

# BIOS

**Modul pre diskové  
operácie v strojovom  
kóde pre D40/D80/Kompakt  
s OS MDOS**

**MIROSLAV**   
B E N Í Č E K

*Príručka pre používateľa*



Release: 27.3.94

# **BIOS**

verzia 1.1

---

## Basic Input Output System

MODUL PRE DISKOVÉ OPERÁCIE POD MDOS

(C) 1994 Miroslav Beníček

---

### **DÔLEŽITÉ UPOZORNENIE**

---

Prekontrolujte si najprv, či je disketa chránená pred nežiadúcim zápisom (u 5, 25" diskety zalepený otvor, u 3, 5" otvorené okienko). Táto disketa a software na nej je stavaný tak, že nebude nikdy požadovať zápis.

Nedoporučujeme odstránenie zábrany voči zápisu, nesprávnou manipuláciou by ste mohli prísť o záznam na diskete!

Pred vložením diskety do mechaniky si tiež prečítajte túto príručku.

Pred reklamovaním tohto produktu si najprv prečítajte túto dokumentáciu.

SADZBA

romantic softman

## ÚVOD

Základným prínosom disketovej jednotky pre osobný počítač v protiklade s obvyčajným magnetofónom je zrýchlenie operácií s vonkajším pamäťovým zariadením z rádovo minút na rádovo sekundy. Ďalšou nezanedbateľnou výhodou je možnosť priameho programom riadeného prístupu ku informáciám na takomto médiu.

Preto je iste veľkým prínosom vyvinutie a dostatočné rozšírenie disketovej jednotky s operačným systémom **MDOS** firmou *Didaktik a.s.* zo Skalice. Na tento operačný systém sú rôzne názory, ale svoju základnú funkciu predsa len splňa.

Spomenieme jeho hlavné výhody:

- veľká rýchlosť diskových operácií
- vlastná tieňová ROM a RAM bez rozšírenia systémovej oblasti v samotnej RAM počítača
- zabudovaný obvod 8255A a tým možnosť komunikácie s perifériami aj pri obsadenom systémovom konektore
- možnosť pripojenia druhej disketovej jednotky
- pomerne dobre vytvorený interpret príkazov pre diskové operácie
- obohatenie BASICu o možnosť práce so sekvenčnými súbormi

Samozrejme tento systém má svoje nedostatky:

- žiadna podpora obsluhy diskových operácií zo strojového kódu
- nemožnosť používania nadstavieb BASICu ako je Beta Basic a pod.
- nekompatibilita s jednotkami Microdrive
- niekoľko programátorských chýb

- nemožnosť jednoduchého používania niektorých podprogramov z diskovej ROM pre ich "tvrdé" orientovanie na prostredie BASICu

Z uvedených dôvodov je veľmi ťažké používať podprogramy z diskovej ROM a už samotné "preklopenie" ROM prináša so sebou mnoho problémov. Toto mnohých programátorov odrádza od programovania diskových operácií a využívajú často len zlomkové informácie, či ich produkty sa pri kolíznych stavoch nesprávajú úplne podľa ich predstáv.

Autor sa rozhodol vydat knižnicu diskových operácií pre operačný systém MDOS, ktorá v maximálne možnej miere odstraňuje problematické miesta iných riešení, pod názvom BIOS. Je určený pre programátorov pracujúcich s assemblerom Z80. Pri vyvíjaní tohto modulu sa prihliadalo, aby mal čo najkratšiu dĺžku a tým najväčšiu možnú využiteľnosť.

BIOS obsahuje niekoľko služieb pre bežné diskové operácie. Každá služba vždy vráti po svojom úspešnom, alebo neúspešnom vykonaní chybový kód, ktorý už môže programátor jednoducho otestovať vo svojom programe. V prípade chyby, sa rozhodnúť ako ju oznámiť, alebo ošetriť. Formu a tvar ošetrovacej rutiny je plne možnostiach programátora.

U niektorých bežne používaných riešení sa buď použilo jednoduché odsmernenie návratu po chybe zmenou istej systémovej premennej, čo však neodstránilo zobrazenie niektorých ohybových hlásení na obrazovku, alebo sa použila metóda úplného prepisu príslušných partií z diskovej ROM a ich modifikácia. To nie je však prípad BIOSu.

BIOS je postavený s ohľadom na vzájomnú kompatibilitu oboch verzií MDOSu. V rámci BIOS Professional Edition poskytujeme ako komentovaný zdrojový text s popisom a krátkymi

príkladmi. Informujte sa o možnostiach získania zdrojového textu BIOSu.

BIOS je prvý z radu možných produktov tejto série. V budúcnosti uvažujeme o verzií, ktorá bude na systémoch založených na MDOSe pracovať s disketami MS-DOS, ďalej podporujeme vytvorenie verzie pre operačný systém Beta Disk. Ako doplnok vyvinieme BIOS for TAPE.

Z uvedeného vyplýva, že používanie BIOSu Vám prinesie jednoduchý prechod na inú platformu. Preto už pri vývoji svojho softwaru počítajte s možnosťou nahradenia inou verziou BIOSu. Ide hlavne o možnosť neexistencie niektorej zo služieb v inej verzií, o inú dĺžku kódu BIOSu a pod.

Táto príručka popisuje základné funkcie, ktoré Vám, ako programátorovi, poskytuje BIOS for MDOS. V závere uvádzame jednoduché príklady použitia BIOSu.

## AUTORSKÉ PRÁVA

Vlastníkom autorských práv na BIOS a BIOS for MDOS je **Miroslav Beníček**.

