CS571-001: Operating Systems

Some Observations and Comments

Some Operating System Examples

- CP/M, MS-DOS
- MS-Windows 3.1, 95, 98, ME, CE, 2K, XP, ...
- VMS
- UNIX (BSD, SysV, Ultrix, OSF/1, Solaris, AIX, HP-UX, Tru64, IRIX, OS X, ...)
- Linux
- PalmOS
- Can you name others? Why are there so many?
- Are operating systems still important?
- Will new operating systems evolve?

What does an OS do?

- Essential:
  - Manages the execution of programs
  - Provides interface & control of processor and connected devices
  - Manages limited resources
- Optional:
  - Manages sharing of limited resources
  - User interface: CLI, GUI

Resources? What resources?

- CPU: single or multiple (SMP)
- Memory: on-chip cache, off-chip cache, DRAM, virtual memory
- I/O: data paths, buffers (caches), queueing requests, file system actions
- Devices: I/O controllers, comm devices, video, sound, keyboard, mouse, ...
- Security User identification, authentication, authorization

Basic OS Questions

- * Why are there OS limits on
  - - Memory size? Process size?
  - - Speed of execution? Speed of data transfer?
  - -Number of processes, files, sockets, objects, ...
- * ...and how do we work within those limits?
- * How do we
  - - Represent, measure, find, change, <u>share</u>, verify, monitor, control, detect?
- * How does an OS
  - - Respond to stress or resource exhaustion?

Relative Access times

Device	Real time	Relative time (if 1 s=1 ns)
CPU Register	2 nsec	2 seconds
CPU cache	20 nsec	20 second
Main Memory	200 nsec	2-3 minutes
Disk	20 msec	7 months

Consequences of Size Decisions:
Important Powers of 2

- $2^{8} = 256$
  - Byte = 8 bits
- $2^{16} = 65,536$
  - Original IBM-PC 8086
- $2^{32} = 4,294,967,296$
  - Sun SPARC , Intel Pentium, IPv4, Jan 19 2038
- $2^{64} = 18,446,744,073,709,551,616$
  - Sun UltraSPARC, AMD Opteron, Intel Itanium, DEC/Compaq/HP Alpha
- $2^{128} = 3.4 \times 10^{38}$
  - IPv6, cryptography

Chapter 4: Processes

- Process Concept
- Process Scheduling
- Operations on Processes
- Cooperating Processes
- Interprocess Communication
- Communication in Client-Server Systems

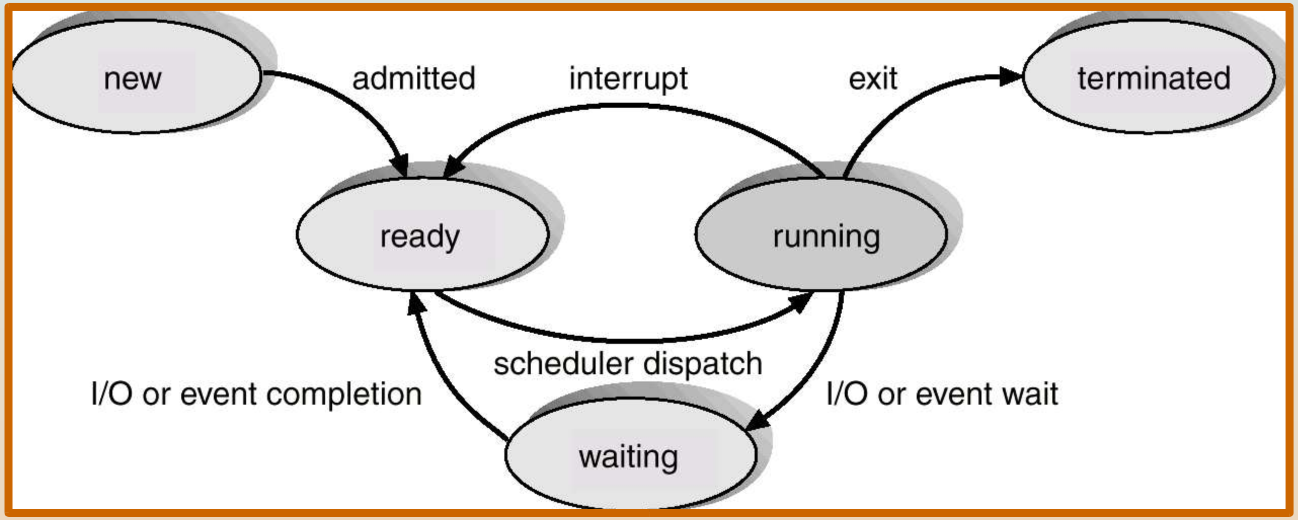
Process Concept

- An operating system executes a variety of programs:
  - Batch system – jobs
  - Time-shared systems – user programs or tasks
- Textbook uses the terms job and process almost interchangeably
- Process – a program in execution; process execution must progress in sequential fashion
  - is this true?
    - “out of order execution”
- A process includes:
  - program counter
  - stack
  - data section

