



Suport de curs

OSCE

- Capitolul 8 – Virtual Memory

MOS

- Capitolul 4 – Memory Management
 - Sectiunile 4.4, 4.5

Cuprins

Memoria virtuala
Page fault
Demand paging
Copy-on-Write
Maparea fisierelor
Inlocuirea paginilor

Memoria virtuala

De ce memorie virtuala?

Pagini

Astractizari ale SO

Spatiul virtual de adrese

Avantaje si dezavantaje memorie virtuala

De ce memorie virtuală?

dificil să plasezi un proces de la adresa 0

decuplare de spatiul fizic

- un proces are acces la propriul spatiu fizic

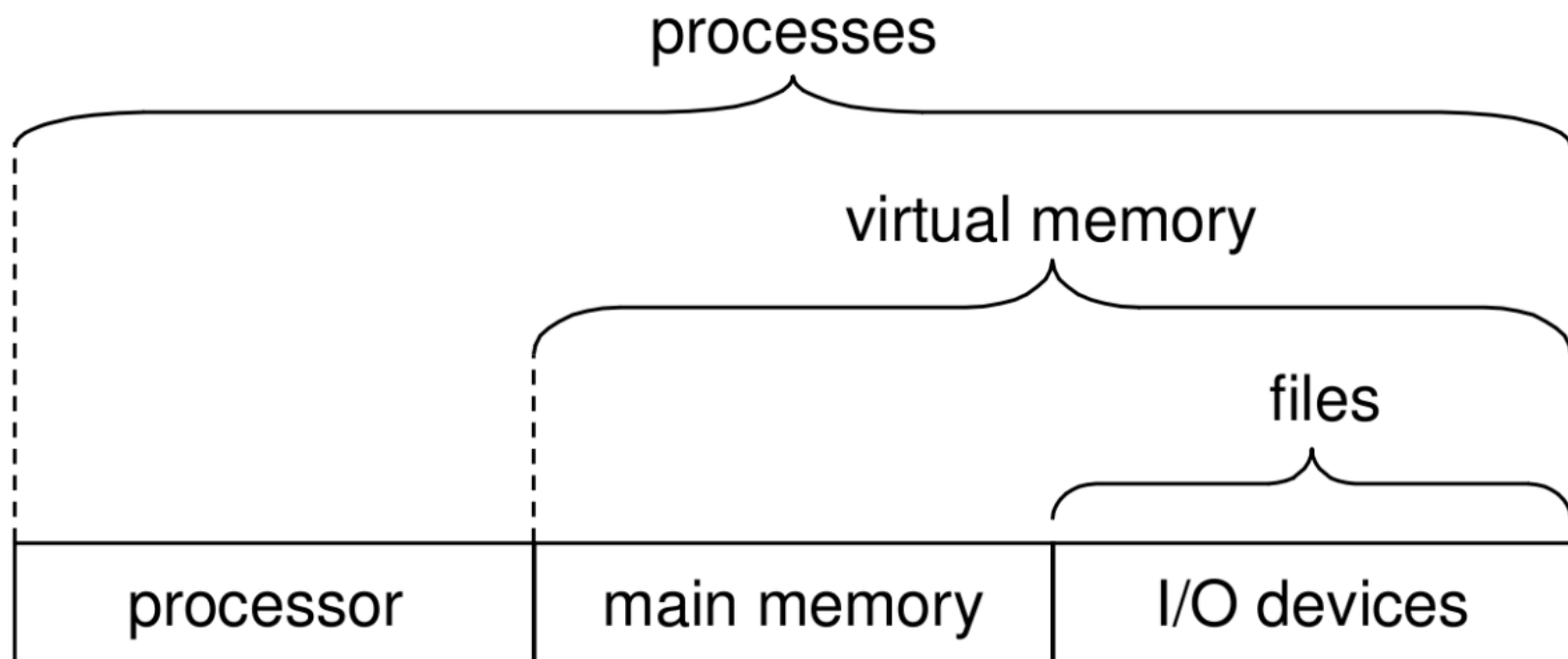
memorie partajata

copy-on-write

maparea fisierelor

poti aloca doar memorie virtuala (demand paging)

