SPECCY-DOS

REFERENCIA ADATOK
GÉPI KÓDÚ
FELHASZNÁLÓI
PROGRAMOKHOZ

kézirat

BUDAPEST
1985 December

©: dr. HORVÁTH FERENC , 1985,1986.

contains 76 sheets/A4 SØ1

TARTALOM

	BEVEZETÉS	/iii/
	SZUBRUTIN INDEX	/ii/
I.	MEMÓRIAFELOSZTÁS	I-1.
II.	A SPECCY-DOS I/O RENDSZERE	II-1.
	1. HARDVARE VEZÉRLŐJELEK KEZELÉSE	II-1.
	2. AZ FDC REGISZTEREINEK KEZELÉSE	II-5.
III.	A SPECCY DISZK FELÉPITÉSE	III-1.
	1. A SPECCY DISZK FIZIKAI FELÉPITÉSE	III-1.
	2. A SPECCY DISZK LOGIKAI FELEPITÉSE	III-2.
	2.1. A DIRECTORY TERÜLET FELÉPITÉSE	III-3.
	2.1.1. TO/S1 FELÉPITÉSE	III-3.
	2.1.2. TC/S2...S7 FELÉPITÉSE	III-7.
	2.2. A TÉRKÉP TERÜLET FELÉPITÉSE	III-10.
IV.	A SPECCY-DOS SZUBRUTINOK ÁTTEKINTÉSE	IV-1.
V.	A SPECCY-DOS SZUBRUTINJAINAK HIVÁSA MÁS PROGRAMOK ÁLTAL	V-1.
	1. ROM VÁLTÁSOK	V-1.
	2. A VÁLTOZÓ CIMÜ SPECCY SZUBRUTINOK HIVÁSA	V-2.
	2.1. ÁLTALÁNOS HIVÁS	V-5.
	2.2. EGYEDI HIVÁS	V-6.
VI.	A SPECCY-DOS RAMTERÜLETE	VI-1.
VII.	A SPECCY-DOS SZUBRUTINJAINAK ISMERETETÉSE	VII-1.
VIII.	MINTAPROGRAM: SKILL	VIII-1.

HORVÁTH FERENC
1045 BUDAPEST /i/
IV. k. TITO u. 8 VII/39

BEVEZETÉS

Ez a kézikönyv azon SPECCY-DOS felhasználók számára készült, akik saját file kezelő programokat kívánnak készíteni, kihasználva a SPECCY-DOS belső szubrutinkészletét. A kézikönyv az összes olyan információt tartalmazza, amely a fenti feladat megoldásához szükséges lehet. A kézikönyv legcélszerübb használati módja az,hogy először gondosan végigolvassuk az elejétől a végéig, a későbbiekben, a programok irása során pedig szubrutin-katalógusként használjuk. Ezt segiti elő a könyv elején található betürendes szubrutin index.

A kézikönyv irása során feltételeztünk bizonyos előismereteket, igy feltételeztük, hogy az olvasó elsősorban alaposan ismeri a SPECTRUM-ot, másodsorban megfelelően járatos a SPECCY-DOS BASIC-ből való kezelésében, végül feltételeztük, hogy az olvasó járatos a Z-8Ø assembly nyelvü programozásában. A SPECCY-DOS-ra vonatkozó alapismeretek a "SPECCY-DOS FELHASZNÁLÓI KÉZIKÖNYV" c. irodalomból szerezhetők meg.

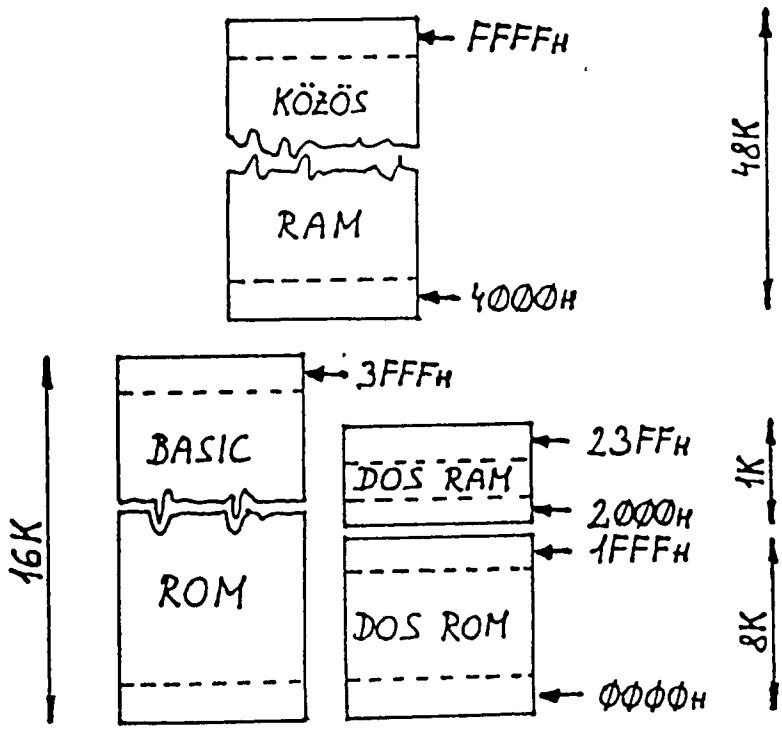
Bizonyos szubrutinok használatához /pld.:TYPONE/,illetve bizonyos fajta feladatok megoldásához a fentieken kivül a SPECCY-DOS-ban alkalmazott floppy diszk kontroller /továbbiakban: FDC/, azaz a WD 177Ø részletes ismerete is szükséges.

A kézikönyv fő részét képező VII. fejezet a szubrutinok interface-ivel foglalkozik, mig az egyéb fejezetek a memória felépitéséét, a DOS rendszerváltozóit, a szubrutinok hivási módját és a diszk felépitését ismertetik. A SPECCY szubrutinok használatához sok sikert kivánunk!

I. MEMÓRIAFELOSZTÁS

A SPECCY-DOS 8 Kbyte ROM és 1 Kbyte RAM memóriát tartalmaz. Ezeket árnyék ROM-nak, ill. árnyék RAM-nak nevezzük, mivel a SPECTRUM-ban lévő memóriával azonos cimeken helyezkednek el. Az I-1 ábra a SPECCY-vel felszerelt 48 k-s SPECTRUM cim szerinti memóriakiosztását mutatja. A BASIC, ill. a DOS ROM közül egyidejüleg csak az egyik lehet kiválasztva. A DOS RAM csak akkor férhető hozzá, ha nem a BASIC ROM van kiválasztva, mig a közös RAM mindkét ROM-ban futó programból hozzáférhető.

A ROM-ok közötti választás módjáról a későbbiekben lesz szó.



I-1 ábra

II. A SPECCY-DOS I/O RENDSZERE

A DOS hardware és software közötti kapcsolat megvalósitására a SPECCY-DOS-ban un. memóriaként kezelt I/O rendszer müködik. Ez azt jelenti, hogy a szükséges input/output müveleteket a processzor nem IN/OUT, henem memória olvasó, ill. iró müveleteken keresztül valósitja meg. A szükséges I/O müveleteket két csoportba sorolhatjuk, miszerint /i/ a DOS hardware vezérlőjeleit kezelő output müveletek, és /ii/ az FDC különböző regisztereinek olvasására, ill. irására szolgáló I/O müveletek.

II-1. HARDWARE VEZÉRLŐJELEK KEZELÉSE

Ebbe a csoportba kizárólag output müveletek tartoznak, amelyekkel a DOS hardware különböző vezérlőjeleit kezelhetjük.

Ha a 3B41 H /15169 D/ címre memória

müveletet hajtunk végre, akkor a processzor adatbusza különböző harware elemekkel kerül kapcsolatba, ezáltal különböző hardware folyamatokat vezérel. Az egyes adatbitek és az emlitett hardware jelek egymáshoz rendelését mutatja a II-1. ábra.

	(MSB) B7	B6	B5	B4	B3	B2	B1	B0 (LSB)
3B41H	DENSITY SELECT DENSEL	DOS ENABLE DOSEN	NEM HASZNÁLT BIT	NEM HASZNÁLT BIT	DOS/BASIC SELECT D/B	SIDE SELECT SS	DRIVE SELECT DS1	DRIVE SELECT DS0

II-1. ábra

1./ Drive kiválasztás: BØ, B1

Diszkműveleteket megelőzően ezen bitek segítségével kell kijelölnünk, hogy a művelet melyik drive-on hajtódjék végre.
A kijelölést követően minden művelet a kijelölt drive-on fog végrehajtódni, egészen addig, amig más drive-ot nem jelölünk ki.

A II-1. táblázat a BØ, B1 bitek kódolását mutatja.

B1 /DS1/	B0 /DS0/	A kiválasztott drive
0	0	drive-0
0	1	drive-1
1	0	drive-2
1	1	drive-3

II-1. táblázat

2./ Lemezoldal kiválasztás: B2

Kétoldalas drive esetén B2 = 0-val, a lemez Ø jelü, míg B2 = 1-el, a lemez 1-jelü oldalát választhatjuk ki. Ez a kiválasztás a következő lemezoldal kiválasztásig érvényben marad.

3./ DOS/BASIC választás: B3

Ha DOS engedélyezve van, és B3=1, akkor a BASIC ROM letiltódik, és soronkövetkező memóriaműveletek esetén a DOS ROM ill. a DOS RAM leszkiválasztva a konkrét cimtől függően. Ha a DOS engedélyezve van /lsdB6/ és B3=Ø, akkor a DOS ROM, ill. DOS RAM letiltódik, és a BASIC ROM lesz az érvényes memória a saját cimtartományában.

4./ DOS engedélyezés /tiltás: B6

Ha B6=Ø, akkor a DOS engedélyezve van, azaz B3 bittel választhatunk a DOS, ill. a BASIC ROM között, és a gép elfogadja és végrehajtja a BASIC programokban elhelyezett DOS /*-os/ parancsokat. Ha B6=1, akkor a DOS le van tiltva, más szóval ki van kapcsolva. Ez azt jelenti, hogy annak ellenére, hogy a SPECCY rá van csatlakoztatva a SPECTRUM-ra, nem fogad el és nem hajt végre DOS parancsokat, és a B3 bittel nem lehet kiválasztani a DOS ROM-ot.

Ez a képesség lehetőséget ad olyan SPECTRUM perifériákkal való kompatibilitás megvalósitására, amelyekkel a SPECCY-DOS egyébként nem volna kompatibilis. E bit ciklikus váltogatásával pld. olyan másolóprogramot készithetünk, amelyik az INTERFACE-1-hez kapcsolt microdrive és SPECCY-hez kapcsolt drive-ok közötti kétirányu file másolásra képes.

5./ Irássürüség kiválasztás: B7

Ezzel a bittel határozhatjuk meg, hogy az FDC milyen irássürüségre számitva végezze el a diszk olvasását, illetve milyen irássürüséggel hajtsa végre a diszk irását.

B7 = Ø: Dupla irássürüség

B7 = 1: Szimpla irássürüség

A kiválasztott irássürüség mindaddig érvényben marad, amig ujabb irássürüség kiválasztást nem végzünk.

Egyéb tudnivalók a 3B4lH címmel kapcsolatban

Mint már hangsulyoztuk, a 3B4lH cimre csak memória irási müveletnek van értelme. Az erről a cimről beolvasott adat a DOS szempontjából nem értelmezhető, és semmiféle kapcsolatban nincs az e cimre kiirt adattal. Mivel ugyanazzal a memória irási müvelettel 5 különböző funkciót vezérlünk, a software -nek mindig tudnia kell az e cimre legutoljára kiadott kódszót. Ez sokféleképpen megoldható, legcélszerübb azonban a felhasználói programokban is azt a módszert követni, amelyet a DOS irásakor alkalmaztunk. Ennek lényege, hogy a 3 B4lH cimre legutoljára kiadott kódszót egy rendszerváltozóban tároljuk és valahányszor uj kódszót irunk ki a 3B4lH-re, mindig beirjuk az uj kódszót e rendszerváltozóba.

E rendszerváltozó neve a DOS-ban DFLAG-l, cime 2000H.

II-2. Az FDC REGISZTEREINEK KEZELÉSE

Az FDC regiszterei software-ből akkor hozzáférhetőek, ha a DOS van kiválasztva, azaz a 3B4lH cimre legutoljára kiadott kódszóban B3=1. Ez esetben az FDC regiszterei memória olvasással, ill. irással férhetők hozzá. A II-2. táblázat az FDC egyes regisztereihez rendelt memóriacimeket tartalmazza.

MEMÓRIA CÍM	OLVASÁS	ÍRÁS
3290H	STÁTUSZ REGISZTER	PARANCS REGISZTER
3291H	TRACK REGISZTER	TRACK REGISZTER
3292H	SZEKTOR REGISZTER	SZEKTOR REGISZTER
3293H	ADAT REGISZTER	ADAT REGISZTER

II-2. Táblázat

Megjegyezzük, hogy az FDC regisztereinek kezelésére a gépi kódu programot készitő felhasználónak nem lesz szüksége feltéve, hogy az e kézikönyvben közölt szubrutinok és belépési pontok segitségével használja a DOS szolgáltatásait. Az itt közölt információkra a felhasználónak csak akkor van szüksége, ha a DOS fizikai szintü rutinjainak megkerülésével ő maga kivánja vezérelni a DOS hardware-t. Ehhez azonban az FDC alapos ismerete is szükséges. Az e kézikönyvben közölt adatok birtokában elvileg nem kizárt, hogy valaki saját diszk operációs rendszert készitsen a SPECCY hardware kihasználásával, de a SPECCY software megkerülésével.

III; A SPECCY DISZK FELÉPITÉSE

A diszk szervezése,felépitése alapvetően meghatározza egy adott DOS minőségi jellemzőit. A SPECCY-DOS diszkjeinek szervezése olyan, hogy lehetőséget nyujt a legkülönbözőbb feladatokhoz való alkalmazásra, mint például soros, ill. véletlen hozzáférésü file-ok kezelése, adatbázis kezelők készitése, stb.