Process State

- As a process executes, it changes state
  - new: The process is being created
  - running: Instructions are being executed
  - waiting: The process is waiting for some event to occur
  - ready: The process is waiting to be assigned to a process
  - terminated: The process has finished execution

Diagram of Process State

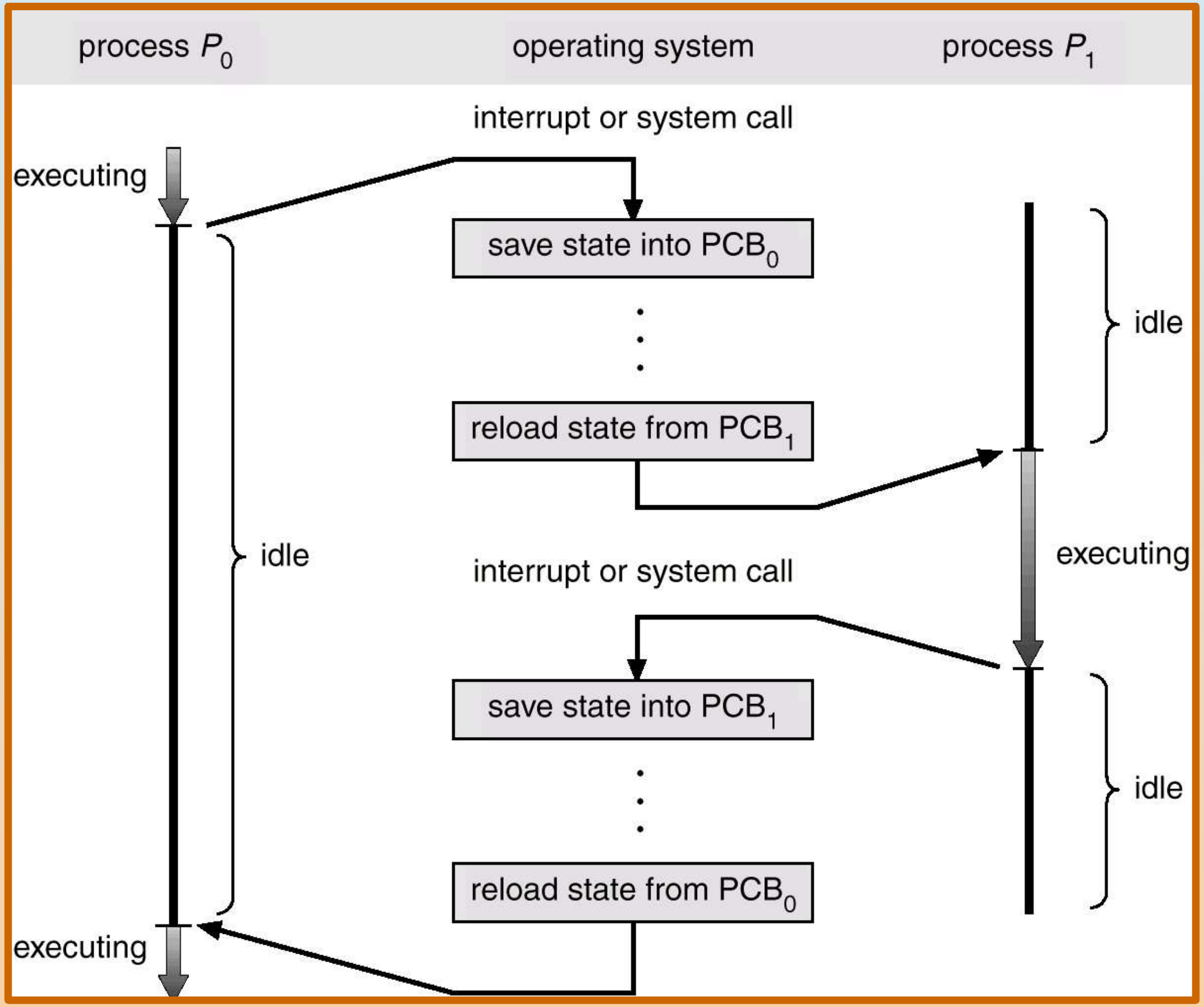


Process Control Block (PCB)

