TI III: Operating & Communication Systems

Memory

Paging & Segmentation,

Virtual Memory,

Swap Policies

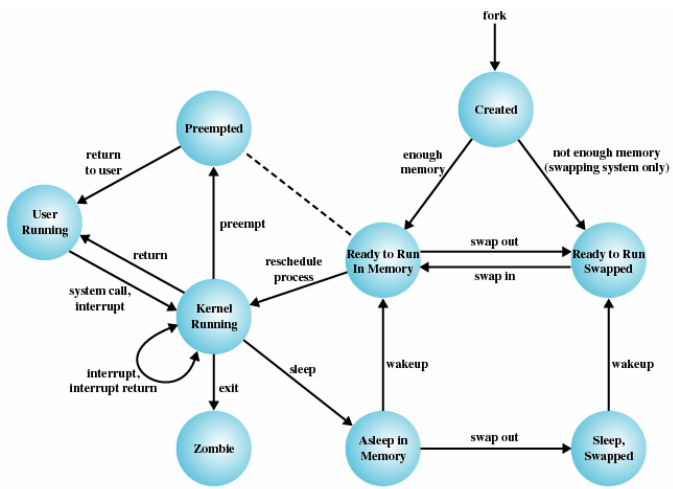


Figure 3.17 UNIX Process State Transition Diagram

Content (1)

1. Introduction and Motivation
   - Tasks
   - Services
   - Virtual Resources
   - Historical Perspective
   - Examples
   - Tools
2. Subsystems, Interrupts and System Calls
   - System Structure
   - Flow of Control
   - System Library
   - POSIX
3. Processes
   1. Definition
   2. Implementation
   3. State Model
4. **Memory**
   - Paging & Segmentation
   - Virtual Memory
   - Swap Policies
5. Scheduling
6. I/O and File System
7. Booting and System Services

Memory Management

- Goals
  - Subdividing memory to accommodate multiple processes
  - Memory needs to be allocated to ensure a reasonable supply of ready processes to consume available processor time
- Requirements
  - Relocation: Location in memory unknown or may change
  - Protection: Disallow access to memory of other processes
  - Sharing: Data for communication, program copy for memory reduction
  - Logical Organization: Support modular programming
  - Physical Organization: Support overlaying to work around insufficient memory