III-1. A SPECCY DISZK FIZIKAI FELÉPITÉSE

A SPECCY diszk egy vagy kétoldalas lehet,az egy oldalon lévő track-ek száma 35,40,45, ill. 80 lehet. Az egy track-en lévő szektorok száma 35,40, ill. 80 track esetén 10/16, 45 track esetén 7/13. /A kisebb szám szimpla, a nagyobb dupla irássürüség esetén érvényes./ 1 szektor hossza 256 byte.
A SPECCY -DOS által kezelt konfigurációkat a III-1 táblázat mutatja.

OLDALAK SZÁMA	TRACK-EK SZÁMA			
	35	40	45	80
1 Egyoldalas diszk	✓	✓	✓	✓
2 Kétoldalas diszk	✗	✓	✗	✓

III-1 Táblázat

Minden kezelt konfiguráció esetén lehetőség van egyszeres, ill. kétszeres irássürüségü rögzitésre.
A track-ek számozása Ø-tól, a szektorok számozása 1-től kezdődik. A Ø-dik track /TOO/ a diszk legnagyobb sugaru track-je, az első szektor az index lyukat követi /SØ1/.
A szektorok számozásánál alkalmazott un.interleave tényező értéke 2, azaz két logikailag egymást követő szektor között két másik szektor található.

III-2. A SPECCY DISZK LOGIKAI FELÉPITÉSE

Logikailag a SPECCY diszk három részre osztható, um. /i/ directory terület,/ii/ térkép terület és /iii/ adatterület.
A directory terület egyrészt a diszk egészére vonatkozó információkat, másrészt a file-okra, mint egészekre vonatkozó információkat tartalmaz. A directory terület minden esetben a diszk első oldalán /side Ø/, a TØ első 7 szektorát /S1...S7/ foglalja le.

A térkép terület az adatterület "térképét" tartalmazza. Ennek a térképnek a segitségével válik lehetségessé, hogy az egyes file-ok szektorai az adatterületen szükség esetén teljesen összevissza, más file-okhoz tartozó szektorokkal összekeveredve helyezkedhessenek el, biztositva ezzel az adatterület tökéletes kihasználását. Ugyancsak a térkép ad lehetőséget a file-on belüli véletlen szektorhozzáférésre, azaz arra, hogy egy file N-edik szektorát az azt logikailag megelőző N-1 szektor végigolvasása nélkül beolvashassuk vagy irhassuk. Ugyancsak a térkép ad lehetőséget file-on belüli előre, ill. hátrafelé való keresésre. A térkép mindig a diszk első oldalán /side Ø/ van.
Az adatterület foglalja le a diszk legnagyobb részét. Az adatterület a file-ok adattartalmát foglalja magában. Az adatterület tartalmazhat hibás, nem használt szektorokat is /max 128/, mig a directory és a térkép terület csak hibátlan szektorokból állhat.

A III-2 táblázat a térkép terület track és szektorfoglalását mutatja. /SS = egyoldalas, DS = kétoldalas diszk, SD= szimpla, DD= dupla sürüség/. Az adatterület a térkép utáni tracken kezdődik.

III-2. táblázat

KONFIGURACIO	TERKEPTERULET SD	TERKEPTERULET DD
35T-SS	T1:S1.....S7	T1:S1....S11
40T-SS	T1:S1.....S8	T1:S1....S12
45T-SS	T1:S1.....S6	T1:S1....S11
80T-SS	T1:S1-S9,T2:S1-S7	T1:S1-S15,T2:S1-S10
40T-DS	T1:S1-S9,T2:S1-S7	T1:S1-S15,T2:S1-S10
80T-DS	T1,2,3:S1-S9, T4:S1-S4	T1,2,3:S1-S15, T4:S1-S4

III-2 táblázat

III-2.1. A DIRECTORY TERÜLET FELÉPITÉSE

A directory terület tartalom szerint két részre osztható. TØS1 a diszkre, mint egészre vonatkozó információkat, valamint az un. directory térképet tartalmazza, mig TØ/S2 .... S7 azegyes file-okra vonatkozó információkat tartalmazza.

1. TØ/S1 FELÉPITÉSE

1.1 RENDSZERAZONOSITÓ BYTE-OK /III-1 ábra./

ØØH	Ø1H	Ø2H	Ø3H	Ø4H
54H	52H	4BH	59H	32H

III-1 ábra

s13

1.2. A DISZK NEVE /III-2 ábra/

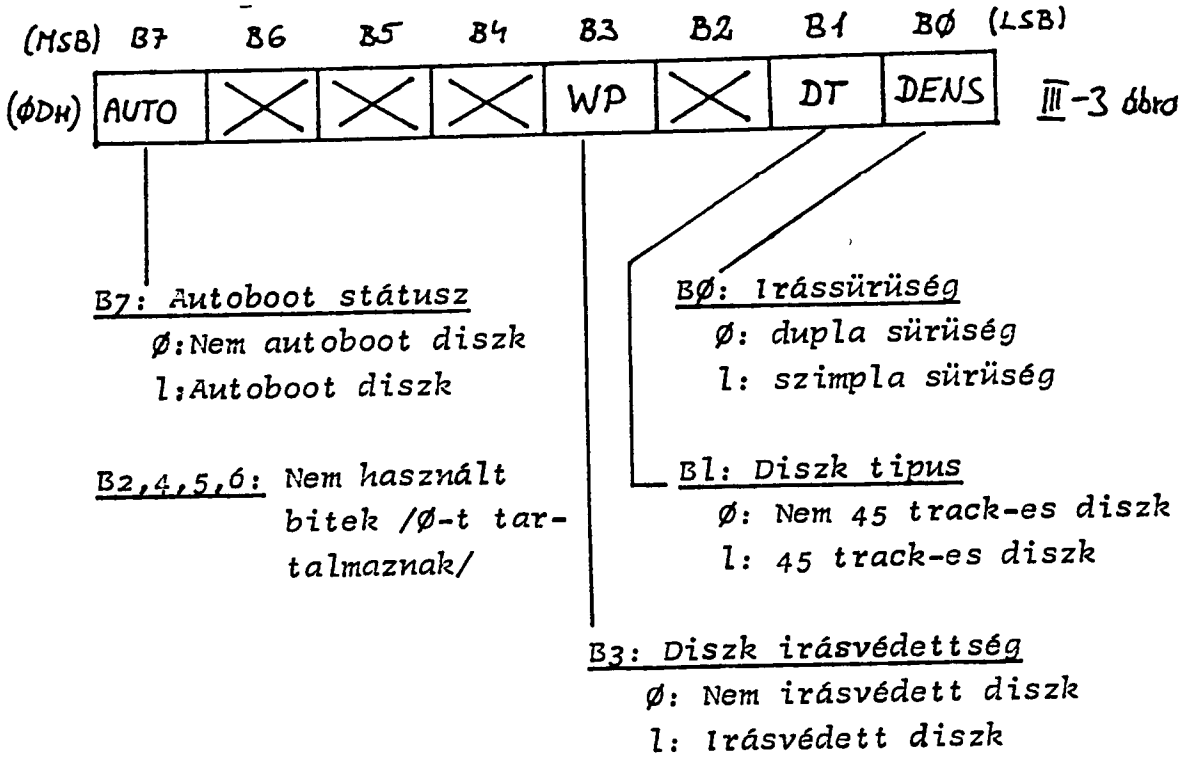
Ø5H	Ø6H	Ø7H	Ø8H	Ø9H	ØAH	ØBH	ØCH
D	I	S	K	N	A	M	E

III-2 ábra

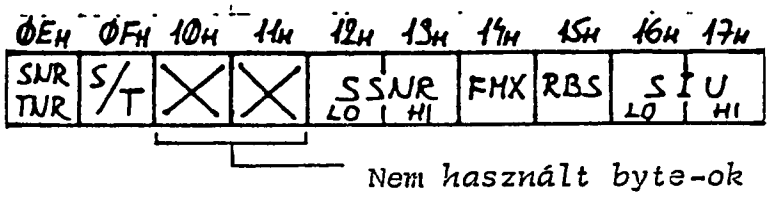
Ez a 8 byte a diszk nevét tartalmazza, szükség szerint space-ekkel 8 karakterre kiegészitve.

1.3. A DISZK LEIRÓ BYTE-JA /ODH/ /III-3 ábra/

A leiróbyte bitenként van értelmezve, amint a III-3. ábra mutatja.



1.4. A DISZKRE VONATKOZÓ EGYÉB INFORMÁCIÓK /III-4.ábra/



III-4.ábra

ØEH: BØ...B6: A diszk egy oldalán levő track-ek száma
B7: Ø, ha egyoldalas, 1, ha kétoldalas diszk.

ØFH: Az egy track-en levő szektorok száma /7/10/13/16/

12H,13H: A diszk összes szektorainak száma, nem levonva a hibás szektorokat, de levonva a directory, valamint a térkép terület szektorait és a nem használt ép szektorokat. (Azaz: az adatterület szektorainak száma.)

14H: A file-ok maximális száma.

15H: Az inicializáláskor felismert sérült szektorok száma.

16H,17H: A létező file-ok által lefoglalt szektorok száma az adatterületen.

1.5. AZ AUTOBOOT FILE-RA VONATKOZÓ INFORMÁCIÓK

A soronkövetkező 25 byte /18H....3ØH/ a létező autoboot file 25 byte hosszu file-directory-jának másolata. Ennek belső felépitését lásd később.

1.6. 27 BYTE HOSSZU KIHASZNÁLATLAN TERÜLET /31H...4BH/

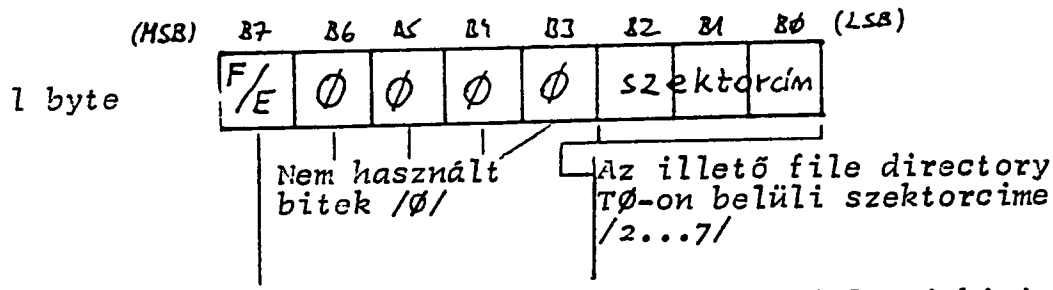
1.7. KÓDOLT FILENEVEK TERÜLETE /4CH...87H/

Ezen a 6Ø byte hosszu területen minden lehetséges file-hez tartozik egy byte, mégpedig az első byte a file-directory-ban első file-hoz tartozik, a második byte a második file-hoz, stb.

Ha az adott file létező file, akkor a neki megfelelő byte-ra a file nevéből képzett 1 byte-os kód kerül. A kódot, ugy képezzük, hogy a 8 karakterre kiegészitett file név karakterkódjainak MOD 256-os összegét vesszük. Ezek a kódok a file-név szerinti keresést gyorsitják meg.

1.8. A DIRECTORY TÉRKÉP

A TØ/S1 utolsó 120 byte-ját a directory térkép foglalja le. Ez a térkép nem más, mint egy táblázat, amely az egyes 25 byte hosszu file directory-k lokalizálását szolgálja a TØ track S2....S7 szektorain belül. A 120 byte-os terület 60 db 2 byte széles, un. directory pointer-t foglal magába. Az első pointer /88H,89H/ a file directory-n belül az első file-helyhez tartozik, a második pointer /8AH,8BH/ a másodikhoz, stb.Az emlitett 25 byte-os file-helyeket, amelyek a TØ track S2....S7 szektorait foglalják le file directory-knak nevezzük, és részletes felépitésükről a következő pontban lesz szó. A 2 byte-os directory pointer a III-5. ábrán látható, bitenkénti bontásban.



Az illető file directory foglaltsági bitje.

B7=1, ha a file directory foglalt, azaz létező file-t tartalmaz,

B7=Ø, egyébként

2.byte: Az illető file directory legelső byte-jának relativ cime az adott szektoron belül. Ez a byte a következő értékeket veheti fel: 00H, 19H, 32H, 4BH, 64H, 7DH, 96H, AFH, C8H, ill. E1H

III-5. ábra

2.TØS2....S7 FELÉPITÉSE

Ezek a szektorok tartalmazzák az un. file directory-kat. Minden lehetséges file-hoz tartozik egy-egy 25 byte hosszu terület, a file directory-ja, amely ha a file létezik, az adott file-re vonatkozó adatokat tartalmazza. Egy-egy szektor lO file directory-t tartalmaz, a szektor legutoltsó 6 byte-ja kihasználatlan. A file directory felépitése a következő./Az egyes byte-okat a file directory legelsö byte-jától kezdve ØØH-tól l8H-ig számozzuk/.

ØØH: - FFH, ha a file directory nem tartalmaz létező file-t,

- A file tipuskódja, ha a file directory létező file-t tartalmaz.

A file tipuskódja: Ø: BASIC program a változóival
l: Számtömb
2: Karaktertömb
3: CODE vagy SCREEN$ file

Ø1H....Ø8H: A file neve szükség szerint space-ekkel /2ØH/ 8 karakterre kiegészitve.

