UNIT - IV MEMORY MANAGEMENT

Points To Be Covered

- Introduction
- Memory Management Requirements
- Memory Partitioning: Fixed Partitioning, Dynamic Partitioning
- Buddy System
- Relocation
- Paging
- Segmentation
- Virtual Memory: Hardware and Control Structures
- Operating System Software.

Introduction [1]

Memory management is the process of allocating primary memory to user programs , reclaiming that memory when it is no longer needed protecting each user's memory area from other user programs.

Memory Management Requirements

Memory management is intended to satisfy the following requirements:

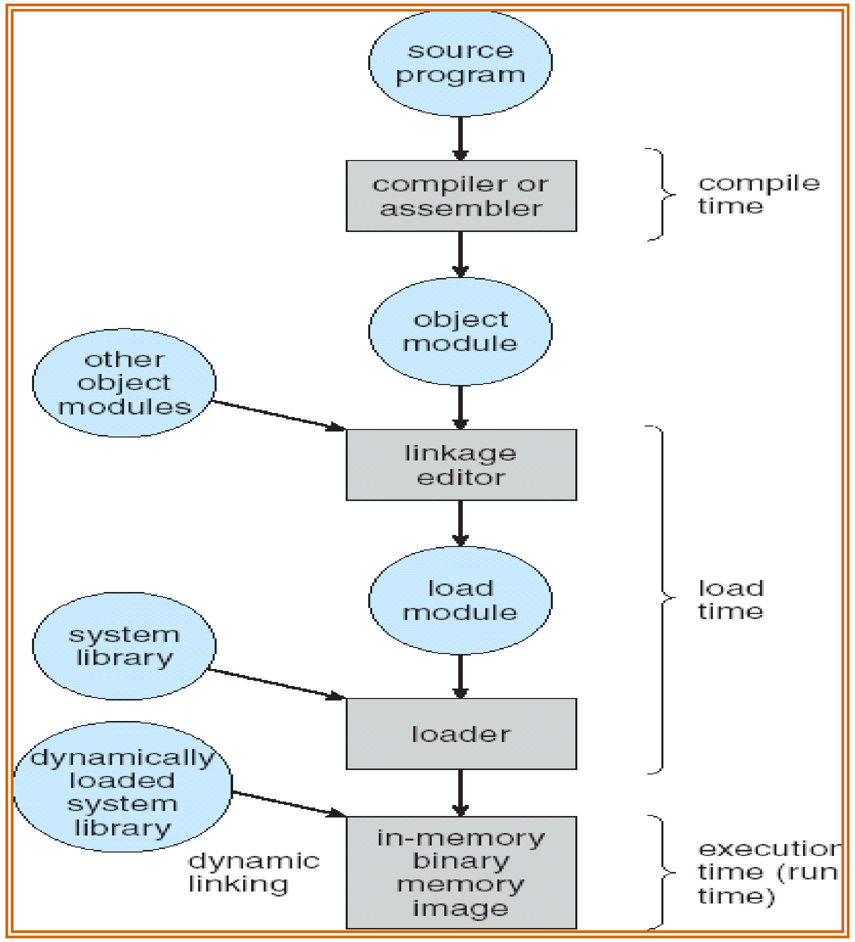
- Relocation
- Protection
- Sharing
- Logical organization
- Physical organization

Relocation Basics [2]

Address binding of instructions and data to memory addresses can happen at three different stages

- **Compile time**: If memory location known a priori, **absolute code** can be generated; must recompile code if starting location changes
- **Load time**: Must generate **relocatable code** if memory location is not known at compile time
- **Execution time**: Binding delayed until run time if the process can be moved during its execution from one memory segment to another. Need hardware support for address maps (e.g., base and limit registers)

Multistep Processing of a User Program [2]



Logical vs. Physical Address Space [2]

The concept of a logical address space that is bound to a separate **physical address space** is central to proper memory management

- **Logical address** – generated by the CPU; also referred to as **virtual address**
- **Physical address** – address seen by the memory unit

Logical and physical addresses are the same in compile-time and load-time address-binding schemes; logical (virtual) and physical addresses differ in execution-time address-binding scheme

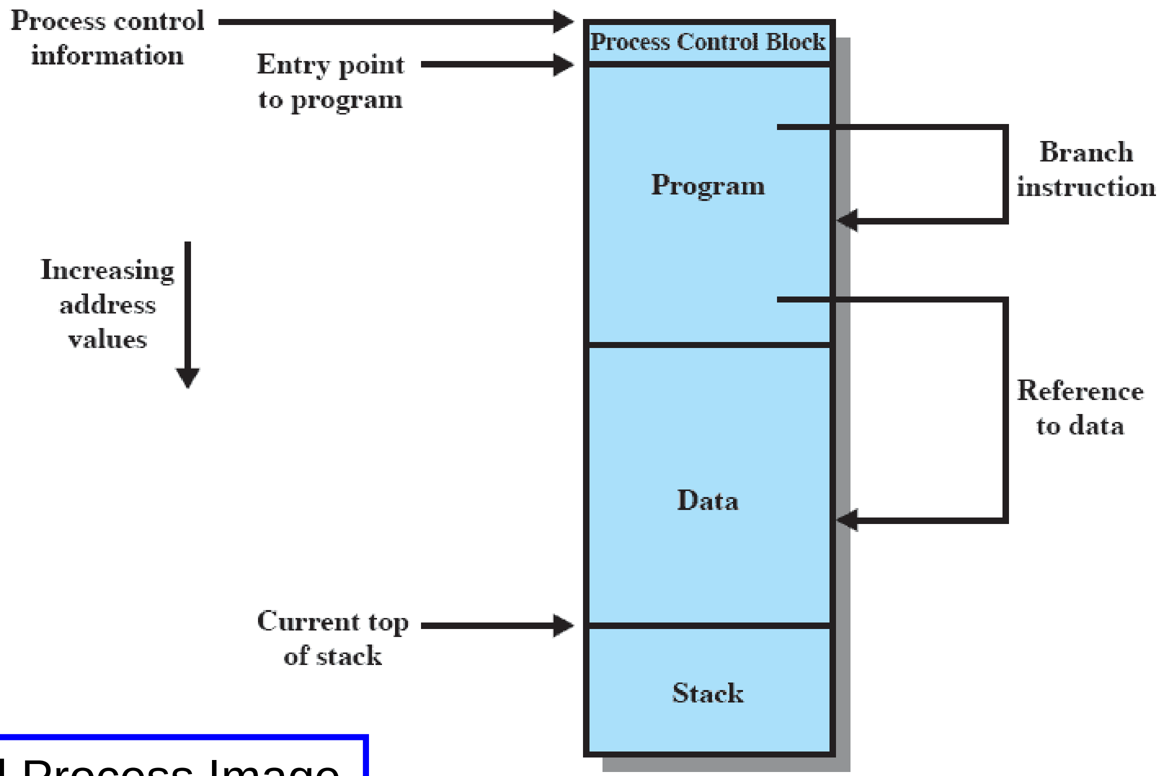
Relocation Definition and Addressing Requirements [1]

Relocation is the process of adjusting program addresses to match the actual physical addresses where the program resides when it executes

Why is relocation needed?

