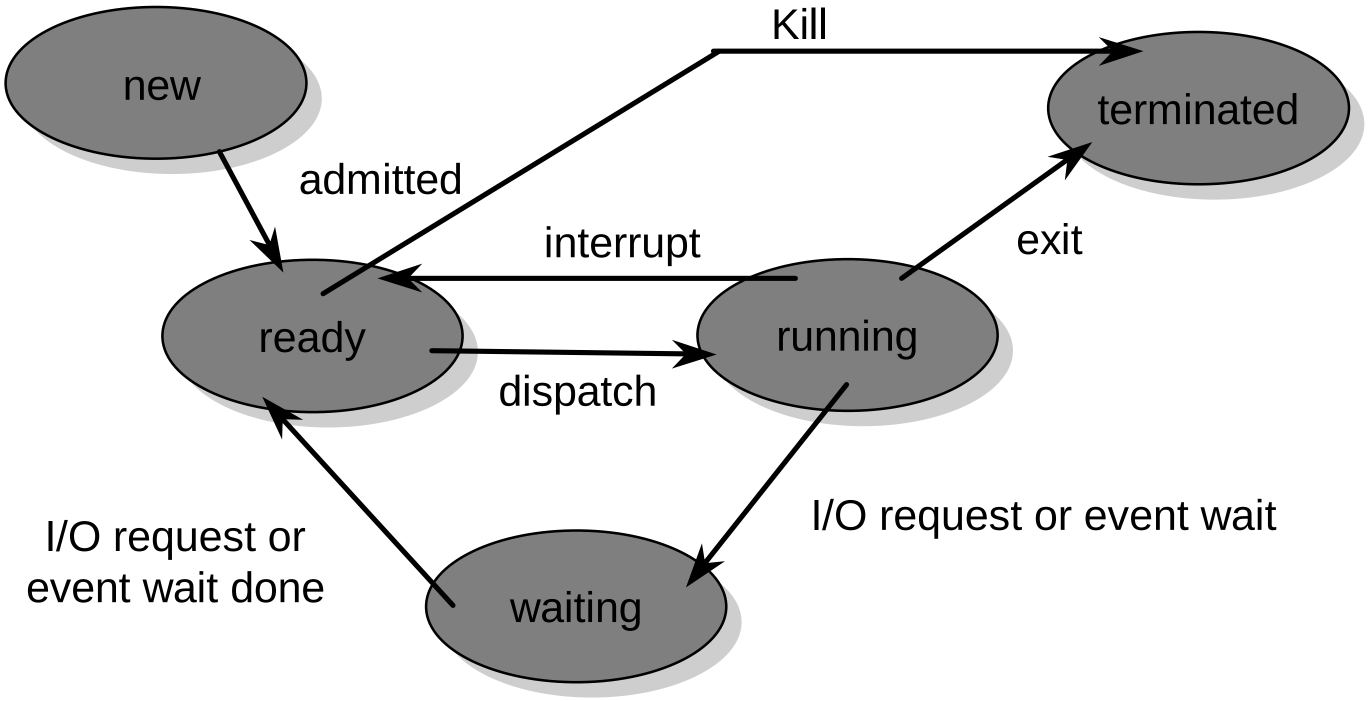
Announcements

- Program #1 is due next Monday
- Reading chapter 4 (4.5-4.6)

Process State

- Processes switch between different states based on internal and external events
- Each process is in exactly one state at a time
- Typical States of Processes (varies with OS)
  - New: The process is just being created
  - Running: Instructions are being executed
    - only one process per processor may be running
  - Waiting: The process is waiting for an event to occur
    - examples: I/O events, signals
  - Ready: The process is waiting to be assigned to a processor
  - Terminated: The process has finished execution

Process State Transitions



Components of a Process

- Memory Segments
  - Program - often called the text segment
  - Data - global variables
  - Stack - contains activation records
- Processor Registers
  - program counter - next instruction to execute
  - general purpose CPU registers
  - processor status word
    - results of compare operations
  - floating point registers

Process Control Block

- Stores all of the information about a process
- PCB contains
  - process state: new, ready, etc.
  - processor registers
  - Memory Management Information
    - page tables, and limit registers for segments
  - CPU scheduling information
    - process priority
    - pointers to process queues
  - Accounting information
    - time used (and limits)
    - files used
    - program owner
  - I/O status information
    - list of open files
    - pending I/O operations