Ø9H,ØAH: Kihasználatlan /Tartalma 2ØH, 2ØH, vagy ØØH ,ØØH/

ØBH,ØCH: A file hossza, azaz byte-jainak száma /ØBH-LO, ØCH-HI/

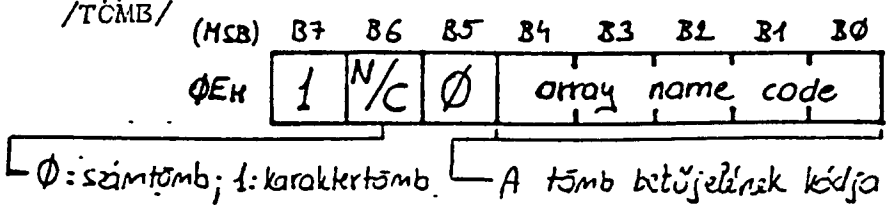
ØDH,ØEH,ØFH,1ØH: Ezen byte-ok tartalma a file tipusától függ /A nem specifikált értékek definiálatlanok./

Tipuskód=Ø: ØEH: 8ØH, ha a file LINE megadása nélkül lett SAVE-elve.
/BASIC/ ØDH,ØEH: A LINE után megadott sorszám, egyébként. /ØDH-LO/

ØFH,lØH: A program hossza a változók nélkül.
/ØFH-LO/
Megjegyzés: ØFH,lØH = /VARS/ - /PROG/
ØBH,ØCH = /E-LINE/- /PROG/ +1

Tipuskód = 1,vagy 2: ØEH tartalmát a III-6 ábra mutatja.

/TÖMB/



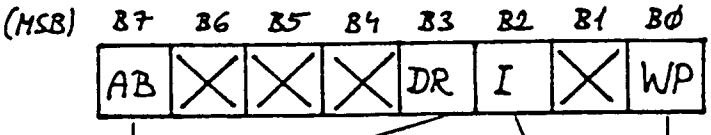
III-6.ábra

(*) Megjegyzés: A tömb betüjelének kódja nem más, mint a tömb betüjelének kisbetüs megfelelőjének ASCII kódja minusz 6ØH.

Tipuskód = 3 ØDH, ØEH: Az a memóriakezdőcim ahonnan a file SAVE-elése történt. /ØDH-LO/
/CODE/

ØFH,1ØH: ØØ,ØØ, ha a file rögzitésekor nem specifikáltak futtatási kezdőcimet. Ha igen, akkor a kezdőcim.

llH: A file attributumbyte-ja, amelynek felépitését a III-7. ábra mutatja.



III-7.ábra

Rögzitési mód
1: Kétszeresen rögzitett
Ø: Egyszeresen rögzitett file

Irásvédettség
1: Irásvédett file
Ø: Nem irásvédett file

Autoboot státusz
1: Autoboot file
Ø: Nem autoboot file

Láthatóság
1: Nem látható file
Ø: Látható file

X: Nem használt bitek, tartalmuk Ø /X=Ø/

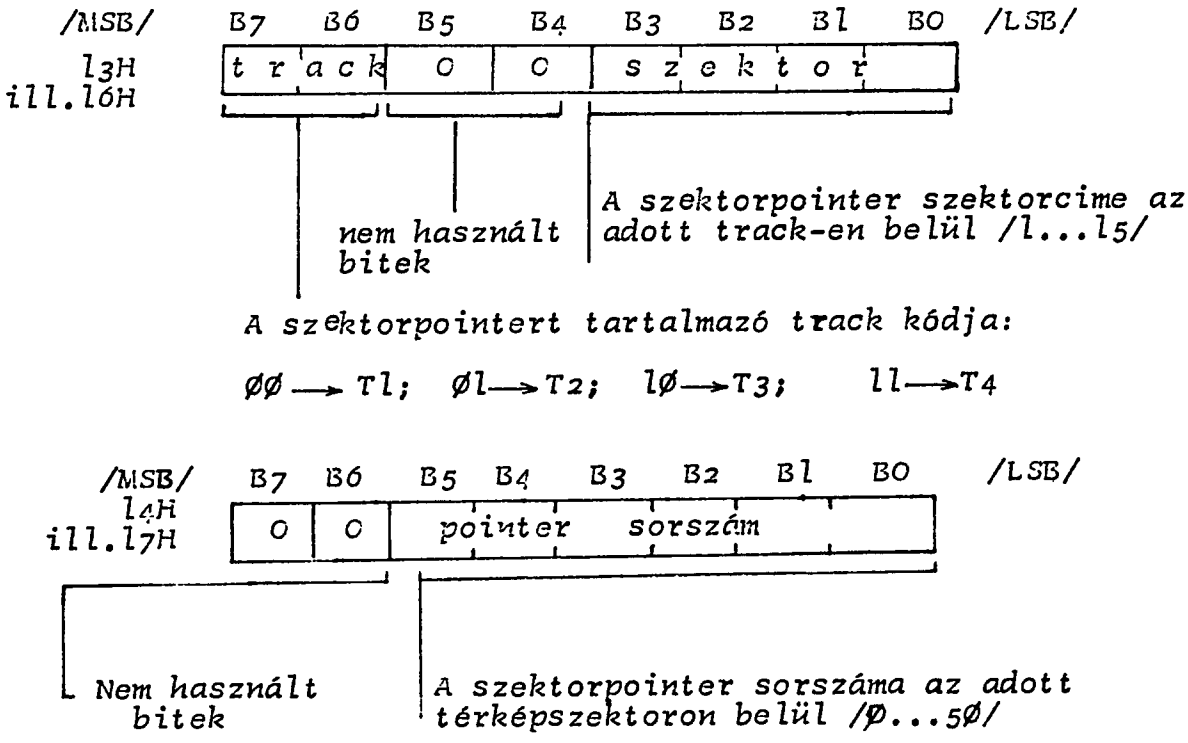
12H: A file egyszeri rögzitéséhez szükséges szektorok száma minusz 1.

13H,14H: A file /ill. kétszeresen rögzitett file esetén a fő file/ első szektorpointerének cime a térkép területén belül. E két byte telepitését a III-8. ábra mutatja.

15H: A file utolsó byte-jának sorszáma /Ø..255/ a file utolsó szektorában.

16H,17H:
Kétszeresen rögzitett file esetén a mellékfile első szektorpointerének cime a térkép területén belül. Lsd: III-8. ábra.

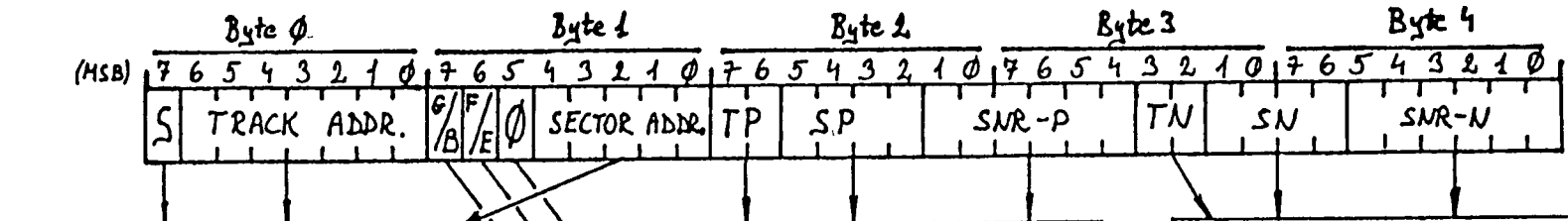
18H: Az adott file-directory sorszáma. /Ø... 59D/



III-8. ábra

III-2.2. A TÉRKÉP FELÉPITÉSE

A térkép terület diszken belüli elhelyezkedését a különböző konfigurációk esetén a III-2 táblázat mutatja. A térkép terület az adatterület struktúráját irja le. Az egyes file-ok szektorainak a diszken való elhelyezkedése a térkép alapján követhető nyomon. A térkép területen az adat-terület minden egyes szektorához tartozik egy 4Ø bit széles, un. szektorpointer. Ha a diszken egy több szektorból álló file-t hozunk létre, akkor ezekből a szektorpointerekből egy un. lánc keletkezik, amely annyi szektorpointert tartalmaz, ahány szektorból áll a file. Kétszeresen rögzitett file esetén mind a fő, mind a mellékfilehoz külön lánc tartozik. A lánc /illetve láncok/ első elemének a térképterületen belüli cimét a file directory 13H,14H, /ill. 16H,17H/ byte-jai tartalmazzák. A szektorpointer 5 féle információt tartalmaz: /i/,jó, vagy hibás-e a szektor, /ii/ tartozik-e file-hoz, vagy nem, /iii/ a szektor adatterületen belüli cime, és /iv,v/ a megelőző és a következő szektorpointer cime. A szektorpointer felépitését a III-9. ábra mutatja.



A szektorpointerhez tartozó szektor címe az adatterületen:

S=OLDAL: Ø: side Ø
1: side 1

TRACK ADDR = TRACK CÍM (Ø..... 79)

SECTOR ADDR = SZEKTORCÍM (1.... 16)

G/B: Ø: hibátlan szektor
1: hibás (nem kasenált) sektor

F/E: Ø: szabad szektor
1: foglalt, azaz valamelyik létező file-hoz tartozó szektor.

Byte-1 / Bit-5: Kihasználatlan bit. (=Ø)

Az előző szektorpointer címe a térkép területen a file-hoz tartozó szektorpointer-láncban:

TP: Az előző szektorpointer track-címének kódja

kód	track
ØØ	T1
Ø1	T2
1Ø	T3
11	T4

SP: Az előző szektorpointer szektorcíme. (1... 15)

SNR-P: Az előző szektorpointer sorszáma a szektoron belül, (Ø...5Ø), ill. 111111, ha a szektorpointer a file legelső szektorához tartozik.

A következő szektorpointer címe a térkép területen a file-hoz tartozó szektorpointer-láncban:

TN: A következő szektorpointer track-címének kódja.

kód	track
ØØ	T1
Ø1	T2
1Ø	T3
11	T4

SN: A következő szektorpointer szektorcíme. (1... 15)

SNR-N: A következő szektorpointer sorszáma a szektoron belül, (Ø....5Ø), ill. 111111, ha a szektorpointer a file legutolsó szektorához tartozik.

III-9. ábra: A SZEKTORPOINTER FELÉPÍTÉSE

S21

IV. A SPECCY-DOS SZUBRUTINOK ÁTTEKINTÉSE

Idáig haladva az olvasásban az olvasó már eleget tud a SPECCY-DOS -ról ahhoz, hogy a következő pontokat, amelyek a SPECCY-DOS szubrutinjait ismertetik, maradéktalanul megértse. E pontban röviden áttekintjük az ismertetésre kerülő szubrutinokat, amelyeket két csoportra oszthatunk, miszerint /i/ változó cimü szubrutinok és /ii/ kötött cimü szubrutinok. A változó cimü szubrutinokra jellemző, hogy belépési cimeik a SPECCY-DOS egyes verzióiban eltérőek lehetnek. Arról, hogy hogyan lehet e szubrutinokat verziótól függetlenül hivni, a következő fejezet ad felvilágositást.

A kötött cimü szubrutinok körébe elsősorban restart területen futó szubrutinokat sorolunk, másrészt olyan belépési pontokat, amelyeknek cime egyéb okokból verziótól függetlenül kötött. A IV.-1 táblázat a változó cimü szubrutinok, a IV-2. táblázat a kötött cimü szubrutinok és belépési pontok funkcióit tartalmazza. /Az alkalmazott elnevezések természetesen nem kötelező jellegüek, a felhasználó módosithatja őket/.

VÁLTOZÓ CIMÜ SZUBRUTINOK

Nr	Név	FUNKCIÓ
1.	SAVE	Ugyanaz, mint a SAVE*....DOS-BASIC parancs funkciója.
2.	LOAD	Ugyanaz, mint a LOAD *...DOS-BASIC parancs funkciója.
3.	DELETE	File törlése a diszkről.
4.	COPY	Ugyanaz, mint a COPY *... DOS-BASIC parancs funkciója
5.	RENAME	File átnevezése a diszken.
6.	SRFILE	Adott file filenév alapján való megkeresése a diszken
7.	CREFIL	Adott méretü,tipusu és nevü üres file létrehozása a diszken.
8.	COLECT	Adott file-hoz tartozó adatterület-beli szektorcimek kigyüjtése a memóriába.

VÁLTOZÓ CIMÜ SZUBRUTINOK /folytatás/

Nr	Név	FUNKCIÓ
9.	FLREAD	Megnyitott file N-edik szektorának leolvasása
10.	FLWRTE	Megnyitott file N-edik szektorának irása.
11.	RDFILE	Megnyitott file összes szektorának leolvasása
12.	WRFILE	Megnyitott file összes szektorának felirása.
13.	SECTRD	Adott diszk-szektor leolvasása fizikai cime alapján.
14.	SCWRIT	Adott diszk-szektor felirása fizikai cime alapján.
15.	RDYCON	Drive müködésre kész állapotának ellenőrzése.
16.	TYPONE	1-es tipusu FDC parancsok kiadása
17.	HEXDC 1	8 bites hexadecimális-decimális konverter
18.	HEXDC2	16 bites hexadecimális-decimális konverter
19.	ERASE	Képernyő és attributumterület törlése

IV-1 táblázat

KÖTÖTT CIMÜ SZUBRUTINOK

Nr	Név	CIM	FUNKCIÓ
20.	ERROR*	ØØ1ØH	Program futásának megszakitása hibajelzéssel.
21.	DSTRAM	ØØ18H	Szöveg kiirása a képernyőre, vagy ZX printerre.
22.	BASRUT	ØØ28H	BASIC ROM-ban lévő szubrutinok hivása, ha a DOS van kiválasztva.
23.	IDENTS	ØØ3ØH	Diszk azonositása
24.	ROMXCG*	ØØ46H	Visszaváltás DOS ROM-ról BASIC ROM-ra

* Nem valódi szubrutin, nem tér vissza a hivási helyre.

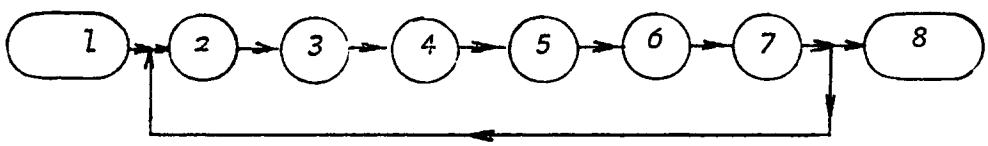
IV-2 táblázat

V. A SPECCY-DOS SZUBRUTINJAINAK HIVÁSA MÁS PROGRAMOK ÁLTAL

Ebben a fejezetben azokat a tudnivalókat ismertetjük, amelyek a SPECCY-DOS szubrutinjainak felhasználói programokból való hivására általánosan, azaz a hivott szubrutin egyedi tulajdonságaitól függetlenül érvényesek.

