# Processes

Eike Ritter

Modified: October 9, 2012

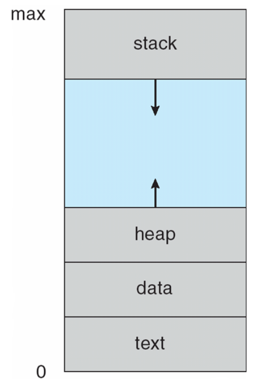
Lecture 6: Operating Systems with C/C++
School of Computer Science, University of Birmingham, UK

## Outline

## Process Concept

- An operating system executes a variety of programs:
  - Batch system - jobs
  - Time-shared systems - user programs or tasks
- Process - a program in execution; process execution must progress in sequential fashion
- A process includes:
  - program counter
  - stack
  - data section

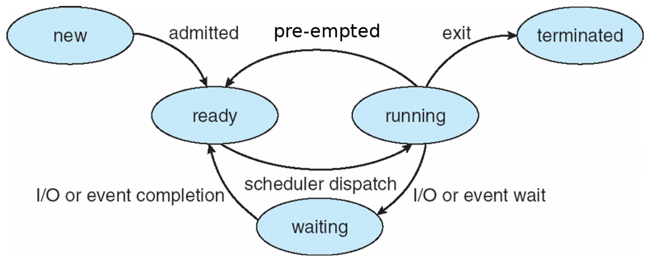
## Process in Memory



## Process States

- 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

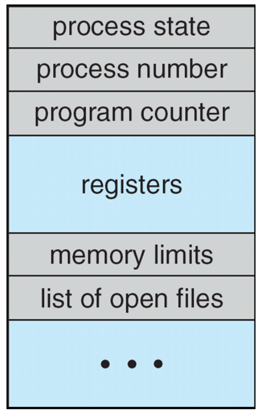
## Process States



## Process Control Block

- Information associated with each process, which is stored as various fields within a kernel data structure:
  - Process state
  - Program counter
  - CPU registers
  - CPU scheduling information
  - Memory-management information
  - Accounting information
  - I/O status information

## Process Control Block



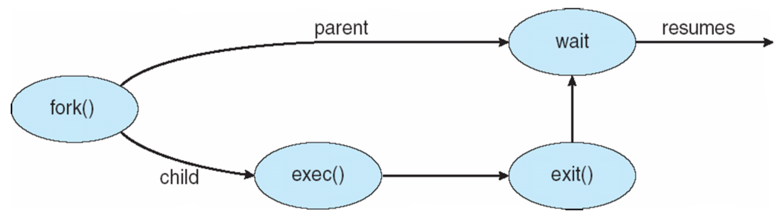
## Process Creation

- Parent process create children processes, which, in turn create other processes, forming a tree of processes
- Generally, process identified and managed via a process identifier (pid)
- Resource sharing
  - Parent and children share all resources
  - Children share subset of parent's resources
  - Parent and child share no resources
- Execution
  - Parent and children execute concurrently
  - Parent waits until children terminate

## Process Creation

- Address space
  - Child duplicate of parent
  - Child has a program loaded into it
- UNIX examples
  - `fork` system call creates new process
    - will look at fork soon, in one of our practical lectures.
  - exec system call used after a fork to replace the process' memory space with a new program

## Process Creation



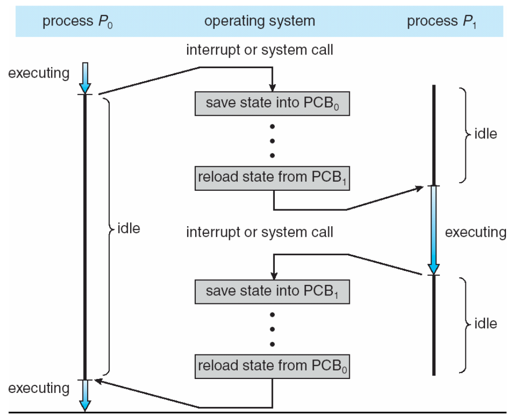
## Process Termination

- Process executes last statement and asks the operating system to delete it (exit)
  - Output data from child to parent (via wait)
  - Process' resources are deallocated by operating system
- Parent may terminate execution of children processes (abort)
  - Child has exceeded allocated resources
  - Task assigned to child is no longer required
  - If parent is exiting:
    - Some operating systems do not allow child to continue if its parent terminates — all children terminated (*i.e.* cascading termination)

