

O P E R A Č N Í S Y S T É M

CCC	PPPP	/	M	M	222	222
C	C P	/	M	M	2 2	2 2
C	P P	/	MM	MM	2	2
C	PPPP	/	M M M		2	2
C	C P	/	M	M	2	.. 2
CCC	P	/	M	M	22222	.. 22222

Implementace na Sharp MZ-800

VERZE: 1.2

Jiří Lamač

Praha, únor 1989

OBSAH:

	str.
1. Předmluva	3
2. Úvod do operačního systému CP/M	5
2.1 Co to je operační systém	5
2.2 Vznik CP/M	6
2.3 Základní informace o CP/M	8
2.4 Interpret příkazů CCP	14
2.5 Standardní služby BDOSu	16
2.6 Modul BIOS	26
3. Implementace CP/M na Sharpu - popis systému	30
3.1 Vlastnosti použitého CCP a BDOS	30
3.2 Zavedení systému	33
3.3 Obsluha klávesnice	35
3.4 Obsluha obrazovky	37
3.5 Tiskárna a seriové rozhraní RS-232	38
4. Diskové operace	39
4.1 Formát záznamu na disketových mechanikách	41
4.2 Výběr mechaniky	43
4.3 Operace čtení a zápisu	44
4.4 Práce s diskovým bufferem	45
5. Základní programové příslušenství	47
5.1 Program FORMAT	49
5.2 Program COPYSYS	50
5.3 Program COPYDISK	51
5.4 Program SETUP	52
5.5 Program DISKDEF	52
5.6 Program DEFKEY	53
5.7 Program KEYB	54
5.8 Programy CLS, EJECT a BOOT	55
5.9 Program DIAPRINT	55
5.10 Programy MLOAD a MSAVE	56
5.11 Program TIME	59
5.12 Program XDOS	59
6. Než začnete pod systémem pracovat	60
6.1 Využití programu z jiného počítače	60
7. Literatura	62

1. Předmluva

Sharp MZ-800 je dnes u nás jedním z nejlepších zahraničních osmibitových počítačů. Jeho hardwarová konstrukce je v mnohém podobná počítačům pro profesionální použití. Uživatel, který se však rozhodl si tento počítač koupit, byl jistě velmi zklamán, když zjistil, že na Sharpa není prakticky žádné firemní programové vybavení. Ideální je samozřejmě používat operační systém CP/M, který běžnému uživateli nabízí prakticky vše, co jen může od počítače požadovat. Je možný jednoduchý přístup k souborům, lze užívat kvalitní programy firem zvučných jmen, jako jsou překladače snad všech významných programovacích jazyků (Turbo Pascal, Ada, Basic, Fortran, Cobol, C, Prolog, Forth, Lisp), textové editory (WordStar, Word Master), databázové programy (dBase II), programy pro tabulkové výpočty (Multiplan, SuperCalc), makroasemblyery a dokonce i křížové asemblyery pro jiné mikroprocesory. Důležitý je také fakt, že implementací CP/M Sharp MZ-800 získává kompatibilitu s jinými osmibitovými počítači u nás, jako je např. TNS, SAPI 1, Robotron 1715, Sinclair ZX Spectrum, Schneider CPC 664 a 6128, ale i Commodore 128 a další.

Firma Sharp pro svůj počítač MZ-800 nabízí diskový operační systém, nazvaný P-CP/M, ve kterém jsou však závažné softwarové chyby, které způsobují "záhadné" mizení částí souborů apod. P-CP/M rovněž neumí spolupracovat s jinými disketovými mechanikami, než firemními čtyřicetistopými a není zcela standardní, protože pod ním nepracují některé programy. Největším nedostatkem P-CP/M je jeho až neuvěřitelná pomalost, připomínající výkon počítače s procesorem I8080 na 1 MHz. Na druhou stranu je v P-CP/M použito některých pěkných myšlenek, např. nepotřebuje stisknout CTRL/C po výměně diskety.

Z výše uvedených důvodů vznikl nový operační systém, který myšlenkově z P-CP/M vychází, v žádném případě však není jeho modifikací nebo kopií. Minimum vnějších změn přináší výhodu, že si uživatel P-CP/M přechodem na nový CP/M nemusí zvykat na žádné novinky. Popisovaný CP/M zachovává dobré vlastnosti původního řešení, ale eliminuje jeho zápory. Mimo jiné umožňuje spolupráci se širokou škálou disketových mechanik, které se u nás mezi lidmi vyskytují. Dále zvyšuje pracovní rychlost (podle druhu činnosti dva- až dvacetkrát).

I přes některé vlastnosti pro CP/M netypické (možnost volně vyměňovat diskety v mechanikách, kopírovat soubory bez nutnosti volání speciálního kopírovacího programu, nebo užívat rozšířených funkcí logického rozdělení disku - USERS), které nový systém má, lze říci, že i přes tyto změny se jedná o naprosto standardní verzi CP/M 2.2, která má určitou nadstavbu. Například umožňuje volbu další činnosti programu při chybě (Retry / Abort / Ignore), lze na ní psát v úplné češtině (podle normy KOI-8-Čs), má zabudované grafické znaky pro kreslení oken a jednoduchých obrázků atd. Dále popisovaný CP/M umí obsluhovat všechny standardně dodávané periferie, počínaje již zmíněnými floppy disky, přes samozřejmou Centronics nebo IRPR tiskárnu, tři různé RAM disky o nastavitelné kapacitě až do 512 K a seriové rozhraní

RS-232C. Formát záznamu na disku je přímým ekvivalentem záznamu používaného na IBM PC, takže diskety jsou bez problémů přenositelné. Součástí systémové diskety je i množství programů pro poměrně variabilní rekonfiguraci systému (kterou si uživatel může provést sám, aniž by přímo zasahoval do strojového kódu systému), definici uživatelských kláves, či rychlé kopírování disket.

Všem uživatelům tohoto systému přejeme co největší pohodu a spokojenost při práci pod naším systémem a zároveň děkujeme za všechny případné konstruktivní návrhy a připomínky.

(c) 1989 Jiří Lamač, autor systému

2. Úvod do operačního systému CP/M

Kapitola 2 se týká operačního systému CP/M obecně, nezávisle na popisované implementaci, proto ti, kteří CP/M již znají odjinud, mohou pokračovat kapitolou 3.

Zároveň doporučujeme těm, kteří hodlají k systému přistupovat jen jako uživatelé (a to je správný postoj), aby přeskočili kapitoly 2.5 a 2.6 popisující BDOS a BIOS. Tyto části, převzaté z [17], jsou potřebné při volání systému ze strojového kódu a v tomto textu jsou zahrnuty spíše pro úplnost.

2.1 Co to je operační systém

Malé osobní počítače, takové jako je např. Sinclair ZX-81 nebo samotný Sharp MZ-800 bez disketových mechanik a RAM disku, nemají skutečný operační systém, i když se to někdy mylně uvádí. Je téměř pravidlem, že bývají vybaveny pevnou pamětí ROM o kapacitě 16 až 48 kB, ve které bývá umístěn interpret jazyka Basic, případně ještě hexadecimální monitor a textový editor. O nějakém standardním systému ani známky. Není pak divu, že člověk, který nikdy nepracoval na větších počítačích (např. SMEP), si o operačním systému, případně o sdílení času, může leda nechat zdát.

O co tedy vlastně jde. Učená definice říká, že operační systém je souhrn určitého množství programů, které přesně definovaným způsobem zajišťují komunikaci mezi hlavním programem a uživatelem. Je to jakýsi mezičlánek, který umí zpracovat informace dodané mu hlavním programem a zároveň obsluhuje periférie (to by zpravidla neměla být starost hlavního programu). U větších počítačů se systém stará též o přidělování strojního času a paměti různým současně běžícím úlohám.

Řeknete si, vždyť to není nic nového. Obsluha klávesnice a obrazovky je přece v pevné paměti každého, i toho nejmenšího počítače. Ale ouha, když si do Sharpa MZ-800 přenesete program třeba z Robotronu 1715, těžko asi bude pracovat. Tady už to nikomu divné nepřijde, vždyť je to úplně jiný počítač. Když ale oba počítače mají podobný mikroprocesor, mělo by to přece nějak jít. A jsme u toho.

Každý počítač obsluhuje určité periférie, které se obecně ovládají různě (jen kolik je např. různých videoprocessorů). Operace prováděné s perifériemi však jsou stále stejné - "přečti znak", "zapiš znak" a u počítačů s disky ještě "přečti z disku" a "zapiš na disk". Jestliže dohodneme nějaký způsob, jakým budou programy volat obslužné podprogramy, mohou být programy bez omezení přenosné mezi naprosto odlišnými počítači. Operační systém CP/M zde představuje takový celosvětově dohodnutý (a na osmibitových počítačích nejvíce rozšířený) standard, definující toto propojení.

2.2 Vznik CP/M

CP/M (Control Program for Microcomputers) je v současné době pravděpodobně nejpopulárnější operační systém pro osmibitové mikropočítače s mikroprocesorem Z80, I8080 nebo 8085. Přes 300 firem vyrábějících mikropočítače dodává CP/M jako součást standardního programového vybavení a odhaduje se, že CP/M je využíván asi na 800 000 pracovištích.

CP/M je sice již starší, ne však zastaralý systém. Jeho první verze byla vytvořena již v roce 1974, jako důsledek značného rozšíření mikroprocesoru Intel 8080. Do té doby totiž byla v programovém vybavení pro tento typ mikroprocesoru značná nejednotnost. Otcem systému je Garry Kildall, původně programový konzultant u firmy Intel. Firma uvažovala o vytvoření standardního systému pro I8080. Pracovalo na něm více programátorů, avšak po rychlém vzestupu firmy byly softwarové problémy odsunuty do pozadí pozornosti (jak se to bohužel běžně dělá u nás). G. Kildall se s firmou Intel nepohodl, navíc bylo konstatováno, že jeho výtvor nemá žádné zvláštní vlastnosti, především mu chybí nepřehledné množství rozličných příkazů, jako mají jiné bohatší a honosnější systémy. G. Kildall však věděl, že v jednoduchosti je síla, a pracoval na dalším vývoji CP/M na vlastní pěst. Zajímavé je, že k pozdějšímu bezkonkurenčnímu rozšíření CP/M došlo i přes nezájem firmy Intel.

V téže době přišel John Torode, zaměstnanec firmy Digital Systems, s řadičem mechaniky pro pružné disky, který umožnil perfektní spolupráci CP/M s diskovou pamětí. Předvádění prototypové verze CP/M mělo značný ohlas mezi nadšenými vyznavači mikropočítačových systémů a zájem projevíly i nově vznikající firmy produkující mikropočítačové stavebnice a systémy. Vznikla tak potřeba instalovat operační systém CP/M na různé mikropočítačové systémy.

V letech 1974 až 1976 přepracoval G. Kildall CP/M tak, že ze systému vyčlenil části závislé na konkrétním technickém prostředí. Systém se tak rozdělil na technicky závislou a technicky nezávislou část, s přesně definovaným rozhraním. Při přenosu systému bylo nutno přizpůsobit technicky závislou část novému prostředí, zbytek systému zůstal beze změn, včetně uživatelských příkazů. Vznikla tak verze označovaná CP/M 1.3, která byla jako první zveřejněna a otevřela CP/M brány do světa uživatelů. V roce 1976 založil G. Kildall firmu Digital Research, která distribuuje CP/M a související programy.

Další vývoj CP/M byl ovlivněn především vývojem diskových pamětí. Vzniká verze CP/M 1.4, která dovoluje ovládat standardní disk firmy IBM (formát IBM 3740 - osmipalcový jednostranný pružný disk s jednoduchou hustotou záznamu). Široké použití tohoto disku dovolilo vysokou přenosnost souborů mezi jednotlivými instalacemi CP/M a tím nebývalou přenosnost programových produktů.

Vývoj pružných disků se však nezastavil - vznikají disky dalších rozměrů (5 1/4 a 3 1/2 ") s rozmanitými formáty záznamu.

Objevily se rovněž pevné disky s velkými kapacitami, použitelné v mikropočítačových systémech. Operační systém CP/M byl proto upraven tak, aby dovolil ovládat co nejširší škálu diskových pamětí. Do technicky závislé části byly zabudovány tabulky, popisující vlastnosti použitých disků. Příslušná verze z roku 1979 nese označení CP/M 2.2, a je nejrozšířenější verzí systému CP/M. U nás je známý spíše pod označením Mikros. Rozšířené jsou i východoněmecké verze SCP, SCPX a odvozené.

Vývoj operačního systému pokračoval ještě dále. Byla vytvořena verze CP/M 3.1 (někdy také označovaná CP/M PLUS), která umí ovládat bankovanou paměť, má rozšířenou škálu systémových služeb, možnost nápovědy apod. Existují také varianty CP/M, umožňující spouštění dalších úloh (Concurrent CP/M), multiuživatelský přístup z více terminálů (MP/M), či sdílení prostředků v rámci sítě (CP/NET). Tyto verze však nedosáhly výraznějšího rozšíření, neboť nejsou zcela standardní a některé programy na nich nepracují.

Navíc se pozice systému CP/M výrazně posílila, když firma Zilog přišla s mikroprocesorem Z80, pro který byla napsána spousta skvělých programů pracujících pod CP/M (odhady hovoří o 5000 programů). Tento mikroprocesor díky podstatně vyššímu hodinovému kmitočtu (až 8 MHz) a bohatšímu instrukčnímu souboru umožnil výrazné zrychlení všech činností systému i uživatelských programů. V roce 1979 byl CP/M oceněn cenou Datapro Software Honor Roll.

Firma Intel kupodivu nakonec systém CP/M přes všechny jeho výhody a velké rozšíření nepřijala a vydala se cestou svého vlastního operačního systému ISIS, který se však neujal a je používán pouze ve vývojových systémech Intel.

V čem spočívá neobyčejný úspěch operačního systému CP/M ? Systém vyniká značnou jednoduchostí a nezávislostí na technických prostředcích. Přitom jde o mocný systém s možností snadného rozšiřování. Jistě by bylo možné navrhnout systémy, které by byly z jistých hledisek rafinovanější než CP/M. Otázkou však zůstává, zda by se podařilo zachovat pružnost při tak malých nárocích na paměť. Jedná se o velmi účelný kompromis mezi mocností příkazů a minimalizací nároků na paměť. Rovněž strategie využívání diskového média je jednoduchá a přitom spolehlivá. Celková struktura systému je navržena tak chytře, že byla převzata ve většině operačních systémů pro mikropočítače.

2.3 Základní informace o CP/M

CP/M je monoprogramový diskově orientovaný operační systém. Je přísně modulární, přičemž návaznost na technické prostředky je soustředěna v jediném modulu, který je lehký modifikovatelný.

Služby systému umožňují tvorbu další programové nadstavby a to systémových programů i účelového aplikačního programového vybavení.

CP/M je rozdělen na tři moduly:

- vstupně/výstupní modul (BIOS, Basic Input/Output System),
- jádro systému (BDOS, Basic Disk Operating System),
- interpret příkazů (CCP, Console Command Processor).

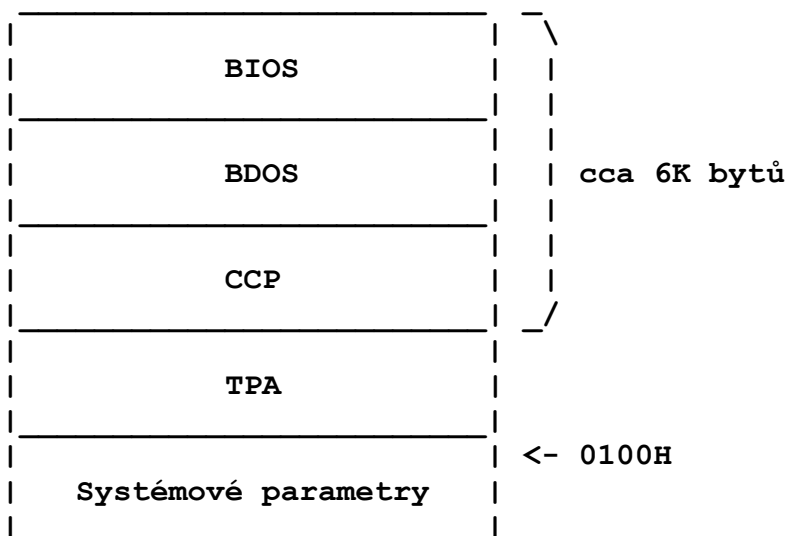
Vstupně/výstupní modul obsahuje fyzické drivery jednotlivých periférií včetně disků. Tento modul představuje základní část operačního systému a umožňuje v konfiguraci technických prostředků používat různé typy disků.

Jádro tvoří na konfiguraci technických prostředků nezávislý operační systém, který zajišťuje vykonávání všech systémových služeb.

Interpretr příkazů, který umožňuje komunikaci operátora se systémem na úrovni konzole, tvoří relativně nezávislý modul CP/M. Je-li to potřebné, může být v paměti překryt uživatelským programem nebo jeho daty.

Po zavedení systému zbyde v paměti RAM volná část, nazývaná pracovní oblast (TPA, Transient Program Area). Slouží k zavádění a běhu služebních a uživatelských programů a k uložení jejich dat. Tato oblast začíná na adrese 0100H, která je též startovací adresou všech programů pracujících pod systémem.