V-1. ROM VÁLTÁSOK

Általános esetben a BASIC RAM-ban futó felhasználói program mind a BASIC-ROM, mind a SPECCY-ROM szubrutinkészletét használhatja. Ehhez természetesen az szükséges, hogy az adott szubrutin meghivása előtt a megfelelő ROM legyen kiválasztva. A ROM-ok közötti váltást a II-1 pontban már ismertettük. Az ott elmondottakat itt annyival egészitjük ki, hogy mindaddig amig a DOS van kiválasztva, az **interrupt-oknak** letiltva kell lenniük. A felhasználói program sematikus folyamatábráját a fentiek alapján az V-1. ábrán láthatjuk.



1./ Kiindulás BASIC-ből. 2,6,; Saját müveletek végzése a BASIC RAM-ban, miközben mindegy, hogy a BASIC ROM /2/, vagy a SPECCY-ROM /6/ van kiválasztva.

3./ A BASIC ROM szubrutinjainak hivása. 4./ Átváltás SPECCY ROM-ra.

5./ A SPECCY ROM szubrutinjainak hivása. 7./ Visszaváltás BASIC ROM-ra.

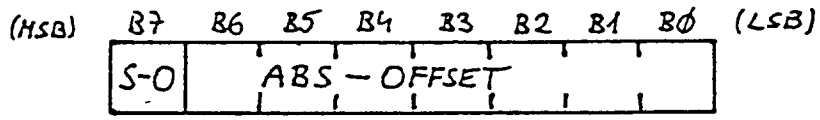
8./ Visszatérés BASIC-be, azaz a program vége.

V-1. ábra

V-2. A VÁLTOZÓ CIMÜ SPECCY SZUBRUTINOK HIVÁSA

A IV-1 táblázatban közölt SPECCY-DOS szubrutinok ROM-beli cime az idő haladtával kibocsátott mind nagyobb tudásu ROM verziókban eltérő lehet. Jóllehet, az eltérések kicsik, első közelitésben, ez mégis lehetetlenné tenné olyan felhasználói programok készitését, amelyek módositás nélkül minden verzióval kompatibilisek. Az ennek a problémának a megoldására a SPECCY-DOS-ban alkalmazott eljárás lényege, hogy az aktuális hivási cimek az adott ROM verzió és egy un. referencia táblázat birtokában egyszerü számitással software uton meghatározhatók, azaz a szubrutinok verzió-függetlenül hivhatók. A referencia táblázat az egyes szubrutinokhoz tartozó un. referencia cimeket tartalmazza. Az egyes ROM verziókban, a verziótól független rögzitett cimeken az egyes szubrutinok referencia cimeinek és aktuális cimeinek különbsége található meg, 8 biten ábrázolva.

Ennek az un. offset byte-nak a felépitése az V-2. ábrán látható.



Az OFFSET definiciója: OFFSET = /Aktuális cim/ - /Referencia cim/

S-O: Az OFFSET előjele: Ø, ha OFFSET ⩾ Ø
1, egyébként

ABS- OFFSET: az OFFSET abszolut értéke: Ø....127

V-2. ábra

Az egyes SPECCY szubrutinok referencia cimeit és a hozzájuk tartozó offset byte-ok SPECCY-ROM-beli cimeit az V-1 táblázat mutatja.
Ezen információk birtokában a felhasználónak megvan a lehetősége, a kivánt szubrutinok verzió-független hivására. Mi a következőkben két kidolgozott módszert ismertetünk, az un. általános hivást és az egyedi hivást.

V-1. táblázat

CODE #	SZUBRUTIN NEVE	REFERENCIA CÍM	OFFSET BYTE CÍME	BEMENETI * REGISZTERHASZNÁLAT
1	SAVE	ØE29H	1FFFH	IX,H,L
2	LOAD	ØE45H	1FFEH	IX,H,L
3	DELETE	106FH	1FFDH	H,L
4	COPY	1184H	1FFCH	D,E,H,L
5	RENAME	111FH	1FFBH	D,E,H,L
6	SRFILE	Ø669H	1FFAH	A,H,L
7	CREFIL	Ø73BH	1FF9H	A,H,L
8	COLECT	ØA93H	1FF8H	A
9	FLREAD	ØBF6H	1FF7H	A,B,H,L
10	FLWRTE	ØC4FH	1FF6H	A,B,H,L
11	RDFILE	ØDØ6H	1FF5H	A,H,L
12	WRFILE	ØD61H	1FF4H	A,H,L
13	SECTRD	Ø1A4H	1FF3H	D,E,H,L
14	SCWRIT	Ø211H	1FF2H	D,E,H,L
15	RDYCON	Ø12AH	1FF1H	———
16	TYPONE	1D14H	1FFØH	———
17	HEXDC1	ØØABH	1FEFH	E
18	HEXDC2	ØØBEH	1FEEH	D,E
19	ERASE	13AFH	1FEDH	———

*: Azon processzorregiszterek, amelyek bemeneti paramétert tartalmaznak

#: Az általános hivásnál használt szubrutinkód

V-2.1. ÁLTALÁNOS HIVÁS

A változó cimü SPECCY-DOS szubrutinok általános hivásának lényege, hogy az egyes szubrutinok egyedi hivása helyett mindig ugyanazt a szubrutint hivjuk, és egy 1 byte-os kóddal tudatjuk ezzel a szubrutinnal, hogy melyik szubrutint kivántuk meghivni. Ha ezt a közös szubrutint SPECCY szubrutinnak nevezzük, akkor az általános hivás a következőképpen néz ki:

```
CALL    SPECCY
DB      SRCODE
```

ahol SRCODE a meghivandó szubrutin kódja /1...19/. Az egyes szubrutinokhoz tartozó kód az V-1. táblázat első oszlopában található meg. A SPECCY szubrutin, amely a felhasználói program része, a szubrutinkód és a referenciatáblázat ismeretében kiszámitja az adott szubrutin aktuális cimét, anélkül, hogy a regiszterek tartalmát módositaná, majd a vezérlést a kivánt szubrutinnak adja át ugy, hogy az lefutása után az eredeti hivási helyre tér vissza. Az általános hivás egyetlen hátránya,hogy a SPECCY szubrutin, a referenciatáblázattal együtt 79 byte programterületet igényel. A SPECCY szubrutint Z8Ø gépi kódban az V-3 ábra mutatja.

```
E3  C5  46  23  33  33  E3  3B  3B  E5
D5  F5  21  XL  XH  11  ØØ  2Ø  23  23
1B  1Ø  FB  1A  5E  23  66  6B  CB  7F
F5  E6  7F  5F  16  ØØ  ED  52  F1  2Ø
Ø2  19  19  F1  D1  C1  E3  C5  E3  C1
C9
256xXH+XL   29  ØE  45  ØE  6F  1Ø  84  11
1F  11  69  Ø6  3B  Ø7  93  ØA  F6  ØB
4F  ØC  Ø6  ØD  61  ØD  A4  Ø1  11  Ø2
2A  Ø1  14  1D  AB  ØØ  BE  ØØ  AF  13
```

V-3 ábra

Az ábra első része a SPECCY szubrutin, második része pedig a referenciatáblázat, amelyet az V-1. táblázatból képeztünk. A szubrutin és a táblázat egymástól függetlenül a programterület bármely részén elhelyezhető, azonban a SPECCY szubrutinnak ismernie kell a referenciatáblázat kezdőcimét. Ha például a referenciatáblázat kezdőcime 8ØØØH, akkor XH és XL értéket a következőképpen számitjuk ki:

X = 8ØØØH - 2H = 7FFEH XH = 7FH és XL = FEH

Az általános hivást akkor célszerü használni, ha viszonylag sok SPECCY-DOS szubrutint használunk egy programban és/vagy ha elegendő terület áll rendelkezésre.

V-2.2.EGYEDI HIVÁS

Az egyedi hivás lényege, hogy a felhasználói programban minden egyes hivandó változó cimü szubrutinhoz tartozik egy kis, néhány byte-os szubrutin, ami az adott szubrutin cimhelyes hivásáról gondoskodik. Szükség van még egy, ugyancsak rövid, az egyedi szubrutinok által közösen használt szubrutinra, amit itt "COMCAL"-nak neveztünk el. Az egyedi szubrutinoknak célszerüen a SPECCY-DOS-ban is használt neveket adhatjuk. Igy pl. a SAVE nevü SPECCY-DOS szubrutin hivása a felhasználói programban CALL SAVE lesz. Az egyedi szubrutinok felépitése a következő:/V-4.ábra/

```
PUSH    HL          | REFADDR:A meghivandó SPECCY-DOS
PUSH    AF          |         szubrutin referenciacime
LD      HL,REFADDR  |
LD      A,/OFFS-ADDR/ | OFFS-ADDR:A meghivandó SPECCY-DOS
CALL    COMCAL      |           szubrutin offset byte-
POP     AF          |           jának cime
EX      /SP/,HL     | Ezek konkrét értékeit az V-1.táb-
RET                 | lázatból vehetjük.
```

V-4. ábra

529

Az alábbi két példa a SAVE és az ERASE nevü SPECCY-DOS szubrutinokhoz tartozó egyedi szubrutinokat mutatja, a szubrutinok eredeti nevét megtartva.

Példák az egyedi szubrutinokra:

```
SAVE:   PUSH    HL          ERASE:  PUSH    HL
        PUSH    AF                  PUSH    AF
        LD      HL,ØE29H            LD      HL,l3AFH
        LD      A,/lFFFH/           LD      A,/lFEDH/
        CALL    COMCAL              CALL    COMCAL
        POP     AF                  POP     AF
        EX      /SP/,HL             EX      /SP/,HL
        RET                         RET
```

Az egyedi szubrutinok által közösen hivott, itt "COMCAL"-nak nevezett szubrutin a következő:/V-5.ábra/

```
COMCAL:     BIT     7,A     | A COMCAL szubrutin a hivandó
            PUSH    DE      |
            PUSH    AF      | SPECCY-DOS szubrutin aktuális
                            | cimét
            AND     7FH     |
            LD      E,A     | /output-HL/ számitja ki a re-
                            | ferenciacim
            LD      DØ      |
            SBC     HL,DE   | /input-HL/ és az offsetbyte
                            | /input-A/
            POP     AF      |
            JR      NZ,ADDR-|
                    DONE    | alapján
            ADD     HL,DE   |
            ADD     HL,DE   |
ADDR-DONE:  POP     DE      |
            RET
```

V-5. ábra

Megjegyezzük, hogy az egyedi szubrutinok hossza esetenként tovább csökkenthető, ha kihasználjuk, hogy nem minden regiszter tartalmának kell változatlannak maradnia az egyedi szubrutin belépési pontjától a SPECCY-DOS szubrutin belépési pontjáig. Ez a hivott szubrutin bemeneti interface-étől, az általa módositott regiszterektől és magától a felhasználói programtól függ. Ezért tüntettük fel az V-1. táblázatban a bemeneti regiszterhasználatot is. Például a fenti ERASE szubrutin hossza lØ byte-ra csökken, ha A és HL módositható:

```
ERASE:   LD      HL, l3AFH,
         LD      A, /lFEDH/
         CALL    COMCAL
         JP      /HL/
```

VI. A SPECCY-DOS RAMTERÜLETE:

A SPECCY-DOS ramterülete a 2ØØØH - 23FFH cimeket foglalja le. Ennek az 1Kbyte-nyi területnek a rendszer csak egy részét használja. A 232ØH - 23FFH közötti 224 byte hosszu terület szabad, ezt a felhasználói programok minden megkötés nélkül használhatják. A felhasználó saját felelősségére a 2ØØØH - 231FH közötti területet is használhatja, ez esetben azonban fennáll annak a lehetősége, hogy a SPECCY-DOS szubrutinok hivásakor ennek a területnek egyes részei felüliródnak. Adott esetben próbálkozással dönthetjük el, hogy a használni kivánt terület egy konkrét felhasználói programban használható-e, vagy nem. Ezt azonban csak akkor tegyük, ha a 232ØH - 23FFH közti terület nem elegendő. Megadunk azonban olyan cimeket, amelyeknél biztos , hogy nem iródnak felül, feltéve, hogy az e kézikönyvben közölt SPECCY-DOS szubrutinokat használjuk. Ezek a következők:

2ØØ1H.... 2ØØDH; 2ØA4H.... 2ØCBH; 22CDH....2317H

Ez további 128Byte-ot jelent a felhasználói programok változóinak számára.
A rendszer által használt RAM terület egyrészt bufferekre, másrészt memóriaregiszterekre oszlik. A DOS által használt bufferterületek a következők:

DOSBUF:	2Ø CCH....21CBH;	A rendszer 256 Byte-os fő buffere
PRTBUF:	21 CCH....22CBH;	256 Byte-os buffer,főként az aktuális file szektorcimeinek tárolására
UNIBUF:	2Ø64H.....2Ø83H;	32 Byte-os,változó célokra használt buffer
DIRBFØ:	2Ø24H.....2Ø3CH;	2 db 25 byte-os buffer, az aktuális file/ok/
DIRBF1:	2Ø3DH.....2Ø55H	file directory-jának tárolására

Azoknak a memóriaregiszteréknek a cimét és felhasználási módját, amelyeknek ismerete a felhasználó számára fontos lehet, a Volük kapcsolatos szubrutinoknál ismertetjük. Végül emlékeztetünk arra, hogy a DOS- RAM csak akkor hozzáférhető, ha a DOS van kiválasztva.

