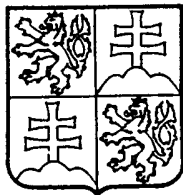


ČESKÁ A SLOVENSKÁ
FEDERATIVNÍ
REPUBLIKA
(19)



FEDERÁLNÍ ÚŘAD
PRO VYNÁLEZY

ZVEŘEJNĚNÁ PŘIHLÁŠKA VYNÁLEZU

(12)

(21) 02702-91.R

(13) A3

5(51) G 06 F 12/00.
12/02

(22) 02.09.91

(32) 03.09.90

(31) 90/116847

(33) DE

(40) 15.04.92

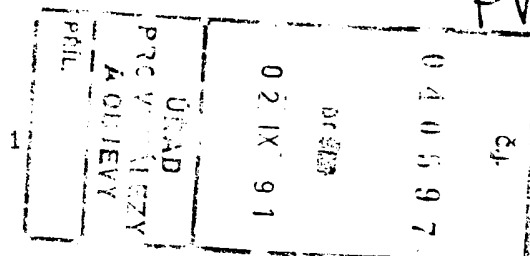
(71) International Business Machines Corporation, New York, New York, US

(72) Salm Ingolf, Gäufelden, DE

(54) Počítač s koncepcí rozšířené virtuální paměti

(57)

V počítači obsahujícím operační systém, který zahrnuje organizační koncepci virtuální paměti, jež poskytuje paměťový prostor pro paralelní provádění programu v předběžně přidělených segmentech paměti virtuálních adres, obsahující sdílené oblasti a vyhrazené oblasti v několika adresních prostorech, jsou poskytnuty dynamické segmenty (P1, P2, ..., N1, N2, ..., O1, O2, ...) které jsou přidělitelné navíc k předběžně přiděleným segmentům (F1-F8, FA, FB, BG) v závislosti na zakázce, která je prováděna, a které jsou uvolňovány po ukončení zakázky, aby oblast paměti byla neobsazená pro následující použití. Tato koncepce rozšiřuje oblast virtuální paměti, efektivněji využívá poskytnutý prostor a dovoluje paralelní zpracování více programu.



Počítač s koncepcí rozšířené virtuální paměti

Oblast techniky

Vynález se týká počítače včetně operačního systému, který obsahuje organizační koncepci virtuální paměti poskytující paměťový prostor pro paralelní provádění programů v předběžně přidělených segmentech paměti virtuálních adres obsahujících sdílené oblasti a vyhrazené oblasti v několika adresních prostorech.

Stav techniky

Takový počítač a speciálně takový operační systém je na trhu například pod názvem Virtual Storage Extended/System Package Version 4 Release 1, v následujícím krátce VSE/SP 4.1, podrobně popsany například v příslušných manuálech, z nichž je zde pro účely odkazu citován Administration, Form Number SC33-6405-01. Operační systém VSE/SP 4.1 obsahuje program VSE/Advanced Functions (v následujícím krátce VSE/AF), stejně jako další, z nichž by měl být také zmíněn program VSE/POWER. Odkazuje se na manuál VSE/Advanced Functions System Management Guide Version 4 Release 1, Form Number SC33-6352-00. V tomto manuálu, zvláště na stranách 1 až 20, je podán podrobný popis týkající se např. multiprogramování, virtuální paměti a organizace reálné paměti, segmentů, stejně jako přidělování paměti.

V běžném systému je virtuální paměť omezena na 12 segmentů v maximálně devíti adresních prostorech. To vyžaduje velmi pečlivé a obtížné přidělování paměťového prostoru v daných adresních prostorech programům, aby se tyto mohly zpracovávat paralelně. Toto podstatně omezuje možnost para-

lelního provádění programů, a tak činí systém poněkud nepružným.

Podstata vynálezu

Základním předmětem předkládaného vynálezu je učinit systém flexibilnějším, zvětšit zdroje virtuální paměti tak, že lehce může být prováděno paralelně více programů, a přijatelným pro existující systém.

Tyto a jiné předměty jsou řešeny zásadně aplikací vlastností ukázaných níže v hlavním patentovém nároku 1.

Takto nově zavedenými dynamickými segmenty, které jsou přidělovány společně se zakázkou a uvolňovány s ukončením zakázky, je poskytnuto ohromné rozšíření oblasti virtuální paměti až o (nebo více) 200 nových segmentů. To znamená, že mnohem více programů může být prováděno paralelně a že dnes existující systémová omezení jsou překonána.

Další příznivé vývody jsou ukázány dole v závislých patentových podnárocích. Jejich specifické rysy a výhody se buď ozřejmí samy nebo budou vysvětleny ve spojitosti s následujícím podrobnějším popisem.

Přehled obrázků na výkresech

V dalším bude vynález podrobněji vysvětlen ve spojitosti s ukázaným a popsáním ztělesněním a obrázky, z nichž:

obr. 1 ukazuje rozvržení paměti známé z dřívějšího stavu znalostí jako předběžně přidělené nebo statické segmenty,

obr. 2 ukazuje vedle sebe rozvržení adresního prostoru pro statické segmenty a pro dynamické segmenty ve shodě s vynálezem,

obr. 3 ukazuje příklad rozvržení paměti s dynamickými segmenty ve shodě s vynálezem,

obr. 4 ukazuje obrazovkové menu se scénářem pro výběr úlohy: udržované dynamické segmenty,

obr. 5 ukazuje obrazovkové menu se scénářem pro přidání dynamické třídy,

obr. 6 ukazuje obrazovkové menu se scénářem definování dynamické třídy X,

obr. 7 ukazuje obrazovkové menu se scénářem pro zapsání změněné tabulky dynamických tříd do knihovny,

obr. 8 ukazuje obraz systémové konzole se scénářem zavedení tabulky dynamických tříd,

obr. 9 ukazuje obrazovkové menu se scénářem vytvoření VSE/POWER zakázky (třída X),

obr. 10 ukazuje obraz systémové konzole se scénářem VSE/POWER zakázky, která je právě prováděna.

