Module 12

Operating System

Module Twelve

- Operating System - Part Four
- In this presentation, we are going to talk about :

  Memory Management

Overview

- Previously we talked about:
  - Process Scheduling
  - I/O Supervision

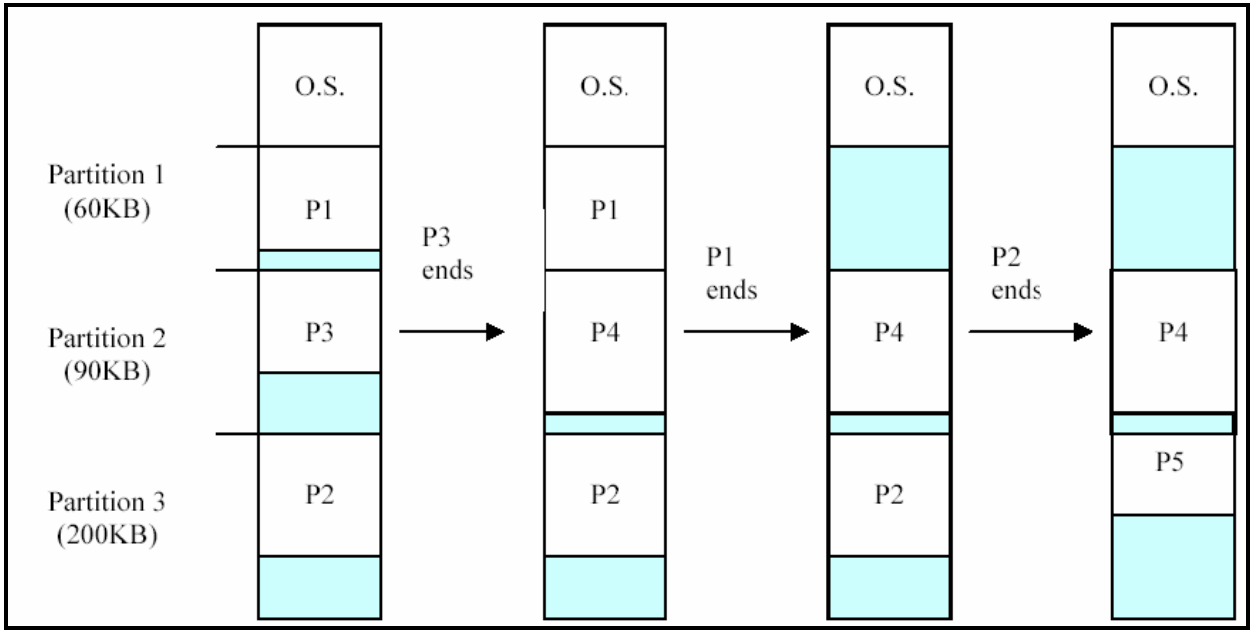
  Now: Memory Management

Real Memory

- Programs are typically stored on disc as executable binary images. The program must be copied into physical memory and a process assigned to it for execution.
- The Operating System assigns the space allocated to the program, reserving enough space for the program to properly execute.
- The challenge is accommodating multiple programs.

Real Memory

- Partitions
  - Divisions of the main memory
  - Fixed - unmovable Predictable program sizes
  - Variable - movable First fit / Best fit / Worst fit

