

Cap. 8.3.

Memoria Virtuala.

Memoria virtuala reprezinta un ansamblu de tehnici si mecanisme folosite pentru a extinde capacitatea aparenta a memoriei principale RAM. Astfel, numarul de biti din instructiune, care specifica adresa, depaseste capacitatea memoriei principale fizic instalata, ca posibilitati de adresare. In aceste conditii, din programul stocat in memoria externa vor fi aduse in memoria principala numai zonele active la un moment dat. Procesul este transparent pentru utilizator si se desfasoara automat. Spatiul de adresare explorat de bitii de adresa din instructiunile furnizate de procesor corespunde capacitatii memoriei virtuale. Sub aspect fizic memoria virtuala va fi organizata in memoria secundara.

Memoria virtuala este structurata, din punct de vedere logic, sub forma de pagini, astfel ca, o adresa furnizata de o instructiune a procesorului consta in doua campuri: numarul paginii si adresa in pagina:

Nr. Pagina Virtuala	Adresa in Pagina
---------------------	------------------

Dimensiunile unei pagini sunt mult mai mici decat dimensiunea memoriei principale M_p . In acest mod, in memoria principala pot fi aduse, din memoria secundara, mai multe pagini active ale unui program sau ale mai multor programe, in cazul exploatarii sistemului in regim de multiprogramare.

Se va considera, ca exemplu educational, organizarea unei memorii virtuale cu capacitatea de 64 Mo, structurata pe cuvinte de 32 de biti: 16 Mcuvinte x 32 de biti/cuvant.

Intr-un cuvant/instructiune, de 32 de biti, 24 de biti sunt alocati pentru adresarea memoriei virtuale. Primii 12 biti, mai semnificativi, vor specifica numarul paginii, iar ultimii 12 biti, mai putin semnificativi, vor constitui adresa in pagina. In figura 10 se prezinta organizarea logica a memoriei virtuale cu 2^{12} pagini, fiecare pagina avand capacitatea de 2^{12} cuvinte, de cate 32 de biti. Organizarea fizica a memoriei virtuale in cadrul memoriei secundare este prezentata in figura 11 de mai jos:

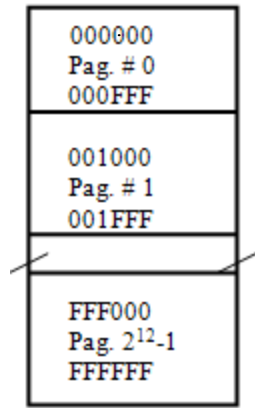


Fig.10 Organizarea
logica

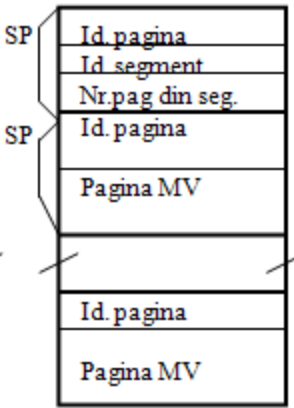
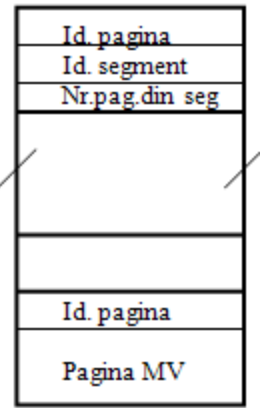


Fig.11.Structura segmentelor. Organizare
fizica (SP = Structura Paginii in Memoria Secundara)



Programul utilizatorului este constituit dintr-un numar oarecare de pagini, organizate intr-un segment. Prima pagina, din cadrul fiecarui segment, contine, pe langa un identificator de pagina, un identificator de segment si numarul de pagini din segmentul dat. Celelalte pagini vor fi constituite dintr-un antet de identificare/identificator de pagina si pagina propriu-zisa.

In cazul de fata se considera ca sistemul de operare se ocupa cu gestiunea segmentelor/fisierelor, in memoria secundara.

Memoria principala va avea o capacitate limitata astfel incat nu toate paginile unui segment pot fi simultan in RAM. Considerand o memorie principala cu capacitatea de 256 k.cuvinte x 32 de biti, se pot organiza in cadrul ei $2^6 = 64$ pagini avand fiecare o capacitate de $2^{12} = 4096$ cuvinte.