Information associated with each process

- Process state
- Program counter
- CPU registers
- CPU scheduling information
- Memory-management information
- Accounting information
- I/O status information

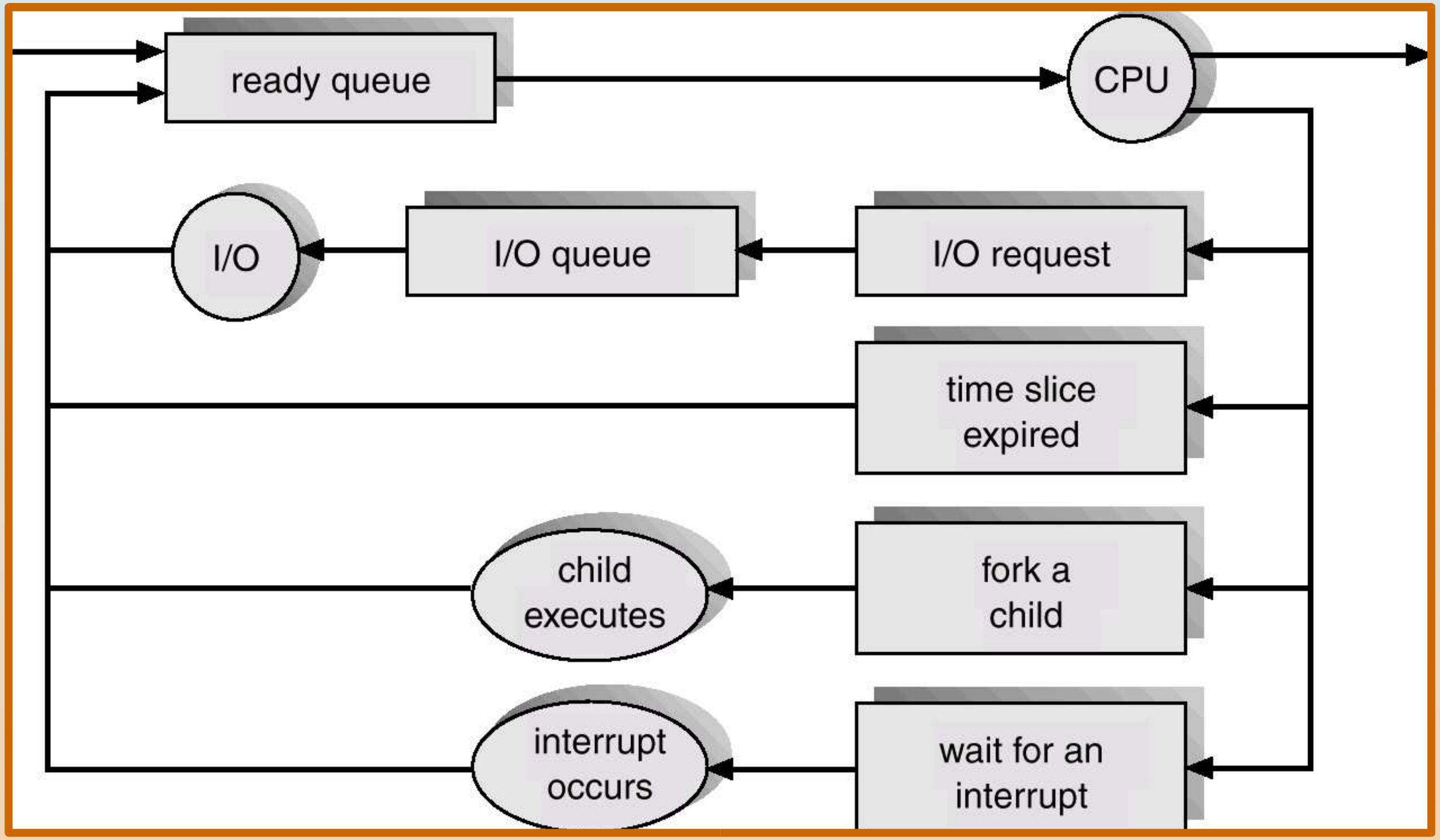
CPU Switch From Process to Process



Process Scheduling Queues

- Job queue – set of all processes in the system
- Ready queue – set of all processes residing in main memory, ready and waiting to execute
- Device queues – set of processes waiting for an I/O device
- Process migration between the various queues

