



(19) Országkód

HU



**MAGYAR
KÖZTÁRSASÁG**

**MAGYAR
SZABADALMI
HIVATAL**

SZABADALMI LEÍRÁS

(21) A bejelentés ügyszáma: 2843/91
(22) A bejelentés napja: 1991. 09. 02.
(30) Elsőbbségi adatok:
90116847.6 1990. 09. 03. EP

(40) A közzététel napja: 1992. 03. 30.
(45) A megadás meghirdetésének a dátuma a Szabadalmi
Közlönyben: 1999. 06. 28.

(11) Lajstromszám:

216 474 B

(51) Int. Cl.⁶

G 06 F 12/00

(72) Feltaláló:

Salm, Ingolf, Gäufelden (DE)

(73) Szabadalmaz:

International Business Machines Corp., Armonk,
New York (US)

(74) Képviseelő:

S. B. G. & K. Budapesti Nemzetközi Szabadalmi
Iroda, Budapest

(54) Eljárás számítógépes rendszer virtuális tárolójának előre meghatározott kapacitású címterületekre osztására

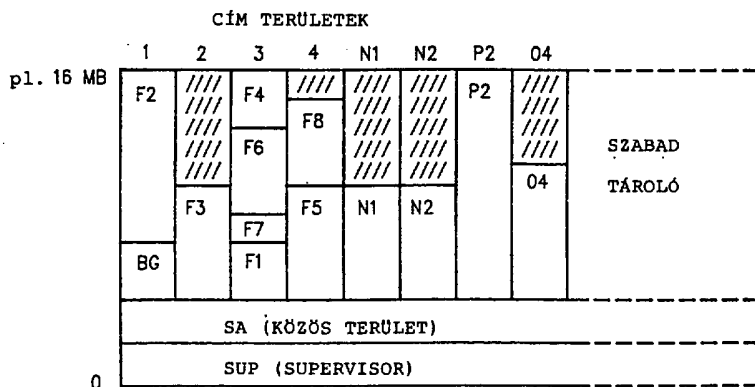
KIVONAT

A találmány tárgya eljárás számítógépes rendszer virtuális tárolójának előre meghatározott kapacitású címterületekre osztására a címterületekhez előre hozzárendelt, adott számú partícióban (F1–F8, BG) párhuzamos programfuttatáshoz, ahol a címterületek közös (shared) és saját (private) területeket tartalmaznak.

A találmány szerinti eljárás során a feladatok (jobs) párhuzamos végrehajtásához egy vagy több dinamikus

partíciót hozzárendelnek a címterületekhez (P2, N1, N2, O4), ahol a hozzárendelt dinamikus partíciók mérete a végrehajtandó feladattól függ, és ahol

mindegyik dinamikus partíciónak saját címterülete (P2, N1, N2, O4) van, és csak egy dinamikus partíció van a címterület (P2, N1, N2, O4) saját területéhez hozzárendelve,



3. ábra

elindítják és vezérlik a feladatok végrehajtását, és megszüntetik a dinamikus partíció hozzárendelését, amikor a hozzárendelt feladat végrehajtása befejeződött, újabb felhasználáshoz felszabadítják a dinamikus partíció által lefoglalt címtérületet (P2, N1, N2, O4), továbbá

ahol a dinamikus partíciókkal együtt átmenetileg dinamikus címtérületeket (GETVIS) jelölnek ki a saját (private) rendszerterületen, amely rendszer adatokat tartalmaz a címtérülethez az adott feladat számára.

A találmány tárgya eljárás számítógépes rendszer virtuális tárolójának előre meghatározott kapacitású címtérületekre osztására a címtérületekhez előre hozzárendelt, adott számú partícióban párhuzamos programfuttatáshoz, ahol a címtérületek közös (shared) és saját (private) területeket tartalmaznak.

Ilyen eljárást alkalmazó operációs rendszerrel, mint például az IBM virtuális, bővített tárolású/rendszer csomaggal (Virtual Storage Extended/System Package) 4. változat, 1. kiadás (a következőkben röviden VSE/SP 4.1.) rendelkező számítógép a piacon ismert, és részletesen le van írva az IBM-kézikönyvekben, amelyek közül az Administration, Form Number SC33-6405-01 számú idézzük mint referenciaanyagot. A VSE/SP 4.1 operációs rendszer magában foglalja egyik programként az IBM által védett VSE/Advanced Functions programot (következőkben röviden: VSE/AF), valamint más programokat, mint például a VSE/POWER programot. Hivatkozunk továbbá az IBM-kézikönyv IBM VSE/Advanced Functions System Management Guide Version 4 Release 1, Form Number SC33-6352-00 anyagra is. Ebben a kézikönyvben különösen az 1-20. oldalakon részletesebb leírás található a többszörös programozásról, a virtuális tárolóhasználatról és a valós tárolószervezésről, a particionálásról és a helykijelölésről.

A jelen rendszerben a virtuális tárolás tizenkét partícióra korlátozódik kilenc címtérületen. Ez a párhuzamosan futtatandó programokhoz a megadott címtérületen igen gondos és bonyolult tárolóhely-kijelölést igényel. Ez a programok párhuzamos végrehajtását számottevően korlátozza, és így a rendszert bizonyos mértékig merevvé teszi.

A jelen találmánynak az elsődleges célja az, hogy a rendszert rugalmasabbá tegye, megnövelje a virtuális tárolási forrásokat úgy, hogy párhuzamosan több programot lehessen futtatni egyszerű eszközökkel és a meglévő rendszerrel összeférhető módon.

A találmány célját a legáltalánosabban olyan eljárással érhetjük el, amelynél a feladatok (jobs) párhuzamos végrehajtásához egy vagy több dinamikus partíciót hozzárendelünk a címtérületekhez, ahol a hozzárendelt dinamikus partíciók mérete a végrehajtandó feladattól függ, és ahol

mindegyik dinamikus partíciónak saját címtérülete van, és csak egy dinamikus partíció van a címtérület saját területéhez hozzárendelve,

elindítjuk és vezéreljük a feladatok végrehajtását, és megszüntetjük a dinamikus partíció hozzárendelését, amikor a hozzárendelt feladat végrehajtása befejeződött, újabb felhasználáshoz felszabadítjuk a dinamikus partíció által lefoglalt címtérületet, továbbá

10 ahol a dinamikus partíciókkal együtt átmenetileg dinamikus címtérületeket jelölünk ki a saját rendszerterületen, amely rendszer adatokat tartalmaz a címtérülethez az adott feladat számára.