Vlastníctvo autorských práv sa vzťahuje na samotný kód BIOSu, štruktúru služieb a použité algoritmy.

Autormi tejto dokumentácie sú Dipl. Ing. **Robert Madaj** a **Roman Kráľovič**. Žiadna časť z tejto príručky nesmie byť publikovaná bez súhlasu autorov.

BIOS for MDOS je možné použiť vo svojom produkte len za podmienky vlastníctva používateľských práv na tento produkt. BIOS for MDOS je zakázané šíriť v žiadnom tvare ako samostatný produkt a ani nijako inak umožňovať jeho

neoprávnené používanie treťou osobou inak ako v rámci produktu s ktorým je distribuovaný.

Pred samotným použitím produktu doporučujeme prečítanie tejto dokumentácie.

## INŠTALÁCIA BIOSu

Pretože v základnom balíku nie je prítomný zdrojový kód BIOSu, je nutné zabezpečiť umiestnenie a jeho funkčnosť na adresu, ktorú používateľ potrebuje.

Preto je na distribučnej diskete uložený program SETUP, ktorý dokáže umiestniť kód BIOSu od ľubovolnej adresy počnúc adresou 24200 až po 64521. Po zadaní zvolenej adresy program SETUP uloží na vloženú disketu samotný kód BIOSu uložený od požadovanej adresy.

**Doporučujeme vám neumožňovať zápis na originálnu disketu, aby nedošlo ku prepísaniu jej záznamu. Na experimenty používajte vždy svoje pracovné diskety.**

Umiestnením BIOSu na ľubovольnú adresu je umožnená práca pod ktorýmkoľvek prekladačom assembleru aj bez znalosti zdrojového kódu samotného BIOSu.

Program SETUP načítate zadaním RUN po vložení originálnej diskety. Po jeho spustení a zadaní požadovanej adresy vložte na požiadanie programu novú disketu (naformátovanú, najlepšie čistou). SETUP na túto disketu uloží samotný kód BIOSu. Tento je už možné bežným povelenom `LOAD*"BIOSXXXXX"CODE XXXXX` uložiť do pamäti a používať vo svojom vývojovom prostredí.

Kód BIOSu je možné krokovať vo Vašom ladiacom systéme. Upozorňujeme na fakt, že použité algoritmy v BIOSe sú vo vlastníctve autora a ich použitie v iných programoch je bez jeho súhlasu zakázané.

## SLUŽBY BIOSU

Volanie služieb BIOSu je mimoriadne jednoduché. Postačí totiž vedieť len počiatočnú adresu BIOSu a tento popis. Pokiaľ máte k dispozícii zdrojový text BIOSu, môžete si ho preložiť na ľubovoľné miesto v pamäti. Samozrejme, využíva oblasť diskovej RAM a preto nemôžete využívať túto oblasť. Umiestnenie BIOSu v RAM si môžete definovať v programe 'setup'.

BIOS poskytuje viacero služieb, ktoré sú očíslované a pri volaní BIOSu sa toto číslo musí uložiť do registra A. Pokiaľ sa pokúsíte volať neexistujúcu službu, BIOS vráti príslušné chybové hlásenie ako číslo v registri A. Číslovanie hlásení odpovedá hodnotám čísiel hlásení, ktoré používa MDOS.

Aby bolo možné BIOS používať, je nutné zadať pred štartom programu, ktorý ho používa, príkaz

### **POKE #247,79**

Tento príkaz zadajte aj pred štartom Vašeho vývojového systému. Pokiaľ tak neurobíte, Váš program zresetuje počítač pri prvom použití niektorej zo služieb BIOSu!

Nasleduje zoznam služieb BIOSu tak ako sú definované v jeho verzií pre operačný systém MDOS.

## **Služba: 0 SAVE**

---

### ***Uloženie časti pamäte do súboru***

Táto služba umožňuje uloženie súboru s rôznou definovanou príponou na disketu so všetkými kontrolami na možné kolízie.

Idie o kontroly pred samotným uložením dát na disketu, napr. kontrola na:

**Drive not ready,  
Disk full,  
Directory full,  
Corrupted FAT structure,  
Bad device type,  
Disk write protect...**

Nekontroluje chyby vznikajúce počas ukladania dát na disketu:

**CRC error,  
Sector not found...**

Pri výskyte chyby sa samozrejme požadovaná služba nevykoná.

Pri vstupe do tejto služby je nutné naplniť registre hl,de,bc parametrami:

**Vstup:**

- HL - adresa identifikácie súboru (vid. identifikácia súboru)

- DE - odkiaľ v pamäti RAM sa majú dáta ukladať do súboru
- BC - koľko bajtov sa má od adresy v reg. DE nahrávať

#### Výstup:

- A=0 a Z=1 - pokiaľ uloženie súboru bolo úspešné
- V prípade chyby (A<>0, Z=0):  
HL=adresa umiestnenia textu chybového hlásenia v RAM (v tele BIOSu).

## Služba: 1 LOAD

---

### Načítanie súboru do pamäte

Načítanie súboru z disku s uvažovaním možných konfliktných situácií a ich hlásenia v podobe návratu čísla chyby v registri A.

Kontroly pred samotným uložením dát na disketu, napr. kontrola na:

**Drive not ready,  
Bad device type,  
File too long...**

Nekontroluje chyby vznikajúce počas ukladania dát na disketu:

**Corrupted FAT structure,  
CRC error,  
Sector not found**

Pri výskyte chyby sa samozrejme požadovaná služba nevykoná.