Abstractizari ale SO



Pagini

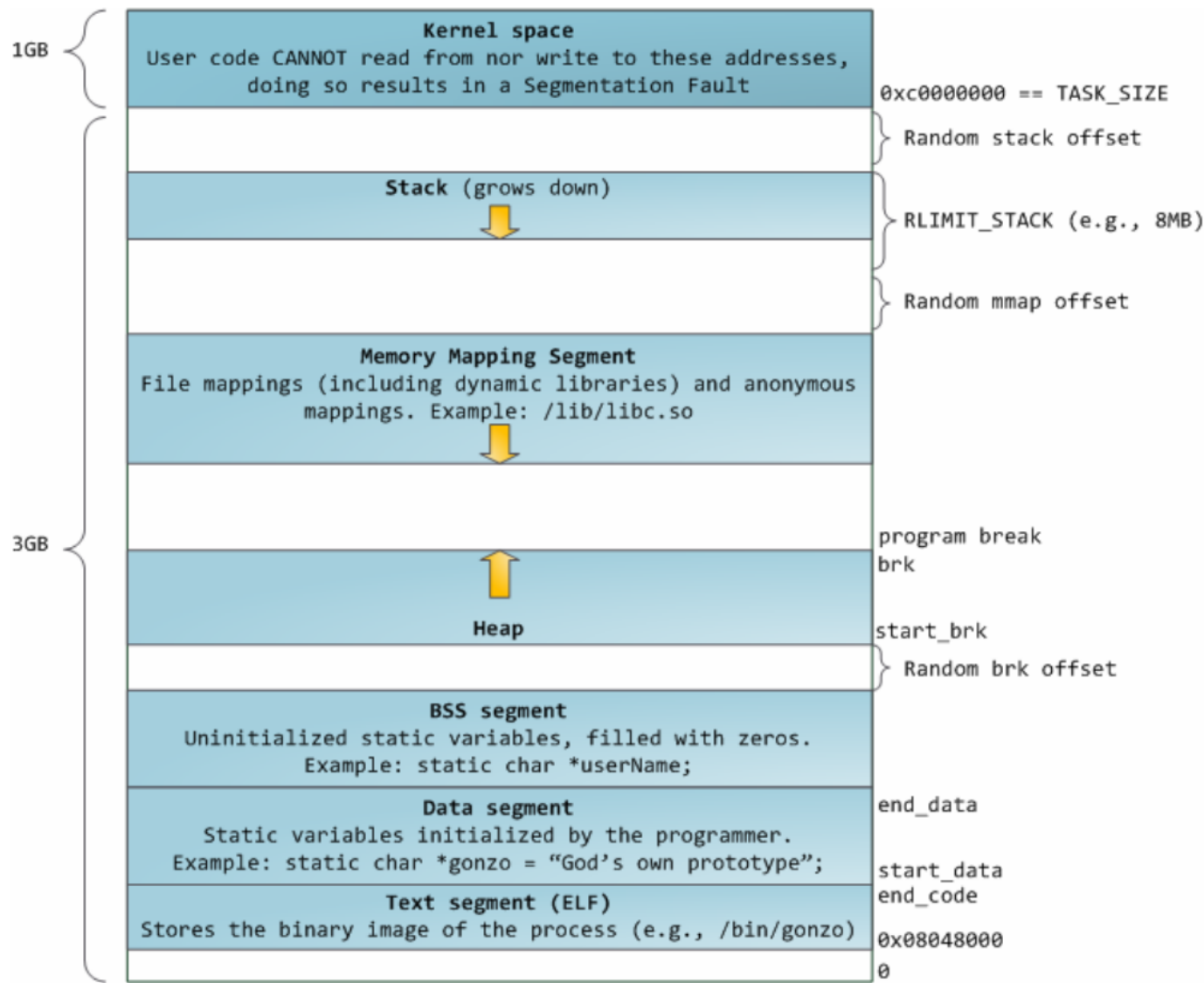
fara fragmentare externa

cea mai mica unitate de alocare la nivel de SO

pagini virtuale (pages)
pagini fizice (frames)

maparea memoriei: asocierea unei pagini virtuale cu una fizica

Spatiul virtual de adrese



<http://duartes.org/gustavo/blog/post/anatomy-of-a-program-in-memory>

Avantaje si dezavantaje memorie virtuala

programatorul este eliberat de

- plasarea codului la o adresa data
- alocarea de memorie fizica

partajare memorie

folosire mai multa memorie decat are sistemul

acces dublu la memorie (tabela de pagini, TLB)

suport hardware (MMU, TLB)

componenta dedicata in SO (complexitate)

Concepte de memorie virtuala

Page fault
Swap
Demand paging

un proces in
• restul sur

o alocare
• pagin

Page fault

acces la o pagina virtuala:

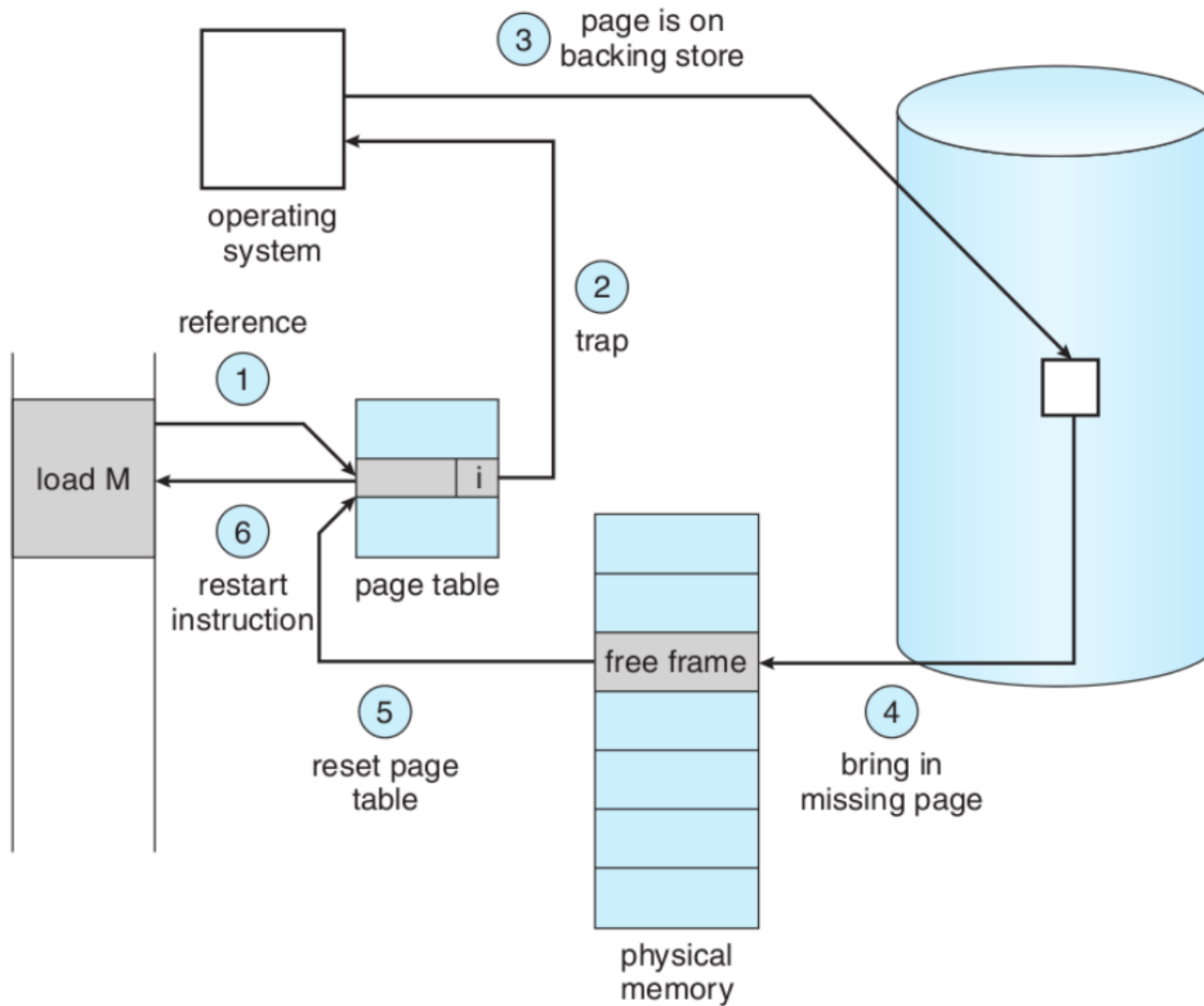
- nemapata (demand paging, swap)
- nevalida (pagina virtuala nealocata)
- drepturi nevalide (scriere pe o pagina read-only, copy-on-write)

se genereaza o exceptie/trap
MMU genereaza exceptia
se ruleaza page fault handler-ul

