LINGI 1113: Systèmes informatiques 2

Mission 4: gestion de la mémoire par les systèmes d'exploitation

ÉCOLE POLYTECHNIQUE DE LOUVAIN

UCL

Gestion de la mémoire

Comment un processus (traditionnel) voit la mémoire:

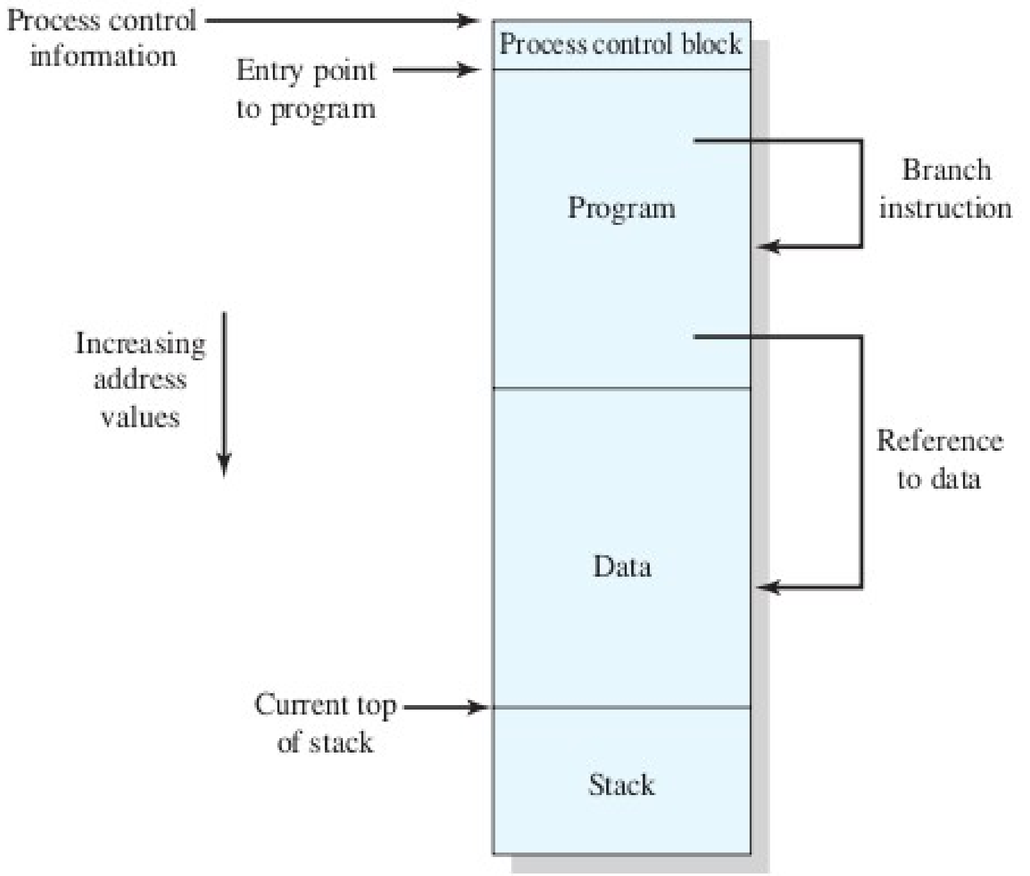


Figure 7.1 **Addressing Requirments for a Process**

Gestion de la mémoire

Problèmes posés à l'OS

- Relocation
- Protection
- Partage
  - Code
    - Tout le code
    - dll/shared objects (ldd)
  - Données
    - Zones de mémoire communes (shared memory: shmop)
    - Memory mapped files (MMAP)

UCL

Gestion de la mémoire

Comment l'OS gère la mémoire (1):

Table 7.2 Memory Management Techniques

Technique	Description	Strengths	Weaknesses
Fixed Partitioning	Main memory is divided into a number of static partitions at system generation time. A process may be loaded into a partition of equal or greater size.	Simple to implement; little operating system overhead.	Inefficient use of memory due to internal fragmentation; maximum number of active processes is fixed.
Dynamic Partitioning	Partitions are created dynamically, so that each process is loaded into a partition of exactly the same size as that process.	No internal fragmentation; more efficient use of main memory.	Inefficient use of processor due to the need for compaction to counter external fragmentation.
Simple Paging	Main memory is divided into a number of equal-size frames. Each process is divided into a number of equal-size pages of the same length as frames. A process is loaded by loading all of its pages into available, not necessarily contiguous, frames.	No external fragmentation.	A small amount of internal fragmentation.

UCL

Gestion de la mémoire

Comment l'OS gère la mémoire (2):

Simple Segmentation	Each process is divided into a number of segments. A process is loaded by loading all of its segments into dynamic partitions that need not be contiguous.	No internal fragmentation; improved memory utilization and reduced overhead compared to dynamic partitioning.	External fragmentation.
Virtual Memory Paging	As with simple paging, except that it is not necessary to load all of the pages of a process. Nonresident pages that are needed are brought in later automatically.	No external fragmentation; higher degree of multiprogramming; large virtual address space.	Overhead of complex memory management.
Virtual Memory Segmentation	As with simple segmentation, except that it is not necessary to load all of the segments of a process. Nonresident segments that are needed are brought in later automatically.	No internal fragmentation, higher degree of multiprogramming; large virtual address space; protection and sharing support.	Overhead of complex memory management.

Gestion de la mémoire

Allocation par zones fixes ou variable:

(a) Equal-size partitions

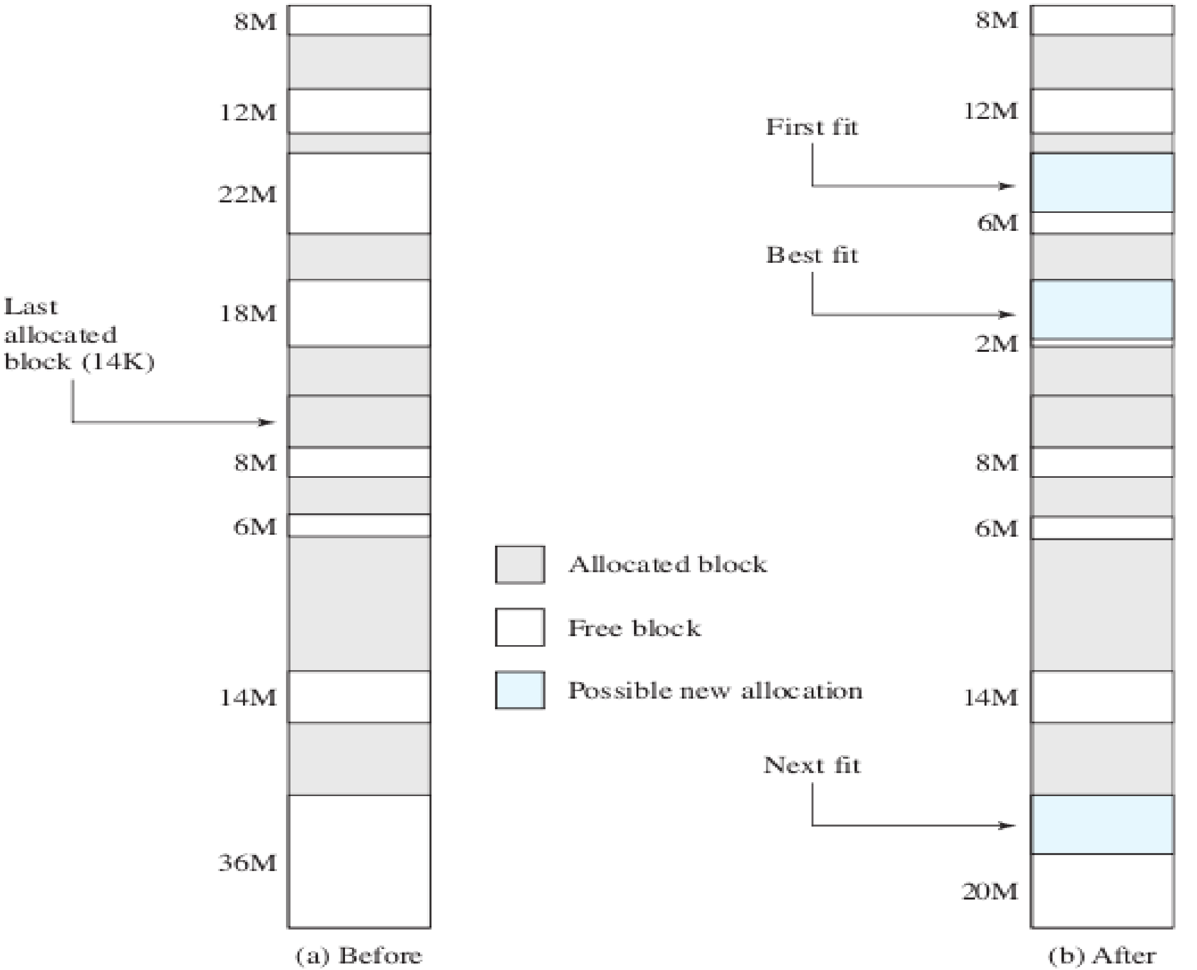


Figure 7.5 **Example Memory Configuration before and after Allocation of 16-Mbyte Block**

Gestion de la mémoire

Allocation par zones variables par tailles 2^n:buddy system (Peterson 77)