Provedení vynálezu

Pro lepší porozumění pojmu dynamického segmentu a tomu, jak je tento pojem zaveden do operačního systému VSE, je podán krátký popis návrhu běžného segmentu. Ve spojitosti s popisem vynálezu jsou běžné segmenty nazývané statické nebo předběžně přidělené segmenty.

Předběžně přidělené, statické segmenty

Běžná verze operačního systému VSE (VSE/SP 4.1) dovoluje přidělit 12 segmentů až v devíti adresních prostorech. Adresní prostor obsahuje sdílenou oblast a vyhrazenou oblast. Sdílená oblast je pro všechny adresní prostory stejná. Vyhrazená oblast se liší v závislosti na přidělení. Je možné přidělit více než jeden segment v jedné vyhrazené oblasti. Statické segmenty mají následující předem definované identifikátory identifikačních statí F1, F2, F3, F4, F5, F6, F7, F8, F9, FA, FB, BG.

Rozvržení paměti na obr. 1 zobrazuje jako příklad známého dřívějšího stavu znalostí systém s pěti sektory BG, F1, F2, F3 a F4 přidělenými v adresních prostorech 1 až 4. Jeden adresní prostor by mohl mít maximální velikost např. 16 megabytů (MB). Adresní prostory jsou rozděleny na sdílené oblasti a vyhrazenou oblast. Vyhrazená oblast je oblast programu pro provádění programu. Operátor nebo ASI (automatizovaná systémová inicializace) procedura může přidělovat a spouštět segmenty a přidělovat prováděcí prioritu každému segmentu. Toto předběžné přidělování statických segmentů je provedeno ve fázi nastavení systému.

Statické segmenty mohou dostat JCL (jazyk pro řízení zakázek) příkazy z fyzického snímače nebo z rozvrhujícího programu prací (job scheduler). V operačním systému VSE je rozvrhování prací, řízení zpracování a asynchronní zpracování výstupu zakázky provedeno programem VSE/POWER, to jest, zakázka je programem VSE/POWER řízena. Program VSE/POWER (POWER znamená Priority Output Writer Execution Reader a je to licenční program firmy IBM prvotně použitý pro asynchronní vstup a výstup) může také spouštět statické segmenty a zpracovávat v nich zakázky, to jest, program VSE/POWER dodává příkazy JCL a data do segmentu. Zakázky jsou umístěny ve čtecí frontě. Výpis nebo výstup na děrovač může být směrován do odpovídajících front.

Statický segment je použitelný pro více než jednu zakázku. Program VSE/POWER dovoluje provádět zakázky ve třídách, v tzv. VSE/POWER třídách, kterýmžto je přiřazen jeden nebo více statických segmentů. Třída znamená skupinu zakázek, které buď přicházejí ze stejného vstupního zařízení nebo jsou nasměrovány na stejné výstupní zařízení.

Dynamické segmenty

Manipulační program pro zprostředkované soubory jako program VSE/POWER přiděluje a spouští dynamický segment pro

jednu zakázku. Program VSE/POWER řídí provádění zakázky a uvolňuje dynamický segment po ukončení zakázky. Uvolněná paměť je použitelná pro jinou zakázku.

Každý dynamický segment má svůj vlastní prostor, to jest, pouze jeden dynamický segment je přidělen ve vyhrazené oblasti takového adresního prostoru. Více dynamických segmentů může být přidělováno najednou.

Aby byla redukována velikost sdílených oblastí, je pro adresní prostory dynamických segmentů zavedena nová (vyhrazená) systémová oblast nazývaná dynamický prostor GETVIS. Oblast dynamického prostoru GETVIS patří do vyhrazené oblasti. Obr. 2 ukazuje srovnání rozvržení adresního prostoru statického a dynamického segmentu.

V levé části obr. 2 je ukázáno rozvržení statického segmentu obsahující supervizor a sdílenou oblast ve sdílených oblastech a oblast programu ve vyhrazené oblasti. Také by tam mohla být obsažena oblast segmentu GETVIS. To je paměťový prostor uvnitř segmentu, který je použitelný pro dočasné přidělení k programu sdruženému s tímto segmentem.

V pravé části obr. 2 je ukázáno rozvržení dynamického segmentu ve shodě s provedením vynálezu. Jak může být vidět v porovnání s rozvržením statického segmentu je ve vyhrazené oblasti poskytnuta oblast nazvaná oblast dynamického prostoru GETVIS.

Oblast dynamického prostoru GETVIS je přidělená dočasně spolu s dynamickým segmentem a obsahuje systémová data pro adresní prostor. Tato systémová data jsou pro zakázku právě jedna, a tím také pro přidělený dynamický segment. Proto tato data nejsou nutná pro jiné segmenty a nemusí být nutně obsažena ve sdílené oblasti. Tedy sdílená oblast je prostá těchto dat, a tak může být udržována menší.

Dynamické segmenty mohou být seskupeny do tříd odpovídajících VSE/POWER třídám, ve kterých mohou být zakázky prováděny. Tyto třídy se nazývají dynamické třídy.

Může být specifikována pouze priorita dynamické třídy.

Dynamické segmenty uvnitř jedné a téže třídy jsou časově rozděleny.

Statické a dynamické segmenty mohou být seskupeny v jedné VSE/POWER třídě.

Tabulka, tzv. tabulka dynamických tříd, obsahuje atributy dynamických tříd. Tabulka dynamických tříd může být katalogizovaná v knihovně, odkud může být zaváděna.