page fault-urilor pot sa nu fie erori

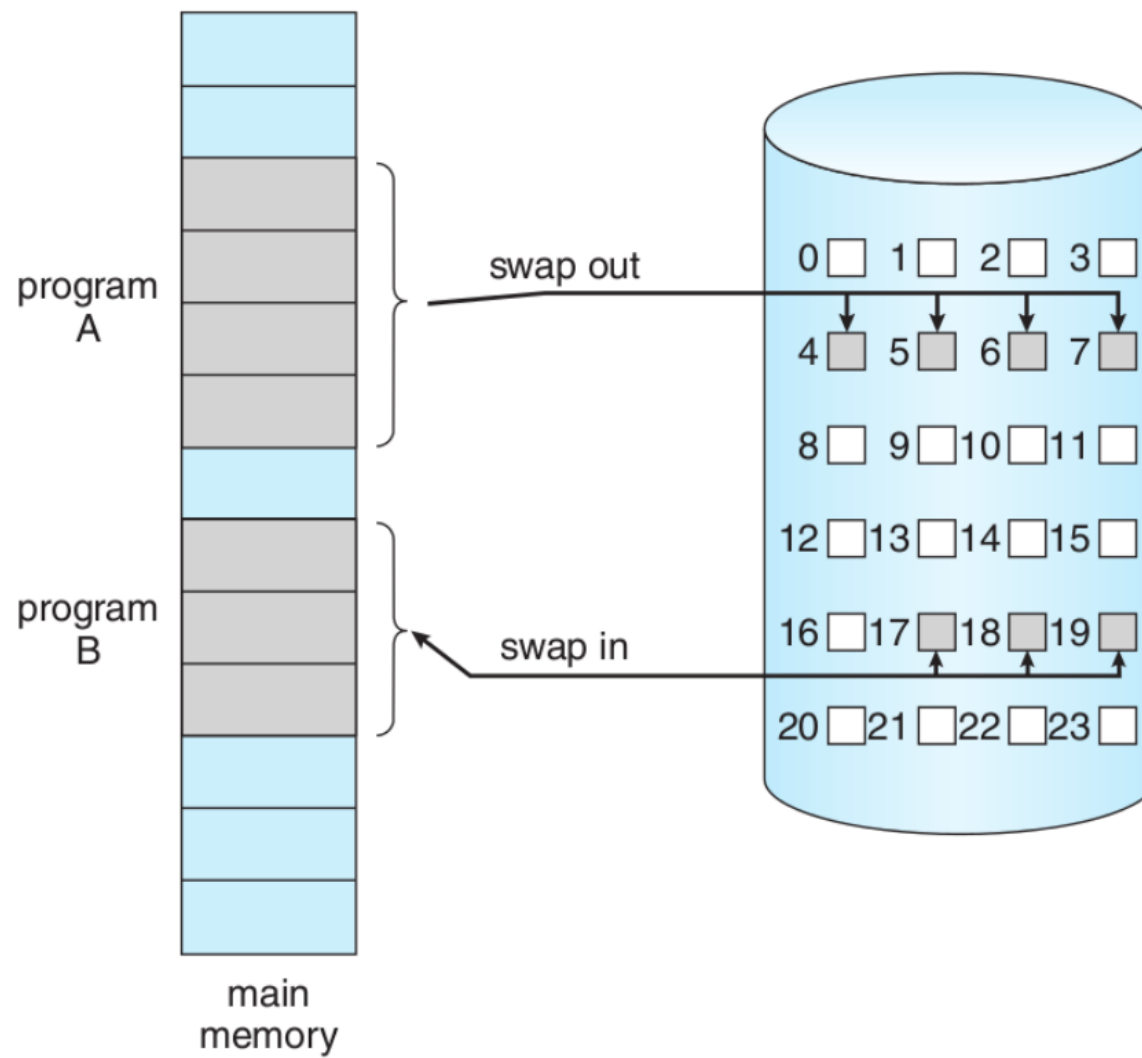
- minor page faults: demand paging
- major page faults: swap

Page fault (2)



OSCE, Chapter 8, pg. 325, Figure 8.6

Swap



Swap (2)

in absenta memoriei fizice, se pot evacua pagini pe disc
spatiu de swap

- swap in (aducerea unei pagini de pe disc in RAM)
- swap out (trimiterea unei pagini din RAM pe disc)

mareste spatiul posibil folosit
mult mai incet decat memoria RAM

algoritmi de inlocuire de pagina

Demand paging

Paginare la cerere

paginare = alocarea unei pagini fizice si maparea acesteia

decuplarea alocarii de memorie virtuala de mapare

lazy allocation/paging

la alocare se alocă intrare în tabela de pagini

- pagina este marcată nevalidă
- este alocată pagina fizică (de la zero sau de swap) la nevoie

