
OPERAČNÍ SYSTÉM
AMOS

UŽIVATELSKÁ PŘÍRUČKA AMOS

operační systém AMOS

Obsah:

Případné připomínky k operačnímu systému a k uživatelské příručce zasílejte na adresu:

KKI MFF UK OS AMOS
Malostranské náměstí 25
118 00 P R A H A 1

UPOZORNĚNÍ PRO UŽIVATELE MODULŮ OS AMOS

1/ OS AMOS je schopen přímo spolupracovat s následujícími zásuvnými moduly a periferiemi:

- modulem BASIC 6
- modulem VIDEO 32, VIDEO 64
- modulem GRAFIK
- modulem STAPER
- minigrafem 0507 (Aritma)
- grafickou jednotkou XY 4131 (Laboratorní přístroje)
- jednotkou pružných disků PFD 252 a PFD 151 (Pragotron).

Tyto moduly tedy mohou být současně zasunuty a provozovány spolu s OS AMOS, který také poskytuje programové prostředky pro práci s nimi (viz manuál OS AMOS).

2/ NEZASUNUJTE SOUČASNĚ DO POČÍTAČE IQ 151 MODUL BASIC-G A MODULY OS AMOS, MOHLO BY DOJÍT K JEJICH POŠKOZENÍ !!!!

Přes tuto nepříjemnou skutečnost lze využívat služeb operačního systému AMOS i pro potřeby BASICu G, například pro tvorbu podprogramů ve strojovém kódu, které chceme volat z Basického programu, a to následujícím způsobem:

- pomocí modulu AMOS/Assembler připravit potřebné podprogramy ve strojovém kódu, a nahrát je na magnetofon prostřednictvím monitoru (tj. monitorským nahráváním). V této době samozřejmě nesmí být současně zasunut modul BASIC G.
- poté počítač vypnout, vyměnit modul AMOS za modul BASIC G, a připravené podprogramy nahrát zpět do paměti, opět prostřednictvím monitoru.

3/ Již v roce 1988 se však připravuje výroba nového modulu grafického BASICU, který bude:

- plně kompatibilní s OS AMOS
- rozšířen o možnost práce se soubory OS AMOS i z Basických programů
- zachovávat všechny možnosti stávajícího Basicu G, t.j. bude moci pracovat i samostatně mimo OS AMOS.

4/ U prototypových verzí OS AMOS (verzí 3.2), které byly v malém počtu již dříve vyrobeny, se občas projevovala závada, způsobená spoluprací samotných zásuvných modulů.

V současné době ve verzi 4.1. je tato závada odstraněna programovými prostředky. Má to však důsledek v tom, že logické moduly AMOS/ASSEMBLER a AMOS/PASCAL odlišných verzí mají omezenou možnost spolupráce. Toto omezení se projeví pouze tehdy, jsou-li v počítači zasunuty současně oba logické moduly (různých verzí !!), a spočívá v tom, že v této konfiguraci nelze volat překladač Pascalu.

Logické moduly stejných verzí spolu samozřejmě spolupracují v plném rozsahu.

1. Charakteristika operačního systému AMOS

Za hlavní rysy operačního systému AMOS považujeme tyto jeho vlastnosti:

- Systém může zcela plnohodnotně pracovat v základní konfiguraci počítače, využívá vícenásobně adresového prostoru, který dosud využíván nebyl, a ponechává uživateli skoro celou paměť RAM (zabírá pro systémové účely méně než 2Kbyty).
- Systém je plně souborově orientován a umožňuje tak uživateli práci na kvalitativně vyšší úrovni, běžné u počítačů vyšší třídy než je většina současných československých mikropočítačů. Programátor pracuje v programu s logickými soubory a teprve v okamžiku spouštění programu specifikuje se kterými konkrétními fyzickými soubory má program pracovat.
- Příkazy operačního systému AMOS jsou navrženy tak, aby se na jedné straně začátečníci velmi rychle mohli naučit provádět jednoduché akce a aby na druhé straně umožňovaly specifikovat i velmi složité operace.
- Operační systém AMOS je prvním OS, ve kterém uživatel pracuje stejným způsobem jak s pružnými disky, tak i bez nich.
- Operační systém obsahuje překladač Pascalu. Jde o plnohodnotnou implementaci jazyka, vyhovující mezinárodní normě ISO na úrovni 0, tj. překladač neakceptuje konformní schémata polí. Míra respektování normy je plně srovnatelná s tím, jak normu respektují překladače používané v ČSSR na "velkých" počítačích. Překladač má velmi dobrou chybovou diagnostiku a řadu rozšíření, které umožňují např. přímý přístup do paměti, spolupráci s podprogramy ve strojovém kódu, ovládání souřadnicového zapisovače atd.
- Systém obsahuje překladač assembleru akceptující obvyklý jazyk assembleru mikroprocesoru 8080 se všemi standardními pseudo-instrukcemi. Jeho vazba na ostatní prostředky OS AMOS umožňuje, že se při řešení standardních úloh programátor ani v assembleru nemusí starat o konkrétní adresy dat nebo podprogramů pro vstup a výstup. To má velký význam nejen pro pohodlí programátora a efektivitu jeho práce, ale především umožňuje soustředit se při výuce programování v assembleru na podstatné problémy a omezit množství technických detailů.
- OS AMOS obsahuje výkonný textový editor, který umožňuje vytvářet a opravovat jak paměťové soubory, tak i soubory na vnějších zařízeních. Obrazovka tvoří "okénko", kterým se lze pomoci pohybů kurzoru po souboru libovolně pohybovat. Na obrazovce je v každém okamžiku vidět aktuální stav právě zobrazené části souboru. Během celé práce s editorem je na obrazovce řádek s informacemi o aktuálním stavu editoru (o nastavených modech jeho práce).

Editor může pracovat ve dvou režimech: obrazovkovém a příkazovém.

Obrazovkový slouží k pořizování a prohlížení textových souborů a k jejich jednoduchým opravám, v příkazovém režimu lze provádět i složitější nelokální opravy v souboru.

V obrazovkovém režimu můžeme vkládat, vypouštět resp. přepisovat text v místě kurzoru, rušit, štěpit a spojovat věty souborem.

V příkazovém režimu je iniciativa zcela na straně počítače, uživateli je nabídnuto menu příkazů, které v daném stavu může použít, počítač se sám dotazuje na potřebné parametry. U všech příkazů se zadává oblast souboru, ve které se daná akce má provádět. Lze například vyhledávat, rušit resp. nahrazovat jiným řetězcem první, všechny nebo jen některé výskyty zadaného řetězce ve specifikované oblasti souboru, vypustit, posunout resp. zkopírovat na jiné místo v souboru celou specifikovanou oblast.

- Operační systém AMOS obsahuje dále debugger, což je soubor prostředků, které umožňuji efektivně ladit programy ve strojovém kódu. Umožňují krokovat nebo volitelnou rychlostí simulovat výpočet programu, simulaci zastavovat a přitom měnit obsahy všech registrů, programového čítače, ukazatele na vrchol zásobníku, položky na vrcholu zásobníku atd. Během simulace na obrazovce stále vidíme v přehledném tvaru velmi podrobnou informaci o aktuálním stavu výpočtu. Průběh simulace můžeme ovlivňovat volbou řady parametrů např. nastavovat a rušit breakpointy, definovat podprogramy, které nemají být simulovány, ale mají být prováděny v reálném čase. Dále může debugger OS AMOS provádět disassembláž obsahu paměti. Většinu z těchto prostředků lze využít účinně také pro názornou demonstraci funkce mikroprocesoru 8080.
- Pro OS AMOS byl vyvinut nový způsob nahrávání na magnetofon. Tento způsob na rozdíl od dosavadního nezávisí na polaritě, se kterou magnetofon nahrává, a je několikrát rychlejší. Přitom se jeho spolehlivost spíše zvýšila, neboť umožňuje provádět bezprostředně po pořízení nahrávky její verifikaci. Navíc si systém sám dokáže najít správný soubor, odpadá tedy dosavadní pracné vyhledávání správného místa na pásce. Pro lepší orientaci uživatele systém při vyhledáváni souboru na pásce vypisuje identifikace všech záznamů, které na pásce čtecí hlava přechází.

- Magnetofonový soubor může být v OS AMOS prakticky libovolně velký, vstup a výstup do něj se děje přes buffer, jehož velikost může uživatel ovlivnit. To spolu se skutečností, že oba překladače OS AMOS jsou jednoprůchodové, umožňuje překládat i velmi dlouhé zdrojové programy. Zvolená koncepce magnetofonových souborů bude mít také velký význam v oblasti tvorby a použití výukových programů, které vyžaduji velké množství textových údajů.
- Operační systém AMOS je kompatibilní se všemi přídavnými moduly k mikropočítači IQ 151, o nichž se podařilo jeho autorům od výrobce dostatečné informace.

Všechny komponenty OS mohou pracovat jak s modulem VIDEO 32, tak i s modulem VIDEO 64.

Je-li připojen modul STAPER, je možné pracovat s tiskárnou, snímačem a děrovačem děrné pásky.

Překladač Pascalu je doplněn o standardní funkce pro práci se souřadnicovým zapisovačem MINIGRAF.

Operační systém obsahuje povely pro zapínání a vypínání jemné grafiky v modulu GRAFIK. Kreslení pomocí modulu GRAFIK půjde ovládat i přímo z programů v Pascalu. Protože však již není v paměti EPROM OS AMOS dost místa, budou se výkonné procedury pro kreslení nahrávat do operační paměti.

OS AMOS je kompatibilní také s modulem BASIC 6.

Není však kompatibilní s modulem Basic G (ten dokonce nesmí být s moduly OS AMOS současně zasunut do počítače). Počítá se však s tím, že již koncem roku 1987 se bude vyrábět nový modul grafického Basicu, který nejen, že bude kompatibilní s AMOSem, ale půjde z něj přímo pracovat se soubory OS AMOS.

V OS AMOS je možné pracovat pomocí modulu DISC 2, vyráběného v ZPA Čakovice, závod Pragotron, i se soubory na pružných discích.

2. Technická realizace OS AMOS

Operační systém AMOS se bude dodávat ve dvou logických modulech: **AMOS/Pascal** a **AMOS/Assembler**. Oba moduly mohou pracovat v počítači zároveň, ale mohou také pracovat každý samostatně.

Modul AMOS/Assembler je realizován v jednom zásuvném modulu, který obsahuje 16Kbytů paměti EPROM. V ní jsou umístěny tyto komponenty OS AMOS:

- vlastní operační systém
- textový editor
- překladač assembleru
- debugger

Modul Amos/Pascal je tvořen dvěma paměťovými prostory po 16Kbytech. Vzhledem k nedostatku 4Kbytových pamětí EPROM 2732 je realizován jako dvojice zásuvných modulů, z nichž každý obsahuje jeden paměťový prostor. Zásuvný modul je však navržen tak, že jej lze beze změn použít i pro realizaci dvou paměťových prostorů pomocí 4Kbytových pamětí. Jejich zajištěním by se mohla zvětšit produkce modulů AMOS/Pascal dvakrát bez dalšího zatížení výroby.

První paměťový prostor modulu Amos/Pascal obsahuje tyto komponenty OS AMOS:

- vlastní operační systém
- obrazovkový editor
  tyto dvě složky jsou totožné s odpovídajícími složkami, modulu AMOS/Assembler
- systém podpůrných programů pro běh pascalských programů, (run-time-support).

Druhý paměťový prostor tohoto modulu obsahuje překladač jazyka Pascal.

Operační systém AMOS tedy zabírá trojnásobně adresový prostor 8000H - 0BFFFH, který dosud nebyl v mikropočítači IQ 151 využíván. Přepínání těchto tří adresových prostorů se děje automaticky (provádí jej operační systém zápisem na řídící port) a uživatel o něm neví.

N.p. Komenium bude distribuovat také páskovou verzi modulu AMOS/Assembler. Tato verze bude až na nevyhnutelné výjimky uživatelsky ekvivalentní verzi realizované v zásuvných modulech s pamětí EPROM.

I. Koncepce operačního systému a ovládání periferií

1. Operační systém a práce se soubory

Názvem operační systém je zvykem označovat souhrn programových modulů, které řídí a přidělují prostředky (vnitřní a vnější paměť, zařízení pro vstup a výstup dat), které jsou na daném počítači k dispozici. Operační systém tedy tvoří mezičlánek mezi technickým vybavením (hardware) a uživatelskými programy. Z tohoto vymezení je vidět, že bez alespoň minimálního operačního systému není možné pracovat na žádném počítači. Takový minimální operační systém je u mikropočítačů zvykem označovat jako monitor. Mikropočítač IQ 151 je také takovým monitorem vybaven.

Vlastnosti operačního systému ovlivňují rozhodující mírou uživatelské vlastnosti počítače i efektivitu práce na něm, mohou podtrhnout či zdůraznit přednosti i nedostatky technických prostředků počítače.

AMOS je nový operační systém pro školní mikropočítač IQ 151, který umožňuje uživateli pracovat na něm způsobem obvyklým na počítačích vyšší třídy než je IQ 151. Při tom je schopen plnohodnotně pracovat v základní konfiguraci mikropočítače (IQ 151 s modulem VIDEO a kazetový magnetofon). Na rozdíl od obvyklé terminologie budeme o překladačích, editoru a debuggeru pro jednoduchost mluvit jako o součásti operačního systému, i když je jasné, že jsou to samostatné programy, které pouze využívají služeb vlastního operačního systému.

Název operačního systému AMOS vznikl jako zkratka názvu

Autonomní Mikropočítačový Operační Systém

nebo, chcete-li anglicky,

Almost Memory Oriented System.

Nemalou roli při volbě názvu sehrálo samozřejmě i to, že mikropočítač IQ 151 je určen především do škol a že jej distribuuje n.p. Komenium. Přívlastky autonomní resp. "memory oriented" mají zdůraznit nezávislost operačního systému i jeho překladačů na existenci výkonné vnější paměti (např. pružných disků).

Operační systém AMOS je plně souborově orientován, to znamená, že všechny manipulace s daty, ač se dějí na systémové úrovni nebo ať jde o vstupně/výstupní akce programů, pracují na souborové úrovni.

2. Soubory v OS AMOS

2.1. Pojem souboru

Termínem soubor označujeme posloupnost údajů, které logicky patří k sobě. Soubor označujeme jménem. Souborem nazýváme např. nahrávku na magnetofonu, program nebo data na disketě. Soubor vytváříme tak, že do něj postupně zapisujeme údaje. Z existujícího souboru můžeme údaje naopak číst. Soubor se může vytvářet také např. na obrazovku nebo na tiskárnu - co zaznamenáme do takového souboru, to se vypíše nebo vytiskne. Soubor lze číst nejen z média, které uchovává údaje, ale např. i z klávesnice - přečte se to, co uživatel napíše.

Soubor je tedy abstraktní pojem, který uživateli dovoluje pracovat jednotným způsobem s daty a nezajímat se o detaily spolupráce s tím či oním periferním zařízením.

Vnitřní reprezentace a organizace souboru značně závisí na tom, na jakém médiu je zaznamenán, resp. z jakého zařízení se čte nebo na jaké se zapisuje. Detailní popis této fyzické úrovně souborů není v tomto manuálu obsažen. Uživatel jej nepotřebuje znát, protože pracuje v OS AMOS se soubory na vyšší úrovni: čte údaje ze souboru nebo je zapisuje do souboru stejným způsobem, ať je soubor realizován na kterémkoliv zařízení.

Uživatel se může odkazovat na soubor dvěma způsoby. Na úrovni operačního systému, např. při kopírování, rušení souborů a spouštění výpočtů programů, uvádí vždy specifikaci konkrétních souborů. Tato specifikace obsahuje (případně implicitně) také údaje o zařízení, na kterém se soubor nachází.

Naproti tomu při programování se uživatel na soubory, s nimiž bude pracovat, odkazuje symbolicky. Odkaz na soubor je vlastně identifikátorem. Konkrétní soubor, s kterým má program pracovat, se určí až při spouštění programu.

2.2. Klasifikace souborů v OS AMOS

Soubory v OS AMOS je možno klasifikovat podle dvou hlavních hledisek:

- podle zařízení, na kterém se soubor nalézá
- podle typu souboru, tj. podle jeho vnitřní struktury a role, kterou v OS AMOS hraje.

2.2.1. Klasifikace souborů podle zařízení

Základním dělením podle zařízení je dělení na soubory paměťové (vnitřní), diskové, vnější a soubory na uživatelem definovaných zařízeních.

Paměťové soubory jsou uloženy v operační paměti mikropočítače.

Jako vnější označujeme soubory, které nejsou uloženy v operační paměti počítače. Může jít buď o soubory vstupní, ze kterých se čtou data, nebo o soubory výstupní, do kterých se data zapisují.

Zvláštní skupinu tvoří diskové soubory, které sice patří tím, že nejsou uloženy v paměti počítače, mezi soubory vnější, mají však svoji povahu velmi blízko souborům paměťovým.