Některé příklady atributů tabulky dynamických tříd jsou:

- přidělování paměti:

specifikuje požadavky na kapacitu paměti dynamického segmentu (vyhrazení dynamického prostoru GETVIS a segmentu) dané třídy,

- velikost segmentu:

definuje velikost souvislé virtuální paměti v segmentu, která je rezervovaná pro provádění programu, zbytek segmentu je použitelný jako oblast segmentu GETVIS,

- velikost dynamického prostoru GETVIS:

definuje množství souvislé virtuální paměti pro oblast dynamického prostoru GETVIS (viz obr. 2),

- profil (procedura):

bude proveden v segmentu před zakázkou (odpovídá proceduře ASI statických segmentů),

- maximální počet dynamických segmentů ve třídě:

specifikuje maximální počet segmentů, které mohou být vyhrazené paralelně uvnitř dané třídy, je-li k dispozici dostatek virtuální paměti a dynamických segmentů,

- indikace nepřístupnosti:

dovoluje znepřístupnit dynamickou třídu, takže žádná

zakázka nemůže být prováděna v této třídě,

- manipulovaná vstupní/výstupní zařízení:

specifikuje manipulovaná zařízení, např. snímač, tiskárny a děrovače, které se propojují na program VSE/POWER.

Identifikace dynamického segmentu jsou sestaveny následovně:

`<class> <pno> ,`

kde `<class>` = dynamická třída (jeden znak definovaný uživatelem),

`<pno>` = číslo segmentu uvnitř dynamické třídy.

Například první dynamický segment vyhrazený v dynamické třídě P přijímá identifikaci P1.

Následující příklad, jak je ukázáno na obr. 3, zobrazuje rozvržení paměti se segmenty BG, F1-F8 v adresních prostorech 1 až 4, dynamické segmenty dynamické třídy N v adresních prostorech N1 a N2, jeden dynamický segment v adresním prostoru P2 a jeden dynamický segment v adresním prostoru O4.

V příkladu rozvržení paměti ukázaném na obr. 3 jsou v dynamické třídě N v dané chvíli přidělené pouze adresní prostory N1 a N2, kdežto ostatní adresní prostory v této třídě nejsou aktivní. To samé platí pro adresní prostor P2 třídy P, kde P2 je jediný aktivní dynamický segment třídy P. Na ukázaném příkladu je ze třídy O aktivní pouze adresní prostor O4. Sestava tohoto příkladu ukazuje, že pouze tyto zmíněné dynamické segmenty jsou aktivní v oblasti virtuální paměti, že skutečně pouze tyto mají prováděné zakázky. Jestliže zakázky přidělené dynamickým segmentům nejsou aktivní, jejich přidělená virtuální paměť může být použita k jiným účelům. Toto ukazuje, že pojem dynamického segmentu poskytuje optimální využití virtuální paměti, které může být použito, poněvadž vždy pouze tato virtuální paměť ze společné oblasti volné paměti je použita, což je skutečně potřebné.

Dynamické segmenty a lehkost použití

Následující položky ukazují, jak je lehké zpracovávat zakázky v dynamických segmentech:

1. Definice:

Tabulka dynamických tříd dovoluje definovat atributy dynamických segmentů pouze v několika příkazech (viz též obr. 5, 6 a 7).

2. Dynamické segmenty vyhrazené v případě potřeby:

Dynamické segmenty jsou přidělovány pouze tehdy, když má být zakázka provedena v dynamické třídě. Toto šetří systémové provozní prostředky. Jestliže VSE/POWER třída obsahuje statické a dynamické segmenty, budou pro provádění zakázky obvykle nejdříve použity statické segmenty.

3. Automatické přidělování, inicializace a uvolňování

Prostředí zakázky a provozní prostředky jsou přidělovány a uvolňovány automaticky bez uživatelského zásahu.

4. Identické přidělování prostředků uvnitř jedné dynamické třídy:

Provozní prostředky potřebné pro zakázku jsou přidělovány před prováděním zakázky.

5. Dávkování času:

Neexistuje žádná potřeba řídit zakázky intenzivně využívající CPU, protože poté, co časové kvantum zakázky je vyčerpáno, je zakázka přesunuta do pozice s nejnižší prioritou uvnitř dynamické třídy.

6. Přesun ze statických do dynamických segmentů:

Je snadné přesunovat zakázky do dynamických segmentů,

protože konstrukce se vyhýbá neslučitelnostem.

Koncepce

Koncepce ukazuje, jak mohou být dynamické třídy definovány a jak jsou dynamické segmenty vytvářeny a uvolňovány, když jsou použitelné systémové provozní prostředky (např. virtuální paměť).

Následující odstavce popisují:

- a/ technickou údržbu tabulky dynamických tříd,
 - b/ přidělování dynamického segmentu,
 - c/ provádění zakázky.
-
- 1/ zadejte zakázku ke zpracování v dynamické třídě
 - 2/ program VSE/POWER požaduje vyhrazení dynamického segmentu
 - vytvořte dynamický segment
 - 3/ program VSE/POWER spouští dynamický segment - učiňte dynamický segment schopným odbavování
 - 4/ inicializace dynamického segmentu
 - inicializace adresního prostoru
 - běh profilu dynamické třídy
 - 5/ provádění zakázky
 - 6/ konec zpracování zakázky
 - 7/ deaktivace dynamického segmentu
 - uvolněte systémové provozní prostředky
 - zastavte segment
 - učiňte dynamický segment neschopným odbavování
 - 8/ uvolněte dynamický segment
 - odstraňte segment z přidělovací fronty

Údržba tabulky dynamických tříd (vytváření, ukládání a zavádění)

Libovolný editor nebo VSE dialog může vytvářet a měnit tabulku dynamických tříd. Jsou použitelné služby pro uložení tabulky do knihovny, odkud operátor nebo VSE/POWER ASI procedura může tabulku zavést.

Zaváděcí proces zkontroluje platnost hodnot tabulky dynamických tříd a přeloží informace do vnitřní reprezentace. Chybné vstupy jsou zobrazeny na konzoli. Pouze bezchybné vstupy mohou být aktivovány, to jest, tabulka dynamických tříd je uložena do systémové paměti, tj. je adresovatelná systémovými standardními programy. Dynamické třídy získávají předem definovanou prioritu výběru ke zpracování. Operátor může změnit prioritu příkazem priority.

