CPSC 457
Operating Systems

Lecture 6

The Rest of Scheduling Algorithms and The Beginning of Memory Management

Last Time

What we need to know about scheduling:

- When to schedule
- Preemptive vs Non-Preemptive
- Scheduling Goals and Metrics

Algorithms:

- First Come, First Served
- Shortest Job First
- Priority Scheduling

This Time

More Scheduling:

- Round Robin
- Multi-Queue Scheduling
  - Multi Level Feedback Queue
- And the rest?
- Windows & Linux

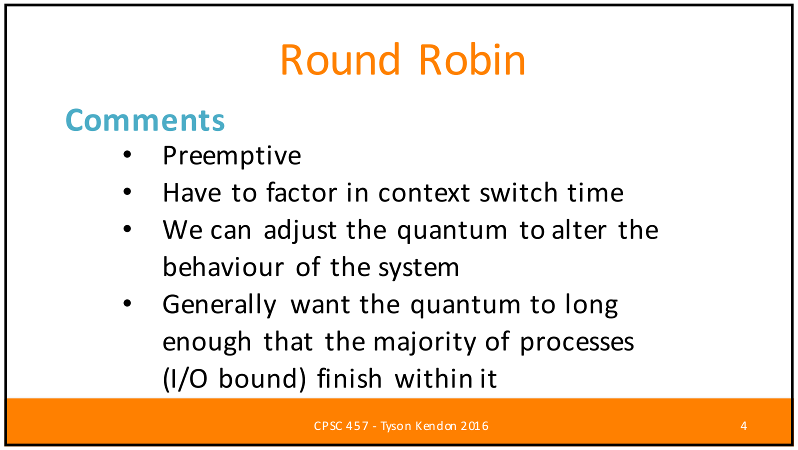
Memory Management:

- Introduction
  - Ideal Memory
- Memory Manager
- Why we need memory management

Round Robin

Algorithm

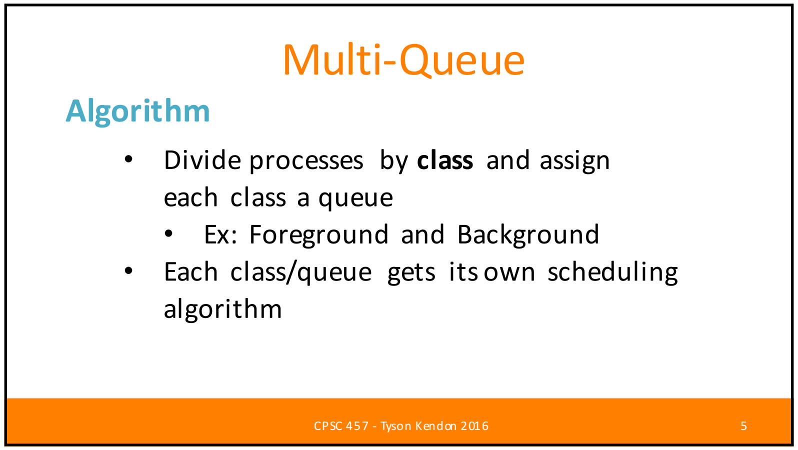
- Start with FCFS
- Preempt processes after a fixed amount of CPU time **(called the Time Quantum or the Time Slice)**
- If you stopped a process for using all of it's time, put it on the back of the queue.