- Programmer/translator don’t know which other programs will be memory resident when the program executes

Addressing Requirements



Simplified Process Image

Figure 7.1 Addressing Requirements for a Process

Protection

Processes need to acquire permission to reference memory locations for reading or writing purposes

Location of a program in main memory is unpredictable

Memory references generated by a process must be checked at run time

Mechanisms that support relocation also support protection

Sharing

Advantageous to allow each process access to the same copy of the program rather than have their own separate copy

Memory management must allow controlled access to shared areas of memory without compromising protection

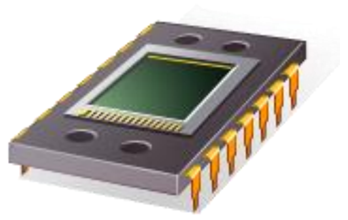
Mechanisms used to support relocation support sharing capabilities

Memory Partitioning

Virtual memory management brings processes into main memory for execution by the processor

- involves virtual memory
- based on segmentation and paging

- Partitioned memory management
  - used in several variations in some now-obsolete operating systems
  - does not involve virtual memory



Partitioning Types to be covered ….

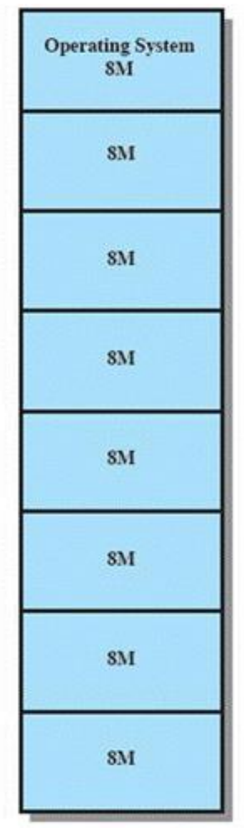
- Fixed Partitioning
- Dynamic Partitioning
- Simple Paging
- Simple Segmentation
- Virtual Memory Paging
- Virtual Memory Segmentation

Fixed Partitioning

Equal-size partitions

- any process whose size is less than or equal to the partition size can be loaded into an available partition

The operating system can swap out a process if all partitions are full and no process is in the Ready or Running state



(a) Equal-size partitions

Disadvantages

A program may be too big to fit in a partition

- program needs to be designed with the use of overlays

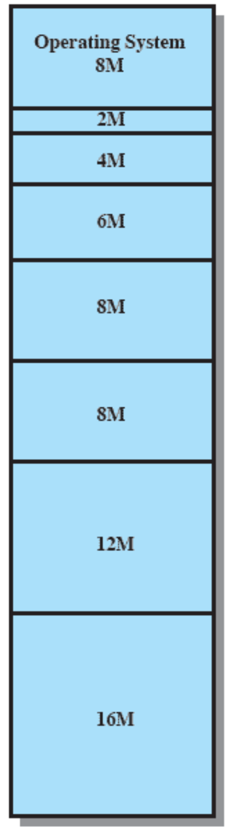
Main memory utilization is inefficient

- any program, regardless of size, occupies an entire partition
- ***internal fragmentation***
  - wasted space due to the block of data loaded being smaller than the partition

Unequal Size Partitions

Using unequal size partitions helps lessen the problems

- programs up to 16M can be accommodated without overlays
- partitions smaller than 8M allow smaller programs to be accommodated with less internal fragmentation



(b) Unequal-size partitions

Memory Assignment

Fixed Partitioning

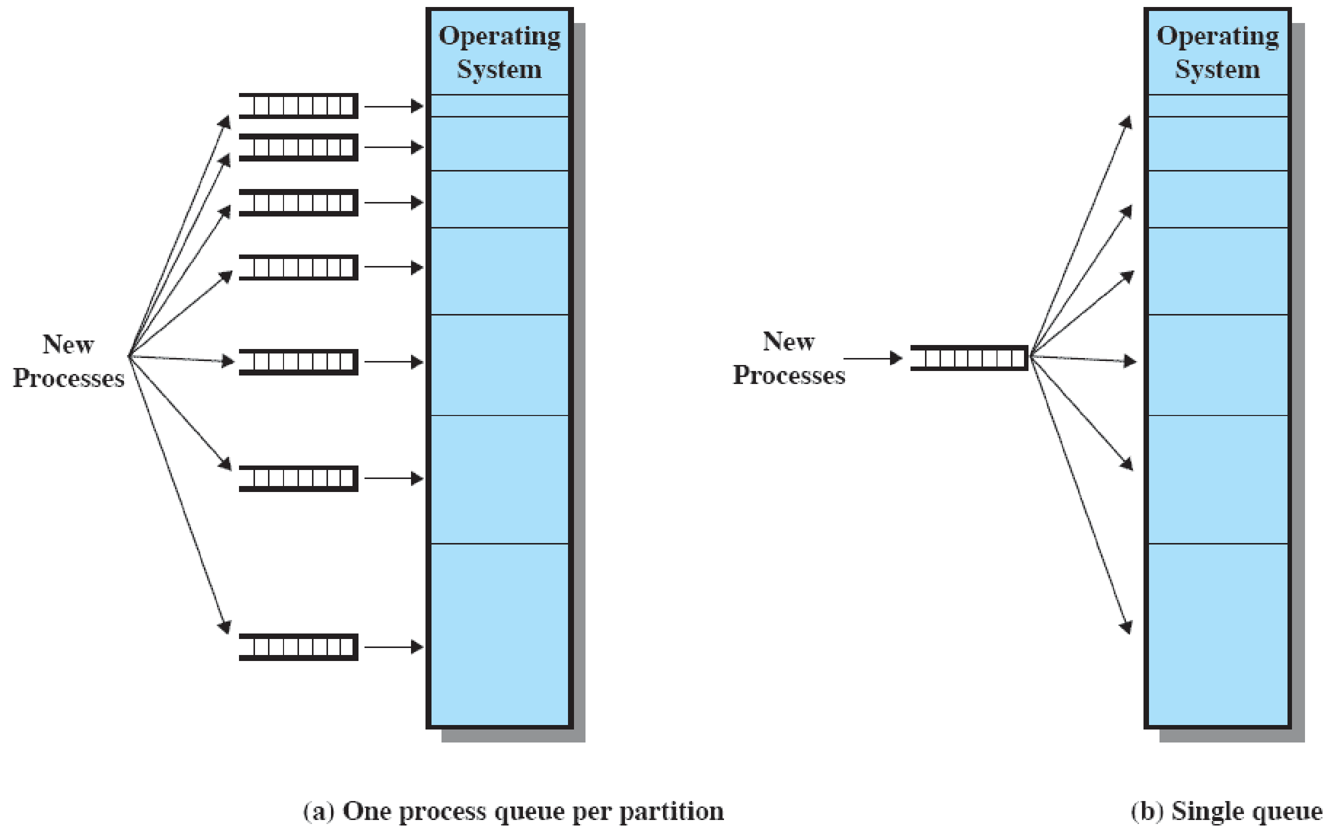


Figure 7.3 Memory Assignment for Fixed Partitioning

Disadvantages

The number of partitions specified at system generation time limits the number of active processes in the system