Výhodou je, použitie parametra v registri **bc**. Takto môžete ľahko obmedziť maximálnu dĺžku načítavaného súboru a tak sa chrániť proti nežiadúcemu prepisu mimo priestor vyhradený Vaším programom pre načítavaný súbor. Ide o chybu File too long.

*Službou 19 HDSET možno prinútiť BIOS, aby uprednostňoval údaje z hlavičky súboru pred zadanými vstupmi (DE, BC).*

#### Vstup:

- HL=adresa identifikácie súboru (vid. identifikácia súboru),
- DE=kam v pamäti RAM sa majú dáta načítavať zo súboru, malo by mať hodnotu viac ako 16384 (počas načítavania je prepnutá ROM/RAM MDOSu),
- BC=umožňuje definovať maximálnu dĺžku súboru, pokiaľ bude načítavaný súbor dlhší vráti chybové hlásenie File too long.

Služba umožňuje nahrávať, pokiaľ je nastavený príznak službou 14 DEFTYP BIOSu, len súbory označené systémovou príponou v hlavičke, doplnenou službou č.0 t.j. SAVE. Vtedy hlási chybu File not found. Pozri službu 19. '

**Výstup:**

- A=0 a Z=1 - pokiaľ načítanie súboru bolo úspešné
- V prípade chyby (A<>0, Z=0):  
HL=adresa umiestnenia textu chybového hlásenia v RAM (v tele BIOSu).

**Služba: 2 ERASE**

---

**Vymazávanie súborov**

Bezkonfliktné vymazanie súboru alebo viacerých súborov s možnosťou mazania súborov aj s vypnutým atribútom D. V identifikácii je možné použiť aj hviezdičkovú a otáznikovú konvenciu.

Pred samotným vymazaním sa kontroluje:

**Drive not ready,  
Bad device type,  
Disk write protect,  
File not found...**

Nekontroluje chyby vznikajúce počas ukladania dát na disketu:

**CRC error,  
Sector not found,  
Corrupted file structure**

**Vstup:**

- HL=adresa identifikácie súboru (viď. identifikácia súboru)

**Výstup:**

- A=0 a Z=1 - pokiaľ definované súbory boli vymazané.
- V prípade chyby (A<>0, Z=0):  
HL=adresa umiestnenia textu chybového hlásenia v RAM (v tele BIOSu).

**Služba: 3 CAT**

---

**Prezeranie adresára diskety**

Načítavanie mien súborov z adresára diskety, ktoré vyhovujú zadanej identifikácii so zohľadnením nastavenia služby 19.

Pracuje sekvenčne, a to tak, že pri volaní služby definujeme vstup a pri návrate dostaneme na výstupe jedno meno súboru, ktorý vyhovuje špecifikácii. Pri každom ďalšom volaní dostaneme ďalšie meno až pokiaľ CAT nenájde už ani jeden súbor vyhovujúci špecifikácii a vracia sa s hlasením File not found (reg. A=27).

Umožňuje vytvoriť si v pamäti zoznam súborov, ktoré nám vyhovujú a tak zobrazovať vo vlastnom programe len tie súbory, ktoré sú pre program naozaj vhodné (aspoň teoreticky). Nie je to teda klasický príkaz CAT z Basicu, ktorý len vypisuje mená



súborov, ale použiteľná a dôležitá služba. Môžete si vytvoriť vo svojom programe jednoduché dialógové okná, ktoré ponúknu používateľovi zoznam mien súborov na disku a on si z nich iba vyberá. Nemusí preto vôbec vytukávať jeho meno a pod. To isté platí aj pre operáciu SAVE.

Vid' príklad programového obsluženía tejto služby.

**Vstup:**

- HL=adresa identifikácie súboru (vid. identifikácia súboru)

**Výstup:**

- A=0 a Z=1 - pokiaľ definovaný súbor bol nájdený,
- V prípade chyby (A<>0, Z=0):  
HL=adresa umiestnenia textu chybového hlásenia v RAM (v tele BIOSu).

## **Služba: 4 LOADIST**

---

### ***Načítanie prvého sektoru zo súboru***

Je podobná ako služba LOAD, načítava vždy len prvý sektor súboru. Je to teda 512 bajtov. Pokiaľ je súbor kratší ako 512 bajtov je zvyšok do 512 vyplnený nejakým bajtom, ktorého hodnota môže byť rôzna.

Môže poslúžiť na rýchlu kontrolu originality obsahu súboru, s ktorým chceme previesť niektorú diskovú operáciu.

**Vstup:**

- HL=adresa identifikácie súboru (vid. identifikácia súboru),
- DE=kam v pamäti RAM sa majú dáta načítavať zo súboru, malo by mať hodnotu viac ako 16384 (počas načítavania je prepnutá ROM/RAM MDOSu),

**Výstup:**

- A=0 a Z=1 - pokiaľ prvý sektor požadovaného súboru bol načítaný.
- V prípade chyby (A<>0, Z=0):  
HL=adresa umiestnenia textu chybového hlásenia v RAM (v tele BIOSu).

## **Služba: 5 RENFIL**

---

### ***Premenovanie súboru***

Premenovanie súboru so zadanou špecifikáciou na novú špecifikáciu.

**Vstup:**

- HL=adresa identifikácie súboru (vid. identifikácia súboru), tj. staré meno súboru
- DE=adresa identifikácie súboru s novým menom, pričom meno disku musí byť v oboch špecifikáciách rovnaké.

**Výstup:**

- A=0 a Z=1 - pokiaľ definovaný súbor bol premenovaný.
- V prípade chyby (A<>0,Z=0):  
HL=adresa umiestnenia textu chybového hlásenia v RAM (v tele BIOSu).