Následující žádost o zavedení může změnit nebo odstranit vstupy tabulky dynamických tříd. Žádost o zavedení je odmítnuta, když dynamická třída, která se má odstranit, má přidělené dynamické segmenty.

Přidělování

Jeden segment operačního systému VSE se skládá z hlavní zakázky a připojených podzakázek, přitom hlavní zakázka má vždy nejnižší priorotu uvnitř segmentu.

Běžný operační systém VSE (VSE/SP 4.1) má dvouúrovňové schéma přidělování:

1. Segment s nejvyšší prioritou připravený ke zpracování je vybrán z dispečerské fronty.
2. (Pod)zakázka s odpovídající nejvyšší prioritou připravená ke zpracování je vybrána a odbavena.

Podporou dynamických segmentů je pro dynamické segmenty zavedena nová úroveň přidělování. Každá dynamická třída má svou vlastní dispečerskou frontu, která má vstupy pro všechny vyhrazené dynamické segmenty dané třídy. Jak bylo dříve zmíněno, dynamické třídy jsou zahrnuté do dispečerské fronty, což vede k tříúrovňovému schématu přidělování:

1. Statický segment s nejvyšší prioritou nebo dynamická třída připravená ke zpracování se vybere z dispečerské fronty.
2. Je-li vybrána dynamická třída, je určen dynamický segment s nejvyšší prioritou z této ke zpracování připravené třídy.
3. Odpovídající (pod)zakázka s nejvyšší prioritou připravená ke zpracování je vybrána a přidělena.

Operátor nebo inicializační procedura systému může změnit prioritu statických segmentů a dynamických tříd. Aby bylo odstraněno to, že segment nebo třída intenzívně využívající CPU může zastavit přidělování segmentů nebo tříd s nižší prioritou, operační systém VSE dovoluje definovat skupinu vyvážených časových intervalů. Člen této skupiny (statický segment nebo dynamická třída), který spotřeboval svůj časový interval, získává uvnitř této skupiny pozici s nejnižší prioritou. Jestliže více než jeden dynamický segment je přidělen uvnitř dané dynamické třídy, je použita stejná metoda.

Provádění zakázky

Výběr zakázky pro zpracování v dynamické třídě je možný, jestliže je dynamická třída zpřístupněna. Toto je oznámeno v zavedené tabulce dynamických tříd a může být změněno jiným požadavkem na zavedení nebo příkazem.

Následující odstavce popisují zpracování poté, co byla nalezena zakázka pro zpřístupněnou tabulku dynamických tříd. Pro přidělování a spouštění dynamického segmentu požaduje program VSE/POWER vyhrazení dynamického segmentu. Přidělovací služba vytvoří segmentový identifikátor, přidělí a inicializuje řídicí bloky a vytvoří adresní prostor. Dynamický segment je zahrnut do dispečerské fronty dynamických tříd. Služba spouštění segmentu činí dynamický segment schopným

přidělování, tj. připraveným ke spouštění, a spouští inicializaci. Je-li druhý dynamický segment vyhrazen uvnitř stejné dynamické třídy, je pro tuto dynamickou třídu spuštěno dávkování času.

Během inicializace dynamického segmentu je adresní prostor zkontrolován, je zaveden program pro řízení prací a dynamický segment je připraven. Několik systémových standardních programů může být provedeno během přípravných prací. Program pro řízení prací provádí profil. Jestliže se inicializace nemůže dokončit, segment bude zrušen, deaktivován, uvolněn a dynamická třída bude zneprístupněna, aby se znemožnilo provádění jiných zakázek. Po úspěšné inicializaci je zakázka zpracovávána, to jest, program VSE/POWER předává JCL a data do dynamického segmentu. Program VSE/POWER vyžaduje deaktivaci, když je dosaženo konce zakázky.

Pro deaktivaci dynamického segmentu se oznamuje systémovým standardním programům vyvolávaným během přípravných prací, že je požadována deaktivace. Systém uvolňuje provozní prostředky, zastavuje dynamický segment a činí segment neschopným přidělování. Program VSE/POWER segment uvolňuje.

Pro uvolnění dynamického segmentu uvolňovací služba uvolňuje segmentový identifikátor a systémový prostor přidělený řídícím informacím (řídící bloky). Odstraňuje dynamický segment z přidělovací fronty dynamických tříd. Jestliže pouze jeden dynamický segment zůstává v dynamické třídě, bude pro tuto dynamickou třídu vynulováno dávkování času.

Ve spojitosti s obr. 4 až 10 bude popsána definice a užití dynamické třídy, tedy dynamického segmentu. Obrázky ukazují obrazovková menu nebo panely. Interaktivní dialog mezi systémem a uživatelem se provádí pomocí těchto panelů.

Ukázaný scénář je použit k definici dynamické třídy, k zaktualizování tabulky dynamických tříd v systémové knihovně, k aktivování tabulky dynamických tříd užitím příkazu PLOAD programu VSE/POWER a provedení zakázky programu VSE/POWER v definované dynamické třídě.

Na obr. 4 je ukázán panel "resource definition" (definice prostředků). V tomto panelu pod číslem 7 může být proveden výběr "maintain dynamic partitions" (udržované dynamické segmenty) vstupem čísla 7.

Výběrem provedeným ve shodě s obr. 4 se objeví panel zobrazený na obr. 5 "maintain dynamic partitions" (udržované dynamické segmenty). Pro definici a připojení nové dynamické třídy v panelu zobrazeném na obr. 5 je vybrána možnost 1=ADD. Toto vede k panelu zobrazeném na obr. 6. Tento obrázek ukazuje v levém sloupci v podstatě atributy dynamické třídy a v pravém sloupci vysvětlení těchto atributů. Pro definování dynamické třídy je vložen do prostředního sloupce například znak X. Tato dynamická třída by měla 8 segmentů, každý s oblastí vyhrazeného prostoru o 3 megabytech (MB) a s velikostí programu 1 megabyte (MB). Oblast dynamického prostoru GETVIS je nastavena na 64 kilobytů (kB), atribut "enabled" je nastaven na 1, což znamená, že třída je zpřístupněna. Profil, který má být zpracován je JCL procedura STDPROF. Všechny ostatní položky jsou nastaveny na implicitní hodnoty. Jestliže jsou všechny tyto hodnoty nastaveny, jsou zavedeny stlačením spouštěcí klávesy.