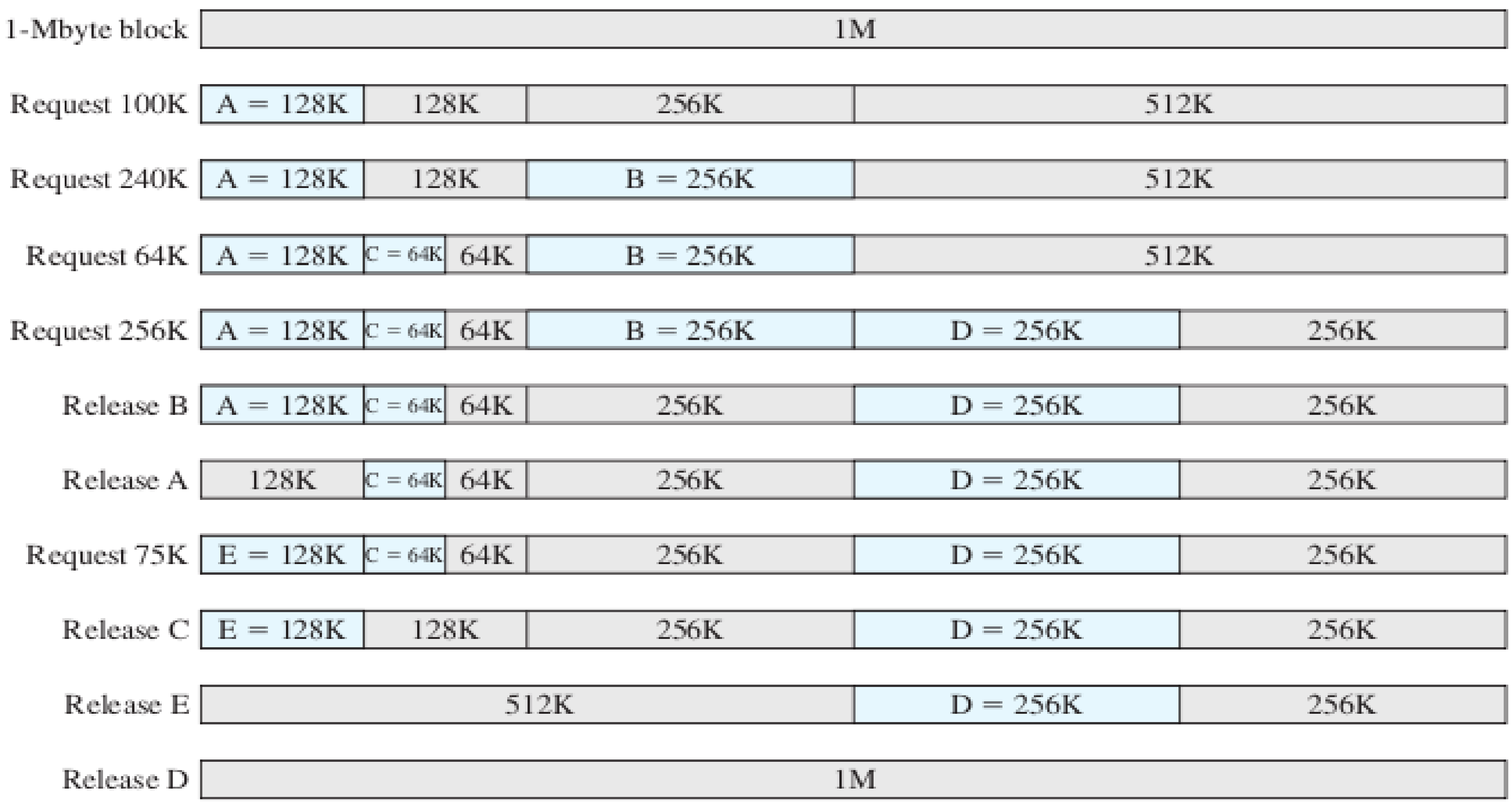


Figure 7.6 Example of Buddy System

Gestion de la mémoire

Le problème de la relocation: le MMU le plus simple

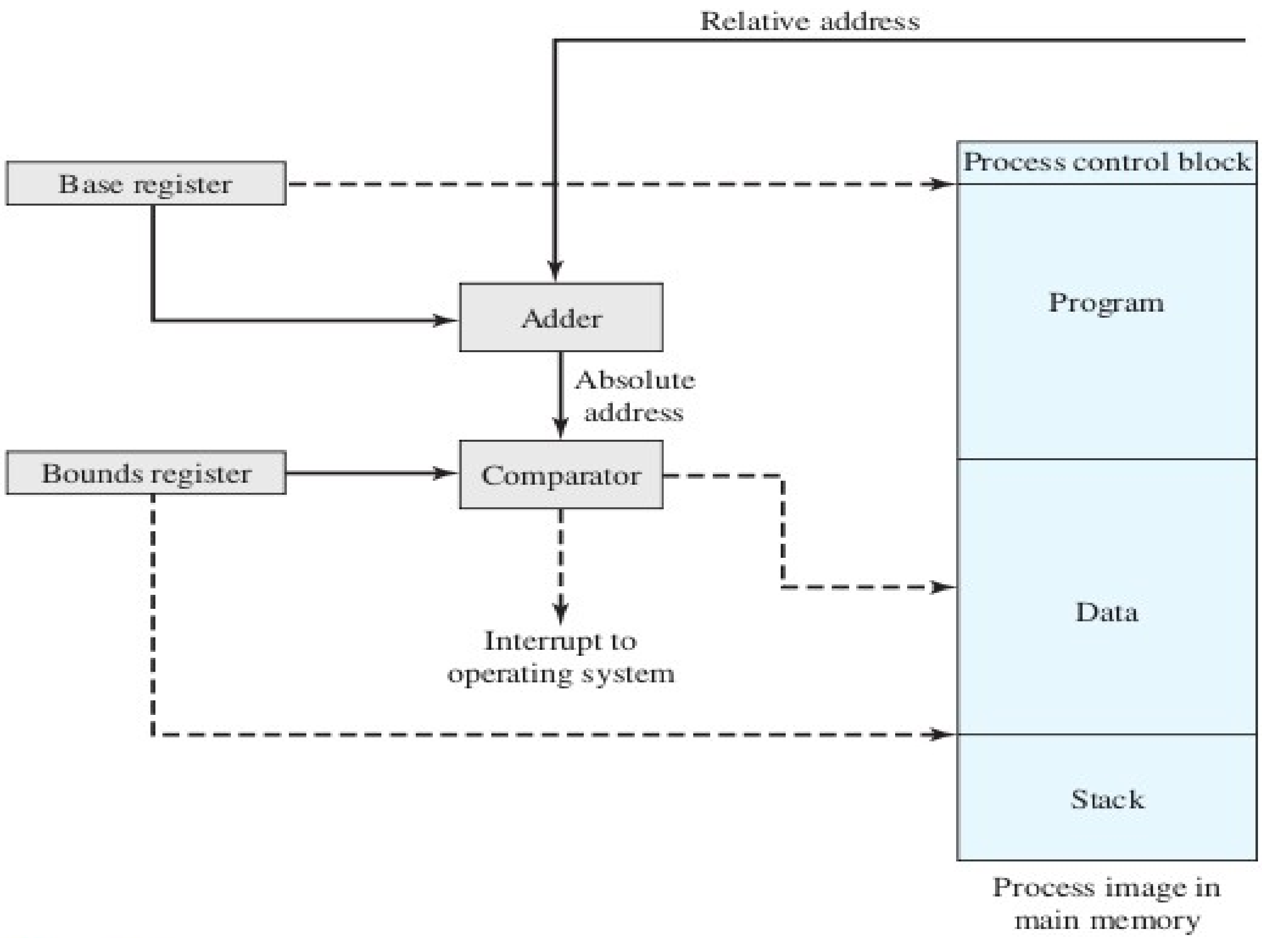


Figure 7.8 Hardware Support for Relocation

Gestion de la mémoire

Le principe des MMU classiques

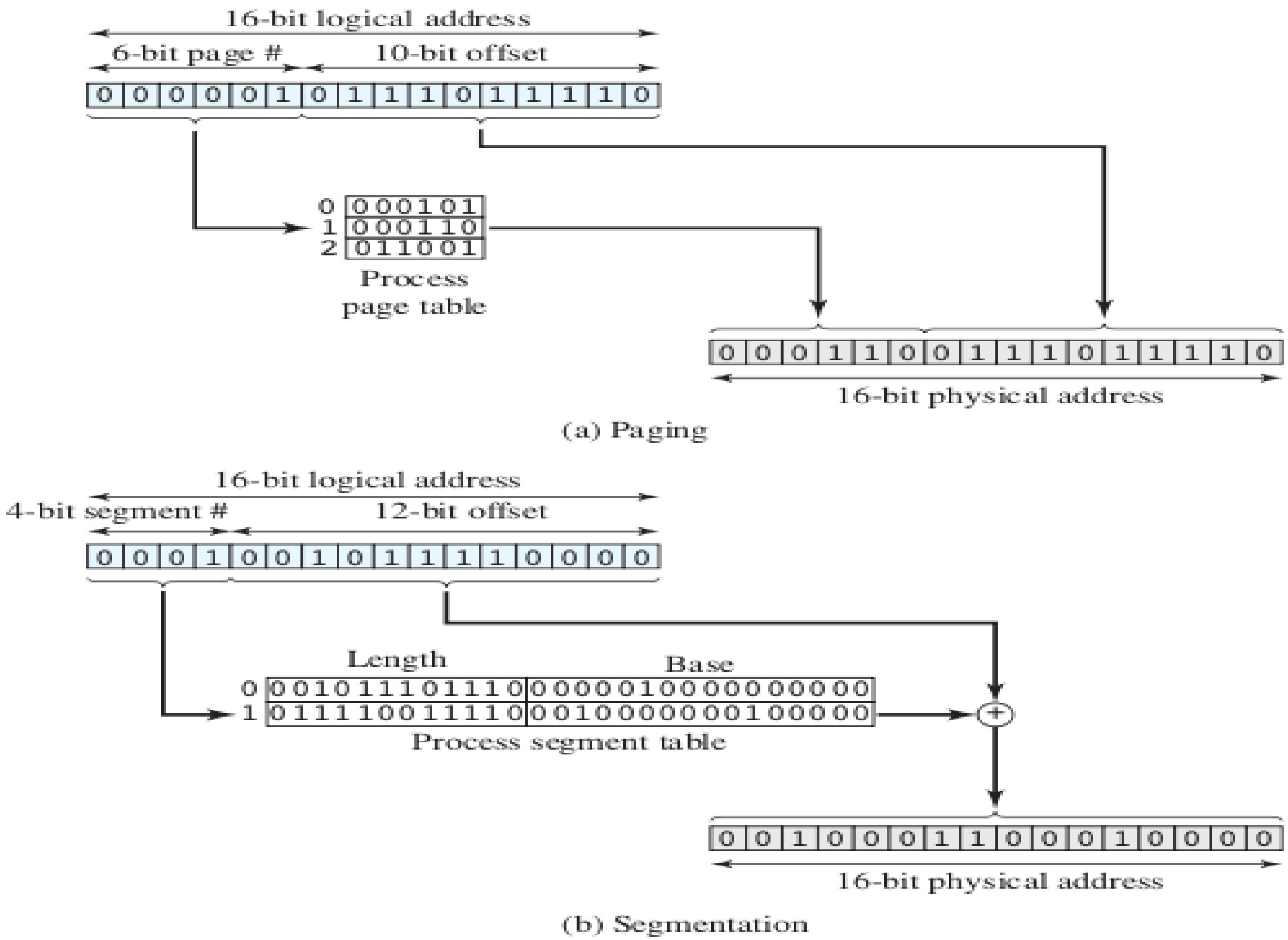


Figure 7.12 **Examples of Logical-to Physical Address Translation**

Gestion de la mémoire

Linking and loading: n'oubliez pas de lire l'appendix 7A

Gestion de la mémoire

La mémoire virtuelle:

- Séparer adresses perçues par le processus des adresses réelles (là où l'information se trouve)
- L'information peut être en mémoire centrale ou sur disque
- Le principe: à un moment donné un processus ne doit avoir accès qu'aux intructions qu'il va bientôt exécuter et aux données qu'il va bientôt utiliser
- En général ça ne change pas trop: on reste dans quelques zones qui évoluents doucement (principe de localité)
- S'il y a assez de place pour ces zones, OK, sinon « thrashing »