Demand paging (2)

un proces incarcat in memorie isi incarca doar o parte din date

- restul sunt incarcate la nevoie

o alocare cu mmap aloca doar pagini virtuale

- paginile fizice sunt alocate la acces

alocarea fizica si maparea se face in page fault handler

Alocarea memoriei virtuale

Rezervare si commit
Demand paging
Pagini rezidente
Page locking

Rezervare si commit

rezervare: alocare memorie virtuala (doar virtuala)

- apelurile malloc (dimensiuni mari), mmap, VirtualAlloc

commit: alocare memorie fizica pentru memoria alocata

- la demand paging

decuplarea celor doua permite alocare rapid de memorie

apelurile de biblioteca malloc fac mai putine apeluri de sistem brk

Demand paging

alocare la cerere

rezervare in prima faza

commit in page fault handler, la acces

Pagini rezidente

paginile prezente in RAM sunt rezidente
altfel pot fi pe swap sau inca nealocate

RSS = Resident Set Size, spatiul ocupat de un proces in RAM

alocare rezidenta (paginile nu pot fi swappate)

- la nivelul nucleului de operare
- sau prin page locking/pinning

Page locking

page pinning, fixed pages, non-pageable

pagina este marcata rezidenta (neswappabila)

utila pentru comunicarea cu dispozitive I/O

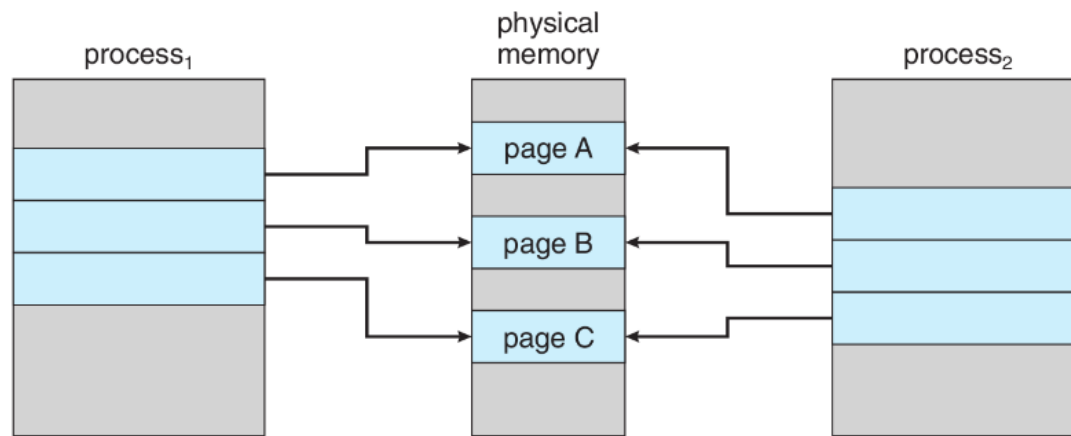
mlock() - Linux

VirtualLock() - Windows

Mecanisme de memorie virtuala

Copy-on-Write
Maparea fisierelor

Copy-on-Write



OSCE, Chapter 8, pg. 330, Figures 8.6 & 8.8

Copy-on-Write (2)