**Služba: 6 RENDSK**

---

**Premenovanie diskety**

Zmena mena disku (ktoré je umiestnené v nultom sektore).

**Vstup:**

- HL=adresa identifikácie disku (A:novémeno)

**Výstup:**

- A=0 a Z=1 - pokiaľ meno diskety bolo zmenené.
- V prípade chyby (A<>0,Z=0):  
HL=adresa umiestnenia textu chybového hlásenia v RAM (v tele BIOSu).

**Služba: 7 HEAD**

---

**Načítanie sektoru**

Načítanie jedného logického sektoru z disku MDOS. Je možné použiť na načítanie sektorov tak, ako idú za sebou bez ohľadu na adresárovú štruktúru a tabuľku FAT.

Pri načítavaní sa použije ten disk, na ktorom bola posledná operácia.

**Vstup:**

- HL=identifikácia disku (A:, B:)
- BC=číslo sektoru diskety, pokiaľ zadáte väčšie ako je skutočný počet sektorov na disku, BIOS hlási chybu.
- DE=adresa v RAM, kam uloží obsah sektoru

**Výstup:**

- A=0 a Z=1 - pokiaľ definovaný sektor bol načítaný.
- V prípade chyby (A<>0,Z=0):  
HL=adresa umiestnenia textu chybového hlásenia v RAM (v tele BIOSu).

**Služba: 8 RESTOR**

---

**Zápis sektoru**

Zápisaniu jedného logického sektoru na disk MDOS. Použije sa ten disk, na ktorom bola posledná disková operácia.

**Vstup:**

- HL=identifikácia disku (A:, B:)
- BC=číslo sektoru diskety, pokiaľ zadáte väčšie ako je skutočný počet sektorov na disku, BIOS hlási chybu
- DE=adresa v RAM, odkiaľ bude zapisovaných 512 bajtov na disk

**Výstup:**

- A=0 a Z=1 - pokiaľ definovaný sektor bol uložený.
- V prípade chyby (A<>0,Z=0):  
HL=adresa umiestnenia textu chybového hlásenia v RAM (V tele BIOSu).

**Služba: 9 VOLUME****Zistenie mena disku**

Zisti meno disku, vloženého do mechaniky, s ktorým sa posledne pracovalo.

**Vstup:**

HL=identifikácia disku (A:, B:)

**Výstup:**

- A=0 a Z=1 - pokiaľ načítanie mena bolo úspešné a potom je platný obsah HL.
- HL=adresa v RAM, kde je 10 znakov mena diskety
- V prípade chyby (A<>0, Z=0):  
HL=adresa umiestnenia textu chybového hlásenia v RAM (v tele BIOSu).

**Služba: 10 FREE****Počet voľných sektorov**

Zistenie počtu voľných sektorov na vloženej diskete.

Kontroly pred volaním služby:

**Drive not ready,  
Bad device type...**

Nekontroluje chyby vznikajúce počas ukladania dát na disketu:

**Corrupted FAT structure,  
CRC error,  
Sector not found**

**Vstup:**

- HL=identifikácia disku (A:, B:)

**Výstup:**

- A=0 a Z=1 - pokiaľ operácia bola úspešná
- HL=adresa, kde je uložený počet voľných sektorov na diskete v dvojбайtovej notácii (prvý bajt nižší, druhý bajt vyšší). Počet voľných sektorov načítate napr. takto:  
LD C,(HL)  
INC HL  
LD B,(HL)

**Služba: 11 MAXSEK****Počet sektorov na diskete**

Zisti počet sektorov na diskete (teda počet stôp krát počet sektorov na stopu).  
Počítajú sa aj sektory adresára a FAT.

**Vstup:**

- HL=identifikácia disku (A:, B:)

**Výstup:**

- A=0 a Z=1 - pokiaľ operácia bola úspešná
- HL=adresa, kde je uložený počet voľných sektorov na diskete v dvojбайtovej notácii (prvý bajt nižší, druhý bajt vyšší). Počet voľných sektorov načítate napr. Takto:  
LD C,(HL)  
INC HL  
LD B,(HL)

**Služba: 12 DFORM****Formát disku**

Vráti počet stôp a počet sektorov na stopu.

**Vstup:**

- HL=identifikácia disku (A:, B:)

**Výstup:**

- A=0 a Z=1 - pokiaľ operácia bola úspešná
- (HL)=počet sektorov na stope  
(HL+1)=počet stôp na disku

## **Služba: 13 DEFTYP2**

---

### ***Nastavenie obsahu 16. a 17. bajtu hlavičky***

Využíva služba SAVE, kedy je hodnota definovaná touto službou ukladaná pri každom použití SAVE. Implicitná hodnota je 32768.

#### **Vstup:**

- BC=požadovaná hodnota

## **Služba: 14 DEFTYP**

---

### ***Definovanie druhej prípony súboru***

Druhá prípona súboru je umiestnená na 19. bajte hlavičky súboru v adresári diskety.

Keď sa pracuje v móde práce s daným typom druhej prípony, bude BIOS zohľadňovať toto nastavenie a pri pokuse o načítanie súboru funkciou LOAD. Pozri INESET.

#### **Vstup:**

- C=hodnota druhej prípony súboru, s ktorou bude BIOS pracovať

## **Služba: 15 RATTR**

---

### ***Nastavenie práce so súbormi HEAD-ONLY***

Atribút READ-ONLY využíva MDOS na súbory, do ktorých nie je možné zapisovať. BIOS umožňuje obchádzať tento atribút, t. z. že R atribút BIOSu má väčšiu prioritu ako zo systému MDOS.