Umístění jednotlivých modulů v paměti RAM je na následujícím schématu:



Ze zobrazeného rozdělení paměti vyplývá, že operační systém je rezidentní na konci paměti RAM. Je to proto, aby TPA začínala vždy na stejné adrese. V případě, že velikost TPA nestačí, mají programy možnost překrýt modul CCP. Tímto způsobem je uživatelským programům v rámci dané konfigurace paměti poskytnuta co největší oblast pro jejich práci.

Nejspodnější část paměti (od adresy 0000 do 00FFH) je rezervována pro systémové informace, které jsou nutné pro práci systému. Tyto informace budou popsány dále.

CP/M je schopen ovládat 1 až 16 různých diskových jednotek, přičemž každý soubor na disku může mít kapacitu až 8 MB. Každý z disků má svůj vlastní adresář, ve kterém jsou uloženy informace o souborech na disku. Počet položek adresáře může být předem určen. Položka adresáře obsahuje informace o rozsahu a umístění příslušného souboru na disku. Systém ovládání souborů umožňuje přímý přístup ke kterémukoli z až 65536 záznamů max. 8 MB souboru. Každému z diskových souborů je možno přiřadit příznaky "systémový" a "chráněný proti zápisu". Přes tuto velkou flexibilitu typů disků zabezpečuje systém ovládání souborů logicky jednotný a rychlý přístup k jednotlivým souborům.

Systém ovládání souborů používá pevnou délku logického záznamu - 128 bytů. Diskům o větší kapacitě vyhovuje (z důvodu rychlosti přístupu) větší délka fyzického záznamu. Na to je v CP/M též pamatováno a v BIOSu je možno implementovat algoritmus kumulující více logických 128 bytových záznamů do delšího fyzického záznamu s jediným omezením, že jeho délka musí být násobkem 128 bytů.

Systém ovládání souborů CP/M rozděluje disk na tzv. alokační bloky. Všechny alokační bloky na jednom disku mají stejnou délku. Jsou tvořeny celistvým počtem sektorů, který je vždy mocninou 2 (např. často používaná délka alokačního bloku je $2^3 = 8$ sektorů = 1 KB, příp. $2^4 = 16$ sektorů). Je-li souboru přidělováno místo na disku, děje se tak po diskretních jednotkách - alokačních blocích. Alokační bloky jsou číslovány 0, 1, 2 atd. Informace o uložení daného souboru na disku je jednoznačně daná čísly alokačních bloků tohoto souboru. Tento způsob má proti tradičním technikám ukládání souboru na disku formou zřetězeného seznamu dvě výhody:

- Nemůže dojít z důvodu přerušování seznamu k ztrátě informace o umístění zbytku souboru,
- je zaručené, že přinejmenším ta část souboru, která vyplňuje alokační blok, je na disku uložena kontinuálně (v těch sektorech, které tvoří alokační blok) a tedy v případě sekvenčního přístupu k tomuto souboru je přístupová doba minimalizovaná. Navíc systém ovládání souborů v případě potřeby přiřazuje souboru další volný alokační blok fyzicky nejbližší k poslednímu alokovanému bloku souboru. Pohyby čtecí hlavy jsou tím omezené na minimum.

Na jednotlivé soubory na disku se CP/M odvolává pomocí jména. Jméno souboru je tvořeno 1 až 8 ASCII znaky; případně je možné za ním udat ještě verzi, která je tvořena 1 až 3 ASCII znaky a je od vlastního jména souboru oddělena tečkou. Příklad: Soubory TURBO a TURBO.COM jsou dva různé soubory, ze kterých první je specifikovaný jen jménem, druhý má jméno stejné jako 1. soubor, ale odlišuje se od něho verzí COM.

Jak už bylo uvedeno, soubory umístěné na disku jsou katalogizované v adresáři. Adresář daného disku je umístěný vždy v počátečních alokačních blocích (0 a výše), v každém případě zabírá alespoň jeden alokační blok, a to nultý. Systém ovládání souborů považuje alokační bloky adresáře za obsazené (alokované).

Položka adresáře je tvořena popisem jednoho souboru na daném disku. To znamená, že kolik položek má adresář, tolik souborů může disk obsahovat. Počet položek se definuje při implementaci systému. Položka adresáře zabírá 32 bytů. To znamená, že do jednoho sektoru adresáře se vejdu 4 položky. Položka obsahuje kromě jiného příznak obsazení položky, jméno a verzi souboru, počet sektorů souboru a čísla alokačních bloků, které byly souboru přidělené. V položce je vždy 16 míst pro čísla alokačních bloků. Jestliže seznam alokačních bloků zabírá více než 16 míst, musí být pro něj v adresáři rezervovaná další položka, která popisuje toto další rozšíření (extension) souboru.

Zrušíme-li soubor, projeví se to v adresáři tak, že do příznaku obsazenosti položky daného souboru se zapíše E5H. Místo v adresáři, které položka zabírala, je možno využít pro položku popisující jiný soubor. Příznak zrušení položky (a tedy i souboru) - konstanta E5H - byla zvolena proto, že distributoři pružných disků popisují disky právě touto konstantou a systému ovládání souborů se takový disk jeví jako prázdný - všechny položky v adresáři mají příznak "zrušený".

Programátor, který pracuje s diskovým souborem, se na tento soubor odvolává pomocí řídicího bloku souboru (FCB - File Control Block). FCB je vlastně položkou adresáře (má v podstatě stejnou strukturu) a při práci se souborem jsou do něho přepisované některé údaje z položky adresáře (např. čísla alokačních bloků souboru), které jsou po dobu práce se souborem aktualizované a při uzavření souboru jsou zapsané zpět do adresáře.

Struktura FCB je následující:

DN	N1	...	N8	V1	V2	V3	EX	S1	S2	RC	AB0	...	AB15	CR	R0	R1	R2
0	1		8	9	10	11	12	13	14	15	16		31	32	33	34	35

FCB zabírá 36 bytů s následujícím významem:

- DN - číslo diskové jednotky, na které chceme vykonávat diskovou operaci;
1 znamená jednotku A,
2 " " B,
.
.
16 znamená jednotku P (max. 16 jednotek označujeme A až P),
0 znamená už předtím (implicitně) vybranou jednotku. Vybraná jednotka (nebo častěji disk), je jednotka na které budou probíhat diskové operace.
- N1 až N8 - jméno souboru, přičemž v případě, jestliže je kratší než 8 znaků, je do tohoto počtu doplněno mezerami,
- V1 až V3 - verze souboru, v případě potřeby doplněná mezerami. Protože verze je tvořena ASCII znaky, může být 7. bit kódu využit pro jiné účely:
Je-li 7. bit V1 nastaven, je daný soubor chráněn proti zápisu (značí se R/O, Read Only), je-li zrušen, soubor není chráněn (R/W, Read/Write),
Je-li 7. bit V2 nastaven, soubor je systémový (značíme SYS) a v seznamu souborů (DIR) se nevypíše, při zrušeném bitu není soubor systémový (značíme DIR).
- EX - číslo "části" souboru (extension) popsané jednou položkou adresáře. Části (EX) číslujeme 0, 1, 2 atd.
- S1, S2 - rezervované pro operační systém,
- RC - počet záznamů (sektorů), které obsahuje daná část (EX) souboru,
- AB0 až AB15 - čísla alokačních bloků daného souboru, které obhospodařuje operační systém,
- CR - číslo záznamu (sektoru), se kterým se bude pracovat (v rámci dané části - extension souboru) při sekvenčních operacích (číst/zapisovat). Chce-li uživatel zpracovávat soubor od začátku, nuluje CR.
- R0 až R2 - tvoří absolutní číslo záznamu (bez ohledu na část EX souboru), které se využívá při přímém přístupu k danému záznamu.

Každý soubor, se kterým programátor pracuje, musí být popsán položkou FCB, přičemž místo pro daný FCB může programátor rezervovat kdekoli ve svém programu. Když přistupuje k souboru, musí vyplnit prvních 16 bytů FCB (obvykle vyplní DN, jméno, verzi a zbytek šestnáctice nuluje) a musí inicializovat CR. V případě přímého přístupu musí namísto CR inicializovat R0 až R2.

V případě prvního přístupu na disk zjišťuje operační systém

o tomto disku některé důležité informace, které potom udržuje v paměti. Jednou z takovýchto velmi důležitých informací je např. obsazenost disku. Operační systém musí zjistit, které alokační bloky jsou již přiřazeny a které jsou ještě volné, aby v případě zápisu nového souboru nepřepsal (použitím už přiřazeného alokačního bloku) některý z existujících souborů. Operační systém přečte celý adresář, kde v jednotlivých položkách jsou čísla už přiřazených alokačních bloků a vyrobí alokační vektor disku. Tento vektor je tvořen posloupností bitů. Kolik alokačních bloků má disk, tolik bitů má alokační vektor. Každý bit odpovídá jednomu alokačnímu bloku. Má-li hodnotu 0, je daný alokační blok volný, v opačném případě je již přiřazen. Operační systém potom během své práce aktualizuje tento alokační vektor (při rušení souboru nuluje, při zápisu nastavuje ty bity, které odpovídají příslušným alokačním blokům).

Jak bylo uvedeno, operační systém vyrobí z adresáře alokační vektor disku jen jednou, v případě prvního přístupu na disk. Potom ho už jen aktualizuje. Pro disk, s kterým systém po svém zavedení ještě nepracoval, nemá vytvořený alokační vektor. Disk, pro který už existuje alokační vektor, nazýváme aktivním diskem. Aktivních disků může být několik. Jestliže systém pracoval s prvním diskem, pak s druhým a naposled se třetím, všechny tyto tři disky jsou již aktivní. O tom, které disky jsou aktivní, má systém informaci v tzv. vektoru aktivních disků.

Představme si, že v době práce systému někdo v jednotce, se kterou se již pracovalo (je aktivní) vymění disk. Je jisté, že alokační vektor tohoto nového disku nebude totožný s alokačním vektorem předcházejícího disku a při zápisu na tento nový disk (kdy systém využívá alokační mapu předcházejícího disku - neví, že disky byly vyměněny) dojde k prepisování jeho už existujících souborů. Aby se to nestalo, vytváří si operační systém při prvním přístupu na disk kontrolní součty jednotlivých sektorů adresáře a ty potom při své práci kontroluje a aktualizuje. Když nenastane shoda kontrolních součtů disku, se kterým pracuje, se součty, které si vyrobil, oprávněně předpokládá, že se jedná o jiný disk a označí ho jako R/O - chráněný proti zápisu a tak zabráni přepsání informace na tomto disku.

Při výměně disku se proto doporučuje stlačit CTRL/C, což způsobí kromě jiného, že všechny disky v systému se označí jako neaktivní, t.j. že se pro všechny budou vytvářet alokační vektory znovu, a ty už budou odpovídat vyměněným diskům.

Na závěr si všimneme oblasti systémových parametrů. Jak už bylo řečeno, tato oblast je na adresách 0000 až 00FFH.

Na adrese 0000 je skok do BIOSu, který způsobí znovuzavedení modulů CCP a BDOS ze systémového disku do paměti, nastaví adresu DMA na 0080H a všechny disky označí jako neaktivní. (Tuto funkci označujeme jako WBOOT, Warm Boot).

Na adrese 0005 je skok do BDOSu a pomocí této adresy jsou volány všechny systémové služby.

Skoky na adresách 0000 a 0005 t.j. skoky do BIOSu a do BDOSu jsou při generování systému nasměrovány na skutečné adresy odpovídající uložení operačního systému pro daný rozsah paměti.

Od adresy 005CH vytváří CCP řídicí blok souboru (FCB) pro soubor daný jako parametr příkazu interpretovaného CCP.

Oblast od 0080H do 00FFH, t.j. 128 bytů, tvoří buffer pro jeden sektor disku. DMA se automaticky nastavuje na tuto adresu při funkci WBOOT. Všechny diskové operace se vykonávají po sektorech (128 bytů) a operační systém nemá vlastní I/O buffer pro tyto operace. Programátor musí tento buffer nastavovat sám. Oblast od adresy 0080H využívá též CCP a odevzdává v ní programátorovi text příkazového řádku, který zadal z konzole.

2.4 Interpretér příkazů CCP

Uživatel komunikuje s CP/M pomocí interpretera příkazů, který čte příkazy zadávané z konzole.

Jednotlivé diskové jednotky označuje CCP písmeny A, B, C, až P (max. počet jednotek, se kterými je CP/M schopen pracovat, je 16), za kterými se píše dvojtečka ":". Říkáme, že disk je vybraný, když CCP adresuje právě tento disk. Aby bylo vždy jasné, který disk je vybraný, CCP informuje operátora kódem tohoto disku (A, B, ..P), který je následován symbolem ">", kterým CCP oznamuje, že je schopen přijmout příkaz. Například, v případě, že je vybraný disk B, CCP se na obrazovce zahlásí výpisem:

B>

a čeká na příkaz operátora. Když chceme vybrat pro následující operace disk A, zadáme jeho kód:

B>A:

a CCP se ohlásí

A>

Kódem disku spolu s dvojtečkou udaným před jménem souboru udáváme, na kterém disku je daný soubor. Jestliže kód disku neudáme, CCP předpokládá, že soubor je na disku, který je právě vybraný.

Příkazy CP/M jsou implementované ve dvou úrovních:

- příkazy zabudované v CCP,
- příkazy realizované pomocí programů rezidentních na discích.

Každý příkaz je aktivován zápisem jeho jména, případně parametrů a stlačením klávesy CR. Jde-li o příkazy DIR, ERA, REN, SAVE, TYPE, USER a d:, zabudované v CCP, jsou ihned vykonány. Ostatní, tzv. tranzientní příkazy, hledá CCP na disku. V případě nalezení je zavede do TPA a odevzdá jim řízení.

Funkce zabudovaných příkazů je následující:

DIR vypíše seznam souborů aktivní nebo zadané jednotky
- např. DIR B:*.COM nebo jenom DIR,
ERA zruší specifikované soubory - ERA B:A*.*,
REN provede přejmenování souboru - REN NEW.MAC=OLD.MAC,
SAVE uloží požadovaný počet stránek oblasti TPA (od začátku)
na disk pod zadaným jménem - SAVE 16 HELP.TXT,
TYPE vypíše obsah souboru (týká se pouze textových souborů),
přičemž výpis lze ukončit stiskem libovolné klávesy,
USER nastaví číslo uživatele - USER 0 (až 15).

Při psaní parametrů příkazů je povoleno užívání hvězdičkové konvence, tj. lze napsat otazník místo libovolného znaku a

hvězdička nahrazuje skupinu znaků.

Interpreter příkazů využívá jako řídicí znaky některé nezobrazitelné znaky ASCII kódu. Tyto znaky jsou generované konzolí displeje při současném stlačení klávesy CTRL a příslušného písmene. Zmíněné znaky se označují jako CTRL/<písmeno>. Např. CTRL/C generuje kód 03H apod. Následuje seznam těchto řídicích znaků i s jejich funkcemi:

CTRL/C	(03)	znovuzavedení operačního systému,
CTRL/E	(05)	fyzické ukončení řádku,
CTRL/P	(10H)	logické zapnutí/vypnutí tiskárny,
CTRL/S	(13H)	přerušení výstupu na konzoli a vykonávání programu až do vstupu jakéhokoli dalšího znaku,
CTRL/R	(12H)	výpis obsahu vstupujícího řádku,
CTRL/U	(15H)	zrušení vstupujícího řádku,
CTRL/X	(18H)	vymazání vstupujícího řádku z obrazovky i z operační paměti,
CTRL/Z	(1AH)	označení konce souboru (u textových souborů).

Pro programy rezidentní na disku konstruuje CCP ze zbytku příkazového řádku, který následuje bezprostředně za jménem programu, dva FCB. První od adresy 005CH, druhý od adresy 006CH. Je-li parametrem příkazu jméno souboru, FCB tohoto souboru připraví CCP od adresy 005CH. V případě, že je uvedený druhý parametr, potom pro tento je vytvořen FCB na místě AB0 až AB15 prvního FCB a musí být přesunut do jiné oblasti paměti, aby nebyl aktivací prvního FCB přepsán.

Dále je pro programy rezidentní na disku oblast od adresy 0080H inicializovaná ASCII řetězcem, který tvoří zbytek příkazového řádku. Na adrese 0080H je délka tohoto řetězce a od adresy 0081H následují jednotlivé ASCII znaky. Tímto způsobem může volaný program získat potřebné parametry.

2.5 Standardní služby BDOSu

Každá služba volaná do systému je realizovaná modulem BDOS, který dle potřeby volá drivery BIOSu. Každá služba má přiřazeno číslo, které se při volání předává v registru C. V případě, kdy služba vyžaduje vstupní parametr, např. adresu FCB, je systému poskytnut v registrech DE. Je-li zapotřebí vstupní parametr jednobytový (např. znak), odevzdává se v registru E, obsah D je pak nepodstatný.

Jestliže služba vrací volajícímu programu dvoubytový parametr, je tento v registrech HL. Jednobytový výstupní parametr vrací v registru A. Všeobecně pro výstupní parametr platí B=H a A=L, t.j. obsahy registrů B, H a A, L jsou stejné.