Storing PCBs

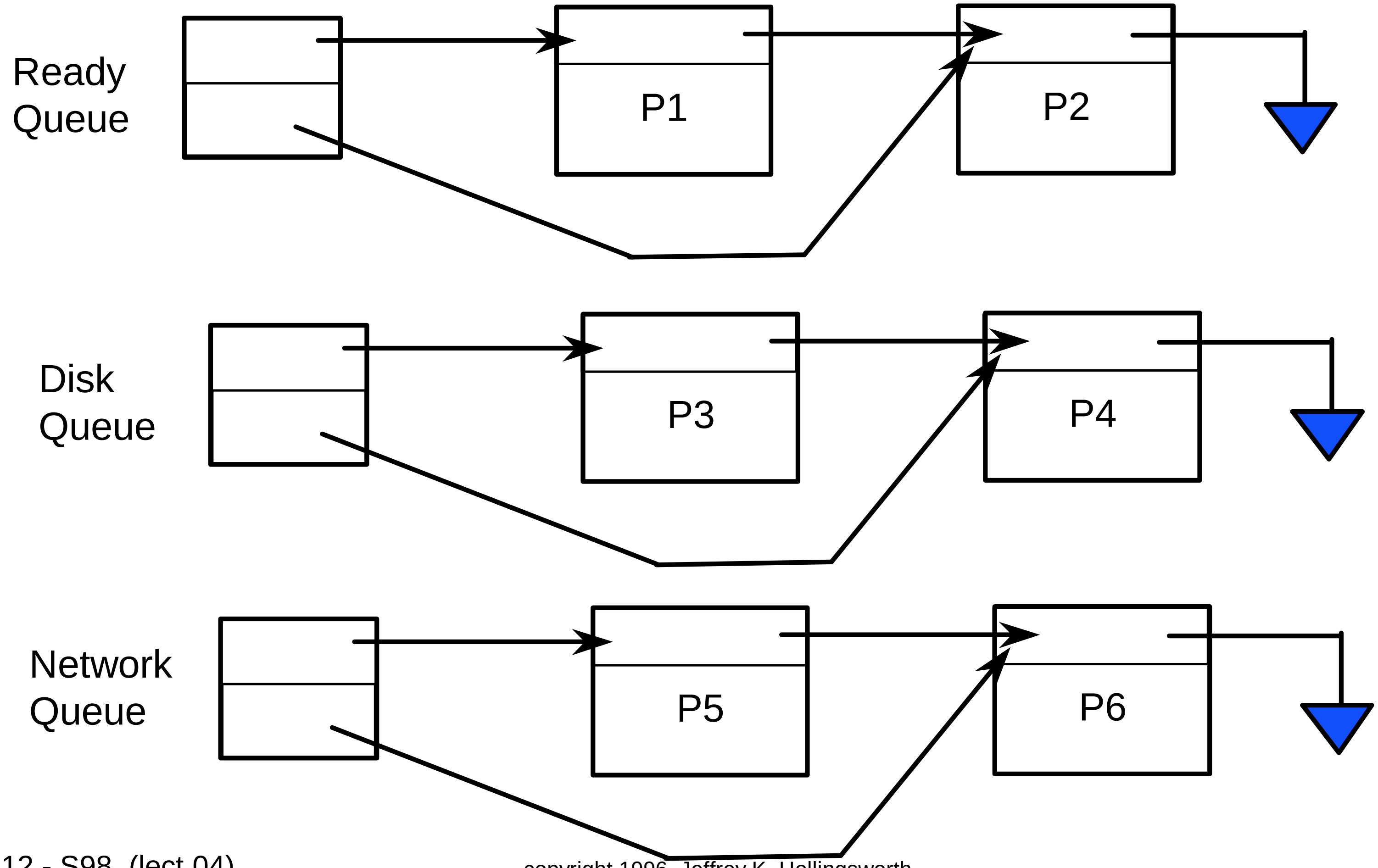
- Need to keep track of the different processes in the system
- Collection of PCBs is called a process table
- How to store the process table?
- First Option:

P1	P2	P2	P3	P4	P5
Ready	Waiting	New	Term	Waiting	Ready

- Problems with Option 1:
  - hard to find processes
  - how to fairly select a process

Queues of Processes

- Store processes in queues based on state



Dispatcher

- The inner most part of the OS that runs processes
- Responsible for:
  - saving state into PCB when switching to a new process
  - selecting a process to run (from the ready queue)
  - loading state of another process
- Sometimes called the short term scheduler
  - but does more than schedule
- Switching between processes is called context switching
- One of the most time critical parts of the OS
- Almost never can be written completely in a high level language

Selecting a process to run

- called scheduling
- can simply pick the first item in the queue
  - called round-robin scheduling
  - is round-robin scheduling fair?
- can use more complex schemes
  - we will study these in the future
- use alarm interrupts to switch between processes
  - when time is up, a process is put back on the end of the ready queue
  - frequency of these interrupts is an important parameter
    - typically 3-10ms on modern systems
    - need to balance overhead of switching vs. responsiveness

Process Priority

- Use multiple run queues, one for each priority
- Who decides priority
  - dispatcher - that mixes policy and mechanism too much
  - when the process is created, assign it a priority
  - have a second level scheduler (often called medium term scheduler) to manage priorities
    - mechanism is to move processes between different queues
- Will discuss scheduling more in a future lecture

Process Creation

- Who creates processes?
  - – answer: other processes
  - – operations is called fork (or spawn)
  - – what about the first process?
- Have a tree of processes
  - • parent-child relationship between processes
- what resources does the child get?
  - • new resources from the OS
  - • a copy of the parent resources
  - • a subset of the parent resources
- What program does the child run?
  - • a copy of the parent (UNIX fork)
    - – a process may change its program (execve call in UNIX)
  - • a new program specified at creation (VMS spawn)