

UPM OPERÁCIÓS RENDSZER



VIDEOTÓN

TV-Computer

209.195.01.01.

**TV COMPUTER
UPM OPERÁCIÓS RENDSZER
FELHASZNÁLÓI KÉZIKÖNYVE**

1. ÁLTALÁNOS	ISMERETEK.....	5
1.1. Bevezetés		5
1.2. Az UPM felépítése.		5
1.2.1. Az operációs rendszer főbb részei.		5
1.2.2. Kapcsolat az operációs rendszerrel		7
1.2.3. Az UPM file-jai		7
1.2.4. Rendszerparancsok		9
1.2.5. File hivatkozások		9
2. AZ	UPM	HASZNÁLATA.....
		10
2.1. A rendszer betöltése, indítása		11
2.2. Vezérlő karakterek		12
2.3. Beépített parancsok		13
2.3.1. ERA		13
2.3.2. DIR		14
2.3.3. REN.		15
2.3.4. SAVE		15
2.3.5.	TYPE.....	16
2.3.6. USER		16
2.3.7. MON.		17
2.4. Tranziens parancsok		18
2.4.1.	STAT.....	19
2.4.2. LOAD.		25
2.4.3. PIP.		26
2.4.4. ED.		35
2.4.5. SUBMIT.		41
2.4.6. DUMP.		44

2.5. Felhasználói programok	45
2.6. Az UPM hibaüzenetei	47
3. AZ UPM ADATKEZELÉSE	49
3.1. Az operációs rendszer a memóriában.	50
3.1.1. A memória felbontása	50
3.1.2. Rendszerváltók (SVA).	51
3.1.3. File Control Block (FCB).	52
3.2. Az operációs rendszer a diszken.	53
3.3. UDOS.	54
3.3.1. Perifériakezelés.	54
3.3.2. Beépített funkciók	56
3.4. UIOS.	69
3.4.1. Az UIOS ugrótáblája	70
3.4.2. UIOS funkciók	71
3.4.3. Diszk paramétertábla	73
3.4.4. Floppy diszkkal kapcsolatos tudnivalók	75
3.4.5. UPM konzol a TV-Computeren.	76
4. UPM BEÉPÍTETT FUNKCIÓK BŐVÍTÉSE	79
4.1. Bevezetés.	79
4.2. Áttekintés.	80
4.3. A bővítés rutinok felsorolása	84
4.4. A bővítés rutinok leírása.	85

1. ÁLTALÁNOS ISMERETEK AZ UPM-RŐL

(Unified Program Monitor = egységesített program monitor)

1.1 Bevezetés

Az UPM operációs rendszer diszk-perifériával rendelkező 8 bites mikro-számítógépek számára készült. Alkalmas a INTEL8080 és a Z80 központi egységű gépeken az ember-gép kapcsolat biztosítására, a perifériák magas-szintű logikai kezelésére; a felhasználó számára intelligens file-kezelést nyújt. Lehetőséget ad HW-független programok írására, és mert felülről kompatibilis a DIGITAL RESEARCH CP/M 2.2. operációs rendszerével, futtathatók rajta a CP/M 2.2. rendszer-hivatkozásokat használó, az alatt futó rendszer- és felhasználói programok.

A parancsok leírásánál a kerek zárójelek közé () írt egységek jelentése a magyarázó szövegben található. Az aktuális jelentést ide mindig kerek zárójel nélkül kell beírni.

1.2. Az UPM felépítése

1.2.1. Az operációs rendszer főbb részei

Az UPM operációs rendszer a következő főbb részekre osztható.

UIOS	UPM I/O rendszer
UDOS	UPM diszk operációs rendszer
CCP	konzol parancs processzor
TPA	átmeneti (tranzien) programterület
SVA	rendszerváltó területe

Az egyes részek funkciói:

UIOS (=UPM Input-Output System)

Megvalósítja az illeszkedést a hardware környezethez.

Lehetővé teszi a diszk háttértárakhoz való hozzáférést és kapcsolatot teremt a perifériákkal (teletype, display, nyomtató, lyukszalagolvasó/lyukasztó, stb.). A 3.3.1. pontban ennek részletes leírása található.

UDOS (=UPM Disk Operating System)

Elvégzi a teljes periféria-, file- és könyvtárkezelést. Meghatározott belépési ponton keresztül hozzáférést tesz lehetővé a programok számára a beépített funkciókhoz. A beépített funkciók a következő jellegű feladatokat látják el:

SEARCH	file keresés a diszken
OPEN	file nyitás a további műveletekhez
CLOSE	file lezárás az elvégzett művelet után
RENAME	file-név megváltoztatása
READ	adott file egy rekordjának beolvasása
WRITE	egy rekord felírása az adott file-ba
SELECT	diszk meghajtó kiválasztása a további műveletekhez

A 3.3.2. pontban ezekről részletes leírás található.

CCP (=Consol Command Processor)

Szimbólikus kapcsolatot teremt a felhasználói konzol és az UPM rendszer további részei között. Beolvassa és feldolgozza a konzolról érkező parancsokat (pl. file katalógus kinyomtatása, file-tartalom listázása, stb.). Részletebben az 1.2.2. pontban foglalkozunk vele.

TPA (=Transient Program Area)

Ez a memóriaterület áll a felhasználói programok és tranzien্স parancsok rendelkezésére. A CCP a lemezről ide tölti be a kijelölt programokat.

SVA (=System Variable Area)

Memóriaterület a rendszerváltozók számára. Az operációs rendszer meghatározott pontjai érhetőek el az itt található ugrási utasítások segítségével, valamint tartalmazza a diszken elérni kívánt file szükséges paramétereit, és puffertérületet biztosít a file műveletekhez. A 3.1.2. pontban ennek a részletes leírása található.

1.2.2. Kapcsolat az operációs rendszerrel

Az UPM lehetővé teszi, hogy a kezelő egyszerű parancsokkal működtesse a számítógépet. A konzollal a CCP tart közvetlen kapcsolatot. A megadott parancsot értelmezi, hibás parancs esetén jelzi a hibát, egyébként a parancsot gépi szintű utasításokra bontja le, és ezeknek a végrehajtását vezérli, ellenőrzi. Az eredményt ismét a felhasználó számára közvetlenül értelmezhető formában adja vissza.

A CCP általában a számítógéppel aktuálisan kapcsolatban levő diszknek címzi a parancsokat. Maximálisan 16 db diszk meghajtó kapcsolható a rendszerhez. Ezekre a diszk meghajtókra betűkkel hivatkozhatunk:

A = 0. diszk meghajtó
B = 1. diszk meghajtó
...
P = 15. diszk meghajtó

A CCP parancsállapotban jelzi az operátornak, hogy melyik meghajtó az aktuális: a konzolra ">" karakter elé kiírja az aktuális meghajtó betűjelét. Pl. a 0. meghajtó esetén "A>" jelenik meg a konzolon. Ez azt jelenti, hogy egyéb utasítás hiányában az ebben levő lemezen keresi a parancsban kijelölt file-t.

Parancs állapotban új meghajtót jelölhetünk ki aktuális kapcsolatra: a meghajtó betűjelét kell megadnunk kettősponttal (:) lezárva. Pl. az 1. meghajtó kijelölése esetén "B:" megadása szükséges. A parancssor lezárása után a CCP az új meghajtó betűjével fog parancselfogadásra jelentkezni: "B>". Az aktuális meghajtó kijelölése a konzolról, vagy felhasználói programból való megváltoztatásig érvényes.

1.2.3. Az UPM file-jai

A felhasználó programjai, adatai, stb., file-okban vannak a lemezeken tárolva. Az operációs rendszer a file-okat egy file katalógusban tartja nyilván. A katalógus tartalmazza a file nevére, státuszára és a lemezen való elhelyezkedésre vonatkozó adatokat. Ezek egyik része a felhasználó, a másik része az operációs rendszer számára nyújt információkat.

Egy meghatározott file-ra a file nevével és a kiterjesztésével hivatkozhatunk. A file neve maximum 8 ASCII karakter lehet. Valamennyi nagybetű és számjegy használata megengedett. A névben nem használható karakterek: S * .. : ; < = > ? valamint a 20H alatti és 5AH feletti karakterek. (Célszerű az A . . Z betűket és a 0 . . 9 számokat használni.)

A file neve után a tartalmára utaló kiterjesztés áll. A kiterjesztés max. 3 karakter, a névben megengedett karakterekből felépítve. A kiterjesztést a file nevéből a pont (.) karakter választja el.

A file-t a név és a kiterjesztés együtt határozzák meg egyértelműen.

Néhány megengedett file elnevezés:

X.86Y , PROBA.HEX , UJ52.ASM

Az UPM-ben egyes kiterjesztések meghatározott típusú file-t jelölnek. A szokásos elnevezések a következők:

COM	Az UPM alatt futtatható program
HEX	Hexadecimális formátumú program
ASM	Assembly forrásnyelvű ASCII karakterekből álló program
PRN	Listázható formátumú ASCII karakterekből álló file
BAK	Háttér file (elmentett eredeti file)
SYM	Szimbólumtáblát tartalmazó file
CAS	Kazetta típusú formátum a TVC-nél .

Az UPM file-jai két különleges státusszal rendelkezhetnek. A csak olvasható (R/O) státuszú file-t addig nem lehet felülírni, letörölni, átszervezni, amíg írható-olvasható státuszúvá (R/W) nem változtatjuk. A SYS státuszú file a DIR parancsban nem kerül kilistázásra, amíg DIR státuszúvá nem változtatjuk. A file-ok státusza a STAT programmal változtatható. (2.4.1. pont)

1.2.4. Rendszerparancsok

Az **UPM** parancsai kétfélek lehetnek: beépített parancsok és tranziens parancsok.

Beépített parancsok

A CCP tartalmazza őket. Az operációs rendszer részeként a rendszer indításakor töltődnek be a lemezről, ezért a parancssor lezárása után azonnal végrehajthatódnak. A beépített parancsok: ERA, DIR, REN, SAVE, TYPE, USER, MON. Részletes ismertetésük a 2.3. pontban történik.

Tranziens parancsok

A parancs egy file betöltését jelenti a lemezről a TPA-ba, majd annak elindítását. A file kiterjesztése COM kell legyen, és a neve nem lehet azonos a beépített parancsok nevével. Az UPM-ben tehát minden COM kiterjesztésű file egyben tranziens parancsot is jelent. A COM file-ok képzési szabályait a 2.5. pontban részletesen ismertetjük. A compilerek és a LOAD program a feltételeknek megfelelő COM file-t generálnak. Az UPM standard tranziens parancsait a 2.4. pontban ismertetjük, ezeket a programokat a felhasználók az operációs rendszer részeként kapják meg.

1.2.5. File hivatkozások

A legtöbb parancs hivatkozik egy file-ra, vagy file-ok egy csoportjára. Egyetlen file-ra való hivatkozás a file nevével és kiterjesztésével történik. Egyszerre több file-ra való hivatkozás esetén a filenévben és a kiterjesztésben két különleges karakter használható: "? " és a "«".

? karakter

Jelentése bármely karakterpozícióban: tetszőleges **karakter**. Az ilyen file hivatkozás az összes olyan file-ra vonatkozik, amelynek neve és kiterjesztése a

? karakterek kivételével megegyezik a hivatkozással (a ? helyén bármilyen karakter állhat). Példák:

X? Z.C? M Ennek megfelel pl. az [XYZ.COM](#) és az X4Z.CIM file
? ? ? .ASM Az összes olyan file-ra vonatkozik, melynek neve legfeljebb 3 karakter, a kiterjesztése pedig ASM. Ilyen file-ok pl.: X.ASM, UJ.ASM.F53.ASM, de nem ilyen pl.: a FILE.ASM, mert ennek neve 4 karakterből áll.

karakter

Jelentése a file nevében vagy a kiterjesztésben: a további karakterek ebben a részben tetszőlegesek. Jelentése ugyanolyan, mintha a hivatkozásnak azt a részét ? karakterekkel töltöttük volna fel. Eszerint az XYZ*.ASM hivatkozás azonos az XYZ? ???? .ASM-el, a FILE.* hivatkozás pedig a FILE.? ? ? -al. A *.* hivatkozás (=?????????.???) az összes file-ra vonatkozik.

A fenti hivatkozások a parancssor elején kiírt aktuális lemezre vonatkoznak. Pl. "A >" kiírása esetén a 0. meghajtóban levő lemezen keresi az UPM a file-okat. Ha az aktuálistól eltérő lemezre akarunk hivatkozni, akkor közvetlenül a file hivatkozás előtt meg kell adni a kívánt meghajtó betűjelét, kettősponttal (:) elválasztva a hivatkozástól (az aktuális diszk nem változik).

Példák:

B:XYZ.ASM Az 1. meghajtóban levő lemezeiről hivatkozik az XYZ.ASM file-ra.
A:*.HEX A 0. meghajtóban levő lemez összes HEX kiterjesztésű file-jára vonatkozik.

2. AZ UPM HASZNÁLATA

2.1. A rendszer betöltése, indítása

Az **UPM** üzemmód indításához a gép kikapcsolt állapotában be kell dugni az UPM operációs rendszert tartalmazó ROM kazettát a gép bal oldalán kialakított csatlakozóba.

Ugyancsak kikapcsolt állapotban be kell dugni a floppy-diszk csatolót, össze kell kábelezni a floppy-t a floppy csatolóval.

Bekapcsolás után néhány másodperc várakozás következik: a SYSTEM ROM programja — mielőtt átadná a vezérlést a ROM kazettában levő UPM rendszernek — lefuttatja a szokásos teszteket és inicializálásokat. Meghívja a floppy csatolón levő memóriabővítésben levő inicializáló rutint is. Ezután a ROM kazetta programja egy "*" karakterrel kijelentkezik, és egy RETURN billentyű leütésére vár. A RETURN leütése után a

```
61K UPM/TVC READY  
>
```

üzenettel kijelentkezik az UPM operációs rendszer, és a szokásos módon parancs leütésére vár.

Bejelentkezés után az aktuális diszk meghajtó betűjelével és a ">" karakterrel jelzi, hogy a CCP kész a parancsok elfogadására. A meghajtó neve az A..P betűk egyike, a 0..15. meghajtónak megfelelően, sorrendben összerendelve.

A rendszer az aktuális lemezről a kiválasztáskor foglaltsági térképet is készít. Ez alapján tud újabb file-okat felírni, vagy törölni a lemezről. A lemez kicserélése esetén logikailag írásvédetté válik a diszk meghajtó. Ezzel elkerülhető, hogy az előző lemez foglaltsági térképe alapján téves helyen írjunk, vagy töröljünk. Az írásvédelem ^C leütésével szüntethető meg. A "f" jelenti a CTRL (CONTROL) billentyű egyidejű lenyomását. A rendszer ilyenkor elkészíti az új foglaltsági térképet.

A ^C egyben részleges rendszerbetöltést is eredményez: a lemezről a **CCP** és az UDOS betöltődik a memóriába. Ezzel lehetővé válik olyan lemezek használata is, amelyeken másik hardware-hez tartozó UIOS van, **mert** a C az **UIOS** részt nem írja fölül, a saját rendszerhez tartozó **UIOS** marad érvényben.

Felhasználói programból a 0 címre ugrás azonos hatású a ^C leütésével.

2.2. Vezérlő karakterek

Az operációs rendszer bizonyos funkciókat a vezérlő karakterek hatására végez el. Hatásuk a funkció egyszeri végrehajtásáig, vagy megszüntetésükig érvényes. A "" jelenti a CTRL billentyű egyidejű lenyomását. A vezérlő karakterek:

- ^C A rendszer CCP és UDOS része újratöltődik és indul, a lemezfoglaltság aktualizálódik. Az UPM parancsainak félbeszakítását is eredményezi.
- ^E Soremelés, amely csak a konzolon hatásos, nem parancslezárás, (fizikai sorvége)
- ^H A parancssor utolsó karakterét törli.
- ^I Tabulátor: a következő karakter a legközelebbi, 8-cal maradék nélkül osztható pozícióra kerül.
- ^J Soremelés, parancssor lezárása (line feed)
- ^M Sor elejére állás, parancssor lezárása (carriage return)
- ^P A konzolra írt karaktereket a **LIST** eszközzel is kiírja (pl. sornyomtatóra). Hatása újabb ^P-vel szüntethető meg.
- ^R Az aktuális sort a következő sorban megismétli.
- ^S A konzolra történő kivetel megállítása. Hatása bármely karakter leütésével megszűnik.
- ^U Sortörlés a konzol kimenő sorpufferében, a konzolon a sor megmarad.
- ^X Sortörlés a konzol kimenő sorpufferében és a konzolon.

Megjegyezzük, hogy a ^R, ^U, ^X esetén a sor végére # kerül. Ez jelzi, hogy azt a sort az UPM nem veszi figyelembe.

Néhány vezérlő karakter a billentyűzet külön nyomógombjaként is előfordulhat, például:

TAB	^I
RETURN	^M
LINE FEED	^J

2.3. Beépített parancsok

A CCP a következő beépített parancsokat tartalmazza:

ERA	Törli a kijelölt file-okat.
DIR	Listázza a file neveket a lemezről.
REN	Átnevezi a kívánt file-t.
SAVE	Elementi a TPA tartalmát egy file-ba.
TYPE	Kilistázza a kívánt file tartalmát.
USER	Független felhasználók definiálása.
MON	Visszatérés felhasználói programba.

