CSCC 69H3
Operating Systems
Winter 2013
Professor Bianca Schroeder
U of T

Today …
- Done with concurrency & synchronization!
- Questions on deadlocks?
- Today:
  - Processes and how the OS manages them.
  - Starting memory management

Remember the process state diagram?

Process Scheduling
- Only one process can run at a time on a CPU
- Scheduler decides which process to run
- Goal of CPU scheduling:
  - Give illusion that processes are running concurrently
  - Maximize CPU utilization
- Let’s talk about CPU scheduling in more detail …

Process Life Cycle
- Processes repeatedly alternate between computation and I/O
  - Called CPU bursts and I/O bursts
  - Last CPU burst ends with a call to terminate the process (exit() or equivalent)
  - CPU-bound: very long CPU bursts, infrequent I/O bursts
  - I/O-bound: short CPU bursts, frequent (long) I/O bursts
- During I/O bursts, CPU is not needed
  - Opportunity to execute another process!

What is processor scheduling?
- The allocation of processors to processes over time
- This is the key to multiprogramming
  - We want to increase CPU utilization and job throughput by overlapping I/O and computation
  - Mechanisms:
    - process states, process queues, context switch
What happens on dispatch/context switch?

- Switch the CPU to another process
  - Save currently running process state
    - Unless the current process is exiting
  - Select next process from ready queue
  - Restore state of next process
    - Restore registers
    - Switch to user mode
    - Set PC to next instruction in this process

What is processor scheduling?

- The allocation of processors to processes over time

  This is the key to multiprogramming
  - We want to increase CPU utilization and job throughput by overlapping I/O and computation

  Mechanisms:
  - process states, process queues, context switch

  Policies:
  - Given more than one runnable process, how do we choose which to run next?
  - When do we make this decision?

When to schedule?

- When the running process blocks (or exits)
  - Operating system calls (e.g., I/O)
  - Signals
- At fixed intervals
  - Clock interrupts
- When a process enters Ready state
  - I/O interrupts, signals, process creation

Scheduling Goals

- All systems
  - Fairness - each process receives fair share of CPU
  - Avoid starvation
  - Policy enforcement - usage policies should be met
  - Balance - all parts of the system should be busy

- Batch systems
  - Throughput - maximize jobs completed per hour
  - Turnaround time - minimize time between submission and completion
  - CPU utilization - keep the CPU busy all the time

More Goals

- Interactive Systems
  - Response time - minimize time between receiving request and starting to produce output
  - Proportionality - "simple" tasks complete quickly

- Real-time systems
  - Meet deadlines
  - Predictability

- Goals sometimes conflict with each other!

Types of Scheduling

- Non-preemptive scheduling
  - once the CPU has been allocated to a process, it keeps the CPU until it terminates
  - Suitable for batch scheduling

- Preemptive scheduling
  - CPU can be taken from a running process and allocated to another
  - Needed in interactive or real-time systems
Scheduling Algorithms: FCFS

- "First come, first served"
- Non-preemptive
- Choose the process at the head of the FIFO queue of ready processes

FCFS Example

<table>
<thead>
<tr>
<th>Process</th>
<th>Arrival Time</th>
<th>Service Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

- Note C waits six times as long as it runs!
- Total wait time is 8, avg. wait is 8/3 = 2.66

Problem with FCFS

- Average waiting time often quite long
  - convoy effect: all other processes wait for the one big process to release the CPU

What can we do to minimize wait times?

Algorithm: Shortest-Job-First

- Choose the process with the shortest processing time

SJF Example

<table>
<thead>
<tr>
<th>Process</th>
<th>Arrival Time</th>
<th>Service Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

- FCFS: Total wait time is 8, avg. wait is 8/3 = 2.66
- SJF: Total wait time is 4, avg. wait is 4/3 = 1.33

We cut wait time in half just by scheduling!