Volání služby se realizuje odevzdáním řízení pomocí instrukce CALL 0005H. Na této adrese je instrukce JP BDOS, která zabezpečí správný vstup do BDOSu nezávisle na tom, kde je BDOS v paměti umístěn. Každá služba se tedy volá následující sekvencí instrukcí:

```
LD C,číslo služby
LD DE,parametr
CALL 5
```

přičemž návrat ze služby je vykonán na adresu následující za instrukcí CALL 5.

Je potřebné si uvědomit, že služby BDOSu nezachovávají obsahy registrů !

služba: Reset systému

číslo služby: 0

vstupní parametr: -

výstupní parametr: -

funkce: Odevzdává řízení do BIOSu, který vykoná znovuzavedení modulů CCP a BDOS z disku do paměti, nastaví adresu DMA na hodnotu 0080H, zachová předtím vybraný disk a číslo uživatele. Potom odevzdá řízení na CCP. Tato služba má stejný efekt jako instrukce JP 0.

služba: Vstup znaku z konzole

číslo služby: 1

vstupní parametr: -

výstupní parametr: ASCII znak

funkce: Přečte znak z konzole (čeká na jeho vstup). Když je kód přečteného znaku <20H (SPACE) a není to znak CR, LF, TAB nebo BS, nevypíše ho na konzoli. Ostatní znaky zobrazí na konzoli.

služba: Výstup znaku na konzoli

číslo služby: 2

vstupní parametr: ASCII znak výstupní parametr: -

funkce: Zobrazí vystupující znak na konzoli, případně i na tiskárně, je-li zapnuta pomocí CTRL/P. Znak TAB zobrazí jako

příslušný počet mezer.

služba: Vstup znaku ze snímače

číslo služby: 3

vstupní parametr: -

výstupní parametr: ASCII znak

funkce: Přečte znak ze snímače děrné pásky (čeká na jeho vstup).

služba: Výstup znaku na děrovač

číslo služby: 4

vstupní parametr: ASCII znak

výstupní parametr: -

funkce: Vyděruje znak na děrovači děrné pásky.

služba: Výstup znaku na tiskárnu

číslo služby: 5

vstupní parametr: ASCII znak

výstupní parametr: -

funkce: Vytiskne znak na tiskárně.

služba: Přímý vstup/výstup na konzoli

číslo služby: 6

vstupní parametr: 0FFH (pro vstup znaku)

nebo ASCII znak (pro výstup)

výstupní parametr: ASCII znak (byl-li požadován vstup)

nebo nedefinovaná hodnota.

funkce: Je-li vstupní parametr 0FFH, čte status konzole (ptá se, zda nebyla stlačena klávesa). Jestliže nebyla žádná klávesa stlačena, vrací jako výstupní parametr 00, v opačném případě vrací ASCII reprezentaci vloženého znaku. Z uvedeného vyplývá, že se nečeká na vstup znaku z konzole! Na rozdíl od služby č. 1 nezobrazuje vstupní znak na konzoli. Je-li vstupní parametr různý od 0FFH, považuje tento parametr za ASCII kód vypisovaného znaku a vypíše ho na konzoli.

služba: Zjistí hodnotu I/O bytu

číslo služby: 7

vstupní parametr: -

výstupní parametr: hodnota I/O bytu

funkce: Poskytne obsah adresy 0003. Na této adrese je u systému Intel I/O byte.

služba: Nastav I/O byte

číslo služby: 8

vstupní parametr: hodnota I/O bytu

výstupní parametr: -

funkce: Na adresu 0003 uloží vstupní parametr, který představuje I/O byte.

služba: Vypiš ASCII řetězec

číslo služby: 9

vstupní parametr: adresa počátku řetězce výstupní parametr: -

funkce: Vypíše na konzoli textový řetězec od zadané adresy. Řetězec musí být ukončen znakem "\$" (24H), před kterým výpis

skončí. Výpis jednotlivých znaků vykonává stejně jako služba č.2.

služba: Čti řetězec znaků z konzole

číslo služby: 10

vstupní parametr: adresa paměťového bufferu

výstupní parametr: -

funkce: Vstupní parametr je adresou počátku bufferu, který má následující strukturu:

MAX - 1 byte

SKUT - 1 byte

ZNAKY - SKUT bytů, kde

MAX je maximální počet znaků (bytů), které je možné uložit do bufferu,

SKUT bude skutečný počet znaků načtených do bufferu. Za tímto bytem jsou ZNAKY, t.j. ASCII reprezentace znaků tak, jak přišly z konzole. Vstup znaků z konzole je ukončen znakem CR nebo LF, které se ale do bufferu již neukládají. Buffer, který je třeba v paměti rezervovat pro načítání řádku z konzole, musí mít délku MAX + 2 byty. Jestliže byl buffer naplněn MAX počtem znaků, je vstup ukončen stejně, jako kdyby vstoupil znak CR.

Při zadávání řetězce znaků je možné vykonávat následující editační funkce:

- DEL vymaže naposled napsaný znak z bufferu a zobrazí tento rušený znak na konzoli (DEL se do bufferu neuloží).
- BS vymaže naposled zapsaný znak z bufferu i z obrazovky (BS se do bufferu neuloží).
- CTRL/E vykoná fyzický přechod kurzoru na nový řádek obrazovky (obsah bufferu se nezmění).
- CTRL/X způsobí vymazání všech vložených znaků z bufferu i z obrazovky.
- CTRL/U způsobí vymazání všech vložených znaků z bufferu a nastaví kurzor na novém řádku pod pozici, odkud začíná načítávaný řádek.
- CTRL/R nastaví kurzor na nový řádek pod pozici, odkud začíná načítávaný řádek a vypíše aktuální obsah bufferu.
- CTRL/C - Jestliže byl tento znak stlačený jako první, způsobí znovuzavedení operačního systému, jinak se uloží do bufferu.
- CTRL/P zapne/vypne tiskárnu, t.j. další vložené znaky se budou zobrazovat i na tiskárně (když tomu dosud tak nebylo), až do následujícího stlačení CTRL/P. Hodnota CTRL/P (10H) se do bufferu neuloží.

Ostatní znaky, pokud jsou zobrazitelné, t.j. jejich kód je $\geq 20H$, se při vstupu zobrazují na konzole (případně i na tiskárně při CTRL/P). Řídící znaky se zobrazují jako dvojznaky a to znak "^" a příslušné písmeno. Například 02 se zobrazí jako ^B (B má kód 42H), 1AH se zobrazí jako ^Z (Z má kód 5AH).

služba: Zjistí status konzole

číslo služby: 11

vstupní parametr: -

výstupní parametr: hodnota odpovídající stavu konzole

funkce: Je-li ve vstupním bufferu klávesnice nějaký znak (t.j. pokud byla stlačena klávesa konzole), vrátí hodnotu 01, v opačném případě vrátí 00.

služba: Zjistí číslo verze systému

číslo služby: 12

vstupní parametr: -

výstupní parametr: číslo verze

funkce: Z důvodu možnosti dalšího rozvoje CP/M a z toho vyplývající možné nekompatibility mezi jednotlivými verzemi (při předpokladu, že vyšší verze může mít další nové služby, které nižší verze nemá), poskytuje tato služba uživatelským programům možnost zjistit verzi operačního systému. Volání vrátí v registru jednobytové číslo představující číslo verze (v našem případě 22H, protože jde o systém 2.2).

služba: Reset diskového systému

číslo služby: 13

vstupní parametr: -

výstupní parametr: -

funkce: Volání zabezpečí inicializaci diskového systému, t.j. nastaví DMA adresu na 0080H, vybere disk A, všechny ostatní disky označí jako neaktivní a všechny disky včetně disku A označí jako R/W. Tato služba není ekvivalentní se službou č. 0. Nedojde k znovuzavedení systému.

služba: Vyber disk

číslo služby: 14

vstupní parametr: číslo vybíraného disku

výstupní parametr: -

funkce: Označí zadaný disk jako "vybraný", t.j. všechny následující diskové operace budou probíhat na tomto disku, pokud není explicitně v FCB uvedeno jinak. Vstupní parametr je 0 pro disk A, 1 pro B atd. až 15 pro P. Toto nastavení platí až do následujícího resetu diskového systému.

služba: Otevři soubor

číslo služby: 15

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Před prvním čtením ze souboru, nebo prvním zápisem do existujícího souboru je potřebné získat informace o tomto souboru. K tomuto účelu používáme službu "Otevři soubor". Soubor, který chceme otevřít, udáváme adresou jeho FCB. BDOS prohlédne adresář zadaného disku pro aktivního uživatele a hledá položku adresáře, která odpovídá bytům 1 až 14 v udaném FCB. (T.j. hledá položku, které odpovídá jméno, verze, EX, S1 a S2, přičemž S2 automaticky nuluje). Jestliže se ve jméně nebo verzi udané v FCB vyskytuje na pozici některého znaku otazník "?", není tento znak s odpovídající pozicí v položce adresáře porovnáván.

Nastane-li shoda s některou položkou adresáře (adresář je prohledáván od začátku), je informace z této položky adresáře přepokopovaná do FCB (t.j. čísla alokačních bloků souboru jsou dosazeny na pozici 16 až 31 v FCB).

Služba Otevři soubor tedy zabezpečí obsazení daného FCB informacemi o rozsahu a umístění daného souboru na disku. Tyto informace v FCB jsou během následujících diskových operací s uvedeným souborem aktualizovány. Opětovné uložení aktualizovaného FCB do odpovídající položky adresáře zabezpečí služba "Zavři

soubor". Výstupní parametr (kód adresáře) bude mít hodnotu 0FFH, jestliže daný soubor nebyl v adresáři nalezen, v opačném případě má hodnotu 0 až 3, což odpovídá pozici dané položky adresáře v sektoru adresáře (- pro aplikační programování nepodstatné).

služba: Zavři soubor

číslo služby: 16

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Soubor, který byl nějakým způsobem aktualizován, je potřebné uzavřít, t.j. informace o rozsahu souboru a jeho uložení na disku, které jsou během práce s tímto souborem průběžně zaznamenávané v FCB, je potřebné uložit do příslušné položky adresáře. Soubor, z kterého se jen četlo, není nutno zavírat, protože jeho rozsah ani umístění na disku se nezměnily.

Jestliže byla operace uzavření úspěšná, výstupní parametr má hodnotu 0 až 3, v případě neúspěšného zavírání (např. soubor není na disku) vrací služba hodnotu 0FFH.

služba: Hledej první odpovídající položku v adresáři

číslo služby: 17

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Služba hledá od začátku adresáře položku, kterou popisuje FCB. V případě, kdy najde zadanou položku, dá k dispozici od DMA adresy záznam (128 bytů) adresáře, ve kterém je daná položka, a jako výstupní parametr vrátí kód adresáře (0 až 3). Volající program má potom tuto položku na adrese DMA + 32 * A. Jestliže hledaný soubor není v adresáři, má výstupní parametr hodnotu 0FFH. Ve jménu, nebo verzi souboru, udaného v FCB se může vyskytovat jeden, nebo více otazníků "?". V tomto případě, stejně jako při službě "Otevři soubor", není odpovídající znak v položce adresáře kontrolovaný na shodu. Otazník tedy zastupuje jakýkoli znak. V případě, že se otazník nachází na pozici DN (0. byte FCB), je prohledáván adresář vybraného disku a služba vrátí první nalezenou položku, která odpovídá danému FCB, i když je tato položka označena jako zrušená.

služba: Hledej následující položku adresáře

číslo služby: 18

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Služba je ekvivalentní se službou č. 17 s tím rozdílem, že hledání je vykonáváno od předtím nalezené položky. Tuto službu je možné volat pouze bezprostředně po službě č. 17 nebo č. 18.

Výstupní parametr má hodnotu 0FFH, jestliže tato položka v adresáři neexistuje, v opačném případě nabývá hodnoty 0 až 3, což je pořadové číslo položky v záznamu (sektoru) adresáře.

služba: Zruš soubor

číslo služby: 19

vstupní parametr: adresa FCB výstupní parametr: kód adresáře

funkce: Na disku se zruší soubor odpovídající zadanému FCB. Jméno a verze souboru mohou obsahovat jeden nebo více otazníků. V tomto

případě jsou zrušeny všechny soubory odpovídající dané specifikaci. Služba vrací hodnotu OFFH, jestliže zadaný soubor nebyl v adresáři nalezen, v opačném případě vrací hodnotu 0 až 3.

služba: Sekvenční čtení

číslo služby: 20

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Ze souboru (udané pomocí FCB), který byl předtím aktivován službou "Otevři soubor", přečte následující 128 bytový záznam do paměti na adresu DMA. Záznam se přečte od pozice CR dané částí souboru a CR se přitom automaticky zvýší tak, že ukazuje na následující záznam souboru. (CR udává počet přečtených záznamů z dané části souboru). Jestliže se CR přeplní, automaticky se otevře další část (EX) souboru a CR se vynuluje.

Služba vrátí hodnotu 00, když byla operace čtení úspěšná. Nenulová hodnota indikuje, že soubor už neobsahuje žádná data (jsme na konci souboru).

služba: Sekvenční zápis

číslo služby: 21

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Do souboru (popisovaného FCB), který byl předtím aktualizován službou "Otevři" nebo "Založ soubor", se zapíše 128 bytový záznam z adresy DMA. Záznam je uložen na pozici, na kterou ukazuje CR a tento se přitom automaticky zvýší tak, že ukazuje na následující budoucí záznam souboru. Když se CR přeplní, automaticky se otevře další část souboru a CR se nuluje. Při vykonání zápisu do existujícího souboru je příslušný starý záznam (daný CR) přepsán novým záznamem.

Služba vrátí hodnotu 00, byla-li operace zápisu úspěšná. Nenulová hodnota indikuje plný disk.

služba: Založ soubor

číslo služby: 22

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Služba "Založ soubor" je podobná službě "Otevři soubor", s rozdílem, že FCB musí obsahovat jméno souboru, který se v adresáři nenachází. Operační systém vytvoří na disku prázdný soubor o nulové délce a jeho jméno zapíše do adresáře. Služba vrací hodnotu 0 až 3 při úspěšné operaci, nebo OFFH, když v adresáři už není místo. Do takto vytvořeného souboru je možno zapisovat pomocí služeb č. 21 a č. 34.

služba: Přejmenuj soubor

číslo služby: 23

vstupní parametr: adresa FCB

výstupní parametr: kód adresáře

funkce: Služba změní jméno souboru, specifikovaného prvními šestnácti byty FCB, na jméno, které je zadáno v druhé šestnáctici FCB. Kód disku na pozici 0 v FCB se využije pro výběr disku. Při návratu ze služby má výstupní parametr hodnotu 0 až 3 při úspěšném přejmenování, nebo OFFH, když přejmenováváný soubor na

disku není.

služba: Vrať vektor aktivních disků

číslo služby: 24

vstupní parametr: -

výstupní parametr: vektor aktivních disků (2 byty)

funkce: Služba poskytne v registrech HL informaci o discích, které jsou aktivní, t.j. pro které systém ovládání souborů vytvořil příslušné tabulky v operační paměti. Bit 0 reg. L odpovídá jednotce A, bit 7 reg. H odpovídá jednotce P. Má-li bit hodnotu 1, znamená to, že odpovídající jednotka je aktivní.

služba: Vrať číslo vybraného disku

číslo služby: 25

vstupní parametr: -

výstupní parametr: číslo vybraného disku

funkce: Služba vrátí číslo vybraného disku, t.j. číslo 0 pro disk A, atd. až č. 15 pro disk P.

služba: nastav DMA adresu

číslo služby: 26

vstupní parametr: požadovaná DMA adresa

výstupní parametr: -

funkce: DMA adresa (DMA = Direct Memory Access) je adresa v operační paměti, kam se čte (odkud se zapisuje) sektor diskového souboru.

Poznámka: V případě většiny řadičů pružných disků nejde o přímý přístup do paměti, jak název DMA napovídá, ale DMA je třeba chápat jako ukládací adresu v operační paměti.