#### **Vstup:**

- C=0 - nezohľadňuje sa pri SAVE READ-ONLY atribút z hlavičky súboru
- C<>0 - pri pokuse o zápis do súboru s READ-ONLY, BIOS vráti chybu.

## **Služba: 16 WATTR**

---

### ***Nastavenie práce sa súbormi WRITE-ONLY***

Atribút WRITE-ONLY využíva MDOS na súbory, ktoré nie je možné načítavať. BIOS umožňuje obchádzať tento atribút, t. z. že W atribút BIOSu má väčšiu prioritu ako zo systému MDOS.

#### **Vstup:**

- C=0 - nezohľadňuje sa pri LOAD WRITE-ONLY atribút z hlavičky súboru
- C<>0 - pri pokuse o načítanie súboru s WRITE-ONLY, BIOS vráti chybu.

## **Služba: 17 DATTR**

---

### ***Nastavenie práce sa súbormi DELETABLE***

Atribút DELETABLE využíva MDOS na súbory, do ktorých nie je možné zapisovať a vymazávať ich. BIOS umožňuje obchádzať tento atribút, tzn. že D atribút BIOSu má väčšiu prioritu ako zo systému MDOS.

#### **Vstup:**

- C=0 - nezohľadňuje sa pri SAVE a ERASE DELETABLE atribút z hlavičky súboru
- C<>0 - pri pokuse o zápis do súboru s DELETABLE, BIOS vráti chybu.

## **Služba: 18 INESET**

---

### ***Práca s inými príponami***

Táto služba povoľuje alebo zakazuje prácu so súbormi, ktoré majú inú príponu ako je definovaná službou DEFTYP.

#### **Vstup:**

- C=0 - zakazuje prácu so súbormi inej prípony
- C<>0 - povoľuje

## **Služba: 19 HDSET**

---

### ***Preberanie údajov z hlavičky súboru***

Pri povolení preberania údajov z hlavičky sa nebudú brať do úvahy vstupy do služieb LOAD, SAVE ale tieto údaje sa zoberú priamo z hlavičky definovaného súboru.

Využijete vtedy, keď je údaj o štartovacej adrese súboru nezmyselný alebo nevhodný na načítanie na Vami zadanú adresu (viď definovanie vstupu do jednotlivých služieb). Pri povolení preberania údajov z hlavičky sa štartovacia adresa zoberie priamo z hlavičky súboru a zadaný vstup sa neberie do úvahy.

#### **Vstup:**

- C=0 - preberanie údajov z hlavičky, vstupy do služieb neberie
- C<>0 - zanedbáva údaje z hlavičky a berie ako platné vstupy

#### **Výstup:**

- A=0

## **Služba: 20 ACTDRV**

---

### ***Posledný disk***

Vrátí meno disku, s ktorým ste posledne pracovali.

#### **Výstup:**

- (HL)='A' alebo 'B'
- A=O (vždy)

## **Služba: 21 MAXDRV**

---

### ***Počet pripojených diskov***

MDOS umožňuje pracovať s jedným alebo dvomi mechanikami. Aby sme zistili ich počet, použijeme túto službu. Táto vracia meno posledného pripojeného disku. Teda, keď máme dve mechaniky vráti 'B'.

Odpovedá teda na otázku: Mám pripojené B: alebo nie.

#### **Výstup:**

- (HL)='A' alebo 'B'
- A=O (vždy)

Počet pripojených diskov si vypočítate takto:

```
LD A,(HL)  
SUB 64
```

## **Služba: 22 VERS**

---

### ***Číslo verzie BIOSu***

#### **Výstup:**

- HL=adresa názvu verzie.

## Identifikácia súboru

Niektoré služby (v intervale 0 až 6 plnú a 7 až 12 čiastočnú) vyžadujú definovanie identifikácie súboru, ktorej začiatok sa ukladá do registra HL.

Identifikácia súboru sa skladá z mena disku (A:, B:) v prvých dvoch bajtoch, mena súboru (max. 10 znakov ukončených bodkou) a prípony za bodkou (posledný bajt identifikácie).

Nie je potrebné, aby malo meno 10 znakov, postačí, keď ukončíte meno bodkou a príponou.

Niektoré služby umožňujú definovanie mena súboru aj za pomoci hviezdikovej a otáznikovej konvencie. Sú to: CAT, ERASE. Vtedy je možné tak ako u príkazov samotného MDOSu definovať meno vložení hviezdiky alebo otázniku. Vtedy bude BIOS u spomenutých služieb pracovať so všetkými súbormi, ktoré vyhovujú zadanej konvencii.

Upravením hodnoty určitej premennej BIOSu, možno doceliť prácu len s vybranými súbormi. Tieto premenné možno nastavovať príslušnými službami BIOSu.

## Práca s atribútmi

BIOS dokáže korektné pracovať aj so súbormi, ktoré majú rôzne nastavené atribúty. Je možné ho prepnúť do režimu, v ktorom úplne ignoruje akýkoľvek atribút v hlavičke súboru a používa súbory akoby mali atribúty 'RWED'. Takto možno definovať, aby BIOS pri svojich operáciách načítaval súbory s vypnutým atribútom R. Pokiaľ je nastavený tak, ako pracuje MDOS, vráti v prípade pokusu o načítanie súboru s vypnutým R atribútom chybové hlásenie.