Pojem uživatelem definovaných zařízení umožňuje uživatelům OS AMOS napsat si vlastní ovládací programy pro některá zařízení a pracovat pak i s nimi na souborové úrovni.

Dále uvádíme soupis všech zařízení, na nichž může uživatel OS AMOS pracovat. Jak uvidíme v kapitole II., specifikace souboru v OS AMOS obsahuje předponu, která udává zařízení, na kterém se soubor nalézá. V pravé části následujícího soupisu se uvádějí tyto předpony i s vysvětlením jejich mnemoniky.

Vnitřní resp. paměťové soubory

soubor v paměti počítače	:MM: (MeMory)

Diskové soubory

soubor na pružném disku ve štěrbině č.0	:D0: (Disk 0)
soubor na pružném disku ve štěrbině č.1	:D1: (Disk 1)

Vnější soubory

Vstupní soubory

soubor vstupující z klávesnice	:CI: (Console Input)
soubor vstupující z magnetofonu	:MG: (MaGnetofon)
soubor vstupující ze čtečky děrné pásky	:RI: (Reader Input)

Výstupní soubory

soubor vystupující na obrazovku	:CO: (Console Output)
soubor vystupující na magnetofon	:MG: (MaGnetofon)
soubor vystupující na tiskárnu	:LP: (Line Printer)
soubor vystupující na děrovač děrné pásky	:PO: (Punch Output)

Fiktivní soubor

soubor na fiktivním zařízení	:BB: (Byte Bucket)

Soubory na uživatelem definovaných zařízeních

soubor na uživ. zařízení č.0	:U0: (User 0)
soubor na uživ. zařízení č.1	:U1: (User 1)

Tiskárnu, snímač a děrovač děrné pásky mohou užívat jen ti uživatelé, kteří mají svůj počítač vybaven interfejsem pro tato zařízení (standardně modul STAPER).

Se soubory na pružných discích mohou pracovat ti uživatelé, kteří mají jednotku pružných disků PFD 151 resp. 251 připojenou pomocí modulu DISC 2.

Uživatel, který chce pracovat se soubory na uživatelských zařízeních musí, definovat podprogramy pro přístup k nim.

2.2.2. Klasifikace souborů podle jejich typu

Typ souboru udává vnitřní strukturu souboru a jeho roli v operačním systému.

Podle typu se soubory v OS AMOS dělí na:

- textové soubory
- object-kódy
- hexadecimální soubory

Typ souboru se v jeho specifikaci (jak uvidíme v kapitole II.) udává pomocí nejvýše třípísmenné přípony. Uživatel může využívat k rozlišování svých souborů všech přípustných přípon, musí však dodržovat následující konvenci:

typ souboru	vyhrazené přípony
textové soubory	všechny přípony nezačínající písmeny O a H
object-kódy	přípony začínající písmenem O, prakticky význam má však jen přípona OBJ
hexadecimální soubory	přípony začínající písmenem H

Typ souboru je při jeho vytváření určen příponou, uvedenou v jeho specifikaci. Systém obsahuje základní kontroly, které nedovolí použití nesprávné přípony v případech, kdy by to mohlo mít katastrofické důsledky. Tím je uživatel poměrně spolehlivě chráněn proti neúmyslné chybě. Systém těchto kontrol však nemůže být tak dokonalý, aby nešel snadno "obelstít". Varujeme uživatele před důsledky takového počínání, přinejmenším si přidělá práci.

2.3. Typy souborů v OS AMOS

2.3.1. Textové soubory

Textové soubory slouží k uchovávání textové informace (tj. např. zdrojových textů programů, textů ve vlastním slova smyslu, tabulek, ale i obrázků). Textový soubor v OS AMOS může obsahovat všechny zobrazitelné znaky včetně inverzních a semigrafických.

Aby se docílilo lepšího využití operační paměti počítače, jsou textové soubory fyzicky uloženy s zapakovanými mezerami (souvislý úsek 3-250 mezer je uložen jako dva byty).

Dvojicemi znaků CR, LF je textový soubor dělen na věty. Délka věty textového souboru není v OS AMOS nijak omezena a nemá žádnou souvislost s délkou řádky na obrazovce či na tiskárně. Má-li však být obsahem textového souboru zdrojový program, je třeba mít na paměti, že překladače požadují omezenou délku věty.

K vytváření a opravám textových souborů slouží systémový program Editor.

Editor převádí při své práci textové soubory do vlastního vnitřního tvaru a po skončeni editace zpět do vnějšího tvaru. Pro účely organizace vnitřního tvaru souborů používá pěti posledních inverzních znaků (tj. znaků s kódy 0FBH až 0FFH). Proto se tyto znaky při otevření editoru na soubor "odinverzní".

Fyzická reprezentace souborů vytvářených editorem neobsahuje byty s nejvyšším bitem rovným 1, inverzní znaky jsou v něm reprezentovány pomocí "přechodu do inverze".

2.3.2 Object-kódy

Překladače OS AMOS produkují absolutní kód, ten se však nezavádí okamžitě při překladu na místo, kde bude počítat, ale generuje se do souborů zvláštního typu tzv. object-kódů. K označováni object-kódů je v OS AMOS vyhrazena přípona OBJ.

Standardně se program zavádí na správné místo v paměti teprve před vlastním výpočtem. Překladače mohou vytvářet object-kód jako do paměťových, tak i do vnějších souborů (např. na magnetofon). Proto nemusí místo, kde bude program počítat, být již v okamžiku překladu volné. Toto řešení umožňuje efektivní využití operační paměti mikropočítače a tak překládat a spouštět i velmi dlouhé programy.

Alokace programu se provádí automaticky při každém spuštění uživatelského programu, po skončení výpočtu není alokovaný program chráněn před přepisem soubory.

Tam, kde je potřeba, aby se alokovaný program zachoval, je nutné jej alokovat do oblasti SPACE (viz. dále), kde je chráněn před přepisem soubory. K zavádění takových souborů do paměti slouží příkaz LOCATE, spouštějí se pak příkazem RUN.

Do object-kódu je možné vygenerovat i tabulku symbolů, kterou pak využívá disassembler při simulaci.

2.3.2. Hexadecimální soubory

Tímto termínem se v OS AMOS označují soubory, které z hlediska OS nemají žádnou vnitřní strukturu a operační systém je tedy chápán jako posloupnost bytů. K označování těchto souborů jsou vyhrazeny přípony začínající písmenem H.

Při vstupně-výstupních akcích s hexadecimálními soubory nedochází k žádným konverzím, neboť jejich fyzická reprezentace odpovídá přesně jejich logickému obsahu. Proto slouží hexadecimální soubory např. k přenosu dat na jiný systém resp. z jiného systému a jako soubory a přímým přístupem.

2.4. Práce se soubory v OS AMOS

Každý soubor v OS AMOS je vlastně posloupnost bytů, ze které je v každém okamžiku přístupný právě jeden byte. Na tento byte souboru ukazuje tzv. pozice souboru.

I když tomu tak v konkrétní fyzické reprezentaci souborů není, je výhodné si představovat, že každý soubor je vždy ukončen jedním speciálním "ukončovacím bytem" EOF (End Of File - Konec souboru). Ukazuje-li pozice souboru na tento "byte" říkáme, že pro soubor platí podmínka EOF. Prázdný soubor se tedy skládá pouze z ukončovacího bytu.

První akcí, kterou je třeba se souborem provést, je jeho otevření. Tak je zvykem označovat souhrn inicializačních akcí se souborem (např. vyhrazení paměti pro buffer u magnetofonových a diskových souborů nebo vyděrovaní zaváděcího úseku při výstupu na děrnou pásku apod.).

Po ukončení práce se souborem je třeba soubor uzavřít, tj. provést nezbytné závěrečné akce (např. vyprázdnit buffery). Uživatel však nemusí provedení této akce explicitně požadovat. Při opětovném otevírání souboru se totiž soubor nejprve uzavře, stejně tak po dokončení výpočtu OS sám uzavře všechny otevřené soubory.

V OS AMOS jsou realizovány soubory se sekvenčním přístupem. Operační systém je však koncipován tak, že po nahrání několika nezbytných procedur do paměti RAM může uživatel pracovat v paměti a na pružných discích i se soubory s přímým přístupem.

Je však dobré si uvědomit, že některé systémové programy, ale i některé uživatelské programy např. všechny programy v Pascalu, pracují se soubory na vyšší úrovni než je právě popsaná. Se souborem nemusí pracovat jen jako s posloupnosti bytů, ale jako a posloupnosti údajů vyššího typu, např. čísel, záznamů apod.

2.4.1. Soubory se sekvenčním přístupem

Při sekvenčním přístupu k souboru nemáme možnost pozici tohoto souboru (tj. místo, které je v souboru právě přístupné) libovolně nastavit. V OS AMOS můžeme otevírat soubory pro čtení, pro zápis nebo pro přípis.

Otevíráme-li soubor pro čtení, musí soubor již existovat, jinak je hlášena chyba. Pozice souboru se nastaví na jeho počátek. Ze souboru otevřeného pro čtení můžeme přečíst byte, na který ukazuje pozice, a tím posunout pozici souboru o jeden byte dál. Ukazuje-li pozice souboru na jeho ukončovací byte EOF, je indikován konec souboru a při dalším pokusu o čtení ze souboru dojde k chybě.

Obdobně při otvírání souboru pro zápis nesmí soubor dosud existovat, jinak dojde k chybě. Pozice souboru se nastaví na jeho počátek. Do souboru otevřeného pro zápis lze zapisovat byte na místo určené pozicí a tím posunout pozici o jeden byte dopředu. Zapisujeme tedy do souboru tímto způsobem pouze od jeho začátku.

To může být uživatelsky velmi nepříjemné, proto OS AMOS umožňuje otevřít sekvenční soubory i pro tzv. přípis. Při tomto způsobu otevření souboru se pozice nastaví na jeho "ukončovací byte" a je dovoleno do souboru zapisovat. Tímto způsobem je tedy možné soubor prodlužovat.

Je-li soubor otevřen pro zápis nebo přípis, pak vždy platí pro soubor podmínka EOF, tj. pozice ukazuje na ukončovací "byte" souboru. Do sekvenčního souboru lze tedy zapisovat pouze na jeho konec.

2.4.2. Soubory s přímým přístupem

Sekvenční přístup k souborům je nejen nejjednodušší, ale také nejobvyklejší organizaci přístupu k souborům. Některá zařízeni, zvláště vstupní a výstupní ani jinou organizaci souborů nepřipouštějí (např. tiskárna, magnetofon, děrná páska atd.). V mnohých aplikacích je však nutné mít možnost přistupovat k libovolnému specifikovanému bytu souboru. Potom mluvíme o tzv. přímém přístupu do souboru.

Při přímém přístupu k souboru má uživatel možnost nastavit pozici souboru na kterékoli místo v něm a z tohoto místa pak může číst resp. na toto místo zapsat byte (při tom se pozice vždy posouvá o jeden byte kupředu). Při pokusu o nastavení pozice souboru za jeho konec, dojde k chybě.

Ve verzi 4 OS AMOS jsou zahrnuty některé prostředky pro realizaci souborů s přímým přístupem. Další však bude třeba nahrávat z vnější paměti (magnetofon nebo disk) do paměti RAM a budou jedním ze systémových prostředků dodávaných k OS AMOS samostatně. Bližší o přímém přístupu bude v dokumentaci k tomuto programu.

Při přímém přístupu k souboru se pozice v souboru bude zadávat pořadovým číslem bytu v souboru (počítá se od nuly, pozice se ukládá ve třech bytech).

2.4.3. Rutiny jádra OS AMOS pro práci se soubory

Uživatelské programy pracují se soubory na logické úrovni pomocí služeb podprogramů jádra OS. Tyto služby zcela odstiňují uživatelské programy od problematiky konkrétní fyzické reprezentace souborů.

K práci se soubory slouží v OS AMOS následující služby jádra operačního systému:

procedury (mění stav souboru)

OPEN	otevření souboru
CLOSE	uzavření souboru
GET	čtení jednoho bytu ze souboru
PUT	zápis jednoho bytu do souboru

V dalších odstavcích charakterizujeme stručně akce, které tyto procedury provádějí. Jejich podrobný popis, který potřebují pouze ti uživatelé, kteří programují v assembleru bude v kapitole V.

2.4.3.4. Otevření souboru - procedura OPEN

Před první akcí se souborem je třeba soubor otevřít, pomocí procedury OPEN.

Podprogram OPEN nejprve soubor uzavře (nebyl-li již uzavřen), a poté vykoná nezbytné inicializační akce se souborem (např. vyhrazení místa pro buffer, vyděrování zaváděcího úseku na děrné pásce apod.). Tyto akce jsou popsány u jednotlivých zařízení.

V OS AMOS jsou možné čtyři typy otevření souboru:

- otevření ke čtení
  Soubor musí existovat, jeho pozice se nastaví na počátek souboru a ze souboru je možné číst,
- otevření k zápisu
  Soubor nesmí existovat, jeho pozice se nastaví na počátek souboru a soubor je připraven k zápisu,
- otevření k připisování
  Soubor musí být paměťový nebo diskový. Je-li diskový, pak musí již existovat. Pozice souboru je nastavena na "ukončovací byte" souboru a soubor je připraven k zápisu,
- otevření pro přímý přístup
  Soubor musí být hexadecimální, a to paměťový nebo diskový, je připraven k přímému přístupu, jeho pozice souboru není definována.

Soubory na výlučně vstupních zařízeních je pochopitelně možné otevřít pouze pro čtení, na výlučně výstupních pouze pro zápis. Pro přímý přístup lze otevřít pouze hexadecimální soubory, protože textové soubory jsou uchovávány úsporně s pakovanými mezerami a uživatel tedy nezná přesný tvar souboru.

Otevření souboru k přímému přístupu má smysl pouze tehdy, jsou-li v paměti RAM nahrány příslušné obslužné rutiny. Proto je v případě, kdy nahrány nejsou hlášena chyba.

2.4.3.2. Uzavření souboru - procedura CLOSE

Procedura CLOSE provede nutné závěrečné akce po ukončení přístupu k souboru, např. vyprázdnění bufferu, výstup spoolovaného souboru (viz 2.5.) na specifikované výstupní zařízení atd.

Akce uzavírání souboru se vyvolá i implicitně při otevírání neuzavřeného souboru a po ukončení úlohy.

2.4.3.3. Čtení bytu ze souboru - procedura GET

Není-li soubor otevřen pro čtení resp. pro přímý přístup, způsobí zavolání procedury GET chybu.

Chyba nastává také v případě, kdy pro soubor platí podmínku EOF (tj. chci-li číst po konci souboru).

Procedura přečte jeden byte ze souboru a posune pozici v souboru o jeden byte kupředu.

2.4.3.4. Zápis bytu do souboru - procedura PUT

Není-li soubor otevřen pro zápis, přípis nebo přímý přístup, je zavolání procedury PUT chybou.

Procedura zapíše na místě pozice do souboru byte a posune pozici o jeden byte dále. Pokud nejde o přímý přístup, (při kterém je možné "zapisovat" dovnitř souboru) soubor se tím prodlouží.

2.4.4. Rutiny pro přímý přístup do souborů

V tomto odstavci popíšeme logiku práce v OS AMOS při přímém přístupu k souborům.

Jak jsme však již řekli, jejich výkonné části je z prostorových důvodů nutné "přihrát" do paměti RAM. Nejsou-li k dispozici, hlásí OS AMOS chybu.

Kromě procedur OPEN, CLOSE, GET a PUT, se kterými jsme se již seznámili slouží k práci se soubory s přímým přístupem ještě

	procedura
SEEK	nastavení pozice souboru
	a funkce
	(nemění stav souboru, pouze vrací informace o něm)
POS	vrací pozici souboru,
LASTPOS	vrací pozici konce souboru (EOF).

2.4.4.1. Nastavení pozice souboru - procedura SEEK

Zavolání procedury na soubor, který není otevřen pro přímý přístup, je chyba.

Procedura dostane trojbytové číslo udávající pořadí bytu v souboru, na nějž je třeba nastavit pozici souboru. Leží-li specifikovaný byte již mimo dosavadní obsah souboru, je hlášena chyba.

2.4.4.2. Zjištění pozice souboru - funkce POS

Zavolání funkce POS na soubor, který není otevřen pro přímý přístup je chyba. Funkce nijak nemění stav souboru, pouze vrací trojbytové číslo, které udává aktuální pozici souboru.

2.4.4.3. Zjištění pozice konce souboru - funkce LASTPOS

Zavolání funkce LASTPOS na soubor, který není otevřen pro přímý přístup, je chyba. Funkce nijak nemění stav souboru, jen vrátí trojbytové číslo, udávající polohu konce souboru (v naší terminologii "ukončovacího bytu" souboru EOF). Pro prázdný soubor tedy vrátí nulu.

2.5. Spoolovaný výstup

OS AMOS umožňuje spoolovat výstup na tiskárnu, obrazovku, děrovač děrné pásky a na obě uživatelem definovaná zařízení.