Nucleul sistemului de operare va fi rezident in memoria principala si va ocupa primele 8 pagini (32 k cuvinte). Sistemul de operare va gestiona o serie de tabele privind locatiile paginilor in memoria secundara. Primele doua cuvinte din RAM vor fi utilizate de hardware-ul de control al memoriei virtuale pentru a informa sistemul de operare asupra numerelor paginilor virtuale, care trebuie interschimbate intre memoria secundara si memoria principala. Locatia cu adresa 0 va contine numarul paginii care trebuie adusa din memoria secundara. Locatia 1 va contine numarul paginii care va fi transferata din memoria principala in memoria secundara. Numarul acesteia din urma se stabileste ca si in cazul memoriei tampon, folosind un tablou de contori de utilizare CU si o unitate logica combinationala PRILIB(CU). Modalitatea de translatore a adreselor virtuale in adrese fizice este prezentata

in cadrul structurii memoriei principale, care suporta mecanismul de memorie virtuala (Fig. 12.).

Sunt utilizate urmatoarele resurse hardware:

- TA[64;12] - Tablou Asociativ in celulele caruia sunt stocate numerele paginilor virtuale aduse in memoria principala;

- CU[64;6] - Contori de Utilizare ai paginilor prezente in memoria principala;

ROMA[64;6] - Memorie cu continut permanent, adresata asociativ, care transforma numarul paginii virtuale in numar de pagina fizica din memoria principala RAM

- RI[8;19]/CP[0;11] - numarul paginii virtuale;

- RI[20;31]/CP1[2;23] - adresa in pagina;

- AM[0;5] - numarul paginii fizice.

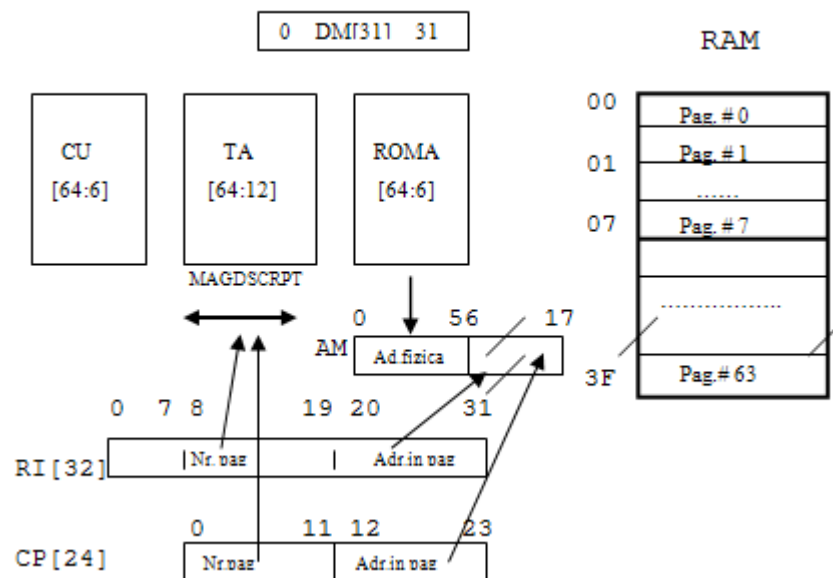


Fig. 12. Organizarea memoriei principale, cu mecanismul asociativ care asigura suportul pentru translatarea adreselor virtuale in adrese fizice.

Resurse hardware:

- AM[18] - Registrul de Adrese al Memoriei RAM;

- DM[32] - Registrul de Date al Memoriei principale RAM;

- RI[32] - Registrul Instructiunii;

- CP[24] - Contorul Programului;

- M[2¹⁸;32] - Memoria principala RAM;

- MAGDSCRPT[12] - MAGistrala DeSCRiPTorilor;
- ASOC(TA;MAGDSCRPT) - Unitate logica combinationala ASOCiativa cu 2^6 iesiri binare;
- DCZ(CU) - Unitate logica combinationala care asigura DeCrementarea pana la Zero a Contorilor de Utilizare;
- PRILIB(CU) - Unitate logica combinationala care sorteaza PRIMul Contor de Utilizare LIBer cu continutul cel mai mic, corespunzind paginii cel mai putin recent utilizate.