## POUŽITIE BIOSU

Prvoradou požiadavkou použitia BIOSu programátorom je vlastníctvo prekladača assembleru pre procesor Zilog Z80, editor zdrojového textu a prípadne debugger. Môže použiť napr. systém MRS, Prometheus alebo ľubovoľný iný, ktorý spĺňa podmienky použitia BIOSu.

Pred samotným začatím programovania musíte zabezpečiť umiestnenie Vášho prekladača a BIOSu do pamäti. Kód BIOSu nesmie kolidovať s umiestnením prekladača, zdrojového kódu ukladaného prekladačom do pamäti a samotným Vaším programom.

Na začiatku písania programu doporučujeme definovať si návestie BIOSu vo Vašom zdrojovom texte a zabezpečiť Váš program (prekladač) tak, aby nepoužil oblasť pamäte obsadenú BIOSom.

Definujte si návestie:



**BIOS equ xxxxx**

kde xxxxx je adresa, na ktorej je BIOS uložený programom SETUP.

Pred štartom Vášho prekladača je nutné zadať príkaz Basicu

**POKE #247,79**

Tento príkaz umožňuje BIOSu preklopiť diskovú ROM/RAM. Bez jeho použitia sa pri pokuse o použitie väčšiny služieb BIOSu počítač resetne.

Pri ladení programu je možné krokovat aj v samotnom kóde BIOSu avšak je potrebné si uvedomiť, že BIOS klopí diskovú ROM a väčšina debuggerov používa pri krokovaní inštrukciu RST a preto je krokovanie v tele BIOSu pomerne rizikové. Preto pri inštrukcii CALL BIOS nevstupujte do jej tela ale ju prejdite plnou rýchlosťou. BIOS bol odladený s maximálnou pozornosťou, preto by mal byť v ňom minimálny výskyt chýb. V prípade problémov sa orientujte najskôr na algoritmy použité vo Vašom zdrojovom texte a správnosť používania služieb BIOSu.

**ZÁVER**

Iste Vás ako majiteľa BIOSu bude zaujímať, že programy TED, ZX602 boli vyvinuté aj za pomoci prvej verzie BIOSu. BIOS bol na nich odskúšaný a väčšina chýb a nedostatkov sa už v tejto verzii nenachádza.

Pre zaujímavosť: TED sa nepredával, nepredáva a pravdepodobne sa už ani predávať nebude. Bol vyvinutý a dokončený. Je mi ľúto, ale bohužiaľ taký bombový editor si nikde nekúpíte. Niežeby som nechcel, ale vyskytli sa faktory nepriamo sa dotykajúce jeho budúcnosti.

Značka "romantic softman" žije, no so Spectrom už nemá nič spoločné.

Už len spomínam na svoje spektrácke časy, keď som začínal s kreslením v Art Studiu, so svojim editorom **HaHaWriter**, podielom na Poster Makeri a vývoji TEDa.

Ešte máte príležitosť stretnúť sa s mojou prácou, ale už to nie sú programy.

Želám programom ZX602, ZX603 a BIOS veľa úspechov. Sú to veľmi dobré programy, iste nebudete ľutovať dať za ne tých pár stoviek korún.

Dipl. ing. Robert Madaj  
"romantic softman"

## Príklady práce s BIOSom

Uvedené príklady spolupráce s Vášho programu s BIOSom sú ilustratívne a ich účelom je ozrejmiť ich konkrétne použitie v programe a slúžia ako doplnok tejto príručky. Mená návěstí sú fiktívne a nie je potrebné držať sa ich pri Vašej práci.

Doporučujeme Vám vyskúšať si prácu BIOSu práve na nich, najmä pre ich jednoduchosť.

Nezabudnite si niekde v pamäti definovať identifikáciu mena súboru. Napr. takto:

```
MENO      db      'A: SUBOR.B'
```