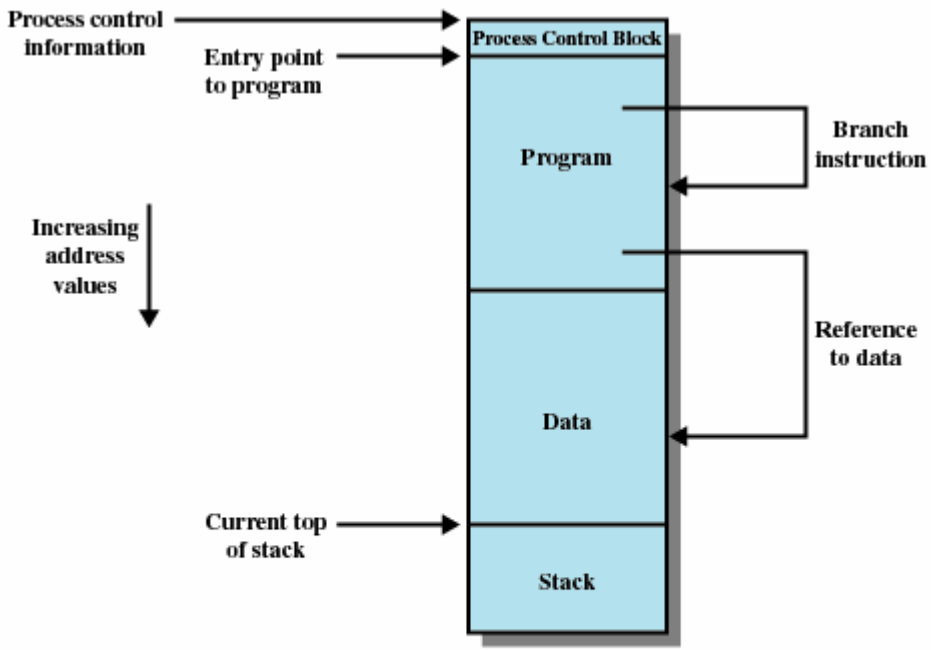


Figure 7.1 Addressing Requirements for a Process

Fixed Memory Partitioning

- Equal-size partitions
- Unequal-size partitions

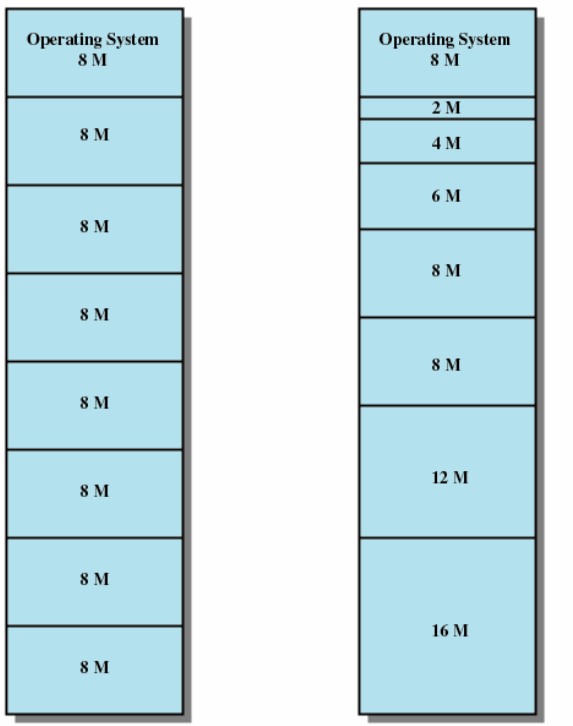


Figure 7.2 Example of Fixed Partitioning of a 64-Mbyte Memory

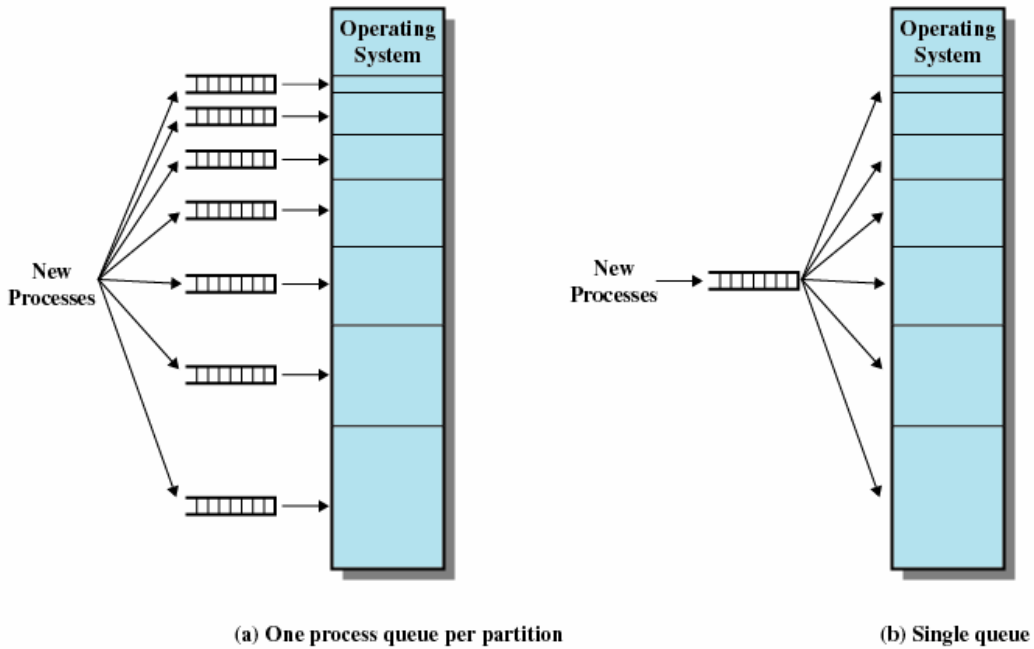


Figure 7.3 Memory Assignment for Fixed Partitioning

Dynamic Memory Partitioning

- Variable length partitions
- Processes allocate memory

- External fragmentation
- Requires compaction

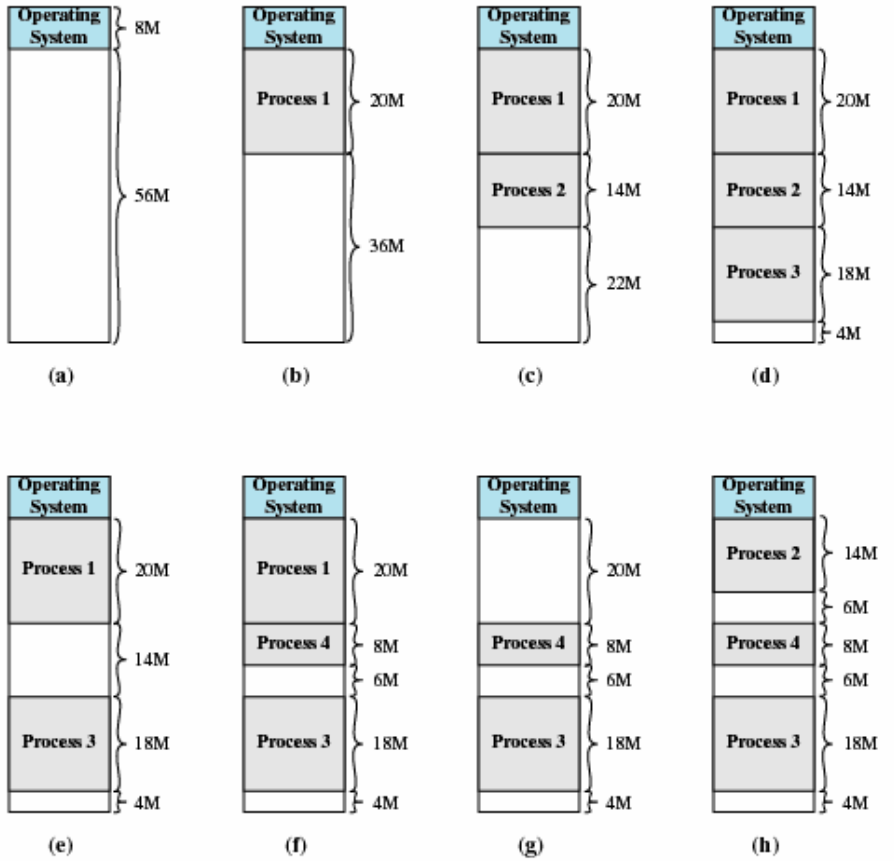


Figure 7.4 The Effect of Dynamic Partitioning

Dynamic Placement Algorithms

- First-fit algorithm:
  - Scans memory from the beginning
  - Chooses first available block that is large enough
- Next-fit algorithm:
  - Scans memory from the location of the last placement
  - More often allocate a block of memory at the end of memory where the largest block is found

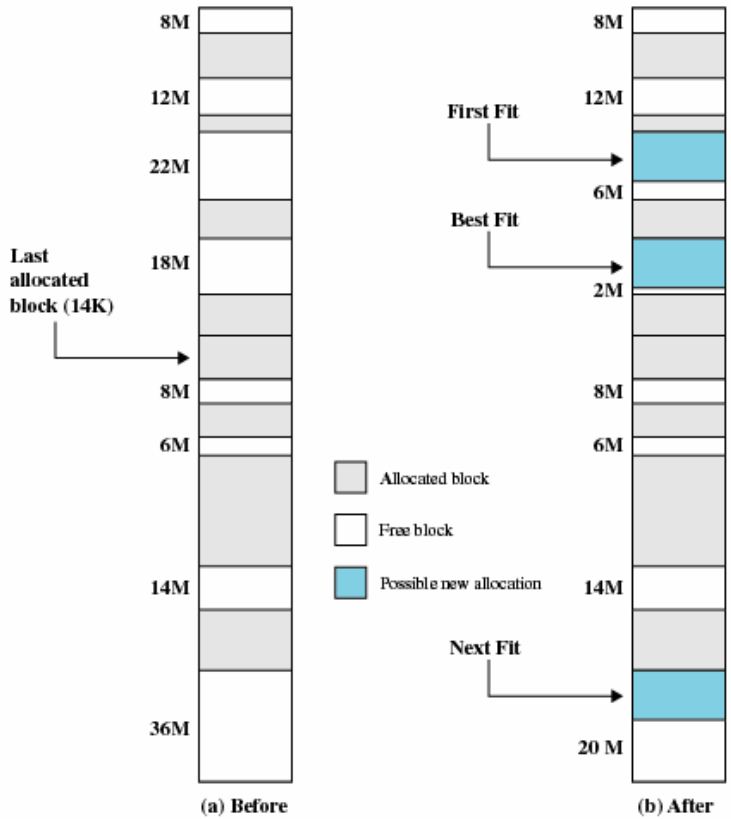


Figure 7.5 Example Memory Configuration Before and After Allocation of 16 Mbyte Block

Buddy System

- Entire space available is treated as a single block of 2^U
- If a request of size s such that $2^{U-1} < s <= 2^U$, entire block is allocated
  - Otherwise block is split into two equal buddies
  - Process continues until smallest block greater than or equal to s is generated

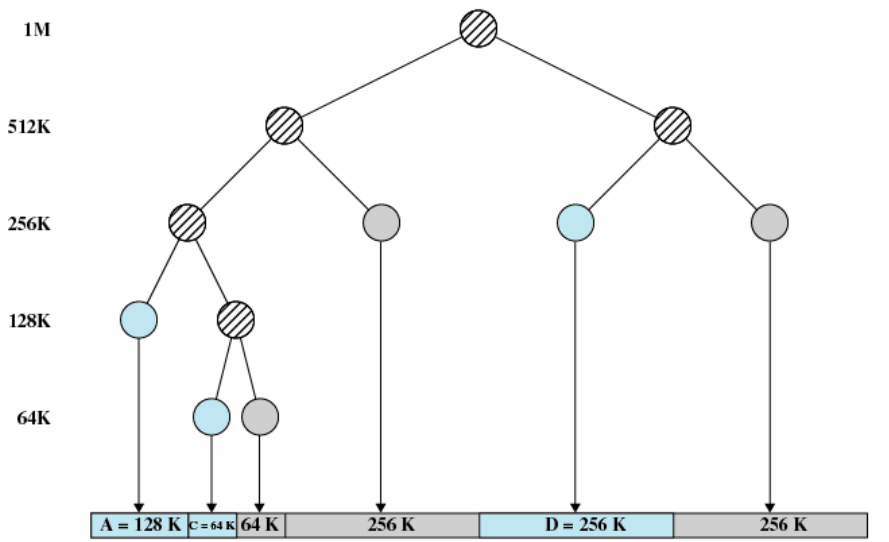


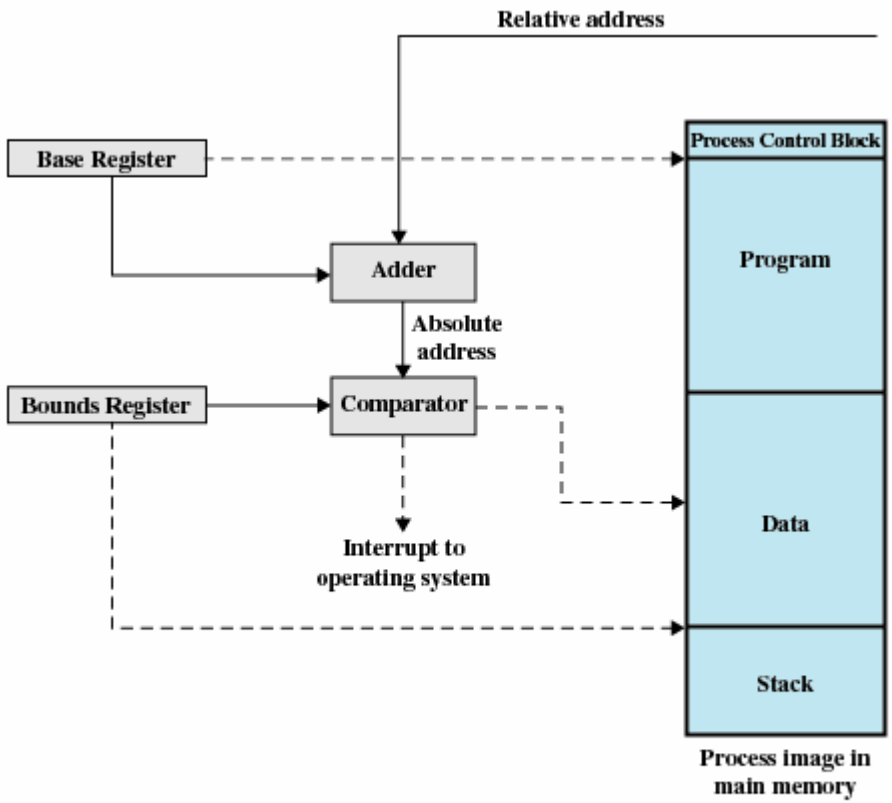
Figure 7.7 Tree Representation of Buddy System