Tím se opět zobrazí panel "maintain dynamic partitions" (udržované dynamické segmenty), ale nyní ukazující v řádku 4 připojenou dynamickou třídu X se všemi atributy, jak je označeno na obr. 7. Tak, jestliže je obr. 7 srovnán s panelem zobrazeným na obr. 5, existují nyní čtyři dynamické třídy C, Y, Z a X. Stisknutím funkce "process" (zpracovat), která je spojena s funkční klávesou 5, se zobrazená tabulka dynamických tříd nyní zapíše jako nová kopie do systémové knihovny. Tak jsou nyní v systému definované a po aktivaci použitelné dynamické segmenty ve čtyřech různých třídách s maximálním počtem 28 dynamických segmentů.

Obr. 8 ukazuje panel pro aktivaci tabulky dynamických tříd. Tabulka dynamických tříd, tedy i dynamická třída X, může být použita po vyslání příkazu PLOAD DYNCTAB programu

VSE/POWER ze systémové konzole. Potom systém zobrazí aktivní dynamické třídy. Jak je znázorněno na obr. 8, žádný ze segmentů ve čtyřech třídách není v danou chvíli aktivní. Jak je indikováno v řídce 09, tabulka dynamických tříd byla úspěšně zavedena.

Nyní je možné zpracovat zakázku. K této závěrečné činnosti je zadán proud prací programu VSE/POWER s dynamickou třídou X na pozici parametru třídy příkazu zakázky programu VSE/POWER. Obr. 9 ukazuje scénář, jak taková zakázka vypadá.

Panel zobrazený na obr. 10 ukazuje, jak zakázka vypadá při svém provádění. Když bude dynamický segment uvnitř dynamické třídy X použitelný, tato zakázka bude provedena, jako na ukázaném příkladu, v dynamickém segmentu X1. Po ukončení této zakázky bude tento dynamický segment uvolněn a bude použitelný pro jinou zakázku.

Výhody koncepce ve shodě s vynálezem jsou, že uživatel musí pouze definovat systémové provozní prostředky použitelné dynamickými segmenty (např. kapacitu virtuální paměti, tabulku dynamických tříd). Provádění zakázky, včetně přidělení segmentu a uvolnění segmentu, je prováděno systémem, to jest, systémové provozní prostředky jsou použity pouze tehdy, jsou-li potřeba. Není obtížné přesouvat zakázky ze statických do dynamických segmentů, poněvadž neslučitelnosti (řídící bloky, služby apod.) jsou pokud možno minimalizovány. Předkládaný vynález poskytuje pro operační systém VSE dodatečné, nové segmenty. Tím může být prováděno paralelně mnohem více programů. Koncepce je velmi pružná a přenositelná do jiných operačních systémů. Tím je pojem dynamického segmentu otevřen pro další rozšiřování.

Dodatek:**Definice**

Některé definice v tomto slovníku jsou zkopírovány z manuálu IBM VSE/AF Planning and Installation Version 4 Release 1, SC33-6351.

A

adress space (adresní prostor)
Pododdíl celkové virtuální paměti.

ASI - automated system initialization (automatizovaná systémová inicializace)

Funkce, která dovoluje, aby řídící informace pro systémovou inicializaci byly katalogizovány pro automatické vyhledávání během systémové inicializace.

ASI procedure (procedura)

Množina řídících příkazů, které specifikují hodnoty pro automatickou systémovou inicializaci.

C

control program (řídící program)

Program ve strojovém kódu navržený k rozvržení programů a dohlížení na provádění programů počítačového systému.

D

dynamic partition (dynamický segment)

Segment přidělený pro jednu zakázku, vytvořený a aktivovaný na základě potřeby. Po zpracování zakázky je obsazený pros-

tor uvolněn.

G

GETVIS area (oblast)

Paměťový prostor uvnitř segmentu nebo sdílené virtuální oblasti použitelný pro dočasné přidělování programům.

J

job (zakázka)

/1/ Množina dat, která pro počítač úplně definuje jednotku práce. Zakázka obvykle obsahuje všechny nutné programy ve strojovém kódu, spojky, soubory a instrukce operačnímu systému.

/2/ Skutečné zpracování jednotky práce počítačem.

JCL - job control language (jazyk pro řízení prací)

Řídící jazyk, který může být použit:

/a/ k identifikaci zakázky operačnímu systému,

/b/ k popsání požadavků zakázky.

M

main task (hlavní úloha)

Segment je spuštěn jednou úlohou zvanou hlavní úloha. Tato úloha provádí zakázku a může zřídit další úlohy (podúlohy).

P

partition (segment)

Oddíl adresního prostoru, který je použitelný pro provádění programu. Avšak řídící program supervizoru nemůže běžet v tomto prostoru.

private area (vyhrazená oblast)

Oddíl adresního prostoru, kde mohou být přiděleny jeden nebo více segmentů, specifický mezi všemi jinými adresními prostory.

program (program)

Posloupnost instrukcí vhodná ke zpracování počítačem.

program area (oblast programu)

Oblast segmentu, kam má být zaveden program.

