CS 3305A

Memory Management

Lecture 15

Nov 13th 2023

Memory Management: Lecture 15 - 19

- Lecture 15
  - Intro to Memory management
  - Memory Hierarchy
  - Memory Allocation Techniques
    - Contiguous allocation
- Lecture 16
  - Memory Allocation Techniques
    - Contiguous allocation
- Lecture 17
  - Memory Allocation Techniques
    - Distributed allocation or Paging
  - Page Table
- Lecture 18-19
  - Virtual Memory & Page Replacement

Agenda

- Intro to Memory management
- Memory Hierarchy
- Memory Allocation Techniques
    - Contiguous Memory Allocation
        - Fixed Allocations
        - Variable or Dynamic Allocation

Introduction

- Our machines today have 10,000 times more memory than the IBM 7094 – leading edge machine of the 1960's

- Memory unit cost has dropped dramatically

- Operating systems must manage memory

Introduction

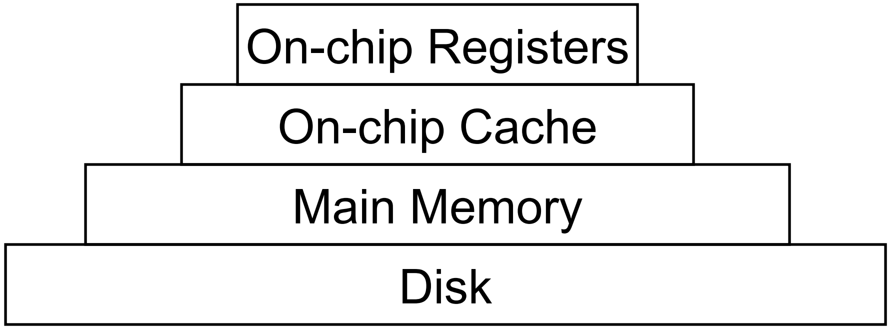
- Memory management requires
  - Allocate memory to processes when needed
  - Managing memory efficiently during runtime
  - Deallocate memory when processes are done

Introduction

- You can think of memory as a large array of bytes
  - Each byte has its own address

- Each of these operations require memory addresses and their management
  - Fetch an instruction from memory (reading)
  - Instruction is decoded
  - After instruction execution
    - Results may be stored back in memory (writing)

Memory Hierarchy



On-chip Registers

On-chip Cache

Main Memory

Disk

- A CPU waiting for data from main memory is not desired
- Remedy: Add fast memory between the CPU and main memory called a cache
- Typical average data transfer rate
  - SSD ~4 x faster than HDD
  - RAM is ~30 x faster than SSD
  - Cache is ~100 x faster than RAM

Memory Allocation Techniques

- Contiguous Memory allocation
  - Entire process has to be allocated in a memory in a contiguous manner
- Distributed memory allocation / Paging
  - Process can be allocated anywhere in the physical memory in a distributed manner

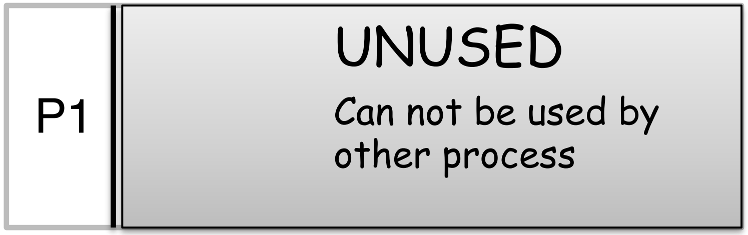
Contiguous Memory Allocation

Contiguous Memory Allocation

- We will start out with the most basic method used that allows multiple processes to reside in memory
- With contiguous memory allocation each process is contained in a single section of memory that is contiguous
    - Fixed partition
    - Variable / Dynamic Partition

Fixed Partitioning

- Any program/process, no matter how small, occupies an entire partition of the memory partition