Addresses

- Logical
  - Reference to a memory location independent of the current assignment of data to memory
  - Translation must be made to the physical address
- Relative
  - Address expressed as a location relative to some known point
- Physical
  - The absolute address or actual location in main memory

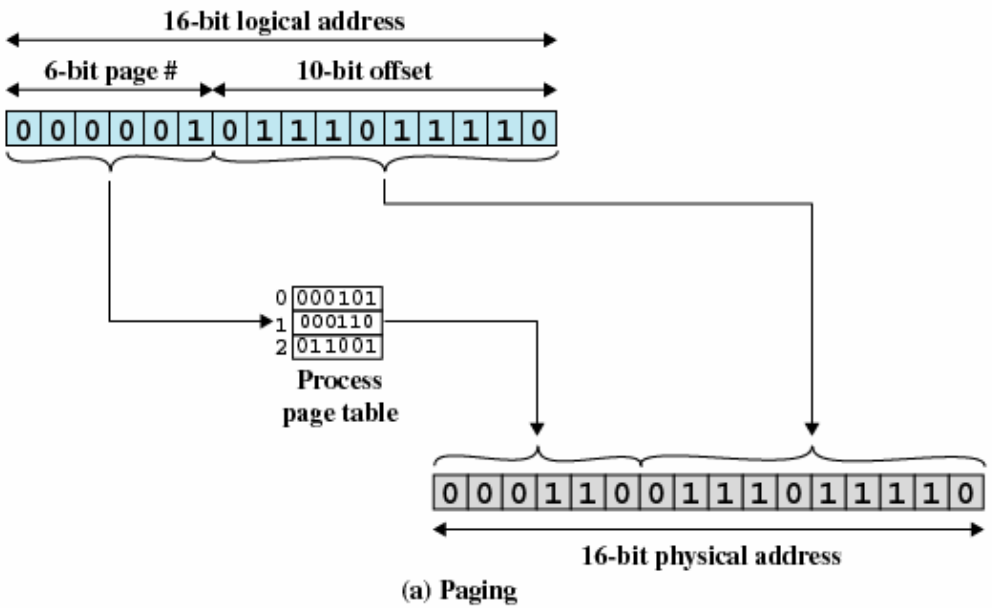
Hardware Support

- Base register (starting address for the process)
- Bounds register (ending location of the process)
- Registers are set when process is loaded or swapped in
- Interrupt (SIGSEGV) if calculated address is out of bounds



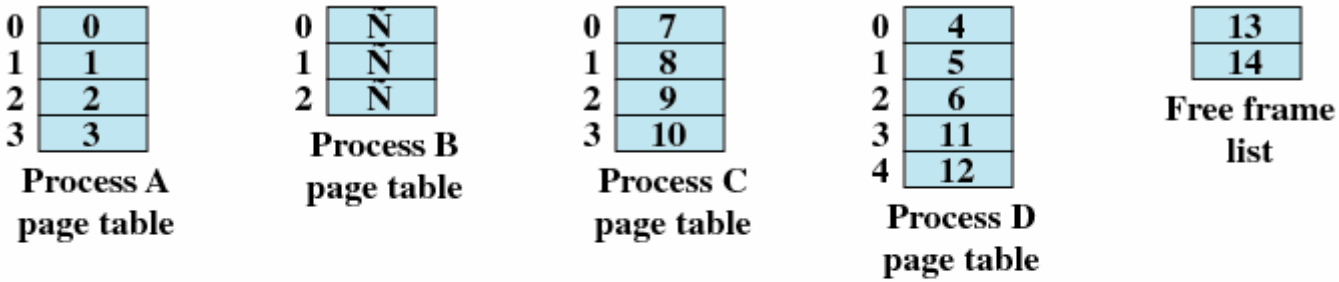
Paging

- Partition memory into small equal fixed-size chunks (“frames”) and divide each process into the same size chunks (“pages”).
- Operating system maintains page table for each process
  - Contains frame location for each page
- Memory address consist of a page number and offset within the page