15 Az így újonnan bevezetett dinamikus partíciókkal, amelyeket egy feladattal együtt jelölünk ki (allokálunk) és a feladat befejezésével a kijelölést töröljük (deallokalunk), egy nagymértékű virtuális tárolásiterület-kiterjesztés hozható létre, egészen 200 vagy annál több új partícióra. Ez azt jelenti, hogy sokkal több programot lehet párhuzamosan futtatni, és a rendszer jelenleg fennálló korlátozásait megszüntethetjük.

A további előnyös kiviteli változatok megismerhetők az aligénypontokból, amelyeket a következő leírásban ismertetünk részletesebben a mellékelt rajzra hivatkozással, ahol az

- 25 1. ábra a technika állásából ismeretes, előre kijelölt (allokált) vagy statikus partíciójú tárolóelrendezés vázlata, a
2. ábra bemutatja egymás mellett a találmány szerinti címtérület-elrendezést a statikus és dinamikus partícióra;
3. ábra a tárolóelrendezés egy példáját mutatja, a találmány szerinti dinamikus partíciókkal;
4. ábra (táblázat) bemutat egy képernyőn megjelenített menüt a feladat kiválasztására szolgáló forgatókönyvvél: fenntartva a dinamikus particionálást;
5. ábra (táblázat) egy képernyőn megjelenített menüt mutat a dinamikus osztály hozzáadásával a forgatókönyvvél;
6. ábra (táblázat) egy képernyőmenüt mutat be, az X dinamikus osztály meghatározásának a forgatókönyvvél;
7. ábra egy képernyőmenüt mutat be, a könyvtár számára megírandó, változtatott dinamikusosztály-táblázatnak a forgatókönyvvél;
8. ábra egy rendszerkonzolképet mutat a dinamikus-osztály-táblázat betöltésére szolgáló forgatókönyvvél;
9. ábra (táblázat) egy képernyőmenüt mutat be egy VSE/POWER feladat létrehozásának a forgatókönyvvél (X osztály) és a
10. ábra (táblázat) bemutat egy rendszerkonzolképet, a VSE/POWER feladat végrehajtásának a forgatókönyvvél.

55 A dinamikus particionálási koncepció jobb megértéséhez, és hogy az hogyan ágyazódik be a VSE operációs rendszerbe, röviden ismertetjük a jelen találmány szerinti particionálási rendszert. A leírásban a jelenlegi partíciókat statikus vagy előre kijelölt partícióknak nevezzük.

Az előre kijelölt, statikus partíciók:

A jelenlegi VSE változat (VSE/SP 4.1) lehetővé teszi, hogy legfeljebb 9 címtérületen 12 partíciót jelöljünk ki. Egy címtérület közös vagy megosztott (shared) és saját (private) területet tartalmaz. A megosztott terület az összes címtérület számára ugyanaz. A saját terület a kijelöléstől függően különbözik. Lehetséges, hogy egynél több partíciót jelöljünk ki egy saját területen. A statikus partícióknak a következő, előre meghatározott azonosítói vannak. F1, F2, F3, F4, F5, F6, F7, F8, F9, FA, FB, BG.

Az 1. ábrán bemutatott tárolóelrendezés a technika állásának egy példaként leír egy rendszert, öt BG, F1, F2, F3 és F4 partícióval, amelyek 1–4 címtérületen vannak elhelyezve. Egy címtérület például maximum 16 megabájt nagyságú lehet. A címtérületek SAS közös területekre és PA saját területekre vannak felosztva. A PA saját terület a programterület, a program végrehajtásához. Egy operátor- vagy ASI (automatikus rendszerinitializálás) eljárás kijelölheti és indíthatja a partíciókat, és mindegyik partíciónak egy végrehajtási prioritást adhat. A statikus partícióknak ez az előre kijelölése a rendszer felépítése (system setup) fázisában történik.

A statikus partíciók JCL (Job Control Language – feladat-vezérlőnyelv) utasításokat kaphatnak egy fizikai olvasótól vagy a feladatütemezőtől. A VSE-nél a feladatütemezést, a végrehajtás-vezérlést és a feladatok kimenőspoolingját a VSE/POWER végzi el, tehát a feladatot a VSE/POWER vezérli. A leírásban a spooling bemeneti vagy kimeneti adatáram olvasását vagy írását jelenti egy közbelső eszközzel és olyan formátumban, ami a későbbi feldolgozást megkönnyíti. VSE/POWER (POWER jelentése Priority Output Writer Execution Reader, és ez egy, az IBM által licenccben adott program, amelyet elsősorban a bemenet és a kimenet spoolingjára használnak) indíthatja a statikus particionálást és futtathat feladatokat azokban, tehát a VSE/POWER JCL-utasításokat és adatokat tölt a partíciókba. A feladatok az olvasósorban vannak elhelyezve. A listázó- vagy lyuk-kártyakimenet a megfelelő sorokhoz irányítható.

A statikus partíció egynél több feladathoz áll rendelkezésre. A VSE/POWER lehetővé teszi, hogy a feladatokat osztályokban, az úgynevezett VSE/POWER osztályokban hajtsa végre, ahol egy vagy több statikus partíciót lehet kijelölni. Egy osztály a feladatok egy csoportját jelenti, amelyek vagy ugyanarról a bemenőkészületről jönnek, vagy ugyanahhoz a kimenőkészülethez mennek.

Dinamikus particionálás:

A spoolingprogram, például a VSE/POWER kijelöl és indít egy dinamikus partíciót egy feladathoz. A VSE/POWER vezérli a feladat végrehajtását, és megszünteti a dinamikus partíció kijelölését a feladat befejezése után. A felszabadított tároló rendelkezésre áll egy másik feladathoz.