Small jobs will not utilize partition space efficiently

Dynamic Partitioning

Partitions are of variable length and number

Process is allocated exactly as much memory as it requires

This technique was used by IBM’s mainframe operating system, OS/MVT

Effect of Dynamic Partitioning

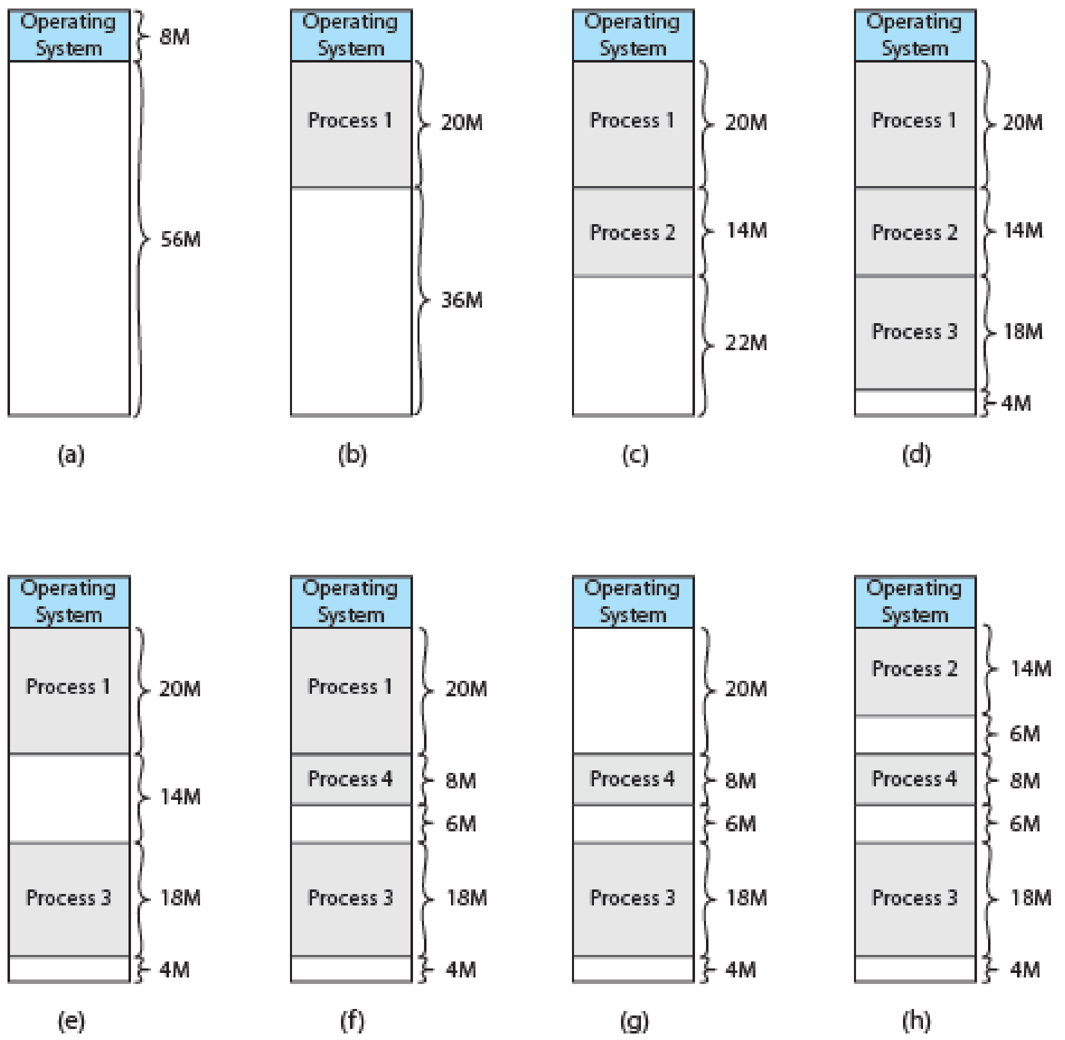


Figure 7.4 The Effect of Dynamic Partitioning

Dynamic Partitioning

External Fragmentation

- memory becomes more and more fragmented
- memory utilization declines

Compaction

- technique for overcoming external fragmentation
- OS shifts processes so that they are contiguous
- free memory is together in one block
- time consuming and wastes CPU time

Placement Algorithms

Best-fit	First-fit	Next-fit
• chooses the block that is closest in size to the request	• begins to scan memory from the beginning and chooses the first available block that is large enough	• begins to scan memory from the location of the last placement and chooses the next available block that is large enough

Memory Configuration Example

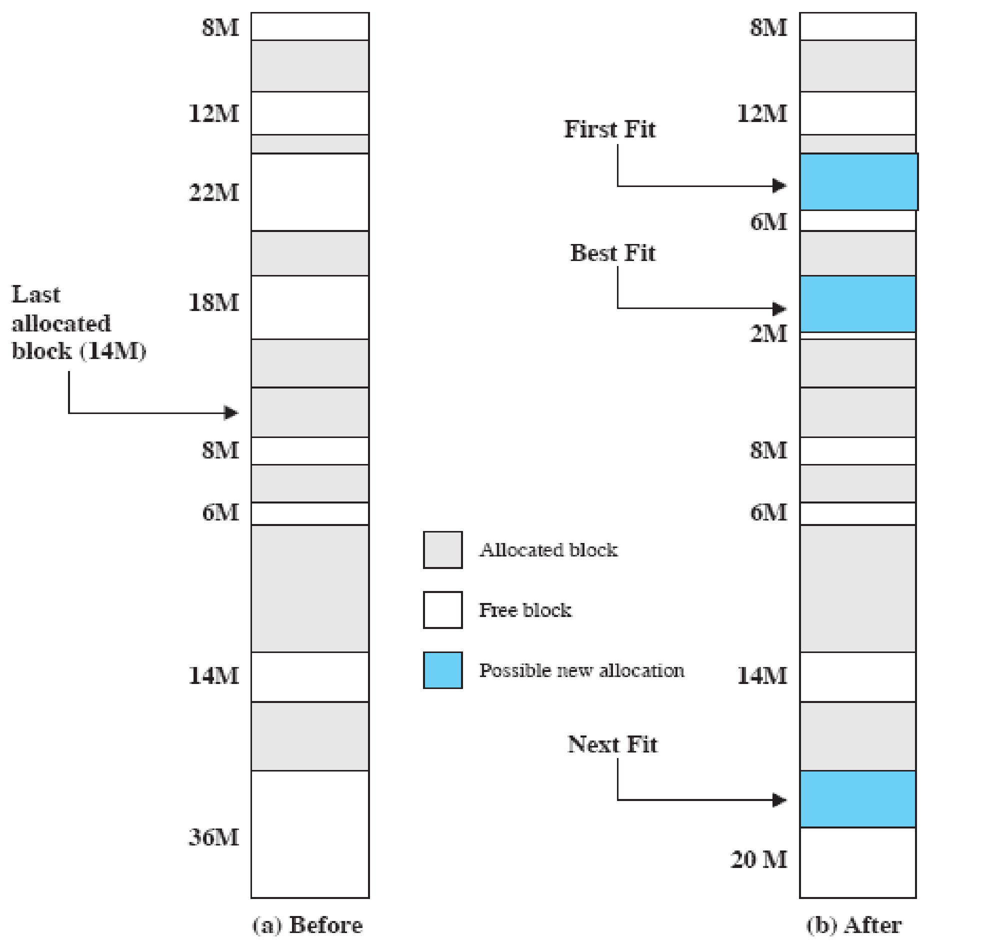


Figure 7.5 Example Memory Configuration before and after Allocation of 16-Mbyte Block