(a) Paging

Assignment of Process Pages to Memory Frames



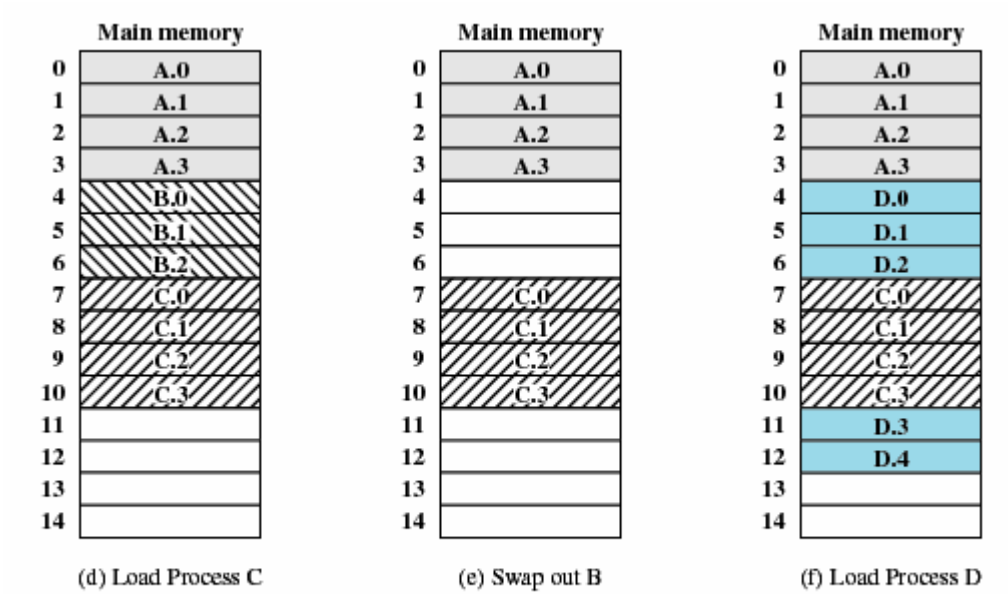


Figure 7.9 Assignment of Process Pages to Free Frames

Paging Address Translation

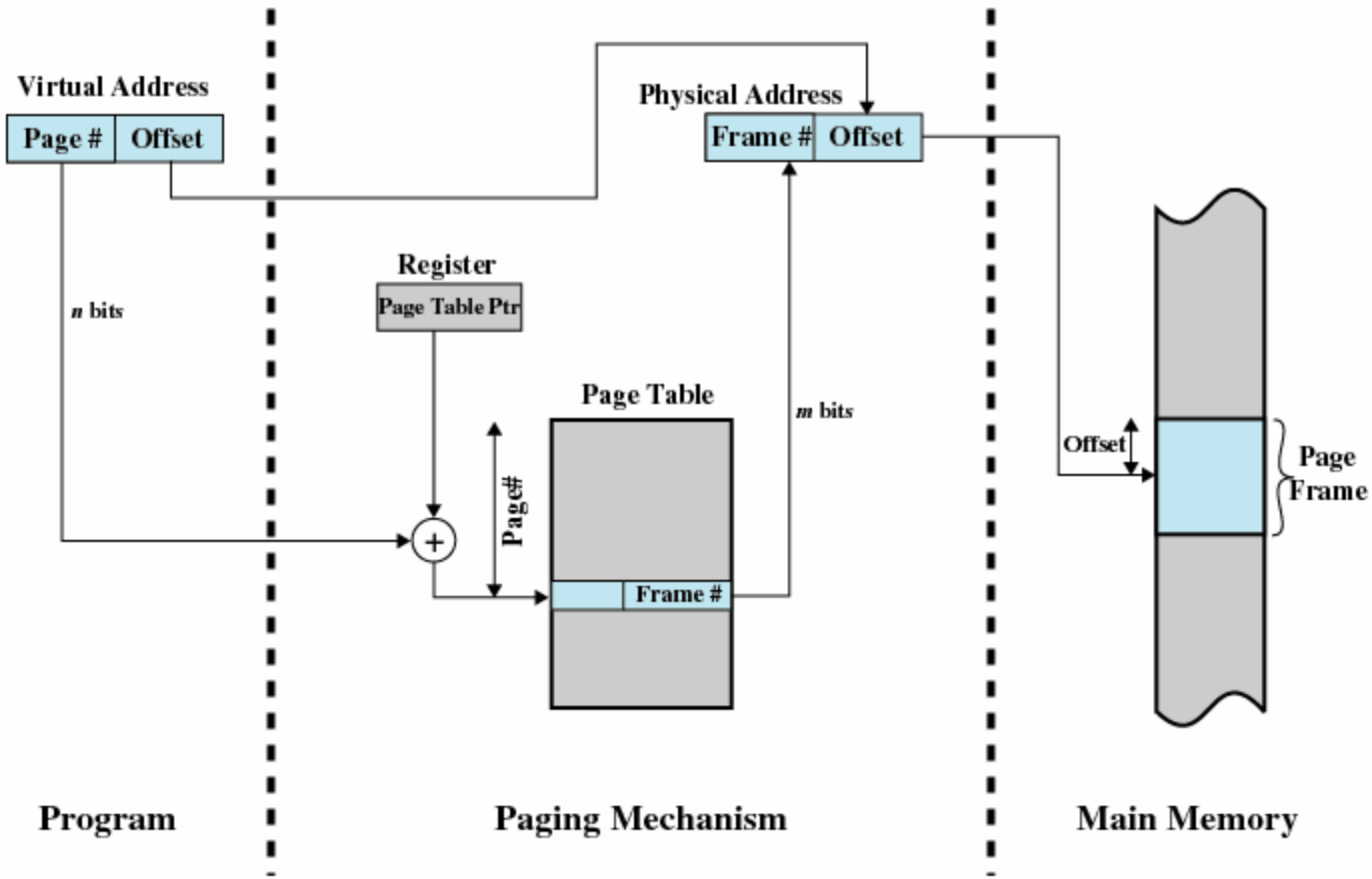


Figure 8.3 Address Translation in a Paging System

Segmentation

- All segments of all programs do not have to be of same length (compare w/ fixed size pages)
- Addressing consist of two parts - a segment number and an offset
- Since segments are not equal, segmentation is similar to dynamic partitioning