VII. A SPECCY-DOS SZUBRUTINJAINAK ISMERTETÉSE

E fejezetben az egyes SPECCY-DOS szubrutinok konkrét software interface-eit ismertetjük. A COPY szubrutin kivételével minden itt ismertetett szubrutinra vonatkozik, hogy az adott diszk művelet azon a drive-on történik, amelyik a szubrutin meghívásakor ki van jelölve. Ezt a 3B4lH cimre legutoljára kiadott kódszó határozza meg, és ugyanezt a kódszót kell tartalmaznia a DFLAGl nevü, 2ØØØH cimü memóriaregisztternek is /Lsd: 11-l pont/. Hasonlóképpen, az ismertetett szubrutinok gondoskodnak arról, hogy kilépéskor DFLAGl-ben a 3B4lH-ra legutoljára kiadott kódszó található.

l; SAVE Ugyanaz a funkciója, mint a SAVE*..... DOS-BASIC parancsé.

Input: - IX egy 18 byte hosszu file leiró terület kezdetére mutat, amely bárhol lehet, a DOS vagy a BASIC RAM-ban.

- HL a diszkre elmentendő terület kezdőcimére mutat.

Output: -

Modified: A,F,B,C,D,E,H,L

A 18 Byte hosszu terület felépitése és HL tartalma.
A terület byte-jait ØØH-tól llH-ig számozzuk. Az itt nem definiált értékek nullák.

ØØH: A létrehozandó file tipuskódja: ØØ: BASIC program a változóival
Øl: Számtömb file
Ø2: Karaktertömb file
Ø3: CODE, vagy SCREEN$ file

ØlH....Ø8H: A file neve/ASCII/ szükség szerint space-ekkel /2ØH/ 8 karakterre kiegészitve, hasznos karakterek balra helyezve.
Megj.: "filenév" alatt a továbbiakban mindig ezt kell érteni!

Ø9H,ØAH: Tartalék byte-ok. Tartalmuk: ØØH,ØØH.

ØBH, ØCH: * A létrehozandó file hossza, azaz byte-jainak száma.

---

* Itt, és a továbbiakban is, a két byte széles értékek első byte-ja az alacsonyabb, a második a magasabb helyiértékü.

ØBH, ØCH értéke a különböző file tipusok esetén. /Lsd: még SPECTRUM leirás/.

VII-1. Táblázat

tipuskód /ØØ4/	/ØBH,ØCH/ = A file hossza.
ØØH /BASIC	/E-LINE/ - /PROG/ +1
OlH,O2H /TÖMB/	A tömb hosszleirója által meghatározott érték
Ø3H /CODE/	A diszkre mentendő byte-ok száma

ØDH, ØEH, ØFH, lØH: Ezen byte-ok értelmezése a file tipusától függ:

tipuskód = ØØH /BASIC/ ØEH: 8ØH-át tartalmaz, ha a programot LINE megadása nélkül kivánjuk SAVE-elni.
ØDH, ØEH: az autorun sorszámot tartalmazza, egyébként.
ØFH,lØH: a program változók nélkül vett hosszát tartalmazza,azaz a (VARS) - (PROG) értéket.

Tipuskód = ØlH, vagy Ø2H: ØEH tartalmát a III-6. ábra mutatja.

Tipuskód = Ø3H: ØDH, ØEH, a diszkre mentendő memóriaterület kezdőcimét tartalmazza.
ØFH, lØH a futási kezdőcimet tartalmazza, ha ez a ØDH,ØEH - tartalmától eltérő,egyébként nulla.

llH: A létrehozandó file attributumait tartalmazza. Felépitése a III-7. ábrán látható, azzal a kiegészitéssel, hogy jelen esetben B7 /MSB/= Ø.

Az input HL tartalma a különböző file tipusok esetén:
/Lásd még a SPECTRUM leirás 24. fejezetét/.

tipuskód /ØØH/	input HL
ØØH /BASIC/	(PROG)
ØlH,Ø2H /TÖMB/	A tömb hossz-leiróját követő byte cime.
Ø3H /CODE/	A diszkre mentendő terület kezdőcime /azonos ØDH,ØEH-vel/

VII-2.táblázat

2. LOAD Ugyanaz a funkciója, mint a LOAD *....DOS-BASIC paranccsé. Az egyetlen kivétel az, hogy olyan BASIC file-ok betöltésekor, amelyeket LINE megadásával rögzitettek, a betöltött BASIC program nem kezd azonnal futni, mint LOAD *-nál.

input: - IX egy 17 byte hosszu file leiró terület kezdetére mutat, amely bárhol lehet a BASIC, vagy a DOS RAM-ban.

- HL a file tipusától függő értéket tartalmaz.

output: -

modified: A,F,B,C,D,E,H,L

A 17 byte hosszu terület felépitése és HL tartalma:

A terület első 11 byte-jának tartalma teljesen megegyezik a SAVE szubrutinnál tárgyalt 18 byte-os leiróterület első 11 byte-jának tartalmával. A többibyte tartalma a file tipusától függ. Az itt nem definiált byte-ok ØØH-t tartalmaznak.

Tipuskód = ØØH: ØBH, ØCH, a SPECTRUM-BAN pillanatnyilag tartozkódó BASIC program és változóinak együttes hosszát tartalmazza, azaz: (E-LINE) - - (PROG) + 1

ØFH, 1ØH a program önmagában vett hosszát tartalmazza, azaz ØFH, 1ØH = (VARS) - (PROG)

input HL = (PROG).

Típuskód = ØlH, vagy Ø2H: ØBH,ØCH a betöltendővel azonos nevü tömb hosszát tartalmazza, ha van ilyen a SPECTRUM-ban.

/TOMB/

input HL: a betöltendővel azonos nevü tömb hosszleiróját követő byte cimét tartalmazza, ha van ilyen.
HL = Ø, ha nincs ilyen tömb.

Típuskód = Ø3H: ØBH, ØCH a betöltendő program megengedhető maximális hosszát tartalmazza, ha specifikáljuk. ØBH, ØCH = ØØ,ØØ, ha nem.

/CODE/

input HL: betöltési kezdőcim, ha specifikáljuk,
HL = ØØØØ egyébként.

3. DELETE File törlése a diszkről. /Lásd: ERASE * parancs/

input: - HL a filenév első karakterére mutat a memóriában.

output: - -

modified: A,F,B,C,D,E,H,L

4. COPY File másolása. /Lásd: COPY * DOS-basic parancs/

input: - HL az output filenév első karakterére mutat a memóriában.
- DE a forrás filenév első karakterére mutat a memóriában.
- A DOS RAM-ban 2ØBBH címen levő /névtelen/ rendszerváltozó az output drive kódját tartalmazza. /VII-1 ábra/
- A DOS RAM-ban 2ØC6H címen lévő /névtelen/ rendszerváltozó a forrás drive kódját tartalmazza /VII-1 ábra/.

/MSB/ B7	B6	B5	B4	B3	B2	B1	B0 /LSB/
Ø	Ø	Ø	Ø	Ø	Ø	drive	

		B1	B0	Drive
2ØC6H	forrás	Ø	Ø	drive Ø
		Ø	1	drive 1
2ØBBH	output	1	Ø	drive 2
		1	1	drive 3

VII-1 ábra

- A DOS-RAM COPY-FG nevü, 22CCH cimü rendszerváltozója: Ø, ha engedélyezzük, hogy egydrive-os COPY esetén a forrás és az output diszk különböző legyen.
COPYFG ≠ Ø, egyébként.

output: - -

modified: A,F,B,C,D,E,H,L

comments: Egy drive-on belüli másolás esetén a rendszer mindig lehetőséget ad arra, hogy a forrás és az output diszk különböző legyen, ha a másolást BASIC-ből végezzük. Ehhez azonban szükséges, hogy a fordulók közötti valaki leüsse az ENTER gombot. A COPYFG által való választási lehetőség az egy drive-on belüli, egy diszken való automatikus /ENTER nélküli/ másolást teszi lehetővé.

5. RENAME File átnevezése /Lásd: FN * DOS-BASIC parancs/

input: - HL az uj filenév első karakterére mutat a memóriában.
- DE a régi filenév első karakterére mutat a memóriában.

output: - -

modified: A, F,B,C,D,E,H,L

6. SRFILE Adott nevü file megkeresése a diszken. Ha a file létezik, akkor a szubrutin a file directory-ját bemásolja a kivánt directory bufferbe.

input: - HL a filenév első karakterére mutat a memóriában.
- A - ban: Ø, ha DIRBFØ-ba/kivánjuk beolvastatni a file,
≠Ø, ha DIRBF1-be/ 25 byte-os directory-ját.

output: - CY = 1, ha nincs a diszken az adott nevü file.
= Ø, ha az adott nevü file a diszken van.

- E-ben annak a TØ-on belüli szektornak a cime, amelyben az adott file directory-ja található. Ennek a szektornak a másolata DOSBUF-ban található.

- HL a DOSBUF-on belül az adott file directory első byte-jának cimét tartalmazza.

modified: A,F,B,C,D,E,H,L

7. CREFIL Formálisan teljesen kész, de még "üres" kitöltetlen file létrehozása a diszken, azaz a szükséges adatterület lefoglalása és a file bejegyzése a diszk directory-ba.
A kialakult file directory bemásolása a DRBFØ/DIRBF1-be.

input: - HL egy 18 byte hosszu file leiró terület kezdőcimet tartalmazza.
E terület felépitése megegyezik a SAVE szubrutinnál ismeretett 18 byte-os file leiró terület felépitésével.

- A-ban: Ø, ha DIRBFØ-ba } /kivánjuk elhelyeztetni a file kialakitott 25 byte-os directory-ját.
≠Ø, ha DIRBF1-be }

output: DIRBFØ-ban,ill. DIRBF1-ben a file kialakitott directory-ja.

modified: A,F,B,C,D,E,H,L.

comments: A szubrutin által létrehozott file csak abban tér el egy kész file-tól,hogy az adatterülete nem tartalmaz hasznos adatot.

8. COLECT *A COLECT szubrutin adott file szektorcimeinek irás, vagy olvasás céljából való összegyüjtésére szolgál.*

A bufferméretek korlátozott volta miatt egy file-hoz tartozó szektorcim nem mindig gyüjthető egyszerre össze. Ezért a COLECT szubrutin ciklikus hivásra készült. A szektorcimek kigyujtése a PRTBUF bufferbe történik. A szubrutin vagy a teljes PRTBUF-ot, vagy csak egyik felét tölti fel, attól függően, hogy egyidejüleg csak egy, vagy pedig két file szektorcimeire van szükségünk.

Ha a file kétszeres rögzitésü, akkor a szubrutin a PRTBUF-nak, /illetve valamelyik felének/ az alsó felét a fő file, a felső felét a mellékfile azonos adattartalmu részeinek szektorcimeivel tölti fel.

input: - *A regiszter, mint input kontrol regiszter. Az egyes bitek értelmezése:*

B0 /LSB/ = 0, ha a DIRBF0	tartalmazza annak a file-nek a directoryját, amelynek szektorcimeit keressük.
= 1, ha a DIRBF1	

B1 = 0, ha a teljes PRTBUF	töltendő fel szektorcimekkel. Ez attól függ, hogy a PRTBUF-ban egyidejüleg **csak** egy, vagy pedig 2 file szektorcimeit tároljuk /pld. COPY-nál két file szektorcimeit kell tárolni/
1, ha csak a fél PRTBUF	

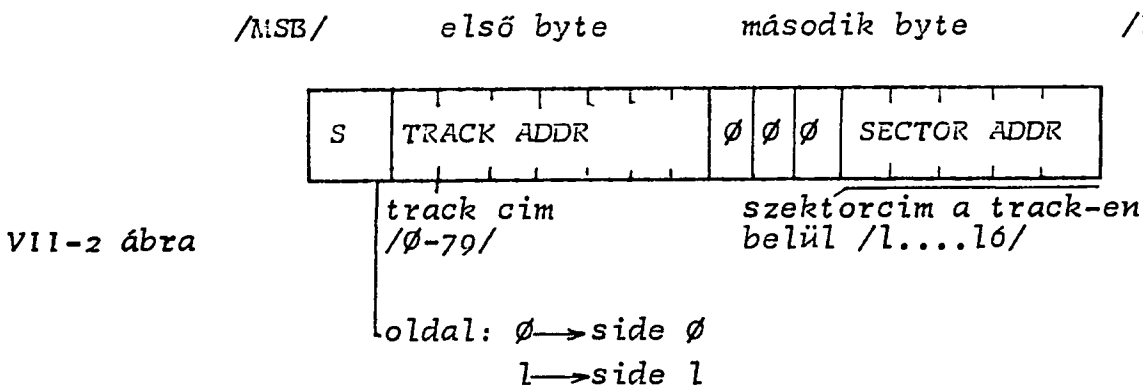
Ha B1 = 1, akkor B0 = 0 esetén a PRTBUF alsó, B0=1 esetén PRTBUF felső felét tölti fel a szubrutin.

B2 = 0, a COLECT szubrutin első hivásakor. Ekkor az FSNR és az LSTMS nevü regiszterekben /lsd. később/ még nincs hasznos adat, és a file szektorcimeinek kigyüjtése DIRBF0, vagy DIRBF1 tartalma alapján történik.

= 1 a COLECT szubrutin soronkövetkező hivásaikor. Ekkor az ujabb szektorcimek kigyüjtése, az FSNR és az LSTMS nevü regiszterek tartalma alapján történik. B2 = 1 esetén e regiszterekben a szubrutin megelőző hivásakor kialakult értékeknek kell lenni /ld: output/

output: /1/A PRTBUF buffer megfelelőrészében a kigyüjtött szektorcimek. Egy-egy szektorcim 2 Byte szélességü és a VII-2 ábrán látható felépitésü.

VII-2 ábra



A VII-2. ábra szerinti felépitésü 2 byte-os szektorcimek a PRTBUF-ban folyamatosan követik egymást. A file N-edik szektorának cimét az N+1-edik szektor cime követi. A szubrutin egy meghivásakor kigyüjtött szektorcimek közül az elsőnek a cimét a VII-3. táblázat tartalmazza.