Buddy System

Comprised of fixed and dynamic partitioning schemes

Space available for allocation is treated as a single block

Memory blocks are available of size 2^K *words,* $L \leq K \leq U$, where

- 2^L = *smallest size block that is allocated*
- 2^U = largest size block that is allocated; generally 2^U is the size of the entire memory available for allocation

Buddy System Example

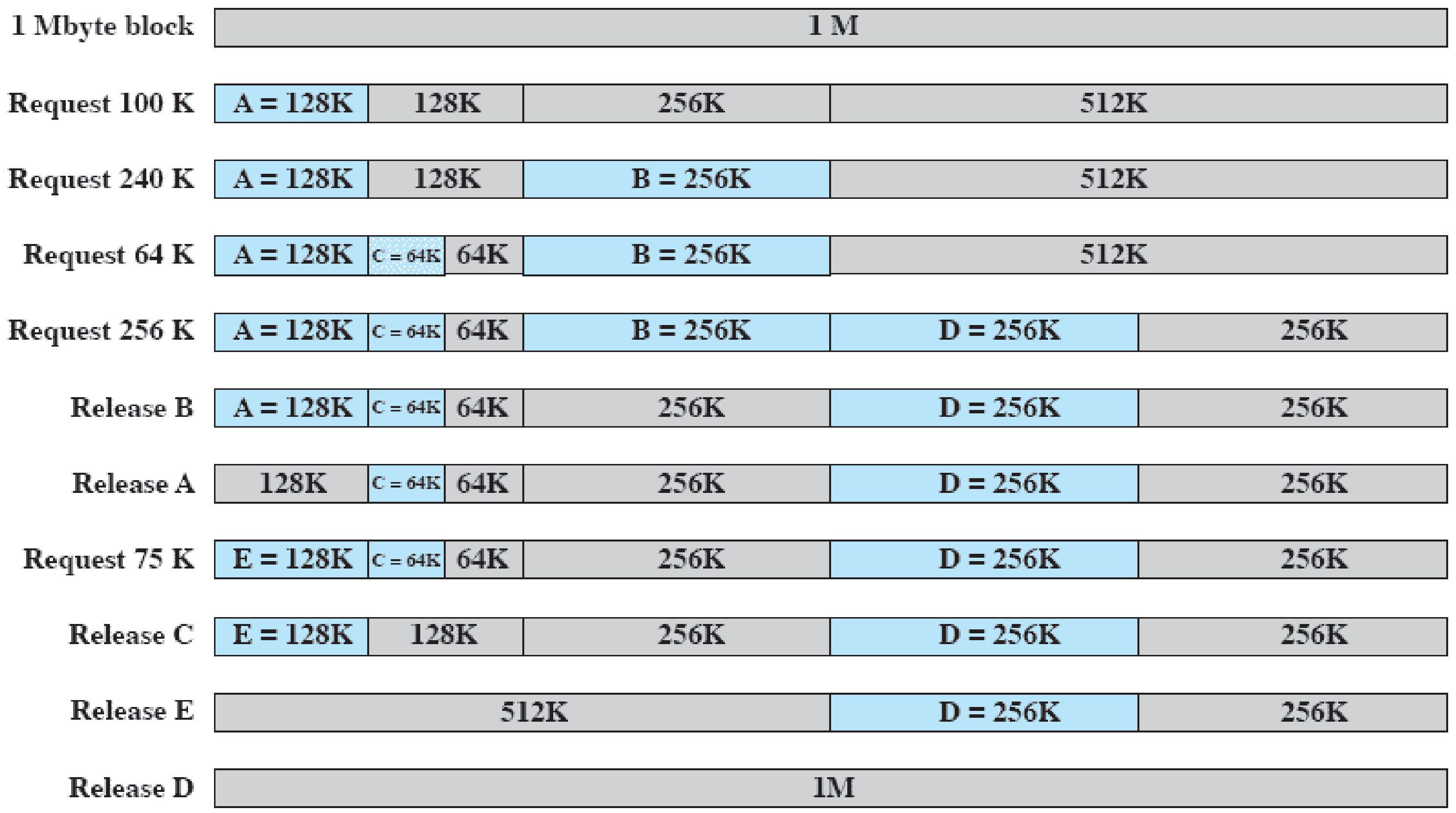

1 Mbyte block | 1 M
Request 100 K | A = 128K | 128K | 256K | 512K
Request 240 K | A = 128K | 128K | B = 256K | 512K
Request 64 K | A = 128K | C = 64K | 64K | B = 256K | 512K
Request 256 K | A = 128K | C = 64K | 64K | B = 256K | D = 256K | 256K
Release B | A = 128K | C = 64K | 64K | 256K | D = 256K | 256K
Release A | 128K | C = 64K | 64K | 256K | D = 256K | 256K
Request 75 K | E = 128K | C = 64K | 64K | 256K | D = 256K | 256K
Release C | E = 128K | 128K | 256K | D = 256K | 256K
Release E | 512K | D = 256K | 256K
Release D | 1M