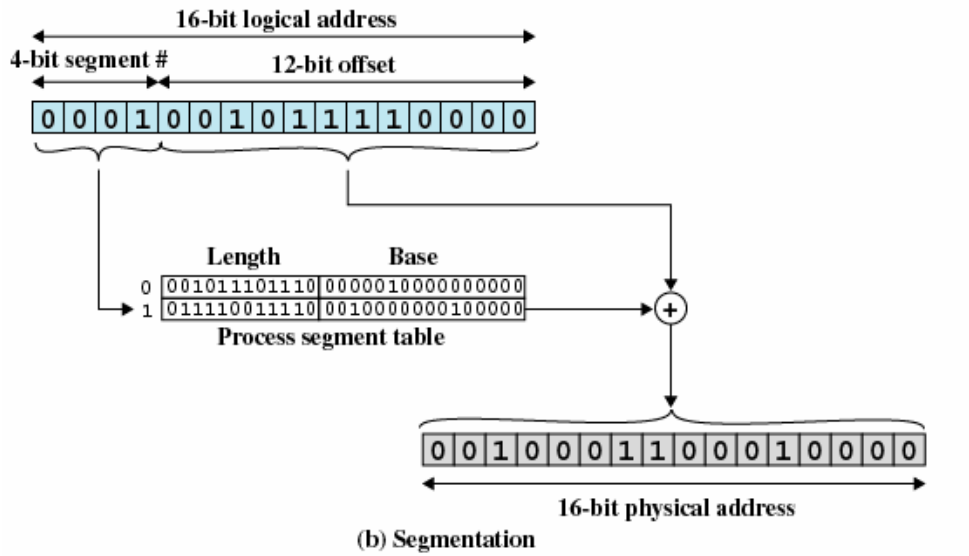


Figure 7.12 Examples of Logical-to-Physical Address Translation

Segmentation Address Translation

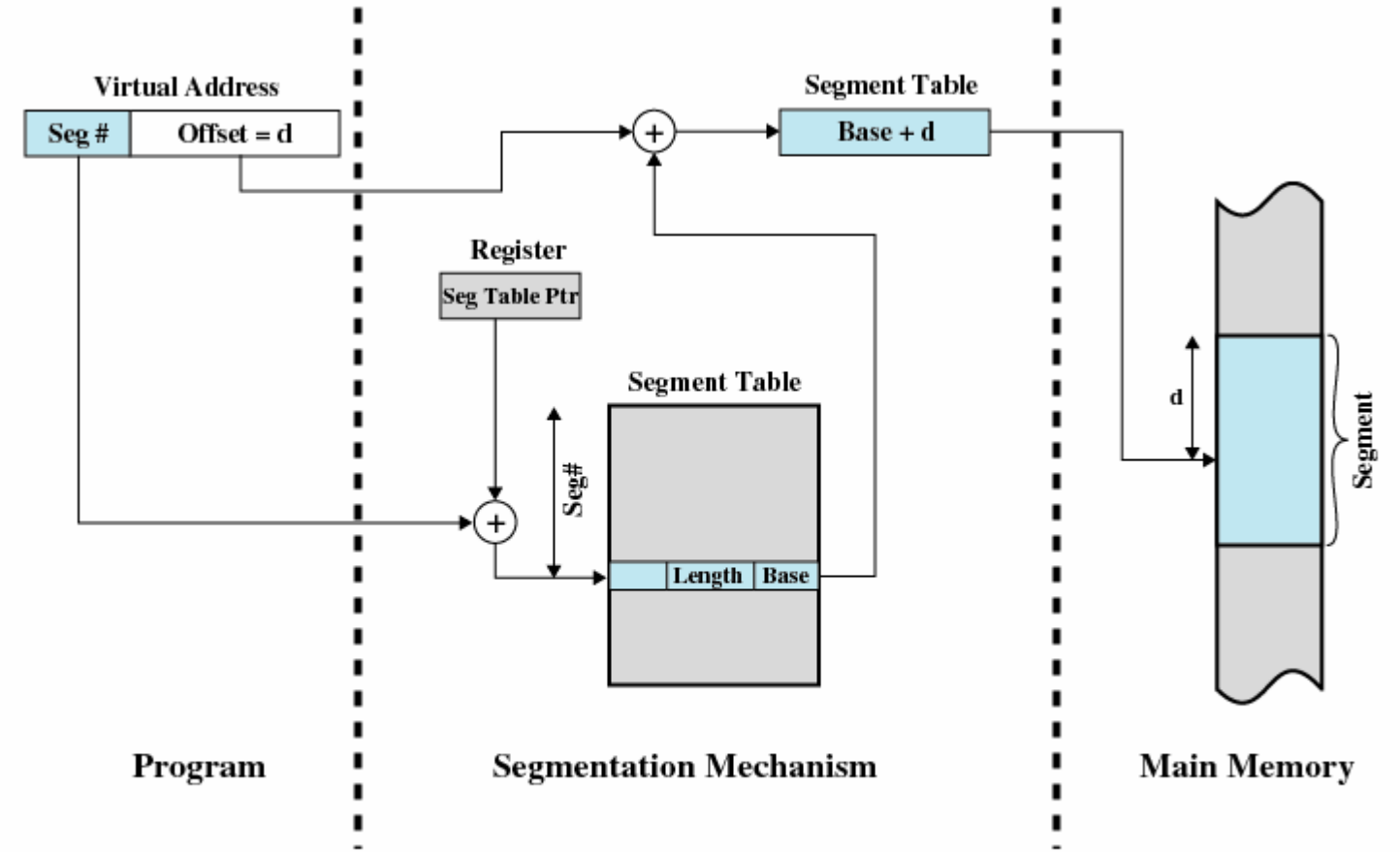


Figure 8.12 Address Translation in a Segmentation System

Comparison

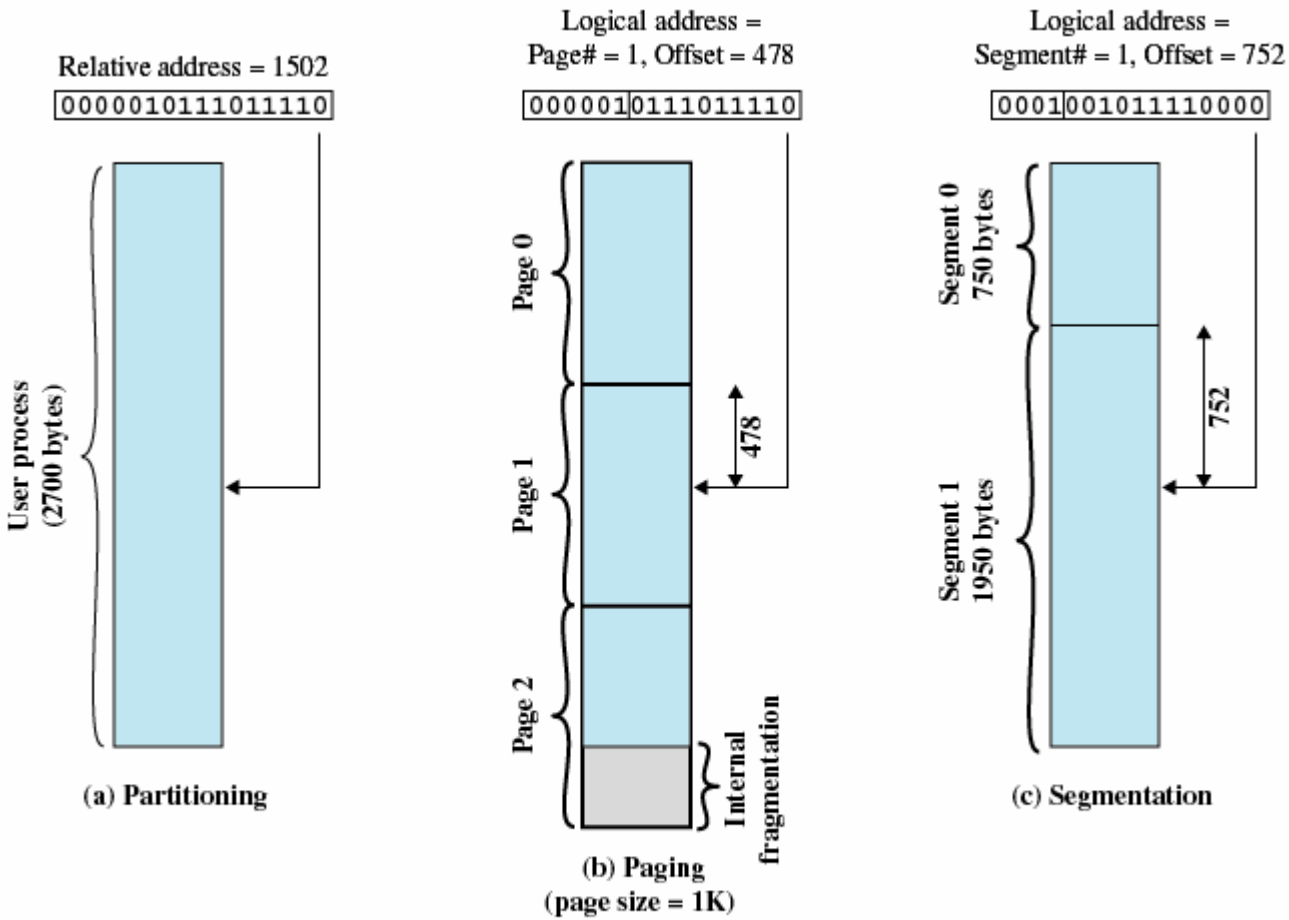


Figure 7.11 Logical Addresses

Execution of a Program

- Operating system brings into main memory a few pieces of the program
- Resident set: portion of process that is in main memory
- Interrupt is generated when an address is needed that is not in main memory
- Operating system places the process in a blocking state
- Piece of process that contains the logical address is brought into main memory
  - Operating system issues a disk I/O Read request
  - Another process is dispatched to run while the disk I/O takes place
  - An interrupt is issued when disk I/O complete which causes the operating system to place the affected process in the Ready state

Real / Virtual Memory

- Real memory: main memory
- Virtual memory: memory on secondary storage / disk
  - Allows for effective multiprogramming
  - Relieves the user of tight constraints of main memory

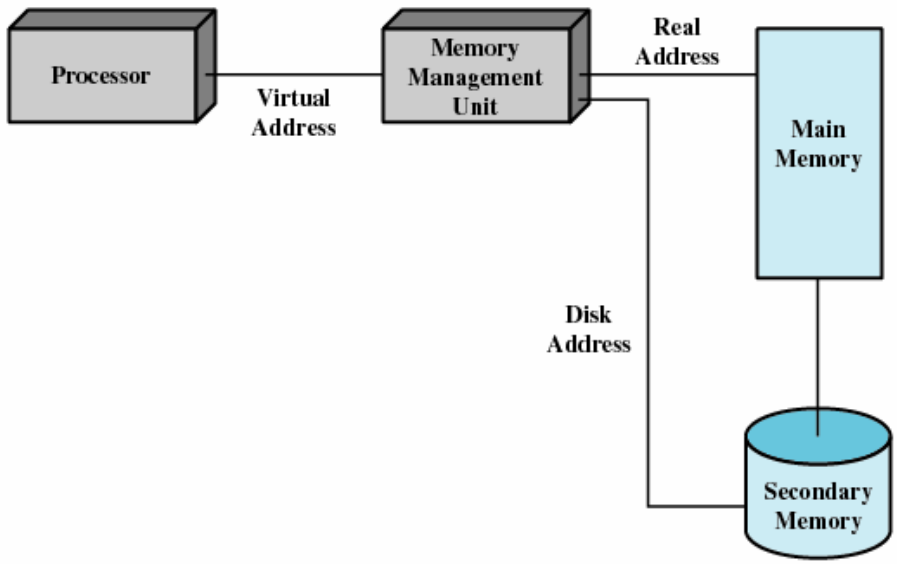
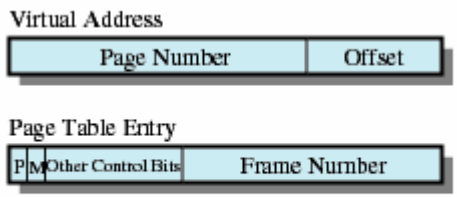


Figure 2.10 Virtual Memory Addressing

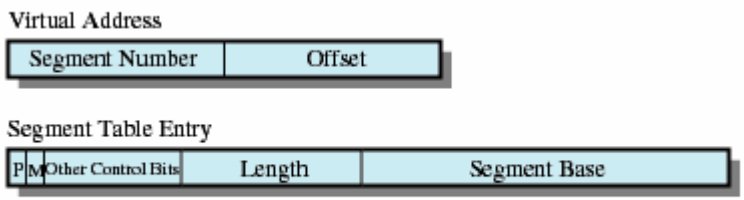
Support Needed for Virtual Memory

- Hardware must support / modified bit
  - Paging



(a) Paging only

  - Segmentation



(b) Segmentation only

- Operating system must be able to manage movement of pages (and/or segments) between primary and secondary memory.