Gestion de la mémoire

La mémoire virtuelle:

exemple de comportement

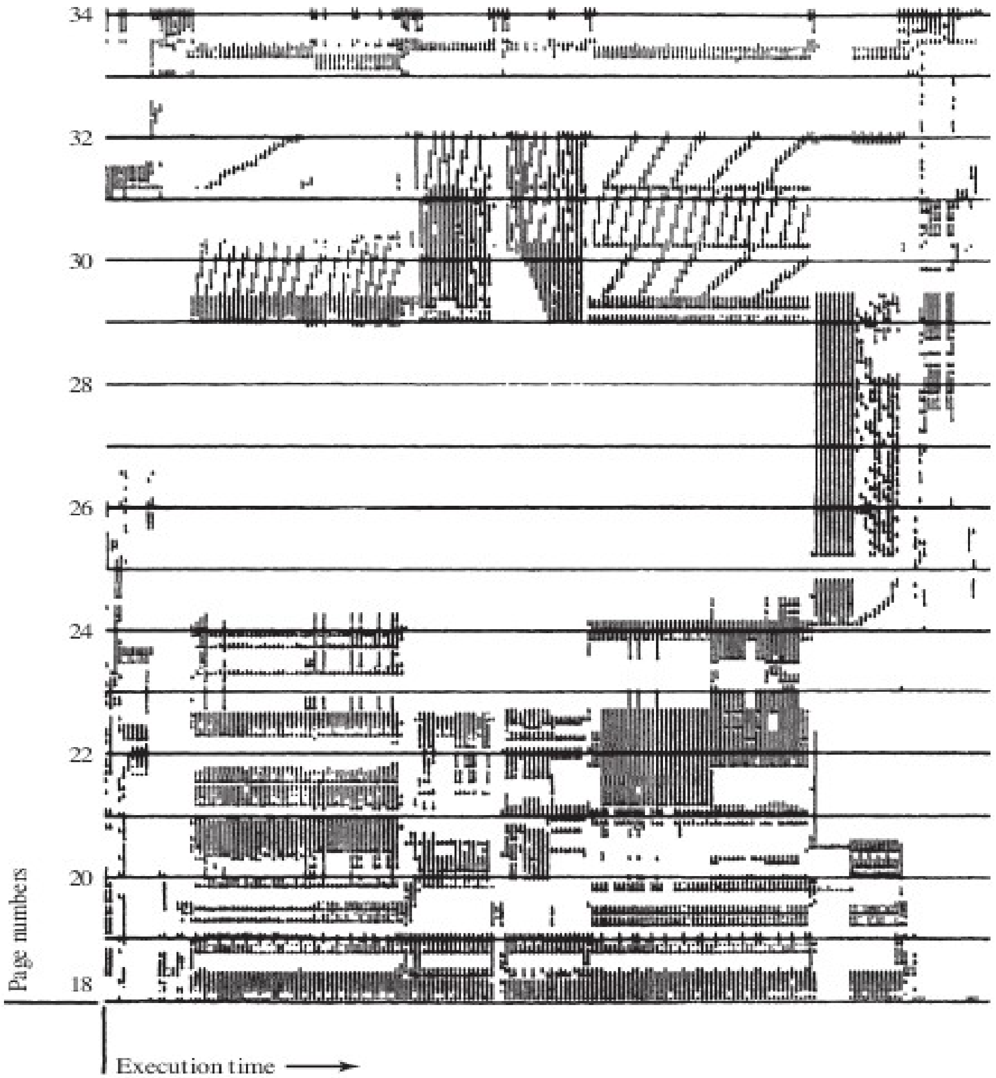


Figure 8.1 **Paging Behavior**

Gestion de la mémoire

La mémoire virtuelle: un système de traduction dynamique d'adresses paginé simple

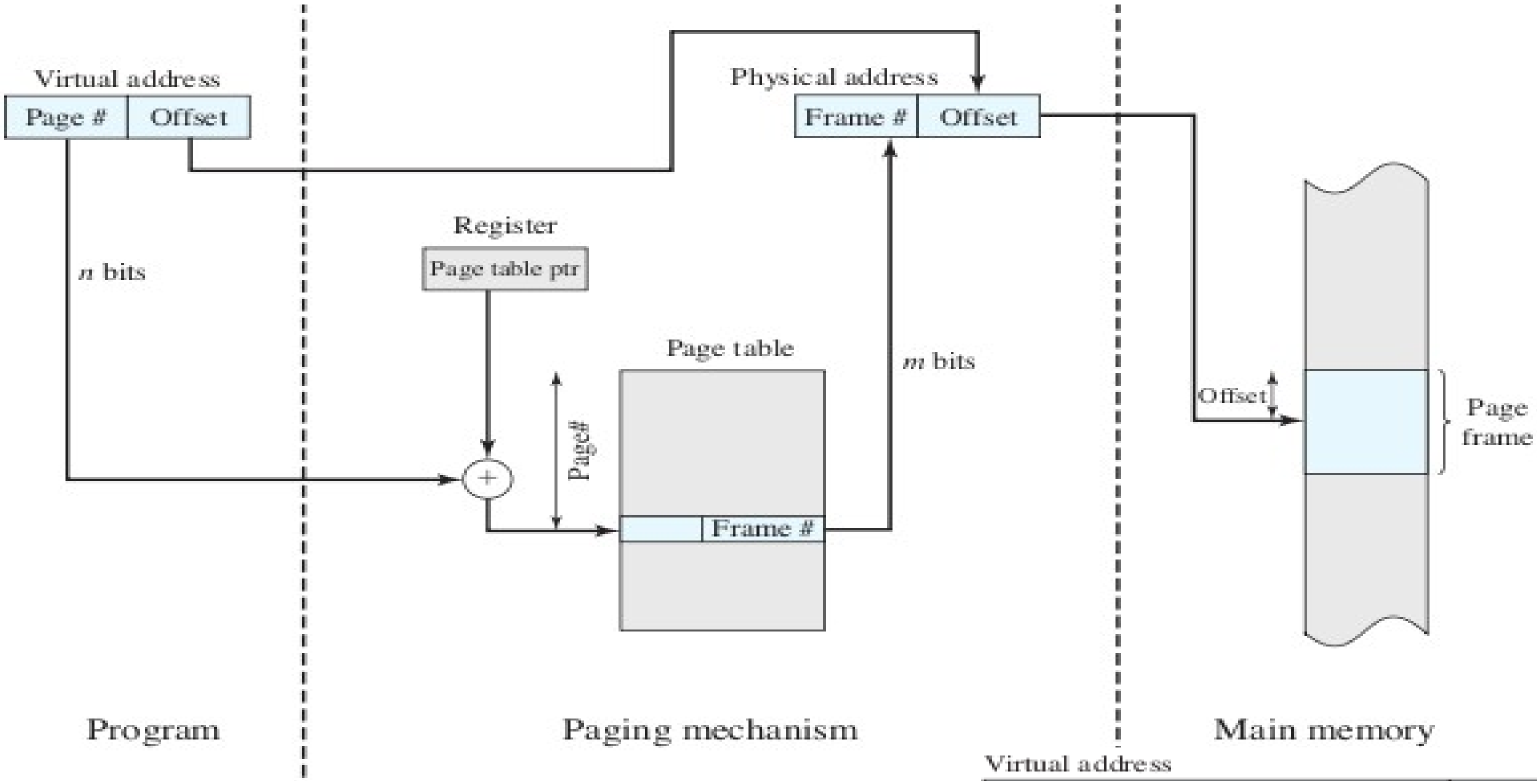
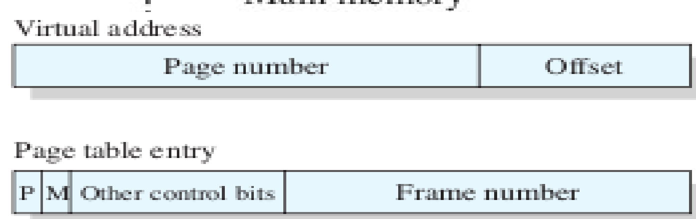


