CS 4410
Operating Systems

Review 1

Summer 2013
Cornell University
System Stack

- Users: Mary, John
- Applications: Web-browser, Word Processor, Video Game
- Operating System: Scheduler, Monitor Driver, Network Driver, Memory manager, Disk manager
- Hardware: CPU, Memory, Disk, Network card, Monitor
HW-OS & OS-App Interface

- device
- device controller
- CPU
- device driver
- OS
- Application
- memory
HW-OS & OS-App Interface

- **Driver to Controller:**
  - Memory-mapped I/O
  - Programmed I/O

- **Controller to driver:**
  - Polling
  - Interrupts

- **Application to Driver:**
  - System Calls (change privilege level using SYSCALL)

- **Driver to Application:**
  - Pass data from OS memory space to application memory space.
Process

• A **process** consists of at least:
  • Code, data, stack, PC, registers, OS resources.

• **State**
  • Ready, running, waiting

• **PCB**

• **Context Switch**
  • Caused by interrupts
  • Store/Restore PCBs
  • Useful for timesharing.
Thread

- Basic unit of CPU utilization.
- It \textbf{belongs} to a process.
- It \textbf{shares} code, data, OS resources (files, etc) with the other threads of the same process.
CPU Scheduling

- Scheduler selects **one** process from the **ready** queue to run.

- **Non-preemptive** or **Preemptive**

- Scheduling **Metrics**: Turnaround time, Waiting time, Response time.

- Scheduling **Algorithms**: FCFS, LIFO, Round Robin, SJF

- Priority Scheduling

- Multilevel Queue Scheduling
Synchronization

- Problem to solve: **Race Condition**
- Solution: **Lock**
- Semaphores
  - wait, signal
  - Usage: mutex, counting, synchronization
- Classic problems: Producer-Consumer, Readers-Writers.
- Monitors
Deadlocks

- **Four** necessary conditions:
  - Mutual Exclusion, Hold and wait, No preemption, Circular wait
- Resource-allocation graph
- Handling deadlocks
  - **Prevention**
    - Negate one of necessary conditions.
  - **Avoidance**
    - The system tries to be in a safe state.
    - Banker's algorithm
  - **Detect & Recover**
Memory Management

• Basic concerns: **Allocation, Relocation, Protection.**

• Paging
  • **Noncontiguous** physical address space of a process
  • **Page Table**: Page Number → Frame Number
  • TLB
  • Structure: Hierarchical Paging, Hashed Page Tables, Inverted Page Tables
Virtual Memory

- The content of a page might be in **memory** or in **hard disk**.
- Add *is_present* bit to every page table entry.
- **Page fault**
- **Page Replacement Algorithms**
  - FIFO, OPT, LRU
- **Thrashing**  →  Working Set, Page Fault Frequency

```
<table>
<thead>
<tr>
<th></th>
<th>Disk</th>
<th>Mem</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>32 :P=1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4183 :P=0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>177 :P=1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5721 :P=0</td>
<td></td>
</tr>
</tbody>
</table>
```

Page Table