Spooling je technika, která se běžně používá v operačních systémech velkých počítačů. Spočívá v tom, že během výpočtu programu se jeho výstupy nedějí přímo na výstupní zařízení, ale do diskového souboru. Převod dat z tohoto souboru na konkrétní výstupní zařízení zařídí po skončení výpočtu operační systém.

Tím se dosahuje lepšího využití času procesu a také toho, že program může posílat více výstupních souborů na totéž zařízení, aniž by došlo k "promíchání" jejich obsahů.

Tento druhý důvod nás vedl k tomu, abychom možnost spoolovaného výstupu zahrnuli i do OS AMOS.

Požadavek, aby výstup do souboru byl spoolován, vyjádříme v dodatku specifikace výstupního souboru (viz kapitola II.). Tento soubor se pak vytváří na tzv. spoolovacím zařízení a při uzavírání souboru jej nechá OS vystoupit na specifikované výstupní zařízení. K uzavření souboru dojde buď automaticky po dokončení výpočtu nebo explicitním voláním procedury CLOSE. Protože můžeme jeden soubor v rámci jediného výpočtu po uzavření opět otevřít, může soubor na výstupní zařízení vystupovat "po částech".

Spoolovacím zařízením je buď paměť (:MM:) v případě, že nejsou připojeny pružné disky nebo nultá štěrbina (:D0:), pokud jsou připojeny. Pro uživatele pružných disků bude k dispozici speciální program, který jim umožní měnit spoolovací zařízení.

Je zřejmé, že je-li spoolovacím zařízením paměť, nemohou být spoolované soubory příliš veliké.

3. Hospodaření s pamětí RAM

OS AMOS a jeho systémové utility mají umístěny svoje proměnné v dolní části adresované prostoru nad proměnnými monitoru.

V operačním systému se pracuje s úseky paměti velikosti 256 bytů, které se nazývají sektory.

Vlastní pracovní prostor se operační systém bere:

- není-li přítomen zásuvný modul BASIC 6:
  celou operační paměť počítače nad úsekem proměnných (s výjimkou posledního sektoru, který je monitorem IQ 151 využíván jinak a do kterého si OS AMOS umísťuje zásobník)

nebo

- je-li zásuvný modul BASIC 6 přítomen:
  oblast USR, kterou uživatel vyhradil Basickým příkazem CLEAR (tato oblast musí být veliká alespoň 4 Kbyty).

Popíšeme, jak OS AMOS s tímto pracovním prostorem zachází.

Jsou-li připojeny pružné disky jsou dva nejspodnější sektory vyhrazeny pro buffer přístupu na disk.

Na spodním kraji zbývajícího prostoru se nalézá souvislá oblast s tabulkou souborů a se soubory, jejíž velikost se dynamicky mění.

Na horním kraji tohoto prostoru může uživatel vyhradit příkazem SPACE oblast pro explicitně alokované programy. Tato oblast má vždy velikost alespoň jeden sektor, tento jeden sektor je rezervován pro speciální účely (budou jej užívat systémové programy).

Oblast paměti mezi soubory a oblasti SPACE se používá jako pracovní oblast pro systémové programy (překladače, debugger, editor).

Paměť pro tabulku souborů a vlastní soubory je přidělována dynamicky po sektorech.

Není tedy počet souborů omezen velikostí tabulky, ale jen velikostí paměti. Soubor zabírá v paměti vždy jen tolik sektorů, kolik odpovídá jeho aktuální velikosti. Další sektor je souboru přidělen teprve tehdy, když se do něj má zapsat byte, pro který již ve stávající alokaci souboru "není místo".

Přitom v každém okamžiku tabulka souborů, každý soubor sám i tabulka a všechny soubory dohromady zabírají souvislý úsek operační paměti. Potřeba připsání jediného bytu do souboru může vyvolat potřebu přesunu všech ostatních souborů v paměti.

4. Činnost interpretu příkazů OS AMOS

Interpret uživatelských příkazů (CCP - control command processor) je ta část operačního systému, se kterou se uživatel přímo komunikuje. Jejím úkolem je dekódovat příkazy uživatele, volat programy, které realizují akce požadované uživatelem, a předávat jim parametry.

Interpret se hlásí uživateli nápovědným znakem "=", po jehož vypsání očekává zadání příkazu.

Po inicializaci systému a vždy po dokončení uživatelova příkazu provede interpret postupně tyto akce:

- uzavře všechny soubory
  (přitom se např. vyprázdní buffery a vystoupí všechny spoolované soubory)
- zruší v tabulce všechny vnější soubory
- zkontroluje kontrolní součet obsahů paměti EPROM (nesouhlasí-li, ohlásí chybu)
- vypíše na obrazovku znak "="
- očekává vstup příkazu
- jde-li o volání systémového programu (utility), předá jí řízení
- jde-li o volání uživatelova programu,
  umístí jeho kód na správné místo v paměti, předá volanému programu odkazy na parametry (na soubory a na řetězec obsahující textový parametr) a pak jej odstartuje.

5. Soubory vnitřní neboli paměťové (:MM:)

Paměťové soubory jsou uloženy v operační paměti mikropočítače a všechny informace v nich uložené jsou tedy programům v každém okamžiku přímo dostupné. Jak později uvidíme, soubor může změnit místo v paměti, kde je uložen, (tj. svoji alokaci) mnohokrát nejen během svojí existence, ale dokonce i během jediného výpočtu. Uživatel o těchto změnách alokace souborů vůbec neví, rutiny jádra operačního systému, které realizují přístup k souborům na logické úrovni, však tyto změny respektují.

Přitom v každém okamžiku každý soubor sám i všechny soubory dohromady zabírají souvislý úsek operační paměti. Potřeba připsání jediného bytu do souboru může vyvolat potřeba přesunu všech ostatních souborů v paměti.

Paměťové soubory mohou být otevřeny všemi způsoby přípustnými v OS AMOS.

6. Diskové soubory (:D0: resp. :D1:)

Organizace souborů na pružném disku je převzata z OS CP/M, a je tedy možný přenos mezi oběma systémy. Pro přenos z OS CP/M je však nutné, aby jméno souboru nemělo více jak šest platných znaků, jinak nebude možné pod OS AMOS s takovým souborem pracovat (ale ani jej přepsat či zrušit). Pro opačný přenos je vhodné se vyhnout komprimaci mezer u textových souborů v OS AMOS, a používat pro přenos hexadecimální soubory.

Diskové soubory se mohou otevírat všemi způsoby definovanými v OS AMOS.

7. Soubory vnější

7.1. Vstup z klávesnice (:CI:)

Vstup z klávesnice do souboru specifikovaného předponou :CI: se děje přes vstupní buffer. Ten má délku 74 znaků a jeho obsah se nám zobrazuje na obrazovce. Obsah vstupního bufferu musíme "odeslat" do souboru klávesou CR, do té doby můžeme obsah vstupního bufferu editovat pomoci pohybů kurzoru a kláves IC a DC, stejně jako při vkládání programu v Basicu. Obsah vstupního bufferu můžeme zrušit kombinaci kláves **CTRL + "X"** a zrušit se současným smazáním obrazovky kombinací **CTRL + "nadtrhovátko"** (klávesa nad CR).

Při odeslání obsahu vstupního bufferu klávesou CR je za něj do souboru přidána dvojice bytů CR, LF (která standardně ukončuje řádku textového souboru).

Soubor ukončíme (tj. bude pro něj platit podmínka EOF), jestliže při naplňování bufferu stiskneme klávesu "**home**" (klávesa se šipkou vlevo nahoru uprostřed kláves pro pohyb kurzoru).

Pokud jsme klávesu "home" stiskli uprostřed bufferu (např. při editování), "odešle" se do souboru pouze část bufferu vlevo od místa, kde stál kurzor v okamžiku stisku klávesy "home". V tomto případě nebude obsah tohoto bufferu ukončen dvojici bytů CR,LF. Tato dvojice bytů se do souboru generuje jen je-li klávesa "home" stisknuta jako první při naplňování bufferu.

Vstup ze zařízení :CI: doporučujeme používat pro interaktivní vstup údajů do programů. Mají-li však vstupující data charakter odpovídající spíše dávkovému zpracování (tj. nezávisejí-li na předchozím výpočtu, ale jsou kompletně známa v okamžiku spouštění programu), je výhodnější připravit si data předem pomoci editoru do textového souboru.

V jiných aplikacích, např. při programování her, kdy je na závadu, že vstup údajů z klávesnice se děje přes buffer, je k dispozici okamžitý vstup znaku z klávesnice bez čekání (viz popis systémových podprogramů přístupných z assembleru v kapitole IV.).

7.2. Výstup na obrazovku (:CO:)

Do souboru se zařízením :CO: mohou vystupovat všechny znaky včetně grafických. Při výstupu na obrazovku platí všechny řídicí znaky jako v monitoru.

Výstup na obrazovku můžeme pozastavit a znovu spustit kombinacemi kláves:

CTRL + S	pozastavení výpisu
CTRL + Q	opětovné spuštění výpisu

Všechny komponenty OS AMOS umí spolupracovat jak s modulem VIDEO 32, tak i s modulem VIDEO 64.

Soubor na zařízení :CO: lze (pochopitelně) otevřít pouze k zápisu.

Poznámka:
Je dobré mít na paměti, že monitorský podprogram pro výstup znaku na obrazovku, kterého využívá i AMOS, špatně interpretuje znak tabelátoru v případě, kdy by se měl nastavit na začátek dalšího řádku. V tomto případě dojde k posunu kurzoru na počátek řádky a ne už k posunu na další řádku.

7.3. Magnetofonové soubory (:MG:)

Protože se práce s magnetofonovými soubory v OS AMOS jak svojí logikou, tak i praktickým ovládáním značně liší od dosavadního způsobu práce s magnetofonem, popíšeme ji nejprve v obecných rysech a teprve poté detailněji.

Operační systém AMOS používá jiný systém zápisu na magnetofonech než dosavadní programové vybavení IQ 151. Tento způsob je několikrát rychlejší než dosavadní. Navíc umožňuje provádět při zápisu na magnetofon verifikaci jeho správnosti a při čtení automatické vyhledávání souboru na pásce.

Magnetofonové soubory v OS AMOS mohou být libovolně velké, nejen značně větší než je velikost právě volné paměti pro soubory, ale i značně větší než je velikost celé operační paměti mikropočítače IQ 151. Je to umožněno tím, že vstup a výstup do magnetofonového souboru se děje přes buffer.

Důsledkem toho je, že zápis souboru na pásce není souvislý, ale skládá se z více bloků, bloky se číslují od nuly, nultý blok tvoří tzv. hlavička, obsahující základní údaje o souboru. Každý další blok obsahuje kromě identifikačních údajů (jméno souboru a číslo bloku), které jsou rutinou pro čtení bloku používány pro jeho vyhledáváni na pásce, vždy

jeden obsah bufferu souboru. Velikost tohoto bufferu má uživatel možnost předepsat při vytváření magnetofonového souboru v dodatku jeho specifikace (viz kapitola II.). Není-li velikost bufferu specifikována, bere se jako implicitní hodnota jeho velikosti 16 sektorů (tj. 4Kbyty).

Magnetofonové soubory lze otevřít pro čtení a pro zápis.

Při otevření výstupního magnetofonového souboru se vyhradí v operační paměti potřebné místo pro buffer magnetofonového souboru a na pásku se zapíše jeho nultý blok ("hlavička").

Zápis do souboru se pak provádí do bufferu v paměti, teprve když je buffer plný, přeruší se provádění právě běžícího programu a uživatel je vyzván, aby připravil magnetofon k zápisu. Na magnetofon se zapíše do dalšího bloku magnetofonového souboru obsah bufferu a uživatel je dotázán, zda chce provést verifikaci právě provedeného zápisu na magnetofon.

Při verifikaci se čte zápis na magnetofonové pásce a porovnává se s obsahem bufferu v operační paměti. Jestliže jsou zjištěny chyby v zápisu, může uživatel buď opakovat zápis bloku na pásku, nebo opakovat verifikaci. Verifikaci lze opakovat i v případě, že dopadla dobře. Po úspěšném zápisu bloku na magnetofon se řízení vrátí do programu, který o zápis požádal. Ten pak zapisuje do bufferu magnetofonového souboru opět od jeho začátku.

Celá činnost se opakuje, dokud výstup do souboru není dokončen. Případný zbytek souboru v bufferu vystoupí na magnetofon při uzavírání souboru. Neuzavře-li soubor programátor explicitním zavoláním rutiny CLOSE jádra OS, je uzavřen po dokončení úlohy interpretem příkazů OS AMOS.

Čtení z magnetofonového souboru se děje takto:

Při otevření magnetofonového souboru se nejprve vyhledá a přečte jeho nultý blok ("hlavička"), z ní se přečte velikost bufferu tohoto souboru a vyhradí příslušné veliké místo v paměti pro tento buffer. Poté se do tohoto bufferu přečte obsah prvního bloku souboru. Rutina pro čtení ze souboru pak postupně čte obsah bufferu, po jeho vyčerpání je uživatel vyzván k přečtení dalšího bloku souboru.

Akce vstupu resp. výstupu souboru na magnetofon může být vyvolána nejen ze systémové utility COPY, která slouží ke kopírováni souborů, ale také z ostatních utilit (např. čtení zdrojového programu z magnetofonového souboru, výstup object-kódu na magnetofon) i z uživatelských programů.

7.3.1 Výstup do magnetofonového souboru

Nejprve popíšeme akci zápisu jednoho bloku magnetofonového souboru na pásku a poté popíšeme celou akci zápisu na magnetofon globálně.

7.3.1.1. Zápis bloku magnetofonového souboru na pásku s jeho verifikací

Iniciativa k zápisu vychází od OS AMOS. Na obrazovce se objeví výzva:

ZAPIS čč/(specifikace souboru)

kde čč je číslo bloku, který se má zapsat, a (specifikace souboru) je specifikace souboru (bez dodatku), v tvaru, v jakém se zapisuje v příkazech OS AMOS a znak

"!"

a systém očekává, že uživatel připraví magnetofon k zápisu a poté stiskne libovolnou klávesu. Po stisku této klávesy se okamžitě začíná nahrávat příslušný blok souboru.

Po ukončení jeho zápisu na pásku se na obrazovce objeví dotaz:

VERIFIKOVAT (A/N)?

Uživatel má tři možnosti, jak reagovat:
(reaguje-li jinak vypíše se ? a systém čeká na správnou reakci)

- Po stisku klávesy "home" se verifikace neprovede a systém nám nabídne možnost opakovat zápis bloku na magnetofon výzvou
  ZNOVA NAHRAT (A/N)? (možné reakce na tuto výzvu budou uvedeny dále),
- po odeslání odpovědi "A", se bude provádět verifikace nahrávky,
- po odeslání odpovědi "N", se verifikace provádět nebude a celý proces zápisu bloku se ukončí.

Poslední dvě odpovědi je nutné "odeslat" klávesou CR, do té doby je možné je "editovat" v bufferu délky 1.

Při verifikaci uživatel musí nastavit hlavu magnetofonu kamkoli před právě pořízený zápis bloku. Systém sám vyhledává tento zápis na pásce.

Při jeho hledání se vypisují na obrazovku identifikace všech bloků magnetofonových souborů, které hlava magnetofonu při vyhledávání přechází. Tento výpis slouží k tomu, abychom věděli, kde se na pásce čtecí hlava právě nachází. Při přecházení čtecí hlavy přes jakýkoli záznam na pásce počítač jemně "vrčí". Čtecí program se "chytne" až na hledaném bloku. To, že hledaný blok byl již nalezen, se pozná kromě

toho, že je na obrazovku vypsána jeho identifikace, také tak, že při jeho čtení se na obrazovce objevuje blikající čtený znak.

Při verifikaci se čte záznam na pásce a porovnává s jeho "vzorem" v paměti počítače.

Pokud verifikace dopadne dobře, objeví se na obrazovce hlášení:

O.K.

a opakuje se dotaz

VERIFIKOVAT (A/N)?

uživatel má k dispozici výše popsané odpovědi.

Pokud je při verifikaci zjištěna chyba v zápisu, objeví se na obrazovce nabídka opakování zápisu bloku, dotazem: **CHYBA, ZNOVA NAHRAT (A/N)?**

- Po stisku klávesy "home" se opakovaná nahrávka ani opětovná verifikace provádět nebudou a celá akce zápisu bloku na magnetofon se ukončí,

- po odeslání odpovědi "A", se objeví na obrazovce opět výzva

  ZAPIS ČČ/(specifikace souboru)

  a celá akce zápisu bloku se výše popsaným způsobem opakuje,
- po odeslání odpovědi "N", se opakuje verifikace zápisu bloku na magnetofonu a celý proces zápisu bloku se ukončí.

7.3.1.2. Globální popis výstupu do magnetofonového souboru

Při otevírání magnetofonového souboru pro zápis se nejprve rezervuje potřebný počet sektorů pro jeho buffer v oblasti pro soubory OS AMOS.

Poté se výše popsaným způsobem zapíše na magnetofon (s verifikací nebo bez ní) nultý blok souboru (tj. jeho hlavička).