Hierarchical Page Table

- Page table itself may grow to considerable size
- ➢ Swap parts of the page table to secondary storage

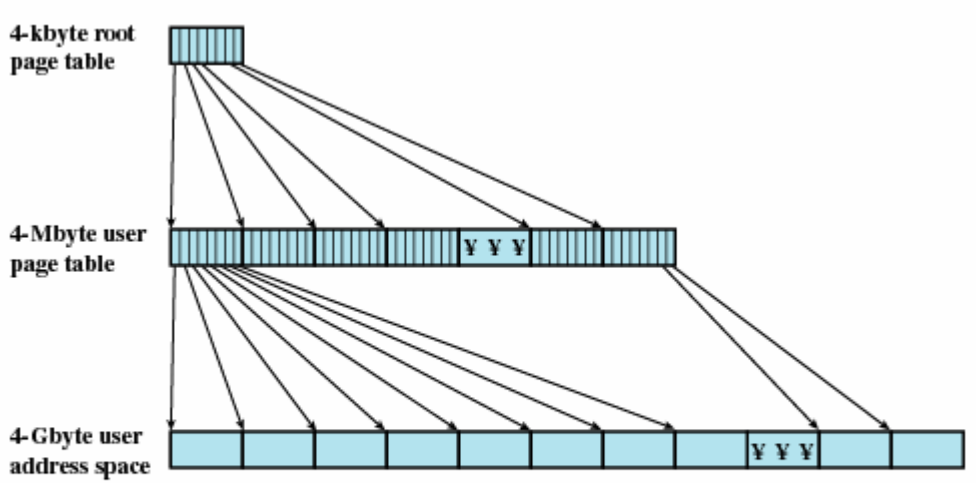


Figure 8.4 A Two-Level Hierarchical Page Table

- Problem: Each virtual memory reference may cause two physical memory accesses
  - One to fetch the page table
  - One to fetch the data

Translation Lookaside Buffer

- Translation Lookaside Buffer (TLB) contains page table entries that have been most recently used

1. Given a virtual address, processor examines TLB
   - If page table entry is present (TLB hit), frame number is retrieved and real address is formed
   - If page table entry is not found in TLB (TLB miss), page number is used to index the process page table
2. OS checks if page is already in main memory
   - If not in main memory a page fault is issued
3. The TLB is updated to include the new page entry

Translation Lookaside Buffer

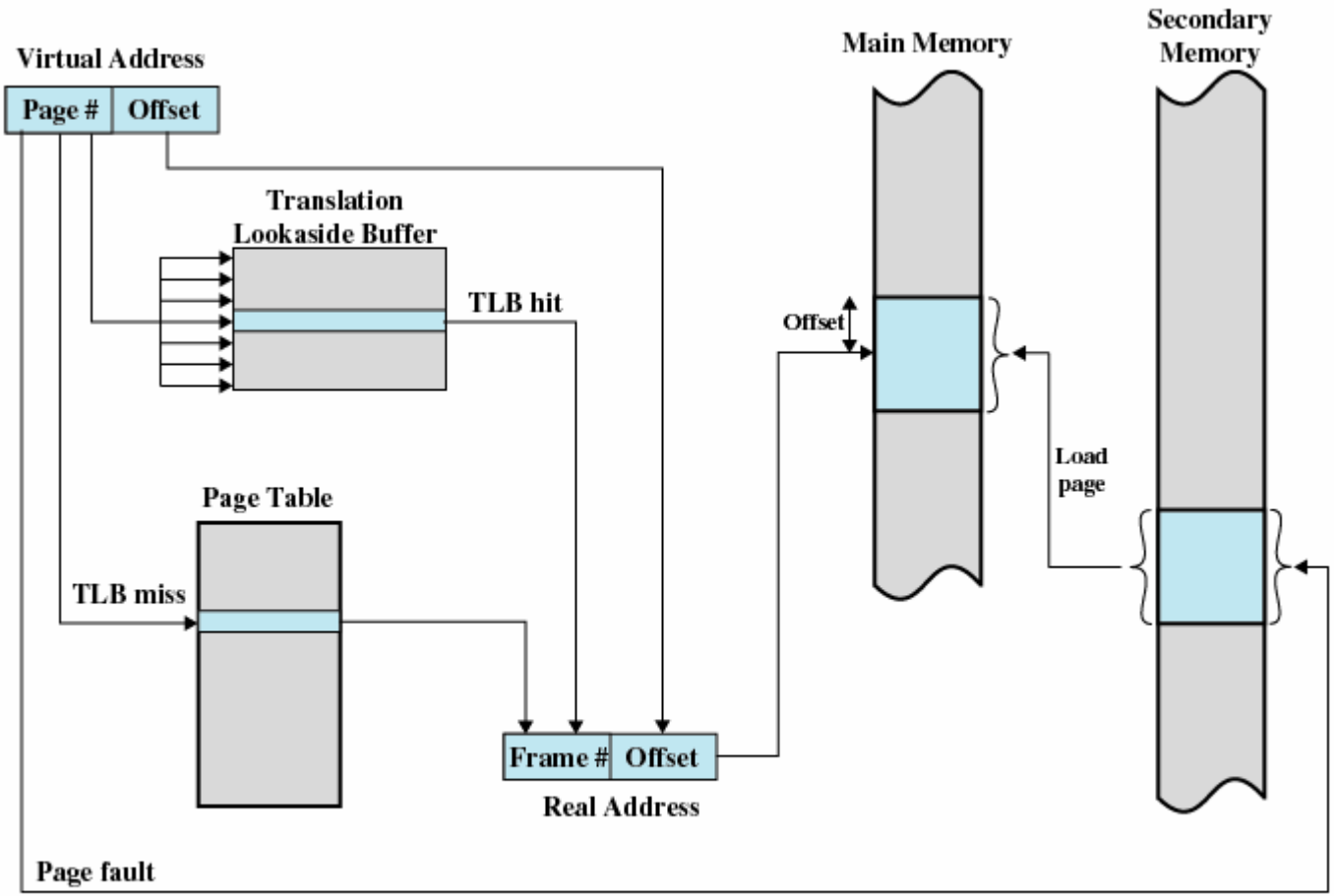


Figure 8.7 Use of a Translation Lookaside Buffer

Problem

- Thrashing
  - Swapping out a piece of a process *just before* that piece is needed
  - Processor spends most of its time swapping pieces rather than executing user instructions

➢ Principle of Locality
  - Program and data references within a process tend to cluster
  - Possible to make intelligent guesses about which pieces will be needed in the future

Algorithms / Policies

- Fetch Policy
  - Determines when page should be brought into memory
  - *Demand paging* only brings pages into main memory when reference is made to location on page
  - *Prepaging* brings in more pages than needed
- Replacement Policy
  - Which page is replaced?
  - Page removed should be the page least likely to be referenced in near future
  - Most policies predict the future behavior on the basis of past behavior

Basic Replacement Algorithms

- Optimal policy
  - Selects page for which the time to next reference is the longest
  - Impossible to have perfect knowledge of future events
- Least Recently Used (LRU)
  - Replaces the page that has not been referenced for the longest time
  - By the principle of locality, least likely to be referenced in the near future
- First-in, first-out (FIFO)
  - Treats page frames allocated to a process as circular buffer
  - Pages are removed in round-robin style
  - Page that has been in memory the longest replaced (may be needed soon)
- Clock Policy
  - When a page is first loaded in memory, *use bit* is set to 1
  - When page is referenced, use bit is set to 1
  - During the search for replacement, each use bit set to 1 is changed to 0
  - When replacing pages, first frame with use bit set to 0 is replaced