VII-3. táblázat

KEZDŐCIM	ESET			
PRTBUF	B1 = Ø	BØ = X	NRF	
PRTBUF	B1 = Ø	BØ = X	DRF	MF
PRTBUF + 128	B1 = Ø	BØ = X	DRF	SF
PRTBUF	B1 = 1	BØ = Ø	NRF	
PRTBUF	B1 = 1	BØ = Ø	DRF	MF
PRTBUF + 64	B1 = 1	BØ = Ø	DRF	SF
PRTBUF + 128	B1 = 1	BØ = 1	NRF	
PRTBUF + 128	B1 = 1	BØ = 1	DRF	MF
PRTBUF + 192	B1 = 1	BØ = 1	DRF	SF

Jelölések: BØ, Bl = input A/bit Ø ill. bit l. X = DON'T CARE NRF = egyszeresen rögzitett file, DRF = kétszeresen rögzitett file, MF = főfile /DRF-nél/, SF = mellékfile /DRF-nél/

A hivásonként kigyüjtött szektorcimek maximális száma Bl = Ø esetén 128, Bl = l esetén 64. Ez kétszeres rögzitésü file esetén a file adattartalma szempontjából 64, ill. 32 szektornak az elérését teszi lehetővé, mivel ez esetben a file minden /logikai/ szektorához két /fizikai/ diszk szektor tartozik, egy a fő-, egy a mellékfile-ban.

Output: /2/ A szubrutin megadja a kigyüjtött szektorok közül a file-on belül legkisebb és legnagyobb sorszámunak a sorszámát. Nevezzük ezeket az értékeket FSNRMIN, ill. FSNRMAX -nak. Mivel egy file maximum 256 szektorból állhat, FSNR értéke Ø....255 lehet. Az FSNR értékek megadása a DRBPTB nevü, 2Ø17H cimü rendszerváltozót követő 4 Byte-nyi területen történik. E területet mutatja a VII-3 ábra. Azt, hogy mely esetben melyik rekeszben található meg FSNRMIN, ill. FSNRMAX, a VII-4. táblázat tartalmazza. /Jelöléseket lsd: VII-3 táblázat/.

2Ø17H	2Ø18H	2Ø19H	2Ø1AH	2Ø1BH
	FSNR MIN	FSNR MAX	FSNR MIN	FSNR MAX
	+1	+2	+3	+4

DRBPTB = 2Ø17H

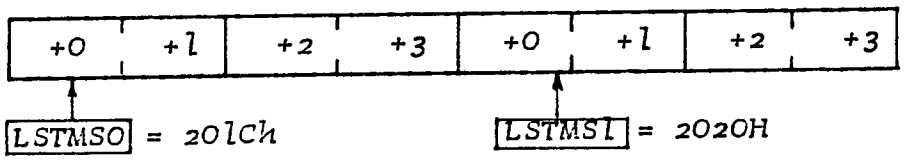
VII-3.ábra

Bl	BØ	$FSNR_{MIN}$ cime	$FSNR_{MAX}$ cime
Ø	X	DRBPTB+1	DRBPtB+2
1	Ø	DRBPTB+1	DRBPTB+2
1	1	DRBPTB+3	DRBPTB+4

X = DON'TCARE

VII-4.táblázat

Output: /3/ A szubrutin megadja annak a szektorpointernek /ill. kétszeres rögzitésü file-nál szektorpointereknek/ a térképterületen belüli cimét, ami az előző pontban emlitett FSNR max sorszámu file szektornak a cimét tartalmazza. Ez az LSTMS nevü, 201Ch kezdőcimü 8 byte szélességü rendszerváltozó területen történik, amelyet a VII-4. ábrán látunk.



VII-4. ábra

+0, vagy +2: A szektorpointer track és szektorcime
+1, vagy+3: A szektorpointer szektoron belüli sorszáma.

A két byte széles mezők /+0,+1, vagy +2,+3/ tartalmáról lsd. III-8. ábra.

Azt, hogy az egyes esetekben melyik két byte-os mező tartalmazza a szektorpointer cimét, a VII-5 táblázat tartalmazza.

MEZŐ CIME		ESET			
LSTMSØ	+0,+1	B1 = 0	BO = X	NRF	
LSTMSO	+0,+1	B1 = 0	BO = X	DRF	MF
LSTMSO	+2,+3	B1 = 0	BO = X	DRF	SF
LSTMSO	+0,+1	B1 = 1	BO = 0	NRF	
LSTMSO	+0,+1	B1 = 1	BO = 0	DRF	MF
LSTMSO	+2,+3	B1 = 1	BO = 0	DRF	SF
LSTMS1	+0,+1	B1 = 1	BO = 1	NRF	
LSTMS1	+0,+1	B1 = 1	BO = 1	DRF	MF
LSTMS1	+2,+3	B1 = 1	BO = 1	DRF	SF

Jelölések: Lsd VII.-3 táblázat

VII-5. táblázat

Output: /4/ A szubrutin a DRBPTB nevü B017H cimü rekesz B6, ill. B7 /MSB/ bitjét 0-ba állitja. Ezek a bitek az FLREAD szubrutin ciklikus hivásai során annak a megjegyzésére szolgálnak, hogy kétszeres rögzitési file esetén az olvasás a fő, vagy a mellékfile-ból történik-e. Ezen bitek nullázásával a COLECT szubrutin arra készteti az FLREAD szubrutint, hogy az olvasást a főfile olvasásával kezdje, és csak akkor térjen át a mellékfilera, ha a főfile-ban hibát talál. /Lásd még:FLREAD/

input - B0 = 0 ⟹ B7/DRBPTB ← 0
input - B0 = 1 ⟹ B6/DRBPTB ← 0

modified: A,F,B,C,D,E,H,L

9. FLREAD: Olvasásra előkészitett file* N-edik szektorának kiolvasása tetszőleges memóriacimre.

input: - A regiszter, mint input kontrol regiszter. Bitenkénti értelmezése:

B0 = 0, ha DIRBF0-ban 1, ha DIRBF1-ben	van elhelyezve az olvasandó file directory-ja.

B1 = 0, ha a teljes PRTBUF 1, ha csak a fél PRTBUF	tartalmazza a file szektorcimét./Lásd még COLECT input/

- B-ben: N /0....255/

- HL-ben az a memóriakezdőcim, ahonnal kezdve a szektort beolvassuk. Ha HL= 0, akkor az olvasás a DOSBUF-ba történik.

- A DRBPTB+1,+2,+3+4 memóriaregiszterek közül az input A-által a VII-4 táblázat szerint meghatározott Kettőben a pillanatnyi állapotra érvényes FSNRMIN és FSNRMAX érték. /Lásd még: COLECT output 2/

- A PRTBUF input A által a VII-3 táblázat szerint meghatározott részében az FSNRMIN és FSNRMAX közé eső file szektorok szektorcimei /Lásd még: COLECT output 1./

- A DRBPTB nevü rendszerváltozó B6, vagy B7 bitjén 0, vagy 1 aszerint, hogy kétszeresen rögzitett file olvasása esetén az N-edik szektort a fő, vagy a mellékfile-ból kell-e olvasni. Ezt a bitet normál esetben a szubrutin előző hivása, vagy a COLECT szubrutin állitja be. DRBPTB meghatározó bitje:
B7 /MSB/, ha input B0 = 0, egyébként B6.
B6 ill. B7 = 0: fő file olvasása, B6, ill. B7 = 1: mellékfile.

Output: - DRBPTB regiszter input-ban meghatározott B6, vagy B7 bitjén a bemeneti érték, ha nem történt olvasási hiba, illetve a bemeneti érték **inverze** ha olvasási hiba történt, de a file kétszeres rögzitésü, és a párhuzamos "másik" file azonos sorszámu szektorának olvasása sikeres volt. /Ekkor vált át a file olvasása a főfile-ról a mellékfile-re, ill. forditva./

modified: A,F,B,C,D,E,H,L

comments: - A szubrutin ciklikus hivásra **készült** de másképp is alkalmazható.

- N-nek teljesitenie kell a következő feltételt:

$$FSNRMIN \leqslant N \leqslant FSNR\ MAX$$

Ellenkező esetben ERROR 53 keletkezik.

- *: A file olvasásra előkészitett, ha a fenti input specifikációk teljesülnek.

10. FLWRTE: Irásra előkészitett file N-edik szektorának irása tetszőleges memóriacimről.

input: - A regiszter, mint input kontrol regiszter. Bitenkénti értelmezése:

B0 = 0, ha DIRBF0 | tartalmazza az irandó file
1, ha DIRBF1 | directory-ját.

B1 = 0, ha a teljes PRTBUF | tartalmazza a file
1, ha csak a féle PRTBUF | szektorcimeit./Lsd: COLECT input/

B4 az elvégzendő müveletet határozza meg. B4 = 0: egyszeres rögzitésü file, vagy kétszeres rögzitésü file főfilejának irása. B4 = 1: kétszeres rögzitésü file mellékfile-jának irása.

- B-ben: N /0....255/

- HL-ben az a memóriakezdőcim, ahonnan a szektort felirjuk, ill. HL = 0, ha a DOSBUF-ból irjuk a szektort.

- DRBPTB+1,+2,+3,+4 memóriaregiszterekben, ill. PRTBUF-ban ugyanaz, mint amit az FLREAD szubrutinnál közöltünk.

Output: - -

modified: A,F,B,C,D,E,H,L

comments: A szubrutinra az FLREAD szubrutinnal közös megjegyzések vonatkoznak. /Lsd. FLREAD-comments/

11. RDFILE *Olvasásra előkészitett file összes szektorának beolvasása adott memóriacimtől kezdve.*

input: - *A regiszter, mint input control regiszter. Bitenkénti értelmezése:*

B0 = 0, *ha* DIRBF0 1, *ha* DIRBF1	*tartalmazza az olvasandó file directory-ját.*

B1 = 0, *ha a teljes* PRTBUF 1, *ha csak a fél* PRTBUF	*tartalmazza a file szektorcimeit. /Lsd: még* COLECT *input/*

- *HL-ben memória kezdőcim a beolvasott szektorok számára.*

output: - -

modified: A,F,B,C,D,E,H,L

Comments: *A file olvasásra előkészitett, ha a fenti input specifikációk teljesülnek.*

12. WRFILE: Irásra előkészitett file összes szektorának felirása adott memóriacimtől kezdve.

input: - A regiszter, mint input kontrol regiszter. Bitenkénti értelmezése:

B0 = 0, ha DIRBF0 | tartalmazza az irandó file
= 1, ha DIRBF1 | directory-ját.

B1 = 0, ha a teljes PRTBUF | tartalmazza a file
= 1, ha csak a fél PRTBUF | szektorcimeit/Lsd: még COLECT input/

- HL-ben az a memória kezdőcim, amelytől kezdve a file felirása történik.

output: - -

modified: A,F,B,C,D, E, H,L

comments: A file irásra előkészitett, ha a fenti input specifikációk teljesülnek.

13. SECTRD
A diszk tetszőleges szektorának beolvasása.

14. SCWRIT
A diszk tetszőleges szektorának irása.

input: - DE-ben a szektor cime a diszken /side, track,szektor/

D-regiszter: B7 /MSB/ = diszk oldal kiválasztás: 0 side 0
1 side 1

B0....B6 = track cim /0....79/

E-regiszter: szektorcim a track-en belül /1....16/

- HL-ben az a memória kezdőcim, amelyiktől az olvasás, ill. az irás történik.

output: A-ban: - 0, ha a művelet sikeres volt
- hibakód, ha a művelet sikertelen volt.

Hibakódok: A hibakód, vagy az FDC-től, vagy a software-től származik. Az FDC-től származó hibakód biteket a VII-5. ábra mutatja.

/MSB/	B7	B6	B5	B4	B3	B2	B1	B0	/LSB/
A:	0	0	0	RNF	CRC	LOST DATA	0	0	

VII-5. ábra

Ha B2=1, akkor "LOST DATA" hiba keletkezett, azaz a software nem volt képes, elég gyorsan kiszolgálni az FDC-t.

Ha B3 = 1, akkor az FDC CRC hibát észlelt, vagy a szektort azonosító, ID mezőben, vagy az adatmezőben.

Ha B4=1, akkor a keresett szektort az FDC nem tudta lokalizálni.

Software-től származó hibakódok: A software ellenőrzi, hogy pontosan 256 byte olvasása, ill. irása történik-e.
Ha ez nem biztositható, akkor a művelettől és az eltérés előjelétől függően a következő hibakódok keletkeznek.

ERROR 70 READ UNDERRUN, byteszám < 256 olvasásnál
ERROR 71 WRITE UNDERRUN, byteszám < 256 irásnál
ERROR 72 READ OVERRUN byteszám > 256 olvasásnál
ERROR 73 WRITE OVERRUN byteszám > 256 irásnál

Megjegyzendő, hogy a software semmilyen esetben sem ir felül 256 byte-nál többet, a memóriában, ill. nem küld ki 256 byte-nál többet az FDC felé.

modified: A,F,B,D,E

comments:

- A szubrutin az irás, ill. olvasás előtt a keresett szektor lokalizálását, is elvégzi, azaz a szubrutin meghivásakor nincs követelmény, az iró/olvasó fej helyzetére nézve.

- Sikertelen lokalizálási kisérlet után a szubrutin a szektor lokalizálását még négyszer megkisérli, és csak ezután adja vissza a vezérlést A-ban a hibakóddal.

- Ha a keresett track-et sem sikerül lokalizálni, akkor az egyes kisérletek között a szubrutin az iró/olvasó fej helyzetét ujrakalibrálja. /RESTORE művelet/.
  Ez a fejnek a TO fölé vitelét jelenti.

- Ha a keresettt szektort maximum 5 kisérletből sikerül lokalizálni, akkor a szubrutin megkisérli az olvasást, ill. irást, és ha az első kisérlet sikertelen, akkor még maximum 5-ször megkisérli az olvasást. /Irás esetén csak lokalizálási hiba észlelhető a szektor teljes visszaolvasása nélkül, ami azonban **több mint felére** csökkentené a sebességet/. A szubrutin tehát max. 5 lokalizálási, ezután max. 6 olvasási kisérletet végez.