Adresa DMA zůstává pro všechny následující diskové operace nezměněna až do dalšího volání služby č. 26. Vyjimku tvoří volání služeb č. 0 a 13, které nastavují DMA adresu na 0080H.

služba: Dodej adresu alokačního vektoru

číslo služby: 27

vstupní parametr: -

výstupní parametr: adresa alokačního vektoru

funkce: Služba poskytne v registrech HL adresu počátku alokačního vektoru vybraného disku. Alokační vektor je tvořen posloupností bitů, přičemž každý bit této posloupnosti odpovídá příslušnému alokačnímu bloku. Je-li alokační blok volný, má příslušný bit hodnotu 0, v opačném případě má hodnotu 1. Délka alokačního vektoru, t.j. počet jeho bitů odpovídá max. počtu alokačních bloků a tedy závisí na kapacitě disku.

služba: Označ disk jako chráněný proti zápisu

číslo služby: 28

vstupní parametr: -

výstupní parametr: -

funkce: Služba označí vybraný disk jako chráněný proti zápisu (R/O). Na tento disk není možné zapsat soubor, mazat nebo přejmenovávat soubor. Tato ochrana trvá až do doby volání služby 0 nebo 13, které všechny disky v systému označí jako R/W, t.j. zruší jejich ochranu.

služba: Dodej hodnotu R/O vektoru
číslo služby: 29
vstupní parametr: -
výstupní parametr: hodnota R/O vektoru
funkce: Služba poskytne v registru HL hodnotu R/O vektoru tvořeného posloupností bitů, z kterých každý odpovídá jednomu disku v systému. Má-li příslušný bit hodnotu 1, je odpovídající disk chráněn proti zápisu. Nejméně významný bit odpovídá disku A, atd.

služba: Nastav atributy souboru
číslo služby: 30
vstupní parametr: adresa FCB
výstupní parametr: kód adresáře
funkce: Služba změní atributy souboru "chráněn proti zápisu" a "systémový" podle hodnoty příslušných bitů verze v FCB, tj. nastavený 7. bit prvního znaku verze označuje R/O, 7. bit druhého znaku verze označuje soubor jako systémový.

služba: Dodej adresu bloku diskových parametrů
číslo služby: 31
vstupní parametr: -
výstupní parametr: adresa bloku diskových parametrů
funkce: Volání této služby poskytne v registru HL adresu v BIOSu, od které začíná blok diskových parametrů vybraného disku. Tento blok udává fyzické charakteristiky příslušného disku.

služba: Poskytni/změň číslo uživatele
číslo služby: 32
vstupní parametr: - 0FFH pro poskytnutí čísla uživatele,
- hodnota různá od 0FFH pro nastavení nového čísla uživatele
výstupní parametr: číslo uživatele (bylo-li požadováno)
funkce: Je-li vstupní parametr 0FFH, poskytne služba číslo právě nastaveného uživatele (0 až 15). V případě, že je vstupní parametr různý od 0FFH, předpokládá se, že je to číslo uživatele, které je třeba nastavit.

služba: Přímé čtení ze souboru
číslo služby: 33
vstupní parametr: adresa FCB
výstupní parametr: kód chyby
funkce: Volání služby poskytne na nastavené DMA adrese 128 bytový záznam ze souboru (předtím aktivovaného službou "Otevři soubor"), popsaného v FCB, přičemž absolutní pořadové číslo tohoto záznamu v souboru bylo udáno v FCB+33 a FCB+34. Při volání služby musí být hodnota bytu na adrese FCB+35 nulová! Tato služba, na rozdíl od sekvenčního čtení, automaticky nezvýší udanou hodnotu čísla záznamu.

Číslo požadovaného záznamu může nabývat hodnoty 0 až 65535 a může proto specifikovat část až 8 megabytů dlouhého souboru.

Služba navíc automaticky nastaví v FCB odpovídající část souboru (EX) i číslo přečteného záznamu (CR).

Vrácený výstupní parametr reprezentuje následující chybové kódy:

- 0 - žádná chyba, operace se vykonala,
- 1 - byla čtena nezapsaná data,
- 2 - tento kód není využit,
- 3 - nemůže uzavřít danou část souboru,
- 4 - chce číst z nezapsané části souboru,
- 5 - plný adresář,
- 6 - pokus o čtení mimo fyzický konec disku.

služba: Přímý zápis do souboru

číslo služby: 34

vstupní parametr: adresa FCB

výstupní parametr: kód chyby

funkce: Služba je podobná službě předešlé s tím rozdílem, že vykonává přímý zápis do souboru. I chybové kódy, které služba vrací, jsou stejné.

služba: Vypočti délku souboru

číslo služby: 35

vstupní parametr: adresa FCB

výstupní parametr: vrácená délka souboru na adresách FCB+33, FCB+34, FCB+35

funkce: Služba vrátí počet záznamů souboru udaného v FCB na adresy FCB+33, +34, +35. Soubor může mít délku 0 až 65536 záznamů (sektorů) a na uložení tohoto čísla jsou nutné 3 byty. Délka souboru podle nejvyššího pořadového čísla záznamu v nejvyšší přiřazené části souboru udává tedy pořadové číslo prvního volného záznamu na konci souboru. Byl-li ale soubor zapsán pomocí služby č. 34, nemusí délka odpovídat počtu uložených záznamů.

služba: Vrať číslo záznamu pro přímý zápis

číslo služby: 36

vstupní parametr: adresa FCB

výstupní parametr: číslo záznamu na adresách FCB+33 a FCB+34

funkce: Při sekvenčním čtení nebo zápisu je udané v FCB číslo části (EX) souboru a pořadové číslo záznamu v této části (CR). Volání služby 36 poskytne na adresách FCB+33 a FCB+34 absolutní pořadové číslo záznamu (od začátku souboru) nehledě na právě otevřenou část souboru.

služba: Reset diskové jednotky

číslo služby: 37

vstupní parametr: vektor diskových jednotek (reg. DE)

výstupní parametr: 0

funkce: Na rozdíl od služby č. 13, která vykonávala reset celého diskového systému (t.j. všech diskových jednotek), zabezpečuje tato služba reset jen zadaných diskových jednotek. Jako vstupní parametr se odevzdává 16 bitový vektor (délka odpovídá max. možnému počtu jednotek v systému), přičemž nejnižší bit odpovídá jednotce A, nejvyšší jednotce P. Jednotkám, na které chceme aplikovat reset (t.j. nastavit jako neaktivní a nechráněné proti zápisu) nastavíme v odevzdávaném vektoru odpovídající bit.

číslo služby: 38

Tato služba není implementovaná

číslo služby: 39

Tato služba není implementovaná

služba: Popiš alokační blok nulami a potom do něho ulož záznam

číslo služby: 40

vstupní parametr: adresa FCB

výstupní parametr: kód chyby

funkce: Služba má stejný efekt jako služba č. 34 s tím rozdílem, že v případě prvního zápisu do nově alokovaného bloku se tento nejdříve popíše binárními nulami a až pak se do něho uloží záznam. Vykonává vlastně před zápisem "inicializaci" obsahu alokovaného bloku.

Chybový kód, který služba vrací, je stejný, jako v případě služby č. 34.

2.6 Modul BIOS

Vstupně/výstupní modul (BIOS) je modul, ve kterém jsou soustředěny všechny části operačního systému, závislé na technických prostředcích, na kterých je systém CP/M implementován. Tento modul má jednoznačně definovaný vstupní vektor, přes který jsou volány drivery (obslužné podprogramy) jednotlivých zařízení. Implementovat CP/M na konkrétní typ počítače znamená v konečném důsledku napsat BIOS pro tento počítač.

Dále se budeme zabývat především společnými znaky různých BIOSů, t.j. vstupním vektorem, přes který může uživatel programovat přímo vstupně/výstupní operace, aniž by volal jádro operačního systému. Popis BIOSu pro Sharp MZ-800 je součástí tohoto textu a je o něm pojednáno později. Případného zájemce o podrobnější popis klasického BIOSu odkazují na [15], resp. [16].

Vstupní vektor se nachází na začátku BIOSu. Má 17 položek, z kterých každá představuje 3 bytovou instrukci skoku na příslušný podprogram BIOSu. Struktura vektoru je následující:

```
JP BOOT    ;vstupní bod do BIOSu při počátečním zavedení
           ;operačního systému z disku do paměti
JP WBOOT   ;zajistí znovuzavedení modulů CCP a BDOS do paměti
JP CONST   ;zjistí stav konzole
JP CONIN   ;čte znak z konzole
JP CONOUT  ;vypíše znak na konzoli
JP LIST    ;vypíše znak na tiskárně
JP PUNCH   ;vyděruje znak do děrné pásky
JP READER  ;přečte znak ze snímače děrné pásky
JP HOME    ;nastaví hlavu na stopu 0 vybraného disku
JP SELDSK  ;vybere diskovou jednotku
JP SETTRK  ;nastaví číslo stopy
JP SETSEC  ;nastaví číslo sektoru
JP SETDMA  ;nastaví DMA adresu
JP READ    ;čte nastavený sektor
JP WRITE   ;zapíše nastavený sektor
JP LISTST  ;vrátí status tiskárny (její připravenost)
JP SECTRN  ;převede logické číslo sektoru na fyzické
```

Vstupní vektor BIOSu je možno rozdělit na tři logické části: inicializaci systému (BOOT, WBOOT), jednoznakové I/O operace (CONST, CONIN, CONOUT, LIST, PUNCH, READER, LISTST) a diskové operace (HOME, SELDSK, SETTRK, SETSEC, SETDMA, READ, WRITE, SECTRN).

O všech jednoznakových operacích se předpokládá, že jsou vykonávány se znaky v kódu ASCII, s nulovým sedmým bitem. Na periferní zařízení CP/M hledí jako na "logická", konkrétní fyzická zařízení definuje BIOS.

Logická periferní zařízení, která CP/M ovládá, jsou:

konzole: Základní I/O periférie, přes kterou operátor komunikuje

se systémem. Nejčastěji je to displej s klávesnicí (terminál), ale obecně to může být i psací stroj, dálnopis apod.

tiskárna: Zařízení pro výpisy textů (mozaiková tiskárna, rychlá řádková tiskárna apod.).

děrovač: Zařízení pro děrování děrné pásky.

snímač: Zařízení pro čtení děrné pásky.

V BIOSu napíše implementátor drivery pro konkrétní zařízení, která mají funkce odpovídající uvedeným logickým zařízením. Je zřejmé, že daný systém nemusí všechny tyto periférie obsahovat. Pro práci CP/M je nezbytná pouze konzole.

Nutno poznamenat, že poslední dvě jmenovaná zařízení (pro práci s děrnou páskou) jsou dnes uváděna jaksí pro úplnost, neboť jejich využití mělo význam spíše v době vzniku CP/M. V dnešní době je výhodnější místo těchto zařízení zabudovat např. přídavná sériová rozhraní.

Diskové operace se provádějí jako posloupnost volání jednotlivých podprogramů, které nastavují číslo diskové jednotky, stopy, sektoru a adresu v paměti pro danou operaci. Příslušné parametry musejí být nastaveny ještě před samotnou operací čtení nebo zápisu a zůstávají nezměněné až do okamžiku, kdy nastavíme jejich novou hodnotu.

V následujícím textu jsou detailně popsány jednotlivé rutiny obsažené v BIOSu.

- BOOT - Vstupní bod do BIOSu, na který odevzdává řízení zavaděč operačního systému (Cold Start Loader) po zavedení celého systému do paměti. Inicializuje některé systémové parametry.
- WBOOT - Na tuto adresu je odevzdané řízení při volání služby č. 0 (reset systému). Adresa 0 v oblasti systémových parametrů obsahuje instrukci skoku na tuto položku vstupního vektoru BIOSu. WBOOT zabezpečí znovuzavedení modulů CCP a BDOS do paměti a odevzdává řízení na CCP. Skok na adresu 0 (JP 0) je obvyklý způsob ukončení práce uživatelských programů a odevzdání řízení operačnímu systému.
- CONST - Přes tento bod se získá stav vstupu konzole. V registru A poskytne tento podprogram hodnotu 0FFH, pokud je na konzoli připravený znak pro vstup, jinak vrací nulu.
- CONIN - Podprogram přečte ze vstupu konzole znak do registru A a nuluje sedmý (paritní) bit. Podprogram čeká, dokud na konzoli není znak k dispozici.
- CONOUT - Podprogram zabezpečí výpis znaku z registru C na konzoli.
- LIST - Znak z registru C se vypíše na tiskárně.
- PUNCH - Znak z registru C se vyděruje do děrné pásky.
- READER - Ze snímače děrné pásky se přečte znak do registru A. Podprogram na znak čeká.
- HOME - Na vybraném disku nastaví hlavu na nultou stopu.
- SELDSK - Vybere disk, kterého číslo je v registru C (0 pro disk

A, 1 pro B atd.) pro následující diskové operace. Podprogram vrátí v registru HL adresu 16 bytové oblasti (tzv. hlavičky bloku diskových parametrů) patřící vybranému disku. Při pokusu vybrat neimplementovaný disk vrací hodnotu 0000.

Struktura hlavičky bloku diskových parametrů a vlastního bloku diskových parametrů přesahuje rámec tohoto přehledu a je popsána v [16].

- SETTRK** - V registru BC se BIOSu předává číslo stopy, na které se má vykonat následující disková operace. Podprogram nastaví tuto stopu.
- SETSEC** - V registru BC se zadává číslo sektoru, na kterém se má vykonat následující disková operace. Podprogram nastaví tento sektor.
- SETDMA** - V registru BC se zadává počáteční adresa oblasti paměti (DMA adresa) pro následující čtení nebo zápis na disk. Například jestliže při BC = 1000H nastavíme DMA adresu, budou následující diskové operace čtení i zápisu vykonávané na adresách 1000H až 107FH tak dlouho, dokud tuto adresu nezměníme.
- READ** - Podprogram čte jeden záznam o délce 128 bytů z vybraného disku, nastavené stopy a sektoru na nastavenou adresu DMA.
V registru A vrátí 0 v případě úspěšné operace a 1 v případě chyby. Chybový kód 1 je jednotný pro všechny druhy chyb, ať už jde o chybu adresace na disku, ztrátu dat, chybu kontrolního součtu atd.
- WRITE** - Podprogram zapíše jeden záznam o délce 128 bytů na nastavený sektor na nastavené stopě na vybraném disku z nastavené DMA adresy.
V registru A je vrácena 0 při úspěšné operaci, nebo 1 v případě chyby.
- LISTST** - V registru A vrátí hodnotu 0, jestliže tiskárna není schopna převzít vypisovaný znak. V případě její připravenosti se vrací hodnota 0FFH.
- SECTRN** - Podprogram zabezpečuje převod logického čísla sektoru na fyzické. Vstupními parametry jsou logické číslo sektoru v registru BC a adresa převodní tabulky v registru DE. Logické číslo sektoru je vlastně indexem v této tabulce. Podprogram poskytne v registru HL výsledné fyzické číslo sektoru.
CP/M nemusí zapisovat jednotlivé sektory na disk v plynule stoupajícím pořadí, ale z důvodu minimalizace přístupové doby mohou být sektory zaznamenány na přeskáčku. Tento způsob záznamu potom v konečném důsledku znamená, že čísla sektorů, ve kterých je sekvenčně zapsán soubor, se neshodují s čísly odpovídajících sektorů na disku. Říkáme, že sektory každé stopy jsou přečíslované. Je-li např. parametr tohoto přečíslení roven 6, vypadá posloupnost po sobě následujících fyzických sektorů takto: 1, 7, 13, 19, 25, 5, 11, 17 atd. Logickému číslu sektoru 0 pak odpovídá fyzický (skutečný) sektor č. 1, logickému číslu 1 odpovídá sektor 7, logickému číslu 2 odpovídá sektor 13 atd.

Ze všech funkcí realizovaných přes vstupní vektor, kromě BOOT a WBOOT vykonává BIOS návrat pomocí instrukce RET.

Na závěr této kapitoly je znázorněn způsob, jak může uživatel programovat diskové operace přes vstupní vektor nezávisle na umístění BIOSu v paměti. Toto je potřebné v případě, kdy vykonáváme konverzi diskových souborů mezi různými operačními systémy apod.

Víme, že na adrese 0 v oblasti systémových parametrů je instrukce skoku na 2. položku, t.j. na instrukci JP WBOOT vstupního vektoru BIOSu. Adresová část této instrukce tedy obsahuje adresu instrukce JP WBOOT. Protože struktura vstupního vektoru je pevně daná, znamená to, že instrukce JP SELDSK, JP SETTRK, JP SETSEC, JP SETDMA, JP READ, JP WRITE je o 24, 27, 30, 33, 36, 39 bytů dále od instrukce JP WBOOT (její adresu můžeme zjistit) a tak můžeme napsat následující podprogramy:

```
DISK:    LD    HL,(1)    ;vyber disk
         LD    DE,24
         ADD  HL,DE
         JP   (HL)
```

```
TRACK:   LD    HL,(1)    ;nastav stopu
         LD    DE,27
         ADD  HL,DE
         JP   (HL)
```

```
WRT:     LD    HL,(1)    ;zapiš sektor
         LD    DE,39
         ADD  HL,DE
         JP   (HL)
```

Funkci vyber disk E a nastav stopu 37H pak můžeme programovat následujícím způsobem:

```
LD    C,4    ;disk E
CALL  DISK
LD    A,L    ;je zadaný disk v systému ?
OR    H
JR    Z,ERROR ;jednotka není v systému
LD    BC,37H ;číslo stopy
CALL  TRACK  ;nastav zadanou stopu
```

3. Implementace CP/M na Sharpu - popis systému

Sharповskému CP/M jsou věnovány celé dvě následující obsáhlé kapitoly. Podrobný a úplný popis programové obsluhy všech zařízení by ovšem byl zbytečně obsáhlý a není ani nutný pro účely pochopení funkce systému jako celku. Snažil jsem se proto zachovat stručnost, aniž by byly vypuštěny některé důležité informace.

Vzhledem k rozšířené funkci některých služeb oproti "klasické" CP/M je nový BIOS poměrně složitý program. Svědčí o tom i jeho délka (kolem 6 kilobytů), která ve srovnání s obyčejným BIOSem je asi pětikrát větší. Rozsah paměti, kterou zabírá celý systém, se tím zdvojnásobuje. Proto je systém implementován tak, jako by pracoval na počítači s 60 KB RAM. CCP začíná na adrese D400H, BDOS na DC00H a BIOS na adrese EA00H, přičemž zabírá místo až do konce paměti RAM (FFFFH). Co se týče obsahu modulů CCP a BDOS, i když mají standardní délku a funkci, jsou přepsané do instrukčního souboru procesoru Z-80 a zároveň mírně přepracované na rozdíl od původního jádra CP/M od Digital Research.

