

## **Tema SO**

### **Implementarea gestiunii memoriei**

**Marica Iulia**

**Grupa 432A**

**Facultatea de Electronica Telecomunicatii si Tehnologia Informatiei**

## **Cuprins**

- 1.Paginarea in linux**
  - 1.1. Cererea de paginare**
- 2.Paginarea in Windows**
  - 2.1.Cum se muta fisierele de paginare**
  - 2.2.Algoritmul de inlocuire a paginii**
    - 2.2.1.Paginarea anticipata**
    - 2.2.2.Free page queue**
    - 2.2.3.Procesul page sealing**
    - 2.2.4.Pre-Cleaning**
- 3.Comparatie Linux-Window**
- 4.Concluzii**
- 5.Bibliografie**

## 1. **Paginarea in Linux.**

In acest caz ne referim la modul in care un sistem de operare utilizeaza capacitatile procesorului astfel incat sa para ca exista mai multa memorie alocata decat exista in realitate. Aceasta poate fi definitia memoriei virtuale (MV). Memoria virtuala in linux este foarte complexa.

Un process trebuie sa contina doar acele pagini necesare in memorie cu care sa lucreze. De aceea se utilizeaza CPU si Kernel care se asigura ca un process nu are acces la locatiile de memorie ale unui alt process. Procesele aderă in general la principiul localitatii spatiale (spatial locality). Kernel-ul poate stabili un set de lucru de pagini pentru fiecare proces, paginile care au fost accesate ultima data sunt pastrate in "n" memorii de referinta.

### 1.1. **Cererea de paginare.**

O data ce o imagine executabila (de memorie mapata) a fost convertita intr-un process virtual poate incepe executia propriu-zisa. Daca doar inceputul unei imagini este fizic, convertit in memoria virtuala el va accesa in curand o zona de memorie virtuala care nu se afla inca in memoria fizica. Cand un proces acceseaza o adresa virtuala, care nu are o adresa valida de intrare in tabelul de pagini, procesorul va raporta o eroare de pagina Linux.

Pagina gresita poate fi descrisa de adresa virtuala in cazul in care paginarea de eroare a avut loc si de tipul de acces la memorie. Linux trebuie sa gaseasca zona de memorie in care este pagina gresita. Acest lucru se face prin intermediul **vm\_area\_struct**, structura de Kernel-uri de date. Cautarea prin intermediul structurilor de date este esentiala pentru eficientizarea defectelor de pagina. Aceste structuri sunt legate printr-o structura de tip **arbore AVL** (Adelson-Velskii si Landis). O structura de tip AVL este un arbore binar echilibrat de cautare in cazul in care inaltimea

celor doi subarbori ai unui nod difera de la cea mai mare parte,optimizand astfel cautarile.

Daca nu exista nicio structura de date `vm_area_struct` pentru aceasta adresa virtuala gresita, acest proces a accesat ilegal o adresa virtuala. Linux va semnala procesul,trimiterea unui **semnal SIGSEGV** si daca procesul nu are un **handler**,acest semnal va fii reziliat.

Linux verifica tipul de pagina gresita care a avut loc impotriva tipurilor de permise de acces pentru aceasta zona de memorie virtuala. In cazul in care procesul este de accesare de memorie in mod ilegal,spunem ca scrierea intr-o zona destinata numai cititului este de asemenea semnalata ca o eroare de memorie.

Daca Linux a stabilit ca pagina gresita este legala, trebuie sa se ocupe de ea.

Linux trebuie sa diferentieze paginile care se afla in fisierul swap si cele care sunt parte ale unei imagini executabile pe un disc. SE face acest lucru prin utilizarea intrarii in tabelul de pagini pentru aceasta adresa virtuala gresita.

Daca la intrarea paginii, tabelul de pagina nu este valabil,dar nici gol,pagina gresita este o pagina din fisierul de swap.

Pentru Alpha, intrarile AXP in tabelul de pagina sunt inregistrate ca date care nu au setul lor de biti variabil,dar care au o valoare non-zero, in domeniul lor de **PFN**. In acest caz, domeniul PFN detine informatii cu privire la cazul in care pagina este executata.