Figure 8.3 **Address Translation in a Paging System**



Gestion de la mémoire

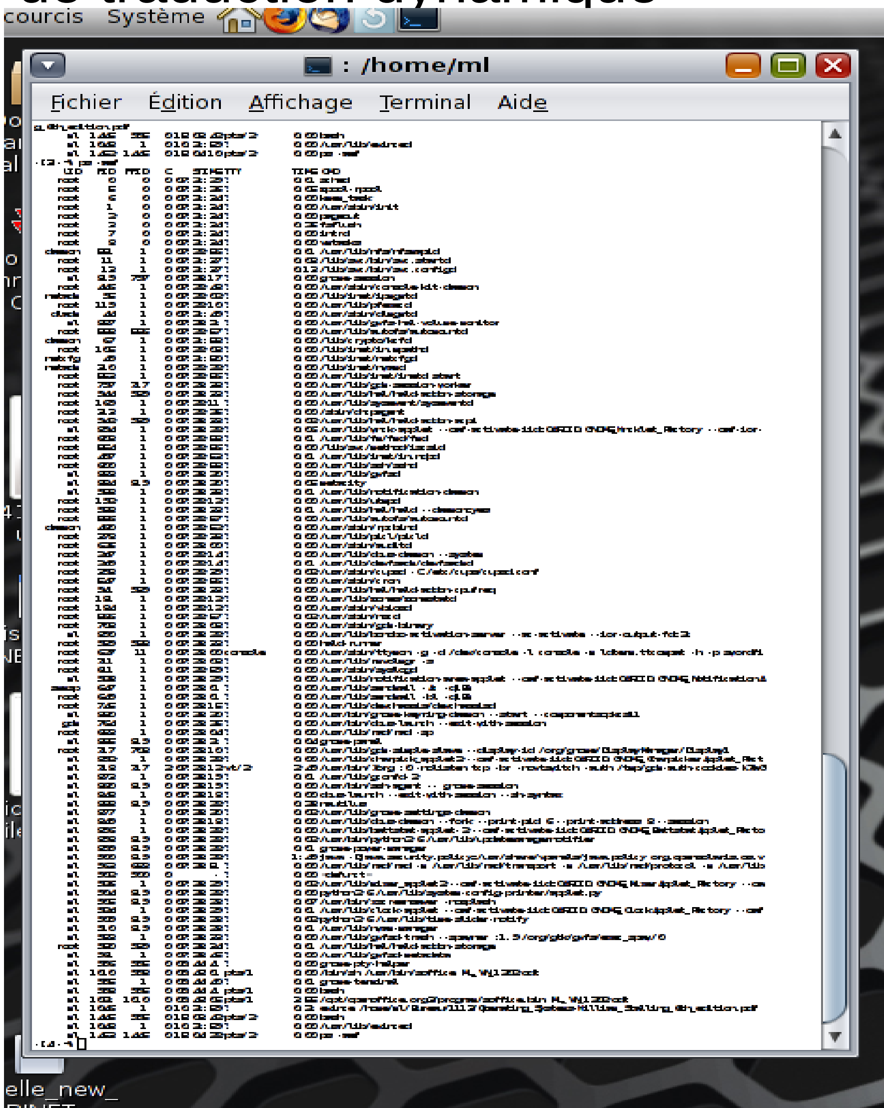
La mémoire virtuelle: un système de traduction dynamique d'adresses paginé simple

Problèmes:

- Si taille page : X et taille mémoire= Y, nombre de lignes de table des pages = x/y
  Si x=4K et y=4G, ça fait...
- Une table des pages par processus (même tout petits) et il y a beaucoup de processus: ex portable SOLARIS: 97

Comment ne pas gaspiller trop de place en mémoire ?

Plusieurs niveaux de tables des pages



Gestion de la mémoire

La mémoire virtuelle: un système de traduction dynamique d'adresses paginé à plusieurs niveaux: 1 table de niveau2 par entrée de table de niveau 1

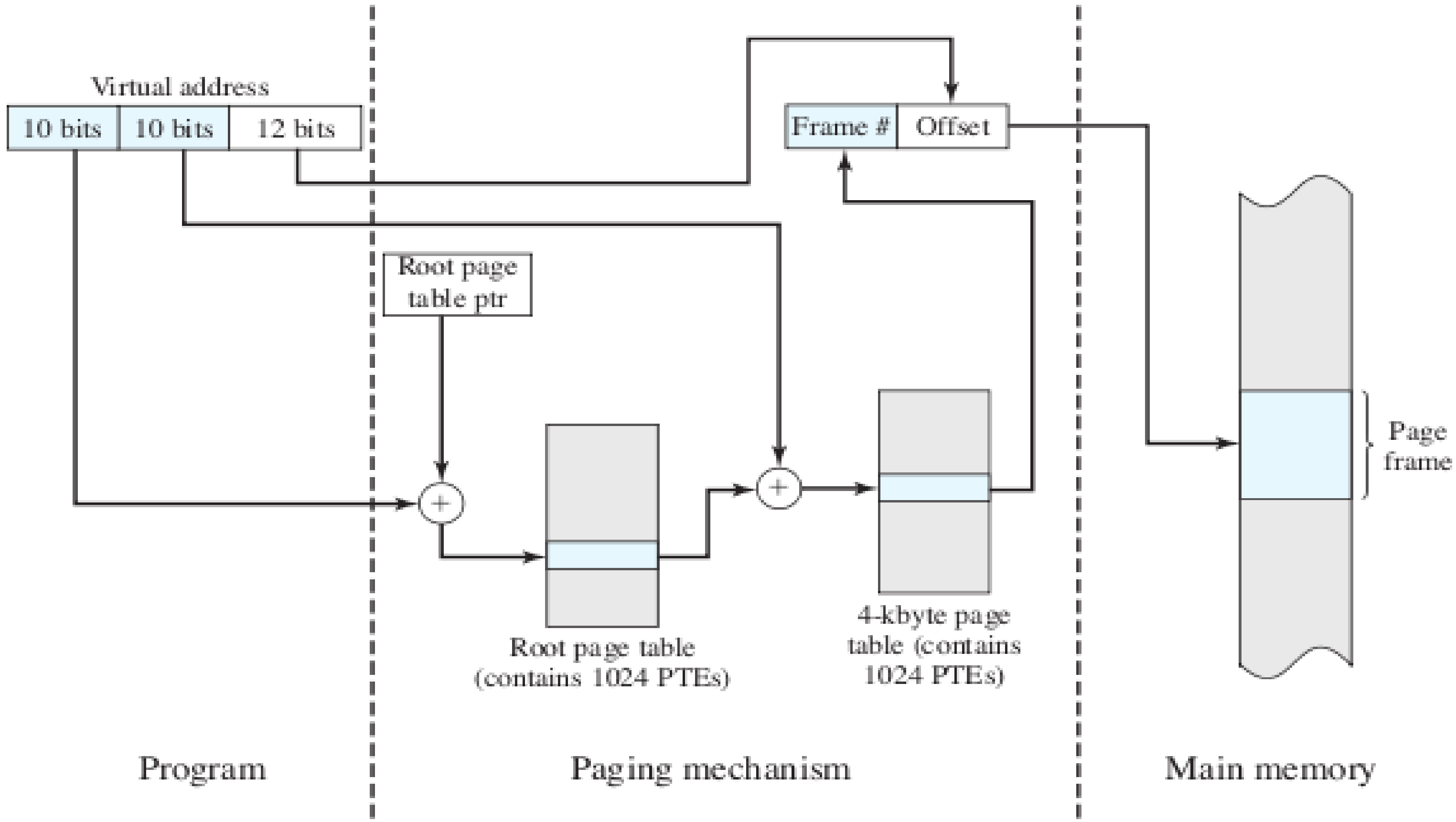


Figure 8.5 Address Translation in a Two-Level Paging System

UCL

Gestion de la mémoire

La mémoire virtuelle: un système de traduction dynamique d'adresses paginé à plusieurs niveaux: 1 table de niveau2 par entrée de table de niveau 1

Problème: nombreux accès mémoire => trop lent donc utiliser une cache associative ds tables des pages: le TLB