BIOS pracuje aj so súbormi s inými príponami ako je štandard MDOSu. Ďalej je potrebné si uvedomiť, že aj so súbormi typu P pracuje ako s bežným B súborom (nemožno očakávať, že použitím služby LOAD na súbor P sa nahrá a spustí program uložený v načítavanom súbore). Na to, aby to bolo možné, musíte zabezpečiť načítanie na správne miesto v pamäti, nastavenie príslušných systémových premenných a návrat do vykonávacej slučky programu v Basicu.

```
;=====
; VZORY VSTUPOV DO JEDNOTLIVYCH SLUZIEB BIOSU
; Ramon 94/03/20
;=====
BIOS      equ     XXXXX
;-----
SAVE      ld      a,0          ; cislo sluzby (t.j.SAVE)
          ld      hl,MENO      ; poloha mena disku a suboru
          ld      de,PAMET     ; zaciatok ukladanej oblasti
          ld      bc,DLZKA     ; dlzka suboru
          call   BIOS          ; volanie BIOSu
          jp      nz,REPORT    ; skok na vypis chyb. Hlasenia
          ret                ; navrat, vsetko je O.K.
;
;-----
LOAD      ld      a,1          ; cislo sluzby (t.j.LOAD)
          ld      hl,MENO      ; poloha mena disku a suboru
          ld      de,PAMET     ; miesto,kam sa bude nahravat
          ld      bc,DLZKA     ; max. pouzitelna dlzka suboru
          call   BIOS          ; volanie BIOSu
          jp      nz,REPORT    ; skok na vypis chyb.hlasenia
          ret
;
;-----
ERASE     ld      a,3          ; cislo sluzby
          ld      hl,MENO      ; poloha mena disku a suboru
          call   BIOS          ; volanie BIOSu
          jp      nz,REPORT    ; skok na vypis chyb.hlasenia
          ret                ; navrat, vsetko O.K.
;
```

```

;-----
CAT      ld  c,$ff      ; znacka "od zaciatku DIR"
CAT2     ld  a,3        ; cislo sluzby
          ld  hl,MENO   ; miesto disku+mena suboru
          call BIOS
          jr  z,CAT3    ; skok na vypis mena suboru
          cp  27        ; File not found, už nie je ani jedno meno
súboru v adresári
          jp  nz,REPORT ; vypis hlásenie
          ret          ; navrat do hlav.programu
;
CAT3     ld  a,(hl)    ; vyber príponu
          rst $10      ; vypis ju
          inc hl
          ld  a," "     ; znak medzera
          rst $10      ; vypis ju
          ld  b,10      ; max.dĺzka mena
CAT4     ld  a,(hl)    ; vyber znak z mena
          and a         ; testuj na koniec mena
          jr  z,CAT5
          rst $10      ; vypis znak z mena
          inc hl       ; posun sa po mene
          djnz CAT4
CAT5     ld  a,13      ; nový riadok
          rst $10
          ld  a,(BIOS+64) ; *** DOLEZITE ! ***
          ld  c,a       ; *** DOLEZITE ! ***
          jr  CAT2     ; vrat sa pre ďalšie meno v DIR
;
;-----
FIRST    ld  a,4        ; cislo sluzby
          ld  hl,MENO   ; poloha mena disku a suboru
          ld  de,PAMET  ; miesto kde bude sektor nacistany
          call BIOS     ; volanie BIOSu
          jp  nz,REPORT ; skok na vypis chyb.hlasenia
          ret
;
;-----
RENFIL   ld  a,5        ; cislo sluzby
          ld  hl,MENO   ; poloha stareho mena disku+suboru
          ld  de,PAMET  ; poloha noveho mena disku+suboru
          call BIOS     ; volanie BIOSu
          jp  nz,REPORT ; navrat cez vypis chyb.hlasenia
          ret          ; navrat, vsetko O.K.
;
;-----
RENDSK   ld  a,6        ; cislo sluzby
          ld  hl,MENO   ; poloha noveho mena disku+volume
          call BIOS
          jp  nz,REPORT ; skok na vypis chyb.hlasenia
          ret          ; navrat, vsetko O.K.
;=====
; Sluzby potrebujúce v mene len identifikáciu disku
;=====
READ     ld  a,7        ; cislo sluzby
          ld  hl,MENO   ; identifikácia disku
          ld  bc,LENGTH ; cislo sektoru diskety
          ld  de,PAMET  ; miesto kam bude nahrany
          call BIOS     ; volanie BIOSu

```

```

        jp  nz,REPORT      ; navrat cez vypis chyb.hlasenia
        ret                ; navrat, vsetko 0.K.
;
;-----
RESTOR  ld  a,8           ; cislo sluzby
        ld  hl,MENO       ; identifikacia disku
        ld  bc,LENGTH     ; cislo sektoru diskety
        ld  de,PAMET      ; miesto noveho obsahu sektora
        call BIOS         ; volanie BIOSu
        ret                ; navrat, vsetko je 0.K.
;
;-----
VOLUME  ld  a,9           ; cislo sluzby
        ld  hl,MENO       ; identifikacia disku
        call BIOS         ; volanie BIOSu
        jp  nz,REPORT     ; skok na vypis chyb.hlasenia
        ret                ; navrat, vsetko je 0.K.
;
;-----
FREE    ld  a,10          ; cislo sluzby
        ld  hl,MENO       ; identifikacia disku
        call BIOS         ; volanie BIOSu
        jr  nz,REPORT     ; skoc, ak prislo ku chybe
; HL ukazuje na polohu cisla poctu volnych sektorov
        ld  c,(hl)        ; vyzdvihni pocet do registra BC
        inc hl
        ld  b,(hl)
        ret                ; navrat, vsetko 0.K
;
;-----
MAXSEK  ld  a,11          ; cislo sluzby
        ld  hl,MENO       ; identifikacia disku
        call BIOS         ; volanie BIOSu
        jr  nz,REPORT     ; skoc, prislo ku chybe
; HL ukazuje na polohu cisla maximalneho poctu sektorov
        ld  c,(hl)        ; vyzdvihni pocet do registra BC
        inc hl
        ld  b,(hl)
        ret
;
;-----
DFORM   ld  a,12          ; cislo sluzby
        ld  hl,MENO       ; identifikacia disku
        call BIOS         ; volanie BIOSu
        jr  nz,REPORT     ; skoc, prislo ku chybe
        ld  c,(hl)        ; vyzdvihni pocet stop
        inc hl
        ld  b,(hl)        ; vyzdvihni pocet sektorov
        ret
;
;=====
; Sluzby SETUPu pre BIOS (nepotrebuju disketu v drive)
;=====
;
DEFTYP2 ld  a,13          ; cislo sluzby
        ld  bc,$8000      ; novy obsah 16.-17.Bytu hlavicky
                                ; suboru, dodany BIOSom pri SAVE
        call BIOS         ; volanie BIOSu
        ret
;