Figure 7.6 Example of Buddy System

Tree Represen ntation

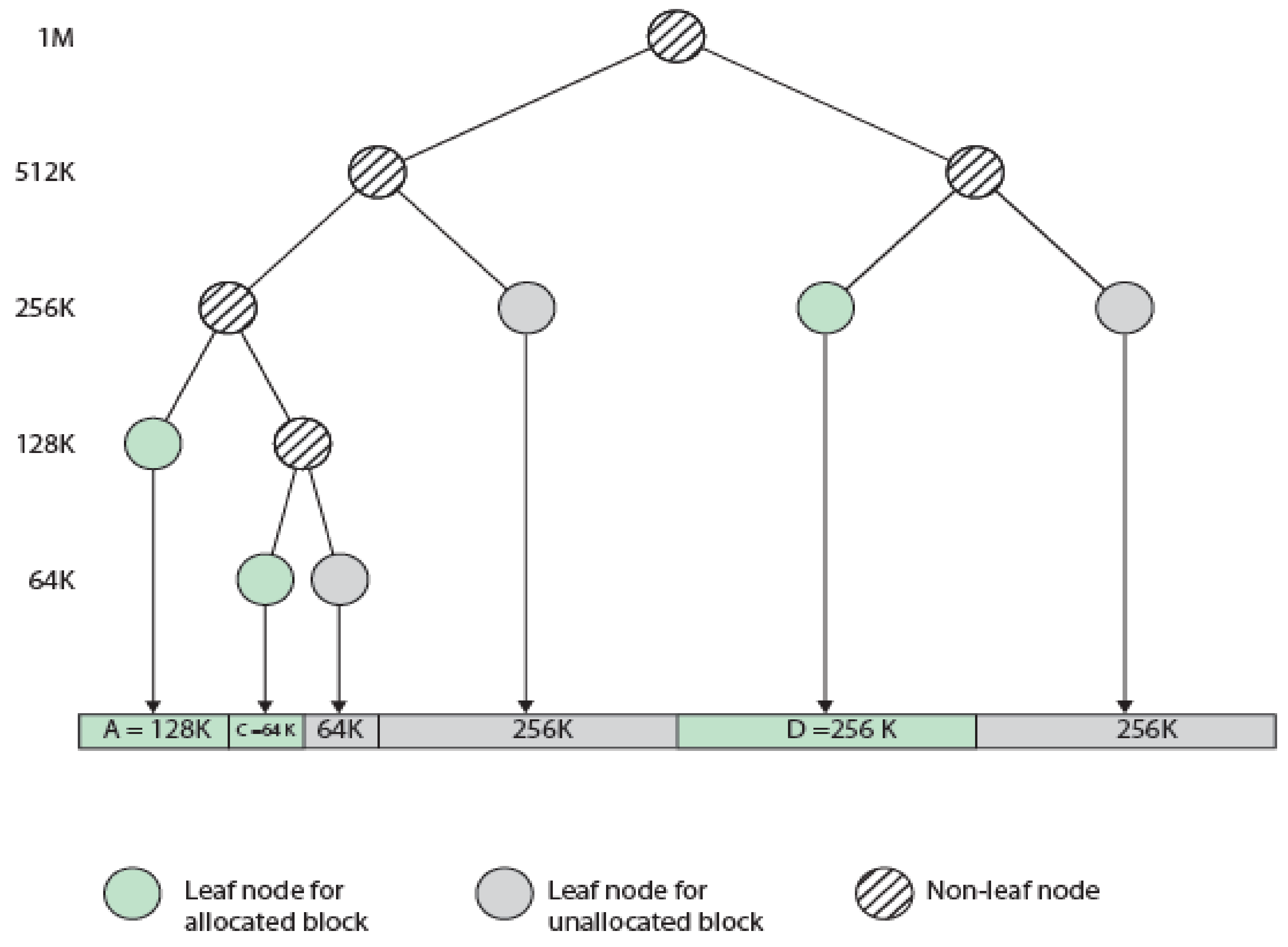


Figure 7.7 Tree Representation of Buddy System

Relocation

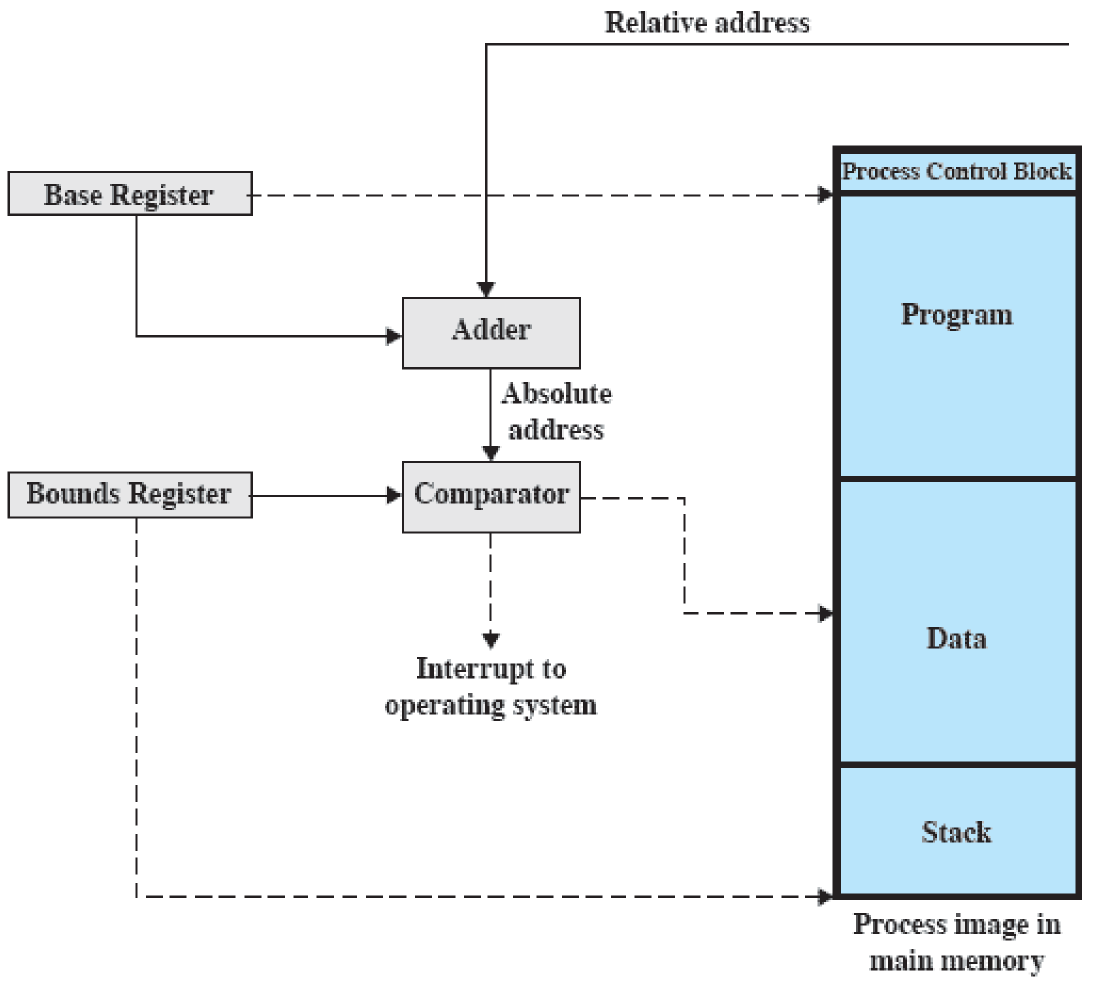


Figure 7.8 Hardware Support for Relocation

Paging

Partition memory into equal fixed-size chunks that are relatively small

Process is also divided into small fixed-size chunks of the same size

Pages

- chunks of a process

Frames

- available chunks of memory

Assignment of Process to Free Frames