Replacement Example

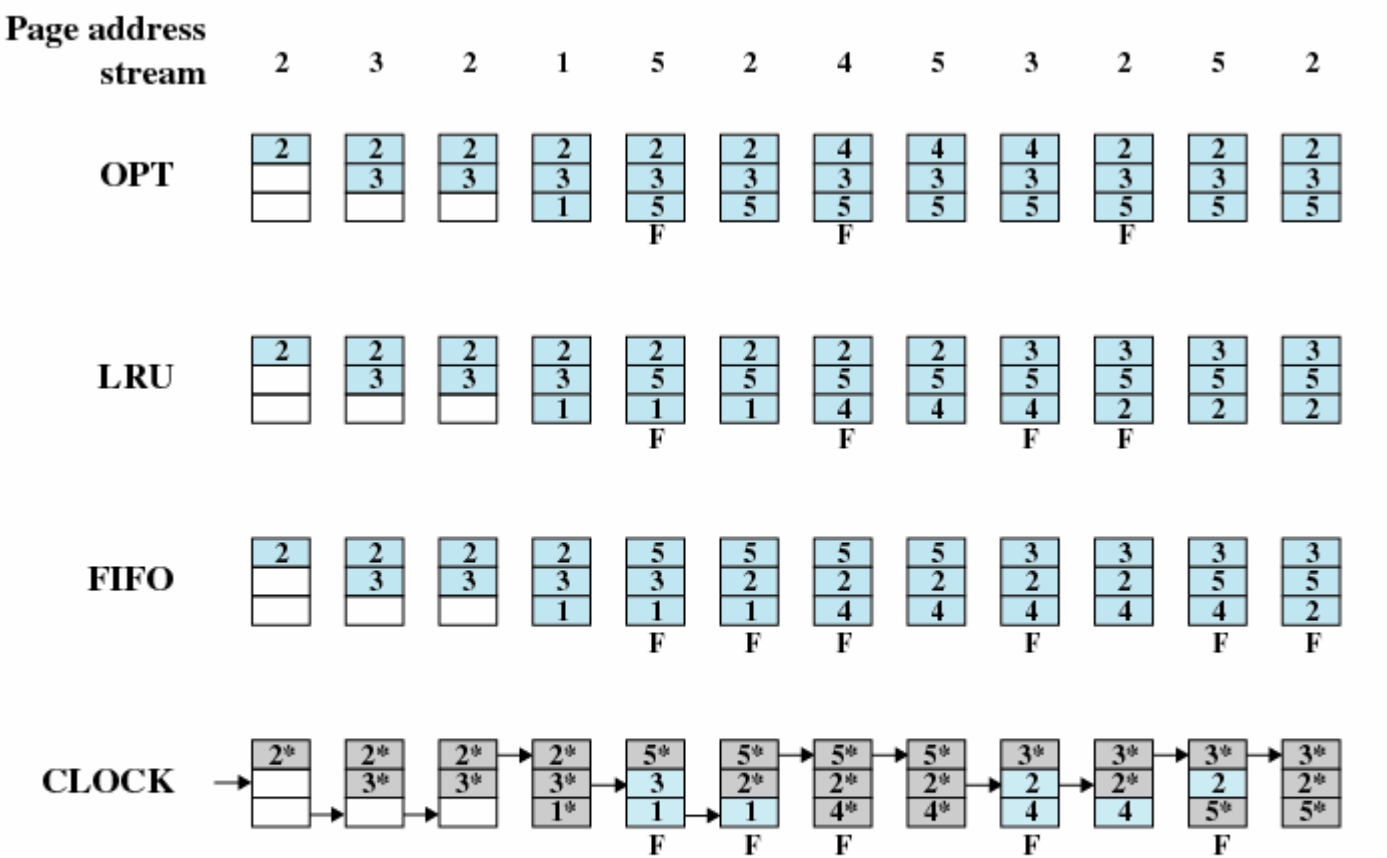


Figure 8.15 Behavior of Four Page-Replacement Algorithms

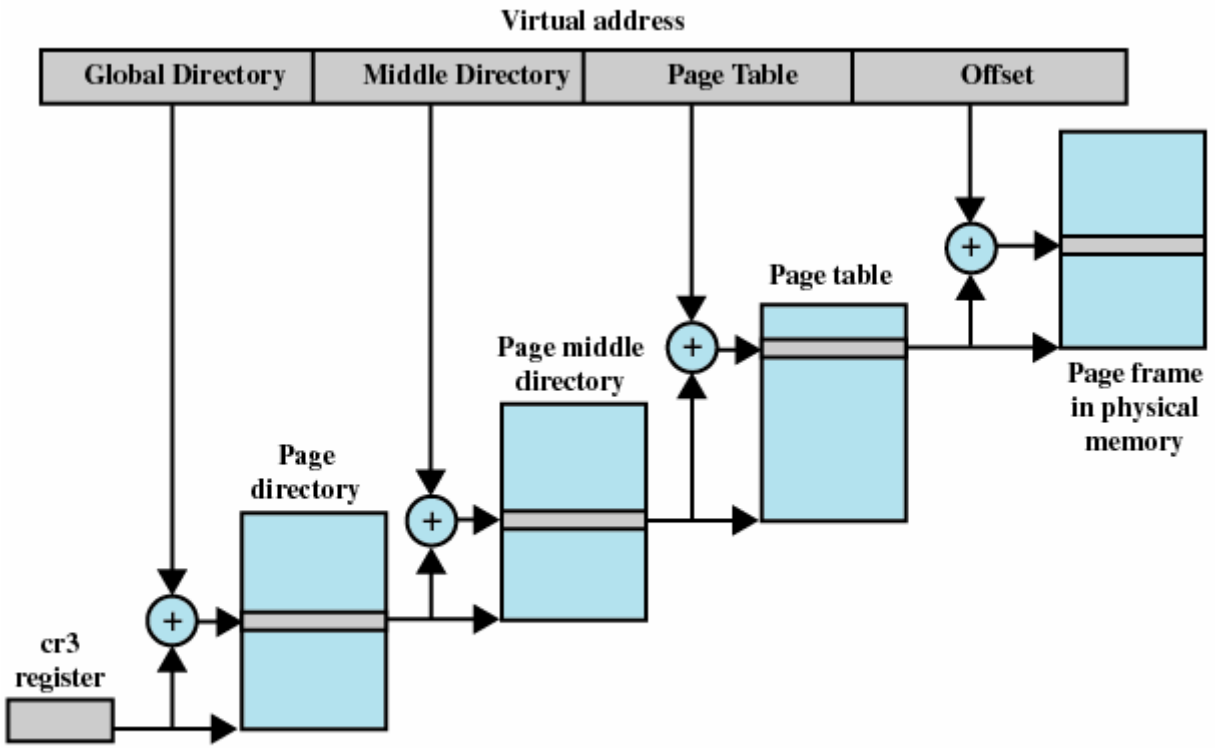


Figure 8.25 Address Translation in Linux Virtual Memory Scheme

Windows Virtual Address Space