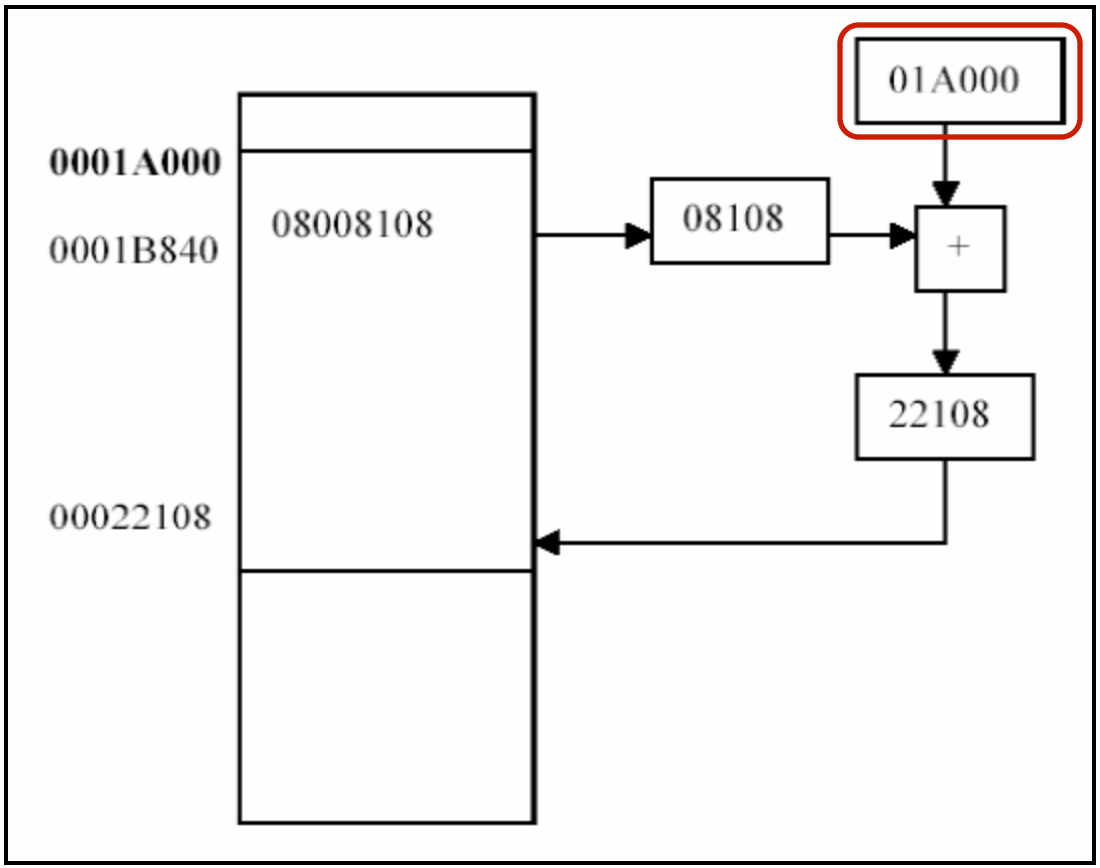


Real Memory

- Memory Fragmentation Problem
  - Relocatable Partitions
  - Moving the executing program within memory
  - Difficult after execution started
  - Hardware support needed

Hardware Support

- Relocation Register
- Program link-edited-loaded as if starting at 000000.
- Every address modified at execution time



Real Memory

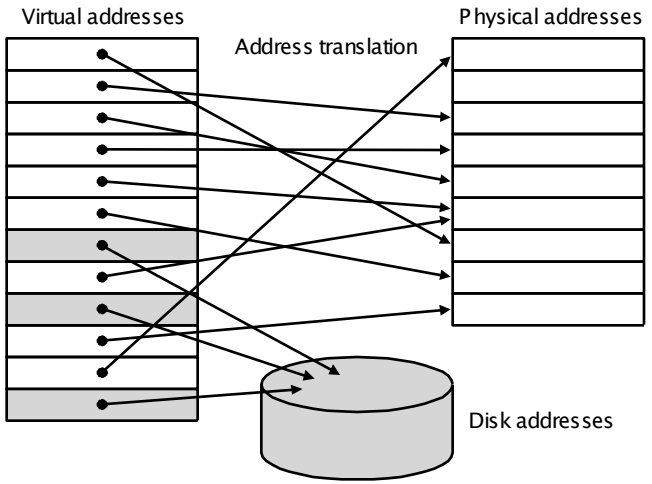
- Memory Protection
  - Bounds Registers — Set by Operating System as program loaded and refreshed when the process is Dispatched for execution.
  - Storage Protection Key — **Identifier** labeling each memory block. Checked before each memory access.

Virtual Memory

- User program is stored on disc.
- Pages of the program loaded into memory as needed.
- Demand Paging
- Fixed length program pages
- Memory divided into fixed length frames of the same size
- Page Map Table

Virtual Memory

- Main memory can act as a cache for the secondary storage (disk)



- Advantages:
  - illusion of having more physical memory
  - program relocation
  - protection

Frames and Pages

- Physical memory is overlain with Frames.
- The Logical Pages are inserted into the Frames.
- The Logical Page number is ‘translated’ via the Page table into the Physical Frame number.
- The Physical frame number plus the page offset is the address.

Page Map Table

- One for each process
- One entry for each page in the process program
  - frame number or disc address
- Uses hardware Dynamic Address translation

Page Tables

Logical program pages on disk.

Page map table to convert logical address to physical memory address.

