CS 4410
Operating Systems

Memory:
Hardware and Allocation

Summer 2013
Cornell University
Today

- How the memory is shared among the ready processes?
- Memory
- Address protection
- Logical vs Physical Address
- Contiguous memory allocation
Storage Hierarchy

- Hard Disk
- Memory
- Cache (L1, L2, L3)
- Registers
Memory

• A large array of words.
• Word = 4 or 8 bytes.
• One address for every word.

• Content:
  • Instructions
  • Data
Instruction execution cycle

- Fetch instruction from memory.
  - The PC saves its address.
- Decode instruction.
- Fetch operands form memory.
- Execute the instruction.
- Store result in memory.

- Program and data should be in Memory to become useful.
Memory Management

- Is memory shared between processes? How?

- Monoprogramming
  - Only one process is ready and loaded into memory.
  - It shares the memory space with the OS.
  - Is it efficient?

- Multiprogramming
  - Fixed or variable partitions for every ready process.
  - 2 problems: relocation, protection. Solutions?

- Timesharing
  - Swapping
    - Entire process (code, data) is transferred from disk to memory, and vice versa.
  - Virtual Memory
    - Processes can run when they are partially in the memory.
Memory Management

• What about the memory addresses?
  • Monoprogramming
    - The physical addresses are known to the programmer.
  • Multiprogramming
    - The physical addresses are known at the loading time.
  • Timesharing
    - The physical addresses are known at the execution.
    - The CPU understands logical addresses.
    - The Memory understands physical addresses.
Memory Management

• Basic concerns:
  • Allocation
  • Relocation
  • Protection
Hardware address protection

- Multitasking OS
- Multiple processes loaded in the memory.
- Each process has a separate memory space.
- HW+OS are responsible for address protection.

![Diagram showing address space allocation with base and limit values for different processes.]
Address Binding

- Logical $\rightarrow$ Physical
- Execution time
  - Logical Address $\leftrightarrow$ Virtual Address
- Memory-Management Unit (MMU)
  - Hardware device
  - Run-time mapping

![Diagram of address binding](chart.png)

<table>
<thead>
<tr>
<th>Logical address</th>
<th>Physical address</th>
</tr>
</thead>
<tbody>
<tr>
<td>346</td>
<td>14346</td>
</tr>
</tbody>
</table>

Relocation register: 14000
Logical vs Physical Address

- Multitasking OS
- Memory management: Swap, Virtual Memory
- Logical Address
  - Address generated by the CPU
  - Address loaded into PC
  - Address used in a program
- Physical Address
  - Address seen by the memory unit
Dynamic Loading

- Using Virtual memory:
- Better memory-space utilization
- The main program is loaded into memory.
- A routing of the program is not loaded until it is called.
- It is users' responsibility.
Dynamic Linking

• Using Virtual memory:
• Without this, each program should include a copy of its language library.
• It waists disk and memory space.

• With Dynamic Linking:
  • A stub substitutes a library-routine reference.
  • When stub is executed:
    – It checks if the routine is in the memory.
    – If not, the program loads the routine.
Contiguous Memory Allocation

- Share memory between OS and multiple processes.
- Each process is contained in a single contiguous section in memory.
- Memory protection:
  - CPU scheduler selects process for execution.
  - The dispatcher loads the relocation and limit registers.

![Diagram of memory allocation process]

- CPU
- Logical address
- Limit register
- Relocation register
- Physical address
- Memory
- Trap: addressing error
Allocation Strategies

- Fixed-sized partitions
  - The degree of multiprogramming is bound by the number of partitions.
- Variable-partition scheme
  - The OS keeps a table indicating which parts of memory are available and which are occupied.
  - The OS tries to fit the memory demands of a process in the available memory space.
- Dynamic storage allocation problem:
  - First fit
  - Best fit
  - Worst fit
Fragmentation

- External fragmentation
  - First-fit, Best fit
  - There is enough total memory space to satisfy a request but the available spaces are small and not contiguous.
- Solution 1: Break the physical memory into fixed-sized blocks and allocate memory in units based on block size.
  - Internal fragmentation: the allocated memory is slightly larger than the requested memory.
- Solution 2: Compaction
Today

• How the memory is shared among the ready processes?
• Memory
• Address protection
• Logical vs Physical Address
• Contiguous memory allocation