3.1 Vlastnosti použitého CCP a BDOS

Prvotním impulsem k přepracování těchto standardních modulů byla nešťastná vlastnost klasického CP/M, že potřeboval stisknout CTRL/C po každé výměně diskety. Při opomenutí této akce došlo pak dříve či později k nucenému opuštění běžícího programu, jakmile se uživatel pokusil na nově vloženou disketu psát. Tuto situaci nebylo možno žádným způsobem obejít, opomineme-li různé programy řešící tento problém přidáním dalšího rezidentního modulu pod CCP.

Nejdříve byla upravena příslušná část BDOSu, ošetřující detekci výměny diskety (nesouhlasící kontrolní součty adresáře) následným výpočtem nových kontrolních součtů a samozřejmě aktualizací celého alokačního vektoru. Tyto změny byly provedeny v původním BDOSu, a to nepřiliš populárním způsobem, kterému se občas říká záplata (přímo v kódu). Protože to ale nebylo elegantní, dospěla situace nakonec k tomu, že originální CCP a BDOS byly podrobeny důkladnému disassemblování a všechny změny v obou modulech pak byly prováděny ve zdrojovém textovém souboru. Současně došlo k přepsání programu do instrukčního souboru procesoru Z-80, což přineslo jednak určité zrychlení a také zkrácení modulů, takže mohly být do nich doplněny některé další funkce.

Vzniklý BDOS má na rozdíl od standardu tyto nové vlastnosti:

- 1) Diskety v mechanikách lze měnit v libovolný okamžik, aniž by bylo třeba po výměně tisknout CTRL/C (BDOS si to provede sám). Není pouze dobré např. měnit disketu v době, kdy je otevřen nějaký soubor, se kterým se pracuje. To ovšem není žádný prohřešek proti standardu, neboť něco podobného nelze dělat snad v žádném operačním systému. Přitom samozřejmě funkce

klávesy CTRL/C zůstala zachována, takže její stisk způsobí aktivaci pouze vybrané jednotky.

- 2) Při volání služby č. 13 (reset systému) nedochází k aktivaci jednotky A:, jak tomu bylo dříve, nýbrž se pouze reaktivuje právě vybraná jednotka. Při návratu ze služby je nastavena jednotka A:, jak to popisuje standard, avšak není aktivována. Aktivace jednotky znamená totiž vytvořit nový alokační vektor, a to podle zaplnění diskety může trvat i jednu až dvě sekundy. Zvláště při běhu dávky (SUBMIT) z jiné jednotky než A: je toto zdržení dost citelné. Obvykle totiž po návratu z resetu stejně nastavujeme jinou než systémovou jednotku a protože nyní vůbec nedojde k její aktivaci, celý proces se urychlí.
- 3) Z BDOSu bylo vypuštěno původní chybové hlášení "R/O disk" (protože nyní už k tomu nemůže dojít) a dále hlášení "Bad sector" a "Select", protože tyto chyby si ošetřuje BIOS. Hlášení "R/O file" bylo ponecháno s tím, že je doplněno volbou "Abort/Ignore". Volbou Abort vypadneme do DOSu, stiskem "I" nastane návrat ze služby tak, jako by byl soubor smazán, přejmenován či přepsán, ale ve skutečnosti k tomuto nedošlo. Při ukládání nového souboru se starým jménem teď budou v adresáři oba soubory zároveň, a je na nás, co s nimi pak uděláme. Každopádně však nemusíme opuštěním programu přijít o třeba i důležitá data.
- 4) Byly rozšířeny možnosti využívání dělení disku do uživatelských oblastí (USERS). Týká se to služeb pro otevření souboru, sekvenční i přímé čtení a výpočet délky souboru. V těchto případech nedochází k obvyklému prohlížení adresáře, ale nové prohlížení má dva průchody. První průchod je totožný s původním prohledáváním adresáře. Je-li nalezen hledaný soubor, služba končí a vrací se do volajícího programu. Jestliže hledání však končí neúspěchem, adresář je prohledáván znovu od začátku s tím rozdílem, že není porovnáváno číslo uživatele. Dojde proto i k nalezení souboru stejného jména, který byl však zaznamenán pod jiným číslem uživatele. To umožňuje volat programy i od jiných uživatelů, avšak nemůžeme tyto programy či soubory mazat či jinak modifikovat. Při výpisu adresáře (DIR) se tyto soubory pochopitelně nevypisují. Navenek se to celé jeví tak, jako by soubory z ostatních uživatelů měly příznak SYS a R/O.

Také modul CCP byl mírně přepracován, a to:

- 1) Na první pohled je vidět změněný prompt. CCP se hlásí

```
A0>_
```

Je zřejmé, že mezi kódem disku a znakem připravenosti se vypisuje ještě číslo aktuálního uživatele, aby bylo vždy jasné, jaký uživatel je právě aktivní.

- 2) Je přidán další rezidentní příkaz, a sice COPY. Tento příkaz je poměrně užitečný, neboť při práci neustále něco někam kopírujeme a jsme nuceni na to volat speciální kopírovací program (PIP nebo POWER) což zdržuje jednak vlastním zaváděním tohoto programu a jednak je zmenšen paměťový prostor využitelný ke kopírování. Rezidentní COPY samozřejmě využívá celou oblast TPA. Syntaxe je stejná jako u příkazu REN:

COPY DestFile=SourceFile

Na místě zdrojového i cílového souboru je nunto zadat úplné jméno souboru, takže není možné např. napsat

COPY a:=b:*.*

což ale není závažné omezení, protože větší počet souborů nejčastěji kopírujeme pomocí dávky (SUBMIT) a zde není problém všechny soubory vypsát.

- 3) Dávka (SUBMIT) běží z libovolné aktivní jednotky a uživatele (není vázána na systémový disk). To je výhodné v případě, že využíváme rychlý RAM disk, který přitom není adresován jako A:.
- 4) V případě zaplnění disku (připadá v úvahu během SAVE nebo COPY) následuje smazání souboru, který zaplnění způsobil. Nehrozí proto situace, že by na disku zůstala část původního souboru a chyběl jeho konec (původní CCP to tak dělal).
- 5) Je ošetřen stav, kdy zavádíme z disku do TPA dlouhý program a zároveň je pod CCP rezidentní nějaký další modul. Nevejde-li se nahrávaný program do zmenšené TPA, je vypsáno hlášení "No space" a program není zaveden. U původního CCP došlo k destruktivnímu přepsání rezidentního modulu.

3.2 Zavedení systému

CP/M maximálně využívá standardního hardwaru Sharp. Proto i bootování systému probíhá standardně pomocí vestavěné ROM. Po zapnutí počítače a vložení diskety do jednotky A: ROM přečte boot sektor (strana 1, stopa 0, sektor 1) a podle údajů v něm natahne vlastní booter. Tento program o délce asi 1 KB je umístěn na disku na straně 1, stopě 0 a sektorech 2 až 5. Po zavedení v módu MZ-700 od obvyklé adresy 1200H se připne RAM od 0000H, booter se přemístí od této adresy a je mu odevzdáno řízení. Následuje inicializace všech I/O obvodů (GDG je přepnuto do módu MZ-800, single read a single write, inicializují se scroll registry) a jsou vyslána řídicí slova do PPI, PIO a PIT. PPI a PIO pracují v obvyklém módu, pouze brána A obvodu PIO se používá pro přerušování CPU v módu IM 2 od signálu VBLN (50 Hz). PIT kanál nula generuje zvukové signály (pípání kláves a BEL) a druhé dva kanály zapojené v serii dělí řádkový kmitočet obrazu pro generování reálného času. Po inicializaci obvodů booter zkoumá záznam na disketě (zda je jedno- nebo oboustranná) a podle tohoto zjištění zavede vlastní operační systém nezávisle na tom, je-li mechanika oboustranná či nikoli a předá mu řízení skokem do sestavovacího programu na konci BIOSu. Zde je třeba říci, že to, co se natahuje z disku, není přímo BIOS v té podobě, jak později běží. Booter prohlédne instalační tabulky na začátku BIOSu (ty popisují konkrétní individuální konfiguraci systému) a podle jejich obsahu přemístí závislé části BIOSu na správná místa, čímž teprve vznikne spustitelný BIOS. Zároveň jsou ztraceny rutiny, které nejsou v konkrétní konfiguraci potřebné a spuštěním BIOSu je přepsán i sestavovací program, protože ten je umístěn v diskovém bufferu. Proto např. není možné již spuštěný BIOS disassemblovat, protože již není kompletní. Nelze rovněž provést v BIOSu nějaké změny a zapsat jej zpět na disk, protože by se podruhé nespustil. Po vygenerování individuální verze BIOSu se inicializuje obvod SIO vysláním asi deseti řídicích slov, jejichž obsah je dán požadavky uživatele a lze jej měnit programem SETUP.

