jelentik. Ezek a blokkok a 128 byte hosszúságú írási/olvasási területre, illetve területről olvasódnak.

Ennek a területnek a kezdőcímét állítja be a 26-os hívás. A terület címe a következő 26-os hívásig, a következő melegindításig, illetve a következő 13-as hívásig változatlan. Az utóbbi két esetben az írási/olvasási cím 128 lesz. Az írási/olvasási területet illetően lásd még 4.3. pontot.

3. Közvetlen olvasás

be: C = 33

D, E = FLT cím

ki: A = hibakód

Közvetlen olvasás esetén az FLT 36 byte hosszú (lásd 4.3. pont). A megnyitott file-t úgy olvashatjuk direkt módon, hogy az olvasandó rekord számot az FLT 33, 34-es byte-jaiba töltjük oly módon, hogy a nagyobb helyiértékű byte a 34-es, a kisebb helyiértékű byte a 33-as helyre kerüljön. A 35-ös byte-ot az első írás előtt 0-ra kell állítani.

Minden direkt olvasás előtt állítani kell az FLT 33-as és 34-es byte-jait, mert a rendszer ezeket nem állítja. Közvetlen olvasásról áttérhetünk soros olvasásra, ekkor a direkt módon utolsónak beolvasott rekordot az első soros olvasás ismét beolvassa, és csak a következő soros olvasáskor kapjuk a következő rekordot.

Például ha a 189-ik rekordot olvastuk direkt módon, akkor az azután következő so-

ros olvasás ismét a 189-ik rekordot olvassa, és csak a következő soros olvasás olvassa a 190-iket.

Soros olvasásról bármikor áttérhetünk direkt olvasásra a 33-as hívás felhasználásával.

A = 0 sikeres olvasást jelent,

A = 1 v. 4 azt jelenti, hogy olyan rekordot akarunk olvasni, amit nem írtunk fel,

A = 3 esetén ismételjük meg az olvasást, és ha nem sikerül helyesen olvasni, vizsgáljuk meg, hogy a lemez nem írásvédett-e, ugyanis ha a lemez írásvédett állapotban van, és a file-t nem csak olvassuk, akkor felléphet ez a hiba.

34. Közvetlen írás

be: C = 34

D, E = FLT cím

ki: A = hibakód

Közvetlen írás esetén az FLT 36 byte hosszú (lásd 4.3. pont). A megnyitott file-t úgy írhatjuk direkt módon, hogy az írandó rekord számot az FLT 33, 34-es byte-jaiba töltjük oly módon, hogy a nagyobb helyiértékű byte a 34-es, a kisebb helyiértékű byte a 33-as helyre kerüljön. A 35-ös byte-ot az első írás előtt 0-ra kell állítani.

Minden direkt írás művelet előtt állítani kell az FLT 33-as és 34-es byte-jait, mert a rendszer ezeket nem állítja. Közvetlen írásról áttérhetünk soros írásra, ekkor a direkt módon utolsónak kiírt rekord az első soros írásnál ismét felülíródik. A következő soros írás kérelem azonban már a következő rekordba íródik a file-on.

A = 0 sikeres írást jelent,

A ≠ 0 hibás futást jelent, pl. mert a lemez írásvédett, vagy megtelt.

„B” függelék

Speciális képernyő kezelés

A PROPOS-8 a képernyőt a legegyszerűbb módon kezeli, azaz csak „ROLL” módusban működik a képernyő. Ennek következtében az „A” függelékben felsorolt 1, 2, 6 és 10-es hívások csak az aktuális kurzor pozícióban hajtódnak végre, és a kurzor csak egyesével jobbra léphet, illetve CR esetén a következő sor elejére.

Kurzor mozgatásra a 2.5.-ben megadott speciális billentyűk segítségével is mód van, ennél rugalmasabb kezelésre azonban nincs lehetőség.