S

shared area(s) (sdílená(é) oblast(i))

Oddíl adresního prostoru společný pro všechny adresní prostory. Sdílené segmenty, sdílená virtuální oblast a supervizor patří do sdílené oblasti.

shared partition (sdílený segment)

V operačním systému VSE segment přidělený programu, který poskytuje služby pro programy a komunikuje s programy v jiných segmentech systémového prostoru virtuálních adres. Sdílený segment je část sdílených oblastí.

shared virtual area (SVA) (sdílená virtuální oblast)

V operačním systému VSE oblast, která obsahuje seznam často užívaných fází, rezidentní programy společné pro segmenty a oblast pro podporu systému.

spooling (spooling, souběžná periferní operace)

Snímání a zapisování vstupních a výstupních proudů dat na pomocném zařízení ve formátu vhodném pro pozdější zpracování nebo výstup.

static partition (statický segment)

Segment, který je přidělen operátorem nebo ASI procedurou a je obvykle použitelný pro více než jednu zakázku.

subtask (podúloha)

Úloha, která je inicializována hlavní úlohou nebo jinou (pod)úlohou.

supervisor (supervizor)

Část řídícího programu, která koordinuje užití provozních prostředků a udržuje průběh procesorových operací.

T

task (úloha)

Základní jednotka synchronního provádění programu. Úloha soutěží s jinými úlohami o systémové provozní prostředky, jako jsou doba zpracování a vstupní/výstupní kanály.

time slicing (dávkování času)

Režim provozu, ve kterém jsou několika procesům přiřazena kvanta času stejného procesoru.

V

VSE/POWER

Licenční program firmy IBM primárně použitý pro zprostředkování vstupu a výstupu, je to zkratka pro Virtual Storage Extended/Priority Output Writer Execution Reader.

040597	02.IX.91	ÚŘAD PRŮMYŠLEVNÍ VLASTNOSTI ČESKÉ REPUBLIKY	Příl.
--------	----------	--	-------

PATENTOVÉ NÁROKY

1. Počítač

S koncepcí ^{rozšířené} virtuální paměti, jež poskytuje paměťový prostor pro paralelní provádění programů v předběžně přidělených segmentech paměti virtuálních adres, obsahující sdílené oblasti a vyhrazené oblasti v několika adresních prostorech vyznačený tím, že v závislosti na zakázce, která má být zpracována, jeden ^{nebo} více dynamických segmentů (P1, P2, ..., N1, N2, ..., O1, O2, ...) je navíc přiděleno ke zmíněným předběžně přiděleným segmentům (F1-F8, FA, FB, BG) a uvolněno po ukončení zakázky, aby používaná oblast paměti byla neobsazená pro následující použití.

2. Počítač podle nároku 1 vyznačený tím, že v něm každý dynamický segment má svůj vlastní adresní prostor a každý dynamický segment je vytvořen ve vyhrazené oblasti takového adresního prostoru.

3. Počítač podle nároku 1 nebo 2 vyznačený tím, že v něm více dynamických segmentů může být přidělováno najednou.

4. Počítač podle libovolného z předchozích nároků vyznačený tím, že v něm pro zmenšení velikosti sdílené oblasti je poskytnuta vyhrazená systémová oblast ve vyhrazené oblasti pro adresní prostory dynamických segmentů, oblast dynamického prostoru GETVIS, která obsahuje systémová data pro adresní prostor a je dočasně přidělována společně se zmíněným dynamickým segmentem.

5. Počítač podle libovolného z předchozích nároků vyz-

načený tím, že v něm zmíněné dynamické segmenty jsou seskupeny do dynamických tříd, ve kterých jsou zakázky proveditelné.

6. Počítač podle nároku 5 vyznačený tím, že v něm může být specifikována priorita zmíněné dynamické třídy.

7. Počítač podle nároku 5 nebo 6 vyznačený tím, že v něm dynamické segmenty uvnitř stejné třídy jsou časově dávkovány.

8. Počítač podle libovolného z nároků 5 až 7 vyznačený tím, že v něm předběžně přidělené statické a dynamické segmenty mohou být seskupeny v jednu a tutéž třídu.

9. Počítač podle libovolného z předchozích nároků vyznačený tím, že v něm je poskytnuta tabulka dynamických tříd obsahující atributy zmíněných dynamických segmentů, přičemž zmíněná tabulka dynamických atributů může být katalogizována v knihovně, odkud může být zavedena.

10. Počítač podle nároku 9 vyznačený tím, že v něm některé atributy dynamické tabulky jsou:

a/ rezervace paměti,

specifikující požadavek na kapacitu paměti dynamického segmentu, zvláště pro dynamický prostor GETVIS a vyhrazení segmentu, dané třídy,

b/ velikost segmentu,

definující množství souvislé virtuální paměti v segmentu vyhrazené pro provádění programu,

c/ velikost dynamického prostoru GETVIS,

definující množství souvislé virtuální paměti pro oblast dynamického prostoru GETVIS,

d/ profil nebo procedura,

specifikující profil nebo proceduru, která bude prove-

dena v segmentu před zakázkou,

e/ maximální počet dynamických segmentů uvnitř jedné třídy,

specifikující maximální počet segmentů, které mohou být paralelně vyhrazeny uvnitř dané třídy, když je použitelný dostatek virtuální paměti a dynamických segmentů,

f/ indikace zneprístupnění,

dovolující zneprístupnit dynamickou třídu, tj., žádná zakázka nemůže být provedena uvnitř této třídy,

g/ manipulovaná vstupní/výstupní zařízení,
specifikující manipulovaná zařízení, např. snímač, tiskárnu, dšrovače, které se propojují na vstupní/výstupní převíjecí mechanismus.