A parancsok hívása az UPM parancsállapotában a parancs nevének és paramétereinek a megadásával történik. A paramétereket a parancsnévtől SPACE vagy TAB karakterrel kell elválasztani. Ha a filenév vagy a file hivatkozás nem az aktuális diszk meghajtóra vonatkozik, akkor a kijelölt diszk meghajtó betűjelét (A. .P) kell a hivatkozás elé írni, attól kettősponttal (:) elválasztva. Ha más előírás nincs, akkor a paramétereket folyamatosan (elválasztó karakter nélkül) kell leírni. A parancssor lezárása a ^M (carriage return) vagy a ^M (line feed) karakterrel történik.

A parancssorban szereplő kisbetűket a CCP nagybetűkké konvertálja, és így használja tovább.

2.1.1 ERA

A parancs a kijelölt diszken törli a file hivatkozásnak megfelelő file-okat. A parancs formátuma:

ERA (file hivatkozás)

A katasztrofális tévedés elkerülése érdekében az

ERA *.*

parancs esetén az operációs rendszer ellenőrző kérdést tesz fel:

ALL (Y/N)?

Csak Y válasz esetén hajtja végre az összes file törlését.

Példák:

ERA FILE.ASM	Az aktuális lemeztől törli a FILE.ASM nevű file-t.
ERA B:*.BAK	A B: meghajtóban levő lemeztől törli az összes BAK kiterjesztésű file-t.
ERA P*.COM	Az aktuális lemeztől törli a P betűvel kezdődő, COM kiterjesztésű file-okat.

2.3.2. DIR

A parancs listázza a kijelölt diszkról a file hivatkozásnak megfelelő file neveket. A parancs formátuma:

DIR (file hivatkozás)

A listázás 4 oszlopban történik.

A kijelölt diszk összes file-jára két módon lehet hivatkozni: a *.* hivatkozással és a file hivatkozás elhagyásával.

Példák:

DIR FILE.???	Az aktuális diszkról listázza 4 oszlopban a konzolra a FILE nevű, tetszőleges kiterjesztésű file-okat.
DIR A: *.HEX	Az A: meghajtóban levő lemeztől listázza a konzolra a HEX kiterjesztésű file-neveket.
DIR	Az aktuális diszkról listázza az összes file-nevet a konzolra.
DIR B:	A B: diszkról listázza az összes file-nevet.

2.3.3. REN

A parancs a diszken már meglevő file-ok átnevezésére szolgál, a file tartalma nem változik. Formátuma:

REN (új filenév)=(régi filenév)

Egy REN paranccsal csak egyetlen file neve módosítható. Az aktuálistól eltérő diszk meghajtó megadása elegendő az (új filenév) előtt. A két filenév előtt nem szerepelhet különböző diszk meghajtó neve, mert a parancs nem végez file mozgatást.

Példák helyes parancshívásra:

```
REN      UJ.ASM=REGI.ASM
REN      B:UJ.ASM=UJ.BAK
REN      A:X.COM=A:Y.COM
```

2.3.4. SAVE

A parancs a TPA elejétől (100H címtől) a memória tartalmát a megadott file-néven a kijelölt lemezre írja. Ha már van ilyen néven file, azt kérdés nélkül felülírja.

Formátuma:

SAVE (lapszám) (filenév)

A (lapszám) decimális számot jelöl, értéke: 1. . .255. 1 lap 256 byte (=100H byte) hosszú. A (lapszám)-tól SPACE vagy TAB karakterrel el kell választani a (filenév) részt.

Példák:

```
SAVE 2 Y.COM  A memória 100H-tól 2FFH-ig terjedő része Y.COM néven
               az aktuális lemezre kerül.
SAVE 30 B:X.COM A memória 100H-tól 1EFFH-ig terjedő része X.COM
               néven a B: lemezre kerül. Ismét felhívjuk a figyelmet
               arra, hogy a lapok száma decimális szám, a 30 lap
               megfelel 1EH számú lapnak.
```

2.3.5. TYPE

A parancs ASCII file-ok tartalmát listázza a konzolra.

Formátuma:

TYPE (filenév)

-

Az aktuálistól eltérő diszkról is lehetséges a listázás.

Példák:

TYPE X.PRN Az aktuális diszkról az X.PRN file listázása a konzolra.

TYPE B:Y.ASM A B: lemezeről az Y.ASM file listázása a konzolon.

2.3.6. USER

A paranccsal lehetőség nyílik egymástól független, egymás programjait nem látó felhasználók definiálására.

Formátuma:

USER (felhasználó száma)

A (felhasználó száma) egy egész szám, értéke: 0. . .15.

Egy USER szám kijelölés a következő USER szám kijelöléséig érvényes. Ezalatt csak a megadott USER számú programokhoz lehet hozzáférni, a lemezre ilyenkor felírt file-ok szintén az aktuális USER szám alatt lesznek csak elérhetők. A gép bekapcsolásakor, vagy rendszer RESET esetén a 0. USER szám lesz aktív. Egy másik USER szám alatti programot az aktuális USER szám alá a PIP másolóprogram G opciójával vihetünk át.

Ha az aktuális user szám nem 0, akkor a rendszer kijelentkezéskor az aktuális user számot is kiírja.

```
pl,   B3 > ; B drive 3. user
      C  > ; C drive 0. user
      A12 > ; A drive 12. user
```

Minden drive-on a 0. user a *.COM és *.SUB file-ok tekintetében globális a többi userre nézve. Ez úgy értendő, hogy ha a *.COM vagy *.SUB file-t a rendszer nem találja az aktuális user alatt, akkor a keresést a 0. user alatt folytatja.

Ha a *.COM file a 0. userről kerül betöltésre, egy nem nullás aktuális user esetén akkor a rendszer megismétli a file nevet.

```
Pl.   C3>STAT
      A:R/W, Space:56k
      C:R/W, Space:135k
      ; STAT.COM volt a 3. userben
```

```
C3>STAT
C:STAT
A:R/W, Space56k
C:R/W,Space:135k
; STAT.COM csak a 0. userben volt
```

2.3.7. MON

A parancs a felhasználó által megadott, fix címre való vezérlésátadást biztosít. A parancs formátuma:

MON

Ennek hatására az UIOS-ban definiált címre (3.4.1. pont) adódik át a vezérlés. Ezt a címet a felhasználó állíthatja be, így lehetővé válik az operációs rendszerből a saját monitor-, vagy egyéb programba való visszatérés.

2.4. Tranziens parancsok

A tranziens parancsok a lemezen található, mint COM típusú file-ok. Hívásuk a parancs nevével és a paraméterek megadásával történik. A parancs lezárásakor a megfelelő nevű, COM típusú file betöltődik a TPA-ba, majd elindul a futása. Az UPM standard tranziens parancsai a rendszerben gyakran használt funkciókat valósítják meg. A standard tranziens parancsok a következők:

STAT	Statisztikai információkat nyújt a lemezokről, file-okról, módosítja a file-státuszt, megadja vagy módosítja a logikai periféria hozzárendelést.
LOAD	Az assemblerrel lefordított file-ból a TPA-ba tölthető, végrehajtható formátumú file-t készít.
PIP	Másoló program, mellyel lehetőség nyílik a rendszer perifériái közötti adatátvitelre.
ED	Az UPM szövegszerkesztő programja
SUBMIT	Parancsfile-ok (file-ba foglalt rendszerparancsok) végrehajtására szolgál.
DUMP	Kilistázza a kijelölt file tartalmát hexadecimális és ASCII formában.

A parancsok hívása az UPM parancsállapotában a parancs nevének és paramétereinek a megadásával történik. A paramétereket a parancsnévtől SPACE vagy TAB karakterrel kell elválasztani. Ha a filenév vagy a file hivatkozás nem az aktuális diszk meghajtóra vonatkozik, akkor a kijelölt diszk meghajtó betűjelét (A...P) kell a hivatkozás elé írni, attól kettősponttal (:) elválasztva. Ha más előírás nincs, akkor a paramétereket folyamatosan (elválasztó karakter nélkül) kell leírni. A parancssor lezárása a ^M (carriage return) vagy a ^J (line feed) karakterrel történik.

A parancsban szereplő kisbetűket a CCP nagybetűkké konvertálja és így használja tovább.

A parancsok, azaz a *.COM file-ok betöltése előre nem definiált meghajtóról is lehetséges, ha X: drive-ot adunk meg. Ilyenkor a CCP, ha megtalálta a parancsot, akkor újra kiírja annak a meghajtónak a nevével együtt, ahol a parancsot megtalálta.

Pl. B>X:STAT

A:STAT

A:R/W, Space:56k

B:R/W, Space:412k

;Az A meghajtón találta meg a [STAT.COM](#) file-t a CCP.

2.4.1. STAT

A parancs a következő jellemzőkről nyújt információkat:

- a lemezek jellemzőiről, tartalmáról, foglaltságáról, státuszáról
- a file-ok méretéről, státuszáról
- az aktív userokról
- a kijelölt és a kijelölhető fizikai perifériákról

A felhasználó a konzolról állíthatja be a

- logikai periféria-névhez tartozó fizikai eszközt
- lemezek és file-ok státuszát

A parancs formátuma:

STAT (paraméterek)

A **STAT** lehetővé teszi a felügyeletet a logikai-fizikai periféria-hozzárendelés fölött (lásd 3.3.1. pont).

Logikai periféria-nevek:

CON: Konzol eszköz, a CCP ezen keresztül tart kapcsolatot az operátorral

RDR: Lyukszalagolvasó

PUN: Lyukszalaglyukasztó
LST: Listát nyomtató periféria

A perifériákra való hivatkozás megkönnyítése érdekében a következő fizikai periféria neveket definiáljuk:

TTY Teletype. Mind a négy logikai perifériához hozzárendelhető.
CRT Display. Lehet konzol, vagy nyomtató eszköz.
BAT Különleges eszköz kijelölés. Hatására az aktuális RDR: eszköz lesz a konzol, a kimenet pedig az aktuális LST: eszközre kerül.
UC1 A felhasználó által definiált konzol
PTR Lyukszalagolvasó
UR1 A felhasználó által definiált olvasó.
UR2 A felhasználó által definiált olvasó
PTP Lyukszalaglyukasztó
UP1 A felhasználó által definiált lyukasztó.
UP2 A felhasználó által definiált lyukasztó.
LPT Sornyomtató.
UL1 A felhasználó által definiált nyomtató.

A logikai és a fizikai perifériák neveit hivatkozáskor minden esetben kettősponttal (:) kell lezárni, hasonlóan a diszk perifériákhoz.

A kiadható STAT parancsok a következők:

a)STATVAL:

A parancs a STAT szolgáltatásairól ad felvilágosítást. Kiliptázza a 4 lehetséges logikai perifériához rendelhető fizikai periféria-neveket és a STAT paramétereire ket:

IOBYTE ASSIGN:

CON:=TTY:	CRT:	BAT:	UC1:
RDR:=TTY:	PTR:	UR1:	UC2:
PUN:=TTY:	PTP:	UP1:	UP2:
LST:=TTY:	CRT:	LPT:	UL1:

GET FILE SIZE	:D:FILENAME.TYP
SET FILE INDICATOR	:FILENAME.TYP R/0,R/W,SYS,DIR
GET DISK CHARACTERICS	:D:DSK:
SET DISKTEMP R/O STATUS	:D:=R/0
GET USER STATUS	:USR:

b) STAT DEV:

A parancs kilistázza a logikai periféria-nevekhez aktuálisan, pillanatnyilag hozzárendelt fizikai periféria-neveket, pl.:

CON:=CRT:
RDR:=UR1 :
PUN:=PTP:
LST:=TTY:

c) STAT (logikaiperiféria): = (fizikaiperiféria):

Az aktuális hozzárendelést egyenként megváltoztathatjuk a STAT VAL: felsorolt lehetőségeinek megfelelően. Például, ha a listázó eszköznek a sornyomtatót akarjuk kijelölni, ezt a következőképpen tehetjük:

STATLST: = LPT:

d) STAT (file hivatkozás)

A parancs az aktuális, vagy a kijelölt lemezen levő, a file hivatkozásnak megfelelő file-okról és azok jellemzőiről készít listát. Ebben szerepel a lemezegység neve, a file neve, a file extentjeinek (az általa elfoglalt könyvtári soroknak) száma, a file rekordjainak száma, a file mérete KBYTE-ban és a file státusza. Például, ha a B: lemez az aktuális, akkor az A: lemezen levő COM típusú file-okról a következő paranccsal kaphatunk információkat:

STAT A: *.COM

A parancsra kapott válasz:

D:FILENAME	.TYP	EXT	RECS	KBYTES	ATT
A:WM	.COM	1	84	11	R/W
A:SID	.COM	1	56	7	R/W
A:MAC	.COM	1	92	12	R/W
A:STAT	.COM	1	41	6	R/W
A:PIP	.COM	1	58	8	R/W
A:GEN	.COM	1	7	1	R/W
A:LOAD	.COM	1	10	2	R/W
A:EDI	.COM	1	34	5	R/W
A:ST	.COM	1	18	3	R/W

REMAINING SPACE ON DISK A: 75 [KBYTE]

Ha csak a lemezen levő üres helyre vagyunk kíváncsiak, akkor csak a lemez nevét kell paraméterként megadni, illetve az aktuális lemez esetén az is elhagyható.

Például:

STAT A:

A parancsra kapott válasz:

REMAINING SPACE ON DISK A: 75 [KBYTE]

A file státusz a következő módon változtatható:

e) *STAT (file hivatkozás) \$ (file státusz)*

A parancs az aktuális, vagy a kijelölt lemezen levő, a file hivatkozásnak megfelelő file-okat a (file státusz)-ban megadott státuszúvá teszi. A file státuszok megváltoztatásukig érvényesek. A lehetséges státuszok:

R/W	A file olvasható/írható
R/O	A file csak olvasható
SYS	A file a DIR parancsban nem kerül kilistázásra, de a STAT-ban látható
DIR	A SYS státusz megszüntetése

Például, ha az A: lemez az aktuális, akkor a rajta levő COM file-okat a következőképpen tehetjük csak olvashatóvá:

```
STAT *.COM$ R/O
```

A parancsra kapott válasz:

```
FILES SET
```

```
A:WM      .COM  SET TO  R/O
A:SID     .COM  SET TO  R/O
A:MAC     .COM  SET TO  R/O
A:STAT    .COM  SET TO  R/O
A:PIP     .COM  SET TO  R/O
A:GEN     .COM  SET TO  R/O
A:LOAD    .COM  SET TO  R/O
A:EDI     .COM  SET TO  R/O
A:ST      .COM  SET TO  R/O
```

f) *STAT (diszk meghajtó) :DSK:*

A parancs a kijelölt lemez jellemzőit adja meg. Az aktuális lemezre elegendő a következő módon hivatkozni:

```
STAT DSK:
```

Az A: lemezre pedig így hivatkozhatunk:

```
STAT A:DSK:
```

A parancsra kapott válasz:

```
A:  DISK      CHARACTERISTICS
    26        SECTOR/TRACK
    4         SECTOR/BLOCK
    243       BLOCK/DISK
    243       KBYTE/DISK
    64        DIRECTORY ENTRY
    2         TRACK OFFSET
```

g) *STAT (diszk meghajtó):= R/O*

A parancs hatására a kijelölt lemez logikailag írásvédetté válik. A beállított státusz a rendszer újra betöltéséig érvényes. A betöltés után a lemez státusza R/W.

h) *STA T (diszk meghajtó): USR:*

A parancs megadja a kijelölt lemezen az aktuális USER számot és felsorolja azokat az user-számokat, amelyek alatt a lemezen program van. Az aktuális lemezre elegendő a következő módon hivatkozni:

```
STAT USR:
```

Az A: lemezre pedig így hivatkozhatunk:

```
STAT A:USR:
```

A parancsra kapott válasz:

```
0 DEFAULT USER NUMBER
USER NUMBERS ON DISK A:0. . .3
```

A STAT parancs működését a funkció végrehajtása jelzi. Hibás parancsmegadás, vagy értelmetlen kijelölés esetén a hibajelzés:

```
INVALID COMMAND
```


2.4.2. LOAD

A parancs a hexadecimális formátumú, tárgykódú, HEX kiterjesztésű file-ból készíti az operációs rendszer alatt futó, COM kiterjesztésű file-t.

A parancs formátuma:

LOAD (filenév)

A filenévben nem kötelező a HEX kiterjesztést megadni, mert elhagyása esetén is ezt tételezi fel a LOAD. A parancs a TPA-ba betölthető programot készíti. Ha a program nem a TPA elején (100H címen) indul, akkor az induló címig levő területre 00 kerül.

A bejelentkező szöveg:

UPM LOAD VER. 1.1

Hibátlan futás esetén a következő információk kerülnek a konzolra:

```
LOAD ADDRESS:XXXX  
LAST ADDRESS: YYY  
RECORDS WRITTEN: VVVV  
END OF LOAD, NO ERROR/S/
```

ahol:

XXXX az első nem 00-val töltött memóriahely

YYY az utolsó töltött memóriahely címe

VVVV a felírt 128 byte-os rekordok száma

A LOAD a következő hibajelzéseket adja:

- LOAD ADDRESS ERROR
a kezdőcím 100H alatti, a program rendszerterületre kerülne.
- INVERTED ADDRESS ERROR
az aktuális programszámláló-értéknél kisebb értékre való töltés fordul elő.