Ebben a függelékben megadjuk a kurzor mozgatásának módját, valamint példát adunk olyan képernyő kezelésére, ami az aktuális kurzor pozícióban hajtja végre az egyes író/olvasó műveleteket. Ez utóbbi a képernyő-kezelő programcsomagok nélkülözhetetlen része. Ezek a lehetőségek természetesen bármely felhasználói programban kihasználhatók, így néhány standard alkalmazói program él is velük.

Az MO8X képernyőjének időről-időre „felfrissítésre” van szüksége; ez azt jelenti, hogy annak tartalmát ismételtelen meg kell jeleníteni, mielőtt a képernyő elhalványulna. Ezt az ismételt megjelenítést a rendszer végzi, a felhasználónak semmit sem kell tennie, hogy ez végbe menjen. Fontos azonban az a memória-cím, ahonnan az előbb leírt „felfrissítés” kezdődik. Ami ugyanis a felfrissítési tartományon van, az mind megjelenik a képernyőn, és nem szükséges semmilyen kiviteli (OUT) műveletet végrehajtani. A képernyő felfrissítési memória kezdőcíme 63536, mert a 64 kbyte nagyságú memória utolsó 2000 byte-ja van fenntartva a 80x25 karakter nagyságú képernyő felfrissítésére.

Tehát például egy 65 (ASCII „A”) a 63536-os címen a képernyő bal felső, a 65535-ös pozícióba helyezve a képernyő jobb alsó pozíciójában jelenik meg.

Az MO8X képernyőjén az adatokat bizonyos attributumokkal lehet megjeleníteni: ezek az „aláhúzott”, „fordított”, „villogó”, vagy „világos” megjelenítési módok a „normál” – tehát sötét alapon világos – mellett. A fenti attributumokat a képernyő felfrissítési területén elhelyezett attributum-byte-ok elhelyezésével állíthatjuk be. Az attributum-byte felépítése a következő: 10UR00BH, ahol

| | |
|-------|--------------------------------|
| U = 1 | „aláhúzott”, |
| R = 1 | „fordított”, |
| B = 1 | „villogó” és |
| H = 1 | „világos” attributumot jelent. |

Az attributum-byte hatása arra az adatmezőre érvényes, ami a felfrissítési területen tőle jobbra áll (tehát a képernyőn jobbra illetve alatta) és egy újabb attributum-byte megjelenéséig tart, vagy a felfrissítési tartomány végéig (a képernyő jobb alsó pozíciójáig).

Az egyes attributumokat egymással kombinálhatjuk. Ha minden attributum bit az attributum byte-ban = 0, akkor a „normál” attributum érvényes. Ekkor az attributum-byte értéke 128.

Példa: a képernyő legfelső sorának az attributumát a következő programrészrel állíthatjuk be „villogó” és „fordított” attributumra (INTEL Assembler nyelven írva):

```
MVI   A,146 ; = 10010010      attr. byte
STA   63536
```

```

MVI    A,128    ; = 10000000    attr. byte, normál
STA    63615    ; = 63536 + 79

```

Mivel az attributum-byte-ok is helyet foglalnak a képernyőn, nem tudunk a képernyőre különböző attributumú szövegeket szomszédos karakterpozícióban megjeleníteni.

A kurzor pozícióját úgy állíthatjuk, hogy a 17-es input/output (port) címre 128-at írunk, amivel a kurzor pozicionálásának szándékát jelezzük az M08X képernyő kezelő vezérlő áramkörével. Ez után a 16-os input/output (port) címre a kurzor pozíció oszlop értékét (0–79), majd ismét a 16-os input/output (port) címre a kurzor pozíció sorának az értékét (0–24) adjuk ki.

Példa: az első sor végére a következő programrészsel állíthatjuk a kurzort (INTEL Ass. nyelven írva):

```

.
.
.
MVI    A,128
OUT    17        ; Parancs mód beállítása
MVI    A, 79    ; sor vége
OUT    16
MVI    A, 0     ; a legfelső sor száma = 0
OUT    16
.
.
.