Daca o pagina oarecare din memoria virtuala a fost adusa in memoria fizica/principala, ea poate fi localizata, ca numar de pagina fizica, prin referirea la o memorie cu continut permanent si cu adresare asociativa ROMA. In cazul coincidentei intre numarul paginii virtuale, furnizate prin MAGDSCRPT, si continutul unei celule din Tabloul Asociativ, celula corespunzatoare, situata pe aceeasi linie, in ROMA, va furniza primii 6 biti mai semnificativi ai adresei fizice. Acestia vor fi concatenati in registrul AM, cu ultimii 12 biti din RI sau CP, pentru a forma adresa fizica, de acces la memoria principala. Contorul asociat, din CU, va fi fortat la valoarea $6\text{T}55$, celalti contori, cu exceptia primilor opt, vor fi decrementati cu ajutorul functiei logice combinationala DCZ(UC).

Primele opt celule din ROMA au continuturi fixe, corespunzand numerelor primelor opt pagini din memoria principala in care se afla nucleul sistemului de operare. Primele opt celule din TA vor contine numerele paginilor virtuale corespunzatoare nucleului sistemului de operare. Continutul acestora va fi fix.

De asemenea, continuturile primilor opt contori de utilizare nu vor suferi decrementari.

Gestiunea memoriei virtuale (Fig.13) incepe prin transferul unui segment sau al unui numar de pagini, dintr-un segment, corespunzator unui program (Blocul 1, Fig.13.), din memoria secundara in memoria principala, sub controlul sistemului de operare, si lansarea in executie (Blocul 2.). Daca se constata lipsa unei pagini, se trece la blocul 3, care este implementat in hardware si care va plasa in prima celula din memoria principala, numarul paginii virtuale, care trebuie adusa din memoria secundara, iar in celula a doua-numarul paginii, care trebuie transferata, din memoria principala in memoria secundara. Dupa terminarea acestor operatii se va genera o intrerupere catre sistemul de operare. Acesta, in cadrul blocului 4. va face interschimbul de pagini.

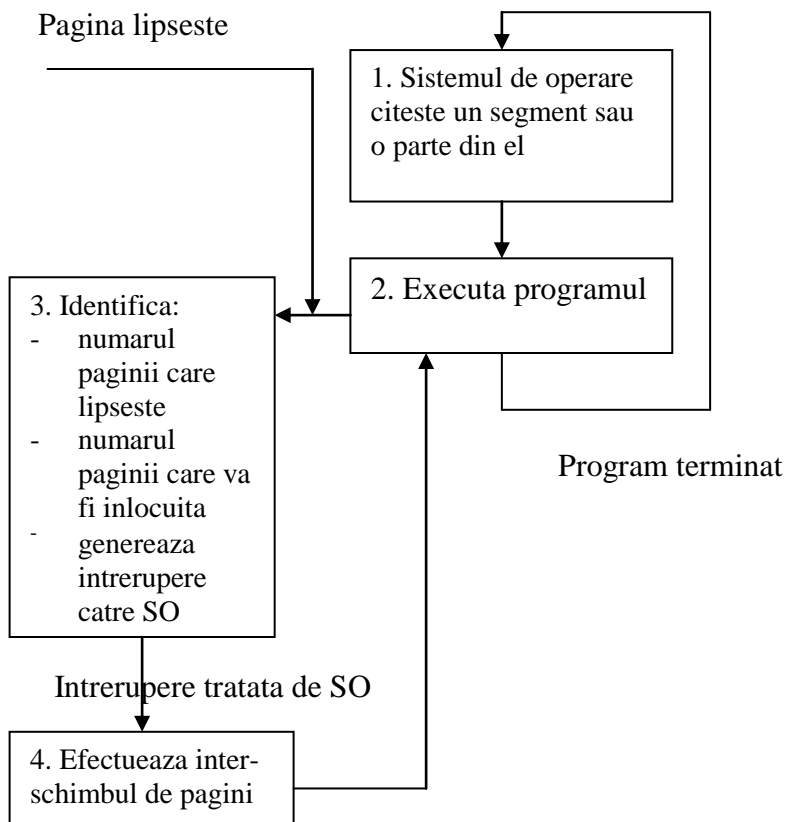


Fig. 13 Gestiunea memoriei virtuale.

Operatiile din blocurile 1 si 4 se executa prin software in timp ce operatiile din blocurile 2 si 3 se executa prin hardware.