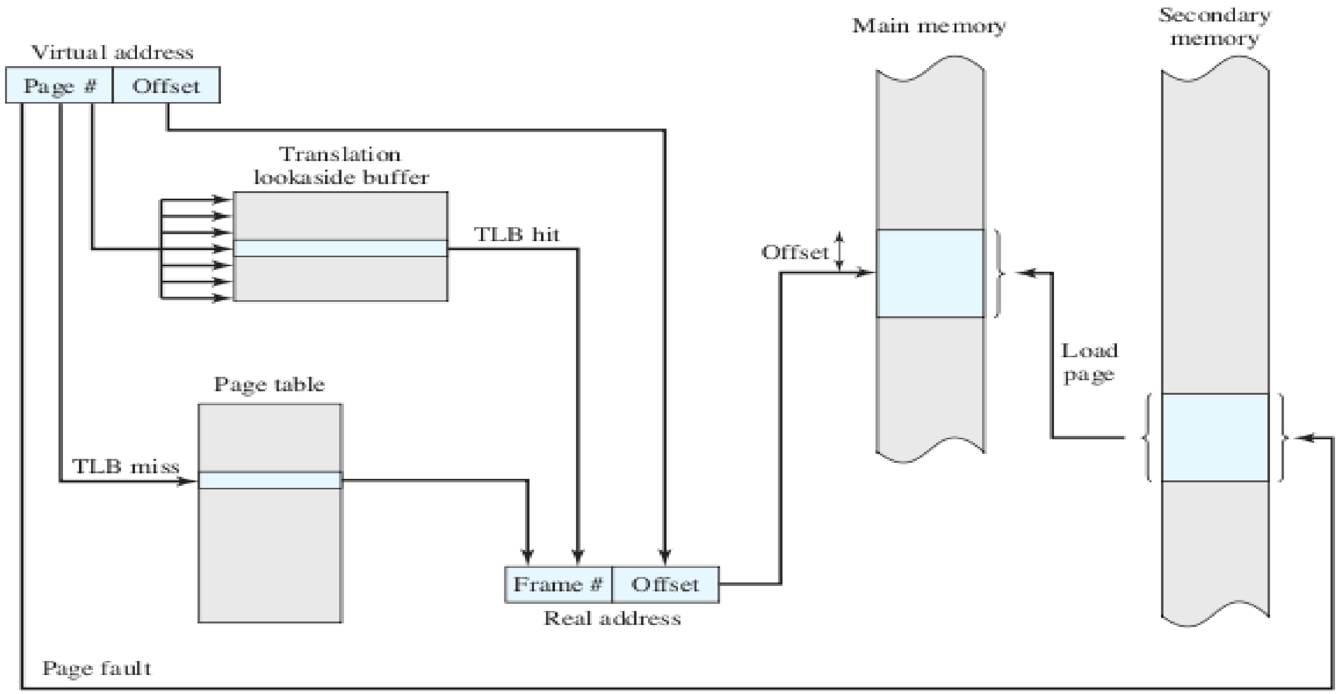


Figure 8.7 Use of a Translation Lookaside Buffer

Gestion de la mémoire

La mémoire virtuelle: Table directe vs TLB

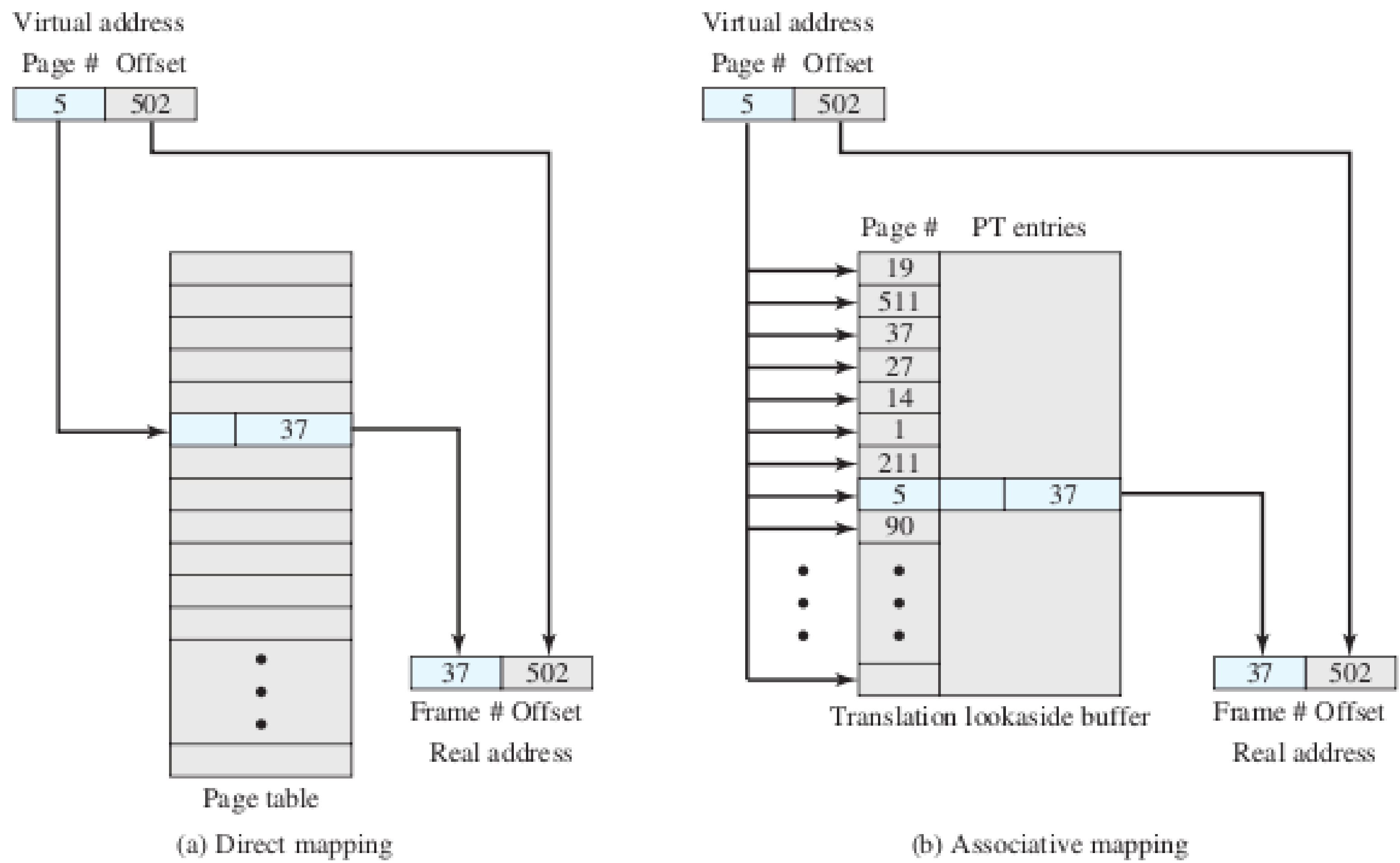


Figure 8.9 Direct versus Associative Lookup for Page Table Entries

Gestion de la mémoire

La mémoire virtuelle: Une lecture en mémoire ne fait aucun accès mémoire si on a 2 caches: le TBLB et la « vraie » cache

