

RNDr Pavel Drbal, CSc  
RNDr Jiří Vaníček

ZAVT, koncern - Výzkumný ústav matematických strojů Praha, k.ú.o.

## METODIKA PROGRAMOVÁNÍ V OPERAČNÍM SYSTÉMU DOS-3/JS

Cílem našeho příspěvku je podat základní informaci o operačním systému DOS-3/JS a zmínit se, jakým způsobem se promítnou některá nová řešení užití při vývoji tohoto systému do metodiky programování a programovacích technik.

V současné době jsou v rámci zemí RVHP uváděny do provozu první počítače "Zdokonalené řady JSEP", tak zvané řady JSEP-2. Při přípravě této řady počítačů bylo rozhodnuto, že budou řešeny dva operační systémy, navazující na existující systémy DOS/JS a OS/JS u řady JSEP-2.

Na řešení operačního systému OS soustředil veškeré své kapacity SSSR a NDR. Vývoj tohoto systému probíhal postupným rozšiřováním existujícího systému OS-4 po vzoru firmy IBM tak, že měl být získán systém zhruba srovnatelný se systémem OS-VSl firmy IBM. Tento systém nese označení OS-6/JS. V současné době je tento systém dokončován, jeho rozpracování však není dosud tak daleko, aby se mohl podrobit mezinárodním zkouškám. Původní záměr rozšířit postupně funkce tohoto systému na úroveň srovnatelnou se systémem OS-VS-2 firmy IBM byl opuštěn a další vývoj v rámci JSEP bude sledovat zřejmě

pouze linii zabezpečení virtuálních strojů.

Úmysl sledovat při vývoji operačního systému DOS/JS linii firmy IBM byl záhy změněn. Hlavním řešitelem systému DOS se stalo ČSSR a zde pod vedením Výzkumného ústavu matematických strojů, za účinné pomoci řady dalších organizací, vznikl operační systém DOS-3/JS, který navazuje na předchozí verze DOS/JS pouze tím, že zajišťuje na potřebné úrovni uživatelskou návaznost. Jeho vnitřní skladba je však oproti předchozím verzím DOS/JS zcela odlišná a rozsah funkcí, které nabízí, je s předchozími verzemi DOS/JS i se systémy DOS a DOS-VS firmy IBM nesrovnatelně bohatší. Systém DOS-3/JS má i řadu funkcí operačního systému OS a v některých směrech (například zabezpečení virtuální paměti v rozsahu prakticky 16 M byte pro každého uživatele nezávisle) je dokonce převyšuje. Kromě bohatých možností, které tento systém poskytuje, lze očekávat, že zajistí i velmi efektivní provoz počítače, protože pro jednoduchou a moderní koncepci svého řešení bude potřebovat oproti rozsáhlým systémům s nepřehlednými vnitřními vazbami pouze menší část strojového času pro vlastní práci systému a ponechá jeho větší podíl pro řešení úloh uživatele. Bližší podrobnosti o vlastnostech operačních systémů JSEP-2, spolu se seznamem literatury k této problematice lze nalézt v [3], řadu uživatelsky zajímavých informací o systému DOS-3 pak v [1].

Podle původního záměru měl být DOS-3/JS určen pro nižší a střední členy řady JSEP-2 (do JS 1035 včetně), kdežto na vyšších členech řady měl pracovat operační systém OS-6/JS. V průběhu řešení se však podařilo rozsah funkcí DOS-3/JS zvýšit tak, že tento systém je vhodný i pro vyšší členy řady JSEP-2. K dnešnímu dni byl již odzkoušen na JS 1015, JS 1025,

JS 1035 (SSSR), JS 1035 (BLR), JS 1055. Domníváme se, že i s jeho prací na JS 1045 nebudou žádné technické problémy. Na druhé straně operační systém OS-6/JS dosud neprošel mezinárodními zkouškami, jeho práce na EC 1025 je vzhledem k technickým parametrům počítače vyloučená, na JS 1035 velmi problematická a zdá se, že ve srovnání s DOS-3/EC bude i na vyšších členech řady JSEP-2 pracovat mnohem méně efektivně. Kromě toho je úmyslem dodavatelů počítačů vybavených OS-6/JS snížit na minimum rozsah komponent systému, které budou dodávány s počítačem zdarma, a ostatní komponenty dodávat samostatně, za zvláštní úhradu.