- CHECKSUM ERROR
a HEX file-ban hibás kontroll szumma fordul elő
- HEX CHARACTER ERROR
a HEX file-ban nem hex karakter szerepel
- CANNOT OPEN SOURCE file
a HEX file-t nem sikerült megnyitni
- NO DIRECTORY SPACE
a lemezen nincs üres DIR belépési pont
- NO MEMORY
a futáshoz kevés a rendszerben levő memória
- CANNOT WRITE ERROR
sikertelen írás a COM file-ban
- CANNOT READ ERROR
sikertelen olvasása HEX file-ban
- CANNOT CLOSE COM file
sikertelen COM file lezárás

2.4.3. PIP

A parancs az UPM operációs rendszerben a file-ok másolását és a perifériák közti adatmozgatást végzi.

A parancs formátuma:

- a) PIP (paraméterek)
- b) PIP

Az a) esetben a parancs végrehajtása után az operációs rendszer parancsállapotába történik visszatérés. A b) esetben a PIP *-gal jelentkezik be, ezzel jelzi, hogy kész az a) híváshoz hasonlóan a paraméterek elfogadására. A parancs

végrehajtása után azonban újból * jelzi, hogy újabb paraméterekkel további parancs hajtható végre. Az UPM parancsállapotába a * után leütött, parancssort lezáró karakterrel (^J vagy ^M) térhetünk vissza.

A paramétersor formátuma:

(output)=(input#1),(input#2),. . .,(input#n)

A paramétereket egymás után folyamatosan, helyköz nélkül kell leírni. Az egyes egységek jelentése a következő:

(output)=(cél periféria):(file-név)
(input#k)=(forrás periféria):(file hivatkozás) [(opciók)]

Az elemek jelentése:

(cél periféria)

Annak a perifériának a neve, amelyre az adatátvitel történik. Megadása esetén mindig kettőspont (;) követi. Az aktuális lemezegység neve elhagyható, ha az (output)-ban meg van adva a (filenév).

(file-név)

Annak a kimeneti file-nak lesz a neve, amelybe a PIP a forrásokból' érkező adatokat folyamatosan bemásolja. Ha a (cél periféria) nem file szervezésű (pl. sornyomatató), akkor nem kell megadni.

(forrás periféria)

Annak a perifériának a neve, ahonnan az adatátvitel történik. Megadása esetén mindig kettőspont (;) követi. Az aktuális lemezegység neve elhagyható, ha az (input#k)-ban meg van adva a (file hivatkozás).

(file hivatkozás)

Annak a file-nak, vagy file-ok egy csoportjának a neve, ahonnan adatokat akarunk átmásolni. Több file-ra egyszerre a ? és *

karakterekkel hivatkozhatunk (lásd. 1.2.5.). Ennek hatására a megfelelő file-okból egyenként végrehajtja a PIP az opciókkal előírt adatátvitelt. Ha a (forrás periféria) nem file szervezésű (pl. konzol), akkor nem kell megadni.

(opciók)

Az adatátvitelt befolyásoló előírások. Az opciókat [] között kell megadni. A megengedett opciók: A . .Z karakterek, a megfelelő paraméterekkel. Érvényességi körük az őket tartalmazó (input#k) kijelölésre érvényes, kivéve a W opciót, melynek hatása az egész input sorra kiterjed. A funkcióval nem rendelkező opciók hatástalanok.

A perifériák kijelölése logikai vagy fizikai periféria-nevekkel lehetséges. A periféria-nevek megegyeznek a STAT-nál használt nevekkkel, valamint még néhány speciális eszköz neve is megengedett. A logikai perifériák aktuális jelentését az I/O byte tartalmazza (3.3.1. pont). A STAT paranccsal lehetséges ennek lekérdezése és módosítása (2.4.1. pont).

Logikai eszköznevek:

CON: (konzol), RDR: (olvasó), PUN: (lyukasztó), LST: (listázó)

Fizikai eszköznevek:

TTY (konzol, olvasó, lyukasztó vagy listázó)
CRT (konzol vagy listázó), UC1: (konzol)
PTR (olvasó), UR1: (olvasó), UR2: (olvasó)
PTP (lyukasztó), UP1: (lyukasztó), UP2: (lyukasztó)
LPT (listázó), UL1: (listázó)

(A BAT: fizikai eszköz név a PIP parancsban nem használható.)

A diszk perifériákra az A . .P betűk egyikével hivatkozhatunk.

Speciális eszköznevek:

- NUL: az output eszközre 40 db ASCII 0-t küld a PIP
- EOF: az output eszközre egy ^Z-t (1AH) küld a PIP, amelynek jelentése: "end of file".
- INP: speciális input periféria kezelést lehet a PIP programba beírni. Az átvitel byte-onként történik. A 103H címre elhelyezett CALL utasítással ugorhatunk a beszúrt programra, a visszatérő karaktert a 109H címre kell elhelyezni. A program a 10AH-3FFH területre írható.
- OUT: speciális output periféria kezelést lehet a PIP programba beírni. A beszúrt programra a 106H címen elhelyezett CALL utasítással ugorhatunk, a kimenő karaktert a 109H címre kell elhelyezni. A program a 10AH—3FFH területre írható, a beszúrás pl. a DDZT paranccsal végezhetjük el.
- PRN: megegyezik az LST:-vel, kivéve, hogy a tabulátort 8 karakternek értelmezi, a sorokat sorszámozza, és 60 soronként beszúr egy lapváltást. A plusz funkció azonos a következő opció hatásával: [T8NP]

A funkciókkal rendelkező opciók:

- Dn : a forrásból érkező sorokból csak a sor első "n" karakterét viszi át az outputba.
- E : minden átvitt karaktert kiír a konzolra is.
- F : kiszűri az átvitt karakterek közül a "form feed" karaktert.
- Gn : az n. userben lévő diszkről másol az aktuális userbe
- H : átvitel HEX-formátum ellenőrzéssel. Hibajelzés esetén lehetőség van az átvitel folytatására egy space leütésével, ehhez az outputba a következő blokk elejétől kerülnek be a karakterek.

átvitel HEX-formátum ellenőrzéssel. A csak hexa 0-t tartalmazó rekordok nem kerülnek átvitelre. Hibajelzés esetén a H-hoz hasonlóan viselked'rk.

L az átvitel során előforduló nagybetűket kisbetűvé alakítva viszi át az outputba.

N a sorokat sorszámmal látja el, és úgy teszi az output eszközre.

O object file átvitel esetén a ^Z karaktereket is átviszi anélkül, hogy termináló karakterként értelmezné.

Pn minden „n”-edik input sor után lapemelést tesz az outputba. Ha n nincs megadva, 60 soros lapokat készít.

QKAR ""^Z: az outputba a KAR minta bejöveteléig viszi át az inputot, beleértve a KAR karaktersorozatát is.

R : az input file-t a SYS és R/O bejelöléssel együtt viszi át az outputba, ha létezett ilyen megjelölés az input név-kiterjesztésének felső bitjein. (FCB 9. . .11 pozíciói)

SKAR ^Z : az outputba a KAR minta bejövetelétől kezdve viszi át az inputot, beleértve a KAR karaktersorozatát is. Az S és Q opciók együttes alkalmazásával lehetőség van egy határokkal definiált rész átvitelére.

Tn a TAB karaktert (09H) "n" karakter hosszúnak definiálja

U az átvitel során előforduló kisbetűket nagybetűkké konvertálja, és így viszi az outputba.

V : diszk output esetén a fölírt rekordot visszaolvassa és összehasonlítja a fölírttal. Különbözőség esetén hibajelzést ad.

- W** : ha az output lemezen a megadott nével van már file, azt a PIP kérdés nélkül átírja a file R/D státusza esetén is. A W nélkül a PIP választási lehetőséget kínál a felhasználónak az átírás vagy változatlanul hagyás eldöntésében.
- Z** : az inputról érkező karaktereket 0 paritással viszi át az outputba.

Példa néhány megengedett opcióra:

```
[OV ]  
[T8D60U ]  
[ QKEZDET^ZSVEG ^ZT8 ]
```

A PIP szolgáltatásai:

A PIP az inputból az outputba másolja a parancsokban kijelölt byte-okat. Másolás közben végrehajtja az opciókkal elrendelt ellenőrzéseket, átalakításokat, és ellenőrzi a helyes perifériakijelölést (pl. a sornyomatató ne legyen forrás periféria).

A parancssor megengedett maximális hossza 128 byte, a PIP utáni space kivételével folyamatosan, helyköz nélkül írva.

Ha az átviendő adatok byte-szervezésű eszköztől érkeznek, (pl. konzol), a file végét a Z karakter jelenti (hexadecimális 1A), amely az outputba is átkerül.

Az input sorban kijelölt adatok balról jobbra folyamatosan, konkatenációval kerülnek az outputba. Ha egy file hivatkozással egyszerre több file-t jelölünk ki egy lemeztől átvitelre, akkor a megfelelő file-ok a lemezen való elhelyezkedés sorrendjében kerülnek átmásolásra. Az input file-ok tartalma nem változik az adatátvitel során.

COPY üzemmód:

A file-ok lemeztől lemezre való átmásolása változatlan néven a **COPY**

üzemmóddal történik. A paramétersora ilyenkor a következő formátumú:

(output)=(output lemez):
(input#k)=(input lemez):(file hivatkozás)[(opciók)]

Az (input lemez)-ről a (file hivatkozás)nak megfelelő file-ok az elhelyezkedés sorrendjében, eredeti nevükön rendre átmásolódnak az (output lemez)-re, miközben a PIP elvégzi az (opciók)-kal kijelölt ellenőrzéseket, átalakításokat. A konzolon a

COPYING FILES

felirat után a PIP kiírja az éppen másolt file nevét. A COPY a SPACE vagy ^C leütésével megszakítható.

Példák PIP parancsokra:

PIP LST:= X.PRN

Az aktuális lemezeről az X.PRN file a listázó eszközre (pl. sornyomtatóra) kerül, majd az UPM parancsállapotába történik visszatérés.

PIP

A PIP *-gal bejelentkezik, ezzel jelzi, hogy kész a parancs paramétersor elfogadására.

*CON:=B:X.ASM[D40]

A B: lemezeről az X.ASM file-t kiírja a konzolra. Az opció szerint minden sorból csak 40 karaktert ír ki, a többi nem.

*A:Z.ASM=X.ASM[SBEGIN^Z].B:Y.ASM[QERROR^Z]

Az A: lemezen létrehozza a PIP a Z.ASM file-t. Először bemásolja az aktuális lemezeről az X.ASM file-t a BEGIN karaktersorozattal kezdve, majd a B: lemezeről az Y.ASM file-t másolja bele, az ERROR karaktersorozattal bezárólag.

PUN:=NUL:,X.HEX[I],EOF:,NUL:

Először kiküld a PIP a lyukasztóra 40 db ASCII 0-t, majd az aktuális lemezről az X.HEX file-t. Elvégzi a hexa formátum ellenőrzést és kihagyja az átvitelből a csak hexa 0-t tartalmazó rekordokat. Ezután ^Z-t küld a lyukasztóra, majd újabb 40 db ASCII 0-t.

A:=.LIB

Az aktuális lemezről az összes LIB kiterjesztésű file-t átmásolja az A: lemezre. Minden file neve az A: lemezen azonos lesz az eredeti névvel. Pl. 3 file átmásolása esetén a konzolon megjelenő üzenet:

COPYING FILES

X.LIB

Y.LIB

SUBR.LIB

*.HEX=CON:,B:Y.HEX,PTR:

Az aktuális lemezen létrehozza az X.HEX file-t. Először a konzolról történik beolvasás, ennek végét a konzolon leütött ^Z karakter jelzi. A file következő részébe bemásolódik a B: lemezről az Y.HEX file, majd ezután a lyukszalagolvasóról kerülnek az adatok a file végére, amíg egy ^Z karakter nem érkezik.

* (parancslezáró karakter)

A ^J vagy "M hatására az UPM parancsállapotába történik visszatérés.

A PIP hívásakor a következő szöveget írja ki:

— PIP VER 1.1 —

Ezután a hívás módtól függően program-végrehajtás, vagy * kiírása után parancs-várás következik.

A PIP hibaüzenetei a következők:

- **COMMAND SYNTAX ERROR!**

A parancssorban szintaktikai hiba van, a parancs nem felel meg a 3. pont szerinti felépítésnek (pl.: túl hosszú file-név; nem megengedett karakter a névben; ? vagy * jel az outputban; nem megfelelő eszközjelölés; hibás elválasztó karakter stb.)

- **CHECKSUM ERROR!**

az I és H opció esetén a számított és beolvasott kontroll összeg nem egyezik meg.

- **CHARACTER ERROR!**

az I és H opcióknál nem megengedett karakter érkezett az inputból.

- **VERIFY ERROR!**

a diszkról visszaolvasott karaktersorozat nem egyezik meg a felírttal.

- **NO SPACE ON DISK**

a file nem fér föl az output lemezre

- **FILE CANNOT OPEN**

a PIP nem tudja megnyitni az input vagy output file-t

- **FILE CANNOT CLOSE**

a PIP nem tudja az output file-t lezárni

- **FILE MAKE ERROR**

a PIP nem tudja az output file-t megnyitni

- **DEVICE ERROR**

a parancs-sorban nem megengedett eszköznév van

- **DISK FULL**

az output file nem fér föl az output lemezre

- **TRANSMITTING ABORTED**

az átvitelt a konzolról félbeszakították.

2.4.4. ED

Az ED az UPM operációs rendszer diszk-orientált szövegszerkesztő programja.

Az ED hívása:

ED (input file-név) (output file-név)

A két file-nevet egymástól és az ED-től SPACE választja el.

(input file-név)

A forrásfile neve. Ha nem az aktuális lemezen található, akkor a megfelelő lemez nevét a file-név elé kell írni, attól kettősponttal (:) elválasztva. A file tartalma a szerkesztés után változatlanul marad.

(output file-név)

Az eredmény file neve. Ha nem az aktuális lemezre akarjuk felírni, akkor a kívánt lemez nevét a file-név elé kell írni, attól kettősponttal (:) elválasztva.

Az (output file-név) elhagyása esetén az eredmény file az (input file-név) nevét kapja, míg a forrásfile kiterjesztése BAK lesz. Új file írása esetén csak az (output file-név) megadása szükséges, ekkor az ED a következő szöveget írja ki:

NEW FILE

A hívás után az ED ******-gal bejelentkezik, ezzel jelzi, hogy kész a szerkesztési parancsok elfogadására. Példák az ED hívására:

ED X.ASM

Az aktuális lemezről az X.ASM file-t kijelöli szerkesztésre. A módosítások után a forrásfile X.BAK néven, az eredményfile S.ASM néven található az aktuális lemezen.

ED A:X.AS, B:Y.ASM

Az A: lemezről az X.ASM file lesz a forrásfile, az eredményfile pedig Y.ASM néven a B: lemezre kerül.

ED B:NEW.ASM

Ha a NEW.ASM file még nem szerepel a B: lemezen, akkor az ED a következő módon jelzi ezt:

NEW FILE

majd a V bejelentkezés után inzertr módban új file írható. Az eredmény file NEW.ASM néven a B: lemezre kerül.

A szerkjsztés a memóriában történik. A forrásfile-ból soronként lehet betölteni az adatokat a memóriába. A betöltés a forrásfile elején kezdődik és minden újabb betöltés onnan indul, ahol az előző befejeződött. A betöltött sorok a memória első szabad helyére kerülnek, itt a sorokat az ED sorszámával látja el. A kész sorokat az eredmény file-ba lehet másolni, a memóriában levő legkisebb sorszámától kezdődően. A szerkesztés befejezésekor a memóriában levő sorokat az ED átmásolja az eredmény file-ba. Az eredmény file tartalmát csak újabb ED paranccsal tudjuk módosítani.

A memória véges méretei miatt egy módosítás alatt több betöltésre és kivitelre is szükség lehet. Új file létrehozásakor természetesen a betöltés elmarad, a memória a betelte után csak a kivitel szükséges. A memória egy része látható a konzolon.

A bejelentkezés után egy sorszám és a '*' karakter jelzi, hogy az ED kész a parancsok elfogadására. A '*.' előtt álló sorszám mutatja, hogy a pointer melyik sorban áll. A pointer a konzolon nem látható, de az összes utasítás a pointertól kezdve érvényes, így helye mindig lekérdezhető.

Az ED-ben használható az összes UPM vezérlő karakter, így a parancssorok lezárás előtt javíthatók, törölhetők, stb.

Az ED parancsok felépítése:

A * bejelentkezés után A. .Z karakterekkel adhatók parancsok a programnak. A betű előtt egy decimális szám állhat, amely mennyiségileg értelmezi az utána álló parancsot. A szám hiánya 1-et jelent. A szám maximális értéke

32.768 lehet. A O-nak speciális jelentése van, ezt a parancsoknál külön értelmezzük. Néhány parancs előtt állhat '-' előjel, ez a végrehajtásnak a pointerhez viszonyított irányát módosítja. Egyes parancsok után minta karaktersorozat következik (pl. F), ennek lezárását a CTRL/Z (1 AH) karakter jelöli.