Mindegyik dinamikus partíciónak megvan a saját címtérülete, tehát csak egy dinamikus partíció van kijelölve egy ilyen címtérület saját területén. Ennek következtében több dinamikus partíció is ki lehet jelölve egyidejűleg.

Annak érdekében, hogy csökkentsük a közös területek nagyságát, egy új saját rendszerterület kerül bevezetésre a dinamikus particionálás címtérületekhez, amit GETVIS dinamikus címtérületnek nevezünk. A GETVIS dinamikus címtérület a saját területéhez tartozik. A 2. ábra egy összehasonlítást ad a statikus és dinamikus particionálású címtérület felosztásáról.

A 2. ábra bal oldali része a statikus partíció szervezését mutatja, amely SUP felügyelő- (supervisor) területet és SA közös területet tartalmaz a SAS közös területeken és PRA programterületet a PA saját területen, ahol egy PGA partíciós GETVIS terület is lehet, amely a partícióhoz tartozó programhoz ideiglenesen hozzárendelődik (allokálódik).

A 2. ábra jobb oldala a találmány egy kiviteli alakjával összhangban a dinamikus partíció szervezését mutatja. Amint az ábrán látható, a statikus partíciók szervezésével ellentétben, a PA saját területen van egy GETVIS dinamikus címtérület is.

A GETVIS dinamikus címtérület a dinamikus partícióval együtt ideiglenesen van kijelölve, és rendszeradatokat tartalmaz a címtérület számára. Ezek a rendszeradatok egyediek a feladathoz, és így a kijelölt dinamikus partícióhoz is. Ezért ezek az adatok nem szükségesek a másik partíciókhoz, és nem szükséges ezeknek a közös területen tartása. Mivel az SA közös terület mentes ezektől az adatoktól, a mérete is kisebb lehet.

A dinamikus partíciókat a VSE/POWER osztályoknak megfelelő osztályokba lehet csoportosítani, ahol a feladatok végrehajthatók. Ezeket az osztályokat dinamikus osztályoknak hívják.

Csak a dinamikus osztály prioritását lehet specifikálni. A dinamikus partíciókat egy és ugyanazon osztályon belül időben felszeleteljük.

A statikus és dinamikus partíciókat egy VSE/POWER osztályba lehet csoportosítani.

Egy táblázat, az úgynevezett dinamikusosztály-táblázat tartalmazza a dinamikus osztályok tulajdonságait. A dinamikusosztály-tábla a könyvtárban katalogizálható, és onnan betölthető.

Néhány példa a dinamikusosztály-táblázat jellemzőire:

- tároló kijelölés/elhelyezés (allokáció): meghatározza egy dinamikus particionálás (GETVIS dinamikus címtérület- és partíciókijelölés) tárolási követelményeit egy megadott osztályban,
- a partíció mérete: meghatározza az összefüggő virtuális tároló nagyságát a program végrehajtáshoz fenntartott partícióban; a partíció fennmaradó része mint partíciós GETVIS dinamikus címtérület áll rendelkezésre,
- a GETVIS dinamikus címtérület nagysága: meghatározza az összefüggő virtuális tároló nagyságát a GETVIS dinamikus címtérület részére (lásd a 2. ábrát),
- profile (eljárás): a partícióban a feladat (job) előtt hajtódik végre (megfelel a statikus partíciók ASI eljárásának)
- osztályon belül a dinamikus partíciók maximális száma:

meghatározza az adott osztályon belül párhuzamosan kijelölhető partíciók maximális számát, ha elegendő virtuális tároló- és dinamikus partíció áll rendelkezésre;

– letiltás jelzése:

lehetővé teszi egy dinamikus osztály letiltását, tehát ezen osztályon belül nem lehet semmilyen feladatot végrehajtani;

– sorba állított (spooled) I/O készülékek:

meghatározza azokat a sorba állított készülékeket (olvasó, nyomtatók és lyukasztók, amelyek a VSE/POWER-hez kapcsolódnak.

– a dinamikuspartíció-azonosítók felépítése:

<osztály> <pno>

ahol <osztály> = dinamikus osztály (egy karakter, amelyet a felhasználó határoz meg)

<pno> = a partíció száma a dinamikus osztályon belül. Például a P dinamikus osztályban kijelölt első dinamikus partíció a P1 azonosítót kapja.

A következő példa, amint azt a 3. ábra mutatja, egy tárolóelrendezést mutat BG, F1–F8 partíciókkal az 1–4 címtérületeken, dinamikus partíciókkal az N dinamikus osztályban, N1 és N2 címtérületeken, egy dinamikus partícióval a P2 címtérületen és egy dinamikus partícióval az O4 címtérületen.

A 3. ábrán bemutatott tárolóelrendezés példájánál az N dinamikus osztályban csak az N1 és N2 címtérületek vannak kijelölve egyszerre, míg a többi címtérület ebben az osztályban nem aktív. Ugyanez áll a P osztály P2 címtérületére, ahol ez az egyetlen aktív dinamikus partíció a P osztályban. Az O osztályból csak az O4 címtérület aktív a bemutatott példánál. Ennek a példának az elrendezése azt mutatja, hogy a virtuális tárolóterületen csak azok a dinamikus partíciók aktívak, amelyeknek ténylegesen futó feladatai vannak. Ha a dinamikus partíciókhoz kijelölt feladatok nem aktívak, azoknak a kijelölt virtuális tárolóit fel lehet használni más célokra. Ez azt mutatja, hogy a dinamikuspartíció-kon koncepció a virtuális tároló optimális kihasználását teszi lehetővé. Ezt fel lehet használni, mivel a szabad tárolókapacitásnak mindig csak az a virtuális tárolója használatos, amire tényleg szükség van.

A dinamikus partíciók és a használatuk egyszerűsége:

A következőkben bemutatjuk, hogy milyen könnyű a feladatok feldolgozása a dinamikus partíciókban.