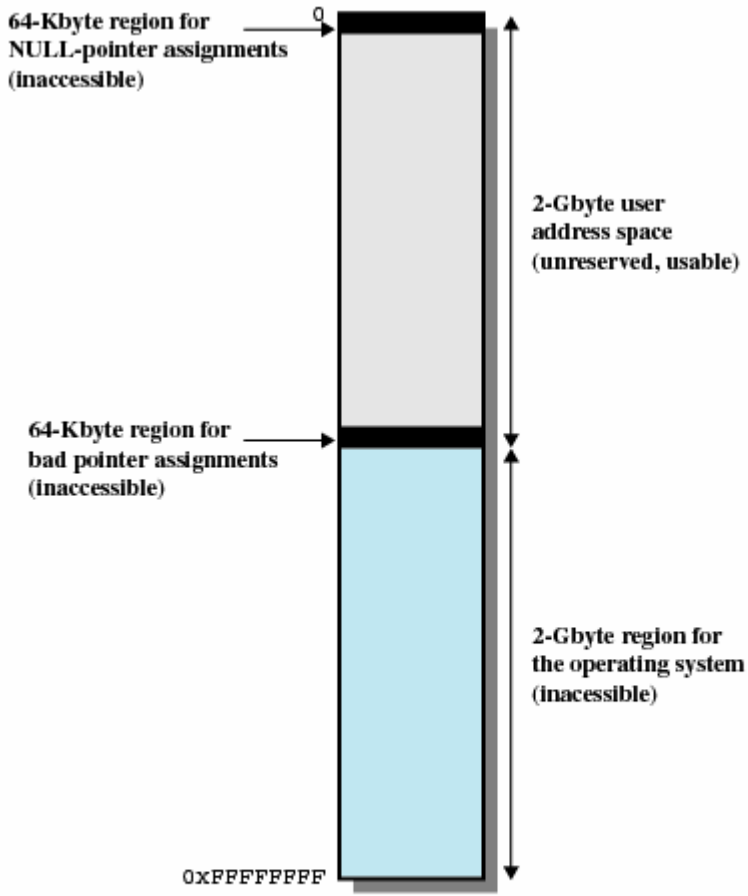
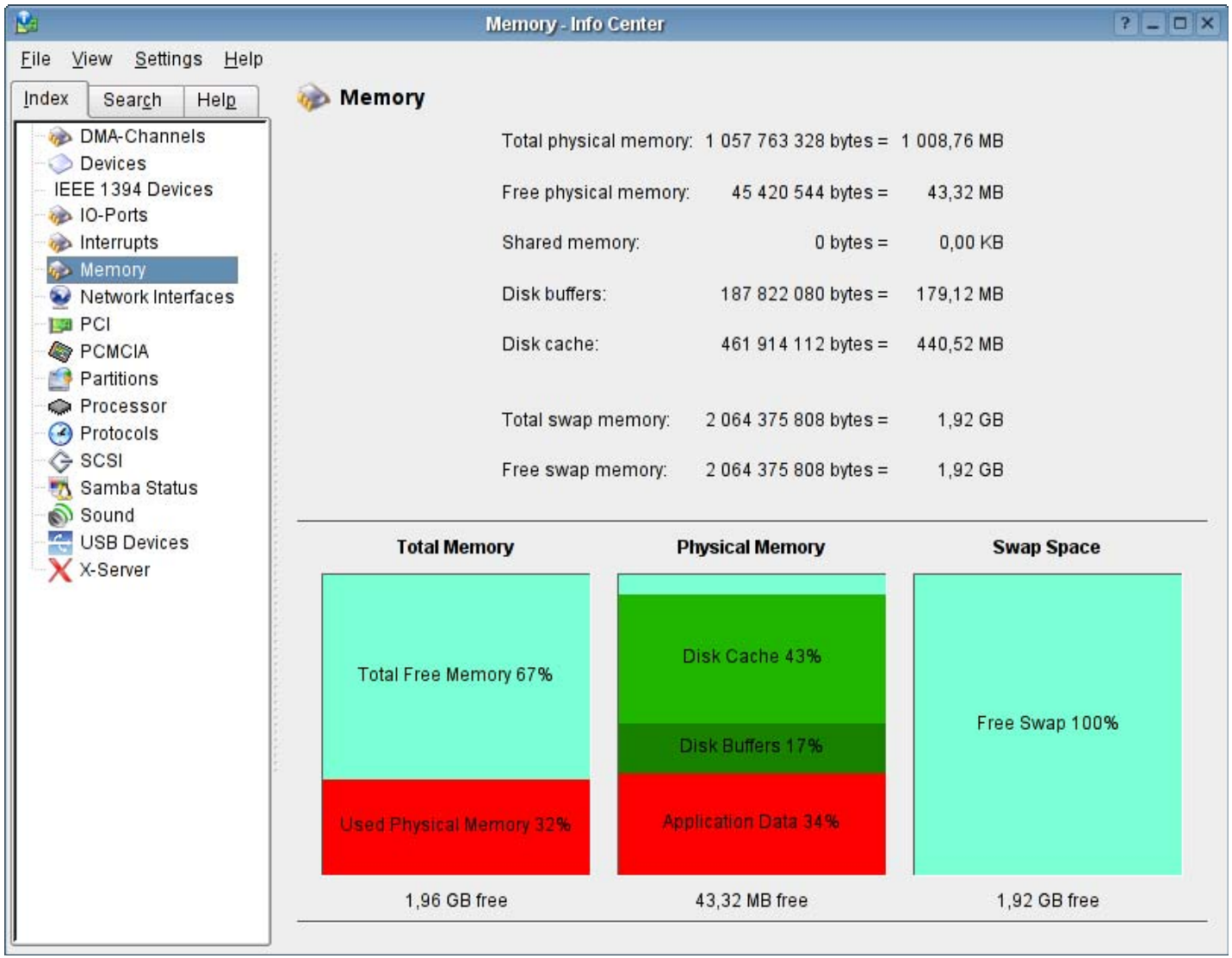
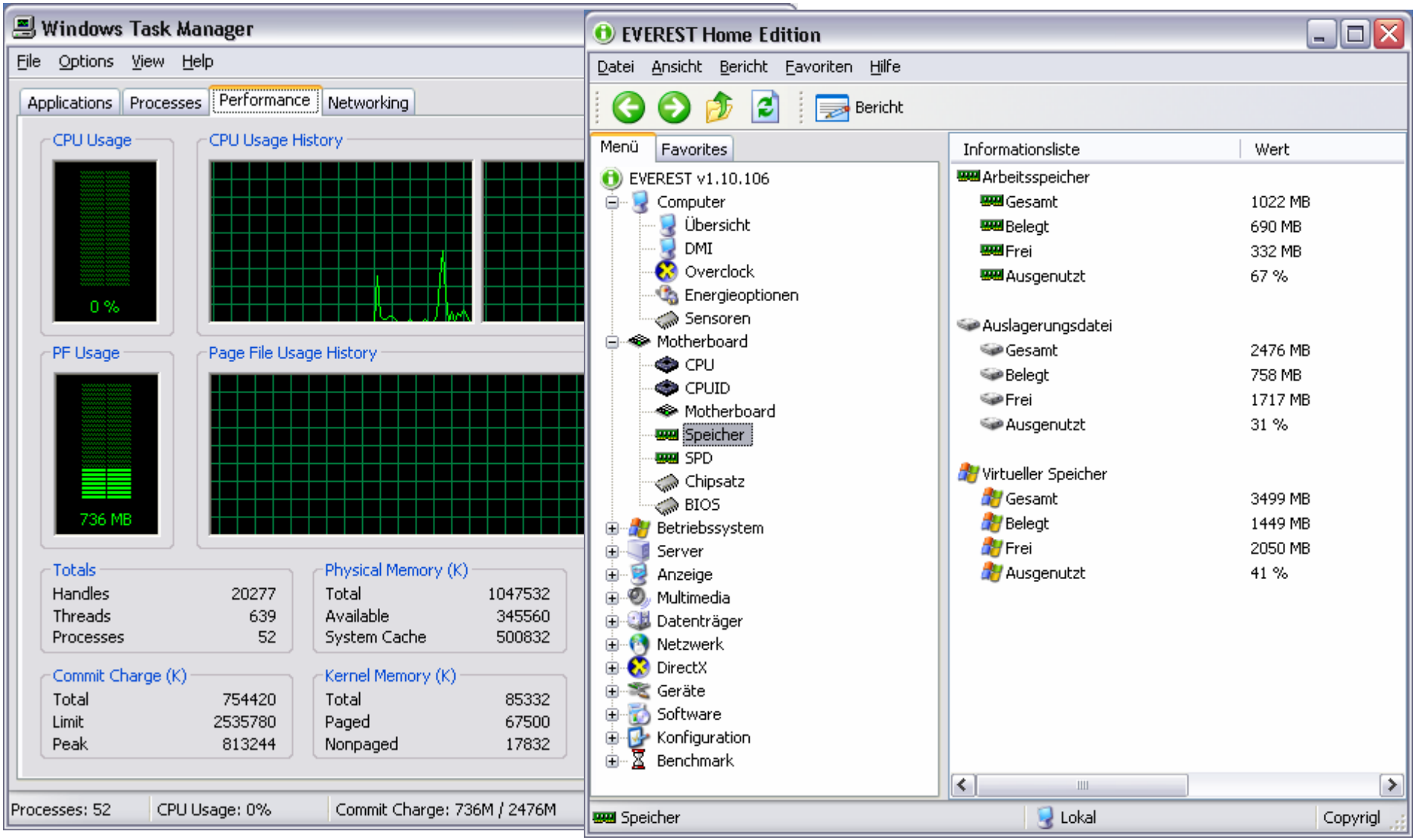


Figure 8.26 Windows Default Virtual Address Space

Linux Memory Utilization



Windows Memory Utilization



Windows Paging