implementare naiva `fork()`: se copiaza spatiul de adresa

implementare eficienta: se duplica tabela de pagini

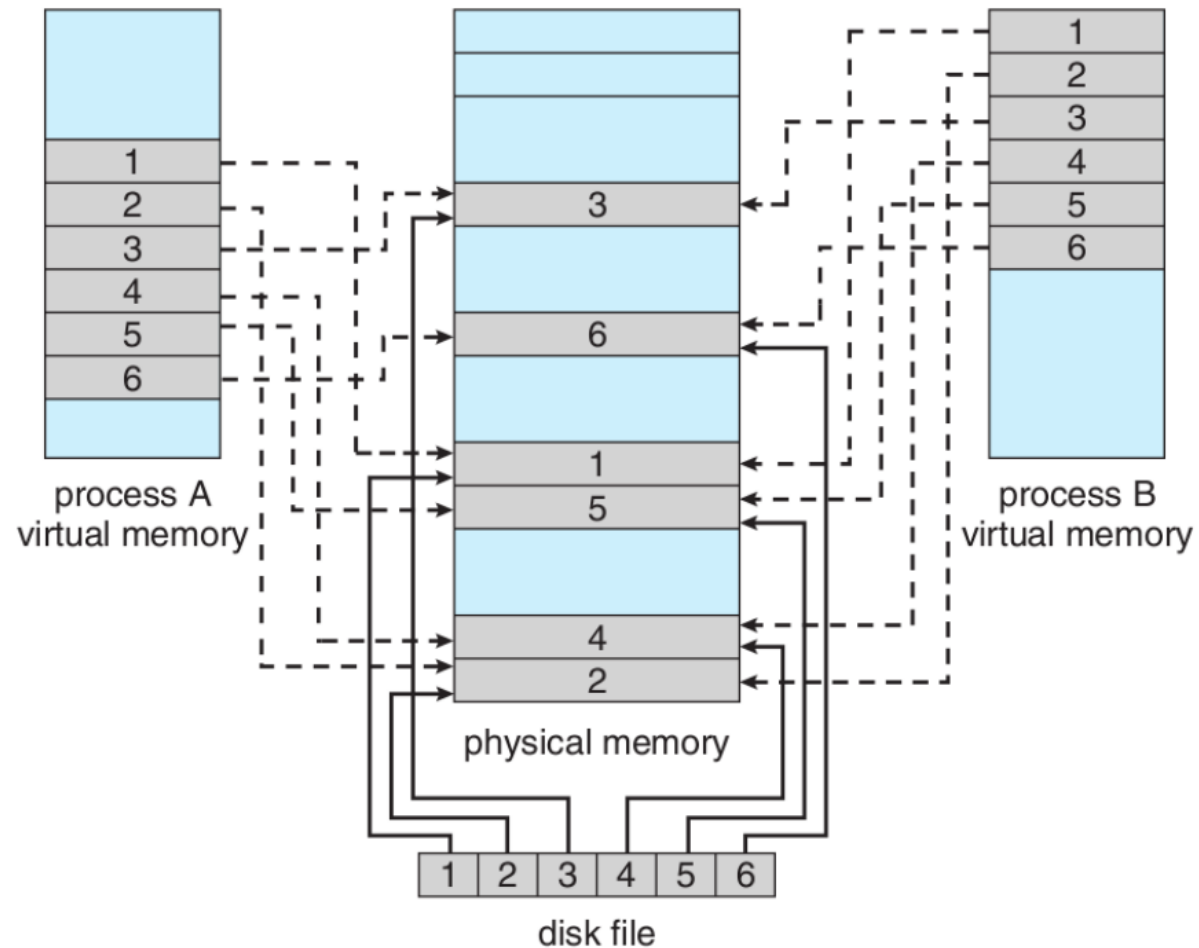
se partajeaza paginile intre procesul fiu si procesul parinte

- paginile sunt marcate read-only

in momentul unui acces de tip scriere se genereaza page fault

- se creeaza o copie a paginii configurata `READ_WRITE` la care va avea acces doar procesul care a generat page fault
- Pagina originala ramane configurata `READ_ONLY` si va avea acces la ea doar celalalt proces

Maparea fisierelor



Maparea fisierelor (2)

un bloc de date al unui fiier este mapat intr-o pagina/set de pagini
lucrul cu fisiere se realizeaza prin operatii de acces la memorie

folositor pentru incarcarea executabilor si bibliotecilor

scrierea la o adresă de memorie inseamna scrierea in fisier
scrierile nu sunt imediate/sincrone (se folosesc apeluri msync)
memoria este folosită pe post de cache