Autoři tohoto příspěvku se proto domnívají, že pro využívání počítačů JSEP-2 v ČSSR by bylo vhodné vybavit všechny počítače této řady jediným operačním systémem DOS-3/JS. Bude-li tato koncepce zvolena, je ovšem třeba ji promítnout i do obchodních jednání o dovozu počítačů JSEP-2 z ostatních zemí RVHP.

V dalším textu svého příspěvku si všimneme některých specifických vlastností systému DOS-3/JS, které mají bezprostřední výrazný dopad na programovací techniky.

Jednou z důležitých vlastností DOS-3/JS je významné obohacení jazyka řízení úloh (monitoru). Tento jazyk obsahuje v DOS-3 nepodmíněné i podmíněné skoky a jeho struktura se tak blíží struktuře programovacího jazyka. V jeho rámci lze vytvářet procedury příkazů jazyka řízení úloh, které mají charakter podprogramů a jejichž volání lze považovat za nový příkaz jazyka řízení úloh. Takovéto vytváření monitorových procedur může být užito ve více úrovních.

Při volání procedury je možné kromě toho zadat požadavek záměny znakových řetězců, který má za následek, že při

provádění procedury je nahrazován jeden z uvedených znakových řetězců existující v příkazech procedury druhým zadaným řetězcem. Tak například je možné v zadané proceduře měnit jména programů, které se mají zpracovávat nebo hodnoty parametrů příkazu OPTION. Tato vlastnost umožňuje širokou flexibilitu procedur příkazů pro řízení úloh a je bohatší než například volba parametrů podprogramu, protože neomezuje možnost modifikace procedury pouze na předem pevně zadané parametry. Uvedené modifikace lze opět provádět na více úrovních.

Jazyk řízení úloh u DOS-3 tak dává vhodný aparát pro spojování více programů do flexibilních parametrických pod-systémů a systémů a usnadňuje tak významně práci při jejich tvorbě.

Operační systém DOS-3/EC je koncipován tak, že umožňuje současně zpracovávat úlohy v dávkovém režimu a vedle toho u úloh, které to vyžadují, provádět zpracování v reálném čase. Toto řešení spojuje zajištění požadavků uživatele na rychlé poskytnutí služeb u úloh, kde je to zapotřebí, a výhodami dávkového zpracování, pokud jde o optimální využití systémových zdrojů počítače.

Základním režimem dávkového zpracování úloh je nepřímý vstup a výstup úloh. Program zpracováváný v tomto režimu nepoužívá přímo pomalých vnějších zařízení, ale data i monitor-ské příkazy odebírá ze vstupních front na magnetické diskové paměti a výstupní data opět ukládá do výstupních front na disku. Z reálných vstupních zařízení jsou data načítána do vstupních front nezávislými systémovými procesy, které probíhají paralelně s ostatními procesy zpracováványými systémem a obdobné systémové procesy realizují výstup z výstupních front na reálná výstupní zařízení.

Toto řešení ovšem nebrání tomu, aby úloha, která má být zpracovávána v reálném čase, získávala svá data přímo mimo tyto fronty, např. z terminální stanice nebo zasílala na ni přímo své výsledky.