## 2. **Paginarea in Windows.**

Fisierul de paginare reprezinta zona de pe hard disk pe care Windows o utilizeaza ca si cum ar fi memoria cu acces aleator(**RAM**). Acest lucru este uneori cunoscut sub numele de **Memorie Virtuala(MV)**. In mod implicit Windows stocheaza acest fisier in aceeași partitie cu fisierele de sistem Windows. Putem crește performanța de Windows și a spațiului liber de partitie de încărcare, prin mutarea acestui fisier pe o altă partitie.

### 2.1. **Cum se muta fisierul de paginare.**

Conectați-vă la computer ca Administrator.

Faceți Click pe Start, apoi faceți clic pe Control Panel.

Faceți clic pe Performance and Maintenance și apoi faceți clic pe System.

Faceți clic pe fila Complex, apoi, sub Performanță, faceți clic pe Setări.

Faceți clic pe fila Complex, apoi, sub Memorie virtuală, faceți clic pe modificare. În Unitate [Etichetă volum] listă, faceți clic pe o altă unitate decât cea pe care este instalat Windows (Windows este instalat de obicei pe unitatea C). Sub dimensiune totală fișier de paginare pentru toate unitățile, notați valoarea afișată lângă Recomandat. Faceți clic pe Dimensiune particularizată, apoi tastați valoarea recomandată în dimensiunea inițială (MO). Tastați dimensiunea maximă pe care doriți pentru a permite paginare la dimensiunea maximă (MO), apoi faceți clic pe Set.

În Unitate [Etichetă volum], faceți clic pe unitatea pe care este instalat Windows (de obicei, unitatea C), apoi utilizați una dintre următoarele măsuri: Dacă nu doriți un fișier de paginare pe unitate, faceți clic pe Fără fișier de paginare, apoi faceți clic pe Set. Un mesaj similar apare următorul mesaj: Dacă fișierul de paginare de pe volumul C: are o dimensiune inițială de mai puțin de 126 de megabytes, atunci sistemul nu poate fi capabil de a crea un fișier cu informații de depanare, dacă apare o eroare STOP.

## **2.2. Algoritmul de înlocuire a paginii**

### **2.2.1 Paginarea anticipată**

Această tehnică, numită uneori “**swap prefetch**” este un proces de **non-resident** al paginilor care sunt susceptibile de a fi menționate în viitorul apropiat (profitând de localitate de referință). Astfel de strategii încearcă să reducă numărul de erori de pagini pe experiențe pe proces. Unele dintre aceste strategii sunt, în cazul în care un program de referință e o adresă virtuală care provoacă un defect pe pagină, probabil o valoare de câteva pagini de spațiu de adrese virtuale vor fi curând folosite, și în cazul în care un program care tocmai a terminat execuția lasând o multitudine de memorie RAM liberă, probabil utilizatorul va reveni la folosirea de programe care au fost recent paginate.

### **2.2.2 Free page queue**

Este o listă de pagini de cadru, care sunt disponibile pentru alocare, după o defecțiune paginată. Unele sisteme de operare necesită recuperarea în cazul în care o eroare de start apare pentru o pagină care a fost “furată” și cadrul paginii nu a fost niciodată mutat, apoi sistemul de operare evită necesitatea de a citi pagini din spate prin atribuirea în cadrul paginii modificate.

### 2.2.3 Procesul Page stealing

Procesul *page stealer* este un proces al nucleului care evacueaza din memorie paginile care nu fac parte mult timp din setul de lucru al unui proces. Nucleul creeaza procesul *page stealer* pe timpul initializarii sistemului si-l apeleaza pe toata durata de viata a sistemului, la nevoie. El examineaza fiecare regiune activa, care nu este blocata, lasând regiunile blocate in expectativa pentru a le examina la urmatoarea parcurgere a listei de regiuni, si incrementeaza apoi câmpul *vârsta* al tuturor paginilor valide.

Nucleul blocheaza o regiune când un proces produce o întrerupere pentru o pagina din regiune, astfel încât procesul *page stealer* nu poate evacua pagina care a generat întreruperea.