Tím je akce otevření magnetofonového souboru pro zápis ukončena a řízení je vráceno programu, který o výstup do magnetofonového souboru požádal.

Tento program nyní zapisuje byty do bufferu magnetofonového souboru vždy tak dlouho, dokud jej nezaplní. Při jeho zaplnění vyvolá akci zápisu dalšího bloku souboru na pásku, jak byla popsána v předchozím odstavci. Po ukončení zápisu bloku na pásku zapisuje byty opět do bufferu od jeho začátku.

Všechny tyto akce se opakují až do ukončení výstupu do tohoto magnetofonového souboru. Poslední obsah bufferu je vypsán na magnetofon při uzavírání tohoto souboru.

7.3.2. Vstup souboru z magnetofonu

Stejně jako u výstupu na magnetofon popíšeme nejprve akci přečtení jednoho bloku magnetofonového souboru a teprve potom celé čtení z magnetofonového souboru globálně.

7.3.2.1. Čtení bloku magnetofonového souboru

Očekává-li systém čtení bloku magnetofonového souboru objeví se na obrazovce výzva:

CEKAM čč/(specifikace souboru)

kde čč je číslo očekávaného bloku a (specifikace souboru) je řetězec specifikující soubor v příkazu OS AMOS (bez případného dodatku). Pokud je magnetofon zapnut ke čtení a čtecí hlava se nachází na pásce kdekoli před hledaným blokem, nemusí uživatel provádět žádné akce.

Systém sám hledá potřebný blok na pásce. Při tom vypisuje na obrazovce identifikaci všech bloků, které čtecí hlava na pásce přechází. Při přechodu přes pásku se zápisem počítač jemně "vrčí". Uživatel může během hledání souboru s magnetofonem jakkoli manipulovat, včetně rychloposuvů oběma směry. Systém se "chytne" jedině, nalezne-li hledaný blok. Uživatel se o tom dozví tím, že se na obrazovce vypíše identifikace bloku a tím, že se v průběhu jeho čtení na obrazovce objeví blikající čtený znak.

Je-li blok správně přečten, vrací se řízení do programu, který o přečtení bloku požádal. Nepodaří-li se blok přečíst správně, objeví se na obrazovce výzva:

ZNOVU
CEKAM čč/(specifikace souboru)

Chyba je ohlášena akusticky (pípnutí). Celou akci čtení bloku je nutné opakovat.

Pokud se hledaný blok na pásce nenalezne, je třeba se vrátit do režimu zadávání příkazu OS AMOS povelem BREAK (viz kapitola II.).

Slovo "CEKAM" ve výzvě se u nultého bloku souboru vypisuje inverzně, aby byl uživatel upozorněn, že se má číst soubor od začátku. Má to význam především u druhého průchodu assembleru.

7.3.2.2. Globální popis čtení z magnetofonového souboru

Při otvírání magnetofonového souboru pro čtení se nejprve přečte výše popsaným způsobem nultý blok souboru (jeho hlavička). V ní je kromě dalších informaci zapsána i délka bufferu tohoto souboru. Systém vyhradí potřebný počet sektorů pro buffer v paměti a poté stejným způsobem přečte z pásky do bufferu první blok souboru.

Poté program, který o přečtení požádal, zpracovává postupně jednotlivé byty souboru umístěné v bufferu až do doby, kdy buffer vyčerpá. Po vyčerpání obsahu bufferu se znovu opakuje akce čtení (dalšího) bloku z pásky. Poslední blok souboru na pásce obsahuje příznak, že je poslední. Po jeho přečtení uživatel již samozřejmě není vyzýván ke čtení dalších (již neexistujících) bloků souboru.

7.4. Vstupní a výstupní zařízení přístupná přes modul STAPER

7.4.1. Výstup na tiskárnu (:LP:)

Soubory směrované na tiskárnu (zařízení :LP:) jde otevřít (pochopitelně) pouze pro zápis. Při uzavírání souboru na tomto zařízení se automaticky odstránkuje, aby výpis dalšího souboru začínal na nové stránce.

Při výstupu na tiskárnu se respektuje formát výstupu (tj. počet řádek na stránce a počet znaků na řádce), který uživatel definuje příkazem PRINTER (viz kapitola II.). Ovladač tiskárny v OS AMOS proto počítá pozici na řádce i počet řádek. Při dosažení konce řádky vloží do souboru dvojici znaků CR, LF, odpovídající přechodu na novou řádku. Při dosažení stanoveného počtu řádek na stránce se do souboru vloží znak FF, odpovídající přechodu na novou stránku.

Při výstupu na zařízení :LP: jsou softwarově obslouženy tyto řídicí znaky:

08	BS	(Backspace)	posun tiskové pozice o jednu pozici zpět
09	Tab	(Tabelátor)	posun tiskové pozice dopředu na nejbližší násobek osmi
0A	LF	(Line Feed)	posun tiskové pozice o jeden řádek dolů
0C	FF	(Form Feed)	posun tiskové pozice na začátek další stránky
0D	CR	(Carriage Return)	posun tiskové pozice na nejlevější pozici na aktuálním řádku

Normálně tedy končí řádek dvojicí řídících znaků CR, LF tak jako v textových souborech.

7.4.2. Čtení z děrné pásky (:RI:)

Soubor na zařízení :RI: lze (pochopitelně) otevřít pouze pro čtení. Při jeho otevření se na obrazovku vypíše výzva:

ZALOZ PASKU

a znak **"!"** a systém čeká na stisk libovolné klávesy. Po něm se začíná číst obsah děrné pásky, přitom se ignoruje úvodní posloupnost zaváděcích znaků (s ASCII kódem 00H).

7.4.3. Výstup na děrnou pásku (:PO:)

Soubor na zařízení :PO: lze (pochopitelně) otevřít jen na zápis. V okamžiku otevírání výstupního souboru na pásku musí být již děrovač připraven k děrování. Nejprve se vyděruje 50 zaváděcích znaků (znak s ASCII kódem 00H) a potom vystupuje byte po bytu příslušný soubor. Při uzavírání souboru se také děruje na konci souboru úsek zaváděcích znaků.

7.5. Fiktivní soubor (:BB:)

Fiktivní soubory jsou obdobou dummy-souborů u velkých počítačů, slouží především pro účely ladění programu pracujících se soubory. Fiktivní soubory jsou v OS AMOS realizovány jako soubory na fiktivním zařízení :BB:.

Soubory na zařízení :BB: lze otevřít pro čtení a pro zápis.

Fiktivní soubor je vždy ve stavu EOF, můžeme do něj nechat "vystoupit" libovolné množství dat, tato data se však nikam nezapisují, z fiktivního souboru nelze žádná data přečíst.

7.6. Soubory na uživatelských zařízeních (:U0: a :U1:)

Soubory na zařízeních :U0: a :U1: mohou být otevřeny pro čtení a pro zápis.

Uživatel musí definovat adresy podprogramů realizujících tyto funkce:

- otevření souboru pro čtení,
- otevření souboru pro zápis,
- uzavření souboru,
- akci GET,
- akci PUT.

Pokud dojde k zavolání některého z těchto případě, že uživatel nedefinoval jejich adresy, je hlášena chyba.

II. Příkazy a povely operačního systému AMOS

V této kapitola uvedeme způsoby, jak odstartovat práci operačního systému a souhrn všech příkazů a povelů, které má uživatel OS AMOS k dispozici.

Příkaz je možno zadávat po tom co se na obrazovce objeví nápovědný znak ("=") interpretu příkazů operačního systému. Během zadávání příkazu je možné vkládaný text příkazu opravovat, interpretu příkazů je předán až po jeho "odeslání" klávesou CR.

Povel je možné zadat kdykoli (přesněji kdykoli, když procesor má povolené přerušení). Povely se zadávají kombinací několika kláves a provádějí se okamžitě po jejích zadání.

Chybová hlášení operačního systému se vypisují na obrazovku jako inverzní nápis

CHYBA Číslo

Seznam chyb OS AMOS spolu s jejich významy je uveden v dodatku B.

1. Spuštění OS AMOS

Práci v operačním systému AMOS lze zahájit jedním ze třech způsobů.

Standardně voláme operační systém z monitoru příkazem:

C8000.

Operační systém lze zavolat také z Basického programu (pod Basicem 6) příkazem:

CALL HEX(8000) resp. **CALL 32768.**

Před tím musí uživatel basickým příkazem CLEAR vyhradit dostatečně velkou oblast USR, v které pak operační systém pracuje. O spolupráci s Basicem blíže v odstavci 5 této kapitoly.

Třetí možnosti, jež však nemá praktický význam v standardních aplikacích, je zavolání operačního systému z programu ve strojovém kódu příkazem:

JMP 8000H resp. **CALL 8000H.**

Po zavolání operační systém vypíše na obrazovku hlavičku "AMOS (číslo verze)" a ohlásí se interpret příkazů operačního systému nápovědným znakem "=" a očekává se zadání příkazu uživatelem.

1.1. Studený a horký start sytému

V předchozím odstavci uváděné příkazy slouží k tzv. studenému startu systému, při tomto způsobu odstartování systému se provádí kompletní inicializace systému včetně vytváření prázdné tabulky souborů. Tento způsob vstupu do systému je nejběžnější. Pokud však například provedeme (ať už úmyslně nebo omylem) **reset** počítače, pak bychom při studeném startu přišli o všechna data v počítači. Proto je obvyklé dát uživateli ještě možnost tzv. horkého startu.

Při horkém startu se neprovádí inicializace všech proměnných ani mazání tabulky souborů. Pokud provádíme horký start po nějaké havárii, je nutné mít na zřeteli, že obsah tabulky souborů a systémových proměnných může být poškozen a tudíž nekonzistentní. Proto je lépe provést jen "záchranné" akce a poté provést opětovný studený start.

Horký start se provádí stejnými příkazy jako start studený, pouze musíme nahradit adresu studeného startu 8000H adresu horkého startu 8003H.

2. Specifikace souborů

Specifikace souboru má v operačním systému AMOS tvar:

(předpona) (jméno) .(přípona) (dodatek) .

Hranaté závorky jsou zde i v dalším textu použity pro vyznačení nepovinných částí.

Poznámka: bohužel při tisku nedošlo ke změně závorek a z originálního výtisku nejde poznat co mělo být v normálních a co v hranatých závorkách.

2.1. Specifikace zařízení

Pomoci řetězce (předpona) specifikujeme zařízení, na kterém se soubor nalézá. V operačním systému AMOS jsou přípustné tyto specifikace zařízení:

:MM:	soubor v paměti počítače	(MeMory)
:CI:	vstup z klávesnice	(Console Input)
:CO:	výstup na obrazovku	(Console Output)
:MG:	soubor na magnetofonu	(MaGnetofon)
:BB:	fiktivní zařízení	(Byte Bucket)
:LP:	výstup na tiskárnu	(Line Printer)
:PO:	výstup na děrovač děrné pásky	(Punch Output)
:RI:	vstup z čtečky děrné pásky	(Reader Input)
:U0:	uživatelské zařízení číslo 0	
:U1:	uživatelské zařízení číslo 1	
:D0:	soubor na pružném disku ve štěrbině číslo 0	
:D1:	soubor na pružném disku ve štěrbině číslo 1	

Pokud není (předpona) ve specifikaci uvedena, je specifikace chápána s implicitní hodnotou. Touto implicitní hodnotou předpony je :MM:, tj. předpokládá se, že jde o paměťový soubor.

Pokud uvedeme předponu :D0: resp. :D1: v případě, že nemáme v době startu připojen modul DISC s jednotkou pružných disků, je hlášena chyba. Obdobně se hlásí chyba, pokud jednotka má pouze jednu štěrbinu a specifikujeme u souboru předponu :D1:.

Poznámka:

Uživatelé, kteří mají připojeny pružné disky pomocí modulu DISC 2, mohou zaměnit implicitní hodnotu předpony specifikace souboru na některou z hodnot :D0:, :D1:. Tuto změnu je možno provést buď provedením speciálního programu, který bude dodáván na systémové disketě nebo (méně čistě) přímým přepsáním hodnoty příslušné proměnné v paměti.

2.2 . Specifikace jména souboru a jeho typu

Řetězcem (jméno) pojmenováváme soubor a řetězcem (přípona) specifikujeme jeho typ. Oba řetězce musí být identifikátory, tj. musí se skládat jen z alfanumerických znaků (písmena, číslice a znak "O") a nesmí začínat číslicí.

Řetězec (jméno) je identifikátor délky nejvýše šest. Pokud jej při specifikaci souboru neuvedeme, je vzato jako implicitní hodnota šest mezer.

Řetězec (přípona) je identifikátor délky nejvýše tři. Pokud jej při specifikaci souboru neuvedeme, jsou vzaty jako implicitní hodnota tři mezery.

V operačním systému AMOS má speciální význam přípona **OBJ**, které se používá k označování tzv. object-kódů, tj. souborů obsahujících přeložené programy.

2.3. Specifikace spoolovaných a magnetofonových souborů

U výstupních magnetofonových souborů a výstupních souborů na zařízeních :CO:, :LP: a :PO: je možno uvést ve specifikaci souboru ještě dodatek tzv. (dodatek).

Řetězec (dodatek) má tvar buď:

(S) Tento (dodatek) ve specifikaci souboru, vystupujícího na některé zařízení :CO:, :LP;, :U0;, :U1 a :PO: předepisuje, že výstup do tohoto souboru má být spoolován

nebo

((číslo)) tento (dodatek) ve specifikaci výstupního magnetofonového souboru udává velikost jeho bufferu v sektorech.

Je-li řetězec (dodatek) uveden v takovém jméně souboru, kde jeho uvedení není přípustné, je hlášena chyba.

Nespecifikujeme-li při vytváření magnetofonového souboru velikost bufferu, bere se implicitní hodnota 16 sektorů.

Spoolovacím zařízením je implicitně paměť (:MM:) v případě, že není připojena jednotka pružných disků.

Je-li tato jednotka připojena, je při inicializaci systému nastaveno jako spoolovací zařízení :D0:.

Poznámka:

Uživatelé, kteří budou mít jednotku pružných disků připojenu, budou moci předefinovávat spoolovací zařízení (přípustné jsou pouze obě štěrbiny pro pružné disky a paměť) pomocí systémového programu.

2.4. Existence souboru

V popisech vstupních podmínek pro příkazy OS AMOS budeme pro stručnější a přesnější vyjadřování používat těchto dvou pojmů:

soubor musí existovat znamená, že pro soubor musí platit jedna z podmínek:

- soubor je paměťový a má záznam v tabulce souborů
- soubor je diskový a má záznam v direktoráři na pružném disku
- soubor je vnější

soubor nesmí existovat znamená, že pro soubor musí platit jedna z podmínek:

- soubor je paměťový a nemá dosud záznam v tabulce souborů
- soubor je diskový a nemá záznam v direktoráři na pružném disku
- soubor je vnější.

3. Tvar příkazů operačního systému AMOS

Většina příkazů (výjimkou jsou volání některých utilit) operačního systému AMOS má tvar:

(příkaz) (mezery) (seznam souborů) ; (params)

kde

(příkaz)	jméno utility nebo uživatelského programu
(mezery)	alespoň jedna mezera
(seznam souborů)	seznam specifikací souborů, oddělených čárkami (jsou-li pro daný příkaz definovány implicitní hodnoty pro některé soubory, není nutné specifikace těchto souborů vypisovat)
(params)	řetězec specifikující parametry volaného programu resp. utility, program si je musí zpracovat sám, od systému dostane v systémové proměnné PARAMS ukazatel na středník ve vstupním bufferu

Příkazy OS se zadávají z klávesnice, celková délka příkazu nesmí přesáhnout 73 znaků. Příkaz musíme "odeslat" stiskem klávesy CR, do té doby můžeme jeho text editovat pomoci posunů kurzoru a kláves IC a DC, stejně jako při zadávání příkazů Basicu. Současným stlačením kláves CTRL a X můžeme dosud napsanou část příkazu zrušit, kombinací kláves CTRL a "nadtrhovátko" (nachází se bezprostředně nad klávesou CR) navíc smažeme obrazovku. Pro zadávání příkazů platí tedy totéž, co platí pro vstup ze zařízení :CI:.

Pokud je v příkazech OS třeba zadat číslo, je možné ho zapsat desítkově (dekadicky), šestnáctkově (hexadecimálně) i dvojkově (binárně). Šestnáctkový zápis čísla musí začínat číslicí a být ukončen písmenem "H", dvojkový zápis čísla musí být ukončen písmenem "B". Tedy např. 163, 0A3H a 1010001B jsou správné zápisy téhož čísla. Naproti tomu zápisy A3H, A3 a 101111201B nejsou správné. Za zápisem čísla musí být oddělovač, např. mezera (0BBH je zápis hexadecimálního čísla).

Interpret příkazů operačního systému AMOS rozlišuje příkazy volání utilit, od příkazů volání uživatelských programů podle prvních tři písmen příkazu. Chceme-li, aby bez ohledu na toto pravidlo byl příkaz považován za volání uživatelského programu, napíšeme jako první znak příkazové řádky znak mínus "-". Tento znak se zobrazí na obrazovce, není však částí příkazu, pouze indikuje, že jde o volání uživatelského programu.