mai putine apeluri de sistem pentru lucrul cu fisiere

permite shared memory

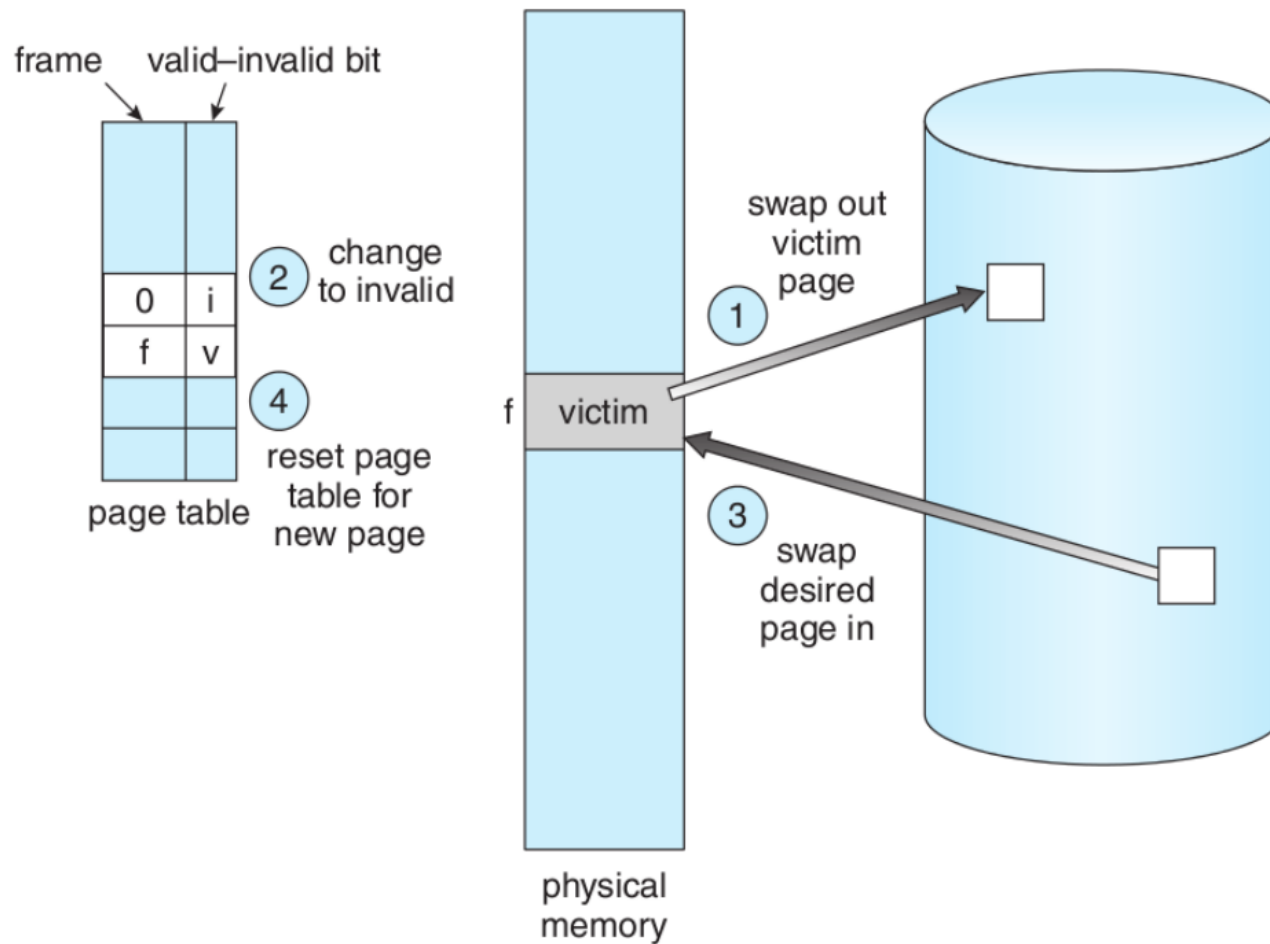
Inlocuirea paginilor

Inlocuirea paginilor
Algoritmi de inlocuire a paginilor
Anomalia lui Belady
Thrashing

Second chance
• se tine cont
• se inspecteaza
• daca $R=0$ pagina este
• daca $R=1$, pagina este

asa cea

Inlocuirea paginilor



Algoritmi de inlocuire a paginilor

ce pagina va fi evacuata pe disc

optim: se inlocuieste pagina care va fi referita cel mai tarziu

biti de modified (M) si referenced (R) in pagina

- o pagina cu M a fost scrisa
- o astfel de pagina este "dirty"
- o pagina cu R evacuata, va fi doar invalidata nu si copiata pe disc

NRU: Not Recently Used (se inlocuieste clasa cea mai mica)

- clasa 0: $R = 0$, $M = 0$
- clasa 1: $R = 0$, $M = 1$
- clasa 2: $R = 1$, $M = 0$
- clasa 3: $R = 1$, $M = 1$

Algoritmi de inlocuire a paginilor (2)