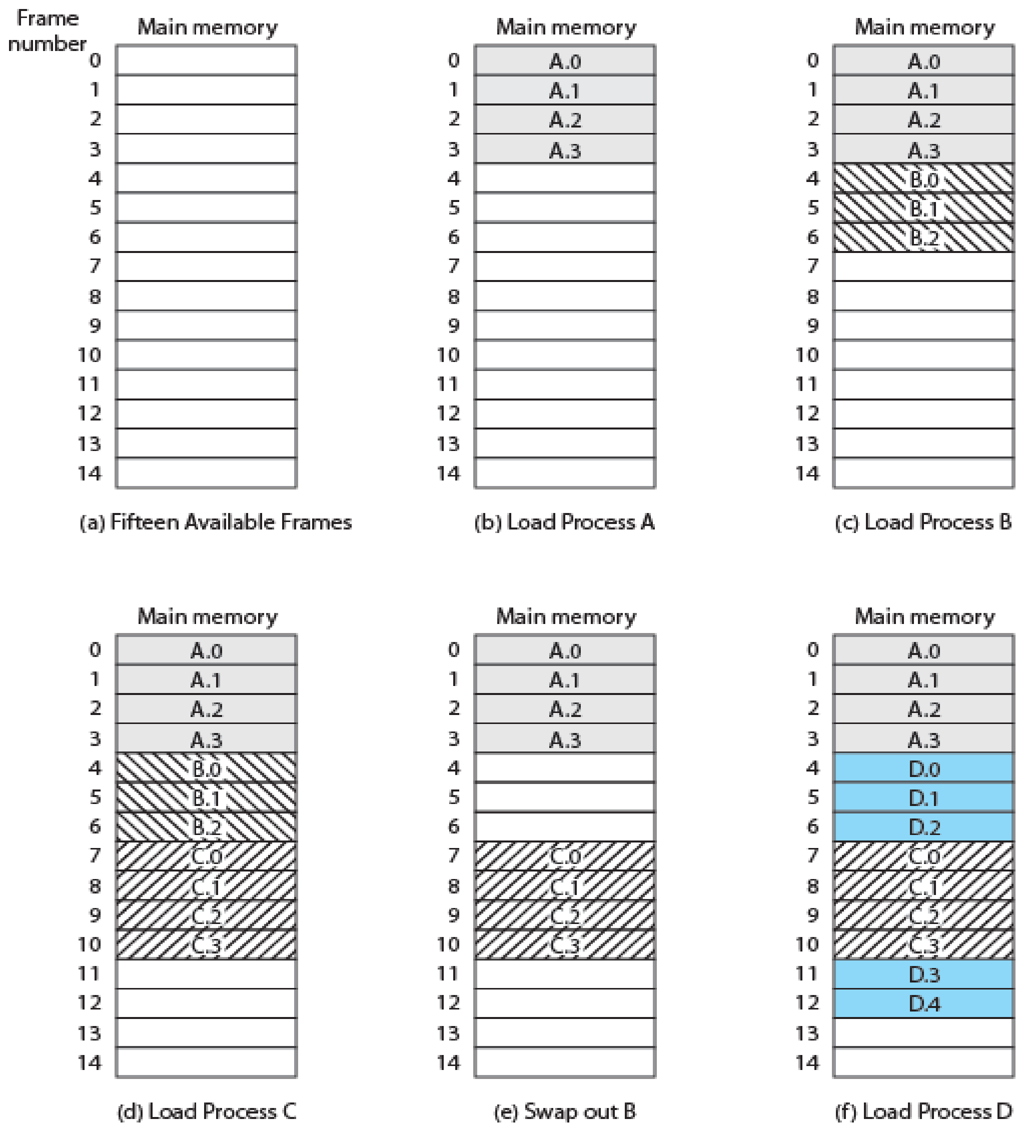


Figure 7.9 Assignment of Process Pages to Free Frames

Page Table

Maintained by operating system for each process

Contains the frame location for each page in the process

Processor must know how to access the page table for the current process

Used by processor to produce a physical address

Data Structures

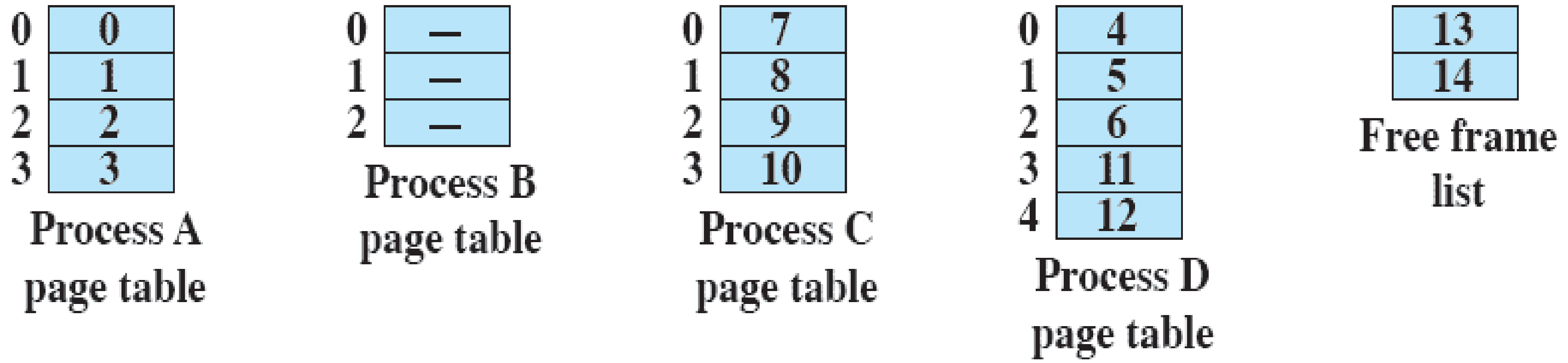


Figure 7.10 Data Structures for the Example of Figure 7.9 at Time Epoch (f)