1. Meghatározás:

A dinamikusosztály-táblázat lehetővé teszi, hogy a dinamikus partíciók jellemzőit (attributes) már néhány utasítással (statements) meghatározzuk (lásd az 5., 6. és 7. ábrát is).

2. Dinamikus partíciók igény esetén kerülnek kijelölésre:

A dinamikus partíciók csak akkor kerülnek kijelölésre, ha egy dinamikus osztályban egy feladatot kell végrehajtani. Ez takarékoskodik a rendszer erőforrásaival. Ha egy VSE/POWER osztály statikus és dinamikus partíciókat tartalmaz, általában a statikus partíciókat használjuk először a feladat végrehajtásához.

3. Automatikus kijelölésinicializálás és deaktiválás:

A feladatok környezetét és erőforrásait automatikusan jelöljük ki és indítjuk a felhasználó beavatkozása nélkül.

4. Azonos erőforrás-kijelölés egy dinamikus osztályon belül:

A feladathoz szükséges erőforrások a feladat végrehajtása előtt kerülnek kijelölésre.

5. Időosztás:

Nincs szükség arra, hogy a CPU intenzív feladatokat vezéreljünk, mivel, miután egy feladatnak az időrése ki lett merítve, a feladatot átesszük a legalacsonyabb prioritású helyzetbe a dinamikus osztályon belül.

6. A statikusból a dinamikus partícióba való mozgás:

15 A feladatokat könnyű áttenni a dinamikus partíciókba, mivel a rendszer kiküszöböli az összeférhetetlenségeket.

A koncepció:

20 A koncepció mutatja, hogy hogyan lehet meghatározni a dinamikus osztályokat, és hogyan lehet létrehozni és felszabadítani a dinamikus partíciókat, ha a rendszer erőforrásai (például virtuális tároló) rendelkezésre állnak.

A következő szakaszok leírják

25 a) a dinamikusosztály-táblázat karbantartását,
b) a dinamikus partíció szervezését (dispatching),
c) a feladat végrehajtását.

1) Az elvégzendő feladatot egy dinamikus osztályba helyezzük,

30 2) VSE/POWER dinamikuspartíció-kijelölést igényel – a dinamikus partíció létrehozása,

3) VSE/POWER indítja a dinamikus partíciót – a dinamikus partíciót kioszthatóvá (dispatchable) tesszük,

35 4) a dinamikus partíció inicializálása – a címtérület inicializálása,
– a dinamikusosztály-profil futtatása,

5) a feladat végrehajtása,

6) a feladatfeldolgozás vége,

40 7) a dinamikus partíció felszabadítása – a rendszererőforrások szabaddá tétele,
– a partíció leállítása,

– a dinamikus partíciót kioszthatatlanná (undispatchable) tesszük,

45 8) a dinamikus partíció kijelölésének törlése

– a partíció eltávolítása a szervezési (dispatching) sorból.

A dinamikusosztály-táblázat karbantartása (létrehozás, tárolás és betöltés).

50 Bármelyik szerkesztő vagy a VSE dialógprogram létrehozhatja és megváltoztathatja a dinamikusosztály-táblázatot. A táblázat a könyvtárban tárolódik, ahonnan a kezelő vagy a VSE/POWER ASI eljárás betöltheti a táblázatot.

55 A betöltőfolyamat érvényesíti a dinamikusosztály-táblázat értékeit, és átveszi az információt egy belső formátumba. Az érvénytelen beadott adatok a kijelzőn megjelennek. Csak az érvényes beadott adatok aktiválhatók. A dinamikusosztály-táblázat a rendszertárolóban tárolódik, tehát az címezhető a rendszerrutinok által.

A dinamikus osztályok egy előre meghatározott kiosztási prioritást kapnak. A kezelő megváltoztathatja a prioritást a prioritásutasítással.

Egy következő betöltési utasítás megváltoztathatja, vagy törölheti a dinamikusosztály-táblázat beadott adatait. Egy betöltési utasítás visszautasításra kerül, ha az eltávolítandó dinamikus osztálynak kijelölt dinamikus partíciói vannak.

A szervezés/kiosztás (dispatching).

Egy VSE partíció egy főfeladatból és ehhez tartozó alfeladatokból áll, ahol a főfeladatnak mindig a legalacsonyabb prioritása van a partíción belül.

A jelenlegi VSE rendszer (VSE/SP 4.1) kétszintes szervezési sémával rendelkezik:

1. Kiválasztjuk a futtatásra kész, legmagasabb prioritású partíciót a kiosztási sorból (dispatcher queue).
2. Kiválasztjuk és kiosztjuk a megfelelő legmagasabb prioritású, futtatásra kész alfeladatot.

A dinamikus partíció segítségével a dinamikus partíciók szervezésének egy új szintjét vezetjük be. Mindegyik dinamikus osztálynak megvan a saját kiosztási sora, amelynek bejegyzései vannak a megadott osztály összes kijelölt dinamikus partíciójához. Amint korábban említettük, a dinamikus osztályok szerepelnek a kiosztási sorban, amely így egy háromszintes szervezési sémához vezet:

1. A legmagasabb prioritású, futtatásra kész statikus partíciót vagy dinamikus osztályt kiválasztjuk a kiosztási sorból.
2. Ha egy dinamikus osztályt választunk ki, meghatározzuk az osztály futtatásra kész, legmagasabb prioritású dinamikus partícióját.
3. A megfelelő legmagasabb prioritású, futtatásra kész alfeladatot kiválasztjuk és hozzárendeljük (dispatch).

A kezelő vagy egy indítófolyamat megváltoztathatja a statikus partíciók és dinamikus osztályok prioritását. Annak érdekében, hogy elkerüljük azt, hogy egy CPU intenzív partíció/osztály megállíthassa az alacsonyabb prioritású partíciók/osztályok kiosztását, a VSE lehetővé teszi, hogy meghatározzunk egy időosztásos (kiegyensúlyozott) csoportot. A csoportnak (statikus partíció vagy dinamikus osztály) az a tagja, amely kiesik az időrésből, a csoporton belül a legalacsonyabb prioritást kapja. Ha egynél több dinamikus partíció kerül kijelölésre egy adott dinamikus osztályon belül, ugyanezt az eljárást használjuk.