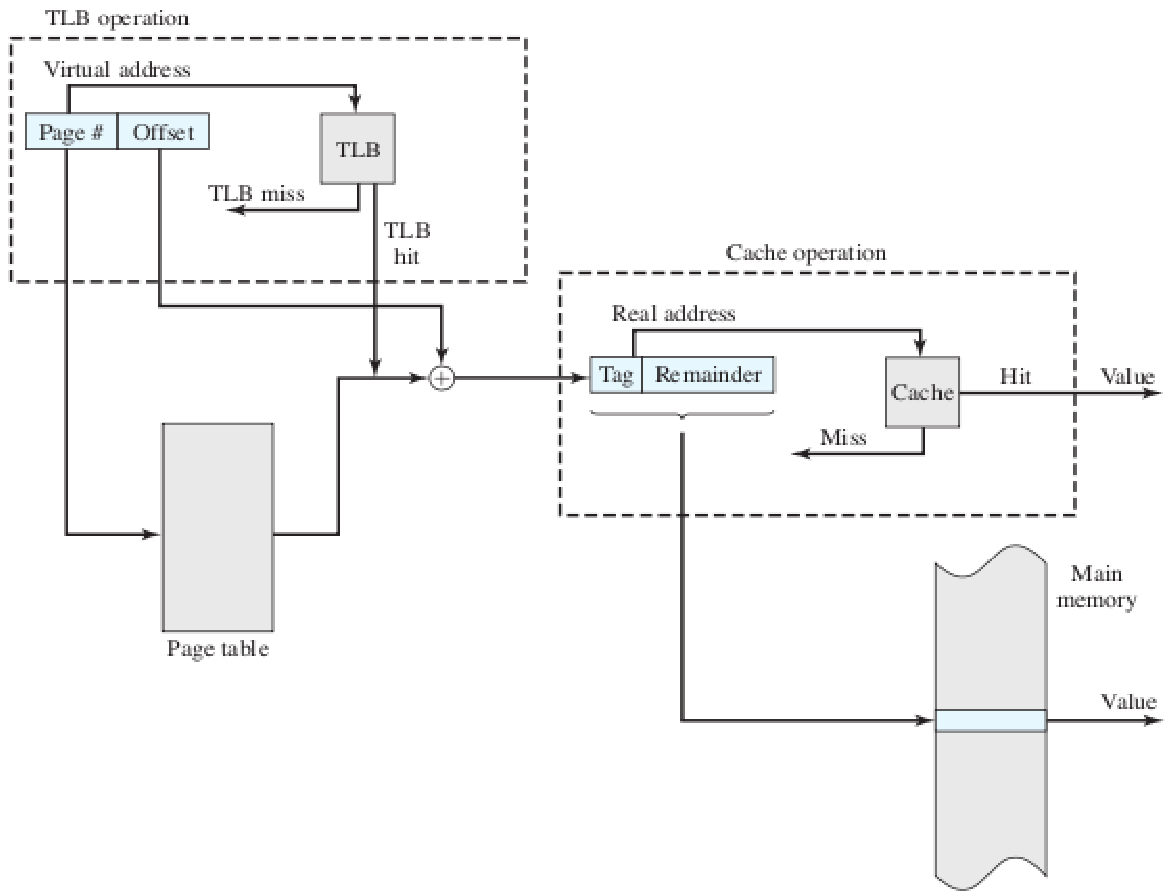


Figure 8.10 Translation Lookaside Buffer and Cache Operation

Gestion de la mémoire

La mémoire virtuelle: gestion par l'OS

Placement policy: irrelevant if paging is used

Optimal replacement: eject page not to be used for the longest time in yhe future

Clock: mark pages that are used & use fifo but don't eject marked pages but unmark them
(« used » bit needed in page tables)

Table 8.4 Operating System Policies for Virtual Memory

Fetch Policy	**Resident Set Management**
Demand	Resident set size
Prepaging	Fixed
Placement Policy	Variable
Replacement Policy	Replacement Scope
Basic Algorithms	Global
Optimal	Local
Least recently used (LRU)	**Cleaning Policy**
First-in-first-out (FIFO)	Demand
Clock	Precleaning
Page buffering	
	Load Control
	Degree of multiprogramming

UCL

Gestion de la mémoire

La mémoire virtuelle: gestion par l'OS

Avoir assez de place est plus important que le choix de l'algorithme de remplacement

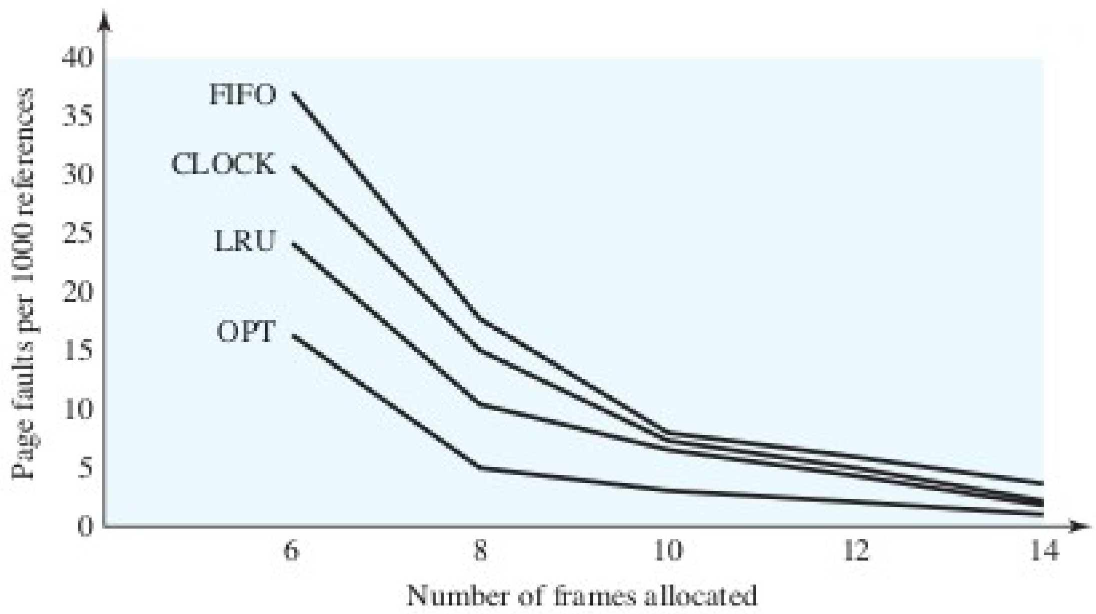


Figure 8.17 **Comparison of Fixed-Allocation, Local Page Replacement Algorithms**

Gestion de la mémoire

La mémoire virtuelle: gestion par l'OS

Où chercher les pages à éjecter ?

Table 8.5 Resident Set Management

	Local Replacement	Global Replacement
Fixed Allocation	• Number of frames allocated to process is fixed. • Page to be replaced is chosen from among the frames allocated to that process.	• Not possible.
Variable Allocation	• The number of frames allocated to a process may be changed from time to time, to maintain the working set of the process. • Page to be replaced is chosen from among the frames allocated to that process.	• Page to be replaced is chosen from all available frames in main memory; this causes the size of the resident set of processes to vary.