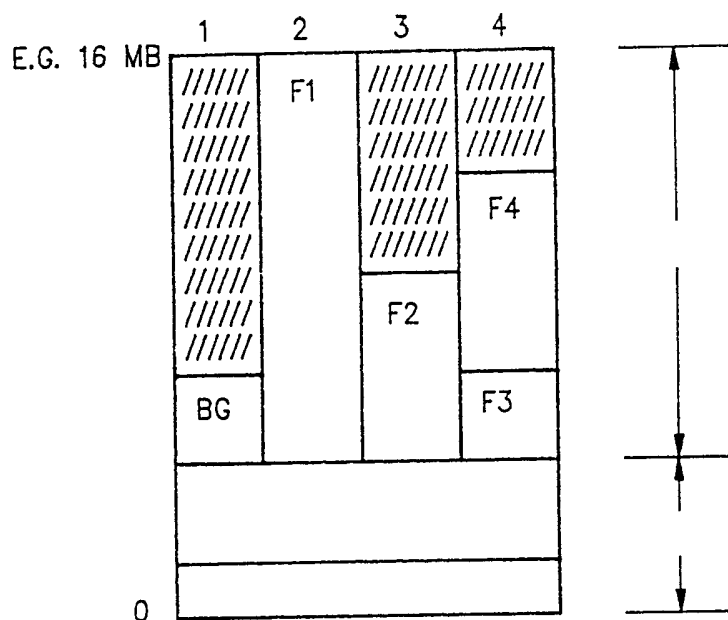


FIG. 1

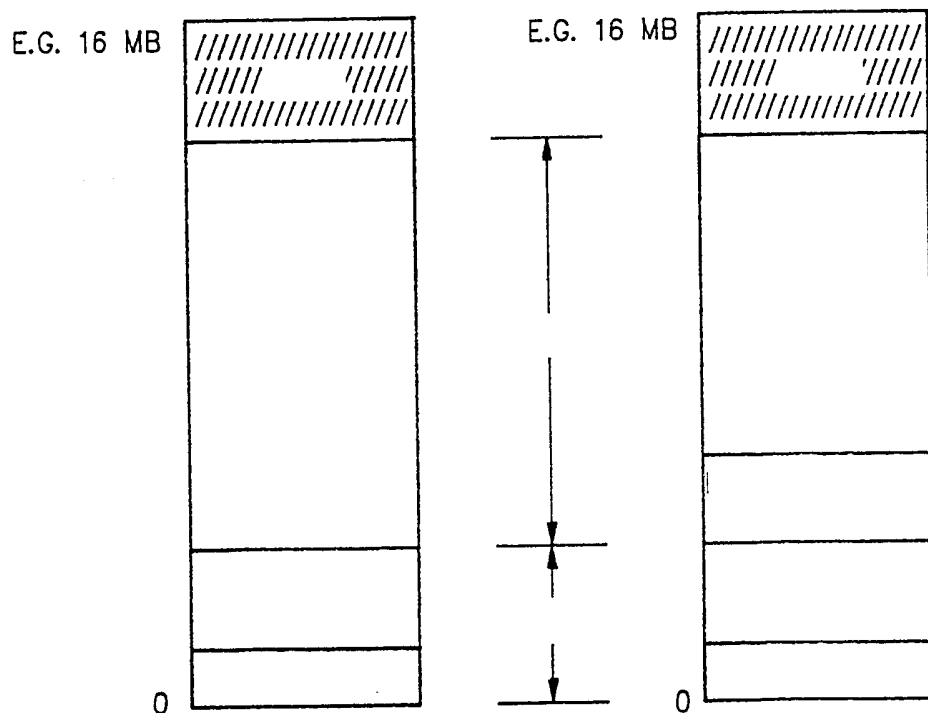


FIG. 2

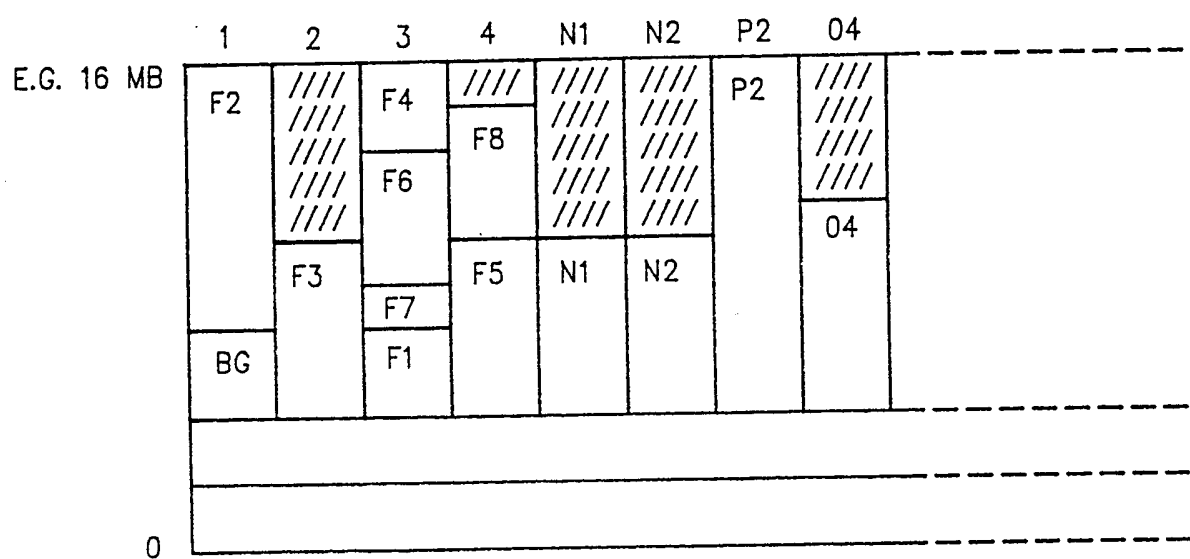


FIG. 3

IESADMSL.IESEDEF RESOURCE DEFINITION APPLID: CICS

Enter the number of your selection and press the ENTER key:

- 1 User Interface Tailoring
- 2 File and Catalog Management
- 3 Display VTOC
- 4 Hardware Configuration and IPL
- 5 Maintain VTAM Application Names
- 6 Maintain VTAM Startup Options
- 7 Maintain Dynamic Partitions

6=ESCAPE(U)

4=RETURN

3=END
9=Escape(m)