A feladat végrehajtása:

Egy dinamikus osztályban a feladat kiválasztása a végrehajtásra akkor lehetséges, ha a dinamikus osztály engedélyezve van. Ez bekerül a betöltött dinamikusosztály-táblázatba, és megváltoztatható egy másik betöltési igényvel vagy utasítással.

A következő fejezetek leírják az eljárást, miután egy engedélyezett dinamikusosztály-táblázat számára egy feladatot megtaláltunk.

A dinamikus partíció kijelöléséhez és az indításhoz a VSE/POWER egy dinamikus partíció kijelölését igényli. A kijelölőprogram létrehoz egy partícióazonosítót, kijelöli és inicializálja a kontrollblokkokat, és egy

címterületet hoz létre. A dinamikus partíció bekerül a dinamikusosztály-kiosztási sorba. A partícióindító program a dinamikus partíciót kioszthatóvá teszi, tehát futtatásra készre teszi, és indítja az inicializálást. Ha ugyanazon dinamikus osztályon belül egy második dinamikus partíció is kijelölésre került, indul a dinamikus osztály időosztása.

A dinamikus partíció inicializálása alatt a címterület érvényesítve lesz, a feladatvezérlés (JCL – job control) betöltődik, és a dinamikus partíciót előkészítik. Néhány rendszerrutin végrehajtásra kerülhet az előkészítés alatt. A feladatvezérlés végrehajtja a profile-t. Ha az inicializálás nem fejeződik be, a partíció törölve lesz, inaktívvá válik, a helykijelölés megszűnik és a dinamikus osztály le lesz tiltva, hogy elkerüljék egy másik feladat végrehajtását. Az eredményes inicializálás után a feladat végrehajtásra kerül, tehát a VSE/POWER átadja a JCL-t és az adatokat a dinamikus partíciónak. A VSE/POWER az aktiválás megszüntetését kéri, ha a feladat befejeződik.

A dinamikus partíció aktiválásának a megszüntetéséhez az előkészítés alatt behívott rendszerrutinokat értesítik, hogy az aktiválás megszüntetését kéri. A rendszer felszabadítja az erőforrásokat, leállítja a dinamikus partíciót, és a partíciót kioszthatatlanná teszi. A VSE/POWER megszünteti a partíció kijelölését.

A dinamikus partíció kijelölésének a megszüntetéséhez a megszüntetőprogram felszabadítja a partícióazonosítót és a vezérlőinformáció részére kijelölt rendszerterületet (vezérlőblokkok). Eltávolítja a dinamikus partíciót a dinamikusosztály-kiosztási sorból. Ha csak egy dinamikus partíció marad a dinamikus osztályban, a dinamikus osztály időosztása vissza lesz állítva.

A 4–10. ábrák, illetve táblázatokkal kapcsolatban a dinamikus osztályok használatát és a dinamikus partíciót ismertetjük. Ezek az ábrák képernyőn feltüntetett menüket vagy táblázatokat tartalmaznak. A rendszer és a felhasználó között egy interaktív párbeszéd jön létre ezeknek a táblázatoknak a segítségével.

A bemutatott forgatókönyvet használjuk egy dinamikus osztály meghatározására, a rendszerkönyvtárban a dinamikus osztálytábla frissítésére, a dinamikus osztálytábla aktiválására a VSE/POWER PLOAD utasítás használatával, és arra, hogy végrehajtsunk egy VSE/POWER feladatot a meghatározott dinamikus osztályban.

A 4. ábrán az „erőforrás-meghatározás” táblázat látható. Ebben a táblázatban a 7 szám alatt „a dinamikus partíciók beállítása” választható a 7 számbillentyű megnyomásával.

A 4. ábra szerint végzett kiválasztás által az 5. ábrában bemutatott „dinamikus partíciók beállítása” táblázat jelenik meg. Egy új dinamikus osztály meghatározására és hozzáadására az 5. ábrán mutatott táblázatban az 1=ADD funkciót választjuk ki. Ez a 6. ábrán bemutatott táblázathoz vezet. A bal oldali oszlopban lényegében a dinamikus osztály jellemzői, és a jobb oldali oszlopban ezeknek a jellemzőknek a magyarázata látható. Egy dinamikus osztály meghatározására például az X karaktert szűrjük be a középső oszlopban. Ennek a dinamikus

osztálynak nyolc partíciója van, mindegyik 3 megabájt saját címerülettel és egy 1 megabájt programterülettel. A GETVIS dinamikuscímerület-terület 64 kilobájtra van állítva, az „enabled” jellemző egy értékre van állítva, ami azt jelenti, hogy az osztály engedélyezve van. A futtatandó „profile” az STDPF JCL eljárás. Az összes többi változó alapértékre van állítva. Ha mindezeket az értékeket beállítottuk, akkor ezek az „enter” billentyű megnyomásával bevihetők.

Ez újra a „dinamikus partíciók beállítása” táblázatot hozza elő, de most a 4. sorban a hozzáadott dinamikus X osztályt mutatja az összes jellemzőjével, amint azt a 7. ábra tartalmazza. Így, ha a 7. ábrát összehasonlítjuk az 5. ábrán bemutatott táblázattal, most négy dinamikus C, Y, Z és X osztály van. Az 5 funkcióbillentyűn feltüntetett „feldolgozásművelet” megnyomásával a kijelzett dinamikusosztály-táblázat beírásra kerül új másolatként a rendszerkönyvtárba. Így a rendszerben most meg van határozva és a dinamikus partíciók aktiválási billentyű megnyomása után rendelkezésre áll négy különböző osztály, maximum 28 dinamikus partícióval.

