

## MDOS3 – popis služeb pro programátory

---

### **Využití:**

Pokud potřebujete napsat nějaký program, který spolupracuje přímo s harddiskem, nemusíte psát svoje vlastní rutiny, ale můžete využít rutin MDOS3. Je to hlavně výhodné v tom, že se nemusíte starat o geometrii disku, pracujete s LBA adresací sektorů a můžete využít některé rutiny, které jsou již hotové.

### **Volání služeb:**

Volání služby se provádí po nastránkování do paměti divIDE. Vzhledem k tomu, že se MDOS3 chová jako D80, můžeme k tomu využít již hotové rutinky pro přestránkování:

```
SHADE      ld      a, $4F
           ld      de, TAB-26
           call   $25AB
           ld      hl, 0
           ld      (TAB), hl
           ld      hl, $3EF7
           ld      (TAB+2), hl
           rst    0
           ret
TAB         dw      0
           dw      $3EF7
```

A pro přestránkování zpět do ROM ZXS:

```
ZXROM      ld      a, 32
           ld      (16119), a
           jp      $1700
```

Všechny služby se volají instrukcí `call $24` (36 dec), číslo služby je v registru **A** a parametry služby v dalších registrech. Žádná ze služeb nemění registr **IY**.

Příklad:

```
ld      a, ($3E6B)
ld      e, a
ld      a, 4
call   $24
```

Čtení dat z virtuální diskety probíhá naprosto stejně, jako čtení z fyzické mechaniky. Změněny jsou až rutiny DREAD a DWRITE, které programátor pouze volá, níž než tyto rutiny nejde, neb je to zbytečné. Stejně tak ošetření chyb je stejné jako u fyzické mechaniky. Prostě virtuální disketa se značí co nejméněji emulovat opravdovou disketu.

Pro úspěšné použití všech služeb je zapotřebí, aby uživatel po nabootevání MDOS3 poprvé stisknul NMI. Pak dojde k načtení informací o oddílech všech (tj. dvou) připojených discích. Vzhledem k tomu, že po nabootevání není nastavena žádná disketa, dá se předpokládat, že první kroky povedou právě do NMI menu. Pro všechny případy služba GET\_VER vrací status inicializace MDOS3.

### **Popis jednotlivých služeb:**

#### **0 GET\_VER**

out: L = den, H = měsíc, DE = rok, A = init MDOS3

mění: BC

popis: Vstupní parametry nejsou žádné, vrací v registrech HL,DE verzi MDOS3. Verze nejsou číslovány, na číslování se používá datum. Čím vyšší je verze, tím vyšší je „čtyřregistr“ DEHL. Pokud je A = 0 (a flag z nastaven) není provedena inicializace MDOS3 a není možné správně využívat služby označené hvězdičkou a tabulky budou prázdné. Inicializace znamená, že uživatel MUSÍ nejprve stisknout NMI menu.

### 1 GET\_MAPTAB

out: HL = adresa tabulky MAPTAB

popis: Vstupní parametry nejsou žádné, v HL bude adresa tabulky MAPTAB.

### 2 GET\_GEOM

out: HL = adresa tabulky GEOM

popis: Vstupní parametry nejsou žádné, v HL bude adresa tabulky GEOM (geometrie disků).

### 3 GET\_AKTPAR

out: HL = adresa tabulky AKTPAR

popis: Vstupní parametry nejsou žádné, v HL bude adresa tabulky AKTPAR. AKTPAR je tabulka, kde jsou údaje o nastavené aktuální partition (v NMI klávesa W).

### 4 GET\_PART\_TABLE

out: HL = adresa tabulky PART\_TABLE

popis: Vstupní parametry nejsou žádné, v HL bude adresa tabulky PART\_TABLE. Tato tabulka obsahuje údaje o zjištěných oddílech na připojených discích.

### 5\* DRIVEPART

in: E = číslo mechaniky (0-3)

out: IX = adresa údajů o partition, která přísluší disketě v dané mechanice, A = číslo oddílu, flag Z = chyba (v mechanice není disketa)

mění: BC,HL,DE

popis: Vstupní parametr je číslo mechaniky s virtuální disketou (nemusí být mapováno na HDD), v IX pak bude adresa údajů o tom oddíle, do kterého disketa patří. Struktura těchto údajů je shodná s se strukturou tabulky AKTPAR. V registru A pak navíc bude číslo oddílu. Pokud nebude v mechanice žádná disketa, bude nastaven flag Zero.