Operatiile din cadrul blocului 2 sunt operatii obisnuite de citire/interpretare si executie a instructiunilor programului, cu mentiunea ca activitatile de citire/scriere din/in memorie presupun noi secvente de comanda, dupa cum se va vedea mai jos.

Operatiile din cadrul blocului 3 necesita introducerea unei noi secvente de operatii la nivelul unitatii de comanda a procesorului. Secventa va fi activata ca urmare a constatarii lipsei paginii adresate din memoria principala. Secventa va fi activata cu linia 100, in microprogramul de comanda al procesorului.

Pentru fiecare citire din memoria virtuala procesorul trebuie sa execute urmatoarea secventa de comenzi:

C1. MAGDSCRPT = "nr. pagina"; /* RI[8:19] sau CP[0:11] */

AM ← BUSFN(ROMA; ASOC(TA; MAGDSCRPT)), "adresa in pagina";

/*in AM se plaseaza adresa pentru accesul la memoria fizica RAM*/

$$\overline{DM^*(\cup / ASOC(TA; MAGDSCRPT))} \leftarrow 20_T 0, "nr. pagina";$$

/*in cazul cand pagina cautata lipseste se stocheaza in registrul DM numarul paginii care lipseste*/

$$\rightarrow (\cup / ASOC(TA; MAGDSCRPT)) / (100)$$

/*daca pagina solicitata lipseste se trece la pasul 100 in secventa de comanda*/

C2. CU*(ASOC(TA; MAGDSCRPT)) ← 6_T 55;

/*contorul asociat celulei selectate este fortata la valoarea 6_T 55 deoarece in memoria principala pot fi aduse numai 55 de pagini ale programelor din memoria secundara; 8 pagini sunt ocupate de sistemul de operare*/

$$\overline{CU^*(ASOC(TA; MAGDSCRPT))} \leftarrow DCZ(UC);$$

/*ceilalti contori sint decrementati*/

DM ← BUSFN(M; DCD(AM))

/*se citeste data din memoria principala M*/

C3. pasul urmator.

Pentru scrierea in memoria procesorul va executa urmatoarea secventa de comenzi care, cu exceptia ultimei linii, este asemanatoare cu secventa de citire:

S1. MAGDSCRPR = "nr. pagina";

AM ← BUSFN(ROMA; ASOC(TA; MAGDSCRPT)), "adresa in pagina";

$$\overline{DM^*(\cup / (ASOC(TA; MAGDSCRPT))} \leftarrow 20_T 0, "nr. pagina";$$

$$\rightarrow (\cup / (ASOC(TA; MAGDSCRPT)) / (100)$$

S2. CU*(ASOC(TA; MAGDSCRPT)) ← 6_T 55;

$CU^*(\overline{ASOC(TA;MAGDSCRPT)}) \leftarrow DCZ(UC);$

$M^*DCD(AM) \leftarrow DM$

S3. pasul urmator.

Registrul DM este afectat la C2 si S2 numai in cazul lipsei unei pagini. Executia instructiunii va fi intrerupta/suspendata pana

la aducerea paginii solicitate in memoria principala. Astfel, nu apare nici un conflict in utilizarea registrului DM.

In vederea pregatirii interventiei sistemului de operare pentru efectuarea transferului necesar de pagini, intre memoria principala si memoria secundara, trebuie sa se plaseze in locatiile cu adresele 00000H si 00001H informatiile referitoare la:

- numarul paginii care lipseste ($DM[20:31]$);
- numarul paginii care trebuie inlocuita in memoria principala:
($BUSFN(TA;PRILIB(CU))$).

Secventa care se lanseaza de la linia 100 va asigura aceste date, va incrementa/decrementa contorii de utilizare si va genera in final o intrerupere.

100. $AM \leftarrow 18T0$

101. $M^*DCD(AM) \leftarrow DM$

102. $AM \leftarrow INC(AM);$

$DM[20:31] \leftarrow BUSFN(TA;PRILIB(CU));$

$TA^*PRILIB(CU) \leftarrow DM[20:31]$

103. $M^*DCD(AM) \leftarrow DM$

104. $CU^*PRILIB(CU) \leftarrow 6T55;$

$CU^*(\overline{PRILIB(CU)}) \leftarrow DCZ(CU);$

cerere de intrerupere = 1

Dupa generarea intreruperii intreaga responsabilitate pentru transferul paginilor revine sistemului de operare.