BUCKNELL UNIVERSITY
Computer Science
CSCI 315 Operating Systems Design

Processes

Notice: The slides for this lecture have been largely based on those accompanying an earlier version of the textbook *Operating Systems Concepts with Java*, by Silberschatz, Galvin, and Gagne. Many, if not all, the illustrations contained in this presentation come from this source.
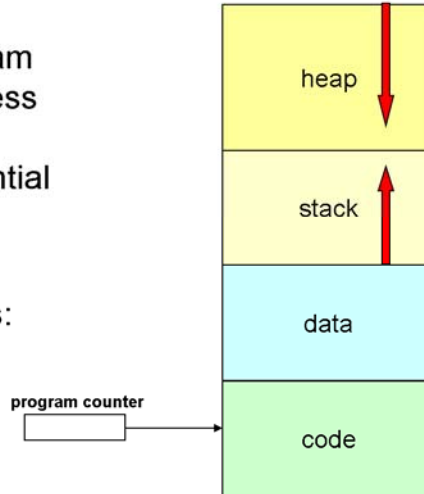
Chapter 3

Processes

Process Concept

- Process – a program in execution; process execution must progress in sequential fashion.

- A process includes:
  - program counter,
  - stack,
  - data section.

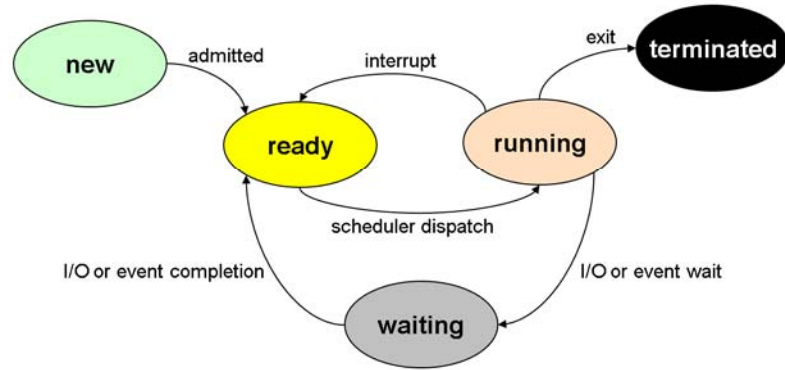| heap |
| --- |
| stack |
| data |
| code |

program counter

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 processor.
- **terminated**: The process has finished execution.

4

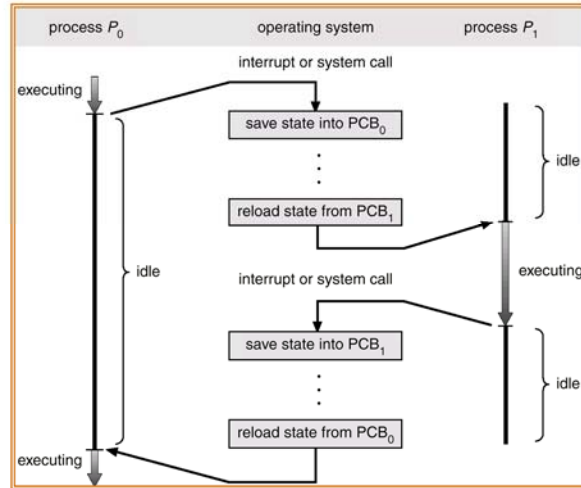# Process State Transition Diagram

5

Process Control Block (PCB)

OS bookkeeping information
associated with each process:

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

process id
process state
program counter
registers
memory limits
list of open files
⋮

CPU Switching

process P_0	operating system	process P_1
	interrupt or system call	

executing

save state into PCB_0

⋮

reload state from PCB_1

idle

interrupt or system call

executing

save state into PCB_1
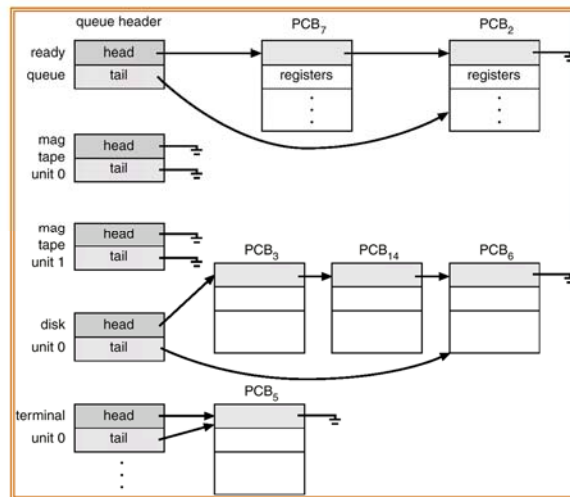
⋮

reload state from PCB_0

executing

idle

idle

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.

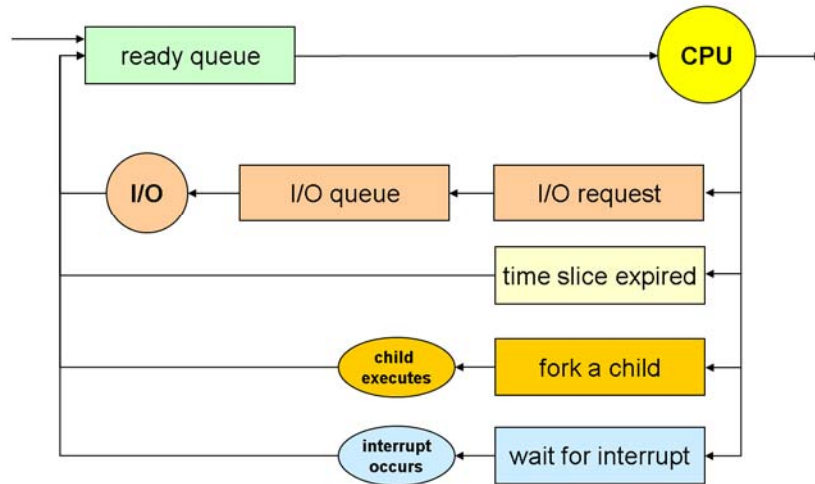Processes migrate between the various queues.

Processes and OS Queues

Process Scheduling

ready queue → CPU

I/O ← I/O queue ← I/O request

time slice expired

child executes ← fork a child

interrupt occurs ← wait for interrupt