Fennáll a lehetőség egy parancssorban több parancs megadására, ezek elválasztás nélkül egymás után írandók.

Bizonyos parancsok azonban csak önmagukban adhatók meg. A parancssort a "carriage return" karakterrel kell lezárni, ezután a parancssor a megadás sorrendjében végrehajtódik.

A parancsok végrehajtása egy SPACE leütésével megszakítható. A parancsok leírásában szereplő '+' előjelet a parancs kiadásakor nem kell kiírni.

Az ED parancsai:

- nA n sor betöltése az input file-ból a memóriába. A parancs nem engedi meg a memória végén túli töltést. A 0A jelentése: a még üres memóriahelyet félig tölti meg. A #A hatására az egész szabad memória feltöltődik.
- +—B a pointer állítása a memória elejére (+ esetén), Ml. a végére (— esetén).
- +—nC a pointer mozgatása karakterenként. A +n esetén előre, —n esetén hátrafelé. A memóriahatárok elérésekor a parancs hatástalan.
- +—nD n karakter törlése a pointertől kezdve (+ esetén), Nl. a pointer előtt (— esetén).
- E a szerkesztés vége. A teljes memória, Ml. a még az inputból be nem olvasott sorok az outputba kerülnek, és lezáródik a output file. Az E parancs sorában nem lehet több parancs.

nFMINTA^Z a pointert a memóriában a "MINTA" karaktersor n-edik előfordulása utánra állítja, lehetőséget adva a memóriában való keresésre. Ha a kijelölt mintát nem találta meg n-szer a memóriában, a 'BREAK' felirat jelenik meg.

H a szerkesztés lezárása és újraindítása az ED újratöltése nélkül, az output lezárásával és inputként való újra megnyitásával. A H output név megadásával való indítása az operációs rendszerbe való visszatérést eredményez. A parancs sorában több parancs nem lehet.

I a pointer elé karakterek beszúrását teszi lehetővé. A parancsot csak a parancssor végén, vagy egyedül lehet kiadni. A beszúrás a következő sorban kezdhető, a beszúrás végét a ^Z karakter jelöli:

I

MINTABESZÚRÁS1

BESZÚRÁS2

~Z

Ha a karakterek száma több, mint amennyi a memóriába befér, a BREAK felirat jelenik meg.

+—nK n sor törlése a pointertől kezdődően (+ esetén), ill. a pointer előtt (— esetén). Ha a pointer a sor belsejében áll, ezen sor a pointertől törlődik.

+ -nL a pointer mozgatása n sorral előre (+ esetén), vagy hátra (— esetén), a sor elejére. A OL jelentése: az aktuális sor elejére áll a pointer.

nM parancs-makrózás: a sorban utána következő parancsokat n-szer hajtja végre.

A OM jelentése: a parancssort annyiszor hajtja végre, ahányszor a file-ban lehetséges.

Pl.: B0MSMINTA1 ~ZMINTA2~Z

a memória elejétől a végéig az összes MINTA1 karaktersort MINTA2-re cseréli.

Egy utasítássorban több makró is definiálható. Ebben az esetben a makró utáni utasítások a következő makró megadásáig képeznek egy makró operandust.

```
PI.:0MS8~Z9~ZMB0MS9~Z8 Z
```

A példabeli utasítássorozat először kicseréli az összes '8'-ast '9'-esre és utána a puffer elejére áll, majd az összes '9'-est kicseréli '8'-asra.

Speciális az I parancs makrózása, mert a beviendő karaktereket továbbra is mindig a konzolról kéri az utasítás .

.. -
nNMINTA^Z a pointert a MINTA karaktersorozat n-edik előfordulása utánra állítja. Ha a memóriában nem találta meg ezt a helyet, az inputból újabb sorokat olvas be, a memória megtelte után pedig sorokat visz ki az outputba, ha a kijelölt mintát nem találta meg n-szer, a BREAK felirat jelenik meg.

+rP a pointertől kezdve n képernyőt (22 sor) listáz ki a konzolra, előtte mindig 22 sorral áthelyezi a pointert. Egyenértékű az nM22L22T utasítással.
A OP jelentése: a pointertől kezdve 22 sort kilistáz.

Q a szerkesztés befejezése, az input file változatlanul hagyásával, a bevitt módosítások figyelmen kívül hagyásával. A parancs véletlen kiadását elkerülendő ellenőrző kérdést tesz fel az ED, és csak "Y" válasz esetén hajtódik végre. A sorban több parancs nem szerepelhet.

nSMINTA1~ZMINTA2^Z
a MINTA1 karaktersorozatot a pointertől kezdve n-szer megkeresi, és mind az n előfordulásában kicseréli MINTA2-re. A keresés és csere csak a memóriában lévő szövegre érvényes.

+nT a pointertől kezdve n sort listáz ki a konzolra. A T elhagyásával a parancs értelme:

$+_n = H_nLT$

tehát n sort előre (+) vagy hátra (-) lép, és az aktuális sort kiírja.
Speciális esetek:

$\sim M$ "carriage return" = LT

a következő sort kiírja.

$— = —LT$ az előző sort kiírja.

megadja a memóriában levő szabad területet. A parancssorban több parancs nem lehet. Az EO válasza:

XXXXX: EMPTY BYTE

ahol XXXXX decimális szám mutatja a még felhasználható szabad byte-ok számát.

nW a memória elejétől n sort ír az output file-ba. Az utasítás végrehajtása után a puffer elején azon sorszámú sorral kezdődik a szöveg, amely az output file-ba kivitt sorok után közvetlenül található. A #W hatására a memóriában levő összes sor az outputba íródik.

n: az n. sorszámú sor elejére áll a memóriában a pointer.

A fel nem sorolt betűk hatástalanok, a további parancs-funkciókra vannak fenntartva.

Az ED üzenetei:

A program bejelentkezés után az ED a következőket írja a konzolra:

— ED VERSION 1.2 —

és * kiírásával jelzi készségét a parancs elfogadására.

Sikertelen parancsvégrehajtási kísérlet esetén az ED hibaüzenetet ad:

- NOSOURCE FILEPRESENT
az input file nem létezik a kijelölt diszken és van output név megadva.
- NOSPACEONDIRECTORY
a file nem fér föl a lemezre
- OUPUTFILEEXISTS
DELETE?
az outputnak megadott néven már **van** fiJe-ja a kijelölt lemezen, kérésre az ED ezt törli.
- NEW FILE
az input file a kijelölt diszken nem létezik, az ED új file-t generál.
- CATASTROPHICDISCERROR
CHECKING THE DIRECTORY
a diszken file lezárás és átnevezés közben hiba következett be. Ebben az esetben ajánlott a katalógust ellenőrizni és amennyiben lehetséges, a file-okat visszamenteni.
- AREYOUSURE? (Y/N)
a Q parancs ellenőrző kérdése
- WRITE ERROR
disz írás hiba

2.4.5. SUBMIT

A parancs lehetőséget ad gyakran használt parancssorozatok „kötegbe fogására” (batch), egy parancssal való elindítására és beavatkozás nélküli lefuttatására. A parancs csak akkor adható ki, ha az A: lemez az aktuális. A parancs formátuma:

SUBMIT (file-név) (par1) (par2). . .(par9)

Az egyes egységeket egymástól SPACE választja el. A (file-név) egy editorral összeállított, SUB kiterjesztésű file. A kiterjesztést nem szükséges megadni. A file max. 128 sorból állhat, az egyes sorok az UPM parancsállapotában kiadható parancsokat kell, hogy tartalmazzanak. A SUB file-ban szimbolikus paraméter-hivatkozás lehetséges: S1, S2, . . . S9-cel. Az aktuális paraméterek a parancssorban kerülnek megadásra (par1), (par2), . . . , (par9)-cel, ezeket a végrehajtás során a SUBMIT rendre behelyettesíti a S1, S2, . . . ,S9 szimbolikus paraméterek helyébe.

Egy példa a PROBA.SUB file összeállítására:

```
DIR
ERAS1
TYPE S2
DIR B:
```

A file PROBA.SUB néven szerepel az A: lemezen.

A parancsfile futtatása:

```
A>SUBMIT PRÓBA *.SYM PROBA.ASM
```

A SUBMIT bejelentkezik, majd sorban lehívja a PROBA.SUB file-ba lévő sorokat és végrehajtja a parancsokat.

```
A > DIR
    (végrehajtás)
A > ERA» .SYM
    (végrehajtás)
A > TYPE PROBA.ASM
    (végrehajtás)
A > DIR B :
    (végrehajtás)
```

A program minden lehíváskor a fenti feladatsort hajtja végre, a paraméterektől függően törli és listázza a file-okat.

Hibátlan futás esetén a bejelentkező szövegen kívül nincs üzenet.

Bármilyen hiba esetén a SUBMIT futása félbeszakad, hibaüzenet íródik a konzolra, és az UPM újratöltése következik be (JMP 0)

A hibaüzenetek:

- TOO LONGSUBFILE

A SUB file nem fér el a SUBMIT vége (600H) és az UPM kezdete közötti memóriaterületen, vagy a kötegben 128-nál több parancssor van.

Ebben az esetben vagy a SUB file-t kell csökkenteni, vagy a programot nagyobb méretű rendszeren kell futtatni.

- TOO LONG COMMAND LINE

A parancssor max. 127 byte hosszú lehet, éppúgy, mint az UPM-nek adott bármely parancssor.

- INVALIDCONTROLCHARACTER

A megengedett vezérlő karakterektől (^M, "M, ^Z) eltérő vezérlő karakter szerepel a parancssorban.

- INVALID PARAMÉTER

A parancssorban a S jel után álló karakter nem 1 ... 9 közötti.

- FILE NOTFOUND

Nem tudja megnyitni az A drive-on a SUB kiterjesztésű FILENEV file-t.

- NO DIRECTORYSPACE

A SUBMIT nem tud egy átmeneti file-t megnyitni, ami a működéséhez szükséges.

- WRITE ERROR

írási hiba az átmeneti file írásakor.

- CLOSE ERROR

A SUBMIT nem tudja lezárni az átmeneti file-t.

2.4.6. DUMP

A parancs egy file tartalmát írja a konzolra rekordonként. Minden rekord előtt kiírja:

YYY.record:

ahol **YYYY** 0-ról induló logikai rekordszám.

A kiírt rekordban egy sor felépítése a következő:

XXXX D1 D2...D8 A1 A2...A8

ahol

XXXX	0-ról induló címszámláló
D1...D8	a byte-ok hexadecimális értéke
A1...A8	a byte-ok ASCII megfelelője, ha ilyen nincs, akkor oda pont (.) kerül.

A parancs hívása:

DUMP (file-név)

Indítás után a következő módon jelentkezik be:

———TVC/UPM FILE **DUMP** (T2.0)———

A listázási módok a következő parancsbetűkkel választhatók ki:

<SPACE>	az aktuális rekord listázása (szóköz)
N	a következő rekord listázása
P	a megelőző rekord listázása
C	folyamatos listázás, megszakítás (szóköz) karakterrel
RA	rekord listázás adott címszámláló alapján
RR	rekord listázás logikai rekordszámláló alapján
A	ASCII karakterekből álló minta megadása kereséshez
H	hexadecimális karakterekből álló minta megadása a kereséshez

SF	mintakeresés a file elejétől
SN	a következő minta megkeresése
"^C	a DUMP program befejezése
l,?	a parancsbetűk listázása a képernyőn

A sor elején egy csillag (*) jelzi, hogy kész a parancsbetűk elfogadására.

Ha a program a megadott file-t nem találja az aktuális, vagy a kijelölt lemezen, hibaüzenetet ad:

NOTFOUND INPUT FILE

A parancs-"" C-vel megszakítható.

·J C C IU Ali-

2.5. Felhasználói programok

A felhasználó a programok írásakor egyrészt segítséget kap, másrészt bizonyos kötöttségeknek kell eleget tennie. A programírás két szintjét különböztetjük meg:

- magas szintű nyelven írt program
- gépi kódú vagy assembly nyelvű program

A magas szintű nyelven írt programok készítése a programnyelv behívásával kezdődik. Ezután a felhasználó a nyelv szabályai szerint járhat el. A compiler vagy interpreter eleget tesz az UPM követelményeinek: a magas szintű nyelv parancsai olyanok, hogy a kész program futtatható az operációs rendszer alatt.

Assembly nyelven írt programokhoz az UPM részletesebb ismerete szükséges, ehhez célszerű a 3. fejezetet áttanulmányozni. A forrásprogram megírása szerkesztőprogrammal célszerű (pl. ED), ez a megfelelő assemblerrel fordítható le (pl. ASMZ80). Az eredményül kapott hexa file alapján a hibák kijavítása történhet interaktív tesztelő programmal (pl. DDZT). A kijavított HEXA kiterjesztésű file-ból a LOAD program készít futtatható formátumú, COM kiterjesztésű file-t, amit már tranziens parancsként tud értelmezni az UPM.

A felhasználói programok az UPM parancsállapotában a következő módon hívhatók:

(file-név) (operandusok)

A parancssor lezárása után a CCP a kisbetűket nagybetűkké konvertálja. A (file-név) a megfelelő COM kiterjesztésű felhasználói program neve, kiterjesztés nélkül. Ha nem az aktuális lemezen található a hívott program, akkor a megfelelő lemez az A..P betűk egyikével és kettősponttal (:) kijelölhető, közvetlenül a (file-név) előtt.

A felhasználói program neve nem lehet azonos az UPM beépített, vagy tranzien্স parancsainak nevével.

A program hívása után az UPM betölti a felhasználói programot a memóriába a TPA elejétől (100H címtől) kezdődően, az operandusokat pedig elhelyezi a rendszerváltozók (SVA) területén a következő módon:

- az első operandusból az 5CH címtől kezdődően képzí egy file control block (FCB) első 16 byte-ját. Ez file műveletekhez felhasználható (lásd 3.1.3. pont)
- a második operandusból a 6CH címtől kezdődően képzí egy FCB első 16 byte-ját (lásd 3.1.3. pont)
- a 80H címre kerül a parancssorban a (file-név) utáni byte-ok száma, a 81H címtől kezdve pedig elhelyezi ezeket a byte-okat az operációs rendszer. (A parancssort lezáró karaktert nem veszi figyelembe.)

A betöltés után egy CALL 100H utasítással a felhasználói programra kerül a vezérlés. Ha ez nem a 100H címen indul, akkor ide egy JMP utasítást kell írni.

A beépített funkciók használatával lehetőség van hardware-független program írására. Ezek leírását és hívási módját a 3.3.2. pont tartalmazza.

A felhasználói program végén biztosítani kell az operációs rendszerbe való visszatérést. Ennek két módja lehetséges:

- A programot RET utasítással zárjuk, így a program az operációs rendszer szubrutinjaként fog végrehajtódni (ugyanis a programot a CALL 100H utasítás indította).

- A programot a JMP 0 utasítással zárjuk. Ennek hatására a CCP és az UDOS újratöltődik, és az operációs rendszer jelentkezik be.

Példák:

A>COMPARE X.ASM B:X.ASM

Az A: lemezen levő [COMPARE.COM](#) nevű összehasonlító program összehasonlítja az A: és B: lemezen levő X.ASM nevű file-okat.

A>B:MERGE X.HEX Y.HEX B:Z.HEX

- A B: lemezen levő [MERGE.COM](#) nevű, file-okat összefűző program összefűzi az A: lemezeről az X.HEX és Y.HEX nevű
- file-t és Z.HEX néven a B: lemezre teszi.

2.6. Az **UPM** hibáüzenetei

A hibáüzenetekre válaszként megadható ~ M (carriage return) karaktert » jelöli. Hibás parancs-megadás vagy file-hivatkozás esetén az UPM a következő hibáüzeneteket adja:

- a hibás sor visszairása kérdőjellel: szintaktikai hiba (pl. nem megengedett helyen ? vagy * használata; túl nagy SAVE, USER érték megadása stb.)
A sor helyesen megismétlendő!

- NOT FOUND
hivatkozás a lemezen nem létező file-ra

- READERROR
lemezhiba miatt nem olvasható file

WRITE ERROR
lemezhiba miatt nem írható file

- NODIRSPACE
a file már nem fér fel a könyvtárba

- NO SPACE ON DISK
a file nem fér fel a lemezre

- BAD LOAD
felhasználói program sikertelen töltése olvasási hiba miatt

- READ ERROR IN DIRECTORY ON A:
DIR szektor sikertelen olvasása

space válasz esetén az UPM újra próbálja a szektor olvasását
» válaszra kihagyja az olvasásból a hibás szektort

- WRITE ERROR IN DIRECTORY ON A:
DIR szektor sikertelen írása

space válasz esetén az UPM újra próbálja a szektor írását
» válaszra kihagyja az írásból a hibás szektort

- BAD SECTOR IN READ MODE ON A:
sikertelen szektor olvasás

space válasz esetén az UPM újra próbálja a szektor olvasását
» válaszra kihagyja az olvasásból

- SELECT ERROR
UIOS-ban nem definiált drive-ra való hivatkozás

- FILE R/O ON A:
£• file R/O státuszú (írásvédett)

- DISK R/O
a diszk nem írható

"C feloldja az írásvédettséget

- TOOLARGE FILE ON A:
a megengedettnél nagyobb a fi le-méret

- DV FLT
A kijelölt periféria nem kész az adatmozgatásra.