Pentru o pagina din memorie exista doua stari: pagina a fost încarcata recent si nu este inca eligibila pentru *swapping*, sau pagina este eligibila pentru *swapping* si este disponibila pentru reasigurarea la alte pagini virtuale. Prima stare arata ca un proces a referit recent pagina, ea fiind deci in setul sau de lucru. Unele masini seteaza bitul *referinta* când acceseaza o pagina, inasa metodele software pot suplini lipsa acestei facilitati daca hardware-ul nu dispune de ea. Procesul *page stealer* pune pe 0 bitul *referinta* al paginilor referite si retine câte examinari au fost facute de la ultima referire a paginii. Deci prima stare consta in câteva substari corespunzatoare numarului de examinari pe care le face procesul *page stealer* pâna când pagina devine eligibila pentru evacuare (Figura 12. 7). Când acest numar depaseste o valoare de prag, nucleul trece pagina in cea de-a doua stare, gata pentru evacuare. Perioada maxima care poate trece pâna când pagina devine eligibila pentru evacuare este dependenta

de implementare, și este limitată de numărul de biți disponibili în intrarea din tabela de pagini.

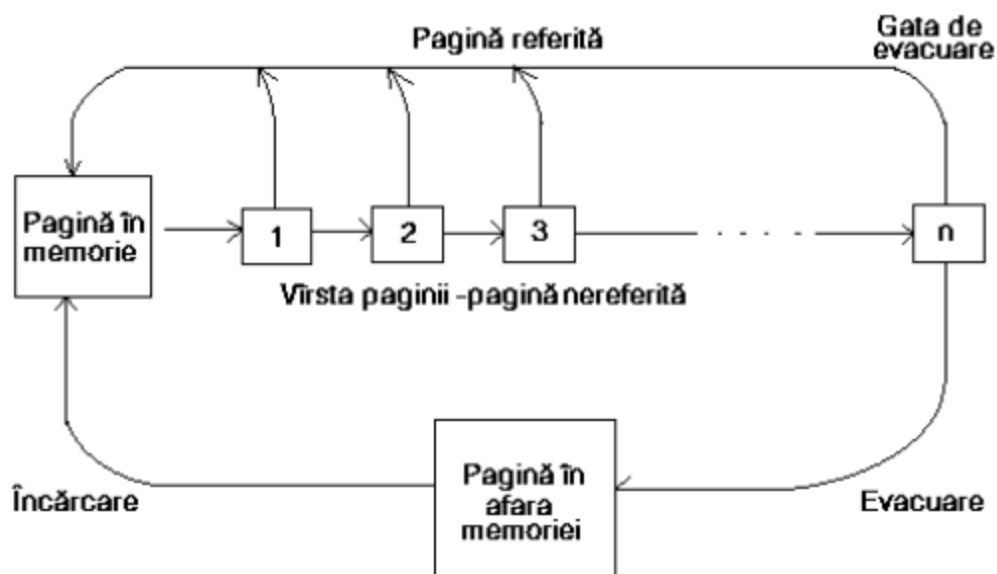


Figura 12.7 Diagrama stărilor în funcție de timpul de rezidență al paginilor

Figura preluată de pe

<http://andrei.clubcisco.ro/cursuri/3so/labs/04. Gestiunea memoriei. pdf>

## 2.2.4 Prea-Cleaning-ul

Sistemele de operare UNIX folosesc periodic sincronizarea la pre-cleaning-ul pentru paginile murdare, pentru a salva toate paginile pe hard-disk. Sistemele de operare Windows fac același lucru prin firele “**modified page writer**”. Pre-cleaning-ul face ca deschiderea unui fișier de date sau începerea unui program să se realizeze mult mai rapid. Hard-disk-ul poate cauta imediat acel fișier și să citească consecutiv întregul dosar în pre-



cleaning de cadre de pagina. Fara pre-cleaning,hard-disk-ul este fortat sa caute inapoi intre scriere si cadrul paginii murdare pe disc si citeste pagina urmatoare a dosarului in cadre.