Assigned memory frames as available

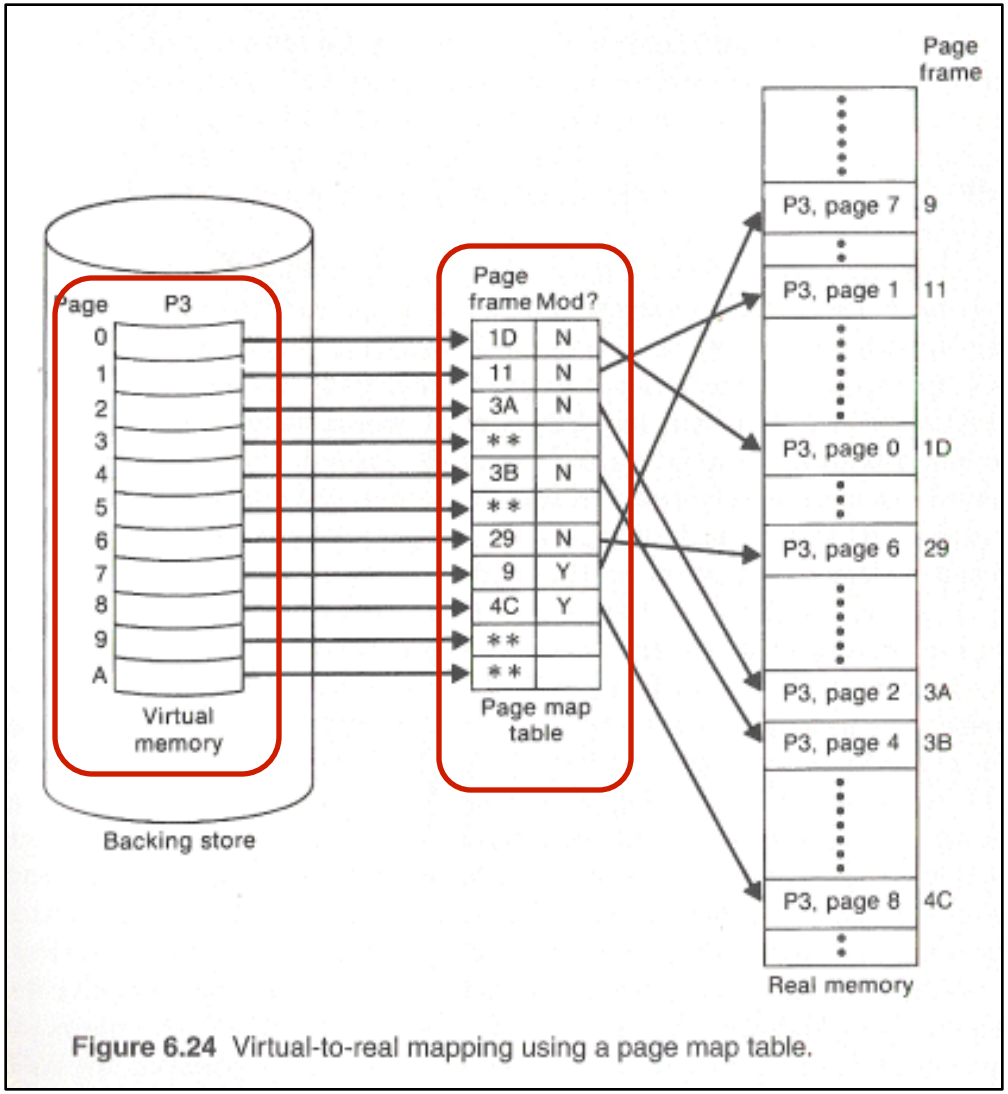
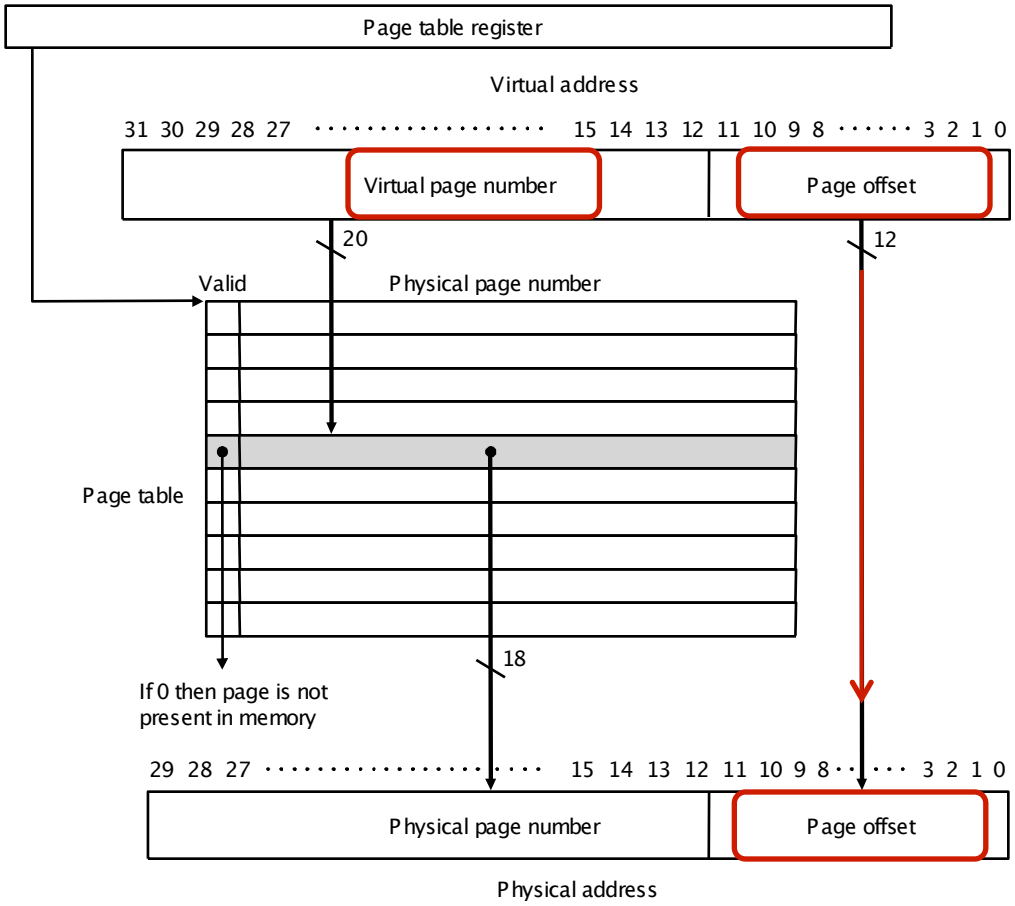