Logical Addresses

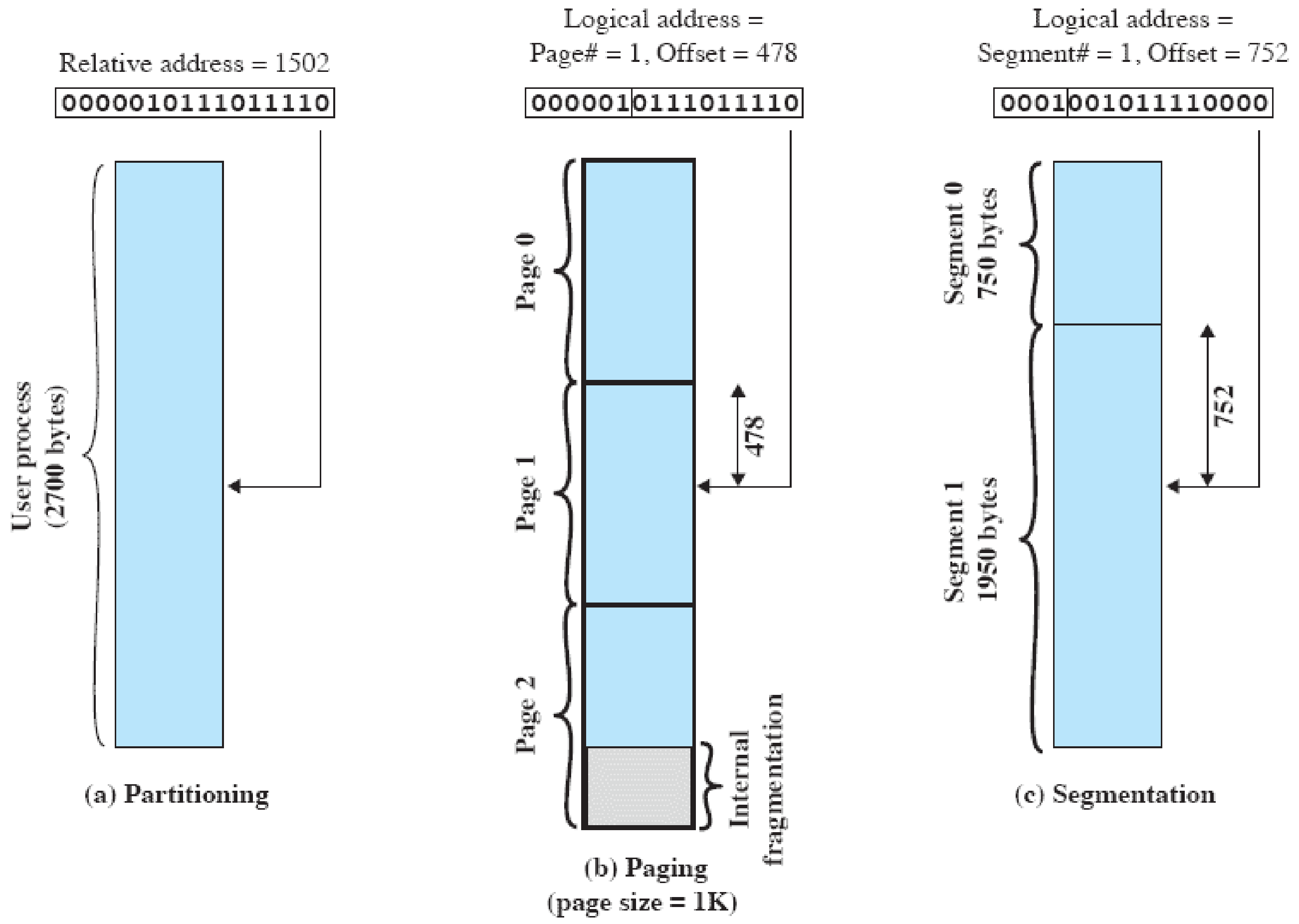
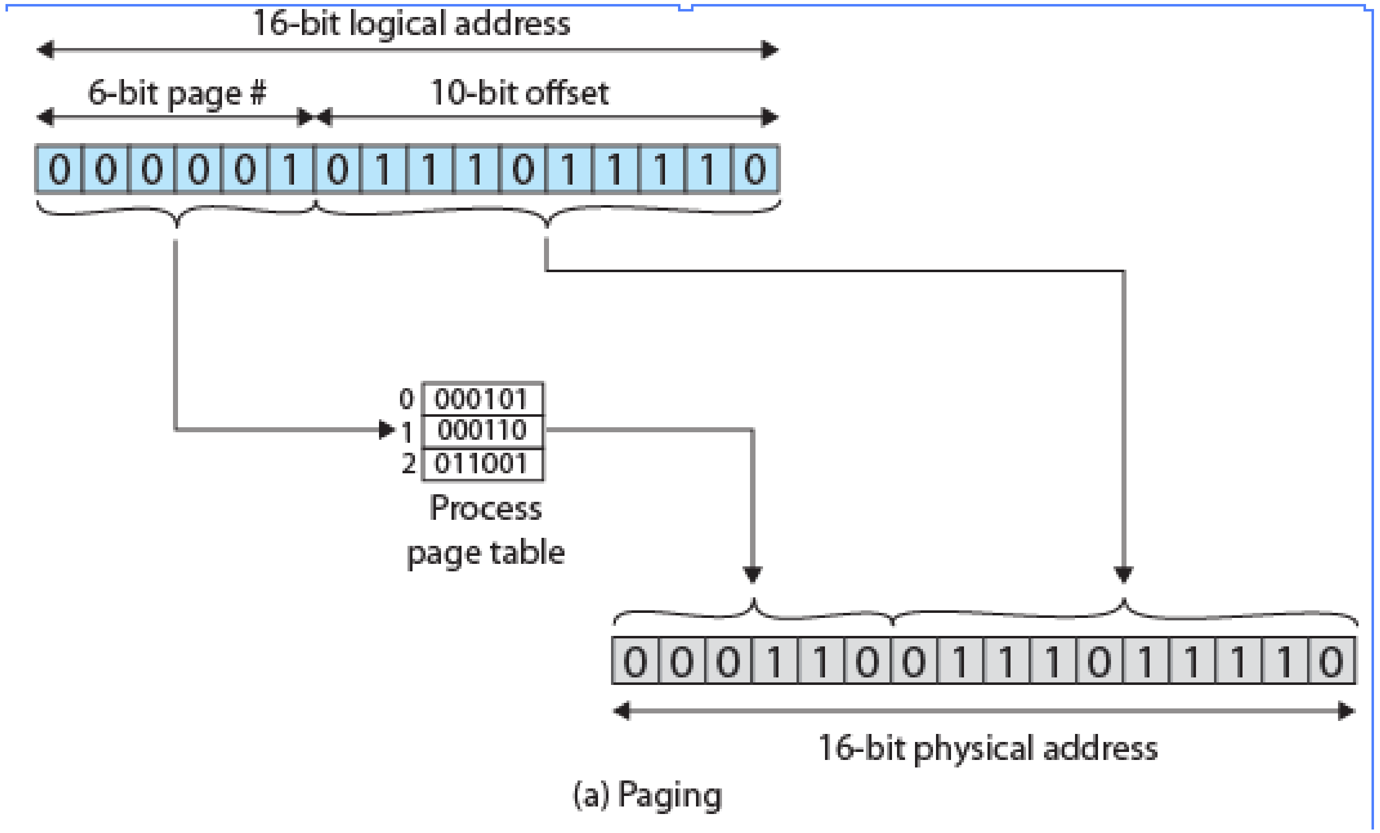