A 8. ábra egy táblázatot mutat a dinamikusosztály-tábla aktiválásához. A dinamikus X osztály a VSE/POWER PLOAD DYNCTAB utasítás kibocsátása után használható. Akkor a rendszer kijelzi az aktív dinamikus osztályokat. Amint a 8. ábra mutatja, a négy osztályban a partíciók egyike sem aktív ebben a pillanatban. Amint a 09-es sor jelzi, a dinamikusosztály-táblázat eredményesen be lett töltve.

Most már végre lehet hajtani egy feladatot. Ehhez egy VSE/POWER munkafolyamatot alkalmazunk dinamikus X osztállyal a VSE/POWER feladat utasítás osztályparaméterében. A 9. ábra mutatja a forgatókönyvet, hogy hogyan néz ki ez a feladat.

A 10. ábrán bemutatott táblázat ábrázolja, hogy hogyan néz ki ez a feladat a végrehajtás során. Ha a dinamikus X osztályon belül egy dinamikus partíció áll rendelkezésre, ez a feladat úgy kerül végrehajtásra, amint a bemutatott példánál az X1 dinamikus partícióban. E feladat befejezése után ez a dinamikus partíció fel lesz szabadítva vagy meg lesz szüntetve a kijelölése, és rendelkezésre áll egy másik feladathoz.

A találmánnyal összhangban lévő koncepció előnye, hogy a felhasználónak csak a dinamikus partícióhoz rendelkezésre álló rendszererőforrásokat kell meghatározni (például a virtuális tároló nagyságát, dinamikusosztály-táblázatot). A feladat végrehajtását, beleértve a partíció kijelölését, és a kijelölés megszüntetését a rendszer végzi, tehát a rendszer erőforrásai csak akkor vannak használatban, amikor az szükséges. Könnyű feladatokat a statikusból a dinamikus partícióba áthelyezni, mivel az összeférhetetlenségek (vezérlőblokkok, szolgáltatások stb.) a lehetséges legalacsonyabb szinten vannak. A jelen találmány további új partíciókat biztosít a VSE operációs rendszerhez. Így sokkal több programot lehet párhuzamosan végrehajtani. A koncepció igen rugalmas és átvihető más operációs rendszerekre. A dinamikus partíció koncepciója így nyitva áll további kiterjesztéshez.

SZABADALMI IGÉNYPONTOK

1. Eljárás számítógépes rendszer virtuális tárolójának előre meghatározott kapacitású címerületekre osztására a címerületekhez előre hozzárendelt, adott számú partícióban (F1–F8, FA, FB, BG) párhuzamos programfuttatáshoz, ahol a címerületek közös (shared) és saját (private) területeket tartalmaznak, *azzal jellemezve*, hogy feladatok (jobs) párhuzamos végrehajtásához egy vagy több dinamikus partíciót hozzárendelünk a címerületekhez (P1, P2,... N1, N2,...O1, O2,...), ahol a hozzárendelt dinamikus partíciók mérete a végrehajtandó feladattól függ, és ahol

mindegyik dinamikus partíciónak saját címerülete (P1, P2,...N1, N2,...O1, O2,...) van, és csak egy dinamikus partíció van a címerület (P1, P2,...N1, N2,...O1, O2,...) saját területéhez hozzárendelve,

elindítjuk és vezéreljük a feladatok végrehajtását, és megszüntetjük a dinamikus partíció hozzárendelését, amikor a hozzárendelt feladat végrehajtása befejeződött, újabb felhasználáshoz felszabadítjuk a dinamikus partíció által lefoglalt címerületet (P1, P2,...N1, N2,...O1, O2,...), továbbá

ahol a dinamikus partíciókkal együtt átmenetileg dinamikus címerületeket (GETVIS) jelölünk ki a saját (private) rendszerterületen, amely rendszer adatokat tartalmaz a címerülethez az adott feladat számára.

2. Az 1. igénypont szerinti eljárás, *azzal jellemezve*, hogy több dinamikus partíciót jelölünk ki egyszerre.

3. Az 1. vagy 2. igénypont szerinti eljárás, *azzal jellemezve*, hogy a dinamikus partíciókat a végrehajtható feladatok dinamikus osztályaiba csoportosítjuk, miközben dinamikusosztály-táblázatot készítünk, amely tartalmazza a rendszer saját dinamikuscímerület- (GETVIS) jellemzőit (tárkiosztás, partícióméret, saját rendszerterület-méret).

4. A 3. igénypont szerinti eljárás, *azzal jellemezve*, hogy a dinamikus osztályok mindegyikéhez végrehajtási prioritást rendelünk hozzá, és dinamikus partíciót csak akkor jelölünk ki, ha a feladatot dinamikus osztályban kell végrehajtani.

5. A 3. vagy 4. igénypont szerinti eljárás, *azzal jellemezve*, hogy a dinamikus osztály előre kijelölt statikus és dinamikus partíciókkal rendelkezik, ahol a statikus partíciókhoz rendelt feladatoknak elsőbbsége van a dinamikus partíciókhoz rendelt feladatokhoz képest.

6. A 3–5. igénypontok bármelyike szerinti eljárás, *azzal jellemezve*, hogy a dinamikus osztályok dinamikus partícióit időben felosztjuk.

7. A 3–6. igénypontok bármelyike szerinti eljárás, *azzal jellemezve*, hogy az osztálytáblázatot könyvtárban tároljuk el, ahonnan az kiolvasható a dinamikus partíciók kiosztásához, a feladatok végrehajtásához, valamint a táblázat frissítéséhez és karbantartásához.