Poznámka:

Syntaxe zde popisována není přesným popisem jazyka příkazů, který akceptuje interpret příkazů OS AMOS, ale doporučením, jak příkazy psát.

Interpret ve skutečnosti akceptuje i poněkud volnější syntaxi a nevyžaduje ve všech případech oddělení řetězce (příkaz) od zbytku volací věty mezerami. Užívání těchto odchylek od zde doporučené syntaxe, může však být, zvláště pro začátečníka, nebezpečné.

3.1. Příkazy volání utilit operačního systému

Shoduje-li se řetězec (příkaz) na prvních třech místech se jménem některé utility operačního systému, chápe se příkaz jako volání příslušné systémové utility. Zbývající znaky řetězce (příkaz) až do prvního nepísmenového znaku nejsou významné.

Počet parametrů jednotlivých příkazů, jejich význam a implicitní hodnoty budou uvedeny u každé utility zvlášť.

3.2. Příkazy volání uživatelských programů

Neshodují-li se první tři písmena příkazu s žádným jménem systémové utility nebo jestliže jako první znak na příkazové řádce vstoupil znak "-", je příkaz pokládán za volání uživatelova programu.

Řetězec (příkaz) musí v tomto případě být specifikací souboru. Tento soubor musí existovat a musí obsahovat object-kód, může však být na libovolném zařízení (např. na magnetofonu). Bez ohledu na to, jakou příponou specifikujeme (nebo zda ji nespecifikujeme vůbec) se vždy pracuje s příponou OBJ.

Operační systém nejprve tento object-kód naalokuje, předá programu odkazy na soubory uvedené ve volací větě, uloží do systémové proměnné PARAM odkaz na počátek textového bufferu obsahujícího parametry a nakonec odstartuje uživatelský program. Při alokaci programu se jeho startovací adresa zapamatuje do systémové proměnné, její hodnota se pak bere jako implicitní při spouštění programů příkazem RUN (viz 4.8.3.).

V seznamu souborů se uvádějí soubory, se kterými program pracuje. Takto specifikované soubory pak odpovídají souborům v programech podle následujících pravidel:

- u pascalských programů odpovídají tyto soubory i co do pořadí parametrům programu (jejichž seznam se uvádí v hlavičce programu)
- u assemblerských programů odpovídají tyto soubory i co do pořadí souborům definovaným ve zdrojovém textu programu pseudoinstrukcemi DF.

Specifikujeme-li v příkazu volání uživatelského programu jiný počet souborů, než kolik jich bylo definováno ve zdrojovém programu, nedojde k chybě. Je-li specifikovaný seznam souborů příliš krátký, doplní se automaticky implicitními hodnotami na správnou délku. Je-li seznam příliš dlouhý, uplatní se pouze jeho začátek.

V příkazech volání uživatelských programů je implicitní hodnota všech souborů :CO: s výjimkou souboru, který odpovídá pascalskému standardnímu vstupnímu souboru INPUT, pro který je implicitní hodnotou :CI:.

Například:

A	Spuštění programu, který je v souboru A.OBJ
A B,C	Spuštění programu, který je v souboru A.OBJ, s předáním odkazů na soubory B a C
:MG:A C;123	Spuštění programu obsaženého v magnetofonovém souboru :MG:A.OBJ, předání odkazu na paměťový soubor C a předání parametru za středníkem
APROX :CI:,,:MG:BBA	Spuštění programu, který je v souboru APROX.OBJ, a předání odkazů na tři soubory: první soubor vstup z klávesnice, druhý je výstup na obrazovku, třetí je magnetofonový se jménem BBA
-REN :D0:A,H	Spuštění programu REN, který je v souboru REN.OBJ, a předání odkazů na soubory :D0:A a H.

4. Seznam příkazů volání utilit

Pro odlišení uvádíme první tři významná písmena názvu utility velkými písmeny.

Ve všech příkladech předpokládáme, že nebyla změněna implicitní hodnota předpony ve specifikaci souboru a je tedy (tak jako standardně) :MM:.

4.1. Manipulace se soubory na systémové úrovni

4.1.1. Výpis tabulky souborů

CATalog

Výpis tabulky souborů. Jednotlivé položky tabulky se vypisují ve tvaru:

(specifikace souboru) a b

kde a,b jsou hexadecimální zápisy pořadových čísel prvního a posledního sektoru, které soubor v paměti zabírá.

Za poslední položkou tabulky symbolů je ve výpisu uveden řádek udávající informace o rozdělení paměti, která není obsazena soubory. Má tvar:

FREE f / s

kde **f** je počet volných sektorů a **s** počet sektorů v oblasti vyhrazené příkazem SPACE pro uživatelské programy, obě čísla jsou vypsána hexadecimálně.

Výpis tabulky na obrazovku lze pozastavit dvojici kláves CTRL + S a opět spustit dvojicí kláves CTRL + Q.

Akci, kterou provádí utilita CATalog, jde vyvolat také povelem (viz odstavec 5.).

4.1.2. Kopírování souborů

COPy f1, f2

Zkopíruje soubor f1 do souboru f2, přitom se vykonají všechny potřebné formátové konverze a pomocné akce. Protože specifikace souboru obsahuje (byť třeba jen implicitně) i údaj o zařízení, na kterém se soubor nalézá, stačí v OS AMOS jediná utilita COPY obhospodařit všechny vstupně/výstupní operace se soubory.

Soubor f1 musí existovat, implicitně se bere :CI:, soubor f2 nesmí existovat, implicitně se bere :CO:.

Například:

COP :MG:A,B	načtení souboru A z magnetofonu do paměti (dostane jméno B)
COP A,:MG:A	zápis souboru A na magnetofon (se jménem A)
COP A,:MG:F(32)	zápis souboru A na magnetofon (se jménem F a s bufferem o velikosti 32 sektorů)
COP A,:LP:	výstup souboru A na tiskárnu
COP A,:CO: resp. zkráceně COP A nebo COP A,	výstup souboru A na obrazovku
COP :D0:A,:D1:A	zkopírování souboru A z pružného disku ve štěrbině č.0 do stejnojmenného souboru na pružný disk ve štěrbině č.1.
COP A,:D0:PRG	zkopírování paměťového souboru A na pružný disk ve štěrbině č. 0. pod názvem PRG
COP :D0:A,:D0:A.H	zkopírování textového souboru A na pružném disku ve štěrbině č.0. do souboru A.H (tedy hexa-decimálního) na stejném pružném disku
COP :RI:A,:LP:	zkopírování souboru z děrné pásky na tiskárnu

4.1.3. Přejmenování souborů

REName f1, f2

Přejmenuje paměťový nebo diskový soubor f1 na soubor f2, atributy souboru přitom zachovává.

Soubor f1 musí být paměťový nebo diskový a musí existovat, soubor f2 nesmí existovat. Předpona ve specifikaci souboru f2 se ignoruje a je násilně přepsána na stejnou předponu, jaká byla specifikována u souboru f1.

Implicitní specifikace ani jednoho ze souborů není přípustná.

Například:

REN AHA,AHOJ	přejmenování paměťového souboru AHA na AHOJ
REN :D0:S,T	přejmenování diskového souboru S na T.

4.1.4. Zrušení paměťového nebo diskového souboru

DELete f1

Odstraní paměťový soubor f1 z tabulky souborů resp. diskový soubor z adresáře na disku a tak uvolní sektory, které soubor v paměti resp. na disku zabíral.

Soubor f1 musí být paměťový nebo diskový a musí existovat (tj. mít v tabulce souborů resp. v adresáři na disku svůj záznam).

Implicitní specifikace souboru není dovolena.

Například:

DEL A	zrušení paměťového souboru A
DEL :D0:A	zrušení diskového souboru A na disku ve štěrbině č.0.

4.1.5. Připojení jednoho souboru k druhému

APPend f1 , f2

Připojí obsah souboru f2 do souboru f1 za jeho konec, soubor f2 zůstane zachován.

Soubor f1 musí být paměťový nebo diskový a musí existovat. Jeho implicitní specifikace není povolena.

Soubor f2 musí existovat, může jít o soubor na libovolném zařízení (kromě výlučně výstupních). Implicitní specifikace souboru f2 není dovolena.

Například:

APP A,B	připojení obsahu souboru B na konec souboru A
APP A,:MG:GG	připojení obsahu magnetofonového souboru GG na konec souboru B
APP :D0:A,:D1:B	připojení obsahu diskového souboru B z disku ve štěrbině č. 0. na konec souboru A na disku ve štěrbině č. 1.

4.1.6. Vypsání adresáře pružného disku

DIRectory č.1.

Vypíše na obrazovku adresář souboru na pružném disku, který je právě zasunut ve štěrbině č.1. Implicitní hodnota parametru č.1. je nula.

Formát výpisu je obdobný jako u výpisu tabulky souborů příkazem CATalog. Velikost souborů se udává také v sektorech o délce 256 bytů. Vypisuje se také velikost volného prostoru na disketě.

Výpis adresáře na obrazovku lze pozastavit dvojicí kláves CTRL + S a opětovně spustit dvojicí kláves CTRL + Q.

Je-li na pružném disku nalezen soubor se jménem INFORM a příponou začínající písmenem H, pak se z prvního takového souboru vypíše jeho prvních 256 znaků ještě před vlastním výpisem adresáře. To uživatelům umožňuje označovat si diskety.

Například:

DIR 0	výpis adresáře diskety ve štěrbině č.0.
DIR 1	výpis adresáře diskety ve štěrbině č.1.

4.2. Editace textových souborů

O způsobu práce se systémovým programem editor a o jeho příkazech pojednává třetí kapitola manuálu, zde uvádíme pouze způsoby jeho zavolání. Editor lze zavolat, pokud jsou v paměti volné alespoň dva sektory.

4.2.1. Vytváření resp. oprava textového paměťového souboru

(editace paměťového souboru)

EDItor f1

Otevře na soubor f1 systémový program editor. Pokud soubor dosud neexistoval, vytvoří se nový prázdný soubor specifikovaného jména, pokud již existoval, otevře se editor na tento soubor.

Soubor f1 musí být paměťový textový soubor, není přípustná jeho implicitní specifikace.

Například:

EDI A	otevření editoru na soubor A

4.2.2. Vytváření souboru editací z jiného souboru

(editace obecného souboru)

TRAnsfer f1, f2

Tímto příkazem se z libovolného (tj. na libovolném zařízení) textového souboru vytváří jeho editací jiný textový soubor. Po zavolání příkazu se otevře systémový program editor na prázdný pomocný soubor, který se edituje.

Při jeho editaci je kromě obvyklých akcí editoru možno načíst na konec tohoto pomocného souboru zadaný počet vět vstupního souboru f1 a nechat vystoupit zadaný počet vět ze začátku pomocného souboru do výstupního souboru f2.

Soubor f1 musí existovat a být textový. Implicitní hodnota je :BB:.

Soubor f2 nesmí existovat a musí být textový. Implicitní specifikace souboru f2 není přípustná.

Při zavolání editoru příkazem TRAnsfer je repertoár příkazového režimu rozšířen o dva další příkazy:

I - načtení zadaného počtu vět ze souboru f1 na konec pomocného souboru (je-li soubor f1 ve stavu EOF - konec souboru - jde o prázdnou akci),
O - výstup zadaného počtu vět z počátku pomocného souboru do souboru f2.

Po zadání některého z těchto příkazů se na obrazovce objeví výzva:

"**I.. KOLIK?**" resp. "**O.. KOLIK?**"

a systém očekává reakci operátora. Ten má tři možnosti, jak odpovědět:

- Stiskem klávesy home zrušit tento příkaz a pokračovat v editaci pomocného souboru v nezměněném stavu.
- Specifikovat s kolika větami se má akce provést. Zadat můžeme nejvýše třímístné číslo. Číslo je zadáno až v okamžiku jeho odeslání klávesou CR, do té doby je možné jej editovat v tříznakovém bufferu.
- Stiskem klávesy CR nechat provést požadovanou akci se stejným počtem vět, jaký byl použit při poslední akci (při první akci je použita implicitní hodnota 30).

Po provedení této akce se opět otevře nad požadovaným způsobem změněným pomocným souborem editor (máme stále k dispozici i příkazy I a O).

Zadáme-li příkaz E pro opuštění editoru, na obrazovce se objeví dotaz

"**E.. ZBYTEK VYPUSTIT?**"

a systém očekává reakci operátora. Ten má tři možnosti, jak odpovědět:

- Stiskem klávesy home zrušit zadaný příkaz a vrátit se do editace pomocného souboru.
- Odeslat znak A, tím se práce editoru ukončí a do výstupního souboru f2 se dokopíruje obsah pomocného editačního souboru.
- Odeslat znak N, tím se práce editoru ukončí a do výstupního souboru f2 se dokopíruje nejprve obsah pomocného editačního souboru a za něj i dosud "nezpracovaná" část vstupního souboru f1.

Znaky A resp. N musí uživatel odeslat klávesou CR, do té doby je lze editovat v bufferu délky 1.

Poznámka:

Pokud je požadováno provedení příkazu I s tak velkým počtem vět souboru f1, že by se pak pomocný editační soubor "nevešel" do aktuálního volného prostoru pro soubory, načte se ze souboru f1 jen tolik bytů, kolik je možné. Je to jediný případ, kdy příkaz TRA "přetrhne" věty vstupního souboru. Je dobré se takové situaci vhodnou volbou počtu čtených vět a včasným vyprazdňováním pomocného souboru příkazem O vyhnout. Dojde-li k ní přece, je vhodné si před další prací "přetržené" místo prohlédnout.

Pokud dojde k přeplnění paměti a soubor f2 je paměťový, může opuštění editoru příkazem E trvat velmi dlouho.

Například:

TRA :BB:,:MG:A resp. zkráceně	
TRA ,:MG:A	vytváření magnetofonového souboru A editorem
TRA :MG:A,:MG:B	aktualizace magnetofonového souboru (vstup ze souboru A, výstup do souboru B)
TRA :MG:A,:LP:	výstup některých částí (případně aktualizovaných) magnetofonového souboru A na tiskárnu
TRA :D0:A,:D1:B	vytváření diskového souboru B (na disku ve štěrbině č. 0.) editací z diskového souboru A (na disku ve štěrbině č.1.).

4.3. Příkazy volání překladačů

Na tomto místě uvádíme pouze příkazy pro volání překladačů. Oběma překladačům lze ve volací větě předat celou řadu parametrů a tak řídit způsob překladu, popis těchto parametrů naleznete vždy v kapitole věnovaná příslušnému překladači.

4.3.1. Překlad assemblerského programu

ASSembler f1, f2, f3 ;(params)

Zavolá překladač assembleru. Soubor f1 musí obsahovat zdrojový program, do souboru f2 se generuje object-kód a do souboru f3 se vytváří listing o překladu.

Parametry, které je možno překladači zadat v řetězci (params), jsou popsány v kapitole věnovaná assembleru.

Soubor f1 musí existovat. Jeho implicitní specifikace není povolena.

Pokud soubor f2 není paměťový, pak nesmí existovat. Pokud soubor je paměťový, pak při každém volání překladu se automaticky smaže případný "starý" object-kód téhož jména, pokud v tabulce souborů již existoval.

Přípona souboru f2 je ignorována a je násilně přepsána na standardní příponu object-kódů OBJ.

Jako implicitní hodnota specifikace souboru f2 se bere hodnota jména souboru f1, doplněná předponou :MM: a příponou OBJ.

Soubor f3 nesmí existovat, implicitní hodnota jeho specifikace je :CO:.

Pokud chceme vytvářet listing do paměťového souboru, musíme pomocí parametru překladu Z rezervovat pro něj místo (viz popis assembleru).

Pokud jsou při překladu assemblerského programu nalezeny chyby, vypisují se chybné řádky a čísla nalezených chyb na obrazovku i v případě, že listing je vytvářen na jiné zařízení.

Po výpisu chybového hlášení vypíše překladač na obrazovku nabídku:

(home)/CR/oprava

a očekává reakci operátora, ten má tři možnosti:

- Stiskem tlačítka home ukončíme překlad.
- Stiskem klávesy CR požadujeme, aby kompilátor chybný řádek ignoroval a pokračoval dále v překladu s tím, že po překladu nevznikne object-kód.

- Třetí možností je napsat nový text chybné řádky a pokračovat dále v překladu, jako by vstupoval zdrojový program s provedenou opravou.

V tomto třetím případě je však třeba mít na paměti, že oprava se v souboru se zdrojovým programem neprovedla.

Tato třetí možnost se dá s výhodou použít v situaci, kdy často překládáme dlouhý zdrojový program, v němž chceme na několika místech měnit konstanty. Na těchto řádcích uděláme úmyslně chyby a opravíme je na správné hodnoty až při samotném překladu.

Všechny uživatelské programy, které mají pracovat pod OS AMOS, by měly končit příkazem JMP CCP, aby se po skončení práce vrátily do OS.