Při nepřímém vstupu a výstupu úloh jsou na vstupu vytvářeny dvě fronty: pro příkazy řízení úloh (BCH) a pro vstupní data úloh (RDR) a na výstupu opět dvě fronty: dat ve formátu tiskárny (LST) a děrných štítků (PCH). Jednotlivé položky v těchto frontách mají pro program povahu vnějších zařízení, takže nepřímý vstup a výstup úloh nevyžaduje žádné změny při přípravě programu. Potřebné přiřazení lze provést až příkazy jazyka pro řízení úloh. Příkazy tohoto jazyka je možné (pokud to připouštějí jejich charakteristiky) vzájemně přiřazovat jednotlivé položky ve zmíněných frontách. Tato skutečnost umožňuje například vytvářet programem posloupnosti monitorovských příkazů a automaticky je provádět nebo jednoduchým způsobem měnit vstupní data programů, uložená na systémových knihovnách.

Užíváme-li například připravenou soustavu programů v různých kombinacích a s různými parametry, lze vhodně vytvořeným konverzačním programem zjistit konkrétní požadavky uživatele a podle nich může již program vytvořit potřebnou posloupnost příkazů řízení úloh s požadovanými parametry a provést ji.

Při provozu od terminální stanice existuje kromě možnosti zadat úlohu od terminálu do vstupní fronty (RJE) i možnost pracovat přímo s již připravenými položkami ve frontách úloh a dat a provádět v nich i změny. Tím je umožněno například od terminálu převést program ze zdrojové knihovny do vstupní fronty, zde provést potřebné opravy, spustit překlad,

vyžádat jeho výsledky na obrazovku stanice a podle nich rozhodnout zda opravený program bude katalogizován, případně přeložený program spustit a v dalším postupu rozhodnout na základě získaných výsledků. Při tomto postupu všechny operace probíhají pouze prostřednictvím obrazovky terminálu, bez užití výstupu na tiskárnu. Uvedená možnost ovšem nevylučuje běžné užití terminálu jako vstupního a výstupního zařízení některým zpracovávaným programem.

Významným novým rysem počítačů řady JSEP-2 je zavedení virtuální adresace paměti. Toto technické řešení dává možnost zbavit programátora do určité míry řady starostí spojených<sup>6</sup> obsluhou paměťových hierarchií. Před tvůrci operačních systémů JSEP-2 vyvstal úkol začlenit obsluhu virtuální paměti do operačních systémů. U operačního systému OS-6 byla tato obsluha doplněna do existujícího systému, u DOS-3 byl zvolen jiný postup. Virtuálního přístupu k paměti bylo užito již pro sám operační systém a díky tomu, že jádro DOS-3 bylo řešeno nově, podařilo se obsluhu virtuální paměti využít jako integrální část celého systému. Daň, kterou je ovšem třeba zaplatit za toto moderní řešení spočívá v tom, že systém DOS-3 nemůže na rozdíl od OS-6 ani s omezenými funkcemi pracovat na počítačích, které nemají virtuální adresaci paměti, tj. na počítačích JSEP-1. Podrobnější informace o využití virtuální paměti v operačním systému DOS-3 lze nalézt v [2].

Pro přípravu programů, které mají pracovat ve virtuálním prostředí, platí některé obecné zásady vyplývající z toho, že je vhodné minimalizovat přesuny mezi reálnou pamětí a odkládacím prostorem pro paměťové stránky na magnetickém disku. Při úvahách o práci programu ve virtuální paměti je třeba si uvědomit, že virtuální adresový prostor je pouze teoretickou