PF1=HELP

Path: 2

==> 7

FIG. 4

```

TAS$DYN1          MAINTAIN DYNAMIC PARTITIONS

ENTER OPTION(S) AND PRESS ENTER.

OPTIONS:  1 = ADD          2 = ALTER          5 = DELETE

OPT  DYNAMIC  ENABLED  MAX NO. OF  STORAGE  MAXIMUM  DYNAMIC  PROFILE
CLASS  CLASS    1 = YES  PARTITIONS  ALLOCATION  PROGRAM  SPACE   GETVIS   SIZE
-----
1      C      1      9          1M        500K     64K     STDPROF
-      Y      1      8          2M        1024K    64K     STDPROF
-      Z      1      3          5M        1024K    64K     STDPROF
-
-
-
-
-
-
-
-
PF1=HELP          2=REDISPLAY  3=END          5=PROCESS

```

FIG.5

TAS\$DYN2 MAINTAIN DYNAMIC PARTITIONS

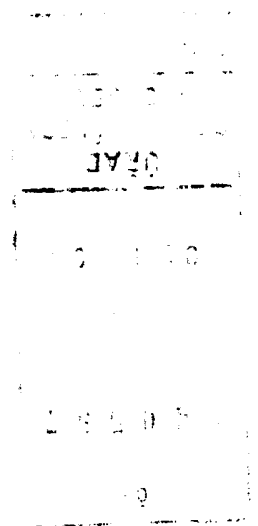
Enter the required data and press ENTER.

DYNAMIC CLASS.....	X	Enter one of the classes C - E, G - Z
NUMBER OF PARTITIONS...	8	Enter a number between 1 and 32
STORAGE ALLOCATION.....	3M	Specify in M bytes
MAXIMUM PROGRAM SIZE...	1M	Specify in K or M bytes
DYNAMIC SPACE GETVIS...	64K	Specify in K bytes
ENABLED.....	1	1=YES, 2=NO
PROFILE.....	STDPROF	Name of the JCL procedure
NUMBER OF LOGICAL UNITS	50	Enter a number between 10 - 255

READER-DEVICE.....	FEC	----	----	----
LIST OF PRINTER DEVICES	FEE	----	----	----
LIST OF PUNCH DEVICES..	FED	----	----	----
	----	----	----	----

PF1=HELP 2=REDISPLAY 3=END

FIG. 6



TAS#DYN1

MAINTAIN DYNAMIC PARTITIONS

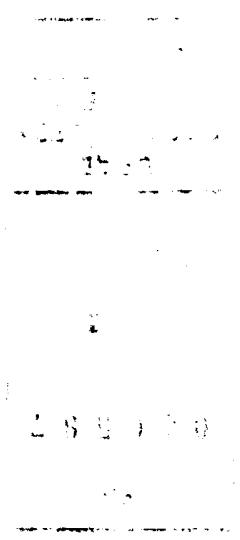
ENTER OPTION(S) AND PRESS ENTER.

OPTIONS: 1 = ADD 2 = ALTER 5 = DELETE

OPT	DYNAMIC CLASS	ENABLED 1 = YES 2 = NO	MAX NO. OF PARTITIONS	STORAGE ALLOCATION	MAXIMUM PROGRAM SIZE	DYNAMIC SPACE GETVIS	PROFILE
-	C	1	9	1M	500K	64K	STDPROF
-	Y	1	8	2M	1024K	64K	STDPROF
-	Z	1	3	5M	1024K	64K	STDPROF
-	X	1	8	3M	1024K	64K	STDPROF

PF1=HELP 2=REDISPLAY 3=END 5=PROCESS

FIG. 7



```

01 PLOAD DYNCTAB
02 AR 015 1C39I COMMAND PASSED TO VSE/POWER
03 F1 001 1Q6AI ***** DISPLAY OF ACTIVE
04 F1 001 1Q6AI CLS STATE ACT/MAX ALLOC DYNAMIC CLASS TABLE *****
05 F1 001 1Q6AI C ENAB 0 9 1M 500K 64K STDPROF 50
06 F1 001 1Q6AI Y ENAB 0 8 2M 1024K 64K STDPROF 50
07 F1 001 1Q6AI Z ENAB 0 3 5M 1024K 64K STDPROF 50
08 F1 001 1Q6AI X ENAB 0 8 3M 1024K 64K STDPROF 50
09 F1 001 1Q6BI DYNAMIC CLASS TABLE LOADED SUCCESSFULLY
10
11
12
13
14
15
16
17
18
19
20

```

FIG. 8

```

=====
<<.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+..... MEM=TESTX >
*  * JOB JNM=TESTX,DISP=L,CLASS=X
// JOB TESTX
// EXEC TEST
/&
*  * EOL
***** END OF FILE *****

```

FIG. 9

```

01 AR 015 1C39I COMMAND PASSED TO VSE/POWER
02 F1 001 1Q6AI ***** DISPLAY OF ACTIVE
03 F1 001 1Q6AI CLS STATE ACT/MAX ALLOC DYNAMIC CLASS TABLE *****
04 F1 001 1Q6AI C ENAB 0 9 1M 500K 64K STDPROF 50
05 F1 001 1Q6AI Y ENAB 0 8 2M 1024K 64K STDPROF 50
06 F1 001 1Q6AI Z ENAB 0 3 5M 1024K 64K STDPROF 50
07 F1 001 1Q6AI X ENAB 0 8 3M 1024K 64K STDPROF 50
08 F1 001 1Q6BI DYNAMIC CLASS TABLE LOADED SUCCESSFULLY
09 F1 001 1Q47I X1 TESTX 00286 FROM (SALN) , TIME=13:07:56
10 X1 020 // JOB TESTX
11 DATE 08/29/90,CLOCK 13/07/56
12 X1 020 EOJ TESTX
13 DATE 08/29/90,CLOCK 13/08/13,DURATION 00/00/17
14 F1 001 1Q3EI DYNAMIC CLASS 'X' WAITING FOR WORK
15
16
17
18
19
20

```

FIG.10