### 3. Comparatie Linux-Windows

Din punct de vedere software, Linux 2. 6 cu Kernel,fișierele de swap sunt la fel de rapide ca partițiile de swap. Nucleul pastreaza o harta in care exista fișierul de swap si acceseaza discul direct, ocolind sistemul de fișiere cache si aeriene. **Red Hat** recomanda utilizarea unei partiții de swap.

Flexibilitatea fișierelor de swap poate depasi si alte avantaje ale partițiilor de swap. De exemplu un fișier de swap poate fi plasat pe orice unitate, poate fii setat la orice dimensiune dorita si poate fii adaugat sau modificat daca este necesar. O partiție de swap cere ca acesta sa fie stabilit pentru intregul hard-disk si in cele mai multe hard-disk-uri exista un maxim de 4 partiții, iar o data ce marimea partiției de swap este setata nu poate fii schimbata fara utilizarea instrumentelor pentru a redimensiona intreaga unitate. De asemenea in cazul in care computerul este destinat special pentru a rula mai mult de un sistem de operare( un proces numit **multi-boot**)se reduce numarul de sisteme care pot fii utilizate pe aceasta masina.

Linux utilizeaza practic un numar nelimitat de dispozitive de pompare, fiecaruia fiindui atribuita o prioritate. Atunci cand sistemul de operare are nevoie pentru a schimba pagini din memoria fizica, se foloseste dispozitivul cu cea mai mare prioritate,cu spatiu liber. In cazul in care mai multe dispozitive au atribuite aceeasi prioritate, acestea sunt utilizate intr-un mod similar ca acordurile la nivel de **RAID 0**.

Acest lucru ofera o performanta mai buna,atata timp cat dispozitivele pot fii accesate eficient in paralel. Prin urmare,trebuie sa fie considerate atribuirile prioritatilor. De exemplu,Swap-urile situate pe acelasi disc fizic nu ar trebui sa fie utilizate in paralel,dar in ordine de la cel mai rapid la cel mai lent( de exemplu: cel mai rapid avand cea mai mare prioritate).

In NT bazate pe versiunile de Windows(Windows XP,Vista si Windows 7) fisierul utilizat pentru paginare este numit **pagefile. sys**. Locatia implicita a fisierului pagina este in directorul radacina al partitiei unde este instalat Windows. Windows poate fii configurat pentru a utiliza spatiul liber cu privire la orice unitati disponibile pentru pagefiles. Este necesar,cu toate acestea,pe pentru partitia de boot (adica unitatea care contine partitia de Windows), pentru a avea un pagefiles pe el,daca sistemul este configurat pentru a scrie Kernel sau integral dispozitivele de memorie dupa un accident. Windows utilizeaza fisierul de paginare ca de stocare temporara pentru groapa de memorie. Atunci cand sistemul este repornit,Windows copiaza groapa de memorie din fisierul de paginare de la un fisier separat si elibereaza spatiul care a fost folosit in fisierul de paginare.

## 4. Concluzii

Aceasta tema are ca scop însușirea cunoștințelor legate de paginarea în Linux și Windows.

Subiectele sunt foarte variate și multiple; aici prezentând partea cea mai importantă legată de paginare în aceste sisteme de operare.

Tehnica paginării nu este una extrem de complicată, aceasta constând în mapearea unei porțiuni a unui program într-o zonă de memorie. Spațiul adreselor virtuale este împărțit în pagini. Paginarea este strâns legată de memoria virtuală deoarece memoria virtuală combină RAM-ul computerului cu un spațiu temporar de pe hard disk. Atunci când RAM-ul devine insuficient, memoria virtuală mută date din RAM într-un spațiu denumit **fișier de paginare**. Mutarea datelor în și din fișierul de paginare determină eliberarea RAM-ului în scopul terminării lucrului acestuia.

## 5. Bibliografie:

<http://en.wikipedia.org/wiki/Paging>

Andrew S. Tanenbaum, Operating Systems. Design and Implementation, Prentice Hall, 1987.

Andrew S. Tanenbaum, Sisteme de operare moderne, Byblos, 2004