Je-li připojen RAM disk, je také inicializován jeho adresář. Do všech příznaků obsazenosti je zapsáno E5H, takže disk je prázdný, ale v případě potřeby lze programem POWER pomocí příkazu RECLAIM obnovit původní obsah RAM disku (zbytek adresáře je zachován). Zálohovaný RAM disk lze také naplnit sekvencí, která zajistí, že BIOS při bootu disk vůbec neinicializuje - tato sekvence je vzestupná řada čísel od 8 do 0FH včetně, zapsaná od adresy 0008H do nulté stránky RAM disku. Vznikne tak vlastně fiktivní položka adresáře, která je-li zapsána pod číslem uživatele 31, je trvalá a systémovými službami přitom nepřístupná. Chceme-li, aby tato položka nezabírala na disku žádné místo (jako fiktivní soubor), musíme do druhé šestnáctice bytů této položky uložit samé nuly. Začátek RAM disku tedy bude vypadat např. takto:

```
Adr: 0000 Data: 1F 00 00 00 00 00 00 00 08 09 0A 0B 0C 0D 0E 0F
      0010      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Nakonec celé bootovací sekvence je vypsán seznam všech

nastavených logických disků i s jejich kapacitami a je předáno řízení do CCP.

Teplý start systému funguje standardním způsobem. Po skoku JP BIOS+3 jsou z disku zavedeny moduly CCP a BDOS a je jim předáno řízení.

Určitou negativní vlastností je nemožnost použít obvyklého CTRL/RESET k teplému restartu CP/M. Je to dáno tím, že po resetu původní ROM inicializuje všechny obvody a BIOS pak již nemůže v jejich registrech obnovit původní hodnoty, protože je jednoduše nezná. Kromě toho inicializaci většiny obvodů z prostorových důvodů neprovádí samotný BIOS, ale předcházející booter. Použití CTRL/RESET tedy vede sice ke skoku do CP/M, ale obvody pracují v nesprávných režimech. Ze stejného důvodu nemohou také uživatelské programy měnit režimy periferních obvodů, protože BIOS je zpět již nepřeprogramuje.

3.3 Obsluha klávesnice

Čtení klávesnice je napsáno znovu v BIOSu, nepoužívá se tedy rutin v ROM. Klávesnice má proto několik dalších funkcí, jako je předávání znaků přes buffer o délce 256 bytů (je umístěn na konci VRAM), opakování již odeslaných příkazů, funkce smazání klávesnicového bufferu, používání funkčních kláves atd. Klávesnice se čte přes přerušeni v módu IM 2 od obvodu PIO (50 Hz). Zároveň s obsluhou klávesnice je zobrazován kurzor (některé programy vyžadují viditelný kurzor i v době, kdy se nečeká na klávesnici). Přerušeni rovněž hlídá, zda se nemá vypnout motor některé mechaniky (při dosažení 250 přerušeni od poslední operace s mechanikou). Stará se i o zobrazování systémového řádku (mód klávesnice, reálný čas a obsah funkčních kláves) a umožňuje pravidelný autorepeat (s tím byly v původní P-CP/M problémy, protože zde nebylo přerušeni využito).

Z klávesnice lze zapsat všechny znaky o kódech 0 až FFH (!), což lze s výhodou použít k psaní češtiny v kódu KOI-8-Cs. Vedlo to k určité komplikaci, protože klávesnice kromě těchto znaků musí vracet i interní řídicí znaky (funkční klávesy, ovládání bufferu apod.) a proto BIOS používá ke čtení klávesnice dvoubytové číslo pro kód klávesy.

Klávesnice je čtena neustále nezávisle na ostatních činnostech počítače. V případě, že běžící program nestačí znaky z klávesnice zpracovávat, klávesnice pracuje do vyrovnávacího klávesnicového bufferu. Je proto možné např. při překladu souboru assemblerem již zadat příkaz pro spuštění překládaného programu, který v době zadání tohoto příkazu ještě vlastně neexistuje.

Jinou funkcí je opakování již zadaného příkazu. Text několika naposled napsaných řádek je totiž i po odeslání klávesou CR uložen v bufferu a jednotlivé příkazy je možno vyvolat klávesou BLANK. Chceme-li např. opakovat poslední zapsaný příkaz, stiskneme BLANK a CR. O více příkazů se dá vracet vícenásobným stiskem BLANK. Pro vymazání nahromaděného textu z bufferu na obrazovce lze s výhodou použít CTRL/X. Klávesnicový buffer je cyklicky uzavřen do kruhu, takže v případě "přejetí" hledaného příkazu se lze k tomuto příkazu zpět vrátit.

Následuje tabulka popisující funkci editačních a jinak využitých kláves. Je mi jasné, že někteří lidé budou po přečtení této tabulky pochybovat o mém zdravém rozumu, nicméně si myslím, že lépe to vymyslet nejde:

Klávesa	Funkce
GRAPH	prefix pro znaky ≥ 128 (mód klávesnice "GRAP"). GRAPH/cokoli generuje znak odpovídající stisklé klávese, avšak má nastaven 7. bit. Lze použít při psaní češtiny, kdy GRAPH/písmeno dává odpovídající český znak (norma KOI-8-Cs). ALPHA CAPS LOCK on/off (indikace jako mód "NORM"/"CAPS")
BREAK	CTRL/C
BLANK	návrat k předcházejícím příkazům (z kláv. bufferu)

SHIFT/BLANK	mazání znaků nahromaděných v klávesnicovém bufferu
"LIBRA"	přepíná rychlý/pomalý scroll
SHIFT/"LIBRA"	přepíná zobrazenou sadu funkčních kláves (F1 - F5, nebo F6 - F10)
INST	BS (pozor, CCP např. používá INST namísto DEL!)
SHIFT/INST	CTRL/T
DEL	DEL
SHIFT/DEL	CTRL/G
KURZOR VLEVO	CTRL/S
SHIFT/K. VLEVO	CTRL/A
KURZOR VPRAVO	CTRL/D
SHIFT/K.VPRAVO	CTRL/F
KURZOR NAHORU	CTRL/E
SHIFT/K.NAHORU	CTRL/R
KURZOR DOLŮ	CTRL/X
SHIFT/K. DOLŮ	CTRL/C
ŠIPKA VLEVO	CTRL/W
ŠIPKA VPRAVO	CTRL/Z
"PI"	_ (underline)
SHIFT/CR	LF

Zapnutý CAPS LOCK se týká pouze písmen, ostatní znaky zůstávají beze změny. SHIFT/písmeno při zapnutém CAPS LOCKu dává odpovídající malé písmeno, tak jak to dělají jiné rozumné počítače. Prefix GRAPH se vztahuje jen na bezprostředně následující klávesu, a to bez ohledu na to, zda je řídicí či nikoli. Zcela zvláštní význam mají klávesy SHIFT/CTRL/BREAK, jejichž současné stisknutí má za následek okamžité opuštění programu bez závislosti na momentální činnosti počítače a skok do operačního systému. Je to výhodná funkce v případě, že program uvízne v nekonečném cyklu, nebo chceme-li např. tisknout na nepřipojené tiskárně. Ostatní klávesy fungují tak jako na jiných počítačích, tzn. CTRL/písmeno dává odpovídající řídicí znak.

3.4 Obsluha obrazovky

Obsluze obrazovky byla věnována veliká pozornost, zvláště co se týče rychlosti obrazovkových operací. Například při výpisu znaku se vůbec nevypočítává adresa v obrazovce z aktuální pozice kurzoru, ale na pointer do obrazovky je využita zvláštní proměnná. Znak se nezobrazuje cyklem, ale přímo sekvencí instrukcí. Blokované přesuny v paměti, potřebné pro rolování kusů obrazovky, nejsou realizovány instrukcí LDIR, nýbrž sekvence šestnácti bezprostředně za sebou následujících instrukcí LDI, resp. LDD. Roluje-li se celá obrazovka, je využito schopností GDG rolovat hardwarově. Podle nastaveného módu rolování (klávesa "LIBRA") se roluje buď po znakovém, nebo grafickém řádku. Tento režim funguje při rolování oběma směry. Součástí výstupního programu je i podprogram pro rychlé mazání řádky, což ve spojení s již zmíněnými přesuny částí obrazovky výrazně zvyšuje komfort při editování textu. Jako určitou výhodou proti původní P-CP/M lze hodnotit i zabudovaný nový generátor znaků, protože nové znaky se podstatně lépe čtou. Je také možné oba typy znaků mezi sebou přepínat. Obyčejné ASCII znaky o kódech 20H až 7EH, BEL, BS, LF a CR jsou zobrazovány obvyklým způsobem. Znaky s nastaveným sedmým bitem jsou použity jednak pro pseudografiku, kreslení rámečků a oken (kódy 80H - BFH) a dále pro české znaky s diakritickými znaménky (C0H - FFH). Seznam ostatních řídicích kódů je v následující tabulce:

Znak	Funkce
ESC (LOWLIGHT (přepnutí na znaky z ROM)
ESC)	HIGHLIGHT (přepnutí na znaky BIOSu - jsou tlustší)
ESC 0	INVERS VIDEO
ESC 1	NORMAL VIDEO
ESC * nebo CTRL/Z	CLS+HOME
ESC T	ERASE TO EOLN
ESC R	DELETE LINE
ESC E	INSERT LINE
ESC = y x	CURSOR LEAD (souřadnice y a x mají offset 20H)
CTRL/W	prefix nastavující 7. bit u následujícího znaku (využívá se v případě, že chceme psát česky z programu, který neumí využívat 7. bit kódu).

BIOS neumí zpracovat znaky TAB, FF a DEL. Řídicí znaky byly zvoleny ve snaze po co největší standardizaci BIOSu a jsou shodné s terminálem Televideo 925, případně i 912 nebo 950.

3.5 Tiskárna a děrná páska

Obsluha tiskárny je řešena přes tiskárenský konektor na zadní straně počítače, připojený přes oddělovače na obvod PIO. Rutina neprovádí žádné znakové konverze. Lze připojit tiskárnu s rozhraním Centronics, IRPR, i jiné, protože podprogram testuje sestupnou i vzestupnou hranu vstupního uvolňovacího signálu.

Na místě driverů pro práci s děrnou páskou je pochopitelně implementována obsluha standardního seriového rozhraní RS-232C. Je využit kanál A obvodu SIO a pomocí programu SETUP lze nastavit všechny potřebné parametry komunikace, lze dokonce zapnout i vypnout handshaking (to původní P-CP/M neuměl). Opět se neprovádějí žádné znakové konverze. BIOS nepočítá s volbou přenosové rychlosti, protože originálních Sharp firemních karet pro seriovou komunikaci je u nás velmi málo a karty vyráběné amatérsky nebo na zakázku mají baudovou rychlost nastavitelnou pomocí miniaturních přepínačů. Jestliže někdo vlastní originální stykovou kartu, může si rychlost přenosu nastavit z nějakého programu (např. TURBO Pascal), protože BIOS s porty pro nastavení rychlosti nic nedělá. Kanál B obvodu SIO je naprogramován do stejného režimu jako kanál A, takže uživatel jej může obsluhovat ve svém programu.

4. Diskové operace

Přístup na disk je základní vlastností každého operačního systému. I když obsluha disku je probádané a často omílané téma, stále jsou ještě detaily, ve kterých se dá komunikace s diskem vylepšit. Popisovaný BIOS např. obchází řadič a vše, co může, dělá raději softwarově. Rutiny jsou sice delší a méně průhledné, ale je tím umožněno veškeré operace časovat přímo programem, nezávisle na tom, co dělá hardwarově řadič. Mimo jiné to až dvakrát urychluje práci disku, navíc všechny časové konstanty obsluhy disku si může uživatel nastavit podle vlastností svých vlastních mechanik (program SETUP). Podmínkou správné funkce jsou jen konstantní hodiny procesoru.

Konkrétně: Řadič Western Digital, obsluhuje-li mechaniku 3 1/2 nebo 5 1/4 '', umí jen čtyři časy kroku diskové hlavy, a to 6, 12, 20 a 30 ms. Při použití osmdesátistopé mechaniky (to je dnes standard) je doba SEEKu (nastavení hlavy na stopu) důležitým faktorem, který má vliv na rychlost disku. Vzhledem k tomu, že naprostá většina dnes používaných mechanik má dobu kroku hlavy (step time) pouhé tři milisekundy, je softwarové obcházení řadiče téměř nezbytné. Při požadavku na SEEK se programově generují jednotlivé STEP pulsy, mezi kterými program běží v časové smyčce, na jejímž konci volá do řadiče příkaz "Force Interrupt", aby řadič byl znovu schopen komunikace.

Jiné urychlení se dosahuje v případě chyby adresace na disku. Obvykle se výskyt této chyby ošetřuje tak, že se s hlavou odjede na stopu 0 (HOME) a poté se znovu provádí SEEK. Uvážíme-li, že se celá procedura může opakovat několikrát, představuje to několikasekundové zpoždění. Přitom chyba adresace nejčastěji vznikne tím, že při výměně diskety se hlava mírně posune mimo svou polohu na stopě a pak dochází ke čtení z prostoru mezi dvěma stopami. Náš BIOS ošetřuje vzniklou chybu takto:

- 1) Provede se jeden krok hlavou (nachází-li se hlava na vnějších stopách, krokuje se směrem dovnitř, v opačném případě směrem ven), čímž se hlava dostane přesně na některou stopu. Pak se provede SEEK na správnou stopu a čte se znovu.
- 2) Je-li předchozí krok neúspěšný, došlo zřejmě k tomu, že hlava se nachází na jiné stopě než si myslíme. Proto je do řadiče volán příkaz "Read address", koriguje se číslo aktuální stopy a čte se znovu.
- 3) Při neúspěchu se nejspíš hlava nachází mimo aktivní plochu diskety. Teprve v tom případě se provádí HOME (opět softwarově) a znovu se čte.

Dojde-li k chybě na disku, pokouší se BIOS provést čtení nebo zápis pětkrát, než hlásí chybu. Chybové hlášení se zobrazuje v systémovém řádku, takže nenarušuje výpisy programu. Uživatel má možnost výběru "Retry/Abort/Ignore", význam čehož je následující:

Retry - opětovný pokus provést danou operaci. Je výhodné třeba při pokusu o zápis na chráněnou disketu. Stačí

disketu vyjmout, přesunout ochranný segment, disketu vrátit zpět do mechaniky a stisknout "R". Zápis se provede.

Abort - návrat ze služby BIOSu s chybovým příznakem. Jestliže si chybu neošetřuje sám program, předá se řízení operačnímu systému.

Ignore - návrat z BIOSu, jako kdyby k chybě nedošlo. Při pokusu o zápis není dobré tuto variantu používat, neboť data v diskovém bufferu jsou označena za zapsaná. I použití při pokusu o čtení je třeba si rozvážit, abychom např. nespustili chybně přečtený program apod. Mohlo by to být osudné pro Vaše diskety.

BIOS správně reaguje i na stav, kdy např. není v mechanice disketa. Je až neuvěřitelné, kolik různých BIOSů zůstane "viset", nedáme-li do mechaniky disketu. Naše rutiny pro čtení sektoru a adresní značky mají časové omezení asi 1.2 sekundy, a když do této doby řadič neukončí příkaz, je tento ukončen násilně.

Pozornost byla věnována také tomu, aby diskové rutiny pokud možno co nejméně zakazovaly přerušování, protože přes něj běží klávesnice. Ideální by bylo, kdyby klávesnice byla čtena nepřetržitě, což by ovšem znamenalo mít přerušování neustále povolené a z disku bychom toho asi mnoho nepřečetli. Přerušování se proto zakazuje pouze na nejnutnější dobu, tj. při vlastním čtení nebo zápisu dat. Ostatní dobu (při operacích HOME, SEEK, čekání na rozběh motoru apod.) je stále povoleno, i když to má za následek netypické zvuky linoucí se z disketových mechanik. Na jejich životnost to však nemá vliv, nepravidelné řízení by mechanikám nemělo vadit.

Dokonalý je i způsob kontroly adresace na disku - je totiž na rozdíl od jiných CP/M (i původní P-CP/M) úplný. Je to sice náhoda, ale i mně se stalo, že vodič vybírající stranu diskety ztratil kontakt a následně došlo k přepsání diskety. To se v novém CP/M nemůže stát! BIOS pozná i to, když mu vytáhnete nějaký vodič z řadiče.

4.1 Formát záznamu na disketových mechanikách

Je jasné, že v dnešní situaci, kdy člověk skoro několikrát denně zakopává o IBM PC, není možné používat disketu s jiným formátem než pécéčkovým. Fyzická stopa diskety tudíž obsahuje devět sektorů o délce 512 bytů. Pokud jde o ostatní parametry stopy (délky všech čtyř Gapů, adresní značky atd.), souhlasí s doporučením IBM a Western Digital. Při psaní programu FORMAT byly tabulky pro zápis stopy převzaty z přenosného počítače IBM PC s mechanikami 3 1/2'' 720 KB.

Z originální pécéčkové diskety byl převzat i PC boot sektor, který je při kopírování systému CP/M na Sharpu též přenesen na novou disketu. To umožňuje přímé čtení disket ze Sharpa na IBM PC, protože Sharpovská disketa obsahuje tabulku s popisem parametrů diskety. Hodnoty v boot sektoru se vztahují k disketě 3 1/2'' 720 KB, protože tyto diskety jsou dnes již u PC častější než dříve montované 5 1/4''. Pokud by někdo potřeboval číst Sharpovské diskety na PC s "pětačtvrtkami", musel by tabulky v pécéčkovém boot sektoru přepsat tak, aby popisovaly tuto disketu.

Z hlediska optimalizace přístupových dob jsou samozřejmě fyzické sektory diskety přečísleny. Aby však diskety byly čitelné na jiných počítačích, přečíslení neprovádí BIOS pomocí tabulek, ale sektory jsou přímo přeházeny na disketě, a sice s parametrem dvě. Stopa tedy obsahuje sektory v tomto pořadí:

1, 6, 2, 7, 3, 8, 4, 9, 5.

Adresní značky jednotlivých sektorů jsou samozřejmě přeházeny zároveň se sektory, takže disketa se navenek jeví jako nepřechíslená.

Aby to však nebylo tak jednoduché, autor CP/M byl požádán, aby bylo možné používat i jednostranné mechaniky. Tím bylo nutno vytvořit dva různé formáty, vzhledem k nutnosti původní Sharp boot stopy. Logické rozdělení disket těchto formátů je v následujících tabulkách.

Oboustranný záznam

IBM PC boot sektor	strana 0, stopa 0, sektor 1
IBM PC FAT (vynechané místo)	strana 0, stopa 0, sektor 2-5
Začátek Sharp BIOSu	strana 0, stopa 0, sektor 6-9
Sharp boot sektor (formát ROMky)	strana 1, stopa 0, sektor 1
CP/M booter	strana 1, stopa 0, sektor 2-5
Zbytek Sharp boot stopy	strana 1, stopa 0, sektor 6-16
Pozor! Sharpovská boot stopa obsahuje šestnáct sektorů o délce 256 bytů. I když je to z hlediska IBM PC nestandardní, tento formát je nutný, protože jiný formát Sharp ROM nepřechte a CP/M by tudíž pomocí standardní ROM nešel zavést.	
Konec Sharp BIOSu	strana 0, stopa 1, sektor 1-7
Sharp CCP a BDOS	strana 0, stopa 1, sektor 8 až strana 1, stopa 1, sektor 9
CP/M adresář	strana 0, stopa 2, sektor 1-8
Začátek prostoru pro soubory	strana 0, stopa 2, sektor 9

Jednostranný záznam

Sharp boot sektor	stopa 0, sektor 1
CP/M booter	stopa 0, sektor 2-5
Zbytek Sharp boot stopy	stopa 0, sektor 6-16

Boot stopa má stejný formát jako v předešlém případě.

IBM PC boot sektor	stopa 1, sektor 1
IBM PC FAT (vynechané místo)	stopa 1, sektor 2-5
Sharp BIOS	stopa 1, sektor 6 až stopa 2, sektor 7
Sharp CCP a BDOS	stopa 2, sektor 8 až stopa 3, sektor 9
CP/M adresář	stopa 4, sektor 1-8
Začátek prostoru pro soubory	stopa 4, sektor 9

Logické parametry CP/M diskety jsou tyto:

Kapacita diskety:	720 / 360 / 180 KB (podle počtu stop a stran)
Místo pro soubory:	698 / 338 / 158 KB
Počet stop:	160 / 80 / 40
Počet sektorů na stopě:	36
Velikost adresáře:	4 KB
Max. počet souborů na disku:	128
Velikost alokačního bloku:	2 KB
Počet systémových stop:	4

Logické parametry CP/M RAM disku:

Kapacita disku:	nastavitelná v rozmezí 0 až 512 KB po 16 KB
Počet stop:	0 až 16 podle kapacity
Počet sektorů na stopě:	256
Velikost adresáře:	4 KB
Max. počet souborů na disku:	128
Velikost alokačního bloku:	2 KB
Počet systémových stop:	0

4.2 Výběr mechaniky

Používáme-li více mechanik, je do BIOSu volaná služba výběru mechaniky, se kterou se bude dále pracovat. Tato služba spočívá především ve zjištění, zda je požadovaná mechanika logicky připojena (počet mechanik se nastavuje pomocí SETUP). V případě pokusu vybrat neexistující mechaniku se do systémového řádku vypíše hlášení

```
Disk select error: Abort/Ignore
```

příčemž volbou "A" BIOS vrací příznak chyby (tj. obvykle vypadneme do DOSu), volba "I" způsobí návrat do volajícího programu s tím, že k chybě nedošlo, avšak zůstal vybrán naposled nastavený disk.

Při úspěšném výběru mechaniky BIOS vrací pointer na tabulku diskových parametrů. Těchto tabulek je celkem pět, tj. každá mechanika má tabulku svou. V případě potřeby je proto možné parametry libovolné mechaniky změnit, aniž by to mělo vliv na parametry mechanik ostatních (program DISKDEF).

Jednotlivé disketové mechaniky adresuje BIOS postupně A:, B:, C: až případně D:, disk E: je RAM disk. Výhledově se počítá s připojením ještě dalších dvou mechanik, a to o rozměru 8'', které se budou značit F: a G:.

Používáme-li systém pouze s jednou mechanikou, tato mechanika má význam dvou logických jednotek (A: a B:). BIOS si uchovává v paměti číslo naposled používané mechaniky a je-li požadován přístup na mechaniku druhou, vypíše se do systémového řádku hlášení

