Agenda

- Intro to Memory management
- Memory Hierarchy
- Address Binding
- Memory Allocation Techniques
  - Contiguous Memory Allocation
  - Fixed and Dynamic Allocations
Introduction

概述：

1. 当今我们的机器比1960年代的IBM 7094（当时的尖端机器）拥有10,000倍更多的内存。
2. 内存的成本大幅下降。
3. 西部大学（用于生物信息学研究）有一台1太字节内存的高性能机器。

结论：内存成本的大幅下降使得当今的机器拥有更多的内存，从而支持了生物信息学等领域的研究。
Introduction

- Software and data sets expand to fill the memory available
  - The 1 terabyte of memory - the researchers already want more

- Operating systems must manage memory
Introduction

- Memory management requires
  - Allocate memory to processes when needed
  - Keep track of what parts of memory are in use
  - Deallocate memory when processes are done
Introduction

- You can think of memory as a large array of bytes
  - Each byte has its own address
- Fetch an instruction from memory
- Instruction is decoded
- After instruction execution
  - Results may be stored back in memory
- Each of these operations require memory addresses
A CPU waiting for data from main memory is not desired.

Remedy: Add fast memory between the CPU and main memory called a cache.
Address Binding

- Program execution requires that a program be brought into memory from the disk
- The process can reside in any part of the physical memory (unknown to programmer)
- Compiler allocates logical addresses to the source program/code
- The loader will in turn bind the logical addresses to physical addresses
Address Binding

- Memory references in the code (virtual or logical) must be translated to actual physical memory addresses.

- Run-time Mapping from virtual to physical addresses is done by a hardware device called the memory-management unit (MMU).
Simple MMU for Address Mapping

- The **base** register holds the physical memory address
- The **limit** register specifies the range
- These registers can be loaded only by the operating system
- Ensures the user program doesn’t access anything beyond the range
Simple MMU for Address Mapping

- Simple mapping from logical addresses to physical
- Relocation register: Value is added to every address generated by a user process
Memory Allocation Techniques

- Contiguous Memory Allocation
- Paging
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 and Dynamic Partitioning.
Fixed Partitioning

- Any program, no matter how small, occupies an entire partition.

- Equal-size partitions
Multiprogramming with 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
Dynamic Partitioning

- Partitions are of variable length and number
- Processes are allocated to the closest possible match
- Leads to external fragmentation
- Compaction is required to obtain a large contagious block
  - Shift processes so they are contiguous and all free memory is in one block
### Dynamic Partitioning

- **Variable-size partitions**

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8 M</td>
</tr>
<tr>
<td></td>
<td>2 M</td>
</tr>
<tr>
<td></td>
<td>4 M</td>
</tr>
<tr>
<td></td>
<td>6 M</td>
</tr>
<tr>
<td></td>
<td>8 M</td>
</tr>
<tr>
<td></td>
<td>8 M</td>
</tr>
<tr>
<td></td>
<td>12 M</td>
</tr>
</tbody>
</table>
Placement Algorithm with Partitions

- Variable-size partitions
  - Each process can be assigned to the smallest partition within which it will fit
  - Processes are assigned in such a way as to minimize wasted memory within a partition
Dynamic Partitioning Placement Algorithm

- Operating system must decide which free block to allocate to a process
  - Best-fit, and First-fit algorithms
  - Best-fit algorithm
    - Choose the block that is closest in size to the request
    - This has the worst overall performance
    - The smallest block is found for a process
Dynamic Partitioning Placement Algorithm

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