— A programot a JMP 0 utasítással zárjuk. Ennek hatására a CCP és az UDOS újratöltődik, és az operációs rendszer jelentkezik be.

Példák:

A>COMPARE X.ASM B:X.ASM

Az A: lemezen levő [COMPARE.COM](#) nevű összehasonlító program összehasonlítja az A: és B: lemezen levő X.ASM nevű file-okat.

A>B:MERGE X.HEX Y.HEX B:Z.HEX

A B: lemezen levő [MERGE.COM](#) nevű, file-okat összefűző program összefűzi az A: lemezeről az X.HEX és Y.HEX nevű file-t és Z.HEX néven a B: lemezeze teszi.

2.6. Az **UPM** hibáüzenetei

A hibáüzenetekre válaszként megadható "" M (carriage return) karaktert» jelöli. Hibás parancs-megadás vagy file-hivatkozás esetén az UPM a következő hibáüzeneteket adja:

- a hibás sor visszairása kérdőjellel: szintaktikai hiba (pl. nem megengedett helyen ? vagy * használata; túl nagy SAVE, USER érték megadása stb.)
A sor helyesen megismétlendő!
- NOT FOUND
hivatkozás a lemezen nem létező file-ra
- READERROR
lemezhiba miatt nem olvasható file
- WRITE ERROR
lemezhiba miatt nem írható file
- NODIRSPACE
a file már nem fér fel a könyvtárba
- NOSPACEONDISK
a file nem fér fel a lemezeze

BAD LOAD

felhasználói program sikertelen töltése olvasási hiba miatt

READ ERROR IN DIRECTORY ON A:

DIR szektor sikertelen olvasása

space válasz esetén az UPM újra próbálja a szektor olvasását

» válaszra kihagyja az olvasásból a hibás szektort

WRITE ERROR IN DIRECTORY ON A:

DIR szektor sikertelen írása

space válasz esetén az UPM újra próbálja a szektor írását

» válaszra kihagyja az írásból a hibás szektort

BAD SECTOR IN READ MODE ON A:

sikertelen szektor olvasás

space válasz esetén az UPM újra próbálja a szektor olvasását

» válaszra kihagyja az olvasásból

SELECT ERROR

UIOS-ban nem definiált drive-ra való hivatkozás

FILE R/OON A:

c- file R/O státuszú (írásvédett)

DISK R/O

a diszk nem írható

""C feloldja az írásvédettséget

TOOLARGE FILEON A:

a megengedettnél nagyobb a fi le-méret

DV FLT

A kijelölt periféria nem kész az adatmozgatásra.

3. AZ UPM ADATKEZELÉSE

Az operációs rendszer működésének — a rendszer HW függetlensége mellett — 3 feltétele van:

- diszk és konzol periféria
- egy minimális operatív memória: az UPM tárigénye UIOS nélkül 7 Kbyte alatt van, de a rendszerprogramok optimális működéséhez célszerű lehetőleg 64K kiépítésű memória.
- a 0 címen ROM—RAM átkapcsolási lehetőségnek kell lennie, hogy a rendszerváltókat a 0. . .100H tartományon lehessen elhelyezni.

Az operációs rendszer a file-okat 128 byte hosszúságú rekordokból építi fel. A mágneslemezen azonban egy nagyobb egység, a blokk lefoglalása történik. A blokk a konyvtarazott írás alapegysége; a lemezek foglaltsági térképe blokkokra vonatkozik. A blokk mérete az UIOS-ban van definiálva, a rekordokból 2 hatványai szerint épül fel. (Pl. 8" -os floppy esetén 1 blokk = 8 rekord, 5 MB-os lemeznél 1 blokk = 16 rekord.)

Éz a fejezet részletesen ismerteti az adatkezelés módját.

3.1. Az operációs rendszer a memóriában

3.1.1. A memória felosztása

Az operációs rendszer a programmodulból betöltődik az user memóriába. Betöltés után az user memória felosztása a következő:

esrcs	UIOS	FFFFH C-Vi
$\frac{i \ i \ w}{ipll'-t}$	UDOS	EEOOH Cwt
$\frac{s-?M}{5?S«}$	CCP	E00OH sWi\
$\frac{ís29S}{59 \ WS}$	TPA felhasználói terület	D800H 55W6
$\frac{?5t}{255}$	SVA	0100H Z56
0		

Bekapcsoláskor és rendszer RESET esetén betöltődik a CCP, UDOS és UIOS, ezután a rendszer kitölti a rendszerváltozók (SVA) területét.

Az UPM részleges újratöltésénél (^C vagy JMP 0) a lemeztől csak a CCP és az UDOS töltődik be újra, valamint a lemezfoglaltság aktualizálódik.

A felhasználó számára az átmeneti programterület (**TPA**) áll rendelkezésre, ide töltődnek be a lemeztől a programok. A 06H címen található az aktuális UDOS cím, ennek lekérdezésével kapunk információt a felhasználható memória méretéről. A CCP területét a felhasználói program elfoglalhatja, ekkor azonban JMP 0-val kell befejeződnie a programnak, hogy a CCP újratöltődjön.

3.1.2. Rendszerváltozók (SVA)

A memória első 128 byte-ja a rendszerváltozók, paraméterek számára fenntartott terület (O-OFFH). A felhasználói programok elsődlegesen (de nem kizárólag) ezen keresztül tarthatnak kapcsolatot az operációs rendszerrel. Az SVA a következőket tartalmazza:

- 0000. . .0002 H JMP utasítás a rendszer újrabetöltési címével; értéke: UIOS+3
- 0003. . .0003 H rendszer I/O byte
- 0004. . .0004 H a felső 4 bit adja a USER számot (0. . . 15), az alsó 4 bit pedig az aktuális drive-szám; értéke: "A" esetén 0, . . ."P" esetén 15
- 0005. . .0007 H JMP utasítás az operációs rendszer beépített funkcióinak eléréséhez.
- 0008. . .002F H |T-nek fenntartva.
- 0030. . .0037 H a rendszer továbbfejlesztésének fenntartott terület
- 0038. . .003A H IT-nek fenntartva.
- 003B. . .005B H a rendszer továbbfejlesztésének fenntartott terület
- 005C.. .007F H a file control block(CB) területe. A felhasználói programok betöltésekor automatikusan kitölti a CCP az operandusok alapján: egy operandus esetén képi a teljes FCB-t, két operandus esetén 5CH-tól az első, 6CH-tól pedig a második operandusból képzett FCB első 16 byte-ját. (lásd. 3.1.3. pont)
- 0080. . .00FF H puffer-terület a rendszer DMA-nak és a parancsdekódernek
A felhasználói program indítása után az UPM a lekezelt operandust a 80-FFH területre is leteszi: a parancsnév utáni operandus karakterszáma a 80 H-ra, az operandusok pedig a 81 H-tól kezdődően vannak letéve.
Alapértelmezésben ez a terület a DMA műveletek puffer-területe, ide kerül a lemezről az aktuális beolvasott rekord, illetve ez lesz a lemezre felvitt rekord tartalma. Erre a célra a felhasználó más területet is kijelölhet a memóriában. A DIR kóddal visszatérő, file műveleteket végző beépített funkciók használata esetén a kiválasztott katalógus szektor

itt található meg. Az FCB-ben megadott file katalógus sorának a kezdőcíme a pufferben: 80H+DIRKOD«32, ahol a DIR-kód azt jelzi, hogy a katalógus szektorban hányadik a kijelölt file (0. . .3)

3.1.3. File Control Block (FCB)

Az FCB teszi lehetővé a lemezen levő file-okhoz való hozzáférést az operációs rendszerben. Az FCB soros file műveletek esetén 33, sorszám szerinti (random) műveletek esetén 36 byte-ból áll:

dr	f1	f8	ti	t2	t3	ex	s1
00	01	08	09	10	11	12	13
s2	re	dO	dn	cr	rO	r1	r2
14	15	16	31	32	33	34	35

Az egyes mezők használata és jelentése:

- dr Drive kód, értéke aktuális lemezre való hivatkozáskor 0, az A. . .P lemez kijelölésekor rendre 1. . .16.
- f i , . .f8 File-név (ASCII nagybetűkből és számokból). Ha a file neve nem 8 karakteres, a fennmaradó helyekre SPACE (20H) kerül. A legfelső bit f1...f4-nél jelzőbitként használható, f5.. .f8-nál a rendszer számára van fenntartva a későbbi bővítésekhez.
- t 1 . . .t3 A file-név kiterjesztése (ASCII nagybetűkből és számokból). Ha a file kiterjesztése nem 3 karakteres, a fennmaradó helyekre SPACE (20H) kerül. A legfelső bit a file státuszát adja:
- t1-nél: 1 R/O (read/only)
0 R/W (read/write)
- t2-nél 1 SYS
0 DIR
- t3-nál: a további rendszerbővítésre fenntartva.

ex	A file-hoz tartozó könyvtársorok közül az aktuális könyvtársor száma (0. . .31)
s1	Operációs rendszer státuszbyte
s2	Könyvtári kötet száma
re	Rekordszámláló. Értéke megadja, hogy az aktuális könyvtári sor a file hány rekordját tartalmazza.
dO.. .dn	File biokkszámok
cr	Aktuális rekord száma a könyvtári soron belül, soros file-műveletek esetén.
rO. . .r2	Opcionális random rekord szám, ahol rO a legalacsonyabb, r2 a legmagasabb helyiértékű byte. Csak random file-műveletek esetén szükséges a megadása.

A file hivatkozások az FCB kezdőcímének megadásával történnek. A 36 byte-ból a felhasználónak a feladata a 0. . .11 byte kitöltése a kívánt értékkel, valamint a 12. . .15 és a 32. byte kitöltése 0-val. A többi mezőt az operációs rendszer tölti ki a file műveletek során.

Az 1...11 byte-okon a file keresés, megnyitás és törlés funkcióknál megengedett a "?" (3FH) használata. A ?-lel töltött karaktereket a keresés során az operációs rendszer figyelmen kívül hagyja, az első olyan file nevét tekinti aktuálisnak, amelyik a ? kivételével megegyezik az FCB-ben levővel.

3.2. Az operációs rendszer a diszken

Az UPM tetszőleges diszk használatát megengedi. A diszk leírását a diszk paramétertábla tartalmazza az UIOS-ban.

A lemez eleje üres: az üres trackek száma a track offset-ben van megadva.

A lemez következő sávjain a katalógus (directory), majd a file-ok következnek. Egy katalógusrekordon belül 4 könyvtári sor található, ezek egyenként 32 byte hosszúak. A katalógus rekordon (szektoron) belül egy könyvtári sor sorszámát a DIR-kód adja meg (0. . .3).

A katalógussor felépítése:

```
us  f1  . . .  f8  t1  t2  t3  ex  s1  s2  re  dO . . .  dn
00  01  . . .  08  09  10  11  12  13. 14  15  16 . . .  31
```

Az egyes elemek jelentése:

us	A file USER-száma (0.. .15). Ha itt hexadecimális E5 van, akkor az UPM üresnek tekinti a katalógussort.
f i . . . f8	File-név ASCII nagybetűkből és számokból
t i . . . t3	A file nevének kiterjesztése ASCII nagybetűkből és számokból
ex	A file-hoz tartozó könyvtári sorok közül az aktuális sor száma
s1	Operációs rendszer státuszbyte
s2	Könyvtári kötet száma
re	Rekordszámláló. Ez mutatja, hogy a könyvtári sor a file hány rekordját tartalmazza.
dO.. .dn	File blokkszámok

Az operációs rendszer a katalógussor 1. . .31. byte-ját másolja be az aktuális FCB-be file-műveletek esetén.

3.3. UDOS

Az UDOS (UPM Disk Operating System) feladata a perifériák, file-ok és a könyvtár kezelése. Tartalmazza a beépített funkciókat, amelyek segítségével a felhasználók hardware-független programokat írhatnak.

3.3.1. Periféria kezelés

Az UPM-nek logikai periféria-kezelése van: a hivatkozott perifériát a memória a 3H címén található I/O byte határozza meg, az alapján történik az adatforgalom. Az I/O byte-ban bitpárok jelölik ki az egyes logikai perifériákhoz tartozó fizikai perifériákat.

Az 1/0 byte felépítése:

LIST 7-6	PUNCH 5-4	READER 3-2	CONSOL 1-0
-------------	--------------	---------------	---------------

Minden típusú periférián a bitpár 4 értéket vehet fel, eszerint 4 fizikai periféria közül lehet választani.

Logikai periféria	bitpár értéke			
	0	1	2	3
CONSOL (CON)	teletype (TTY)	display (CRT)	(BAT)	felh.def. (UC1)
READER (RDR)	teletype (TTY)	gyorsolv. (PTR)	felh.def. (UR1)	felh.def. (UR2)
PUNCH (PUN)	teletype (TTY)	gyorslyuk. (PTP)	felh. def. (UP1)	felh.def. (UP2)
LIST (LST)	teletype (TTY)	display (CRT)	sornyomt. (LPT)	felh.def. (UL1)

A BAT jelentése: input esetén CONSOL eszköz, output esetén LIST eszköz.

Az UC1, UR1, UR2, UP1, UP2, UL1 eszközöket a felhasználó definiálhatja (pl.: UR1 lehet kazettás mágnesszalag-olvasó, UP1 kazettás mágnesszalagra író).

Az 1/0 byte alapján az UIOS a kijelölt perifériára küldi a karaktert. Az 1/0 byte kezdőértéke az UIOS-ban definiálható. Futás közben a beépített funkciókkal lehet módosítani, az UPM parancsállapotában pedig a STAT parancs-csal.

3.3.2. Beépített funkciók

A beépített funkciók a hardware-független programírás segédeszközei. A funkciókat a CALL 05H hívással lehet elérni.

A funkció sorszáma a C-regiszterben, paramétere a DE, vagy E regiszterben adandó meg a hívás előtt. A visszatérési érték a HL vagy A regiszterben van. Visszatéréskor a regiszterek híváskori értéke megváltozik.

Beépíthető funkciókat a felhasználó is definiálhat, ezek kezdőcímét az UIOS-ban kell megadni. (Ezek a funkciók már nem lesznek CP/M 2.2. kompatibilisek!)

A DIR-kóddal visszatérő funkciók esetén az aktuális DMA címen megtalálható az a könyvtárszektor, amelyben az FCB-vel kijelölt file szerepel. A DIR-kód megadja, hogy ebből a rekordból hányadik 32 byte az aktuális könyvtársor.

A beépített funkciók:

0.: rendszer reset

IN: C = 0

OUT: _____

A rendszer parancsbevitel szintjére kerül, újra inicializálódik, mint a rendszer betöltésekor.

A; A drive a szelektált.

1.: konzol input

IN: C = 1

OUT: A = ASCII karakter

Egy karaktert olvas be a konzolról.

A beolvasott karakter a konzolra is kiíródik.

A~ I, ~P, ~S, ~M, ~J, ~H végrehajtásra kerül.

A karakter beérkezéséig a funkció nem tér vissza a hívási helyre.

2.: konzol output

IN: C = 2

E = ASCII karakter

OUT: _____

Egy karaktert visz ki az E regiszterből a konzolra.

A~I,~P,~S, végrehajtódik.

3.: olvasó input

IN: C = 3

OUT: A = ASCII karakter

Az I/O byte szerinti olvasóról a következő karaktert az A-ba olvassa.

A karakter beérkezéséig a funkció nem tér vissza a hívási helyére.

4.: lyukasztó output

IN: C = 4

E = ASCII karakter

OUT: _____

Az I/O byte szerinti lyukasztóra egy karaktert küld ki az E regiszterből.

5.: lista output

IN: C = 5

E = ASCII karakter

Az I/O byte szerinti list eszközre egy byte kiküldése az E regiszterből.

6.: direkt konzol input-output

IN: C = 6
 E = OFFH input esetén
 = ASCII karakter output esetén
OUT: A = karakter vagy státusz

Egy karakter ki- vagy bevittele, a vezérlő funkciók végrehajtása nélkül.
Input esetén akkor is visszatér, ha nincs készen karakter, ilyenkor A=0.

7.: I/O byte olvasás

IN: C = 7
OUT: A = 1/0 byte

Az aktuális 1/0 byte lekérdezése.

∴ 1/0 byte állítás

IN: C = 8
 E = 1/0 byte
OUT: _____

Az 1/0 byte állítása az E regiszter szerint.

9.: string listázás

IN: C = 9
 DE= string kezdőcíme a memóriában
OUT: _____

A kezdőcímtől a S karakterig (24H) listázza a konzolra a karaktereket

A "N, ^~P, ""S végrehajtódik.

10.: konzol puffer olvasás

IN: C = 10
DE = puffer kezdőcím
OUT: konzol karakterek a pufferben.

A funkcióval egy szerkesztett sort olvashatunk be a konzolról. A DE kezdőcímen megadott konzol puffer felépítése a következő:

	mx	ne	d1	d2	...	dn
DE:	+0	+ 1 + 2	+3	...		+(n+1)

ahol

mx a pufferbe maximálisan beolvasható karakterek száma:
n (1..256)
ne a pufferbe beolvasott aktuális karakterek száma
d1..dn a konzolról beolvasott karakterek

A konzolsor szerkesztése során a következő vezérlő karakterek használhatók:

~C (sor elején), ~H, ~R, ~U, ^X.