### 6 PREPOCET

in: IX = adresa systémových proměnných mechaniky (např. \$3E00)

out: IX = adresa do MAPTAB pro příslušnou mechaniku

popis: Změní adresu systémových proměnných za adresu MAPTAB, například, pokud je potřeba zjistit, kam je příslušná mechanika přemapována. Není ošetřen proti vstupu nesmyslných hodnot.

### 7 CMMAPTAB

in: E = číslo mechaniky (0-3)

out: IX = adresa do MAPTAB pro příslušnou mechaniku

popis: Je to jiný způsob, jak vypočítat MAPTAB pro danou mechaniku. Není ošetřen proti vstupu nesmyslných hodnot.

### 8 DRVCMPS

in: E = číslo fyzické mechaniky (fd0, fd1)

out: IX = adresa parametrů drivu fd0 nebo fd1

mění: BC

popis: Pro fyzický reset mechanik, jsou parametry fyzických mechanik uloženy úplně jinde. Po změně mapování jsou parametry brány odtud. Tyto parametry se nastavují při výběru mapování na fd0 nebo fd1. Do té doby nejsou použitelné.

### 9 DIVIDEO

in: DEHL = dělenec, C = dělitel

out: DE = výsledek, A = zbytek

mění: B

popis: 32-bitové dělení. DE = vyšší část „čtyřregistru“, HL = nižší část „čtyřregistru“

### 10 INC32

in: BCDE = 32-bit číslo

out: BCDE = 32-bit číslo, flag Z = BCDE=0

popis: zvýší „čtyřregistru“ o 1. BCDE je většinou LBA adresa sektoru.

### 11 **DEC32**

in: BCDE = 32-bit číslo

out: BCDE = 32-bit číslo, flag Z = BCDE=0, flag NC = BCDE=-1

popis: sníží „čtyřregistru“ o 1. BCDE je většinou LBA adresa sektoru.

### 12 **ADD32HL**

in: BCDE = 32-bit číslo, HL = 16-bit číslo

out: BCDE = 32-bit číslo, flag C = přetečení

popis: k „čtyřregistru“ BCDE přičte dvojregistr HL.

### 13 **DEC32HL**

in: BCDE = 32-bit číslo, HL = 16-bit číslo

out: BCDE = 32-bit číslo, flag C = podtečení (záporná hodnota)

popis: od „čtyřregistru“ BCDE odečte dvojregistr HL.

### 14 **ADD1693**

in: BCDE = 32-bit číslo

out: BCDE = 32-bit číslo, nenastavuje flagy

popis: k „čtyřregistru“ BCDE přičte 1693

### 15 **DEC1693**

in: BCDE = 32-bit číslo

out: BCDE = 32-bit číslo, nenastavuje flagy

popis: od „čtyřregistru“ BCDE odečte 1693

### 16\* **NEXTDISK**

in: BCDE = 32-bit číslo coby ukazatel na virtuální disketu, IX = ukazatel na tabulku oddílu příslušný k disketě.

out: BCDE = 32-bit číslo coby ukazatel na další virtuální disketu. Flag C = mimo oddíl

mění: HL

popis: Nastaví do BCDE další disketu, pokud by byl překročen oddíl, je nastaven flag C. IX lze zjistit službou 5. Služba netestuje, zda se na tomto sektoru skutečně nachází platná virtuální disketa.

### 17\* **PREVDISK**

in: BCDE = 32-bit číslo coby ukazatel na virtuální disketu, IX = ukazatel na tabulku oddílu příslušný k disketě.

out: BCDE = 32-bit číslo coby ukazatel na předchozí virtuální disketu. Flag NC = mimo oddíl

mění: HL

popis: Nastaví do BCDE předchozí disketu, pokud by byl překročen oddíl, je nastaven flag NC. IX lze zjistit službou 5. Služba netestuje, zda se na tomto sektoru skutečně nachází platná virtuální disketa.

### 18 **NASDRNMI**

mění: HL, BC, DE, IX

popis: Služba nastaví systém do správného stavu. Pokud aplikace provedla změny v mapování v tabulce MAPTAB, je třeba zavolat tuto službu, aby se provedly změny v parametrech mechanik a taky provede návrat hlav fyzických mechanik na začátek disků a taktéž případný HW reset fyzických mechanik při prvním výběru fd0 nebo fd1.

### 19 **BACKSCR**

popis: Služba provede zálohu VRAM1 do paměti divIDE. Možno obnovit službou 20.