```

```

;-----
DEFTYP  ld  a,14      ; cislo sluzby
         ld  c,$00    ; novy obsah 19.Bytu hlavicky
         call BIOS    ; suboru, dodany BIOSom pri SAVE
         ret          ; volanie BIOSu
;
;-----
RATTR   ld  a,15      ; cislo sluzby
         ld  c,1      ; nova hodnota BIOS R-attributu
         call BIOS    ; volanie BIOSu
         ret
;
;-----
WATTR   ld  a,16      ; cislo sluzby
         ld  c,1      ; nova hodnota BIOS W-attributu
         call BIOS    ; volanie BIOSu
         ret
;
;-----
DATTR   ld  a,17      ; cislo sluzby
         ld  c,1      ; nova hodnota BIOS D-attributu
         call BIOS    ; volanie BIOSu
         ret
;
;-----
INESET  ld  a,18      ; cislo sluzby
         ld  c,1      ; nova hodnota BIOS I-attributu
         call BIOS    ; volanie BIOSu
         ret
;
;-----
HDSET   ld  a,19      ; cislo sluzby
         ld  c,0      ; 0-neprebera udaje, 1-prebera
         call BIOS    ; z hlavicky pri LOADE
         ret
;
;-----
LDDRV   ld  a,20      ; cislo sluzby
         call BIOS    ; volanie BIOSu
         ld  a,(hl)   ; vyzdvihni znak drivu
         ret
;
;-----
MAXDRV  ld  a,21      ; cislo sluzby
         call BIOS    ; volanie BIOSu
         ld  a,(hl)   ; vyzdvihni znak drivu
         ret
;
;-----
VERS    ld  a,22      ; cislo sluzby
         call BIOS    ; volanie BIOSu
         jp  REPORT   ; navrat cez vypis verzie BIOSu
;
;-----
REPORT  ld  a,(hl)    ; vyzdvihni znak hlasenia
         res  7,a
         rst  $10
         ld  a,(hl)
         inc hl       ; posun ju
         rl  a
         jr  nc,REPORT

```

```

ld    a,13          ; nový riadok
rst   $10
ret                   ; koniec vypisu, navrat

```

### Posledná Správa:

BIOS je možné rozširovať o ďalšie služby.

Pokiaľ by ste chceli vyvinúť verziu BIOSu pre iné prostredie OS, radi budeme spolupracovať. Záujem máme napr. o tieto platformy:

BIOS na OS Beta Disk pracujúci s disketami MDOS

BIOS na OS MDOS pracujúci s disketami MDOS

BIOS na OS ZX Diskface QUICK fy Dataputer

BIOS na OS ZX Diskface QUICK pracujúci s disketami MDOS

BIOS na OS Microdrive

BIOS na OS Disciple

popřípade ďalšie OS platformy

BIOS for TAPE je vhodný pre majiteľov ZX Emulátora na PC, ktorý nemá emuláciu MDOSu. Preto vyviniete SW pod TAPE verziou a po dokončení programu použijete BIOS for MDOS - dobré nie?

Očakávame odozvu od firiem zaoberajúcich sa vývojom SW pre Didaktik a ZX Spectrum.

BIOS sa snaží zjednotiť všetky platformy a uľahčiť tak programátorom programovanie diskových operácií.

To je zatiaľ všetko.

Ďakujeme,  
že ste si prečítali túto príručku.



# PRED OTVORENÍM OBALU SI PREČÍTAJTE LICENČNÉ PODMIENKY

Autor si ponecháva vlastníctvo tejto softwarovej kópie. Túto kópiu môžete používať za nasledujúcich podmienok:

Je dovolené:

- používať software na akomkoľvek kompatibilnom počítači za predpokladu súčasného použitia jediným používateľom na jednom počítači,
- použitie softwaru v sieti, pri údržbe súborov alebo na virtuálnom disku za predpokladu, že je program používaný jediným používateľom na jednom počítači.

Nie je dovolené:

- robiť kópie dokumentácie a diskety s programami alebo robiť kópie diskiet s programami mimo výnimiek popísaných v dokumentácii (vytvorenie záložnej kópie,
- meniť, modifikovať alebo adaptovať software alebo dokumentáciu vrátane a to nielen s obmedzením na to, prekladať, dekompiľovať, disasemblovať alebo robiť odvodené práce. Táto licencia končí automaticky nedodržaním podmienok licenčnej dohody.

Perpetuum garantuje, že programy zodpovedajú publikovanej špecifikácii a dokumentácii za predpokladu, že budú používané na počítači a operačnom systéme, pre ktoré boli určené.

Záruka alebo náhrada zhora uvedená je exkluzívna a namiesto všetkých ostatných ústnych alebo písomných, explicitných alebo implicitných. Žiadny autorizovaný predajca programu, distribútor, agent alebo zamestnanec nie je oprávnený robiť akékoľvek zmeny alebo doplnky ku tejto záruke.

Po odoslaní registračnej karty ste oprávnený:

- ku zákaznickemu servisu (HOTLINE),
- ku výmene produktu formou (UPGRADE alebo UPDATE).

Ďalšie Vaše práva a záruky sú uvedené v dokumentácii.

Pokiaľ nesúhlasíte so znením tejto zmluvy, vráťte software v NEPORUŠENOM obale svojmu predajcovi do 10 dní od zakúpenia spolu s originálom dokladu o zaplatení. V tom prípade Vám bude vrátená čiastka, ktorú ste pri nákupe zaplatili.

**OTVORENÍM OBALU AUTOMATICKY VSTUJUJE DO PLATNOSTI TÁTO ZMLUVA**

## NEPREHLIADNITE TECHNICKÉ ÚDAJE

**VERZIA**

Česká

Slovenská

**FORMÁT**

5,25"

3,5"

**POČET**

**HARDWARE**

ZXS, Didaktik

IBM PC

D40, D80

= NUTNÉ  = VHODNÉ

**MIROSLAV**  
B E N I Č E K

**Miroslav Beniček, P. Horova 1/41 841 08 Bratislava, tel.: 07/774111**

ORIGINAL DESIGN BY ROMANTIC SOFTMAN