Figure 6.24 Virtual-to-real mapping using a page map table.

Page Tables

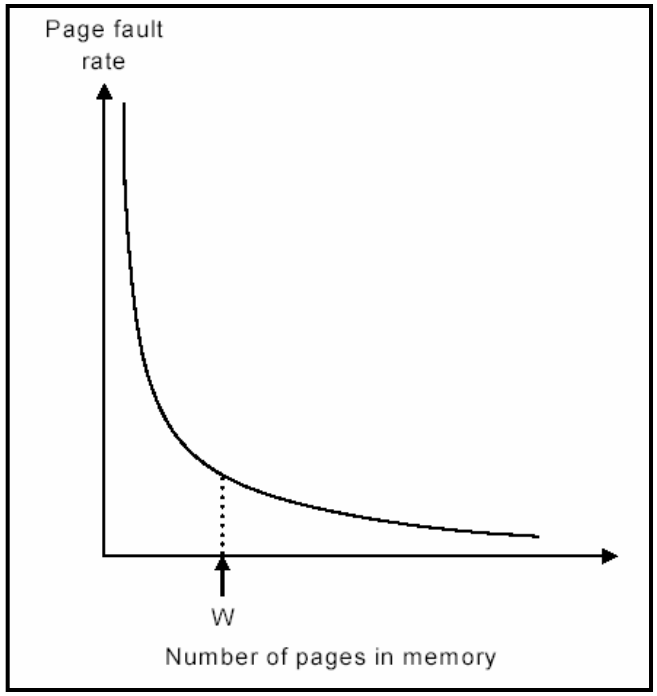

Page table register
Virtual address
31 30 29 28 27 ····· 15 14 13 12 11 10 9 8 ····· 3 2 1 0
Virtual page number
Page offset
20
12
Valid
Physical page number
Page table
18
If 0 then page is not present in memory
29 28 27 ····· 15 14 13 12 11 10 9 8 ····· 3 2 1 0
Physical page number
Page offset
Physical address

Pages: virtual memory blocks

- **Page faults**: the data is not in memory, retrieve it from disk
  - Huge miss penalty, thus pages should be fairly large (e.g., 4KB)
  - Reducing page faults is important (LRU is worth the price)
  - Can handle the faults in software instead of hardware
  - Using write-through is too expensive so we use writeback

Page Fault

- Generates a program interrupt
- Operating System code executed to
  - Select a new memory page frame
  - Least Recently Used - LRU
  - Oldest in Memory – FIFO
  - Random choice
- Working Set - the number of frames allocated to the process
- **Thrashing**



Real Memory Review

- Program must be copied into physical memory and a process assigned to it for execution.
- The challenge is accommodating multiple programs.
- Partitions
- Memory Fragmentation Problem
- Hardware support
- Virtual Memory

Summary

- Operating Systems
- Basic Function : Make the Computer easier to Use
- Operating System types
- Hardware Dependent Tasks
  - Interrupt Processing
  - Process Scheduling
  - I/O Supervision
  - Memory Management