Například:

ASS A	spuštění překladače na zdrojový program v souboru A
ASS A,,A.LST;Z(5)	spuštění překladače na zdrojový program v souboru A, vytváření listingu do paměťového souboru A.LST
ASS :MG:A.ASS	spuštění překladače na zdrojový program v magnetofonovém souboru A.ASS, object-kód je generován do souboru :MM:A.OBJ
ASS :D0:A,:D0:A	spuštění překladače na zdrojový program v souboru A na disku ve štěrbině č.0.; object-kód se generuje do souboru A.OBJ na témže disku, listing vystupuje na obrazovku.

4.3.2. Překlad pascalského programu

PAScal f1, f2, f3 ;(params)

Zavolá překladač Pascalu. Soubor f1 musí obsahovat zdrojový text pascalského programu, do souboru f2 se generuje object-kód, do souboru f3 se vytváří listing.

V řetězci (params) je možno předat překladači parametry pro překlad, blíže o nich v kapitole o překladači Pascalu.

Soubor f1 musí existovat, jeho implicitní specifikace není dovolena.

Pokud soubor f2 není paměťový, pak nesmí existovat. Pokud soubor je paměťový, pak při každém volání překladu se automaticky smaže případný "starý" object-kód téhož jména, pokud v tabulce souborů již existoval.

Přípona souboru f2 je ignorována a je násilně přepsána na standardní příponu object-kódů OBJ. Jako implicitní hodnota specifikace

souboru f2 se bere hodnota jména souboru f1, doplněná předponou :MM: a příponou OBJ.

Soubor f3 nesmí existovat, implicitní se bere hodnota :CO:. Chceme-li generovat listing do paměti, je třeba na něj vyhradit místo parametrem překladu Z (viz popis překladače Pascalu).

Například:

PAS A	překlad zdrojového programu obsaženého v souboru A
PAS A,,A.LST;Z(20)	překlad zdrojového programu obsaženého v souboru A, generování listingu do paměťového souboru A.LST, pro tento soubor se rezervuje 20 sektorů
PAS :MG:A	překlad zdrojového programu obsaženého v magnetofonovém souboru :MG:A
PAS :D0:A,,:D1:A.LST	překlad zdrojového programu obsaženého v diskovém souboru :D0:A, generování listingu do diskového souboru :D1:A.LST

4.4. Zavolání Basicu

BASic

Tento příkaz je ve verzi 4 OS AMOS připraven pro volání připravované nové verze Basicu G (pozor, současný modul Basic G, vyráběný v roce 1987 nesmí být v počítači současně s moduly OS AMOS !!!), která bude moci pracovat pod OS AMOS. Podrobný popis efektu tohoto příkazu bude popsán v dokumentaci připravovaného Basicu.

4.5. Nastavení parametrů pro tiskárnu

PRInter č1, č2.

Tento příkaz nastavuje formát tisku při výstupu na zařízení :LP:, tj. na tiskárnu. Číslo č1 udává počet znaků na řádce, číslo č2 počet řádek na stránce.

Po startu systému platí tyto implicitní hodnoty:
132 znaků na řádce a 65 řádek na stránce.

Například:

PRI 60,30	výstup s 30 řádky po 60 znacích na jedné stránce

4.6. Archivace aktuálního stavu systému na magnetofon

Musíme-li z nějakého důvodu přerušit práci se systémem a při tom chceme pokračovat později z přesně stejného stavu, můžeme aktuální stav systému archivovat jako jeden celek nemusíme nahrávat na pásku jeden soubor po druhém.

4.6.1. Zápis aktuálního stavu na magnetofon

SAVe

Nahraje aktuální stav souborů a některých systémových proměnných na magnetofon. Na obrazovce se objeví výzva **"!"**, připravíme magnetofon k nahrávání a po stisku libovolné klávesy se začíná nahrávat. Po dokončení nahrávky systém nabízí verifikaci (protože v tomto případě jde vždy o velká množství dat, důrazně uživateli doporučujeme verifikaci vždy provést raději dvakrát).

Příkaz **SAV**e nahrává aktuální stav systému na pásku v jiném formátu než obvyklé nahrávání souboru na magnetofon, tento záznam nemá hlavičku a tvoří ho jen jediný blok.

4.6.2. Obnovení stavu systému uloženého na magnetofon příkazem SAVE

LOAd

Přečtete z magnetofonu stav systému nahraný příkazem SAVE a obnoví zcela jeho původní stav.

Při čtení přejede všechny soubory, která jsou na magnetofonové pásce případně před tímto archivačním záznamem (nevypisuje však jejich hlavičky jako v případě čtení souboru). Pokud však máme na pásce více záznamů archivace příkazem SAVE, musíme "najet" před ten "správný" manuálně. Proto je lepší neukládat na jednu magnetickou pásku více než jednu archivaci.

Pokud při čtení dojde k chybě, vypíše se na obrazovku inverzně hláška "ZNOVU" a očekává se opakované čtení z magnetofonu.

4.7. Zavolání Debuggeru

DEBugger (příkaz volání uživat. programu)

Zavolá systémový program debugger, který slouží k ladění programů ve strojovém kódu a k demonstraci funkce mikroprocesoru 8080.

Je-li příkaz zadán bez parametru, jde o prosté vyvolání debuggeru.

Parametr (příkaz volání užívat. programu) může být specifikován dvojím způsobem. Buď jej můžeme specifikovat tak, jak byl popsán v paragrafu 3.2., nebo tak, jak je popsán u příkazu RUN viz paragraf 4.8.3. První způsob je standardní, druhý slouží uživatelům, kteří potřebuji simulovat explicitně alokované programy.

První způsob zavolání debuggeru se používá k simulaci výpočtů dosud nealokovaných programů, v tomto případě operační systém nejprve provede všechny systémové akce, které předcházejí spuštění programu (předání odkazů na soubory a případný textový parametr, uvedený za středníkem, a alokování programu) a poté zavolá debugger s dosazenou startovací adresou programu.

Druhou možnosti je zavolání s parametrem jako u příkazu RUN viz paragraf 4.8.3., přičemž je povolen pouze jeho tvar se znakem dolar.

V tomto případě se předají programu odkazy na soubor a textový parametr a zavolá se debugger s implicitní startovací adresou, nastavenou při poslední alokaci object-kódu. Pokud tato startovací adresa uživateli nevyhovuje, je možné ji změnit v debuggeru příkazem A.

Například:

DEB	zavolání debuggeru
DEB A,B,C;123	zavolání debuggeru na program A, nejprve se mu předají odkazy na souboru B a C a text obsažený v příkazu za středníkem
DEB $,A,C,H	předání odkazů na souboru A, B a H a zavolání debuggeru s implicitní startovací adresou nastavenou při posledním alokování programu.

4.8. Příkazy pro explicitně alokované programy

V operačním systému AMOS se ve standardních situacích ani programátor v assembleru nemusí starat o konkrétní umístění svých programů a dat v paměti. V některých aplikacích je však nutné, aby programátor měl možnost alokaci svých programů přímo řídit. K práci s takovými programy je určena následující skupina příkazů. K jejich úspěšnému užívání je třeba již více zkušeností.

4.8.1. Vyhrazení oblasti pro explicitně alokované programy

SPAce č1

Vyhrazuje na konci paměti RAM, kterou má k dispozici OS AMOS oblast, do které nemohou zasáhnout soubory. Tuto oblast budeme v dalším nazývat oblasti SPACE. Do této oblasti pak uživatel může umísťovat svoje programy, které chce chránit před přepsáním soubory.

Číslo č1 může být specifikováno jak dekadicky, tak i hexadecimálně a binárně, vyhradí se oblast velikosti (č1+1) sektorů. Oblast SPACE obsahuje tedy vždy alespoň jeden sektor, doporučujeme uživatelům ROM-verze OS AMOS, aby svými programy neobsazovali poslední sektor této oblasti, budou jej využívat některé systémové programy, s jejichž vývojem se počítá v budoucnosti.

O velikosti vyhrazené oblasti se lze dovědět jak příkazem CATalog (viz 4.1.1.), tak i jemu odpovídajícím povelem (viz 5.1.).

Například:

SPA 1AH	vyhrazení oblasti 27 sektorů
SPA 0	zrušení oblasti SPACE
SPA 28	vyhrazení 29 sektorů

4.8.2. Alokace uživatelských programů

LOCate f1 ;(posunutí)

Příkaz slouží k alokování object-kódu ze souboru f1 do paměti. Pokud je v alokovaném object-kódu souboru obsažena tabulka symbolů, je také umístěna do paměti.

Soubor f1 musí obsahovat object-kód, může být na libovolném zařízení (pochopitelně kromě výlučně výstupních). Implicitní specifikace souboru f1 není povolena.

Bez ohledu na specifikovanou příponu se pracuje se souborem se standardní příponou pro object-kódy OBJ.

Standardní použití příkazu je bez specifikace parametru (posunutí). Příkaz pak alokuje program uložený v object-kódu f1 na místo v paměti, kam je při překladu "naorgován". V tomto případě se startovací adresa alokovaného programu zapamatuje v systémové proměnné a slouží pak jako implicitní hodnota v příkazu RUN, kterým se spouštějí již naalokované programy.

Specifikujeme-li parametr (posunutí), pak také dochází k alokaci kódu programu do paměti, avšak obecně nikoli na místo, kde může počítat. Parametr (posunutí) musí být specifikován jako číslo, jeho hodnota se při alokaci programu přičítá ke každé zaváděcí adrese. Je-li tedy nenulová je kód programu umístěn na místo, kde nemůže počítat. Tento tvar příkazu slouží např. k tomu, abychom, vyvíjíme-li pod OS AMOS program pro jiný počítač, mohli získat binární tvar programu orgovaný do míst, kde je na IQ 151 paměť ROM.

Pokud je požadovaná alokace mimo oblast SPACE, pak v případě, že alokace proběhla jinak úspěšně, je hlášena chyba číslo 33, která je vlastně varováním, že program nepůjde spustit příkazem RUN bez zvětšení oblasti SPACE.

Například:

LOC A	alokování object-kódu obsaženého v souboru A.OBJ
LOC :MG:A.OBJ	alokování object-kódu obsaženého v magnetofonovém souboru A.OBJ

4.8.3. Spuštění již alokovaných programů

RUN (start. adresa) ,(seznam souborů) ;(params)

Příkaz spustí již alokovaný program od specifikované startovací adresy, předtím mu systém předá odkazy na specifikované soubory a odkaz na začátek bufferu obsahujícího řetězec (params), pokud byly příslušné parametry příkazu RUN specifikovány.

Startovací adresa musí být specifikována buď explicitně jako číslo (je možné je zadat dekadicky i hexadecimálně), nebo můžeme napsat znak dolar a tím požadovat použití její implicitní hodnoty. Implicitní hodnota startovací adresy je vyzvednuta ze systémové proměnné, kam byla uložena při posledním alokování programu (ať již příkazem LOCATE nebo příkazem pro volání uživatelského programu - viz 3.2.).

Hodnota startovací adresy musí patřit do oblasti SPACE, jinak dojde k chybě. Uvedení nesprávné startovací adresy může mít nepredikovatelné důsledky.

Například:

RUN $	spuštění naposledy alokovaného programu
RUN 74AFH,A,B	předání systémových odkazů na soubory A a B a následné spuštění programu od adresy 74AFH

4.9. Příkazy pro opuštění OS AMOS

4.9.1. Skok do monitoru

MONitor

Vyvolání monitoru IQ 151. Monitorským příkazem >R se pak vrátíme do OS AMOS.

4.9.2. Návrat do programu, z kterého byl OS AMOS zavolán

BYE

Návrat do programu, z kterého byl OS AMOS zavolán. Do OS AMOS se můžeme vrátit zpět na jeho horký (přitom se zachovají soubory i stavové proměnné operačního systému) nebo studený start (s úplnou inicializací systému). Užívá se např. při spolupráci s BASICEM 6 viz odstavec 6.

4.10. Příkaz definovaný uživatelem

USEr

Význam tohoto příkazu může definovat uživatel tím, že předepíše adresu podprogramu, který se má v případě zavolání příkazu provést. Pokud byl příkaz zavolán, aniž by byla adresa výkonného podprogramu definována, dojde k chybě.

Nepředpokládá se, že by to byla obvyklá uživatelská akce, spíše budou této možnosti využívat systémové programy dodávané zvlášť. Jsou-li při startu systému přítomny diskety, nastavuje se jako USEr přečtení prvního sektoru nulté stopy do paměti na adresu 7000H a jeho odstartování (číslo štěrbiny lze v příkazu zadat).

5. Povely OS AMOS

Klávesa BR (break) generuje signál přerušení mikroprocesoru. V OS AMOS je toho využito pro zadávání povelů.

Protože je klávesa BR na počítači v těsné blízkosti "uživatelských" kláves F1 - F5 a je tedy značná pravděpodobnost jejího neúmyslného stlačení, zadávají se všechny povely OS AMOS současným stlačením několika kláves. Povel je zadán v okamžiku stlačení klávesy BR a podle toho, které další klávesy jsou v této době stlačeny, se pozná o který povel jde. Proto je třeba při zadávání povelu nejprve stlačit všechny "doplňkové" klávesy a držet je stlačené až do okamžiku stisku klávesy BR.

Povel je akceptován vždy (tedy i během práce nějakého programu), když nemá procesor právě zakázané přerušení. Proto je nutné vždy před zadáním povelu dobře zvážit, zda jeho provedení nebude mít na práci právě probíhajícího programu nežádoucí vliv.

5.1. Ukončení právě probíhajícího programu - Break

Shift + CTRL + BR

Povel ukončí práci programu, který právě probíhá, a řízení se vrátí do interpretu příkazů OS AMOS. Na obrazovku se vypíše BREAK adresa, uzavřou se všechny soubory, na obrazovce se vypíše výzva "=" a je očekáván příkaz OS AMOS. Tento povel používáme např. tehdy, máme-li důvodné podezření, že se program zacyklil, nebo když již nechceme čekat na jeho dokončení (např. víme-li již, že skončí špatně).

Pokud právě počítal pascalským program, je tímto povelem uměle vygenerována běhová chyba, která způsobí výpis standardního pascalského post-mortem-dumpu. To může podstatně pomoci při ladění pascalských programů.

5.2. Přepínání hustého a řídkého řádkování na obrazovce

Shift + FA + BR	zapíná husté řádkování
FA + BR	zapíná řídké řádkování

5.3. Přepínání zobrazovací grafiky

Tyto povely mají význam pouze, je-li zasunut modul GRAFIK, jinak jsou ekvivalentní prázdné akci.

Shift + FB + BR	zapne zobrazování obsahu grafické paměti modulu GRAFIK
FB + BR	vypne zobrazování obsahu grafické paměti modulu GRAFIK

5.4. Výpis tabulky souborů

CTRL + BR

Povel způsobí výpis tabulky souborů na obrazovku v přesně stejném formátu jako příkaz CATALOG.

Je zde však jeden rozdíl. Zadáme-li povel v okamžiku, kdy interpret příkazů OS AMOS čeká na příkaz (na obrazovce svítí nápovědný znak "="), je efekt povelu naprosto stejný jako kdybychom zadali příkaz CATALOG. V tabulce souborů jsou totiž v tomto okamžiku pouze neprázdné paměťové soubory.

Povel však vypisuje okamžitý stav tabulky symbolů, a protože může být zadán kdykoli, mohou se v tabulce vyskytovat i soubory vnější. Z počtu volných sektorů, které se vypisují za slůvkem FREE můžeme také vidět, kolik paměti si pro sebe zabral právě probíhající program.

Výpis tabulky symbolů na obrazovku lze, jako každý jiný výstup na obrazovku, pozastavit kombinací kláves CTRL + S a opět spustit kombinací CTRL + Q.

6. Spolupráce OS AMOS s Basicem 6

Operační systém AMOS můžeme používat i současně s modulem Basic 6. Spolupráce Basicu s modulem AMOS/Pascal nemá praktický význam. Pomocí modulu AMOS/Assembler můžeme však vytvářet, překládat, opravovat a simulovat assemblerské podprogramy, která chceme volat z basického programu. Můžeme také využít z basických programů služeb OS AMOS pro manipulace se soubory.

Před zavoláním OS AMOS z Basicu musíme pro něj rezervovat paměť basickým příkazem CLEAR. Operační systém AMOS obsadí celou oblast USR, rezervovanou tímto příkazem. Je třeba dbát no to, aby tato oblast byla dostatečně velká, alespoň 11 sektorů. Je-li menší, je hlášena chyba 99. Tato oblast nesmí také být větší než 30Kbytů.

OS AMOS při inicializaci uklidí proměnné Basicu. Z OS AMOS se zpět do Basicu vrátíme příkazem BYE. Přitom se opět obnoví obsah jak proměnných interpretů Basicu, tak i proměnných programů v Basicu a uschovají se systémové proměnné OS AMOS. Nezmenšíme-li oblast USR, pak při opětovném zavolání OS AMOS horkým startem se můžeme vrátit do původního stavu souborů v OS AMOS.