A konzolsor lezárása az n. karakter leütése után automatikusan történik, egyébként ^J vagy ^M leütésével. A sorlezáró karakter a pufferbe nem kerül beírásra.

11.: konzol státusz lekérdezés

IN: C = 11
OUT: A = konzol státusz

Ha a konzolon volt leütött karakter, az A=1, egyébként A=0.

12.: verziószám lekérdezés

IN: C = 12
OUT: HL = verziószám

A visszatérési érték jelzi, hogy a futó operációs rendszer alatt melyik verzió funkciói használhatók. Az UPM visszatérési értéke a 2.2 verzióval való kompatibilitás miatt HL=22H. Az ebben írt programok nem feltétlenül futnak a HL<22H visszatérési érték esetén, de biztosan futnak a HL>21H visszatérési érték esetén.

13.: diszk rendszer reset

IN: C = 13
OUT: _____

A funkció a felhasználói programból visszaállítja a rendszert RESET állapotba: minden lemez státuszát írható/olvashatóra (R/W) állítja, az A: lemez lesz az aktuális lemez, a DMA címet pedig 80H-ra állítja. Ezt a funkciót lehet használni pl. akkor, ha a felhasználói programnak diszk cserére van szüksége, rendszer újratöltés nélkül.

14.: aktuális lemez kijelölése

IN: C = 14
E = kijelölt lemez
OUT: _____

Az E regiszterben kijelölünk egy lemezegységet. A foglaltsági té kép aktualizálódik, az új lemez lesz az aktuális. Az E regiszterben 0 jelenti az A:, 1 jelenti a B:15 jelenti a P: lemezt.

A funkció a felhasználói program által használt FCB-ben nem változtatja meg a dr értéket, emiatt dr=0 az új aktuális lemezre fog vonatkozni.

15.: file nyitás

IN: C = 15
DE= FCB kezdőcím
OUT: A= DIR-kód vagy hibajelzés

A diszken lévő file aktiválását végzi file-írás és olvasás előtt. Meg nem nyitott file-hoz nem lehet hozzáférni. Az operációs rendszer a keresést a DE-ben megadott FCB file . .s2 pozícióban álló karakterei alapján végzi. Az file . .f8 pozíciókon van a file neve, a t1 . .t3 pozíciókon a kiterjesztése, ASCII nagybetűkből és számokból. Az üres helyekre itt SPACE (20H) kerül. A többi pozíciót normális esetben 0-val kell kitölteni.

Az file . .t3 pozíció bármelyikén megengedett a "?" (3FH) használata, ez tetszőleges karaktert jelöl. A funkció ekkor az első olyan file-t fogja megnyitni, amelynek neve a ? kivételével azonos az FCB-ben megadott névvel. Ha az FCB alapján az UDOS megtalálja a kijelölt file-t, a katalógusból átmásolja a könyvtársort, a DE-vel kijelölt FCB-be. A visszatérési érték a file megnyitása esetén a DIR-kód, sikertelen megnyitása esetén OFFH.

A DIR kód nem más mint a katalógus tétel sorszáma az adott katalógusrekordon belül (0 . .3).

16.: file zárás

IN: C = 16
DE = FCB kezdőcím
OUT: A= DIR-kód vagy hibajelzés

A funkció megfelelő katalógussorban másolja a DE-ben megadott FCB-ből a szükséges adatokat (re. .dn pozíciók) és lezárja a katalógussort. Minden megnyitott file-t az írás befejezése után le kell zárni. A megfelelő katalógussor keresését a funkció a DE-ben megadott FCB file . .s2 pozíciója alapján végzi. A "?" (3FH) használható az f1 . .t3 pozíciók bármelyikén a 15. funkciónál leírt értelmezés szerint.

Visszatérési érték sikeres lezárás esetén a 0 . .3 értékű DIR-kód, vagy sikertelenség esetén OFFH.

17.: file keresés a könyvtár elejétől

IN: C = 17
 DE = FCB kezdőcím
OUT: A = DIR-kód vagy hibajelzés

Az FCB-ben megadott nevű file-t keres a könyvtárban. Ha megtalálta, bemásolja a könyvtári szektort az aktuális DMA címre és kitölti a visszatérési kódot a megtalált file DIR-szektoron belüli sorszámaival, a DIR-kóddal.

Sikertelen keresés esetén a visszatérési érték A=OFFH. A "?" (3FH) használható az fi. .t3 pozíciók bármelyikén, a 15. funkciónál leírt értelmezés szerint.

18.: file keresés adott belépési ponttól

IN: C = 18
 DE = FCB kezdőcím
OUT: A = DIR-kód, vagy hibajelzés

Egyező az előző funkcióval, de nem a könyvtár elején kezdi a keresést, hanem ott, ahol előzőleg abbahagyta egy 17-es, vagy 18-as funkcióhívás.

19.: file törlés

IN: C = 19
 DE = FCB kezdőcím
OUT: A = DIR-kód, vagy hibajelzés

Az FCB fi. .t3 pozíciói alapján a megfelelő katalógussorokat kitörli a lemezről. A "?" (3FH) használható az f1. .t3 pozíciókban. Értelmezése azonos a 15. funkciónál leírtakkal.

20.: sorrendi olvasás

IN: C = 2 0
 DE = **FCB** kezdőcím
OUT: A • visszatérési kód

A 15. vagy a 22. funkcióval megnyitott file olvasását végzi: beolvassa azt a rekordot az aktuális **DMA** címre, amelyre az FCB-ben a cr számláló mutat. A beolvasás után a cr számlálót eggyel növeli: a cr számláló a file következő rekordjára fog mutatni. Ha szükséges, akkor automatikusan elvégzi a file következő könyvtári sorára az áttérést.

Sikeres olvasás esetén visszatéréskor A=0, file-on kívüli olvasás esetén (EOF) A=1.

21.: sorrendi írás

IN: C = 21
 DE = FCB kezdőcím
OUT: A = visszatérési kód

A 15. vagy a 22. funkcióval megnyitott file-ba ír aktuális DMA címről. Az írás a cr számlálóval kijelölt rekordba történik. írás után a cr számlálót eggyel növeli. Ha a megnövelt cr számláló túlmutat az aktuális könyvtársoron, akkor a könyvtársort lezárja és új könyvtársort nyit az írás folytatásához. Egyéb esetben a 16. funkcióval kell az írás befejezése után a változtatásokat a könyvtársorban dokumentálni.

Sikeres írás esetén visszatéréskor A=0, egyébként A=1 file-on kívüli hivatkozás, A=2 a diszk tele van.

22.: file létrehozás

IN: C = 22
 DE = FCB kezdőcím
OUT: A = DI R kód, vagy hibajelzés

A funkció létrehoz egy üres file-t a katalógus első sorában az FCB alapján. Nem vizsgálja, hogy az adott néven van-e már föle a katalógusban. Használata előtt célszerű a 19. funkcióval előbb egy törlést végezni, ezzel elkerülhetjük, hogy több file is ugyanazt a nevet kapja. A létrehozott file-t a funkció meg is nyitja íráshoz/olvasáshoz, ezért utána a 15. funkció hívása szükségtelen.

Sikertelen végrehajtás esetén a visszatérési érték: A=OFFH, egyébként pedig A=0. . .3 között, a file DIR-szektorbeli sorszáma szerint.

23.: file átnevezés

IN: C = 23
DE= FCB kezdőcím
OUT: A= DIR-kód vagy hibajelzés

Ebben a funkcióban az FCB-t speciálisan kell kitölteni. Az első 16 byte az átnevezésre kijelölt file-ra, a második 16 byte az új nevű file-ra vonatkozik. Mindkettőbe az FCB-ből a dr. . .re pozícióknak megfelelő adatok kerülnek. A két dr pozícióban azonos lemezre kell hivatkozni, az f1. . .t3 pozíciókba kerül a file eredeti ill. új neve, az ex, s1, s2 és re pozíciókba 0-t kell írni. A funkció hatására az átnevezés az eredeti file összes katalógus-sorában végrehajtódik.

Ha nem találta meg a file-t, a visszatérési érték: A=OFFH, egyébként

24.: drive on-line státusz lekérdezés

IN: C = 24
OUT: HL = on-line státusz
(drive • diszk meghajtó)

A HL regiszter bitjei közt lévő 1-esek jelölik az on-line drive-okat. Az L regiszter legelső bitje az A drive-nak, a H regiszter legfelső bitje a P drive-nak felel meg.

Egy drive akkor van on-line állapotban, ha az utolsó rendszerinicializálás óta volt szelektálva, azaz aktualizálódott a foglaltsági térképe.

25.: aktuális drive lekérdezés

IN: C = 2 5
OUT: A = az aktuális drive kódja
(drive = diszk meghajtó)

A funkció az aktuális drive kódját adja az A regiszterben: a 0 felel meg az A drive-nak, a 15 a P drive-nak.

26.: DMA címbeállítás

IN: C = 2 6
DE = DMA cím
OUT: _____

Annak a címnek a beállítására szolgál, ahová a lemezolvasási funkció olvas, ill. ahonnan az írás-funkció veszi a felírandó karaktereket. (Nem takar feltétlenül tényleges DMA funkciót.) A DMA cím értékét az operációs rendszer 80H-ra állítja reset vagy betöltés után.

27.: diszk foglaltsági térkép kezdőcímének lekérdezése

IN: C = 2 7
OUT: HL = diszk foglaltsági térkép kezdőcíme

A funkció az aktuális lemezen még meglévő szabad hely meghatározását teszi lehetővé. A foglaltsági térképet a ^C aktualizálja. Egy adott blokk foglaltságát 1 értékű bit jelenti. A lemezen lévő blokkok a foglaltsági bitekkel sorrendben vannak összerendelve.

28.: írástiltás a diszken

IN: C = 2 8
OUT: _____

A funkció az aktuális diszkre ideiglenes, a következő rendszer-újrátöltésig érvényes írásvédelmet biztosít. írási kísérlet esetén az operációs rendszer hibajelzéssel figyelmeztet az írástiltás tényére.

29.: drive R/O lekérdezés

IN: C = 29
OUT: HL = R/O vektor

(drive = diszk meghajtó)

A HL regiszterpár 1 értékű bitjei jelentik az adott drive-ok csak olvasható állapotát: az L regiszter legalsó bitje tartozik az A drive-hoz, a H regiszter legfelső bitje a P drive-hoz. Az operációs rendszer lemezcseré, vagy C esetén állítja a biteket, a felhasználó a 28. funkcióval.

30.: file státusz állítás

IN: C = 30
DE = FCB kezdőcím
OUT: A = DIR-kód vagy hibajelzés

A funkció a DE-ben megadott FCB szerint a hivatkozott file Read Only, SYS file és felhasználói státuszát állítja az f 1 . . .t3 pozíciók legfelső bitjei szerint.

Sikertelen végrehajtás esetén visszatéréskor A=0FFH, egyébként A=0, 3 értékű.

31.: diszk paraméter-tábla kezdőcímének lekérdezése

IN: C = 31
OUT: HL = kezdőcím

A funkció az aktuális diszk-paraméterek lekérdezésére vagy pillanatnyi átírására ad lehetőséget, a leíró blokk kezdőcímének megadásával.

32.: USER kód lekérdezés és állítás

IN: C = 3 2
E = OF FH lekérdezés esetén
E = USER kód állítás esetén (0..15)
OUT: A = aktuális USER kód lekérdezés esetén
A = érdektelen állítás esetén

Az aktuális USER kód lekérdezését vagy megváltoztatását teszi lehetővé a program futásának idejére.

33.: file random olvasás

IN: C = 3 3
DE = FCB kezdőcím
OUT: A = visszatérési kód

A funkció beolvas egy rekordot az FCB-ben megadott file-ból logikai rekordszám alapján. Az r0, r1, r2 pozíciókba kell írni a rekordszámot: r0 a legalacsonyabb, r2 a legmagasabb helyiértékű byte. A kijelölt rekordot az aktuális DMA címre olvassa be. Beolvasás után a funkció beállítja az aktuális rekordnak megfelelően az FCB ex. . .cr pozícióit, ezzel lehetőséget ad a soros írás/olvasás funkciókra való áttérésre. A soros írás/olvasás funkciókkal ellentétben a cr számláló a beolvasott (és nem a következő!) rekordra mutat.

Az r0, r1, r2 számláló csökkentését vagy növelését a programozónak kell elvégeznie.

Visszatéréskor az A=0 érték jelenti a sikeres végrehajtást, egyébként:

A=1 nem felírt blokk olvasási kérése

A=3 sikertelen lezárás

A=4 nem létező rekord keresése

34.: file random írás

IN: C = 34
DE = FCB kezdőcím
JT: A = visszatérési kód

A funkció az FCB-ben megadott file-ba ír egy rekordot logikai rekordszám alapján. A logikai rekordszámot az r0. .r2 pozíciókba kell írni: r0 a legalacsonyabb r2 a legmagasabb helyiértékű byte. A file-ba írás az aktuális DMA címről történik. Felírás után az aktuális rekordnak megfelelően beállítódik az FCB ex..cr pozíciója. Ezzel lehetőség nyílik a soros írás/olvasás funkciókra való áttérésre. A soros írás/olvasás funkciókkal ellentétben a cr számláló a felírt (és nem a következő!) rekordra mutat. Kivételt képez az, amikor a könyvtári sor utolsó rekordjára történik az írás. Ebben az esetben a funkció lezárja az aktuális könyvtársort és megnyitja a következőt, ezzel lehetőséget ad a file következő rekordjához való hozzáférésre. Egyéb esetben a 16. funkcióval kell a változtatásokat a könyvtársorban dokumentálni.

Random írás esetén a file logikai méretét nem a ténylegesen elfoglalt rekordok száma, hanem az utolsó és a 0. rekordok közötti különbség adja.

Az r0..r2 számláló csökkentését vagy növelését a programozónak kell elvégeznie.

Sikeres végrehajtás esetén A=0 a visszatérési érték.

A hibajelzések:

A=3 sikertelen lezárás

A=4 nem létező rekord keresése

A=5 a könyvtár megtelt

35.: file méret számítás

IN: C = 35
DE = FCB kezdőcím
OUT: _____

A funkció a DE regiszterben megadott FCB random mezőjét állítja be. A név alapján megkeresett file utolsó utáni rekordjának címe kerül az rO.. .r2 byte-okba, lehetőséget adva ezzel a file vége utáni írásra vagy sorosan írt file esetén a fizikai méret számítására.

36.: random mező beállítás

IN: C = 3 6
DE = FCB kezdőcím
OUT: _____

A funkció a DE regiszterben megadott FCB ex, s2 és cr pozíciói alapján kitölti az rO.. .r2 byte-okat az aktuális rekordnak megfelelően, lehetőséget adva ezzel random operációra való áttérésre soros file-operációról.

240,241.:

IN: C = 240. .241

A 240-es funkció a TVC-ben különleges konzol műveletekre foglalt.

A bővített funkciók leírása a 4. fejezetben található.

A 241-es funkció paramétereit és jellemzőit a felhasználó definiálhatja saját igényei szerint.

3.4. UIOS

Az UIOS (UPM Input-Output System) feladata, hogy a hardware környezet-höz a közvetlen illeszkedést biztosítsa.

Felépítése a következő:

UIOS kezdőcím
ugró tábla
UIOS program és
adatterület

3.4.1. Az UIOS ugrótáblája

Az **UIOS** ugrótáblája az **UIOS** program elején található:

CIM		FUNKCIÓ
UIOS	JMP BOOT:	rendszer inicializálás
UIOS+03	JMPWBOOT:	rendszer betöltés, inicializálás
UIOS+06	JMP CSTS:	konzol státusz lekérdezés
UIOS+09	JMP CI:	konzol karakter bolvasás
UIOS+0CH	JMPCO:	konzol karakter kivitel
UIOS+0FH	JMP LO:	list eszközre karakter kivitel
UIOS+12H	JMP PO:	lyukasztóra karakter kivitel
UIOS+15H	JMP RI:	lyukszalag-olvasóról karakter bevétel
UIOS+18H	JMP HOME:	diszk fej 0. track-re
UIOS+1BH	JMPSELDSK:	diszk szelektálás
UIOS+1EH	JMPSETTRK:	track-szám beállítás
UIOS+21H	JMP SETSEC:	szektorszám beállítás
UIOS+24H	JMPSETDMA:	DMA-cím beállítás
UIOS+27H	JMPDREAD:	diszk olvasás
UIOS+2AH	JMPDWRITE:	disz írás
UIOS+2DH	JMP LISTST:	nyomtató státusz lekérdezés
UIOS+30H	JMP SECTTRAN	szektor translációs tábla kezdőcím lekérdezés
UIOS+45H	JMP. :	A felhasználó által definiált ugrási cím a MON beépített programhoz.
UIOS+48H	JMPHVIDEO:	bővített konzol funkciók.
UIOS+4BH	<u>JMP.</u>	a felhasználó által definiált UDOS funkció kezdőcíme sorrendben, ha ilyen nem létezik, a memóriaterület szabadon felhasználható.