```

Ha saját képernyő kezelést (kurzor mozgatást és attributum kezelést) akarunk végezni, az „A” függelékben ismertetett 6-os hívást kell karakter olvasásra használnunk, mivel ekkor a rendszer nem végez semmilyen kiírási funkciót az olvasással egyidőben. A beolvasott karaktert a képernyő tetszőleges helyére írhatjuk vissza (pl. az aktuális kurzor pozícióba), ezután a kurzort mozgathatjuk.

Az előbbieket attributum kezeléssel kombinálva, már bonyolult képernyő kezelést valósíthatunk meg.

C.ITOH 8510 típusú sornyomtató kezelése

A sornyomtató kicsomagolását és uzernbehelyezésének módját nem írjuk le, mivel ez a gép szállítójának a feladata.

Ha a nyomtatót előlről nézzük, bal oldalon találjuk a hálózati kapcsolót. Bekapcsolás előtt („ON” vagy „I”-es állás), dugjuk be a készülék hálózati dugóját 220 V, 50Hz konnektorba. A hálózati csatlakozó zsinór a készülék hátulján (hátról nézve) jobb oldalon található. Ellenőrizzük, hogy a készülék hátulján (hátról néve) a bal oldalon található adat bemene-tí csatlakozó az M08X-el össze van-e kötve. A csatlakozót az M08X oldalon vagy rögzített módon bekötötték, vagy a csavaros csatlakozó dugót a „PRINTER” felíratú csatlakozó helyre kell rögzíteni.

Előlről nézve a vezérlő panelt látjuk. A bal oldalon három lámpa van, balról a „SEL” (zöld), a „PE” (vörös) és a „POWER” (zöld) lámpák. A világító „SEL” lámpa a sornyomtató működésre kész állapotát jelzi. A világító „PE” lámpa azt jelzi, hogy a sornyomtatóból kifogyott a papír. A világító „POWER” lámpa jelzi, hogy a készülék hálózati feszültséget kap.

A vezérlő panel jobb oldalán a három nyomógomb található, a „SEL”, az „LF” és a „TOF” jelzésűek.

Működésükről később lesz szó.

Papír kivétele és befűzése

Ha a papír elfogy, a „PE” lámpa világít, ekkor a papírt ki kell a nyomtatóból venni, hogy új papírt fűzhessünk be. A kivételhez csavarjuk a nyomtató jobb oldalán található gombot az óramutató járásával egyező irányba, amíg a papír vége a továbbító fogakat elhagyja. Ekkor a papírt kiemelhetjük.

A papír befűzésének módja a következő:

1. A nyomtató tetejének hátsó részén található fedelet vegyük le (emeljük meg és húzzuk hátra).
2. Nyissuk fel a henger előtt található átlátszó műanyag lemezt.
3. Húzzuk magunk felé a hengerhez illeszkedő, papírt rögzítő görgőket.
4. Húzzuk magunk felé a nyomtató tetejének bal oldalán lévő kart („FRICTION” állásba).
5. A henger mögött található papírtovábbító fogaskerekhez illeszkedő rögzítő füleket emeljük fel.
6. A befűzendő papír szélén található első két lyukat illesszük a fogakba. Ha papírszélesség nem egyezik a fogaskerek távolságával, a fogaskerek mögött található (fehér színű) emetyűket mozgassuk a nyomtató hátoldala felé; ekkor a fogaskerek jobbra-balra elcsúsztathatók. A papírszélesség beállítása után az emetyűket húzzuk előre, ekkor a fogaskerek rögzítettek.
7. A fogaskerekhez illeszkedő rögzítő füleket nyomjuk le kattanásig.
8. A nyomtató jobb oldalán található papírtovábbító gombot forgassuk (az óramutató járásával egyező irányba) addig, amíg a papír eleje a henger és a nyomtató fej között min. 3 cm-re kiemelkedik.