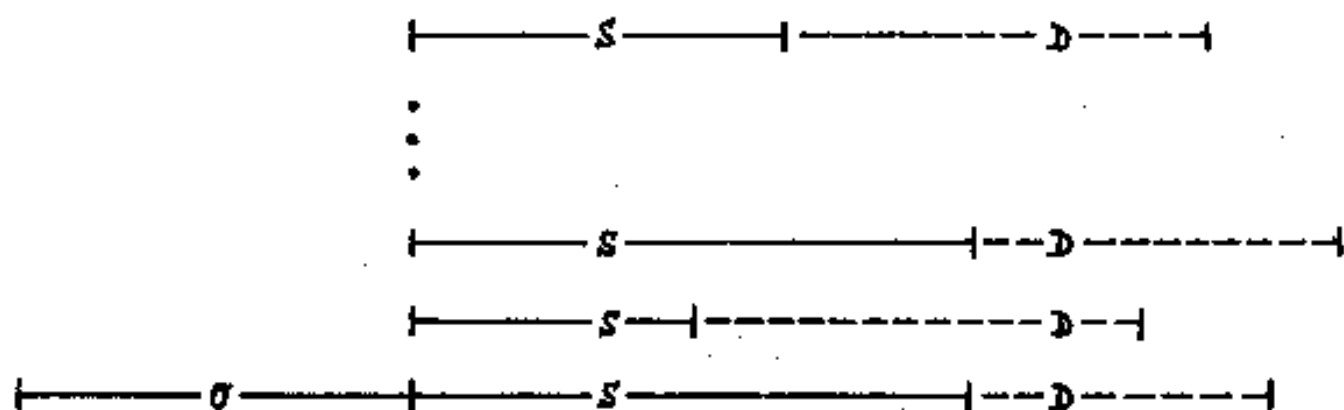
adresovou množinou, která, pokud jednotlivé stránky nejsou používány, nemá ve výpočetním systému nikde žádný obraz. Při prvním užití adres se vytvoří pouhým vzájemným přiřazením položek v tabulkách příslušná stránka v reálné paměti a její obraz v odkládacím prostoru na disku se vytváří až tehdy, je-li třeba potřebný prostor v reálné paměti vyklidit pro jinou stránku a momentální stav vyklizované paměti uchovat.

Z uvedeného vyplývá, že ve virtuálním prostředí není třeba s virtuálními adresami nikterak šetřit, že však je vhodné minimalizovat počet těch stránek, které bude třeba na disk ukládat. To jsou ovšem ty stránky, které se v průběhu programu mění, byť v jediném byte. Je proto účelné soustředit veškeré proměnné do jedné části programu a vyhnout se dynamickým změnám instrukcí v paměti.

Operační systém DOS-3 respektuje požadavek soustředění dat do jedné oblasti a při programování vstupů a výstupů dává možnost uživateli nezadávat vyrovnávací paměť pro blok fyzického záznamu, ale pracovat pomocí pracovní oblasti (work area) pro logický blok. Operační systém vyhrazuje místa pro fyzický záznam tak, aby přesouvaná data byla soustředěna na jediné místo. Ve virtuálním prostředí nemá obvykle smysl provádět vlastní segmentaci programu, protože mechanismus záměny stránek zpravidla pracuje efektivněji než programátorem plánovaná výměna informací mezi reálnou pamětí a diskem. Rovněž snaha, aby nejčastěji prováděné cykly programu byly prováděny v rámci jediné stránky, nevede obvykle k zlepšení práce programu, protože stránkovací mechanismus pracuje tak, že řadu frekventovaných stránek ponechává v paměti trvale.

K tomu, abychom uvedené obecné zkušenosti o práci ve virtuálním prostředí doplnili o doporučení, která vyplývají

ze specifiky systému DOS-3, je třeba se zmínit poněkud podrobněji o tom, jak se virtuální paměť jeví v tomto systému uživateli. V systému DOS-3 má každý paralelní proces (uživatel) k dispozici vlastní virtuální adresový prostor velikosti 16 M byte, dělený na segmenty délky 64 K byte a stránky délky 2 K byte. Schema na obr.



obr.

Tento prostor je dělen z hlediska každého uživatele na tři části:

- O - část paměti pro operační systém,
- S - statická část paměti uživatele,
- D - dynamická část paměti uživatele.

Část operačního systému je všemi procesy sdílěna, uživatelských částí je tolik, kolik je spuštěných procesů a jsou zřizovány a rušeny v průběhu práce systému dynamicky. Statická část je zřizována po segmentech, pro každý krok úlohy (job step) při jeho spuštění, dynamická část je zřizována též po segmentech až na požadavek programu (makro "get virtual storage").