3.4.2. UIOS funkciók

Az **UIOS** ugrótábláján keresztül érhetőek el ezek a funkciók. Az egyes funkciók a következő feladatokat látják el:

BOOT: feladata az operációs rendszer inicializálása: be kell állítania az aktuális kezdőértékeket a 0. . .7 címeken

0H JMPUIOS+3

3H I/O byte kezdőérték

4H aktuális diszk-szám

5H JPM UDOS+6

A bejelentkező szöveg kiírása után az operációs rendszerbe kell ugrania.

A C regiszterben a 4H címen lévő aktuális diszk érték megfelelő kódjának kell lennie: A drive esetén 0, P drive esetén 15.

WBOOT: feladata a CCP és az UDOS betöltése és inicializálása a BOOT funkció szerint.

CSTS: konzol státusz lekérdezés:

— ha van beolvasásra kész karakter: A = OFFH

— ha nincs beolvasásra kész karakter: A = 0

CI: az A regiszterbe beolvas egy karaktert a konzolról, — ha nincs, vár rá.

CO a C regiszterből egy ASCII karaktert visz ki az I/O byte szerinti konzolra.

Természetesen a programnak tartalmaznia kell a konzol által megkívánt különleges igények kielégítését, pl. időzítést.

LO: a C regiszterből egy ASCII karaktert küld ki az I/O byte által kijelölt **LIST** eszközre.

a C regiszterből egy ASCII karaktert visz ki az 1/0 byte szerinti PUNCH eszközzre.

az A regiszterbe beolvassa a következő karaktert a lyukszalag-olvasóról. EOF esetén a rutinnak ^-Z-t kell adnia.

az aktuális diszk fejet a 0. track pozícióba állítja.

a C regiszter tartalma alapján szelektálja a drive-okat:

C = 0 esetén A drive

C = 1 esetén B drive

C = 15 esetén P drive

Visszatéréskor a HL regiszterben a diszk paramétertábla aktuális sorának kezdőcímét adja a handlernek. Ha a C-ben nem létező drive-ra történik hivatkozás, visszatéréskor HL=0 értéknek kell lennie.

A BC regiszterpárban lévő érték alapján az aktuális drive-on kijelöli a track számot.

A BC regiszterben lévő érték alapján az aktuális drive-on kijelöli a szektorszámot.

A BC regiszterben lévő érték alapján kijelöli annak a 80H byte-os területnek a .kezdőcímét, ahová, illetve ahonnan a diszk az írást, illetve az olvasást végzi.

A kijelölt track- és szektorszám, ill. a DMA cím alapján egy rekordot leolvass a diszkről. Sikeres olvasás esetén a visszatérési érték A=0, egyébként A nem 0.

A kijelölt track- és szektorszám, illetve a DMA cím alapján egy rekordot ír a diszkre. Sikeres írás esetén a visszatérési érték A=0, egyébként A nem 0.

LISTST: Az A regiszter 0 értéke jelenti, hogy a LIST eszköz kész karakter fogadására, az A=OFFH pedig a foglaltságot jelzi.

SECTTRAN: A BC regiszterben lévő logikai szektorszámból számítja ki és adja vissza a HL regiszterben a tényleges fizikai szektorszámot. A DE regiszterpárban kell megadni a szektor eltolási vektor kezdőcímét.

3.4.3. Diszk paramétertábla

Az UPM tetszőleges felépítésű diszket tud kezelni, a drive leírását, paramétereit azonban meg kell adni az UIOS-ban. A paramétertábla egy-egy sora 16 byte-ból áll. Egy sor egy diszk meghajtóra vonatkozik: ez a diszk paraméter vektor. A diszk szelektáló szubrutin (SELDSK) feladata az, hogy a HL regiszterben megadja a diszk paramétertáblából az aktuális diszk paraméter vektor kezdőcímét.

A diszk paraméter vektor 2 byte hosszú szavakból áll.

Felépítése a következő:

TRV	00. ..00	DIRBUF	DPB	CSV	ALV
00,01	02. ..07	08,09	10,11	12,13	14,15

ahol:

TRV Az eltolási vektor kezdőcíme (0, ha nincs eltolás). A vektor a diszk egy sávjában (track) a logikai rekordszámhoz fizikai rekordszámot rendel.

Azonos típusú lemezekhez megadható közös eltolási vektor, ha az eltolási érték is egyezik.

A TVC-nél nincs eltolás.

00: Az UDOS számára fenntartva.

- DIRBUF: 128 byte méretű puffer kezdőcíme, amit az UDOS a könyvtárműveletekhez használ. Az összes diszkhez megadható ugyanaz a puffer.
- DPB: Diszk paraméter blokk kezdőcíme. Azonos karakterisztikájú lemezekhez megadható egy közös blokk. A blokk felépítését ennek a pontnak a végén ismertetjük.
- CSV: Kontroll szumma vektor kezdőcíme. Minden lemezhez külön vektort kell megadni.
- ALV: Diszk foglaltsági térkép kezdőcíme. Minden lemezhez külön foglaltsági térképet kell megadni. A térkép egyes bitjei rendre megfelelnek a lemez egyes adatblokkjainak. A foglaltsági térképet az UDOS kezeli.

A diszk paraméter blokk (DPB) hossza 15 byte, 1 és 2 byte-os elemekből áll. Felépítése a következő:

SPT	BSH	BLM	EXM	DSM	DRM	DAL	CKS	OFF
00,01	02	03	04	05,06	07,08	09,10	11,12	13,14

Az egyes elemek jelentése a következő:

- SPT Az egy sávon (track) levő szektorok száma.
- BSH Blokk eltolási faktor. Értékét a rekordokban megadott blokkméret (BLS) határozza meg:

$$\text{BLS} = 2 \quad (\text{BSH})$$

- BLM Blokk maszk. Értékét a rekordokban adott blokkméret (BLS) határozza meg:

$$\text{BLM} = \text{BLS} - 1$$

EXM	Könyvtári sor (extent) maszk. Értékét a rekordokban adott blokkméret (BLS) határozza meg. Ha $DSM > 255$ és $BLS = 16$, akkor $EXM = 0$
DSM	A lemezre írható utolsó adatblokk sorszáma. Az adatblokkok számozása 0-val kezdődik.
DRM	Az utolsó könyvtári sor sorszáma a lemezen. A könyvtársorok számozása 0-val kezdődik.
DAL	A könyvtárnak lefoglalt blokkokat jelzi. A legnagyobb helyi-értéktől kezdve annyi bit 1-es, ahány blokkból áll a könyvtár.
CKS	A könyvtár kontroll vektorának mérete. Értéke a könyvtári szektorok számával azonos.
OFF	A rendszer által lefoglalt sávok száma (track offset).

3.4.4. Floppy diszkkal kapcsolatos tudnivalók

A TV-Computerhez 4 db 5, 1/4"-os kétoldalas, dupla sűrűségű floppy meghajtó csatlakoztatható. A VIDEOTON ugyanazt a floppy meghajtó típust szállítja a TV-Computerhez, mint a VT—16 számítógéphez, és a floppy-t mindkét gépen azonos módon használjuk, tehát az UPM rendszerű könyvtárakat tartalmazó lemez a két gép között cserélhető.

A lemez 512 byte-os szektorokat tartalmaz, sávonként 9 db-ot, cilinderenként 18-at, azaz a "szektor/track" értéke 72, mivel az 128 byte-os CP/M szektorra vonatkozik.

Ezt figyelembe véve az "offset" értéke 2, azaz a lemez elején 36 db 512 byte-os szektor kihasználatlanul marad. Ez a VT—16 kompatibilitás miatt van így. Ez a két sáv azért van fenntartva, mert a hagyományos CP/M rendszerű lemezeknél az operációs rendszer a hideg és meleg boot eljárás során a lemez 0. sávjárói töltődik be. A TV-Computer esetén ez nincs kihasználva, mivel a boot eljárás alaphelyzetben a ROM kazettából tölt. Ezeket a kihasználatlan

szektorokat a felhasználó speciális célokra használhatja, ha az UIOS DWRITE, DREAD rutinjait hívja. Természetesen előtte meg kell hívni a SETDMA, de SETSEC, SETTRK, SETDSK rutinokat is.

Az 512 byte-os szektorok fölbontását (ill. összeállítását) a szokásos szektor blokkoló-deblokkoló algoritmus (az UIOS része) végzi.

További inicializálási adatok a floppy-ról:

blokk méret	:	2048 byte
diszk méret		351 blokk (= 702 kbyte)
directory méret		128 tétel.

3.4.5. UPM konzol a TV-Computeren

Az UPM konzol a BASIC alatt működő TVC-konzolon alapszik, annak módosítása. Jellemzően 24 sor x 64 karakter üzemmódban működik, de képes kezelni 32 és 16 karakteres képernyőt is.

A konzol output nem tartalmazza a BASIC-ben megszokott „logikai sor” kezelést. Az így kialakított konzol jobban hasonlít a nagyobb gépeken megszokott változatokra. Ha egy képernyősor megtelik, akkor a következő sor elején folytatódik a kiírás. Ha a képernyő utolsó pozíciójába írunk, akkor minden sor eggyel feljebb lép, azaz a képernyő „rollozódik”.

Karakter törlésnél a kitörölt karaktertől jobbra levő karakterek eggyel balra lépnek. Karakter beszúrásnál a kurzortól kezdve a karakterek eggyel jobbra lépnek a sor végén levő karakter elveszik. A sortörlés a kurzorsort követő sorokat eggyel feljebb mozdítja. Sorbeszúrásnál a kurzortól kezdve a sorok eggyel lejjebb lépnek, a legalsó sor elveszik. Mindkét esetben a kurzor a sor bal szélére kerül.

Konzol output vezérlőkódok:

" 04h	Kurzor jobbra
5 05h	Kurzor föl
? 07h DC	Karakter törlés a kurzor alatt

? 08h	DEL	Karakter törlés a kurzortól balra
◀ 09h	TAB	Tabulátor karakter. A tabulációs pozíciók a képernyő 1, 9, 17, 25, 33, 41, 49 és 57-es oszlopában vannak
¹⁰ 0Ah	LF	Soremelés. A logikai sort követő sor azonos oszlopába megy a kurzor.
'< 0Bh	CEL	Törlés a sor végéig
U 0Ch	HOME	A kurzor az 1. sor 1. pozícióba megy
'- 0Dh	CR	A kurzor a képernyősor elejére megy
''' 0Eh	IL	Sorbeszúrás
(fe 10h	CURADR	Kurzorcímzés kezdőkódja. Ezt követi a sor és az oszlop kódja. A képernyő bal felső pozíciója az 1. sor 1. oszlop. A 0. sorpozíció ill. 0. oszloppozíció a megfelelő pozíciót változatlanul hagyja. A kiküldendő kódok: sorpozíció +20h oszloppozíció + 20h
<i>ti</i> 11h	CFLASH	Kurzor villogás engedélyezése
<i>ti</i> 12h	CNOFLS	Kurzor villogás tiltása, a kurzor nem látható
<i>to</i> 13h	< -	Kurzor balra
16h	INS	Egy szóköz karakter beszúrása a kurzorpozícióba
18h		Kurzor le
19h	DL	Sortörlés
1Fh	CLEAR	Képernyőtörlés. A kurzor a bal felső sarokba áll

A fix karakterek ^{II 453} 20h-9Fh, a szabadon definiálható bővítés karakterek OAOh—OFFh kódtartományba esnek.

Figyelem: A OEOh-OFFh kódú karakterek kiírathatók, de a billentyűzet-ről nem lehet őket generálni.

3.4.6. Memória Flagekkel változtatható paraméterek

WBFL(=0EE56H)

Meleg-boot flag. Értéke alaphelyzetben 0, ez azt jelenti, hogy a meleg boot ROM-ból történik. Átírható 16H értékre, ebben az esetben a meleg boot egy UPMSYS.SYS nevű file-ból tölti az operációs rendszert. A file 1600H byte hosszú, és a 0D800H-ra töltendő memóriaképet kell tartalmaznia.

WBUNIT(=0EE57H)

Meleg boot floppy diszk egysége. Alaphelyzetben COH (= "A" lemez).

ERFLAG (=0EE59H)

Hibakódot tartalmaz a DWRITE, DREAD hívások után:

- 0 = írás-olvasás rendben
- 1 = CRC hiba
- 2 = write fault
- 3 = seek error
- 5 = record not found
(nem találja a kért számú szektort)
- 6 = lost data (adat vesztes)
- 7 = írásvédelem megsértési kísérlet
- 8 = drive not ready
- 10 = csatoló hiányzik.

HPUF (= EE5AH)

Ezen a címen megtalálható a hőst puffer címe, tehát ahol az egész 512 byte-os szektor egyben megvan.

4. UPM BEÉPÍTETT FUNKCIÓK BŐVÍTÉSE

4.1. Bevezetés

Az UPM operációs rendszer működéséhez egyszerű karakteres képernyő is elegendő. A TV-Computer azonban rendelkezik olyan hardverrel, amely nagyfelbontású színes grafikus képet tud kezelni.

BASIC alatt a VIDEÓ rutinokkal kényelmesen tudunk ábrát, rajzot készíteni. Ugyanerre a célra, azaz rajzok ábrák készítéséhez, a grafikus lehetőségek eléréséhez ezt a rutincsomagot beépítettük az UPM-be, kiegészítve még néhány rendszerváltozót beállító és lekérdező rutinnal.

A bővítést a 240-es számú beépített funkció tartalmazza, hívása CALL 05h utasítással történik. A processzor regisztereinek kiosztása a következő:

híváskor:

C = 240
I = 0..15 konkrét rutin száma
DE, IX, IY = hívási paraméterek

visszatéréskor:

A, HL, IX, IY = visszatérési paraméterek

A rutinok nem mentik el a regisztereket, erről a hívó félnek kell gondoskodnia.

A bővítés — ellentétben a többi beépített funkcióval — nem használ saját vermet (STACK), hanem a hívó program veremtár-mutatóját (STACK POINTER) használja. Ez a tény két megszorítást jelent a hívó program számára:

1. A verem csak meghatározott címtartományban lehet a bővítés híváskor:

4000h - = 7FFFh
vagy 0C000h-0D7FFh

2. A vermet a bővítés rutinjai is mélyítik. Ajánlatos erre legalább 128 bájtot fenntartani.

Kivételes a FILL rutin működése (hívás: 1=10). Ez verem-orientált, emiatt előfordulhat a 128 bájtot meghaladó verem-mélyítés is.

A rutin csupán az ellen van védve, hogy a vermet ne mélyítse 0C000h illetve 4000h alá. Emiatt a **FILL** rutin használata esetén a vermet ajánlatos az 1. pontban adott tartományok alsó határán elhelyezni.

Általában is igaz az 1. pontban leírt címtartomány-korlátozás azokra a rutinokra, amelyek RAM puffert használnak:

- string kiírás (hívás: l=2)
- bővítés karaktergenerátor definiálás (hívás: **1=11**)
- fix karaktergenerátor definiálás (hívás: **1=15**)

4.2. Áttekintés

A TV-Computer bit-térkép szervezésű grafikus megjelenítéssel rendelkezik három különböző színelbontásban. A képernyő felbontása 512X240 pont. A grafikus és szöveges információk nincsenek a képernyőn megkülönböztetve. Karakterek a képernyőn bárhol megjeleníthetők, ehhez először a képpont, aza;: a sugár pozícióját kell beállítani, majd a karaktert kiíratni a bővítés megfelelő rutinjaival. A karakter az aktuális színelbontásnak megfelelően fog megjelenni.

A karakterek 8X10 bitből állnak (készínű mód). Egy sorba kétszínű módban 64 (bekapcsolás és RESET után), négy színű módban 32, tizenhat színű módban pedig 16 db karakter írható. A függőleges felbontás minden módban 24 szövegsor. A grafikai koordináták mindhárom felbontásban 1024X960 értékre vannak normalizálva, vízszintesen 0. . .1023, függőlegesen 0. . .959 értékig használhatók. Finom scroll és oldalirányú scroll lehetőségek nincsenek biztosítva. A karakteres és grafikus képernyőt egyszerre 10 pontsorral tudja a szoftver léptetni.

A videó rutin a következő rendszerváltozókat használja:

SL_MODE	vonalmód
SL_STYLE	vonaltípus
SV_FLAG	karakter felülírás
SÍNK	tintasín (közös a konzol outputtal)
SPAPER	papírsín (közös a konzol outputtal)

Ezeket a változókat tetszés szerint át lehet írni. Használat előtt a rutinok kiválasztják az értékes biteket, ezért nem okoz hibát, ha túl nagy az értékük.