### 20 **RESTSCR**

popis: Služba provede obnovu VRAM1 z paměti divIDE uloženou službou 19.

### 21 **SET\_BOOTDISK**

in: E = číslo mechaniky (0-3)

mění: HL, BC, DE, IX

popis: Nastaví do mechaniky disketu, ze které byl systém nabootován. Provede změnu do tabulky MAPTAB, avšak nenastavuje správné mapování na HDD. Původní disketa je zapamatována.

### **22 RES\_BOOTDISK**

in: E = číslo mechaniky (0-3)

mění: HL, BC, DE, IX

popis: Nastaví zpět to, co provedla služba 21. Pokud by nebyla služba 21 zavolána, nastaví nesmyslné hodnoty do tabulky MAPTAB.

### **23 CLSMAPTA**

mění: HL, BC, DE, IX

popis: Vynuluje tabulku MAPTAB, AKTPAR tak, jak je nastaveno po nabootování systému.

### **24 DRVSELS**

in: E = číslo fyzické mechaniky (0-1)

popis: Roztočí danou fyzickou mechaniku (fd0, fd1)

### **25\* READHDD**

in: BCDE = LBA sektor, HL = data, 4.bit registru B nastavuje zařízení master/slave

out: BC = číslo chyby

mění: HL, BC, DE, IX

popis: Načte sektor. Je automaticky přepočítán na adresu CHS podle geometrie disku. Master/slave je nastaven pomocí 4.bitu registru B. Číslo chyby je stejné jako při výstupu z rutiny DREAD. Pro čtení dat doporučuji spíše nastavit do mechaniky nastavit požadovanou disketu a použít rutiny pro čtení sektoru z diskety (DREAD, BREAD). Pokud je operace úspěšná, pak HL=HL+512.

### **26\* WRITEHDD**

in: BCDE = LBA sektor, HL = data, 4.bit registru B nastavuje zařízení master/slave

out: BC = číslo chyby

mění: HL, BC, DE, IX

popis: Zapiše sektor. Je automaticky přepočítán na adresu CHS podle geometrie disku. Master/slave je nastaven pomocí 4.bitu registru B. Číslo chyby je stejné jako při výstupu z rutiny DWRITE. Služba nekontroluje WRITE PROTECT mechaniky. Pro zápis dat doporučuji spíše nastavit do mechaniky nastavit požadovanou disketu a použít rutiny pro zápis sektoru na disketu (DWRITE, BWRITE). Pokud je operace úspěšná, pak HL=HL+512.

### **27 TST\_BSY**

in: 4.bit registru B nastavuje zařízení master/slave

out: A = 0 + flag Z: READY, A = 255 + flag NZ: BUSY

popis: testuje nebo čeká (cca 2 sec.) na to, až ve status registru bude shozený bit BUSY. Pak je možno posílat do registru parametry a příkazy. Všechny příkazy pro čtení a zápis z/na HDD tuto službu již používají a není nutno ji používat před nimi.

### **28 READIDATA**

in: HL = data, 4.bit registru B = master/slave

out: BC = číslo chyby

mění: HL, BC, DE

popis: Načte ID sektor z ATA zařízení na adresu v HL. Tedy posílá command ECh. Výstup chyb je stejný jako u služby 24 (stejný jako u interních rutin pro ovládání mechaniky DREAD).

### **29 READIDATAPI**

in: HL = data, 4.bit registru B = master/slave

out: BC = číslo chyby

mění: HL, BC, DE

popis: Načte ID sektor z ATAPI zařízení na adresu v HL. Tedy posílá command A1h. Výstup chyb je stejný jako u služby 24 (stejný jako u interních rutin pro ovládání mechaniky DREAD).

## **Tabulky:**

## MDOS3 – popis služeb pro programátory

V textu se vyskytuje spousta tabulek. Zde si vysvětlíme struktury jednotlivých tabulek. Zcela nejzákladnější tabulka je MAPTAB, na které stojí základy celého MDOS3.

<b>MAPTAB</b>	db	2,0,0,0,0	;drive A:
	db	3,0,0,0,0	;drive B:
	db	4,0,0,0,0	;drive C:
	db	5,0,0,0,0	;drive D:
	db	3	;emulace TAPE
	db	0	;Rewrite mód

Tabulka mapování mechanik. Definuje, zda je mechanika mapována na fd0, fd1 nebo HDD. Pokud je mapováno na HDD, pak obsahuje i číslo LBA sektoru, kde se nachází disketa.