- Meghiváskor a diszknek már névleges ford. számon kell lennie.
- Irásvédő szalaggal irásvédett diszkre kiadott irási parancsot az SCRIT szubrutin fatális hibának vesz, és hibajelzés /ERROR5/ kiséretében a rendszerbe tér vissza.
  Az FDC státuszregiszterének olvasásával /Lsd: II-2./ a felhasználói program a hardware irásvédelmet érzékelheti.

15. RDYCON A szubrutin a kiválasztott drive READY státuszának megvizsgálására szolgál. A szubrutin 1-es tipusu FDC parancsok kiadása /lsd TYPONE/ után hivandó meg abban az esetben, ha nem akarjuk, hogy esetleges NOT READY státusz esetén a program futása hibajelzéssel megszakadjon.

input: - -

output: CY = 0, ha a drive READY
1, egyébként

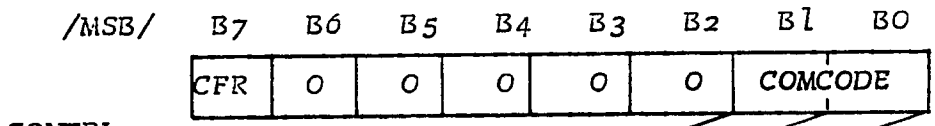
modified: A,F

comments: - Ha az RDYCON szubrutint akarjuk használni, akkor előtte a TYPONE szubrutin hivásakor a CONTRL-byte-ban B7 = 0-t kell specifikálnunk. /Lsd még TYPONE szubrutin/.
Assembly példa egy RESTORE müveletet követő READY vizsgálatra, ha esetleges "NOT READY" státusz esetén a vezérlést a programban kivánjuk tartani:

```
        CALL    SPECCY      : A TYPONE szubrutin
        DB      16 D        : általános hivása
        DB      0           : CONTRL/B7 = 0
        CALL    SPECCY      : Az RDYCON szubrutin
        DB      15 D        : általános hivása
        JP      C,NOT
                READY       : Elágazás, ha "NOT
                              READY"
READY   EQU     $           : Drive READY
```

16. TYPONE A szubrutin az FDC un. 1-es tipusu /fejmozgató/ parancsainak kiadására szolgál. E szubrutin eredményes használatához az FDC ismerete is szükséges. A szubrutin opcionálisan a drive "READY" státuszának vizsgálatára is alkalmas a megfelelő fejmozgató parancs kiadása után.

input: - A szubrutin belépési pontján a stack tetején lévő cim által kijelölt memóriarekesz tartalma, amelyet a továbbiakban CONTRL byte-nak nevezünk. E byte bitenkénti értelmezését mutatja a VII-6 ábra.



CONTRL:

B7:1, ha a parancs kiadása után szükséges drive "READY" vizsgálat

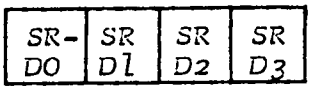
0, ha nem szükséges

B1	B0	A végrehajtott parancs angol neve
0	0	RESTORE
0	1	SEEK WITH VERIFY
1	0	SEEK WITHOUT VERIFY
1	1	STEP IN

VII-6 ábra

input: /folytatás/ - Az STRATE nevü, 209FH kezdőcimü, 4 byte széles rendszerváltozó területen, az egyes drive-okra kijelölt fej-léptetési sebesség /stepping rate/ kódja. E terület felépitését mutatja a VII-7. ábra, a kódolást pedig a VII-6. táblázat.

VII-7.ábra



STRATE	= 209FH

SR-DX: Az X-drive-hoz tartozó léptetési sebesség /X = 0,1,2,3/

SR DX	FEJ LÉPTETÉSI IDŐ /TRACK-RÓL TRACK-RE
0	6 msec
1	12 msec
2	20 msec /2 msec*/
3	30 msec /3 msec* /

VII-6. táblázat

* : WD 1772 esetén

output: - -

modified: A,F

comments: - A szubrutin hivásának javasolt módja következő:

```
CALL SPECCY:  általános      CALL TYPONE: Egyedi
DB    16D                    DB   CONTRL
DB    CONTRL
```

- A szubrutin által kibocsátott FDC parancsok kódjait a VII-7. táblázat mutatja.

A végrehajtott parancs angol neve	Fej léptetési idő /msec/			
	6	12	20 (2)	30 (3)
RESTORE	0CH	0DH	0EH	0FH
SEEK WITH VERIFY	1CH	1DH	1EH	1FH
SEEK WITHOUT VERIFY	18H	19H	1AH	1BH
STEP IN	58H	59H	5AH	5BH

VII-7. táblázat

- Ha az input CONTRL byte-ban "READY" vizsgálatot is előirtunk, akkor "NOT READY" státusz esetén a szubrutin nem tér vissza a hivás helyére, hanem hibajelzéssel a rendszerbe tér vissza. Ha ez nem kivánatos, akkor a "READY" vizsgálatot az RDYCON szubrutinnal végeztessük el.

- A szubrutin a STEP és a STEP OUT nevü fejmozgató. FDC parancsok kiadására nem alkalmas.

17. HEXDC1 1 byte-os hexadecimális szám konvertálása 3 byte-os ASCII-ben ábrázolt decimális számmá.

input: E-ben a konvertálandó szám.

output:- A COUTBF nevü, 205CH kezdőcimü 3 byte hosszu területen az eredmény. A vezető nullák space-ekkel vannak helyettesitve. A COUTBF+2 cimen MSB=1, mint text delimiter.
A legkisebb helyiértékü decimális számjegy COUTBF +2 cimen van.

- HL-ben COUTBF kezdőcime, azaz HL=205CH.

modified: A,F,D,E,H,L

comments: A szubrutin csak pozitiv egész számok konvertálására alkalmas.

18. HEXDC2 2 byte-os hexadecimális szám konvertálása 5 byte-os, ASCII-ben ábrázolt decimális számmá.

input: DE-ben a konvertálandó szám. /D-Hl, E-LO/

output: - A COUTBF nevü, 205CH kezdőcimü 6 byte hosszu területen az eredmény. A vezető nullák space-ekkel vannak helyettesitve. A COUTBF+4 cimen MSB=1 mint text delimiter. A legkisebb helyiértékü decimális számjegy COUTBF+4 cimen van.

- HL-ben COUTBF kezdőcime, azaz HL=205CH.

modified: A,F,D,E,H,L

19. ERASE A képernyő teljes törlése, a képernyő attributumok alapállapotba állitása és a képernyő pointernek a bal felső sarokra állitása.

input: - -

output: Az SCRPOS nevü, 2057H cimü két byte széles rendszerváltozóban 0, azaz 0-ik oszlop, 0-ik sor. E rendszerváltozóról lásd bővebben a DSTRAM szubrutinnál.

modified: - -

20. ERROR /cim:10H/

Az ERROR belépési ponton hibajelzés kisérétében térhetünk vissza DOS-ból. BASIC parancsmódba.

input: A stack tetején levő cim. Ha e cim által kijelölt memóriarekesz tartalma nem nulla, akkor ez az érték a kijelzendő hibakód, ha pedig nulla, akkor a hibakód az. A regiszterben van.

comments:

- Ha a hibakód az FDC-től származik, és az A regiszterben van, akkor hibakód konverzió történik, amelynek eredményeképpen előálló végleges hibakód 1, 2 vagy 3 lehet aszerint, hogy az FDC VII-5. ábrán látható státuszbitjei közül B2, B3, vagy B4 a legmagasabb helyiértékü 1-es bit.

- Ha a hibakód < 17, akkor a hibakód kiirását az ismert szöveges hibajelzés kiséri. Egyéb esetben szöveges hibajelzés nem lehetséges.

- A felhasználói programok járulékos hibajelzéseinek számozására fenntartott hibakód tartomány: 100.... ,199.

- 1. Példa: Hibajelzés szektorolvasási hiba esetén:

```
CALL    SPECCY          :  SECTRD szubrutin
DB      13H                általános hivása
AND     A               :  /Hibakód A-ban/
JP      Z,READ OK       :  Hibavizsgálat, és
                           elugrás, ha nem volt
                           olvasási hiba.
RST     ERROR           :  Hiba esetén visszaté-
                           rés BASIC-be.
DB      0               :  Hibakód az A regiszter-
                           ben.
```

- 2. példa: Hibajelzés felhasználói program olyan részén, ahol a DOS van kiválasztva, 152-es hibakóddal.

```
RST     ERROR     : Visszatérés BASIC-be.
DB      152D      : A kijelzendő hibakód.
```

21. DSTRAM /cim:18H/

A DSTRAM szubrutin szövegeknek a képernyőre, vagy ZX printerre való kiirására szolgál. A kiirandó szövegnek a memóriában kell lennie ASCII karakterek formájában. A szöveg delimitere a legutolsó kiküldendő byte 1-be állitott B7 /MSB/ bitje. Ha a kiirás a képernyőre történik és a szubrutin eléri a képernyő jobb alsó kiirási pozicióját, a kiirás átmenetileg megáll és SCROLL? kérdés jelenik meg. Ez esetben n vagy N beütésére a szubrutin müködése megszakad, y,Y, vagy ENTER leütésére a képernyő beállitható számu sorral feljebb tolódik igy, vagy a felső sorok elvesznek.

input:

- HL-ben a kiirandó szöveg első karakterének cime.

- Az SCRPOS nevü, 2057H cimü két byte széles rendszerváltozóban a kiirás kezdőpoziciója, ha a kiirás a képernyőre történik. E rendszerváltozó felépitését mutatja a VII-8. ábra.

VII-8.ábra

2057H 2058H

COL: Az oszlop száma: 0....31. ROW:A sor száma:0...21.

- A PRINT nevü, 2062H cimü rendszerváltozóban 0, ha a képernyőre, ≠0, ZX printerre történik a kiirás.

- Az NROLLS nevü, 2OE3H cimü rendszerváltozóban az a szám, ahány sorral feljebb kell tolni a képernyőt SCROLL esetén.

output: - SCRPOS-ban a legutolsó kiirt karaktert követő kiirási pozició, ha a kiirás, a képernyőre történt.

- CY =1, ha SCROLL? kérdésre n vagy N a válasz.= a bemeneti érték, egyébként.

modified: F, ha SCROLL? kérdésre n ,vagy N volt a válasz, egyéb esetben minden regiszter módositatlan marad.

comments: - Ha a hivó programban elágazás van aszerint, hogy SCROLL? kérdésre mi a válasz, biztositani kell, hogy a szubrutin hivásakor CY=0 legyen.

- Lsd még: ERASE szubrutin.

22. BASRUT A BASIC ROM-ban levő szubrutinok meghivása a DOS kiválasztott állapotában.

/cim: 28H/

input: - A stack tete jén levő cim a meghivandó BASIC szubrutin ciménck alacsonyabb helyértékü byte-jára mutat.

- Minden egyéb input a meghivandó szubrutinéval szonos.

output: - Azonos a meghivott szubrutin outputjával.

modified: Csak a meghivott szubrutin módositja a regisztereket.

comments: - Hivás módja: RST BASRUT BAS-ADDR a meghivott
DW BAS-ADDR BASIC szubrutin cime.

- A szubrutin nem alkalmazható olyan BASIC szubrutinok hivására, amelyek módositják az FFEOH cimen kezdődő, 26 byte hosszu terület tartalmát.

23. IDENTS Diszk formátumának azonositása. A szubrutin a következő paramétereket azonositja.

/cim: 30H/

- irássürüség /szimpla vagy dupla/
- Track-enkénti szektorszám
- A diszket a SPECCY FORMAT x parancsával formálták-e.

input: - -

output: - A DFLAG1 nevü, 2000H cimü rendszerváltozó B7 /MSB/ bitje 0, ha dupla 1, ha szimpla irássürüségü a diszk.

- A SECMAX nevü, 2056H cimü rendszerváltozóban a Track-enkénti szektorok száma, ami 7, 10, 13 vagy 16 lehet.

- A DOSBUF buffer a TRACK-0 első szektorát tartalmazza.

modified: A,F,B,C,D,E,H,L.

comments: - Ha az irássürüség nem felismerhető, a program futása megszakad és ERROR 15 hibaüzenet keletkezik. Ha a diszket nem a SPECCY-DOS formálta, a program futása megszakad és ERROR 16 hibaüzenet keletkezik. Ha a drive nem "READY", a program futása ERROR 4 hibaüzenettel megszakad.

- Bármely diszkhez való első hozzáfordulást az IDENTS szubrutin meghivásával ajánlatos kezdeni. Ezenkivül a felhasználói program futása során minden olyan programállapot után célszerü a szubrutin meghivása, ahol lehetőség volt az egyszer már azonositott diszk kicserélésére, még akkor is, ha ez bizonyos mértékben lassitja a felhasználói program futását.

24. ROMXCG /cim: 46H/ A ROMXCG belépési pont a DOS kiválasztott állapotából a BASIC kiválasztott állapotába való végleges visszatérést valósitja meg a BASIC RAM tartalmának módositása nélkül.

input: - A stack tetején az a BASIC ROM, ill. RAM cim ahová a vezérlés kerül.

modified: A,F,H,L Ezek a regiszterek módosulnak, mielőtt a vezérlés az input-ban megadott cimre kerül.

comments: Ez a belépési pont használható, ha a visszaváltás a BASIC RAM-ban futó programmal bármely okból nem valósitható meg.

VIII. MINTAPROGRAM: SKILL

E fejezetben egy gyakorlati példán mutatjuk be, hogyan használhatjuk a SPECCY-DOS szubrutinkészletét felhasználói programokban.