8. A 3–7. igénypontok bármelyike szerinti eljárás, *azzal jellemezve*, hogy az egyes dinamikus osztályokhoz diszpécserlistát rendelünk, amely a dinamikus osztály minden egyes kijelölt partíciójához tartalmaz bejegyzéseket, a diszpécserlistáról kiválasztjuk a legmagasabb prioritású statikus partíciót vagy dinamikus osz-

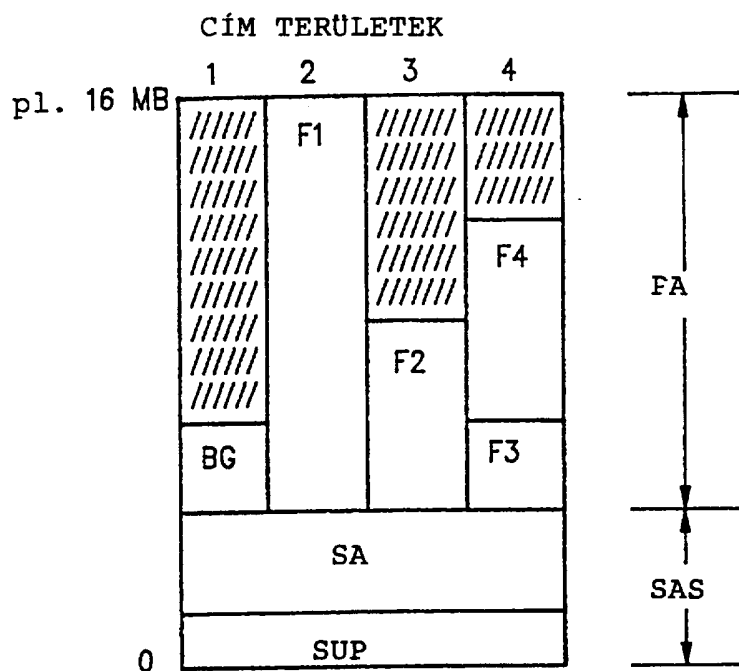
tályt a futtatásra, ahol, ha egy dinamikus osztály van kiválasztva, akkor azon belül meghatározzuk a legmagasabb prioritású, futásra kész partíciót, és a megfelelő, legmagasabb prioritású feladatot választjuk ki és indítjuk el.

9. A 3–8. igénypontok bármelyike szerinti eljárás, *azzal jellemezve*, hogy a következő műveleteket hajtjuk végre:

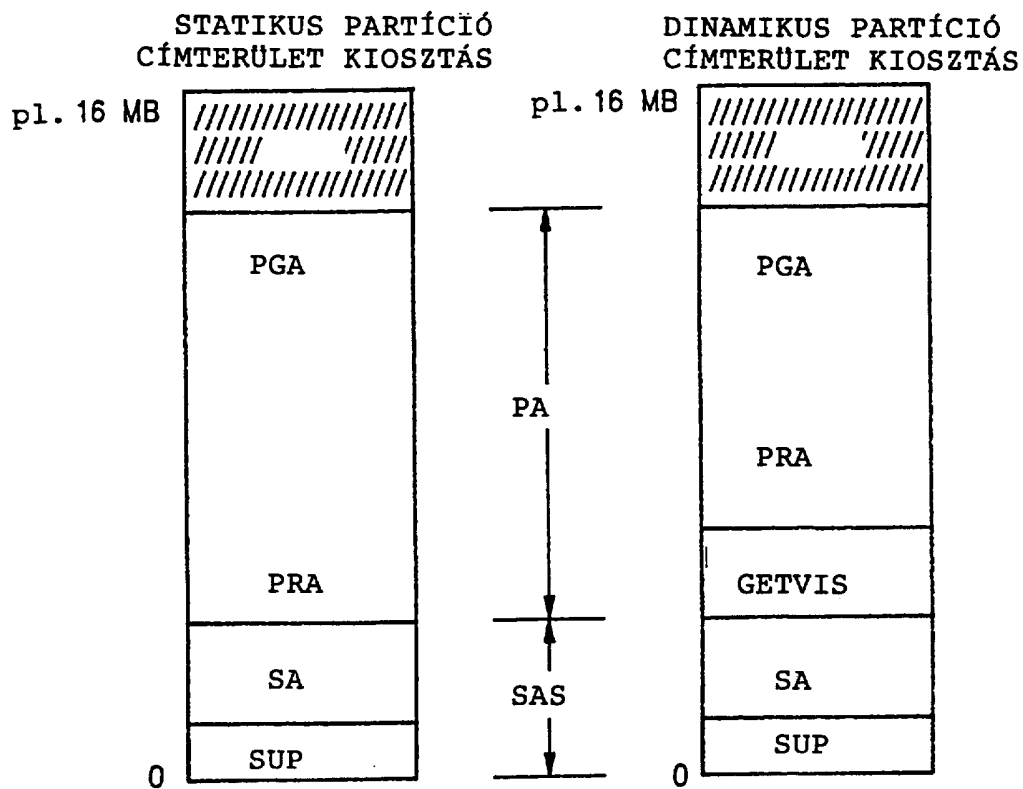
- dinamikuspartíció-kijelölést kérünk, ha engedélyezett dinamikusosztály-táblázathoz elvégzendő feladatot találtunk;
- aktiválunk egy kijelölési szervizprogramot, amellyel partícióazonosítót hozunk létre, kontrollblokkokat jelölünk ki és inicializálunk, valamint létrehozunk egy címterületet;
- aktiválunk egy partíciót indító szervizrutint, amellyel a dinamikus partíciót felkészítjük a futásra;
- a dinamikuspartíció-inicializálás alatt érvényesítjük a címterületet, betöltjük a feladat-vezérlőutasításokat (job control), és előkészítjük a dinamikus partíciót;
- sikeres inicializálás után végrehajtjuk a feladatot;
- dinamikus partíció esetén megszüntetjük a partíciókijelölést, felszabadítjuk a partícióazonosítót és a vezérlőinformáció számára kijelölt rendszerterületet, és ezzel töröljük a dinamikus partíciót a dinamikusosztály-diszpécserlistáról.

