Memory Management

Instructor: Adam C. Champion, Ph.D.
CSE 2431: Introduction to Operating Systems
Reading: Sections 9.1–9.3, [OSC]
Outline

• Basic Memory Management
• Swapping
• Variable Partitions
• Memory Management Problems
Basic Memory Management

• Storage Hierarchy
• Memory Management Problems
• Fixed Partitions for Multiprogramming
• Variable Sized Partitions
• Memory Allocation Strategies
Storage Hierarchy

Cost

$600 a chip

$10^{-2}$ per byte

$10^{-4}$ per byte

$10^{-8}$ per byte

Size

$2^{13}$ bytes

$2^{27}$ bytes

$2^{30}$ bytes

$2^{40}$ bytes

CPU Reg

32–64 bits

Cache

4–128 words

Memory

512–16K words

Secondary Storage
General Memory Problem

• Limited (expensive) physical resources: Main memory
  – E.g. Windows 10 needs 1G, prefers 2G RAM
  – We want to use it as efficiently as possible
• Abundant, slower resources: Disk
• OS needs to provide an abstraction of memory hierarchy to user-level applications
Responsibilities of Memory Manager

• Manage memory hierarchy
  – Monitor used and free memory
  – Allocate memory to processes
  – Reclaim (De-allocate) memory
  – Swapping between main memory and disk
Scenarios: Memory Management Problems

• One program, the size is less than memory size
  – The simplest case

• One program, using lots of memory
  – Can you only keep part of the program in memory?

• Many programs, total size is less than memory size
  – Technically possible to pack them together
  – Will programs know about each other’s existence?

• Lots of programs, total size exceeds memory size
  – What programs are in memory, and how to decide?
Mono-Programming No Swapping

- Run one process at a time
  - simplest possible memory management scheme
- Memory is shared only between OS and the process.
- Three different ways to organize memory

![Memory Organization Diagram]

0xFFFF...

User Program

OS in RAM

OS in ROM

Device drivers in ROM

User Program

OS in RAM
Overlaying

Used when the process memory requirement exceeds the physical memory space.
Multiprogramming with Fixed Partitions

• Divide memory into $n$ (possibly unequal) partitions.

• Problem:
  – Internal Fragmentation

![Diagram showing memory partitions and free space](image)
Fixed Partition Allocation

• Separate input queue for each partition
  – Requires sorting the incoming jobs and putting them into separate queues
  – Problems?

• One single input queue for all partitions.
  – Find a job for fitting in an available partition
    • Available Fit
    • Best Fit
  – Problems?
Problems?
Relocation

- Correct starting address when a program should start in the memory
- Different jobs will run at different addresses
  - When a program is linked, the linker must know at what address the program will begin in memory.
- Logical addresses, Virtual addresses
  - Logical address space, range (0 to max)
- Physical addresses, Physical address space
  - range (R+0 to R+max) for base value R.
- User program never sees the real physical addresses
- Who translates virtual to physical addresses?
  - Program rewriting at loading time
  - Help from relocation registers at execution time
Relocation Register

CPU Instruction Address

Logical Address MA

Base Register BA

Physical Address MA+BA

Memory

+
Protection

• Problem:
  – How to prevent a malicious process to write or jump into other user's or OS partitions

• Solutions:
  – Memory protection code
  – Base bounds registers
Base Bounds Registers

- CPU Address
- Logical Address (LA)
- Base Address (BA)
- Memory Address (MA)
- Physical Address (PA)
- Base Address
- MA + BA
- Limit Address

Fault
Review

• Basic Memory Management
  – Memory Manager’s Responsibilities
  – Mono-Programming
  – Multi-Programming with Fixed Partitions
    • Internal fragmentation
  – Relocation and Protection
Memory Management (More…)

• Batch System
  – Multiprogramming with fixed partitions
  – In the memory until job finishes
  – Keep CPU busy

• Timesharing Systems
  – *Not enough memory to hold all active processes*
  – Swapping (whole process)
  – Virtual memory (partial process)
Swapping

- Move the whole process to/from disk
- Allows several processes to share a fixed partition
- Processes that grow can be swapped out and swapped back in a bigger partition
Outline

• Basic Memory Management
• Swapping
• Variable Partitions
• Memory Management Problems
Swapping (1)

- Monitor
- User Partition
- Disk
- Job 1
Swapping (2)

Monitor

Job 1

User Partition

Disk

Job 1
Swapping (3)