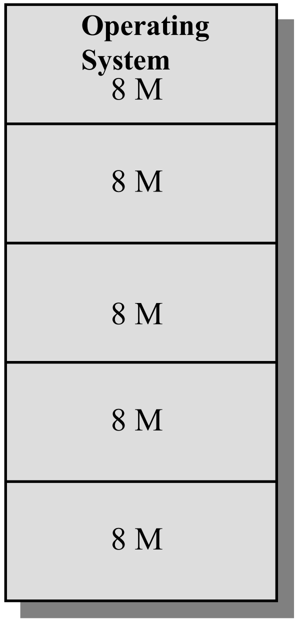


P1

UNUSED

Can not be used by other process

- Equal-size or unequal sized partitions



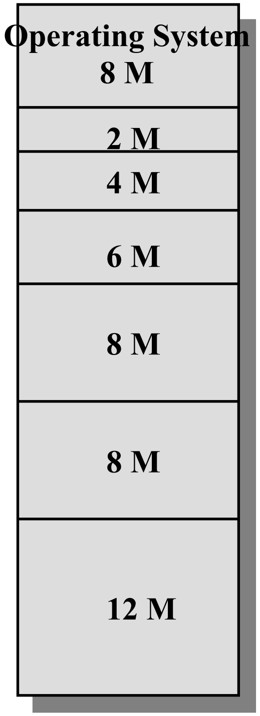
Operating System 8 M

8 M

8 M

8 M

8 M



Operating System 8 M

2 M

4 M

6 M

8 M

8 M

12 M

Fixed Partitions

- Leads to internal fragmentation
- Was used by OS/360 on large IBM mainframes for many years
- Today **no modern OS uses fixed partitions**
- Fixed Partition Memory Allocation Algorithms:
  - First Fit
  - Best Fit
- Operating system must decide which free block to allocate to a process
  - Best-fit, and First-fit algorithms

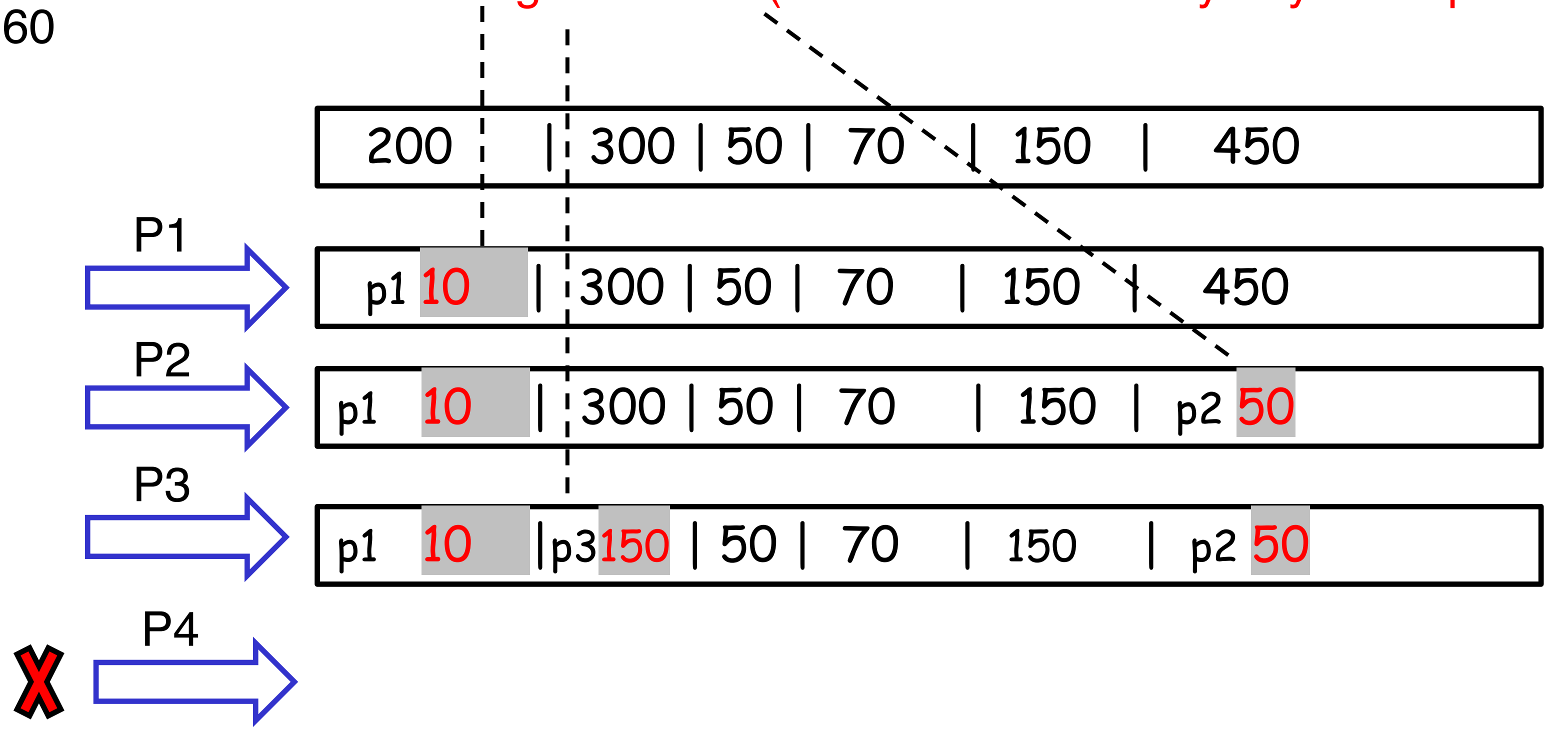
Fixed Partitioning Placement Algorithm

- First-fit algorithm
    - Starts scanning memory from the beginning and chooses the first available block that is large enough.