10. A 3–9. igénypontok bármelyike szerinti eljárás, *azzal jellemezve*, hogy a dinamikusosztály-táblázat a következő jellemzőket tartalmazza:

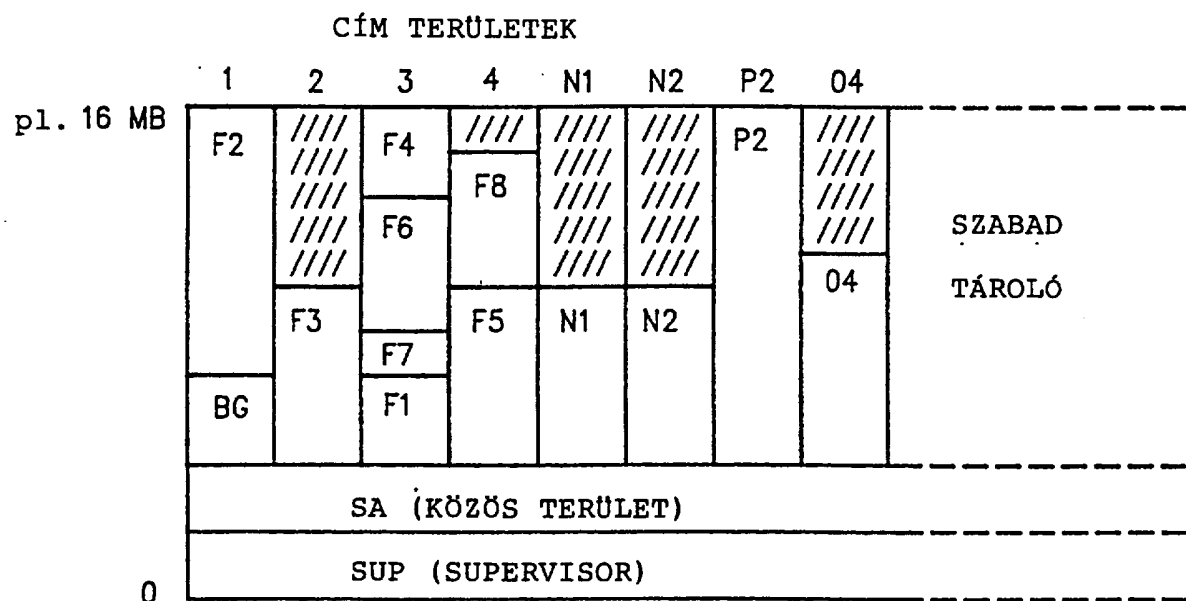
- a) tárolókiosztás (storage allocation),
5 amellyel egy dinamikus partíció, különösen a dinamikus címterület (GETVIS) partíció kijelölésének tárolási követelményeit határozzuk meg egy adott osztályban;
- b) partícióméret,
10 amellyel a program végrehajtására fenntartott partícióban az összefüggő virtuális tároló méretét határozzuk meg;
- c) dinamikus címterület (GETVIS) mérete,
15 amellyel egy dinamikus címterület összefüggő virtuális tároló méretét határozzuk meg;
- d) profile vagy procedúra,
amelyet a feladat előtt hajtunk végre a partícióban;
- e) dinamikus partíciók maximális száma egy osztályban egy osztályon belül párhuzamosan kijelölhető
20 partíciók maximális száma, amennyiben elegendő virtuális tároló és dinamikus partíció áll rendelkezésre;
- f) letiltásjelzés (disable indication),
amellyel lehetővé tesszük egy dinamikus osztály letiltását;
- g) sorba állított I/O eszközök,
25 amellyel megadjuk a sorba kapcsolt eszközöket, például olvasó, nyomtató, lyukasztó, amelyek I/O sorba állítóhoz kapcsolódnak.



1. ábra



2. ábra



3. ábra

4. ábra

TAS\$DYN1

Válasszon opció(ka)t és nyomja meg az ENTER billentyűt

Opciók: 1 = HOZZÁAD 2 = VÁLTOZTAT 5 = TÖRÖL

OPC.	Din.	Eng.	Part.max.	Tárolóhely	Max.prog.	GETVIS	Profile
	Oszt.	1=igen 2=nem	száma	kijelölés	méret	din.ter.	
1	C	1	9	1M	500K	64K	STDPFOP
-	Y	1	8	2M	1024K	64K	STDPFOP
-	Z	1	3	5M	1024K	64K	STDPFOP

PF1=HELP 2=FRISSÍTÉS 3=VÉGE 5=FELDOLGOZÁS

5. ábra

TAS\$DYN2

DINAMIKUS PARTÍCIÓK BEÁLLÍTÁSA

Adja meg a kért adatot és nyomja meg az ENTER billentyűt

Dinamikus osztály	X	Adjon meg egy osztály C-E, G-Z közül
Partíciók száma	8	Adjon meg egy számot 1 - 32 között
Tárolóhely allokáció	3M	Adja meg Mbyt-ban
Maximális program méret	1M	Adja meg Kbyte-ban vagy Mbyt-ban
GETVIS dinamikus címtérület	64K	Adja meg Kbyt-ban
Engedélyezve	1	1=igen, 2=nem
Profile	STDPF	A JCL eljárás neve
Logikai egységek száma	50	Adjon meg egy számot 10 - 255 között
Olvasó eszköz	FEC	---
Nyomtató eszközök listája	FEE	---
	---	---
Lyukasztó eszközök listája	FED	---
	---	---
PF1=HELP	2=FRISSÍTÉS	3=VÉGE

6. ábra

TAS\$DYN1 DINAMIKUS PARTÍCIÓK BEÁLLÍTÁSA

Válasszon opció(ka)t és nyomja meg az ENTER billentyűt

Opciók: 1 = HOZZÁAD 2 = VÁLTOZTAT 5 = TÖRÖL

OPC.	Din. Oszt.	Eng. 1=igen 2=nem	Part.max. száma	Tárolóhely kijelölés	Max.prog. méret	GETVIS din.ter.	Profile
1	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=FRISSÍTÉS 3=VÉGE 5=FELDOLGOZÁS

7. ábra

```

01 FLOAD 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 LUBS
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

```

8. ábra

```

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

```

9. ábra

```

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 STDPFROF 50
05 F1 001 1Q6AI Y ENAB 0 8 2M 1024K 64K STDPFROF 50
06 F1 001 1Q6AI Z ENAB 0 3 5M 1024K 64K STDPFROF 50
07 F1 001 1Q6AI X ENAB 0 8 3M 1024K 64K STDPFROF 50
08 F1 001 1Q6EI 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

```

10. ábra