Pro každou mechaniku je definováno 5 bytů, pro emulátor kazetofonu 1 byte.

offset 0: mapování. Číslo 0 = fd0, 1 = fd1, 2-6 HD, přičemž pro mechaniku A je 2, pro B je 3, pro C je 4, pro D je 5 a pro TAPE emulátor 6. Pokud se neemuluje, je zde nula.

offset 1-4: virtuální disketa. Zde je uloženo číslo LBA sektoru, kde se nachází virtuální disketa. Tento sektor ukazuje na infosektor. Číslo je uloženo od nižšího k vyššímu bajtu. V posledním (4) bajtu, může být 4.bit nastaven na 0 v případě zařízení MASTER, a 1 v případě zařízení SLAVE.

Emulace TAPE:

Rewrite mód: 0 = Když emulace SAVE na kazetofon zjistí stejný soubor, jaký je už uložen, vypíše standardní dotaz „Rewrite old file“.

1 = Umožní uložit na disk více souborů se stejným názvem.

2 = Přepíše starý soubor bez dotazu.

<b>GEOM</b>	db	\$D2,\$03,\$08,\$20	;master
	db	\$EA,\$01,\$02,\$20	;slave

Geometrie zařízení MASTER a SLAVE.

Pro každé zařízení (MASTER, SLAVE) jsou definovány 4 byty. Jsou v něm zapsané geometrie jednotlivých zařízení (HDD, CF).

offset 0-1: Cylinders

offset 2: Heads

offset 3: Sectors

<b>AKTPAR</b>	db	1	;číslo partition
	db	\$20,0,0,0	;začátek partition
	db	\$E0,D1,3,0	;délka partition

V této tabulce se nachází informace o právě nastavené aktuální partition v NMI menu (klávesa W)

Číslo partition může nabývat těchto hodnot:

offset 0: 1-4: primární oddíl na zařízení MASTER

5-8: primární oddíl na zařízení SLAVE

254: celý disk MASTER

255: celý disk SLAVE

Poznámka: Protože lze vybrat v NMI menu i celý disk (třeba hda.) je to označeno takto. Začátek oddílu je v tomto případě nastaven na místo, kde byla nalezena první virtuální disketa a délka oddílu je pak přepočítána až na konec celého disku.

offset 1-4: Číslo partition: je to LBA sektor, kde začíná oddíl.

offset 5-9: Délka partition: je to LBA adresa délky oddílu. Poslední LBA sektor lze spočítat: začátek+délka-1.

<b>PART_TABLE</b>	db	1	;číslo partition
	db	\$20,0,0,0	;začátek partition
	db	\$E0,D1,3,0	;délka partition

## MDOS3 – popis služeb pro programátory

---

db	6	; číslo partition
db	\$80,\$3E,0,0	; začátek partition
db	0,\$3C,0,0	; délka partition
db	0	; ukončovací znak

V této tabulce jsou údaje o nalezených oddílech při inicializaci (první vstup do NMI menu). Jsou to pouze oddíly určené pro MDOS3 a nebo pak celý disk (pokud na něm není žádný MDOS3 oddíl). Pro každý oddíl je vyhrazeno 9 bytů. Těchto 9 bytů má shodnou strukturu jako tabulka AKTPAR:

offset 0: 1-4: primární oddíl na zařízení MASTER  
5-8: primární oddíl na zařízení SLAVE  
255: celý disk SLAVE, první disketa nalezena, údaje přepočítány  
254: celý disk MASTER, první disketa nalezena, údaje přepočítány  
253: celý disk SLAVE, první disketa zatím nebyla hledána  
252: celý disk MASTER, první disketa zatím nebyla hledána

Celou tabulku uzavírá nula.

Poznámka: Protože lze vybrat v NMI menu i celý disk (třeba hda.) je to označeno takto. U výběru celého disku je to navíc rozděleno na 2 stavy:

- 1) Byl proveden výběr tohoto disku, NMI menu tudíž provedlo nalezení první diskety a údaje přepočítalo.
- 2) Nebyl proveden výběr tohoto disku a tudíž se zatím neví, kde začíná první disketa.

Jestliže ještě nebyla nalezena první disketa, začátek oddílu je nastaven na sektor číslo 0 a délka na délku celého disku v sektorech. Jestliže byla již nalezena první disketa, je začátek oddílu nastaven na tuto disketu a délka oddílu je přepočítána tak, aby dosáhla přesně na poslední sektor na disku.