```
Select disk d: Push any key.
```

což uživateli umožňuje pracovat na jedné mechanice tak, jako by měl mechaniky dvě.

4.3 Operace čtení a zápisu

Čtení a zápis sektoru na disk se provádí standardně, používá se jeden diskový buffer pro jeden fyzický sektor (512 bytů). Je použit obvyklý blok/deblokovací algoritmus (viz dále). Čtení a zápis obstarávají rutiny s několikastupňovým hierarchickým uspořádáním. Nejvyšší rutiny, do kterých se řízení dostává při každém volání služby, zajišťují převod mezi logickými a fyzickými sektory a podle potřeby volají rutiny nižší. To jsou zejména rutiny pro přímé čtení nebo zápis konkrétního fyzického sektoru, které v případě chyby vypisují také chybové hlášení

Read/write error: Retry/Abort/Ignore

význam čehož je zřejmý. Tyto rutiny volají hierarchicky ještě nižší stupeň, který ošetřuje a napravuje vzniklé chyby a stará se o přímý transport dat mezi diskovým bufferem a diskem bez ošetření vzniklých chyb.

Je užitečné vědět, co BIOS dělá s diskovým bufferem, dojde-li k chybě. Volba "Retry" je jasná, vše se děje znovu. Při volbě "Abort" nebo "Ignore" dojde k tomu, že je nulován příznak zápisu do bufferu, a data v bufferu jsou označena za platná. Proto v případě pokusu o nové čtení vadného sektoru BIOS chybu již nehlásí (je tím umožněno se alespoň pokusit načíst část sektoru). Při zápisu to má ovšem za následek, že se buffer na disk vůbec nezapíše a proto je třeba si použití varianty "Abort" nebo "Ignore" rozvážit.

4.4 Práce s diskovým bufferem

To není nic jiného než blokovací a deblokovací algoritmus. BIOS musí zajistit převod logických sektorů, které používá jádro CP/M, o délce 128 bytů, na fyzické sektory, které mají 512 bytů. Algoritmus funguje takto:

Při požadavku na čtení se testuje, zda sektor z požadovaného fyzického sektoru se již nenachází v bufferu. Když ano, je předán volajícímu programu a s disketovou mechanikou se nic nedělá. Jestliže buffer obsahuje jiný sektor, je třeba buffer napřed naplnit daty z disku. Před vlastním načtením se samozřejmě ještě vyhodnotí stav příznaku zápisu do bufferu, a je-li příznak nastaven, předchozí obsah bufferu se uklidí na disketu. Výjimku tvoří čtení logického sektoru, který je ve fyzickém sektoru umístěn jako první. Tehdy dochází k fyzickému čtení z disku vždy, bez ohledu na to, zda sektor již v bufferu je nebo není. Tím je zajištěno, že při výměně diskety jsou zapomenuta stará data z předcházející diskety. (Každá operace prováděná po výměně diskety nejdříve přistupuje do adresáře, a tudíž čte nejprve první logický sektor z fyzického).

Zápis na disk je mírně komplikovanější. BIOS především vyhodnocuje vstupní parametr služby WRITE (v registru C). Tento parametr má následující význam:

Nula:	Běžný zápis sektoru.
Jedna:	Zápis do adresáře.
Dvě:	Zápis do prvního sektoru nově alokovaného alokačního bloku.

Jestliže se při zápisu na disk nachází odpovídající fyzický sektor v bufferu, je logický sektor přemístěn do bufferu a je nastaven příznak požadavku na zápis. Jestliže je buffer obsazen jiným sektorem, musí se buffer přiřadit novému sektoru a provést přesun tohoto sektoru do bufferu. Před přesunem je starý obsah bufferu označen za neplatný, a v případě nastaveného příznaku zápisu do bufferu se ještě vlastní buffer odklízí na disketu (operace vyčištění bufferu je stejná jako při čtení sektoru). Každý zápis do bufferu má za následek současné nastavení příznaku zápisu, při fyzickém zápisu bufferu na disketu je příznak nulován.

Při překopírování logického sektoru do bufferu je třeba rozlišovat, zda se jedná o zápis do adresáře, o zápis sekvenční nebo náhodný. Rozdíl je totiž v tom, že při přístupu do adresáře je třeba před přesunem sektoru do bufferu tento buffer inicializovat daty z disku, aby nedošlo ke ztrátě ostatních logických sektorů v sektoru fyzickém. Při zápisu do adresáře se také ihned potřebné změny v bufferu zaznamenávají na disk a příznak zápisu je nulován (zápis do adresáře je mimo jiné průvodním znakem uzavírání souboru, což může předcházet výměně diskety). Podobná situace je při náhodném zápisu na disk, rovněž je nutno buffer předečist z diskety. Rozdíl je jen v tom, že fyzický zápis bufferu není proveden ihned, ale až je-li to nutné

(tj. když potřebujeme buffer uvolnit pro jiný sektor).

Zcela speciální případ tvoří sekvenční zápis sektorů, např. při zápisu sekvenčního souboru. Bohužel BDOS při volání zápisu do BIOSu neinformuje, zda požadovaný zápis má charakter sekvenční nebo náhodný. Kdyby se totiž sekvenční zápis prováděl jako náhodný (s předečítáním bufferu z diskety před přesunem logického sektoru), zápis souborů by byl velmi pomalý (zaznamenal by se nejvýš jeden sektor za otáčku diskety). Proto BIOS používá při zápisu příznak sekvenčního zápisu. O co vlastně jde. Je-li příznak sekvenčního zápisu nastaven, neprovádí se při požadavku na zápis předečítání bufferu z diskety. To umožňuje využívat maximální možné záznamové rychlosti mechaniky, která je prakticky totožná s rychlostí čtení. Jde jen o to, kdy příznak sekvenčního zápisu nastavit a kdy nulovat:

- Je-li při volání zápisu sektoru do BIOSu vstupní parametr rovný dvěma, jedná se o zápis prvního sektoru nějakého alokačního bloku. To má vždy za následek nastavení příznaku sekvenčního zápisu.
- Příznak je zrušen vždy, je-li nějak porušen předpokládaný průběh sekvenčního zápisu. BIOS vždy při zápisu vyhodnocuje, zda fyzické umístění sektoru, který se má zapsat, je následující za umístěním sektoru, který se zapisoval posledně (je nutno si pamatovat číslo disku, stopy a sektoru minulé operace i současné operace). Jestliže pořadí těchto dvou sektorů neodpovídá sousední poloze sektorů na disku, je příznak sekvenčního zápisu nulován.
- Příznak je nulován vždy po dosažení konce aktuálního alokačního bloku. V našem případě je použit speciální čítač, který počítá všechny zapsané sektory od okamžiku, kdy byl vstupní parametr zápisové služby roven dvěma (začátek alokačního bloku). Dosáhne-li tento čítač hodnoty 16 (= 2KB), dojde k nulování příznaku sekvenčního zápisu.
- Příznak sekvenčního zápisu je nulován každou službou čtení. Zároveň je čítač zapisovaných sektorů nastaven na koncovou hodnotu.

5. Základní programové příslušenství

K systému CP/M pro Sharp MZ-800 byly napsány další programy, tvořící jeho nezbytný doplněk. Jsou to hlavně programy FORMAT a COPYSYS pro inicializaci disket, programy SETUP pro modifikování hardwarové konfigurace a DISKDEF pro nastavení logických parametrů disketových jednotek, programy DEFKEY a KEYB pro definování klávesnice a programy MLOAD a MSAVE pro zálohování programů na magnetofonových kazetách.

Bohužel popis dalších uživatelských programů, které pracují pod systémem, jako jsou překladače různých programovacích jazyků, editory a databáze, není do tohoto textu zahrnut. Znamenalo by to totiž napsat knihu o několika stech stránkách. Zájemci se proto musí obrátit na jinou dostupnou literaturu popisující uživatelské programy CP/M. Poměrně rozsáhlý sortiment takové literatury např. nabízí JZD Agrokombinát Slušovice jako příslušenství svých počítačů TNS, nebo podnik Kancelářské stroje k počítačům Robotron.

Je důležité se zmínit o jedné věci. Některé systémové programy potřebují pro svou funkci znát některé systémové parametry (např. program FORMAT potřebuje vědět, zda je mechanika jedno- nebo dvoustranná, program TIME potřebuje zjistit, kolik je hodin apod.). Protože služby pro zjištění takovýchto informací nejsou v standardní CP/M dostupné, řeší to naše CP/M jakousi tabulkou, ve které jsou tyto informace uvedeny a programy mohou údaje z této tabulky číst. Tabulka vypadá následovně:

Adresa	Hodnota, význam
EA33H	0-15, barva pozadí obrazovky
EA34H	0-15, barva tisku na obrazovku
EA35H	0-255, délka pípnutí klávesnice v milisekundách
EA36H	0-3, číslo poslední disketové jednotky v systému
EA37H	0-7, fyzické parametry disketové mechaniky A: takto: bit 2 - jeho nastavení signalizuje dvojitý krok disketové hlavy (jestliže simulujeme čtyřicetistopou mechaniku na osmdesátistopé). bit 1 - nula indikuje jednostrannou mechaniku, jednička dvoustrannou. bit 0 - při nule má mechanika 40 stop, nastavený bit informuje o osmdesátistopé mechanice.
EA38H	0-7, fyzické parametry jednotky B:. Význam stejný jako v předešlém případě.
EA39H	0-7, parametry jednotky C:.
EA3AH	0-7, parametry jednotky D:.
EA3B,3CH	0-9999, doba rozběhu disketových mechanik v milisekundách.
EA3DH	0-99, doba ustálení diskové hlavy na stopě (Settling time).
EA3EH	0-99, doba kroku diskové hlavy v milisekundách
EA3FH	0-32, velikost RAM disku v jednotkách 16 KB EA40H,41H,42H reálný systémový čas v sekundách
EA43H	0-1, příznak přerušeni (zakázáno/povoleno) při diskových operacích.

EA44H a výše řetězec bytů, kterými se při spuštění systému programuje obvod SIO.

Vzhledem k tomu, že některé z těchto parametrů jsou potřebné jen při spuštění systému, je jejich místo při běhu systému využito takto:

Adresa Hodnota, význam

EA33H 0-255, doba v jednotkách 20 ms do vypnutí pohonných motorů disketových mechanik.

EA34H 0-2, vstupní parametr služby WRITE.

EA44H a výše řetězec, kterým se při WBOOT inicializuje oblast systémových parametrů.

5.1 Program FORMAT

Tento program slouží k formátování disket použitelných pak pod CP/M. Spuštění programu se provede prostřednictvím CCP napsáním jeho názvu (FORMAT) a odesláním. Program se zeptá:

Type disk drive to format or CR to reboot:

na což lze odpovědět buď zadáním kódu mechaniky A, B, C, nebo D, nebo stlačením klávesy CR se vrátit zpět do CCP.

Po zadání kódu jednotky se vypíše:

Push a key when ready..

Poté se stiskem klávesy odstartuje vlastní formátování. Celá činnost má dvě fáze:

- Zjištění, v jakém formátu se bude formátovat. Za tím účelem program přečte tabulky popisující fyzické parametry připojených mechanik a podle toho zvolí příslušný formát diskety. Jestliže si přejeme naformátovat disketu v jiném formátu, napřed pomocí programu SETUP nastavíme parametry jednotky a pak formátujeme. Formátovat lze jedno- i oboustranně, 40 nebo 80 stop na straně a dokonce s jednoduchým nebo dvojitým krokem hlavy. Je třeba pouze dát pozor na to, abychom při volbě dvojitého kroku zároveň nastavili čtyřicetistopou mechaniku!
- Vlastní formátování diskety. Jsou zaznamenány prázdné sektory o obsahu všech bytů E5H.
- Verifikace. Program se pokouší přes standardní služby BIOSu přečíst všechny sektory na disketě. Pozor! Vznikne-li na disketě chyba, nedojde k jejímu opravení, nýbrž je vypsáno hlášení "Sorry, permanent error occured." a formátování končí neúspěchem. V dnešní době kvalitních 3 1/2" disketových mechanik je chyba na disketě vyloučena, takže k této eventualitě prakticky vůbec nemůže dojít. Proto autor programu FORMAT upustil od procedury vyřazení vadných sektorů z použitelného místa na disketě.

Po úspěšném formátování se vypíše:

Format complete.

s následným

Type disk drive to format or CR to reboot:

a celá činnost se opakuje.

Při špatném zadání některého parametru se program na tento dotazuje znovu. Činnost programu lze v kterémkoli okamžiku přerušit stiskem kláves SHIFT/CTRL/BREAK. Poslední vyvinutá verze programu FORMAT nese označení FORMAT 1.1.

5.2 Program COPYSYS

Formátováním diskety smažeme její veškerý obsah. To znamená, že pokud chceme tuto disketu používat mimo jiné i k zavádění operačního systému, musíme na ni zapsat operační systém. (Pokoušíme-li se znovuzavádět CCP a BDOS z diskety, kde tyto moduly nejsou, BIOS vypisuje hlášení

Boot error

do systémové řádky). Překopírování systému se provede takto:

- Vložit původní systémovou disketu do jednotky A: a zavolat program COPYSYS. Po jeho spuštění dojde k načtení obsahu systémových stop z disku do paměti.
- Program se dotáže

Type disk drive to write or CR to reboot:

na což lze odpovědět buď kódem jednotky, na kterou chceme systém překopírovat, nebo stiskem CR opustit COPYSYS.

- Zadáním kódu jednotky A, B, C, nebo D dojde k zápisu systému z paměti do systémových stop zvolené diskety.

Program COPYSYS provádí fyzické překopírování obsahu systémových stop. To znamená, že se překopíruje i navolená konfigurace. Není proto nutné programem SETUP obsah nové diskety modifikovat.

Je samozřejmé, že má-li jednotka, na kterou provádíme zápis, jiné parametry než jednotka systémová, program COPYSYS automaticky provede konverzi obsahu systémových stop tak, aby záznam systému na nové disketě odpovídal fyzickým parametrům této diskety. Je třeba to mít na pozoru, protože pak např. při vložení této diskety do systémové jednotky nedojde ke správnému načtení systémových stop.

5.3 Program COPYDISK

Potřebujeme-li zhotovit kopii celé diskety, je nevýhodné kopírovat po jednotlivých souborech, tak jak to dělají běžné kopírovací programy (PIP, POWER) protože je to příliš zdlouhavé. Proto byl pro tento účel napsán zvláštní program, který provede rychlé překopírování diskety, a to způsobem fyzického kopírování. Program používá celou využitelnou paměť TPA jako buffer pro kopírovaná data.

Zvláštností programu je, že kopírování není prováděno přímým ovládním řadiče disketových jednotek, nýbrž voláním standardních služeb operačního systému pomocí vstupního vektoru BIOSu. Program COPYDISK si přitom zcela regulérně přečte obsahy diskových tabulek a podle toho dále pracuje, takže lze kopírovat diskety s prakticky libovolným formátem. Jediným omezením pochopitelně je, že obě jednotky musí mít stejné fyzické i logické parametry. Není-li tomu tak, program vypíše hlášení

Disks are different.

čímž kopírování končí. Díky popsané metodě kopírování lze s výhodou program COPYDISK používat i na jiných počítačích s procesorem Z80 (je systémově nezávislý). To bylo úspěšně vyzkoušeno na počítači TNS.

Program COPYDISK se volá příkazem

COPYDISK DestDrive=SourceDrive

Např. pro překopírování diskety A: na C: napíšeme

COPYDISK C:=A:

Zavoláme-li program pouhým zapsáním

COPYDISK

dojde ke kopírování B:=A:.

Program COPYDISK umí kopírovat diskety i na jediné mechanice, s tím, že vždy při naplnění nebo vyprázdnění bufferu si požádá o výměnu diskety. Podmínkou správné funkce při kopírování na jedné jednotce je jednoduchý blok/deblokovací algoritmus, používající jediný buffer pro fyzický sektor diskety (tuto podmínku splňuje naprostá většina počítačů).

5.4 Program SETUP

To je jeden ze základních programů nové CP/M. Umožňuje totiž velmi variabilně měnit vlastnosti systému, především fyzické parametry připojených disketových mechanik, kapacitu instalovaného RAM disku a parametry seriového rozhraní RS-232C. Při zavolání programu si tento nejprve překopíruje parametry stávajícího systému ze systémových stop disku A: (proto si do mechaniky A: dejte disketu, která má sloužit jako výchozí) do paměti, kde pak provádí všechny změny. Při opuštění programu pak dojde ke zpětnému zapsání nových parametrů z paměti na disk A:, je-li to požadováno. Vystoupíme-li z programu bez zápisu na disk, nachází se systém na disketě v tom stavu, v jakém byl před spuštěním SETUPu.

Popis ovládání programu SETUP zde není uveden, protože jeho názornost to ani nepotřebuje (vše se vybírá pomocí menu). Zajímavostí programu je, že byl celý napsán v TURBO Pascalu, tj. bez použití jediné assemblerové instrukce (také to demonstruje, jaké možnosti Turbo Pascal má).

5.5 Program DISKDEF

Tak jako pomocí programu SETUP se daly nastavit fyzické parametry připojených disketových mechanik (počet stop, stran, jednoduchý nebo dvojitý krok hlavy, veškeré časování diskových operací), program DISKDEF slouží k nastavení logických parametrů diskety (velikost alokačního bloku, velikost adresáře, počet systémových stop, počet sektorů na stopě) což je velmi výhodné, potřebujeme-li pracovat s disketou z jiného počítače. V naprosté většině případů, kdy si od někoho disketu půjčíme, vystačíme s programy SETUP a DISKDEF, abychom tuto disketu správně přečetli a také na ni mohli psát. Předpokládá to pouze vědět, jak vlastně tato disketa vypadá (ale i to se dá zjistit).

Jinak o programu DISKDEF platí totéž, co bylo řečeno výše o SETUPu. Ovládání je snadné, a celý program je napsán v Pascalu. Pro zajímavost je na systémové disketě přiložen i zdrojový program (DISKDEF.PAS). Proto by DISKDEF měl být ve zdrojovém tvaru přenositelný i na jiné počítače (nebylo zkoušeno).

5.6 Program DEFKEY

Tento program se používá k definici znakových řetězců, které jsou generovány klávesnicí při stisku funkčních kláves F1 až F10. Práce s programem je jednoduchá - zavoláme jej

```
DEFKEY <Key>=<Text>,<Key>=<Text>,...
```

Např. chceme na klávesu F1 nadefinovat příkaz ERA B:*.REL a na F2 PIP B:=A:*.REL. Provedeme to

```
DEFKEY F1=ERA B:*.REL,F2=PIP B:=*.*           (a CR).
```

Kódy kláves (F1 apod.) lze psát i zkráceně, např.