Nepotřebujeme-li zachovat soubory, ale jen alokované kódy programů v oblasti SPACE operačního systému, můžeme zmenšit oblast USR tak, aby v ní oblast SPACE zůstala zachována (jestliže však nezachováme celý prostor, který jsme pro OS AMOS vyhradili, musíme se vrátit studeným startem).

Poznámka:

Basic G, vyráběný v roce 1987 se rozrostl do adresového prostoru vyhrazeného modulům s OS AMOS. Proto nemůže být s nimi ani současně zasunut do počítače, natož aby mohl spolupracovat. Tato situace bude řešena vývojem nového modulu s grafickým Basicem, který bude zcela kompatibilní a dosavadním Basicem G a bude mít navíc možnost pracovat se soubory OS AMOS. Při tom adresová kolize bude odstraněna, protože nový Basic bude vícenásobně využívat svůj adresový prostor.

III. EDITOR

1. Role EDITORU v OS AMOS

Systémový program editor slouží k opravám a k prohlížení textových souborů. EDITOR lze otevřít pouze na textový soubor, jinak je hlášena chyba 26. Násilný pokus toto omezení obejít může vést k nepredikovatelným následkům.

EDITOR lze otevřít, dvojím způsobem. Buď příkazem EDITOR pro editaci paměťového souboru nebo příkazem TRANSFER pro editaci obecného souboru. Protože k pochopení funkce příkazu TRANSFER je nutná znalost funkce editoru, proto se jim zabýváme až v odstavci 4 této kapitoly.

2. Vyvolání EDITORu na paměťový soubor

Příkaz má tvar:

EDItor f1.

Soubor f1 musí být textový paměťový, jeho implicitní specifikace není povolena. Existuje-li již tento soubor, je editor otevřen k jeho editaci. Neexistuje-li tento soubor, je editor otevřen na nový prázdný soubor.

3. Charakteristika EDITORU

Operační systém AMOS ukládá textové soubory ve speciálním úsporném tvaru s komprimováním mezer. Tím se dosahuje významné úspory operační paměti, potřebné k uchování především zdrojových textů programů, ale také jiných informaci textového charakteru.

Textový soubor může obsahovat všechny znaky zobrazitelné na IQ 151 (včetně grafických a inverzních). Editor však používá posledních pět inverzních grafických znaků s ASCII kódy 0FBH - 0FFH (tj. inverzní znaky {|}~ ) pro svoje vlastní účely, proto je při zavolání editoru na soubor těchto pět inverzních znaků "odinverzněno".

V textových souborech, které byly vytvořeny resp. zpracovávány editorem dojde také k tomu, že inverzní a grafické znaky jsou v souboru fyzicky reprezentovány pomocí přechodů do grafiky a inverze.

Textový soubor je vnitřně členěn na věty. Velikost věty není nijak omezena. Pro uživatele je však výhodnější používat soubory a rozumnou délkou vět.

K vytváření textových souborů slouží program EDITOR. Uživatel má k dispozici dva režimy práce EDITORu: **obrazovkový** a **příkazový**. Po otevření je Editor vždy v obrazovkovém režimu. Uživatel může během opravování souboru přecházet z jednoho režimu do druhého a zpět.

Obrazovkový režim slouží k vytváření souboru, resp. k jeho jednoduchým úpravám, které lze redukovat na vkládání, záměnu, nebo nahrazování znaků.

Příkazový režim umožňuje složitější akce nad souborem, např. vyhledávání řetězců a jejich záměna jinými řetězci, přesun kusu souboru na jiné místo v souboru atd. V příkazovém režimu je iniciativa na straně počítače, uživateli se nabízí formou "menu" přípustné příkazy, které může provést. Také parametry jednotlivých příkazů se zadávají na výzvu.

Všechny akce editoru se provádějí přímo na editovaném souboru, nevytváří se jeho kopie. Toto řešení si vynutila malá paměť, která je k dispozici. Ve všech stádiích práce s editorem se na obrazovce zobrazuje skutečný stav souboru v paměti.

Na spodním okraji obrazovky se po celou dobu práce s editorem zobrazuje tzv. informační řádek, na které jsou soustředěny informace o aktuálním stavu práce EDITORU.

3.1. Informační řádek

Nejspodnější řádek obrazovky je EDITOREM využíván jako tzv. informační řádek. Má tvar:

St:X I± G± S± R± L± B± M1↕ M2↕ (hex)

Jednotlivé položky informačního řádku mají následující význam:

St: kde X je písmeno, které udává, jaký příkaz je právě prováděn, přípustné hodnoty jsou: V,J,F,S,R,Q,D,C,M,L,B.

Dalších pět položek informačního řádku jsou indikátory módů, ve kterých se EDITOR právě nachází. Je-li v položce zobrazeno + je příslušný mód zapnut, je-li zobrazeno -, je vypnut.

Po otevření editoru jsou nastaveny tyto implicitní hodnoty přepínačů: I- G- S- R- L- B+.

Další dvě položky určují polohu značek M1 a M2 vůči kurzoru. Indikátor může mít jednu z hodnot ↑ a ↓ . Znak ↑ znamená, že příslušná značka se nachází před kurzorem, znak ↓ že je za kurzorem.

Poslední položka (hex) obsahuje hexadecimální zápis počtu bytů, které ještě v paměti pro vytváření souboru zbývají.

TABULKA MÓDŮ EDITORU

Mód	Mnemonika	Význam zapnutí módu (+)
I	inverze	znaky vstupují jako inverzní
G	grafika	znaky vstupují jako grafické
S	shift-lock	tlačítka bez shiftu odpovídají malým písmenům, se shiftem velkým (je-li -, je tomu naopak - stand. režim práce IQ 151)
R	replace	znak pod kurzorem je nahrazován právě vloženým znakem (je-li -, jsou znaky vkládány do textu v místě, kde je kurzor, přičemž následující znaky až do konce věty jsou odsouvány doprava)
L	line	mezi věty souboru se na obrazovce vloží vždy jeden prázdný řádek
B	beep	stisk tlačítka je indikován akusticky

3.2. Zobrazení textu na obrazovce

Obsah souboru je zobrazen takovým způsobem, že každá věta souboru začíná na nové řádce obrazovky (je tedy předchozí řádek obrazovky případně doplněn mezerami). Chceme-li mít zvýrazněno členění souboru na věty, je možné přejít do módu L+, v němž se mezi každé věty vkládá na obrazovce jeden prázdný řádek.

Obrazovka slouží jako "okénko" do souboru, které lze po souboru posouvat dopředu a dozadu.

Blikajícím znakem je na obrazovce vyznačeno aktivní místo tzv. kurzor. Po inicializaci souboru stojí kurzor na prvním znaku souboru. Pomocí povelů a příkazů EDITORU lze polohu kurzoru v souboru měnit, stále však zůstává zobrazen na obrazovce.

Je-li právě na obrazovce zobrazen začátek nebo konec souboru, jsou zvýrazněny speciálním řádkem, skládajícím se ze samých inverzních znaků

EDITOR si v souboru pamatuje ještě dvě významné pozice, které jsou označeny značkami M1, M2. Pokud je na obrazovce právě zobrazena část souboru, obsahující některou z těchto dvou pozic, je v odpovídajícím místě do souboru uložen znak odpovídající této značce. Těmito znaky jsou inverzní složené závorky (značka M1 je zobrazena jako "{", značka M2 jako znak "}"). Tyto značky ve skutečnosti v souboru nejsou zapsány a nejde je tady smazat, ani na ně najet kurzorem.

V příkazovém režimu se využívá spodní část obrazovky nad informačním řádkem pro zadávání příkazů (menu příkazů a zadávání jejich parametrů).

3.3. Povely přepínání módů EDITORU

Módy I, G, S, R se přepínají stlačením tlačítek F1, F2, F3, F4 (pořadí odpovídá pořadí, v jakém jsou zobrazena jejich okamžitá nastavení v informačním řádku). Těchto povelů lze použít v obrazovkovém režimu kdykoliv a v příkazovém režimu jen při zadávání parametrů příkazů.

3.4. Povely Editoru

Popíšeme postupně jednotlivé povely a jim odpovídající akce.

3.4.1. Povely pro posun kurzoru

Následujícími pokyny je možné měnit polohu kurzoru v textu. Kurzor při tom stále zůstává na obrazovce, jen dochází k případným posunům "okénka" po textu příslušným směrem.

TABULKA KLÁVES PRO POHYB KURZOREM

klávesa	Akce
ç	posun kurzoru doleva po textu na obrazovce; na levém konci řádku dochází k přechodu na pravý konec řádku předcházejícího
è	posun kurzoru doprava po textu na obrazovce; na pravém konci řádku dochází k přechodu na levý konec řádku následujícího
é	pohyb kurzoru o jeden řádek na obrazovce nahoru; pokud je kurzor na začátku souboru a posun nelze provést, nestane se nic, pokud by byl kurzor na nekorektní poloze (např. "uprostřed tabelátoru") je posun doplněn nutným posunem doleva
ê	obdobný posun o jeden řádek na obrazovce dolů
ë	přesun kurzoru na začátek věty, je-li kurzor na začátku věty, přesune se na začátek věty předchozí
CR	přesune kurzor na začátek další věty; je-li kurzor na konci souboru, ukončí se věta a za ní se přidá nová "prázdná" věta (věta obsahující jednu mezeru) a kurzor se přesune na první znak této prázdné věty

Prodloužením stisku klávesy lze činnost povelu opakovat. Na pozici kurzoru mají vliv i jiné povely, jejichž hlavní poslání je však jiné.

3.4.2. Povely pro vypuštění a vkládání znaku (resp. věty)

TABULKA KLÁVES PRO EDITACI

klávesa	akce
DC	(Delete Character) - vypuštění znaku; zrušení znaku pod kurzorem a sesunutí zbytku věty o jeden znak doleva
DL	(Delete Line) - vypuštění věty; ruší zbytek věty od polohy kurzoru do konce věty. Je-li kurzor na počátku věty, zruší větu celou. Je-li kurzor na konci věty, jde o prázdnou akci.
IL	(Insert Line) - vložení věty; vloží prázdnou větu před větu, na níž je kurzor a přesune kurzor na počátek vložené věty.
IC	(Insert Character) - vložení znaku; vloží v místě kurzoru mezeru a odsune zbytek věty o jeden znak doprava.

3.4.3. Povely pro roztržení resp. spojení dvou vět

TABULKA POVELŮ PRO TRHÁNÍ A SPOJOVÁNÍ VĚT

klávesa	akce
Shift+IL	Roztržení věty; roztrhne větu na dvě v místě, kde stojí kurzor, znak pod kurzorem se stane posledním znakem první z vět.
Shift+DL	Spojení vět; spojí větu, na níž stojí kurzor s větou následující do jedné věty.

3.4.4. Vkládání, resp. náhrada znaků v souboru

Stisk znakové klávesy značí vložení příslušného znaku na místo "pod kurzorem". Efekt akce přitom záleží na nastavení módu R.

Je-li nastaven mód R+ (Replace) pak vstupujícím znakem je nahrazen znak, který doposud stál pod kurzorem.

Je-li nastaven mód R- (Insert) pak se nejprve odsune zbytek textu o jeden znak doprava a na nově vzniklé volné místo se vloží vstupující znak.

Po vložení znaku se kurzor posune o jedno místo doprava

3.4.5. Tabulátor (Shift+mezera resp. CTRL+I)

Stiskneme-li současně klávesy Shift+mezera (nebo CTRL+I), je do textového souboru vložen znak tabulátor, jehož efekt je takový, že kurzor se odsune doprava na nejbližší pozici, která je násobkem 8 (pozice se počítá od začátku řádku na obrazovce resp. na tiskárně, nikoli od začátku věty).

3.4.6. Povely k přesunu značek M1 a M2

Současný stisk Shift+F1 (resp. Shift+F2) umísťuje značku M1 (resp. M2) před znak pod kurzorem.

3.4.7. Přechod do příkazového režimu

K přechodu do příkazového režimu slouží stisk povelové klávesy F5.

3.5. Příkazový režim EDITORu

Příkazový režim EDITORu umožňuje uživateli provádět složitější úpravy souboru. Po přechodu do příkazového režimu se na obrazovce objeví řádek (prompt) s nabídkou příkazů, které jsou v dané situaci přípustné. Některé příkazy si žádají zadání dalších parametrů. Probereme nejdříve způsoby zadávání parametrů.

3.5.1. Předávání parametrů

Příkazy mohou potřebovat až tři parametry. K jejich zadání je uživatel vyzván příslušnou výzvou.

Výzva A: Specifikace úseku, jehož se příkaz týká
F: Specifikace hledaného řetězce
R: Specifikace přepisujícího řetězce

Zadání parametru musí uživatel zakončit stiskem klávesy CR. Do jejího odeslání je možné pohybovat kurzorem tam a zpět a měnit případně špatně zadaný parametr. Za platnou specifikaci parametru se bere řetězec od výzvy do polohy kurzoru při stisknutí CR. Na obrazovce se tento řetězec ukončí inverzním znakem "~". Během psaní parametru je možné klávesami F1 až F4 měnit módy práce EDITORu stejně jako v obrazovkovém režimu. Pokud je některý z parametrů zadán chybně, příslušný příkaz se neprovede, není však vydáno žádné chybové hlášení. Délka specifikace parametru je omezena délkou řádku na obrazovce.

3.5.1.2. Specifikace úseku textu - výzva A

Objeví-li se po zadání příkazu výzva "A", je uživatel žádán o specifikaci úseku textu, kterého se příkaz má týkat. Tato specifikace má tvar dvou znaků . Znakem specifikujeme začátek a znakem konec úseku souboru, kterého se příkaz týká. Pokud je poloha konce úseku blíže začátku souboru než poloha začátku úseku, nebude mít příkaz žádný efekt. Přípustné specifikace v textu jsou:

TABULKA VÝZNAČNÝCH BODŮ V SOUBORU (SPECIFIKACE PARAM A.)

Znak	Význam
<	Začátek textového souboru
>	Konec textového souboru
K	Okamžitá poloha kurzoru
1	Okamžitá poloha značky M1
2	Okamžitá poloha značky M2
B	Poloha kurzoru před posledním provedeným příkazem EDITORu (pamatuje se pouze v rámci příkazového režimu, při přechodu do obrazovkového režimu se zapomene)
V	Druhý znak zobrazený na obrazovce

U příkazu J má, jak uvidíme, smysl pouze jeden specifikátor polohy v souboru. V tomto případě není třeba druhý znak zadávat a je-li zadán, ignoruje se.

3.5.1.3. Specifikace hledaného řetězce - výzva "F:"

Objeví-li se výzva "F:", žádá EDITOR specifikaci řetězce, který se má vyhledat. Nelze vyhledávat řetězec, rozdělený hranicí vět.

3.5.1.4. Specifikace přepisujícího řetězce - výzva "R:"

Objeví-li se výzva "R:", žádá si EDITOR specifikaci řetězce, kterým má nahradit specifikované výskyty hledaného řetězce.

3.5.2. Příkazy EDITORU

Příkaz zadáváme jedním písmenem.

Povolené příkazy jsou:

Příkaz	Mnemonika	Akce	Požadované parametry
J	Jump	Přemístění kurzoru	A
L	Line	Přepínání módu zvýraznění členění souboru na věty	žádné
V	Video	Přechod do obrazovkového režimu	žádné
E	Exit	Ukončuje práci EDITORU	žádné
F	Find	Vyhledává první výskyt hledaného řetězce ve spec. úseku a umístí kurzor za jeho poslední znak. Nevyskytuje-li se tam řetězec, zůstane kurzor na místě.	A,F
S	Search	Pracuje jako opakovaný příkaz F, po nalezení každého výskytu editor nabídne možnosti dalšího pokračování: Yes/No/Stop N a S ukončení příkazu, Y hledání dalšího výskytu	A,F
R	Replace	Nahradí všechny výskyty hledaného řetězce ve spec. úseku řetězcem přepisujícím. Je-li přepisující řetězec prázdný, budou všechny tyto výskyty zrušeny. Před provedením příkazu je žádáno potvrzení. Nabídka: Yes/No/Stop Y znamená provést příkaz, N a S zrušení příkazu.	A,F,R

Příkaz	Mnemonika	Akce	Požadované parametry
Q	Question	Obdoba příkazu S pro nahrazování řetězci. Po nalezení výskytu hledaného řetězce dostává uživatel nabídku: Yes/No/Stop: Y - přepis hledaného řetězce na přepisovaný N - hledání dalšího výskytu bez přepisování S - Ukončení příkazu	A,F,R
D	Delete	Zruší celý specifikovaný úsek. Protože tato akce je velmi nebezpečná, žádá se potvrzení: Yes/No/Stop Y - zrušit N,S - nechat	A
C	Copy	Zkopíruje specifikovaný úsek před okamžitou polohu kurzoru. Kopírovaný úsek zůstane na svém původním místě zachován. Kurzor se přesune na začátek zkopírovaného úseku.	A
M	Move	Přesune specifikovaný úsek souboru před okamžitou polohu kurzoru. Na původním místě se úsek zruší, kurzor se posune na toto místo.	A
B	Beep	Přepíná režim pípání při stisku kláves.	žádné

3.5.3. Opakování naposledy prováděného příkazu

Stlačením klávesy + v okamžiku zadávání povelu je možné vyvolat naposledy prováděný příkaz. Příkaz se provádí se stejnými hodnotami parametrů, s jakými se prováděl naposledy. Prováděný příkaz je indikován v položce St informačního řádku.