9. A bal oldalon található kart nyomjuk „PIN FEED” állásba.
10. A papírt a hengerhez rögzítő görgőket nyomjuk a hengerhez (a papír a görgők és a henger között van).
11. Nyomjuk meg a „TOF” gombot, ekkor a papír kiemelkedik a nyomtatóból, majd megáll. A papír a fogaskerekeknél nem gyűrődhet, a rögzítő füleket nem emeli fel.
12. A papírtovábbító gomb csavarásával állítsuk a következő nyomtatási lap tetejét a nyomtató fej fölé kb. 3 cm-re.
13. Nyomjuk meg a „SEL” gombot.

Ha nyomtatás közben észrevesszük, hogy a papír ferdén fut, gyűrődik stb., nyomjuk meg a „SEL” gombot, mire a nyomtatás leáll. Ekkor a papír kivétele nélkül végezzük el az előbb leírt beállítási műveleteket a 8, 11 és a 12-es művelet kivételével. A „SEL” gomb megnyomása után a nyomtatás folytatódik.

Papír leszakításának helyes módja

A leszakítandó papírt kissé oldal irányban magunk felé húzzuk, hogy a henger előtt található átlátszó műanyag lappal hegyes szöget zárjon be. A papír leszakítását megkönnyítő műanyag lapot szükség esetén a helyén tartjuk. Helytelen leszakítás esetén a papírtovábbító fogaskerekeknél gyűrődés lép fel.

Sornyomtató beépített teszt

A sornyomtató beépített teszt funkciójának elindításához a következőket kell tennünk:

1. Kapcsoljuk be a készüléket, a nyomtató fej a bal oldalon helyezkedik el.
2. Kapcsoljuk ki a készüléket.
3. Szükség esetén helyezzük be a papírt.
4. Miközben a TOF gombot lenyomva tartjuk, kapcsoljuk be a készüléket, a teszt funkció elindul.
5. A teszt befejezéséhez kapcsoljuk ki a készüléket.

Általános megjegyzések

1. Nem szabad szalag nélkül nyomtatni, ez káros a sornyomtatóra.
2. Csak a forgalmazó által ajánlott szalaggal szabad használni a nyomtatót.
3. Szakadt vagy elhasználódott szalagot cseréltsük ki.
4. Csak a forgalmazó által ajánlott papírt használjuk nyomtatási célokra.
5. Az adott papírhoz szükséges vastagságot állítsassuk be a gép forgalmazójával.
6. Ne felejtünk idegen tárgyat a nyomtatóba.

Ha a nyomtató nem működne

Rendellenesség esetén vizsgáljuk meg a következőket:

1. Kap-e feszültséget a nyomtató (POWER lámpa világít-e)
2. Van-e papír a nyomtatóban (PE lámpa nem világít)
3. A nyomtató fedele a helyén van-e.
4. „SEL” lámpa világít-e; ha nem, nyomjuk meg a „SEL” gombot.
5. Biztosíték hibátlan-e (POWER lámpa bekaocsolás esetén világít).

Tartós rendellenesség esetén forduljunk a gép forgalmazójához.

Ismételten felhívjuk a figyelmet a „SEL” lámpára. A nyomtató csak „szelektált” – tehát a lámpa világít – állapotban nyomtat. A „TOF” és az „LF” gombok megnyomására lapdobás illetve soremelés hajtódik végre, de csak „nem szelektált” – tehát a „SEL” lámpa nem világít – esetben. A nyomtató „szelektált” állapotát a „SEL” gomb megnyomásával változtathatjuk.

Hibaüzenetek

A következőkben azokat a hibaüzeneteket adjuk meg, amiket a PROPOS–8 hibás parancs vagy file hivatkozás esetén ad.