- Monitor
- Job 1
- User Partition
- Disk
- Job 1
- Job 2
Swapping (4)
Swapping (5)
Swapping (6)
Outline

• Basic Memory Management
• Swapping
• Variable Partitions
• Memory Management Problems
Variable Partitions and Fragmentation

1. Monitor | Job 1 | Job 2 | Job 3 | Job 4 | Free
2. Monitor | Job 1 | Job 3 | Job 4 | Free
3. Monitor | Job 1 | Job 5 | Job 3 | Job 4 | Free
4. Monitor | Job 5 | Job 3 | Job 4 | Job 6
5. Monitor | Job 7 | Job 5 | Job 3 | Job 8 | Job 6

- Fixed Partitions ↔ Variable Partitions
- External Fragmentation
  - How to solve it?
Compaction

- What is it? What does it looks like?
- Assumes all programs are relocatable
- Processes must be suspended during compaction
- Need be done only when fragmentation gets very bad

<table>
<thead>
<tr>
<th>5</th>
<th>Monitor</th>
<th>Job 7</th>
<th>Job 5</th>
<th>Job 3</th>
<th>Job 8</th>
<th>Job 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Monitor</td>
<td>Job 7</td>
<td>Job 5</td>
<td>Job 3</td>
<td>Job 8</td>
<td>Job 6</td>
</tr>
<tr>
<td>7</td>
<td>Monitor</td>
<td>Job 7</td>
<td>Job 5</td>
<td>Job 3</td>
<td>Job 8</td>
<td>Job 6</td>
</tr>
<tr>
<td>8</td>
<td>Monitor</td>
<td>Job 7</td>
<td>Job 5</td>
<td>Job 3</td>
<td>Job 8</td>
<td>Job 6</td>
</tr>
<tr>
<td>9</td>
<td>Monitor</td>
<td>Job 7</td>
<td>Job 5</td>
<td>Job 3</td>
<td>Job 8</td>
<td>Job 6</td>
</tr>
</tbody>
</table>
Outline

• Basic Memory Management
• Swapping
• Variable Partitions
• Memory Management Problems
Memory Management with Bitmaps

• Use bitmaps for memory status (free or allocated)
• Memory is divided into allocation units.
  – One allocation unit corresponds to 1 bit in the bitmap
  – 0: free, 1: allocated
• Size of allocation unit
  – The smaller the allocation unit, the larger the bitmap.
• Problem: allocation
  – When a new process arrives, the manager must find consecutive 0 bits in the map.
  – Searching a bitmap for a run of a given length is a slow operation.

```
1 0 1 0 1 0 0 0 1 0 1 0 0 1 0
```
Memory Management with Linked Lists

• Use a linked list of allocated and free memory segments (called hole)
  – Sorted by the address or by the size

Before X terminates  |  After X terminates
(a) | (a)  
A  X  B  |  A  B  
(b)  
A  X  |  A  B  
(c)  
X  B  |  B  
(d)  
X  |  B  

Four neighbor combinations for the terminating process X
Memory Allocation Strategies

• **Best Fit**
  – Uses the hole whose size is equal to the need, or if none is equal, the hole that is larger but closest in size.
  – Problem: Creates small holes that can't be used.

• **First Fit**
  – Uses the first available hole whose size is sufficient to meet the need.
  – Problem: Creates average size holes.

• **Next Fit**
  – Minor variation of first fit: search from the last stopped place.
  – Problem: slightly worse performance than first fit.

• **Worst Fit**
  – Uses the largest available hole.
  – Problem: Gets rid of large holes making it hard to run large programs.

• **Quick Fit**
  – Maintains separate lists of holes for some of the more common sizes requested. When a request comes for placement it finds the closest fit.
  – A very fast scheme, but merge is expensive. If merge is not done, memory will quickly fragment in a large number of holes into which no processes fit.
How Bad Is Fragmentation?

• Statistical arguments - Random sizes
• First-fit
• Given N allocated blocks, 0.5*N blocks will be lost because of fragmentation
• Known as 50% RULE
Memory Management Problems

• Fixed partitions
  – Suffer from internal fragmentation
• Variable partitions
  – Suffer from external fragmentation
• Compaction
  – Suffer from overhead
• Overlays
  – Painful for programmers
• Swapping
  – Requires writing to disk sectors
Summary

• Basic Memory Management
• Swapping
• Variable Partitions
• Memory Management Problems