Lekérdezhető még:

X koordináta
Y koordináta
sugár állapot

Egyéb elérhető rendszerváltozók:

SBORDER	képernyő keretszín
COLOUR	képernyő színmód
DELAYKEY	konzol input időzítés
RATE_KEY	konzol input időzítés
EXTFONT	bővítés karaktergenerátor kezdőcíme ^közös a konzol outputtá ¹
FIXFONT	fix karakter generátor kezdőcíme

SL_MODE

Az aktuális pontkiírás illetve vonalhúzás módját adja. A megfelelő logikai műveletet a tintasín és az aktuális képpont eredeti színének bite vagy bitjei között hajtja végre. Csak az alsó két bitet veszi figyelembe:

- 0 - felülírás
- 1 - logikai VAGY (OR) művelet
- 2 - logikai ÉS (AND) művelet
- 3 - logikai KIZÁRÓ VAGY (XOR) művelet

SL_STYLE

Vonalhúzás esetén az aktuális vonal típusát adja. 16 féle vonalat ismer a bővítés. A vonal lehet folyamatos, illetve állhat pontokból és vonalakból. A folyamatos vonal tintaszínű lesz, míg a többi váltakozva tintaszínű és háttérszínű, a mintázatnak megfelelően. Az alábbi táblázat mutatja a vonaltípusokat ("—" jel jelenti a tintaszínt):

1 (—————)	9 (—————)
2 (—————)	10 (—————)
3 (— —)	11 (—————)
4 (—————)	12 (—————)
5 (-)	13 (—————)
6 (—————)	14 (—————)
7 (—————)	15 (—————)
8 (—————)	16 (—————)

SV FLAG

Meghatározza, hogy a bővítés rutin karakterkiírásnál a tinta és papírszínű pontokat ki kell-e írni. A 0. bit a tintaszínre, az 1. bit a papírszínre vonatkozik. Ha a bit értéke zérus, akkor a pontot ki kell írni, ha értéke 1, akkor nem kell kiírni. Gyakorlatilag ez annyit jelent, hogy a karaktert ki lehet írni az öt körülvevő háttérpontok nélkül, illetve fordítva.

SINKésSPAPER

Kétszínű módban a 0. bit jelenti a palettaregisztert, négy színű módban az 1. és 0. bit adja a palettaregisztert, tizenhatszínű módban a 3. 2. 1. és 0. bit adja közvetlenül a színt: 0. bit=kék, 1. bit = vörös, 2. bit = zöld, 3. bit = fényes szín. Ezek alapján a megjeleníthető 16 színhez tartozó értékek:

0 - fekete	4 — sötétzöld	8 - fekete	12 - zöld
1 - sötétkék	5 — sötét cián	9 - kék	13 — cián
2 — sötét vörös	6 — sötét sárga	10 — vörös	14 - sárga
3 - sötétlila	7 — szürke	11 - lila	15 - fehér

SBORDER

A képernyő keretének színét határozza meg.

A következő biteknek van jelentésük:

1. bit = kék, 3. bit = vörös, 5. bit = zöld, 7. bit = fényes szín.

A megfelelő bitek beállításával kialakítható a lehetséges 16 szín.

bitek:	7.	5.	3.	1.	színek	bitek:	7.	5.	3.	1.	színek
	0	0	0	0	fekete		1	0	0	0	fekete
	0	0	0	1	sötétkék		1	0	0	1	kék
	0	0	1	0	sötétvörös		1	0	1	0	vörös
	0	0	1	1	sötétlila		1	0	1	1	lila
	0	1	0	0	sötétzöld		1	1	0	0	zöld
	0	1	0	1	sötétcián		1	1	0	1	cián
	0	1	1	0	sötétsárga		1	1	1	0	sárga
	0	1	1	1	szürke		1	1	1	1	fehér

Beállításával a hatása azonnal érvényre jut, mert a keret színét minden megszakítás lekezelésekor a rendszer újraírja.

COLOUR

A képernyő üzemmódot jelzi. Jelentése:

- 0 - kétszínű mód
- 1 — négy színű mód
- 2 - tizenhatszínű mód

Figyelem! Másik funkció végzi a beállítását (I=4) és másik a lekérdezést (1=0).

DELAY_KEY

Konzol input időzítés. Az automatikus billentyű ismétléshez szükséges várakozási idő 20 ms-os egységben. Alapértéke 30.

RATE KEY

Konzol input időzítés. Az automatikus billentyű ismétlés ideje 20 ms-os egységben. Alapértéke 3.

EXTFONT

A 160-255 kódú karakterek képeinek kezdőcíme. Kezdetben a fix karakter-generátor területére mutat. A felhasználó által tervezett karakterképeket a következő címtartományba kell elhelyezni:

4000h - 7FFFh vagy 00000h - 0D7FFh.

Figyelem: A konzol output is ezeket a karaktereket használja. A beállításnál semmiféle ellenőrzést nem végez az UPM! Megjegyezzük, hogy a 224—255 kódú karaktereket a billentyűzetről nem lehet generálni.

FIXFONT

A felhasználó saját karakterképpel is használhatja a 32—159 kódú karaktereket. A megtervezett karakterek képét itt is a következő címtartományba kell helyezni:

4000h - 7FFFh **vagy** 00000h - 0D7FFh.

Figyelem! A konzol output is ezeket a karakterképeket használja. A beállításnál semmiféle ellenőrzést nem végez az UPM!

4.3. A bővítés rutinok felsorolása

A bővítés rutinok az I regiszter híváskori értéke szerint felsorolva a következők:

- 0 — képjellemzők beállítása, lekérdezése
- 1 — egy karakter kiírása
- 2 — string kiírás

- 3 - sugárbeállítás adott szövegpozícióba
- 4 — videó mód beállítása
- 5 — képernyő-törlés
- 6 — sugárbeállítás abszolút pozícióra
- 7 — relatív sugárpozíció beállítás
- 8 — sugár bekapcsolás
- 9 — sugár kikapcsolás
- 10 — képernyőfeltöltés
- 11 — bővítés karaktergenerátor kezdőcímeinek beállítása, lekérdezése
- 12 — paletta regiszterek beállítása
- 13 — sugárjellemzők lekérdezése
- 14 — konzol input időzítések beállítása, lekérdezése
- 15 — fix karaktergenerátor kezdőcímeinek beállítása, lekérdezése

4.4. A bővítés rutinok leírása

0. Képjellemzők beállítása, lekérdezése

BE: C = 240

1 = 0

D = 0 lekérdezés

KI: H, L, IXH, **IXL**, **IYH**, **IYL**, A = paraméterek (0-6)

D < > beállítás

BE: D, E, IXH, IXL, IYH, **IYL** = paraméterek (0-5)

KI: —

Működés:

A képernyő rendszerváltozóit lehet lekérdezni, illetve beállítani. A COLOUR változó beállítása külön rutinnal (1=4) történik.

A paraméterek jelentése:

0. = SL_STYLE,	vonaltípus (1—16)
1. = SL_MODE,	pontkiírás típusa 0=felülír, 1=AND, 2=OR, 3=XOR
2. = SÍNK,	tintasín (0-15)
3. = SV_FLAG,	karakter felülírás 0.bit=karakterpontok (tintasín) 1. bit=háttérpontok (papírsín) (1=régi pont marad, 0=új pontot kiírja)
4. = SBORDER,	keretsín 1. bit=kék, 3. bit=vörös, 5. bit=zöld, 7. bit=fényerő
5. = SPAPER,	háttérsín (0-15)
6. = COLOUR,	képernyő színmód 0=kétszínű, 1=négyszínű, 2=tizenhatszínű

1. Egy karakter kiírása

BE: C = 240
1= 1

E = karakter kódja

KI: — (A sugár a következő karakterre áll)

Működés:

Kiírja a képernyőre az adott karaktert. Az aktuális sugárpozíció lesz a karakter bal felső sarka. Ha a karakter a kép jobb szélén már nem férne ki, akkor a sugárpozíció a kiírás előtt lefelé lép 10 pontsornyit és a bal szélre áll. Ha a kép alján nem fér ki a karakter, akkor nem történik kiírás.

A „kocsi vissza” (ASCII 13) és „soremelés” (ASCII 10) karakterek a szokásos módon hatnak a sugárpozícióra. Ha ennek hatására a sugárpozíció a képernyőn kívülre kerülne, akkor nem történik végrehajtás. A képernyő nem

fog a soremelés hatására felfelé lépni (tehát nincs ún. képernyő rollozás), ezt a funkciót csak a konzol output hajtja végre. Minden egyéb vezérlő karakter hatástalan.

A karakterkiírás az aktuális SÍNK, SPAPER és SLMODE változók értéke szerint történik. A karakter képében az 1 értékű bitek tintaszínt, a 0 értékűek papírszínt jelölnek. Ha a SVFLAG változóban a megfelelő bit 1 értékű, akkor az ahhoz tartozó pontok nem kerülnek kiírásra.

Miután a karakter kiírása megtörtént, a sugárpozíció továbblép úgy, hogy a következő karakterkiírás jobbról kövesse a most kiírt karaktert. A sor végéről a következő sor első pozíciójára kerül a sugár.

2. String kiírás

BE: C = 240
1 = 2

DE = string kezdőcíme
IX = karakterek száma

KI: -

Működés:

A string kezdőcímétől kezdve egyenként veszi a karaktereket és kiküldi a képernyőre a karakter kiíró rutin meghívásával. A stringet a következő címtartományba kell elhelyezni:

4000h - 7FFFh vagy 00000h - 0D7FFh.

A rutin nem ellenőrzi, hogy a string ebben a tartományban található.

3. Sugárbeállítás adott szöveg pozícióra

BE: C = 240
I = 3

D = oszlop száma
E = sorszáma

KI: A = 249 hibás paraméterek
A = 0 sugárpozíció beállítása

Működés:

A sugárpozíciót beállítja az adott karakterpozíció bal felső sarkába, ezzel előkészíti a szövegkiírást. Ez a rutin soha nem húz vonalat a képernyőn. A képernyő bal felső sarkának (1,1) a szövegpozíciója. Ha D vagy E értéke zérus, akkor a megfelelő pozíció marad a beállítás előtti értéken.

4. Videó mód beállítás

BE: C - 240
I = 4

E = videó mód (COLOUR változó kívánt értéke)
0=kétszínű, 1=négy színű, 2=tizenhatszínű

KI: A = 247, ha hibás a paraméter
A = hibás beállítás, a változók alaphelyzetbe állnak.

Működés:

Beállítja a megjelenítési módot, ezzel együtt inicializálja a konzol outputot és a videó rutinokat, valamint törli a képernyőt. A beállított színek a színikapcsoló állásától függenek. Színes állásban két- és négy színű módban a palettaszínek: 0=fekete, 1=zöld, 2=vörös, 3=kék, tizenhatszínű módban a háttér fekete, a tinta zöld.

Fekete-fehér állásban két- és négy színű módban a palettaszínek: 0=fekete, 1=fehér, 2=zöld, 3=vörös, tizenhatszínű módban a háttér fekete, a tinta fehér.

5. Képernyőtörlés

BE: C = 240
I = 5

KI: — (karakteres képernyőt is törli)

Működés:

A képernyőt feltölti az aktuális háttérszínnel, a sugárpozíciót a képernyő bal alsó sarkába, azaz a (0,0) grafikus pozícióra teszi, a sugarat kikapcsolja, a konzol output ASCII képernyőjét törli, a konzol kiíratási pozícióját az (1,1) szövegpozícióba teszi.

6. Sugárbeállítás abszolút pozícióra

BE: C = 240

DE = Y koordináta (0-959)
IX = X koordináta (0-1023)

KI: A = 249 , hibás paraméterek
A = 0 beállítás

Működés:

A sugárpozíciót az (X,Y) abszolút koordinátákra helyezi. Minden megjelenítési módnál az adatok X=0-1023 és YO-959 tartományba eshetnek. A (0,0) pont a képernyő bal alsó sarka.

Ha a sugár ki van kapcsolva, akkor csak pozicionálás történik, ha be van kapcsolva, akkor vonalat húz az előző pozíciótól az új pozícióig. Vonalat tehát úgy lehet rajzolni, hogy a sugarat kikapcsoljuk, a kezdőpozícióba visszük, bekapcsoljuk és a végpozícióba visszük.

A vonalrajzolás a SÍNK, SPAPER, SL_MODE és SL_STYLE változók értékei szerint történik.

7. Relatív sugárpozíció beállítás

BE: C = 240
1 = 7

DE • Y relatív koordináta
IX • X relatív koordináta

KI: A = 249 hibás paraméterek
fi = 0 beállítás

Működés:

Hasonló, mint az abszolút pozicionálás, de itt a megadott koordinátákat az aktuális értékekhez hozzáadja, és az eredmény lesz az új pozíció.

8. Sugár bekapcsolás

BE: C 240
I 8

KI: -

Működés:

A sugarat bekapcsolja és kitesz egy pontot az aktuális sugárpozícióba a SÍNK és SL_MODE értéke szerint. Ettől kezdve az abszolút és relatív sugárpozíció állítások vonalrajzolást végeznek.

9. Sugár kikapcsolás

BE: C 240
I 9

KI: -

Működés:

A sugarat kikapcsolja, ettől kezdve az abszolút és relatív sugárpozíció állítások nem húznak vonalat, csupán pozicionálást végeznek.

10. Képernyőfeltöltés (FILL)

BE: C = 240
I = 10

KI: -

Működés:

A képernyőn egy zárt területet beszínez. Az aktuális sugárpozícióban kezdődik a beszínezés a SÍNK változó értéke szerint. Addig végzi a színezést, amíg a vízszintesen és függőlegesen szomszédos pontok az aktuális sugárpozícióval azonos színűek. A rutin teljesen rekurzív, lehetővé teszi tetszőleges területek beszínezését.

Működése verem-orientált, ehhez a felhasználói vermet (STACK) használja. Ennek megengedett címtartománya:

4000h - 7FFFh vagy 00000h - 0D7FFh.

Beépített védelem gondoskodik arról, hogy a verem ne mélyüljön az engedélyezett címtartományok alá. Ilyenkor a működés félbeszakad. Kevesebb verem kell akkor, ha a terület feltöltése a képernyőn felülről lefelé történik.

17. Bővítés karaktergenerátor kezdőcímének beállítása, lekérdezése

BE: C = 240
I = 11

DE = 0 lekérdezés
KI: HL = aktuális kezdőcím (EXTFONT)

DE < > 0 beállítás

BE: DE = új kezdőcím (EXTFONT)

Működés.

A 160-255 kódú karakterek képének kezdőcímét adja vissza, illetve állítja be. Egy karakter 8X10 pontból áll, ezt a memóriában 10 egymást követő bájt reprezentálja. Az 1-es értékű bitek jelentik a tintaszínű, a 0. értékű bitek pedig a háttérszínű pontokat. A memóriában az alábbi címtartományba kell helyezni a bővítés karaktergenerátort:

4000h - 7FFFh vagy 00000h - 0D7FFh.

Ennek helyes beállítását a rutin nem ellenőrzi!

A konzol output is ezeket a karakterképeket fogja a beállítás után használni.

Megjegyezzük, hogy a 224-255 kódú karaktereket a billentyűzetről nem lehet generálni.

12. Paletta regiszterek beállítása

BE: C = 240
I = 12

D = 0, regiszter színe
E = 1. regiszter színe
IXH = 2. regiszter színe
IXL = 3. regiszter színe

KI: -

Működés:

Beállítja a palettaszíneket a két- és négy színű üzemmódhoz.

A bitek jelentése: 0.bit=kék, 2.bit=vörös, 4.bit=zöld, 6.bit=fényes szín

A megfelelő bitek beállításával kialakítható 16 szín:

bitek	6.	4.	2.	0.	színek	bitek	6.	4.	2.	0.	színek
	0	0	0	0	fekete	1	0	0	0	fekete	
	0	0	0	1	sötétkék	1	0	0	1	kék	
	0	0	1	0	sötétvörös	1	0	1	0	vörös	
	0	0	1	1	sötétlila	1	0	1	1	lila	
	0	1	0	0	sötétzöld	1	1	0	0	zöld	
	0	1	0	1	sötétcián	1	1	0	1	cián	
	0	1	1	0	sötétsárga	1	1	1	0	sárga	
	0	1	1	1	szürke	1	1	1	1	fehér	

13. Sugárjellemzők lekérdezése

BE: C 240
 I 13

KI: HL = aktuális Y koordináta
 IX = aktuális X koordináta
 A • sugárállapot

Működés:

Megadja az aktuális sugárjellemzőket.

A sugárállapot jelentése: 0 • kikapcsolt sugár
 255 = bekapcsolt sugár

14. Konzol input időzítések beállítása, lekérdezése

BE: C 240
 I 14

0 lekérdezés

KI: H = RATE KEY
 L = DELAY KEY

E O O beállítás

BE: D = RATE_KEY
E = DELAY_KEY

KI: -

Működés:

Beállítja, vagy lekérdezi a konzol input időzítéseket. A paraméterek 20 ms-os egységben értendők.

DELAY_KEY: várakozási idő az első karakter ismétléséig

RATE_KEY: két karakter-ismétlés közötti idő

15. Fix karaktergenerátor kezdőcímének beállítása, lekérdezése

BE: C = 240
I 15

DE = 0 lekérdezés

KI: HL = kezdőcím (FIXFONT)

DE <> 0 beállítás

BE: DE = új kezdőcím (FIXFONT)

KI' —

Működés:

A 32—159 kódú karakterek képeinek kezdőcímét adja vissza, vagy állítja be. Működése minden más szempontból azonos a bővítés karaktergenerátorra vonatkozó rutinnal (1=11).

A beépített fix karaktergenerátor címe OBOOH.

Ezt visszaírva újra az eredeti karakterképeket használja a konzol output és a karakterkiíró rutin.

VIDEOTON

ELEKTRONIKAI VÁLLALAT
SZÁMÍTÁSTECHNIKAI GYÁRA