- szintaktikai hiba esetén a PROPOS–8 a hibás parancsot kérdőjellel lezárva írja vissza a képernyőre. Ekkor a parancsot helyesen meg kell ismételni.
- NOT FOUND – nem létező lemez file-ra való hivatkozás
- READ ERROR – a file lemezhiba miatt nem olvasható
- WRITE ERROR – a file lemezhiba miatt nem írható
- NO SPACE – nincs elég hely a lemezen
- BAD LOAD – program sikertelen töltése
- SELECT ERROR – nem létező meghajtóra történt hivatkozás
- FILE R/O – a file R/O (írás vpdett) állapotú
- DISK R/O – a lemez R/O (írásvédett) állapotú
- TOO LARGE FILE – túl nagy file méret

A READ ERROR . . . és WRITE ERROR . . . üzenetekre a felhasználó a szóköz, illetve a CR billentyű leütésével válaszolhat. Az előbbi esetben a PROPOS–8 megismétli az olvasást, illetve írást, az utóbbi esetben kihagyja a hibás szektort. Néhány ismétlés (szóköz karakter bevitele) után nyomjuk le a CR billentyűt, mentsük el a lemeztől az összes filet egy üres lemezre (lásd PIP parancs), és a hibás lemezt formátumozzuk újra. Ha a hiba az adott lemezen megismétlődik, a lemez fizikai károsodást szenvedhetett a hosszú használat, vagy hibás kezelés miatt.

T Á R G Y M U T A T Ó

- B Bekapcsolás 11
Billentyűzet 7, 15, 37, 41
- C CENTRONICS típusú interface 9, 17
- D DIR Clemez tartalomjegyzék megjelenítése) 21
DZM típusú sornyomtató 9, 17
EDIT (szövegszerkesztő program) 21
ERA (file-ok törlése) 26
- F File attributumok 15, 16, 18, 38, 42, 43
File hivatkozások 13, 14, 38
File másolás 27, 28
File mérés meghatározás 17, 43
File név 13, 37, 38
File típus 14, 36
File-ok összekapcsolása 29
File-ok megjelenítése (TYPE) 27
File-ok nevének megváltoztatása (REN) 26
File-ok törlése (ERA) 26
Fizikai I/O eszközök 16, 17
- H Hajlékony lemez 8, 11, 12
Hibaüzenetek 56
Hidegindítás 11
- I Irásvédettség 15, 16, 18, 38, 42, 43
- K Karaktorsorozat keresése 24
Képernyő 7, 11, 51
Képernyőkeresés 51
Kurzor mozgatása 22, 52
- L Lemez hivatkozások a file névben 14
Lemez tartalomjegyzékének megjelenítése (DIR) 21
Lemez tárolókapacitásainak meghatározása 16, 18
Lemezcsere 12
Logikai I/O eszközök 16, 17
- M Makro parancsok 23
Másolás 23, 28
Másolás vezérlése parancsban 15
Megjelenítési attributumok 51
Melegindítás 12, 19, 36

- O** Operatív tér 7, 35
Operációs rendszer elhelyezkedése az operatív térben 35
- P** Papír kivétele a nyomtatóból 9, 53, 54
Papírszélesség beállítása 9, 53
Paraméterátadás 36
Parancs beadása 15, 36
Parancs formátum 12
Parancs lánc indítása (SUBMIT) 19
PIP (file-ok másolása, összekapcsolása) 27
Programindítás 12
Programindítás közben végrehajtott funkciók 36
PROPOS-8 hívások 35, 41, 47
- R** REN (file-ok nevének megváltoztatása) 25
Rendszerlemez 11
- S** SAVE (tárterület lemezre való írása) 27
Sornyomtató 9, 53
STAT (lemez és a file-ok állapotának megjelenítése) 16
SUBMIT (parancs lánc előállítása és indítása) 19
- SZ** Számítógép 7
Szövegszerkesztő program (EDIT) 21
- T** Tabulátor 15, 25, 47
Tárterület lemezre való írása (SAVE) 27
TYPE (file-ok megjelenítése) 27

A változtatás jogát fenntartjuk.

Kérjük, hogy jelen kiadványra vonatkozó észrevételeit, megjegyzéseit a következő címre küldje meg:

SCI-L
M08X Vevőszolgálat
Budapest, Iskola u. 10.
1011.