```
DEFKEY 1=ERA B:*.REL,2=PIP B:=*.*
```

Na každou klávesu lze nadefinovat max. 11 znaků, v případě delšího řetězce jsou přebytečné znaky ignorovány. Řetězce mohou obsahovat libovolné znaky v rozsahu kódů 00H až 7EH (nelze nadefinovat české znaky). Řídící znaky se zadávají jako dvojznaky "^" a příslušné písmeno. Např. při definování příkazu DIR, který se má ihned provést, napíšeme

```
DIR^M
```

(^M je kód klávesy CR).

Definování obsahu kláves se provádí přepsáním původních řetězců v oblasti VRAM na adresách BE80H až BEFFH, kterou využívá BIOS. Dále uvádíme obsah této oblasti, jak vypadá po zavedení systému:

```
org      BE80H

DefKeys: defm    4,'DIR          '
          defm    5,'TYPE        '
          defm    4,'ERA          '
          defm    5,'USER        '
          defm    7,'SUBMIT       '
          defm    6,'FILES        '
          defm    7,'FORMAT',0Dh,' '
          defm    8,'COPYSYS',0Dh,' '
          defm    9,'COPYDISK',0Dh,' '
          defm    7,'DEFKEY       '
```

5.7 Program KEYB

Autor CP/M byl požádán, aby na CP/M šlo psát také na klávesnici, podobné psacímu stroji. Tak vznikl program KEYB. Volá se buď

KEYB CS

nebo

KEYB GR

První volání (s parametrem CS) zabezpečí předefinování tabulek klávesnice BIOSu na klávesnici podobnou psacímu stroji. Na horní řadě kláves jsou pak umístěny malé znaky české abecedy, číslice zapisujeme se současným stiskem SHIFT. Původní "šiftované" znaky jsou umístěny na klávesy F6 až F10 (jejich původní obsah je pak nedostupný) a na některé klávesy hlavní klávesnice ("LIBRA", SHIFT/"LIBRA", SHIFT/"/" a SHIFT/"?").

Volání KEYB GR předefinuje tabulky klávesnice zpět na původní tvar, bez českých znaků.

5.8 Programy CLS, EJECT a BOOT

Jedná se o kratičké programky s jasnou funkcí. CLS při zavolání smaže obrazovku, EJECT nastaví čistou stránku na tiskárně a BOOT způsobí opuštění CP/M a start rezidentního systému (ROM), což lze použít např. chceme-li vstoupit do Sharp monitoru.

5.9 Program DIAPRINT

Je malý prográmeček v jazyce Pascal. Jak již bylo dříve uvedeno, umí naše CP/M pracovat v plnohodnotné češtině podle normy KOI-8-Cs. Autor například s velkou oblibou používá výborný textový editor WordStar Cs, ve kterém byl mimo jiné napsán i tento text. Je nasnadě, že když náš počítač může psát česky, bylo by dobré, kdyby uměl také česky tisknout. To by měl zajišťovat právě DIAPRINT. Protože autor CP/M měl však k dispozici pouze zapůjčenou tiskárnu D-100 polské výroby, vzniklý program DIAPRINT příliš neodpovídá požadavkům kompatibility s tiskárnami EPSON apod., takže je třeba program DIAPRINT brát jen jako jakýsi impuls k vlastnímu experimentování (i když s tiskárnou D-100 DIAPRINT funguje). Každopádně je třeba přepsat všechny procedury v modulu DIAPRINT.INC, které zabezpečují přepnutí tiskárny do grafického módu a zpět, a také tisk diakritických znamének v grafickém režimu. Je pravděpodobné, že bude třeba mírně pozměnit i vlastní DIAPRINT.PAS. Autor se tímto všem uživatelům omlouvá a přeje slušivý český tisk.

5.10 Programy MLOAD a MSAVE

Tyto programy jsou určeny především pro zálohování (backup) programů a dat zaznamenaných na disketách. Je také možné použít programy MLOAD a MSAVE k přenosu souborů ze systému CP/M do obyčejného Sharpa či naopak. Není problém rovněž přenášet programy mezi Sharpem a jinými počítači, neboť kompatibilní program existuje např. pro Sinclair ZX Spectrum, IBM PC a další.

Základem je struktura záznamu velmi příbuzná se záznamem obyčejného ZX Spectra (jestliže se někdo diví, proč je zvolen formát ZX Spectra a ne Sharp ROMky, mohu vysvětlit, že zvolený záznam je rychlejší, podstatně spolehlivější a úspěch nahrávání nezávisí na fázi přehrávaného signálu). Záznam vytvořený programem MSAVE (čitelný pomocí MLOAD) má následující strukturu:

Magnetofonový pásek

```
-----  
| Blok 0 | Blok 1 | Blok 2 | ... | Blok X (poslední) |  
-----
```

Blok

```
-----  
| Header | Record |  
-----
```

Header

```
-----  
| Leader | Header flag | Data | Header Checksum |  
-----
```

Record

```
-----  
| Leader | Record flag | Data | Record Checksum |  
-----
```

Leader

```
-----  
|+++++----|+++++----|+++++----| ... |+++++++|-----|  
-----
```

```
| | | | | | |
| 1.24 ms | | | 191 us | 210 us |  
|<----->| | | |<----->|<----->|  
| | | | |  
| 5 s pro header, 2 s pro record | |  
|<----->|
```

Data

```
Bit=0      Bit=0              Bit=1              Bit=0
```

```
-----  
|+++++----|+++++----|+++++++-----|+++++----|  
-----
```

```
| | | | | |
| 489 us | | | 977 us | |  
|<----->| | | |<----->|
```



```

-----
|++++| - kladná magnetizace  |----| - záporná magnetizace
-----

```

Formát dat v hlavičce každého bloku je podobný hlavičce, kterou vytváří Spectrum ROM při záznamu CODE. Rozdíly jsou dány nutností použít v CP/M jedenáct znaků pro jméno (8 znaků pro název a 3 pro verzi) a potřebou počítat jednotlivé bloky, neboť programy delší než operační paměť se musí na kazetu ukládat po několika blocích. Tyto bloky mají délku 32 kB a každý z nich má samostatnou hlavičku:

Byte číslo	Význam	Hodnota
0	Typ CODE	03H
1 až 10	Název souboru bez osmého znaku a verze znak (znaky mají zrušený 7. bit)	znak
11 a 12	Délka bloku v bytech	0000-8000H
13 a 14	Startovací adresa	0100H
15	Poslední znak názvu souboru	znak
16	Pořadové číslo bloku počínaje nulou (poslední blok má nastaven 7. bit)	00-FFH

Kontrolní součet bloku (checksum) se určí XORováním všech bytů bloku včetně úvodního (Flag). Programy mají jednoduchou obsluhu. Program MLOAD se zavolá z CCP příkazem

MLOAD <specifikace souboru>

přičemž specifikace souboru může být předznamenána kódem požadované diskové jednotky a je v ní povoleno používat znaků hvězdička "*" a otazník "?". Specifikace souboru musí být uvedena. Program hledá na pásce soubor vyhovující dané specifikaci a když ho najde (přesněji když najde nultý blok tohoto souboru), soubor překopíruje na příslušnou diskovou jednotku. Pokud zde soubor daného jména již existuje, je novým souborem přepsán. Byl-li požadován přenos jen jednoho souboru, program se po vykonání akce vrátí do CCP. V opačném případě budou přeneseny všechny nalezené programy vyhovující specifikaci. Návrat z programu je zprostředkován klávesou SHIFT/BREAK, stisknutou v okamžiku hledání dalšího souboru na pásce.

Hlášení programu MLOAD:

Searching - program hledá na pásce soubor daného jména.
 Ignoring - byl nalezen program jiného jména nebo nalezený blok nemá číslo nula (jsme uprostřed souboru).
 Loading - soubor se nahrává z pásku do oblasti TPA.
 Writing - soubor se kopíruje z oblasti TPA na disk.
 Transferred - soubor byl úspěšně překopírován.
 Operation completed - program úspěšně vykonal požadovanou funkci a vrací se.
 Operation aborted - činnost programu byla předčasně zastavena (klávesa SHIFT/BREAK).
 Tape error - byla detekována chyba při čtení z kazety (ztráta dat nebo chybný kontrolní součet).

Disk write error - chyba při zápisu na disk (není místo na disku nebo v adresáři).

Too long - délka bloku je příliš velká, takže není možno jej nahrát do paměti.

Používání programu MSAVE je analogické programu MLOAD. Po spuštění programu příkazem

MSAVE <specifikace souboru>

program zaznamená na kazetu všechny soubory z vybraného nebo požadovaného disku, které vyhovují uvedené specifikaci. Před začátkem záznamu prvního souboru program napíše

Start tape, then press any key.

a čeká na stisk klávesy, aby obsluha měla čas na spuštění magnetofonu. Mezi jednotlivými soubory jsou asi desetisekundové mezery, aby bylo možno soubory na pásce později snadno vyhledávat. Po zaznamenání všech požadovaných souborů se program vrací zpět do CCP. V případě potřeby lze činnost programu předčasně ukončit klávesou SHIFT/BREAK.

Programy MLOAD a MSAVE lze použít i v režimu rychlého záznamu. Tento režim je výhodný pro uživatele používající výhradně magnetofon a slouží ke krátkodobému zaznamenání pracovních souborů, u kterých nezáleží na spolehlivosti a trvanlivosti záznamu. Příkaz pro záznam na kazetu má tvar

MSAVE <specifikace souboru> <rychlost>

kde parametr "rychlost" může nabývat hodnot 1, 2 nebo 3 a určuje, kolikrát je pořízený záznam rychlejší než klasický ROMkový. Zrychlený záznam není samozřejmě čitelný pomocí Sinclair ROM. Při zpětném nahrávání příkazem MLOAD se rychlost neudává, program si ji nastavuje sám.

Hlášení programu MSAVE:

Reading - soubor se kopíruje z disku do oblasti TPA.

Saving - soubor se přehrává z TPA na kazetu.

Transferred - soubor byl úspěšně zaznamenán.

Start tape, then press any key - program vyzývá obsluhu, aby spustila magnetofon.

Operation completed - program úspěšně provedl záznam všech souborů a vrací se.

File(s) not found - disk neobsahuje žádný soubor, který by vyhovoval zadané specifikaci.

Operation aborted - činnost programu byla předčasně ukončena klávesou SHIFT/BREAK.

Disk read error - chyba při čtení souboru z disku.

Too long - soubor nelze překopírovat z disku do paměti RAM (oblast TPA je menší než 32 kB).

Nejnovější verze programu MLOAD a MSAVE nesou označení 4.0F.

5.11 Program TIME

Tento program slouží k nastavení systémových hodin, jejichž stav se zobrazuje v systémovém řádku a který je také k dispozici všem programům pracujícím pod CP/M. Hodiny pracují ve čtyřadvacetihodinovém cyklu. Po zavolání programu se vypíše

```
Current time is hh:mm:ss
Enter new time:
```

Operátor může dotaz buď odeslat (údaj hodin se nezmění), nebo napsat aktuální čas ve tvaru hh:mm:ss, kde hh představuje stav hodin, mm stav minut a ss sekund (všechno jsou to dvojciferná čísla oddělená od sebe dvojtečkami). Údaj sekund není nutné psát, v tom případě se zapíše čas hh:mm:00.

Program TIME ve skutečnosti nedělá nic jiného, než že přečte stav systémových hodin v sekundách z adres EA40H, EA41H a EA42H, převede jej na odpovídající počet hodin, minut a sekund a to vypíše. Po zápisu nového času převede zpět údaj hodin, minut a sekund na sekundy a toto tříbytové číslo zapíše na adresy EA40H až EA42H.

5.12 Program XDOS

Tento výborný program (napsaný v Turbo Pascalu na IBM PC) umožňuje přenos souborů mezi operačními soubory CP/M a MS-DOS, a to oběma směry. Autor CP/M touto cestou děkuje J. Merclovi, že umožnil modifikaci XDOSu tak, aby fungoval pod naším CP/M a nabídl jej k volnému užívání.

Po přeložení souboru XDOS.PAS pomocí Turbo Pascalu 3.0 pod operačním systémem CP/M vznikne spustitelný program XDOS.COM (překládáme do souboru - to napřed navolíme v menu "Compiler Options, volitelného z hlavního menu). Spuštěním tohoto programu získáme přístup k disketám, zaznamenaným pod MS-DOS.

Jedinou podstatnou nevýhodou programu XDOS je, že umí pracovat pouze s disketami 360 KB, neumožňuje tudíž číst diskety 3 1/2'' 720 KB, které se u IBM PC dosti často vyskytují. Autor CP/M proto požádal J. Mercla, aby XDOS příslušně modifikoval, a doufá, že tato tolik potřebná verze XDOSu brzy spatří světlo světa.

6. Než začnete se systémem pracovat

Předtím, než začnete systém běžně používat, je třeba v něm provést drobné změny, týkající se nastavení některých systémových proměnných (parametry připojených disketových mechanik, velikost RAM disku, parametry seriového rozhraní RS-232C, pípání klávesnice a další). Autor také doporučuje si na systémovém disku založit soubor AUTOEXEC.SUB, který obsahuje inicializaci celé hardwarové konfigurace a může také ze systémové diskety nakopírovat do RAM disku některé základní programy (POWER, WORDSTAR CS apod.). Samočinné spuštění souboru AUTOEXEC.SUB po zapnutí počítače zajistíme tak, že v SETUPu volíme "Auto execute file" a zapíšeme příkaz "SUBMIT AUTOEXEC" (program SUBMIT.COM je součástí systémové diskety).

Především je nutno systém zkopírovat z distribuční diskety na pracovní disketu, z které jej budete zavádět, a původní disketu bezpečně uložit. Nejjednoduššeji to lze provést pomocí programů COPYSYS a pak COPYDISK (popis těchto programů je také součástí tohoto manuálu).

Po uložení distribuční diskety a vložení zkopírované diskety do jednotky A: zavoláme SETUP a nastavíme všechny potřebné parametry. Zvláště důležité je nastavení časování disketových operací. Pro správné nastavení všech tří časů prostudujte prosím manuály svých disketových jednotek, kde jsou tyto časy uvedeny. Jestliže nastavíte časy nesprávně, můžete si být téměř jisti, že za nějaký čas nepůjdou diskety číst, proto nemáte-li manuál od svých mechanik, snažte se konkrétní časy někde vyzvědět. Jedná se především o "Settling time" - doba ustálení hlavy na stopě, a "Start time" - doba rozběhu hlavního motoru. V případě neúspěchu nastavte Settling time na 30 ms a Start time na 1000 ms, což by mělo vyhovovat všem mechanikám, avšak budete se muset spokojit s menší rychlostí Vašeho CP/M. Pokud se týká doby kroku - "Step time", ta se dá najít zkusmo.

6.1 Využití programu z jiného počítače

Nespornou výhodou implementace CP/M na Sharpu je možnost využívat programy a datové soubory vytvořené na jiných počítačích. Je pouze třeba mít tyto programy a data zaznamenané na disketách. Překopírování programů na odpovídající médium lze provést několika způsoby:

1. Nejjednodušší je zřejmě zkopírovat si vše potřebné přímo z diskety na disketu. To bude pravděpodobně zdaleka nejčastější případ.
2. Nemáte-li možnost vypůjčit si někde kompatibilní disketu, budete muset zaznamenat program na zdrojovém počítači na kazetu ve formátu programu MSAVE (varianty programu MSAVE již existují např. pro IBM PC, Sinclair ZX Spectrum apod.).
3. Rychlejší způsob přenosu je po sériové lince, tj. pomocí seriového rozhraní RS-232C. Tuto možnost autor z vlastní zkušenosti vřele doporučuje, protože zmíněné rozhraní je

velice rozšířené, najdeme je prakticky na každém počítači. Kromě toho propojovací kabel má minimum žil, a přenos jde rychle. Přímý přenos po vodiči má jedinou nevýhodu - musíme oba počítače k sobě fyzicky přinést.

Po přenesení programu do Sharpa je nutno některé programy přeinstalovat. Proto nezbyvá než přenést i instalační program a tímto vlastní program instalovat. Pokud původní program neseženeme v nainstalovatelné podobě, je možné, že nebude fungovat. Naopak po správném nainstalování budeme překvapeni, že na Sharpu obvykle program funguje lépe než na původním počítači.

Co se týče přenosnosti programů, je situace příznivá, neboť lze přenést téměř vše. Je nutné se pouze vyhnout programům, které dopředu předpokládají určitý přesně daný formát disku a nezískávají informace o disku čtením systémových tabulek (týká se např. programů určených výhradně pro osmipalcové disky - DOCTOR apod.).

Při přenosu nebo kopírování každého programu je nezbytné spočítat (např. programem POWER) kontrolní součty originálních a překopírovaných programů a porovnat je na shodu. Je až neuvěřitelné, kolik vadných verzí programů koluje mezi lidmi. Chyba v programu může vzniknout jak při přenosu z jiného počítače, tak i při prostém kopírování z disku na disk. Proto radíme mít sadu disket se "zaručenými" programy, a pro práci pořizovat pracovní kopie na jiné médium.

7. Literatura

- [1] Western Digital Corporation, FD 179X-02, Floppy Disk Formatter/Controller Family.
- [2] Sharp Corporation, Service manual, Model MZ-800, MZ-1P16, MZ-1E20, 1984.
- [3] Sharp Corporation, Personal Computer MZ-800, Owner's Manual, 1984.
- [4] NEC Corporation, FD 1036A, 3.5'' Floppy Disk Drive, 1985.
- [5] NEC Corporation, FD 1037A, 3.5'' Floppy Disk Drive, 1987.
- [6] SGS-Ates, Z80 Microcomputer System, 1977.
- [7] Tesla, Katalog polovodičových prvků, 1987.
- [8] Kočí, J.: Řadič pružných disků pro Sharp MZ-800, výrobní dokumentace, 1988.
- [9] Kočí, J.: Karta seriových styků RS-232C pro Sharp MZ-800, výrobní dokumentace, 1988.
- [10] Kočí, J.: RAM disk 256/512 KB pro Sharp MZ-800, výrobní dokumentace, 1988.
- [11] Neznámý autor, dokumentace k řadiči pružných disků pro Sharp MZ-800, 1988. (řadič je odlišný od předchozího)
- [12] Macourek, T.: RAM-disk for Sharp MZ-800, Personal Computer, 1988.
- [13] Neznámý autor, dokumentace k jedné z variant RAM disku pro Sharp MZ-800. (RAM disk je odlišný od předchozího)
- [14] Richta, K., Zajíc, J.: Operační systém CP/M pro mikropočítače, Český výbor elektrotechnické společnosti ČSVTS, 1986.
- [15] Krejča, T.: ZX floppy, připojení floppy disků k mikropočítači, 666. ZO Svazarmu, 1987.
- [16] Navrátil, Z.: OS MIKROS, Příručka pre implementáciu, Kancelářské stroje, 1985.
- [17] Navrátil, Z.: OS MIKROS, Příručka programátora, Kancelářské stroje, 1985.
- [18] Lamač, J.: Operační systém CP/M 2.2, implementace pro ZX Spectrum, 602. Svazarmu, 1988.