Fixed Partitions First Fit

P1 = 190
P2 = 400
P3 = 150
P4 = 160

Internal Fragmentation (can NOT be used by any other process)



200 | 300 | 50 | 70 | 150 | 450

P1

p1 10 | 300 | 50 | 70 | 150 | 450

P2

p1 10 | 300 | 50 | 70 | 150 | p2 50

P3

p1 10 |p3150 | 50 | 70 | 150 | p2 50

P4

Internal Fragmentation = 10 + 50 + 150 = 210
Any External Fragmentation ? = 160

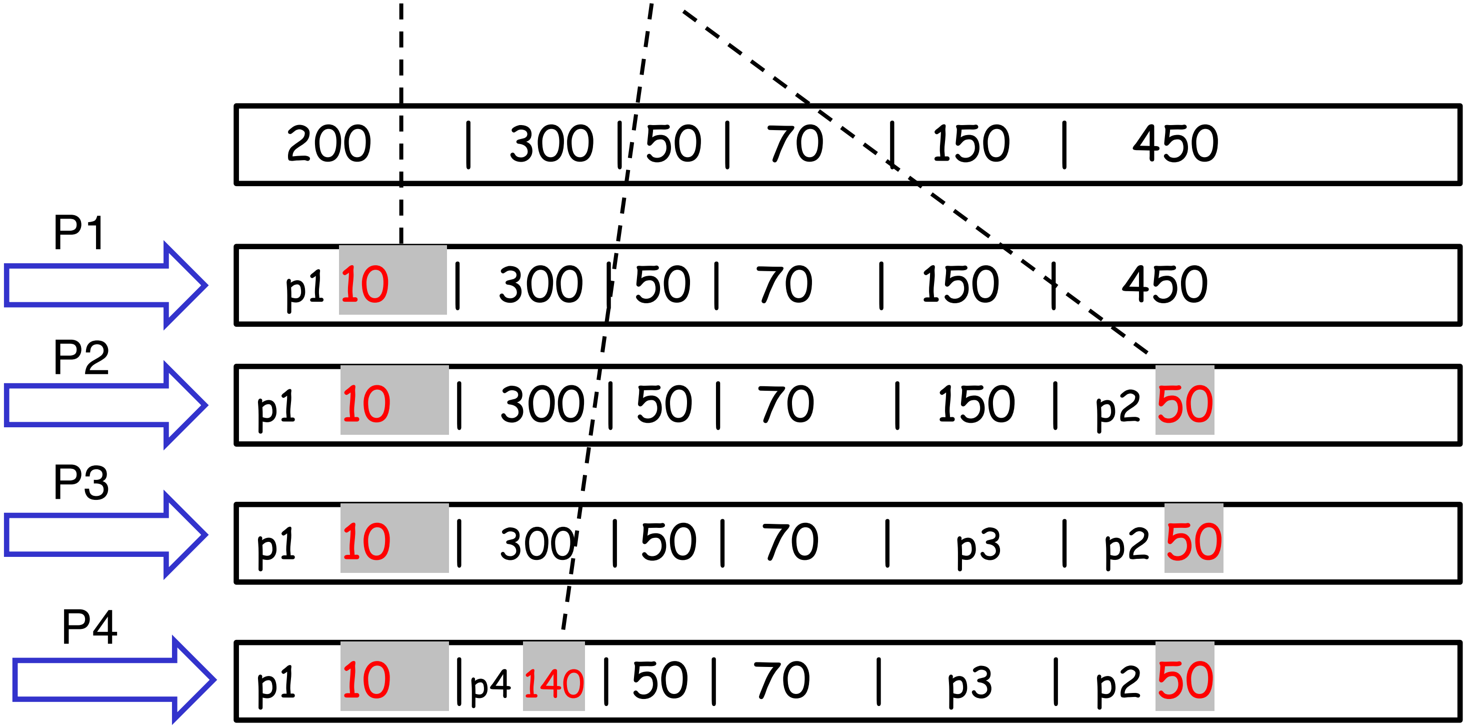
Fixed Partitioning Placement Algorithm

- Best-fit algorithm
  - Choose the block that is closest in size to the request
  - The smallest block is found for a process

- Can we compare the performance between the First-fit and the Best-fit?

Fixed Partitions Best Fit

P1 = 190
P2 = 400
P3 = 150
P4 = 160



Internal Fragmentation (can NOT be used any other process)

200 | 300 | 50 | 70 | 150 | 450

P1

p1 10 | 300 | 50 | 70 | 150 | 450

P2

p1 10 | 300 | 50 | 70 | 150 | p2 50

P3

p1 10 | 300 | 50 | 70 | p3 | p2 50

P4

p1 10 |p4 140 | 50 | 70 | p3 | p2 50

Internal Fragmentation = 10 + 50 + 140 = 200
Any External Fragmentation ? = 0