A program készitésénél fő célunk, a SPECCY szubrutinok használatának bemutatása volt, de arra is törekedtünk, hogy a program használható is legyen. A program, amelynek teljes forrás és tárgykódu listáját közöljük, gyors "nagytakaritást" tesz lehetővé diszkjeinken azaz igen gyorsan lehet vele file-okat törölni, sőt, az összes nem irásvédett file automatikus törlésére is alkalmas. A másik lehetőség az, hogy a program sorra kiirja az összes file-nevét, és dönthetünk, az adott file-t töröltessük-e vagy nem.
A programban, amely a SKILL /=Speccy KILLer/ nevet viseli, a következő SPECCY-DOS szubrutinokat alkalmaztuk. DELETE, ERASE, SECTRD, ERROR DSTRAM, BASRUT, IDENTS. A változó cimü szubrutinokat egyedileghivjuk. /Lsd: V./ A program kezdőcime 61440 /=FØØØH /, hossza 512 byte, startcim 61680 /=FOFOH/

```
****   SKILL   ****
ERROR          EQU    10 H      Az állandó cimü
                                 SPECCY-DOS szubru-
DSTRAM         EQU    18 H      tinok cimei.
IDENTS         EQU    30 H
BASRUT         EQU    28 H
KEYBRD         EQU    02BFH     SPECTRUM consol input
                                 szubrutin
CTPORT         EQU    3B41H     SPECCY I/O kontrol cime
DFLAG1         EQU    2000H     A DOS FLAG-regiszterének
                                 cime.
DOSBUF         EQU    20CCH     256 byte hosszu DOS
                                 RAM buffer
```

A program által használt saját RAM terület

A program által használt saját RAM terület

QYNEED	EQU	2320H	M regiszter:tartalma = 00, ha az összes file automatikusan törlendő, 01, ha file-onként választási lehetőség van
FNAMBF	EQU	2321H	10x9 = 90 byte hosszu buffer, amelyben egy directory szektorban található fileneveket tároljuk. Első 8 byte: filenév; 9. byte: attr.byte. Ez 10-szer ismételve.

```
;**** SKILL ****                    ;MEGJ: A PROGRAM IRÁSÁNÁL A
                                    ;KÓD TÖMÖRSÉGE NEM VOLT CÉL.
                ORG     ØFØØØH      ; =61440D
ERASE:          PUSH    HL          ;A ERASE szubrutin
                PUSH    AF          ;egyedi hívása.
                LD      HL,13AFH    ;(Lsd. V. fejezet)
                LD      A,(1FEDH)
COMWAY:         CALL    COMCAL
                POP     AF
                EX      (SP),HL
                RET
SECTRD:         PUSH    HL          ;A SECTRD szubrutin
                PUSH    AF          ;egyedi hívása.
                LD      HL,Ø1A4H
                LD      A,(1FF3H)
                JR      COMWAY
DELETE:         PUSH    HL          ;A DELETE szubrutin
                PUSH    AF          ;egyedi hívása.
                LD      HL,1ØGFH
                LD      A,(1FFDH)
                JR      COMWAY
COMCAL:         BIT     7,A         ;COMCAL szubrutin.
                PUSH    DE          ;(Lsd V. fejezet)
                PUSH    AF
                AND     7FH
                LD      E,A
                LD      D,Ø
                SBC     HL,DE
                POP     AF
                JR      NZ,ADDRDN
                ADD     HL,DE
                ADD     HL,DE
ADDRDN:         POP     DE
                RET

DRVSEL:         AND     3           ;*** Drive Select Szubrutin ***
                LD      C,A         ;input: BØ,B1/A: drive kód
                LD      A,(DFLAG1)
                AND     ØFCH        ;output: ———
                OR      C
                LD      (DFLAG1),A  ;modified: A,F,C
                LD      (CTPORT),A
                RET

                                    ;*** Szövegek ***
LOGOTX:         DB      ØDH,ØDH,ØDH,' **** SPECCY'
                DB      '-KILLER V 1.Ø ****'
                DB      ØDH,ØDH,ØDH,'DRIVE? (Ø,1,2,3)'
                DB      ØBAH        ;(: OR 8ØH)

QERYTX:         DB      2ØH,ØDH,ØDH,'QUERY? (Y/N)'
                DB      ØBAH        ;(: OR 8ØH)
```

```
;**** SKILL ****

LOADTX:  DB   20H,00H,00H,'LOAD DISK, THEN TYPE ENTE'
         DB   00D2H  ; (R OR 80H)
WPROTX:  DB   '; WRITE PROTECTED'
CRLFTX:  DB   8DH
AUTOTX:  DB   '; AUTOBOOT FILE', 8DH
DELTX:   DB   '; DELETED', 8DH
         DB   0,0,0,0,0,0,0,0,0,0,0,0  ;Kihasználatlan terület

DELAY:   PUSH HL          ; A program működését mester-
         LD   HL,3FFFH    ; ségesen lassítani kell, hogy
DELOOP:  CALL DELIN       ; használható legyen.
         DEC  HL          ; Erre szolgál ez a DELAY szub-
         LD   A,H         ; rutin.
         OR   L           ; input: ---
         JR   NZ,DELOOP   ;
         POP  HL          ; output: ---
DELIN:   EX   (SP),HL     ;
         EX   (SP),HL     ; modified: A,F
         RET              ;

KILLER   EQU  $           ;**** A program kezdete ****

         DI               ; Átváltás BASIC-ről
         LD   A,8         ; SPECCY-DOS-ra.
         LD   (CTPORT),A  ;
         LD   (DFLAG1),A  ;
         CALL ERASE       ; Képernyő törlése
         LD   HL,LOGOTX   ; Bejelentkező szöveg
         RST  DSTRAM      ; kiírása
DRVCIN:  RST  BASRUT      ; Leütött karakter bevétele
         DW   KEYBRD      ; A drive kód nak 0,1,2,3.
         SUB  30H         ; lehet. Egyéb karakkreket
         CP   4           ; nem fogadunk el.
         JR   NC,DRVCIN   ;
         PUSH AF          ;
         CALL DRVSEL      ; Drive kiválasztása
         POP  AF          ;
         ADD  30H         ; A drive kód echózása,
         LD   (QERYTX),A  ; és a következő
         LD   HL,QERYTX   ; szöveg kiíratása.
         RST  DSTRAM      ;
QERYIN:  RST  BASRUT      ; Leütött karakter bevétele
         DW   KEYBRD      ;
         LD   HL,QYNEED   ;
         AND  0DFH        ; kisbetűből nagy lesz.
```

; **** SKILL ****

```
            CP      'Y'          ; <QYNEED> ← 1, ha Y vagy y,
            LD      (HL),1       ; <QYNEED> ← Ø, ha N vagy n.
            JR      Z,FLBYFL     ; Más karaktert nem faga-
            CP      'N'          ; dunk el.
            JR      NZ,QERYIN    ;
            DEC     (HL)         ;
FLBYFL:     LD      (LOADTX),A   ; A következő szöveg kiírása
            LD      HL,LOADTX    ; és az Y vagy N eckózása
            RST     DSTRAM       ;
WFSTRT:     RST     BASRUT       ; Leütött karakter kevétek
            DW      KEYBRD       ;
            CP      ØDH          ; Várakozás amíg ENTER-t
            JR      NZ,WFSTRT    ; nem ütnek.
            RST     IDENTS       ; A diszk azonosítása
            LD      A,(DOSBUF+13) ; Diszk attributumbyte olvasása.
            BIT     3,A          ; Visszatérés BASIC-ke
            JR      Z,NOWPDK     ; ERROR 30 hibajelzéssel,
            RST     ERROR        ; ha írásvédett diszk. (Soft).
            DB      3Ø           ;
NOWPDK:     LD      DE,2         ; T ← Ø, S ← 2 (Első fil.dir.sec).
                                 ;
OUTERL:     PUSH    DE           ; *** Külső ciklus ***
            RST     IDENTS       ; Diszk azonosítás, mert a ciklusban
                                 ; kivehetik a diszket.
            POP     DE           ; DE ← (T,S)
            PUSH    DE           ; (T,S) elmentése stack-ke.
            LD      HL,DOSBUF    ; (T,S) kiolvasása DOSBUF-
            CALL    SECTRD       ; bufferbe.
            AND     A            ; Visszatérés BASIC-ke hiba-
            JR      Z,DSECX      ; jelzéssel olvasási hiba
            RST     ERROR        ; esetén
            DB      Ø            ;
DSECX:      LD      DE,FNAMBF    ; Dest. pointer ← FNAMBF
            PUSH    DE           ; pointer elmentése későbbi célra
            LD      BC,ØAØØH     ; B = directory számláló ← 1Ø
                                 ; C = létező file számláló ← Ø
INTER1:     PUSH    BC           ; *** Belső ciklus-1 ***
            INC     (HL)         ; Ez a ciklus kigyűjti a létező
            JR      NZ,LIVES     ; file-okat az FNAMBF-ke.
            LD      BC,25        ; Nem létező file: Src. pointer a
            ADD     HL,BC        ; következő file directory-ra.
            POP     BC           ; számlálók előhozása.
            JR      COMPAT       ; Ugrás a ciklus végére.
LIVES:      INC     HL           ; Src. pointer a filenév elejére
            LD      BC,8         ; A filenév, majd az attri-
            PUSH    BC           ; butumbyk átmásolása az
            LDIR                 ; FNAMBF-ke, közben az Src.
            POP     BC           ; és a Dest. pointerek
            ADD     HL,BC        ; módosítása.
            LDI
```

s68

```
;**** SKILL ****

                ADD     HL,BC
                POP     BC              ; számláló előhozása
                INC     C               ; létező file számláló ← +1
COMPAT:         DJNZ    INTER1          ; Belső ciklus vége.

                POP     HL              ; stack szint helyreállítás
                LD      A,C             ; A 2. belső ciklus átugrása,
                AND     A               ; ha a sektorban nem volt
                JP      Z,NXTSEC        ; egyetlen létező file sem
                CALL    ERASE           ; Képernyő törlése.
INTER2:         PUSH    BC              ; *** Belső ciklus-2 ***
                LD      BC,7            ; A belső 2. ciklusban végezzük el
                PUSH    HL              ; a file-ok törlését.
                ADD     HL,BC           ; A soronkövetkező filenév
                SET     7,(HL)          ; utolsó karaktere MSB-jének 1-be
                EX      (SP),HL         ; állítása.
                RST     DSTRAM          ; A filenév kiírása
                EX      (SP),HL
                RES     7,(HL)          ; Az eredeti filenév visszaállítása.
                INC     HL              ; Pointer az attr. byte-ra.
                PUSH    HL
                LD      A,(QYNEED)      ; Automatikus törlés, vagy file-onkénti
                AND     A               ; választási lehetőség?
                JR      Z,PREKIL        ; Ugrás, ha automatikus.
QUERY:          RST     BASRUT          ; Várakozás karakter kerítésére.
                DW      KEYBRD          ; (A=FF, amíg nem ütöttek karaktert)
                INC     A
                JR      Z,QUERY
                AND     0DFH            ; kisbetű → nagybetű
                CP      'Y'+1           ; Ugrás előre, ha Y vagy y
                JR      Z,PREKIL
                CP      'Q'+1           ; Ugrás előre, ha Q vagy q
                JR      Z,PREXIT
                POP     HL              ; Egyéb karakter esetén a törlést
                LD      DE,CRLFTX       ; átugorjuk.
                JR      DISPDE
PREXIT:         POP     HL              ; Stack szint helyreállítása a
                POP     HL              ; BASIC-be való visszakeréséhez.
                POP     HL
                POP     HL
EXIT:           XOR     A               ; Visszaváltás SPECCY-DOS-ból
                LD      (DFLAG1),A      ; BASIC-be.
                LD      (CTPORT),A
                EI
                RET                     ; A program vége. ******

PREKIL:         POP     HL              ; attr. byte pointer előhozása
                BIT     0,(HL)          ; Írásvédett file?
                LD      DE,WPROTX       ; A megfelelő szöveg kardjaire.
                JR      NZ,DISPDE       ; Ugrás előre, ha írásvédett file.
```

```
;**** SKILL ****

          BIT   7,(HL)      ;Autoboot file-t sem szabad törölni.
          LD    DE,AUTOTX   ;A törlés átugrása és a megkelő
          JR    NZ,DISPDE   ;szöveg kezdőcímének betöltése DE-be.
          EX    (SP),HL     ;A file törölhető. HL a névre.
          CALL  DELETE      ;A file törlése a diszkről.
          EX    (SP),HL     ;Attr. byte címe HL-be.
          LD    DE,DELTX    ;"DELETED" szöveg kezdőcíme DE-be.
DISPDE:   EX    (SP),HL     ;Az aktuális szöveg kiírása
          EX    DE,HL       ;a filenév után.
          RST   DSTRAM      ;
          CALL  DELAY       ;Meskrséges programlassítás.
          POP   HL          ;HL pointer a következő filenév
          INC   HL          ;első karakterére.
          POP   BC          ;A számláló (C) előhozása
          DEC   C           ;számlálás, és ugrás a belső 2.
          JP    NZ,INTER2   ;ciklus elejére, ha van még file.
NXTSEC:   POP   DE          ;(T,S) előhozása.
          INC   E           ;S ← S+1 ; a követk. dir. sector.
          LD    A,7         ;Ugrás vissza a belső ciklus
          CP    E           ;elejére, ha van még dir. sator.
          JP    C,EXIT      ;Vége, ha nincs több dir. sector.
          JP    OUTERL      ;
;**********************************
```

A program diszkre mentése: SAVE * "SKILL" CODE 61440,512,61680

A program használata: A SKILL programot RUN * "SKILL" CODE-dal indítkatjuk, illetve PRINT USR 61680-al, ha már a RAM-ban van. Először a DRIVE? kérdésre kell válaszolnunk 0,1,2 vagy 3 leütésével, majd a QUERY? kérdésre Y-nal vagy y-nal, ha file-onként döntünk a törlésről, N ill. n-el, ha minden nem védett file törlendő. A törlendő diszk behelyezése után ENTER-t ütünk. Ezután sorra megjelennek a filenevek, és ha a QUERY? -re Y-al válaszoltunk, akkor három lehetőségünk van, úm. (i) Y-t ill. y-t ütünk, ha az adott file törlendő, (ii) Q-t vagy q-t ütünk, ha ki akarunk lépni a programból (Quit), illetve (iii) bármely más karakter ütésére a program folytatódik a következő filenév kiírásával.

A "SKILL" PROGRAM KÓDJA