CSCE 311 - Operating Systems
Contiguous Memory Allocation

Qiang Zeng, Ph.D.
Fall 2018
Previous class...

• Uniprocessor policies
  – FCFS, Shortest Job First
  – Round Robin
  – Multilevel Feedback Queue

• Multiprocessor policies
  – A MFQ per processor
  – A process has affinity with a processor
  – Exception: idle cores can steal processes
Background

- Main memory and registers are the only storage that a CPU can access directly
- Program must be brought (from disk) into memory for it to be run
Outline

• Fixed partitions
• Dynamic partitions
• Buddy system

This Lecture:

Contiguous allocation:
Each process occupies a contiguous memory region in the physical memory

Next Lecture:

Non-contiguous allocation:
Each process occupies multiple memory regions scattered in the physical memory.
Fixed partitions

• The bounds of each partition are fixed/predefined

• Disadvantages:
  – Cause Internal Fragmentation when the allocated space is larger than the need
    • What is “fragmentation”? Small useless chunks
    • E.g., when you put a 13M process in the 16M partition, 3M space is wasted and it is called internal fragmentation
      • Internal: the wasted space is inside allocated space
  – The number of active processes is limited

• Analogy: street parking with meters
Fixed partitions - questions

• If 8M partitions are all used, where do you place a 7M process, and what is the size of the internal fragmentation?
  – The 12M partition is the best choice
  – The internal fragmentation is 5M

• How to resolve the severe internal fragmentation?
  – Dynamic partitions
Dynamic partitions

• Process is allocated exactly the memory it requires
• The partitions are dynamic: the number and locations of partitions are not fixed
• Analogy: street parking without meters
Dynamic partitions - example

(a) 8M
   56M

(b) 20M
    36M
   14M
   22M

(c) 20M
    14M
   18M
   4M

(d) 20M
    14M
   18M
   4M

(e) Process 1 20M
    Process 4 8M
    Process 3 6M
    Process 3 4M

(f) Process 1 20M
    Process 4 8M
    Process 4 6M
    Process 4 4M

(g) Process 1 20M
    Process 4 8M
    Process 4 6M
    Process 4 4M

(h) Process 2 14M
    Process 4 6M
    Process 4 6M
    Process 3 4M
Dynamic partition - disadvantage

- **External fragmentation**
  - Their total size is large enough to satisfy a request, but they are not contiguous, so cannot be used to service the request
  - *External*: fragmentation is outside allocated space

- **Solution: Compaction**
  - OS shifts processes so that they are contiguous; thus, free memory is together in one block
  - But, program execution must be paused for relocation; waste CPU time
### Placement Algorithms

- When there is more than one free block of memory of sufficient size, the system must decide which free block to allocate.

<table>
<thead>
<tr>
<th>Best-fit</th>
<th>First-fit</th>
<th>Next-fit</th>
</tr>
</thead>
<tbody>
<tr>
<td>• chooses the block that is closest in size to the request</td>
<td>• scan memory from the beginning and chooses the first available block that is large enough</td>
<td>• scan memory from the location of the last placement and chooses the next available block that is large enough</td>
</tr>
</tbody>
</table>
Assume a 16M partition is requested
Questions

• Does Fixed Partitions Allocation have external fragmentation?
  – Zero
  – There are no small useless chunks

• Does Dynamic Partitions Allocation have internal fragmentation?
  – Zero
  – The allocated partition size is as needed
Big picture

- Fixed partitions
- Dynamic partitions
- Buddy system

- Segmentation
- Paging

Contiguous allocation:
Each process occupies a contiguous memory region in the physical memory.

Non-contiguous allocation:
Each process comprises multiple memory regions scattered in the physical memory.