Figure 7.11 Logical Addresses

Logical-to-Physical Address Translation - Paging



(a) Paging

Segmentation

A program can be subdivided into segments

- may vary in length
- there is a maximum length

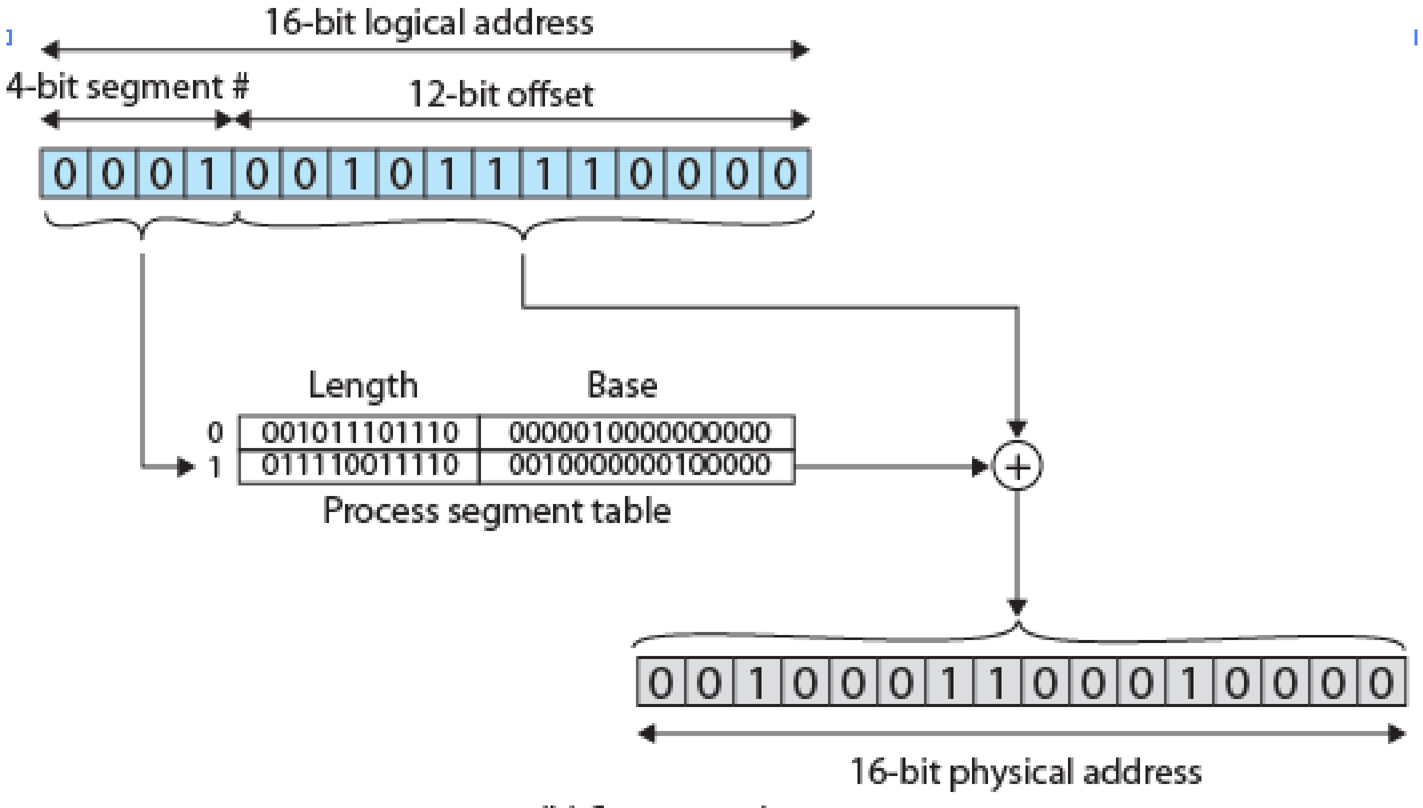
Addressing consists of two parts:

- segment number
- an offset

Similar to dynamic partitioning

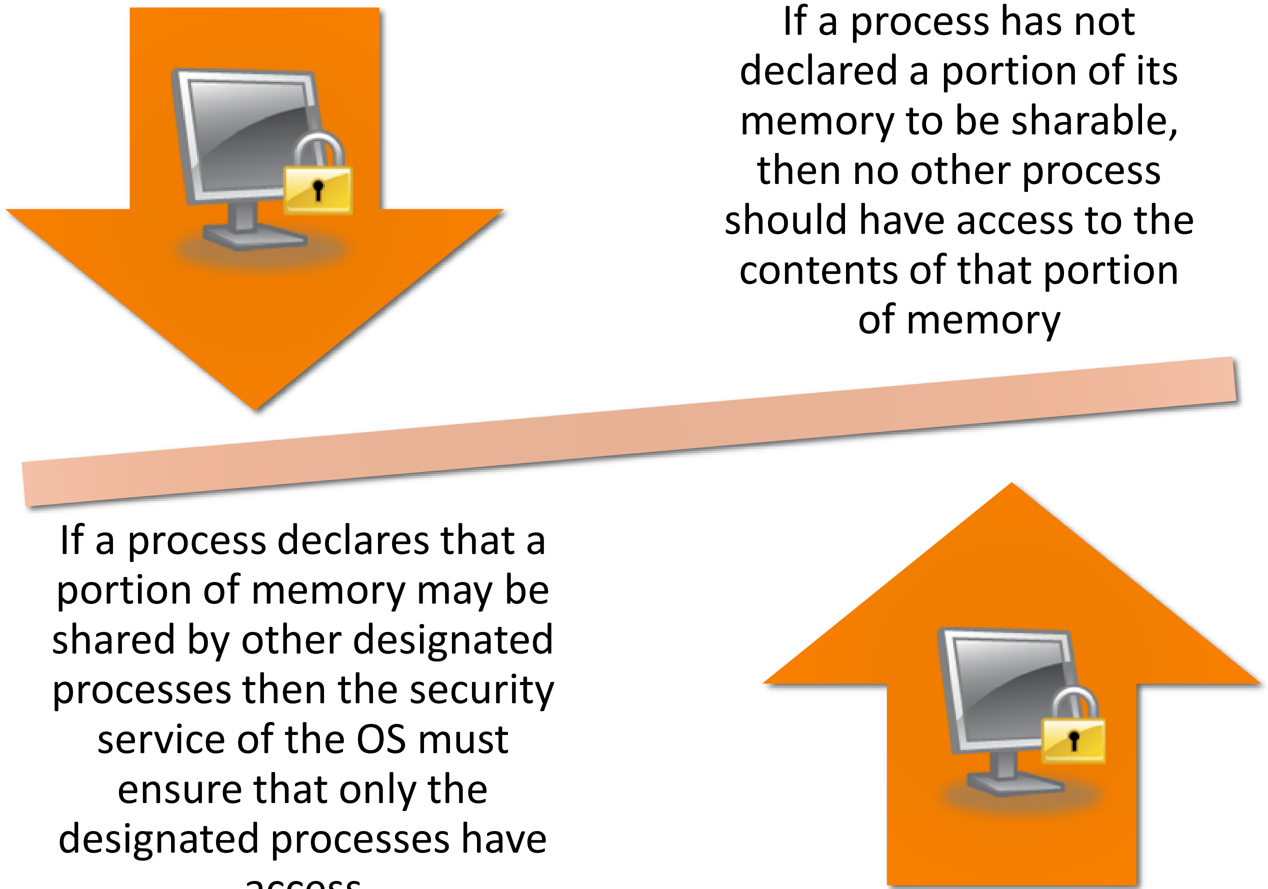
Eliminates internal fragmentation

Logical-to-Physical Address Translation - Segmentation



(b) Segmentation

Security Issues



If a process has not declared a portion of its memory to be sharable, then no other process should have access to the contents of that portion of memory

If a process declares that a portion of memory may be shared by other designated processes then the security service of the OS must ensure that only the designated processes have access

References

[1] William Stallings, Operating System: Internals and Design Principles, Prentice Hall, 8th Edition,2014, ISBN-10: 0133805913

[2] Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts, John Wiley & Sons ,Inc., 9th Edition,2012, ISBN 978-1-118-06333-0