Tímto způsobem nejde zadat přechod do Video-režimu (příkaz V).

3.5.4. Specifikace naposledy uvedených parametrů

Editor si pamatuje řetězce, které byly naposledy specifikovány jako hodnoty parametrů A, F a R. Tyto řetězce lze použít při zadávání parametrů dalšího příkazu. Jejich vyvolání se děje následujícími kombinacemi kláves:

CTRL + A	poslední specifikace parametrů A
CTRL + F	poslední specifikace parametrů F
CTRL + R	poslední specifikace parametru R

3.6. Chybové stavy editoru

Protože uživatel stále vidí na obrazovce efekt akcí, které se souborem provádí, nehlásí Editor žádné chyby. Pouze v některých případech ohlašuje pípnutím, požadujeme-li nemožnou akci, např. pohyb kurzorem v příkazovém režimu. Před provedením nevratných akcí editor žádá potvrzení příkazu.

Pokud již není k dispozici dostatek paměti (její velikost lze přečíst na informačním řádku), může se stát, že požadovanou akci již nelze provést. Vždy však půjde provést výmaz znaku, resp. řádku a pohybovat kurzorem.

Je-li k dispozici jen velmi málo paměti, nemusí jít provést ani ukončení práce editoru. Nejde-li provést příkaz E ukončující práci editoru, objeví se po takovém neúspěšném pokusu v místě St informačního řádku inverzní čtvereček. V tomto případě se doporučuje provést výmaz nezbytné části souboru, pak editor uzavřít a opět otevřít. Pokračujeme-li po neúspěšném pokusu o opuštění editoru v práci, může dojít k některým nepravidelnostem v jeho chování.

EDITOR vždy vyžaduje, aby měl k dispozici stále alespoň 4 byty volného prostoru pro svoji potřebu. Velikost paměti potřebná k uzavření souboru je rovna přibližně počtu vět souboru.

4. Vyvolání editoru na obecný soubor

Tvar příkazu je:

TRAnsfer f1, f2

Soubor f1 musí existovat, implicitní hodnota je :BB:. Soubor f2 nesmí existovat, implicitní specifikace není dovolena.

Oba soubory mohou být i vnější. Soubor f1 slouží jako vstupní, nemůže být tedy souborem na zařízení výlučně výstupním (:CO:, :LP:, :PO:). Obdobně soubor f2 slouží jako výstupní, nemůže být tedy souborem na výlučně vstupních zařízeních (:CI:, :RI:).

Tímto příkazem se ze vstupního souboru f1 vytváří jeho editací výstupní soubor f2. Po zavolání příkazu se otevře systémový program editor na prázdný pomocný soubor, který se edituje. Při jeho editaci je kromě obvyklých akcí editoru možno načíst na konec tohoto pomocného souboru zadaný počet vět vstupního souboru f1 a nechat vystoupit zadaný počet vět ze začátku pomocného souboru do výstupního souboru.

Je tedy při zavolání editoru příkazem Transfer repertoár příkazového režimu rozšířen o dva další příkazy:

- **I** načtení zadaného počtu vět ze souboru f1 na konec pomocného souboru (je-li soubor f1 ve stavu EOF - konec souboru -jde o prázdnou akci).
- **O** výstup zadaného počtu vět z počátku pomocného souboru do souboru f2.

Po zadání některého z těchto příkazů se na obrazovce objeví výzva:

"**I.. KOLIK?**" resp. "**O.. KOLIK?**"

a systém očekává reakci operátora. Ten má tři možnosti, jak odpovědět:

- Stiskem klávesy home zrušit tento příkaz a pokračovat v editaci pomocného souboru v nezměněném stavu.
- Specifikovat s kolika větami se má akce provést. Zadat můžeme nejvýše třímístné číslo. Číslo je zadáno až v okamžiku jeho odeslání klávesou CR, do té doby je možné jej editovat v tříznakovém bufferu.
- Stiskem klávesy CR nechat provést požadovanou akci se stejným počtem vět, jaký byl použit při poslední akci (při první akci je použita implicitní hodnota 30).

Po provedení této akce se opět otevře nad požadovaným způsobem změněným pomocným souborem editor (máme stále k dispozici i příkazy I a O).

Zadáme-li příkaz E pro opuštění editoru, na obrazovce se objeví dotaz:

"**E.. ZBYTEK VYPUSTIT (A/N)?**"

a systém očekává reakci operátora. Ten má tři možnosti, jak odpovědět:

- stiskem klávesy home zrušit zadaný příkaz a vrátit se do editace pomocného souboru.
- Odeslat znak A, tím se práce editoru ukončí a do výstupního souboru f2 se dokopíruje obsah pomocného editačního souboru.
- Odeslat znak N, tím se práce editoru ukončí a do výstupního souboru f2 se dokopíruje nejprve obsah pomocného editačního souboru a za něj i dosud "nezpracovaná" část vstupního souboru f1.

Znaky A resp. N musí uživatel odeslat klávesou CR, do té doby je lze editovat v bufferu délky 1.

Poznámka:

Pokud je požadováno provedení příkazu I na tak velký počet vět souboru f1, že by se pak pomocný editační soubor "nevešel" do aktuálního volného prostoru pro soubory, načte se ze souboru f1 jen tolik bytů, kolik je možné. Je to jediný případ, kdy příkaz TRA "přetrhne" věty vstupního souboru. Je dobré se takové situaci vhodnou volbou počtu čtených vět a včasným vyprazdňováním pomocného souboru příkazem O vyhnout. Dojde-li k ní přece, je vhodné si před další prací "přetržené" místo prohlédnout.

Pokud dojde k přeplnění paměti a soubor f2 je paměťový, může opuštění editoru příkazem E trvat velmi dlouho.

Dodatek A - Příkazy OS AMOS

Tvar příkazu	Význam
CAT	Výpis tabulky souborů
COP f1, f2	Zkopírování souboru f1 do souboru f2. f1 musí existovat, implicitně :CI:, f2 nesmí existovat, implicitně :CO:.
REN f1, f2	Přejmenování paměťového nebo diskového souboru f1 na f2. f1 musí existovat a musí být uveden, f2 nesmí existovat, musí být uveden, jeho předpona je násilně přepsána na předponu f1
DEL f1	Zrušení paměťového nebo diskového souboru. f1 musí být paměťový nebo diskový, musí existovat a musí být uveden
APP f1, f2	Připojení obsahu souboru f2 do paměťového nebo diskového souboru f1 za jeho konec. Soubor f2 zůstane zachován. f1 musí byt paměťový nebo diskový, musí existovat a musí být uveden f2 musí existovat, implicitně :CI:
DIR č1	Výpis adresáře souborů na disku č1 je číslo štěrbiny (0 nebo 1)
EDI f1	Otevření editoru na editaci paměťového souboru f1 musí být paměťový textový soubor, musí být uveden. Pokud neexistuje, je nejprve vytvořen jako prázdný
TRA f1, f2	Otevření editoru na editaci obecného souboru. f1 je vstupní soubor, musí být textový a musí existovat. Implicitně :BB: f2 je výstupní soubor, musí být textový nesmí existovat a musí být uveden
ASS f1,f2,f3 ;(params)	Volání překladače assembleru na překlad zdrojového programu. f1 je soubor se zdrojovým programem, musí existovat a musí být uveden f2 je soubor, do kterého bude generován přeložený program (object-kód), není-li paměťový, nesmí existovat, implicitní jméno=jméno f1, přípona vždy přepsána na OBJ f3 je soubor, do kterého bude vytvářen listing, nesmí existovat, implicitně :CO:. Pro paměťový f3 nutno vyhradit v paměti místo (pomocí Z (n)), (params) jsou parametry překladu.

Tvar příkazu	Význam
PAS f1,f2,f3 ;(params)	Volání překladače Pascalu na překlad zdrojového programu f1 je soubor se zdrojovým programem, musí existovat a musí být uveden f2 je soubor, do kterého bude generován přeložený program (object-kód), není-li paměťový, nesmí existovat, implicitní jméno=jméno f1, přípona vždy přepsána na OBJ f3 je soubor, do kterého bude vytvářen listing, nesmí existovat, implicitně :C0:. Pro paměťový f3 nutno vyhradit v paměti místo (pomocí Z (n)), (params) jsou parametry překladu.
DEB (příkaz volání uživa tel. programu)	Volání debuggeru (příkaz volání uživatel. programu) je buď, volací věta programu, nebo: ,(seznam souborů) ,(params) (jako u příkazu RUN)
SAV	Archivace aktuálního stavu systému na magnetofon.
LOA	Obnovení stavu systému uloženého na magnetofon (příkazem SAV).
PRI č1,č2	Nastavení parametrů tiskárny č1 je počet znaků na řádce, č2 je počet řádek na stránce (po startu OS platí č1=132, č2=65).
SPA č1	Vyhrazení oblasti pro explicitně alokované soubory (oblast SPACE). č1 je počet, vyhradí se č1+1 sektorů.
LOC f1 ;(posunutí)	Alokování object-kódu do paměti f1 je soubor, obsahující object-kód, musí existovat a musí být uveden, přípona se vždy chápe jako OBJ
RUN (start. adresa) ,(seznam sou borů) ;(params)	Spuštění již alokovaného programu (start. adresa) je buď explicitní adresa nebo znak $ (implicitní adresa) (seznam souborů) je seznam specifikací souborů (parami) je textový parametr programu
MON	Volání monitoru.
BYE	Návrat do programu, ze kterého byl OS AMOS zavolán.
USE	Příkaz definovaný uživatelem.

Poznámka:
Soubor musí být uveden = jeho implicitní specifikace není přípustná.

POVELY OS AMOS

Klávesy	Význam
CTRL+BR	Výpis tabulky souborů
Shift+CTRL+BR	Ukončení právě probíhajícího programu (break)
Shift+FA+BR	zapnutí hustého řádkování na obrazovce
FA+BR	zapnutí řídkého řádkování na obrazovce
Shift+FB+BR	zapnutí zobrazování obsahu grafické paměti (modulu GRAFIK)
FB+BR	vypnutí zobrazování obsahu grafické paměti (modulu GRAFIK)
CTRL+S	zastavení výpisu na obrazovku
CTRL+Q	Pokračování výpisu na obrazovku

Dodatek B

6. Seznam chyb operačního systému AMOS

Chybová hlášení se vypisují na obrazovku jako inverzní nápis: "CHYBA (číslo)".

1 Chyba v implicitní specifikaci souboru.
Soubor nebyl specifikován explicitně a přitom nebyla ani přípustně definována jeho implicitní specifikace.

2 Nepřípustné jméno zařízení.
Chyba je hlášena i pro jména :D0: a :D1: v případě, že nejsou připojeny disky.

3 Chyba v dodatku specifikace souboru.
V dodatku je očekáváno číslo nebo znak "S" uvnitř kulatých závorek, číslo smí být uvedeno pouze pro magnetofonové soubory, příznak spoolování "S" může být uveden pouze pro soubory na zařízeních :CO:, :LP:, :P0:, :U0: a :U1:.

4 Chyba ve specifikaci souboru.
Např. specifikace zařízení nekončí dvojtečkou, jméno nezačíná písmenem, specifikace souboru není ukončena oddělovačem (čárka, středník, CR) apod.

5 Příkaz byl špatně ukončen.
Za logickým koncem příkazu ještě následují další nemezerové znaky.

6 Chyba ve specifikaci čísla.

7 Specifikace souborů musí být v příkazu odděleny čárkou.

8 V příkazu bylo specifikováno příliš mnoho souborů,
tj. více než 16 nebo více než 3 při volání překladačů.

9 Soubor (výstupní) již existuje.

10 Soubor (vstupní) neexistuje.
Soubor již musí existovat v tabulce nebo být vnější.

12 Soubor na daném zařízení nejde otevřít pro čtení.
Otevřít pro čtení jdou soubory na zařízeních :MM:, :CI:, :MG:, :RI:, :BB:, :U0:, :U1:, :D0: a :D1:.

13 Soubor na daném zařízení nejde otevřít pro zápis.
Otevřít pro zápis jdou soubory na zařízeních :MM:, :CO:, :MG:, :PO:, :BB:, :U0:, U1:, D0: a :D1:.

14 Nepřípustné spoolovací zařízení.
Přípustné jsou pouze paměť a pružné disky.

15 Nepřípustná hodnota implicitní předpony specifikace souboru. Přípustné jsou pouze :MM:, :D0: a :D1:.

16 Textový soubor nelze otevřít pro přímý přístup.

17 Nepřípustný způsob otevření souboru.

18 Požadavek na čtení z fiktivního souboru, GET nebo DELETE na soubor s identifikací 0000H.

19 Pokus o čtení ze souboru, který nebyl otevřen pro čtení.

20 Pokus o čtení ze souboru po jeho konci.

21 Timeout při čtení z :RI:.
Pravděpodobně není snímač děrné pásky připojen.

22 Pokus o zápis do souboru, který nebyl pro zápis otevřen.

23 V HL není systémová identifikace souboru.

24 Přeplnění operační paměti.
Byla vyčerpána veškerá paměť dostupná pro soubory a tabulku souborů.

25 Soubor musí být paměťový nebo diskový.
Hlásí se u systémových příkazů utilit RENAME, DELETE a APPEND.

26 EDITOR nelze otevřít, protože specifikovaný soubor buď není paměťový nebo není textový.

27 Není k dispozici dost paměti pro příkaz TRANSFER.
K zahájení práce stačí jeden volný sektor.

26 Modul AMOS/Pascal není přítomen.

29 Modul AMOS/Assembler není přítomen.

30 Grafický Basic kompatibilní s OS AMOS není přítomen.
Chyba je hlášena zavoláme-li příkaz BASIC a přitom není ani zasunut modul AMOS/Basic ani nebyl tento program nahrán z kazety.

31 Příkaz LOAD je proveden v jiné konfiguraci počítače, než v které byl stav systému uložen příkazem SAVE na magnetofon. I když je připojen modul DISC 2, nejde diskety přesto použít.

32 Není k dispozici požadovaná paměť pro oblast SPACE.

33 Příkazem LOCATE byl program alokován mimo oblast vyhrazenou pro programy příkazem SPACE. Alokace proběhla jinak v pořádku, přesto program nyní nelze spustit příkazem RUN. Program můžeme spustit, pokud zvětšíme vyhrazenou oblast příkazem SPACE, dříve než budeme jakkoli manipulovat se soubory.

34 Pokus o alokaci do místa paměti, které není volné, v tomto případě se alokace neprovedla.

35 Pokus o alokaci souboru, který není object-kódem.
K této chybě může dojít nejen při explicitním volání příkazu LOCATE, ale i při spouštění uživatelských programů.

36 Příkazem RUN nelze spustit programy, jejichž startovací adresa leží mimo oblast, jež byla pro programy vyhrazena příkazem SPACE.
Chyba může nastat i při příkazu DEB.

37 Není definována implicitní startovací adresa pro příkaz RUN.
Chyba může nastat i při příkazu DEB.

38 Byla volána akce SEEK na paměťový soubor s tak velkou hodnotou parametru udávajícího pořadí bytu v souboru že by výsledná pozice ležela mimo prostor vyhrazený pro soubory.

44 Nesouhlasí-li kontrolní součet obsahu paměti EPROM na desce s operačním systémem.

45 Nesouhlasí kontrolní součet obsahu paměti EPROM na desce s překladačem Pascalu.
Pokud je počítač v pořádku, znamenají tyto chyby, že paměti EPROM "zapomněly" svůj obsah. Závadu musí vyřešit servis.

51-69 Tato čísla chyb jsou vyhrazena pro chyby hlášené modulem pro ovládání disků, jejich význam je uveden v příručce k modulu DISC 2.

70 Není definována akce otevření pro čtení pro zařízení :U0:.

73 Není definována akce otevření pro zápis pro zařízení :U0:.

76 Není definována akce GET pro zařízení :U0:.

79 Není definována akce PUT pro zařízení :U0:.

82 Není definována akce CLOSE pro zařízení :U0:.

85 Není definována akce otevření pro čtení pro zařízení :U1:.

88 Není definována akce otevření pro zápis pro zařízení :U1:.

91 Není definována akce GET pro zařízení :U1:.

94 Není definována akce PUT pro zařízení :U1:.

97 Není definována akce CLOSE pro zařízení :U1:.

99 Není dostatek prostoru pro zavolání OS AMOS.
Tato chyba se hlásí není-li pro OS AMOS při jeho startu z Basicu k dispozici alespoň 10 sektorů.

129 Není definována akce příkazu USEr.

132-152 Nejsou definovány akce pro přímý přístup.