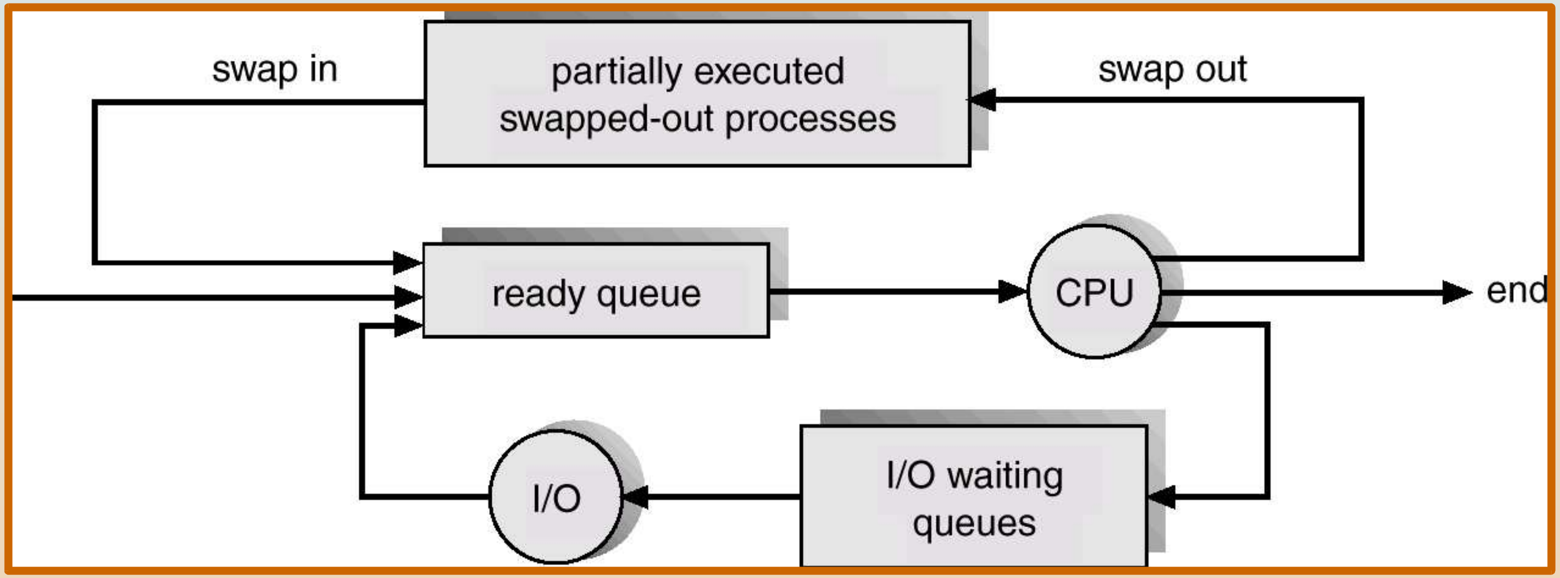
Representation of Process Scheduling



Schedulers

- Long-term scheduler (or job scheduler) – selects which processes should be brought into the ready queue
- Short-term scheduler (or CPU scheduler) – selects which process should be executed next and allocates CPU

Addition of Medium Term Scheduling



Schedulers (Cont.)

- Short-term scheduler is invoked very frequently (milliseconds) (must be fast)
- Long-term scheduler is invoked very infrequently (seconds, minutes) (may be slow)
- The long-term scheduler controls the degree of multiprogramming
- Processes can be described as either:
  - I/O-bound process – spends more time doing I/O than computations, many short CPU bursts
  - CPU-bound process – spends more time doing computations; few very long CPU bursts

Context Switch

- When CPU switches to another process, the system must save the state of the old process and load the saved state for the new process
- Context-switch time is overhead; the system does no useful work while switching
- Time dependent on hardware support

Observing Processes (UNIX)

- How do we know what's happening in a system?
- Various tools to monitor processes, resource consumption
  - top, ps, vmstat, iostat, prstat, mpstat, ...
  - see m a n pages for documentation and options
  - see U N I X Rosetta Stone for equivalents

Homework 1

- Textbook
  - Read Chapters 1-3
    - Especially 2.4.1, 2.4.2
      - Answer Question 2.9, pg 56
    - Especially 3.6, 3.7
      - Answer Question 3.18, pg 99
  - Read Chapter 4
- Review Text Web site
- Next class: Chapter 5, Threads