Round Robin

Comments

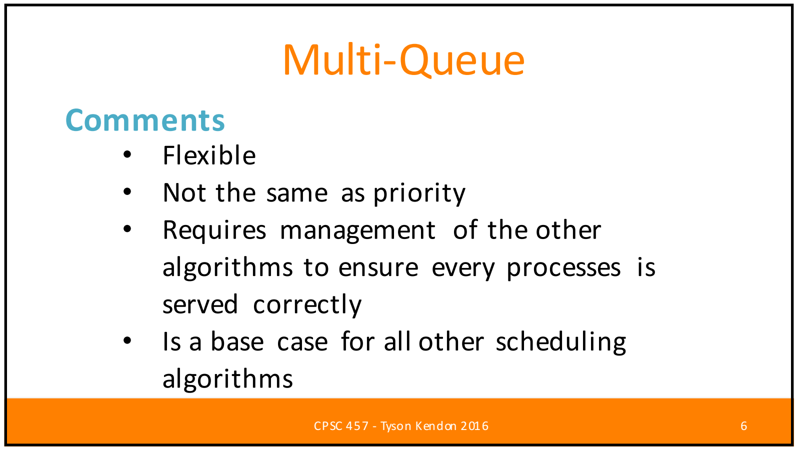
- Preemptive
- Have to factor in context switch time
- We can adjust the quantum to alter the behaviour of the system
- Generally want the quantum to long enough that the majority of processes (I/O bound) finish within it



Multi-Queue

Algorithm

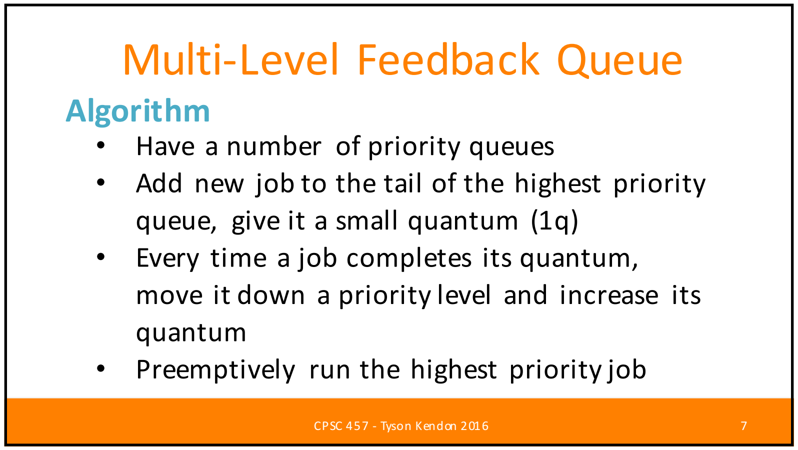
- Divide processes by **class** and assign each class a queue
  - Ex: Foreground and Background
- Each class/queue gets its own scheduling algorithm



Multi-Queue

Comments

- Flexible
- Not the same as priority
- Requires management of the other algorithms to ensure every processes is served correctly
- Is a base case for all other scheduling algorithms



Multi-Level Feedback Queue

Algorithm

- Have a number of priority queues
- Add new job to the tail of the highest priority queue, give it a small quantum (1q)
- Every time a job completes its quantum, move it down a priority level and increase its quantum
- Preemptively run the highest priority job

Multi-Level Feedback Queue

Comments

- Preemptive
- Automatically “sinks” CPU bound processes
- Fewer unnecessary context switches
- Is the basic approach for the “real world” algorithms

“And the Rest”

Guaranteed Scheduling

Lottery Scheduling

The Real World of Schedulers

Two Examples

Windows and Linux

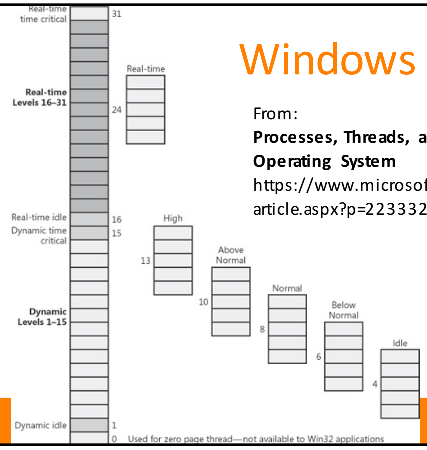
Complex

Different sets of things to run & different priorities

Windows

Dispatcher

Schedules Kernel Threads



Windows Priorities

From:

Processes, Threads, and Jobs in the Windows Operating System

https://www.microsoftpressstore.com/articles/article.aspx?p=2233328&seqNum=7

Windows Scheduling

Algorithm

- Keep 32 priority queues
- Use a bitmap to find the highest priority queue with a job, run that job
- Give each thread a quantum
  - small for desktop, large for server
- Lower priority for long running jobs
- Boost priority to keep the system interactive

Linux

Scheduler

- Standard *nix Scheduler
- The O(1) Scheduler
- Now the Completely Fair Scheduler

Schedules `tasks`

Completely Fair Scheduler

Algorithm

- Keep track of how much time a processes should run
- Keep a list (Red-Black Tree) ordered by time of how long each process has run
- Run the left most task, until it isn't the leftmost task any more

Scheduling

When do we choose to schedule and what are we trying to prioritize?

Different Algorithms give you different benefits

The real world is a little more complex than ideal, but we usually want to run I/O bound processes before CPU bound ones.

Memory Management

How can we create an abstract concept of memory that lets us pretend that all of the data our process may ever need can be accessed instantaneously?

Memory

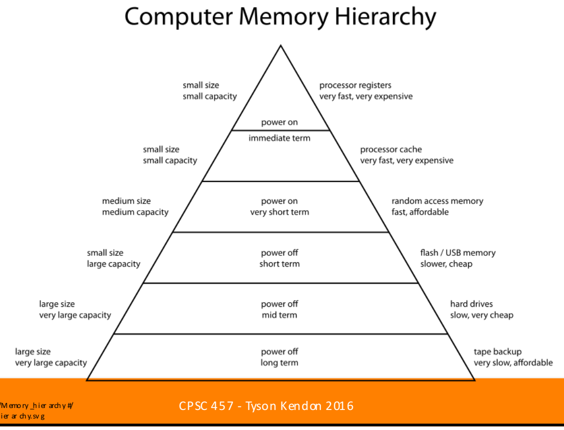
Ideally:

Every process has private, non-volatile, infinitely large, infinitely fast memory

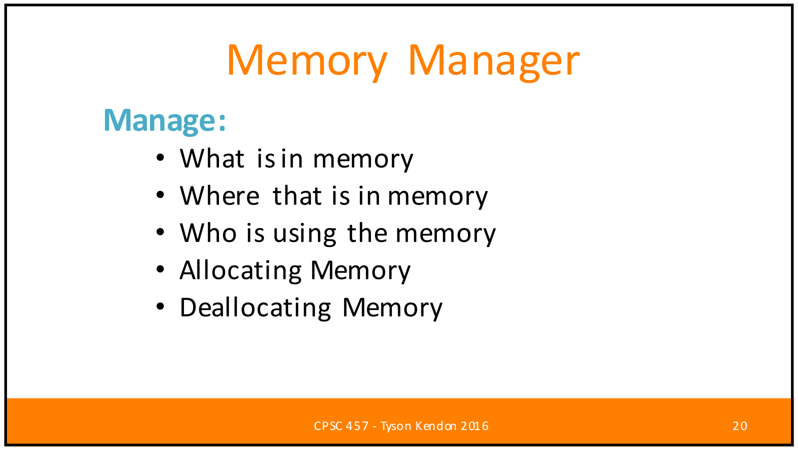
Reality:

Trade off:

- How much memory you can have
- How fast it can be
- How much it costs



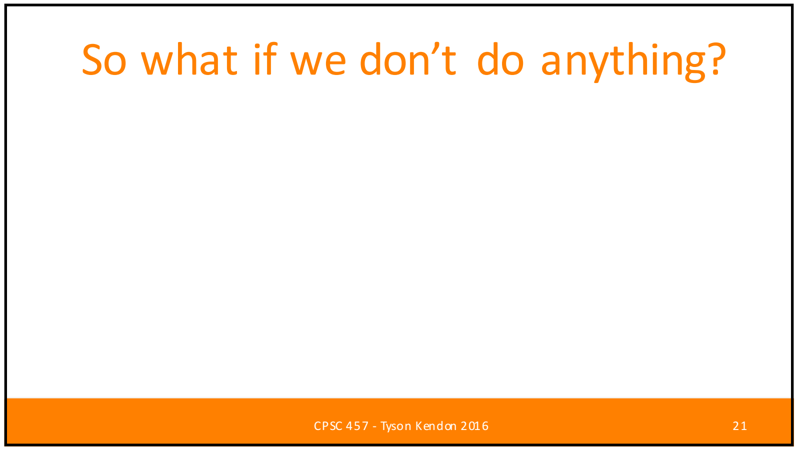
From:
Wikipedia
https://en.wikipedia.org/wiki/Memory_hierarchy#/media/File:ComputerMemoryHierarchy.svg



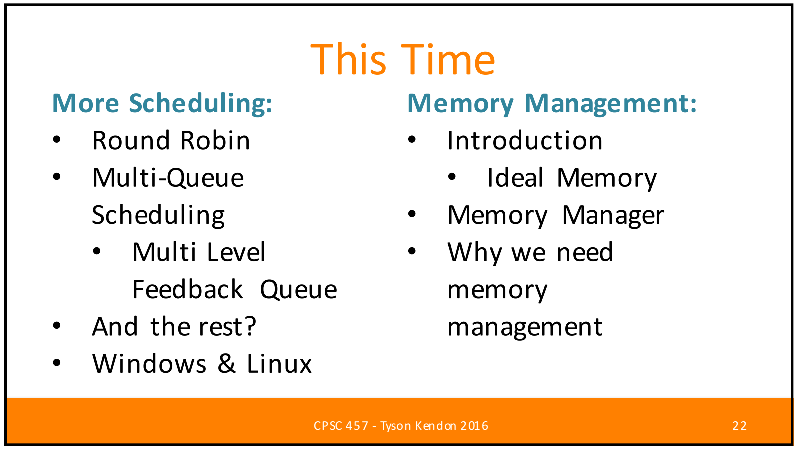
Memory Manager

Manage:

- What is in memory
- Where that is in memory
- Who is using the memory
- Allocating Memory
- Deallocating Memory



So what if we don't do anything?



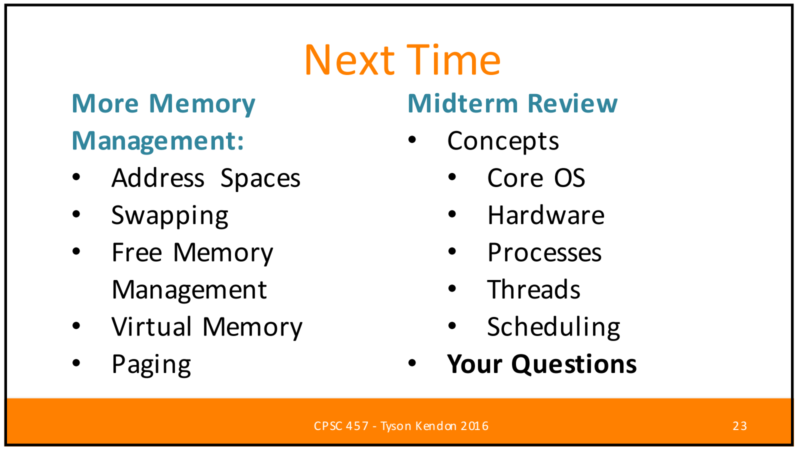
This Time

More Scheduling:

- Round Robin
- Multi-Queue Scheduling
  - Multi Level Feedback Queue
- And the rest?
- Windows & Linux

Memory Management:

- Introduction
  - Ideal Memory
- Memory Manager
- Why we need memory management



Next Time

More Memory Management:

- Address Spaces
- Swapping
- Free Memory Management
- Virtual Memory
- Paging

Midterm Review

- Concepts
  - Core OS
  - Hardware
  - Processes
  - Threads
  - Scheduling
- **Your Questions**