Užitím dvouúrovňového (segmenty, stránky) řízení virtuální paměti je v DOS-3 zajištěna současně vzájemná ochrana



jednotlivých uživatelských programů před nežádoucím vlivem i možnost, aby různé procesy v případě potřeby pracovaly (pod různými virtuálními adresami) s touž sdílenou stránkou virtuální paměti.

Skutečnost, že v operačním systému DOS-3 jsou programy operačního systému sdíleny paralelně probíhajícími procesy si vynucuje, aby tyto programy byly reentrantní, tj., aby je mohly různé procesy užívat současně. Takovéto programy ovšem nemohou obsahovat proměnné, ty musí mít každý proces ve své vlastní dynamické paměti (D) a instrukce těchto programů lze modifikovat pouze beze změny obsahu paměti (pomocí instrukce EX). Požadavek reentrantnosti je účelné uplatnit i na některé programy uživatele, které pak mohou paralelní procesy sdílet tak, že příslušné stránky v částech (S) a (D) u různých uživatelů (pod různými virtuálními adresami) budou reprezentovány týmiž stránkami v reálné paměti. Takovýto program musí opět veškerá měněná data obsahovat pouze ve své vlastní dynamicky zřízené virtuální paměti.

Na rozdíl od jiných operačních systémů není zavedení programu z fázové knihovny do paměti realizováno přesunem textu, ale tak, že fáze je prohlášena za část virtuální paměti a do reálné paměti je přesouvána až v případě potřeby. Je ovšem zajištěno, aby změny textu programu byly platné pouze v rámci jednoho spuštění programu.

Prvkem, který v DOS-3 eliminuje složité realizace přístupových metod pro diskové soubory a umožňuje využít pro tento účel aparátu virtuálních adres, je koncepce tak zvané soukromé virtuální paměti. Při ní lze diskový soubor o délce bloku 2 K byte prohlásit za vzory stránek virtuální paměti a dát k dispozici jejich virtuální adresy. Nadále pak jsou data

tohoto souboru přístupná jako operační paměť a po ukončení zpracování budou provedené změny promítnuty do diskového souboru. Při tomto postupu probíhá zpracování analogicky k přesunům dat mezi operační pamětí a diskem, při kterém roli vyrovnávací paměti hrají ty stránky, které jsou momentálně v reálné paměti počítače. Rozsah této vyrovnávací paměti je tak určován automaticky provozními podmínkami v době provádění výpočtu. Virtuální paměť tak umožňuje velmi pružný a efektivní styk s diskovými soubory.

Souhrnně lze říci, že nové řešení virtuální paměti v DOS-3 poskytuje řadu možností k usnadnění práce programátora a k zvýšení efektivity získaných programů. Z důvodu kompatibility s předchozími verzemi DOS-3 není využívání těchto možností povinné a lze i nadále pracovat "klasickými" metodami. Vzhledem k tomu, že celý systém je řešen tak, aby jeho práce, jmenovitě obsluha stránkového mechanismu, byla maximálně efektivní, doporučujeme jeho možnosti plně využít, postupy obvyklé u minulých verzí užívat jen tam, kde je to bezpodmínečně nutné a pokud systém nabízí možnosti automatizace některých prací, tyto možnosti využívat.

#### Literatura:

- [1] Pilinger, Č.: Diskový operační systém DOS-3/JS. Soubor článků ve zpravodaji Informační služba KP NOTO 1 (1978)-10 (1980).
- [2] Havrátil, V., Sokol, J.: Virtuální operační systém pro malé a střední počítače. Sborník SOPSEM '77 (VVS-Bratislava), str. 179-197.
- [3] Vaníček, J.: Operační systémy řady JSEP-2, MAA, příloha čísel 1 a 2 (1980).