## 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 via a context switch
- Context of a process represented in the PCB
- Context-switch time is overhead; the system does no useful work while switching
- Time dependent on hardware support

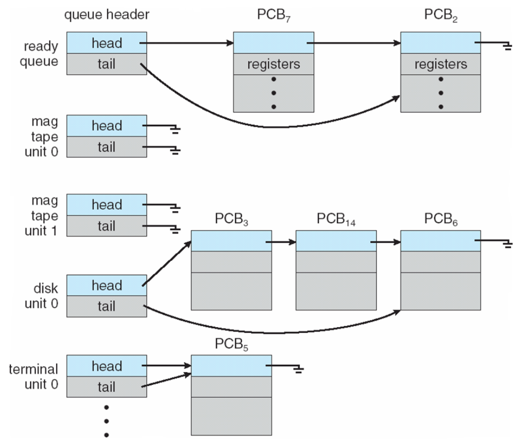
## Context Switch



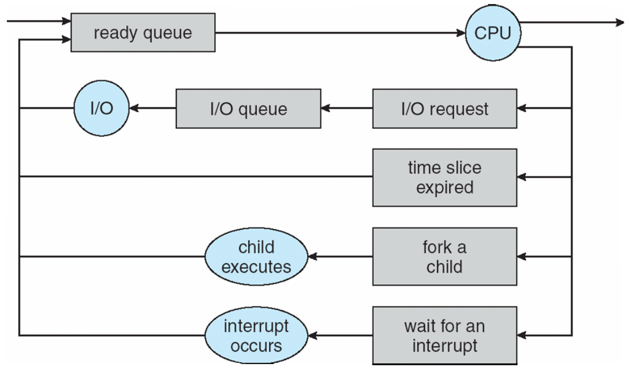
## Process Scheduling Queues

- 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 among the various queues

## Process Scheduling Queues



## Scheduling Workflow



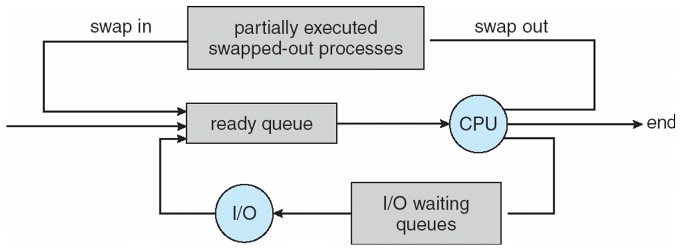
## Schedulers

- Long-term scheduler (or job scheduler) - selects which processes should be brought into the ready queue (*e.g.* loaded from the disk into memory)
- Short-term scheduler (or CPU scheduler) - selects which process should be executed next and allocates CPU

## Schedulers

- 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 (*i.e.* how many processes may compete for the CPU)
  - Long-term scheduling is often minimal or absent on mainstream operating systems, such as Windows and Linux.
- 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
- We will look at the problem problem of scheduling in a later lecture.

## Addition of Medium Term Scheduling



- To reduce contention among ready processes, in some scheduling designs, medium-term scheduling allows some processes to be temporarily swapped out of memory
  - kind of like they are being told to sit on a bench until things quieten down.

## Summary

We looked at:

1. Process Concept
   - What is a process?
   - Process States and Control Block
   - Process Creation and Termination
2. Concurrency Through Context Switching
   - Context Switching
3. Overview of Process Scheduling
   - Scheduling Queues and Workflow
   - Long-, Short-, and Medium-term scheduling.