Gestion de la mémoire

La mémoire virtuelle: gestion par l'OS

Nombre de pages vraiment utilisées (Working set: concept introduit par Denning) w(t,Δ)=nbr de pages référencées depuis un temps Δ

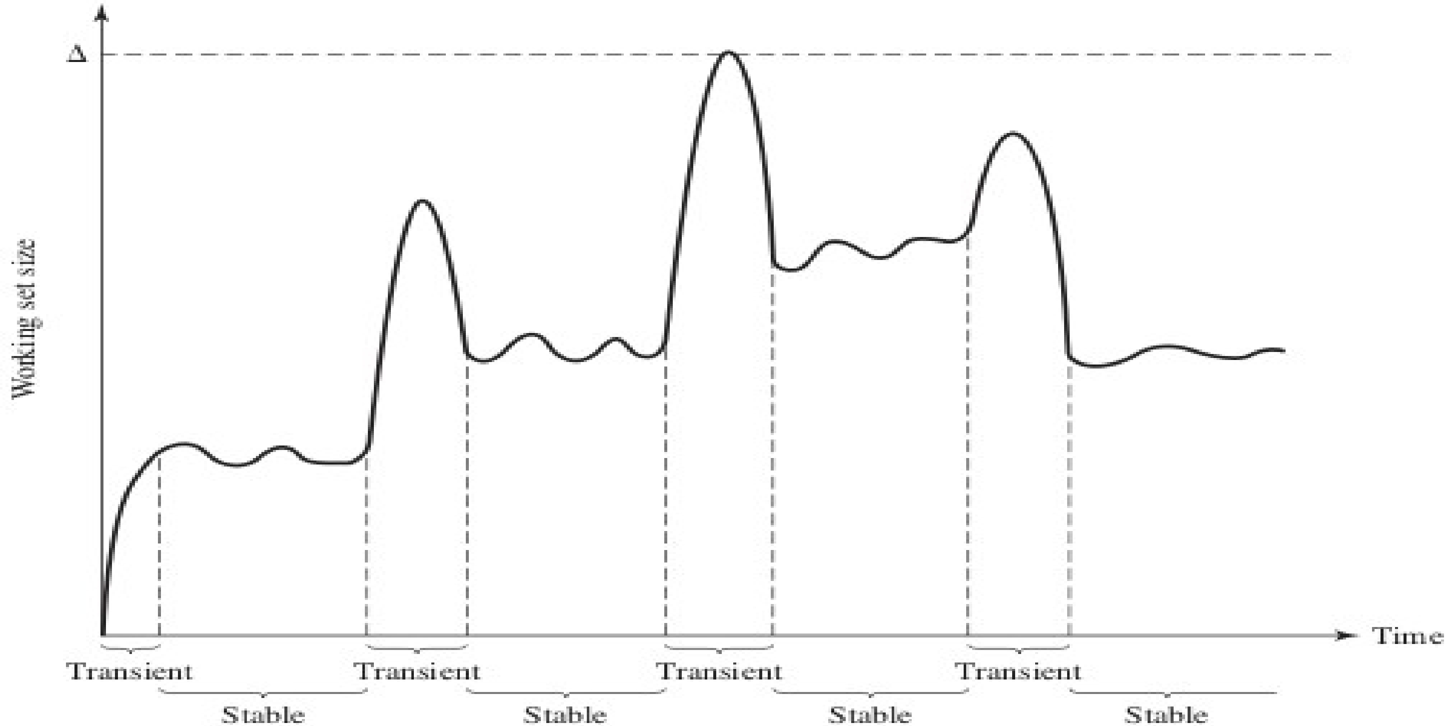


Figure 8.20 Typical Graph of Working Set Size [MAEK87]

Gestion de la mémoire

La mémoire virtuelle: gestion par l'OS

Nombre de pages vraiment utiles pour un processus= le working set pour un Δ raisonable: pas facile à identifier

Autre idée: mesurer fréquence des fautes de pages dans un processus: fréquent: allouer plus de « frames »; jamais: retirer des « frames »

Gestion de la mémoire

La mémoire virtuelle: gestion par l'OS:

exemple de structures de données utilisées: UNIX/SOLARIS

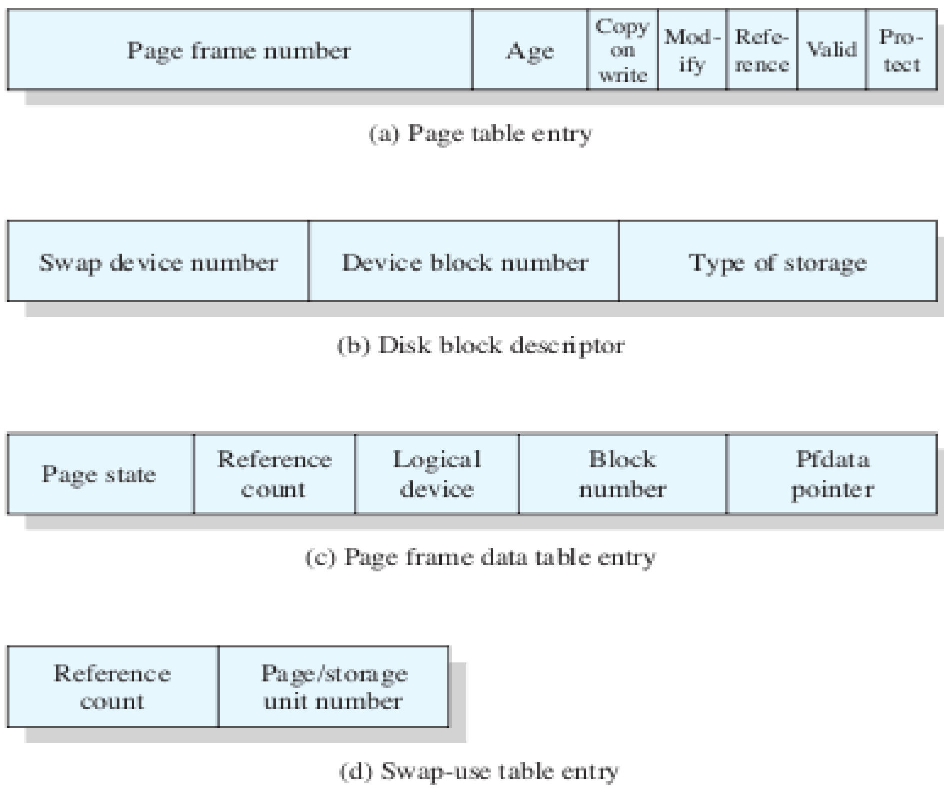


Figure 8.22 **UNIX SVR4 Memory Management Formats**

Gestion de la mémoire

La mémoire virtuelle: gestion par l'OS:

exemple de structures de données utilisées: UNIX/SOLARIS

Table 8.6 **UNIX SVR4 Memory Management Parameters**

Page Table Entry

Page frame number
Refers to frame in real memory.

Age
Indicates how long the page has been in memory without being referenced. The length and contents of this field are processor dependent.

Copy on write
Set when more than one process shares a page. If one of the processes writes into the page, a separate copy of the page must first be made for all other processes that share the page. This feature allows the copy operation to be deferred until necessary and avoided in cases where it turns out not to be necessary.

Modify
Indicates page has been modified.

Reference
Indicates page has been referenced. This bit is set to zero when the page is first loaded and may be periodically reset by the page replacement algorithm.

Valid
Indicates page is in main memory.

Protect
Indicates whether write operation is allowed.

Gestion de la mémoire

La mémoire virtuelle: gestion par l'OS:
exemple de structures de données utilisées: UNIX/SOLARIS

Disk Block Descriptor

Swap device number
Logical device number of the secondary device that holds the corresponding page. This allows more than one device to be used for swapping.

Device block number
Block location of page on swap device.

Type of storage
Storage may be swap unit or executable file. In the latter case, there is an indication as to whether the virtual memory to be allocated should be cleared first.

Page Frame Data Table Entry

Page State
Indicates whether this frame is available or has an associated page. In the latter case, the status of the page is specified: on swap device, in executable file, or DMA in progress.

Reference count
Number of processes that reference the page.

Logical device
Logical device that contains a copy of the page.

Block number
Block location of the page copy on the logical device.

Pfdata pointer
Pointer to other pfdata table entries on a list of free pages and on a hash queue of pages.

Swap-Use Table Entry

Reference count
Number of page table entries that point to a page on the swap device.

Page/storage unit number
Page identifier on storage unit.