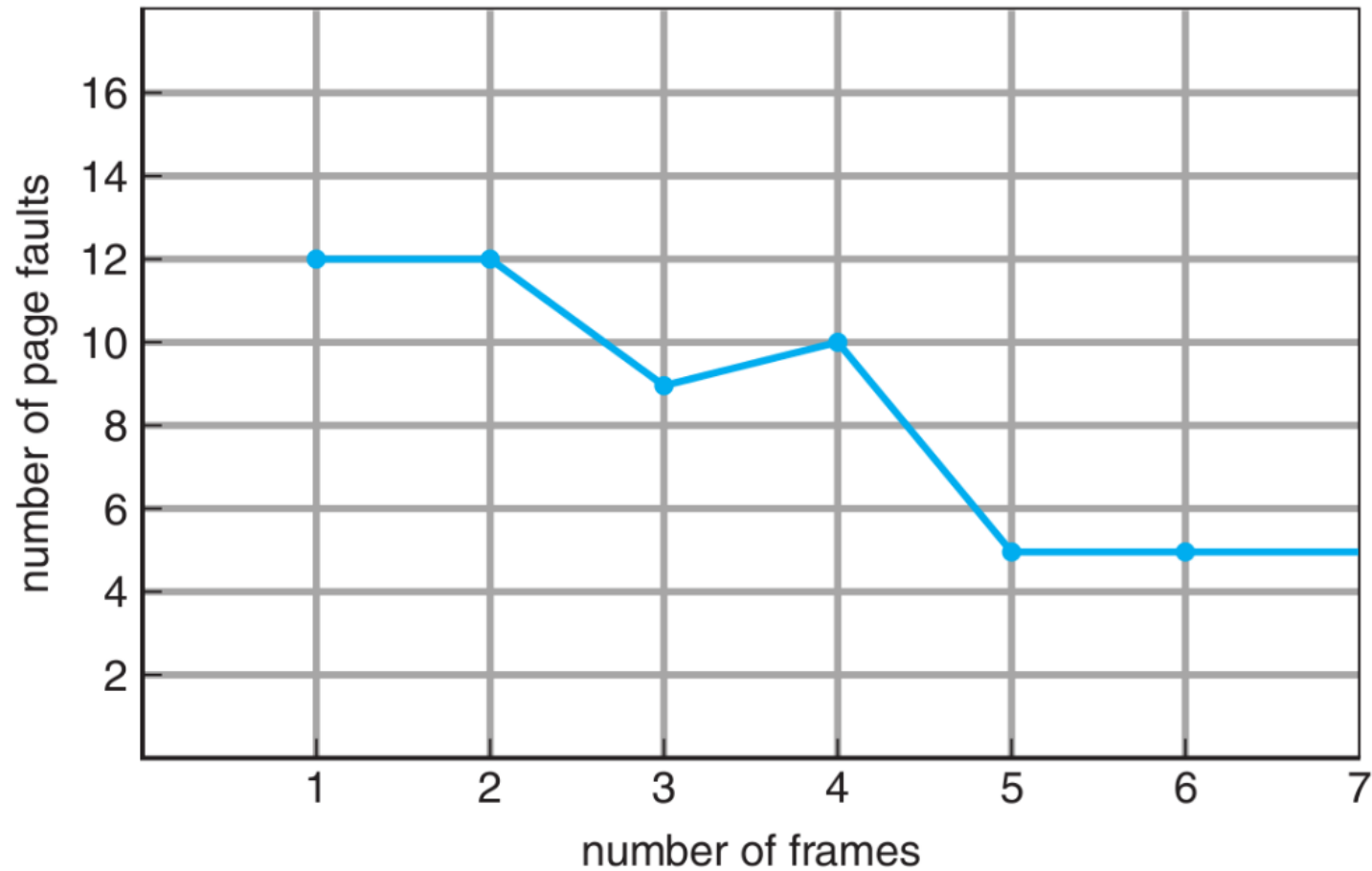
FIFO

- se menine o lista cu paginile din memorie
- noile pagini sunt adaugate la sfarsitul listei
- se inlocuieste prima pagina din lista (cea mai veche)

Second chance (varianta modificata de FIFO)

- se tine cont de bitul R
- se inspecteaza prima pagina din lista
- daca $R=0$ pagina este selectata pentru inlocuire
- daca $R=1$, pagina este mutata la coada listei si $R=0$

Anomalia lui Belady



OSCE, Chapter 8, pg. 337, Figure 8.13

Thrashing

la incarcare mare a sistemului inlocuiri frecvente

schimbarea contextului duce la noi inlocuiri

- swap in/swap out frecvente

se petrece mult timp in page fault-uri (swap in/out)

- ineficienta in folosirea procesorului

exista si notiunea de cache thrashing

Cuvinte cheie

memorie virtuala
spaiu de adresa
swap
demand paging
page fault
copy-on-write

memory-mapped files
memorie partajata
inlocuire de pagina
NRU
FIFO, second chance
thrashing