



Information Technology Institute



Operating System Fundamentals

Chapter Four

PROCESSES

Table of Content

- Process Concept
- Process Scheduling
- Operations on Processes
- Cooperating Processes
- Inter-process Communication

PROCESS CONCEPT

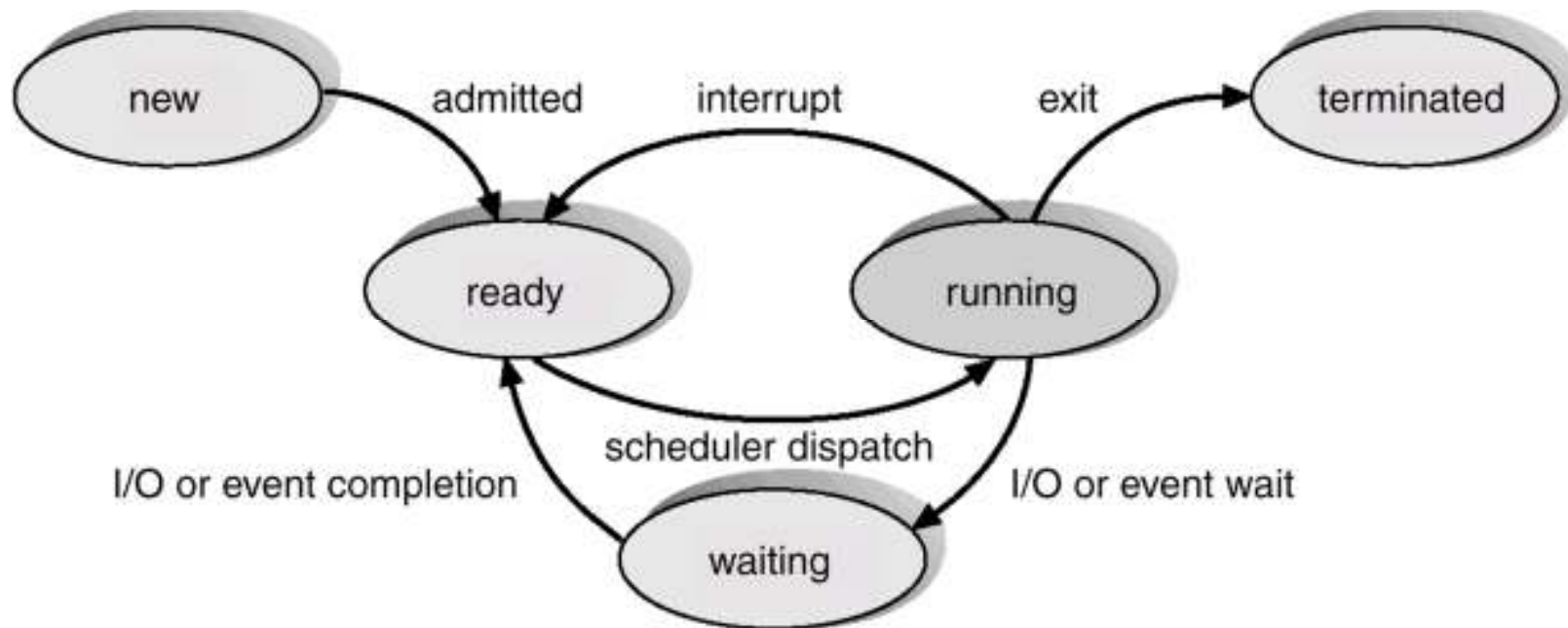
Process Concept

- Process – a program in execution; process execution must progress in sequential fashion.
 - An operating system executes a variety of programs:
 - Batch system
 - Time-shared systems
- * Textbook uses the terms job and process almost interchangeably

Process Contents

- **Text section**
 - Program instructions
- **Program counter**
 - Next instruction
- **Stack**
 - Local variables
 - Return addresses
 - Method parameters
- **Data section**
 - Global variables

Process State



Process State Cont'd

- 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.

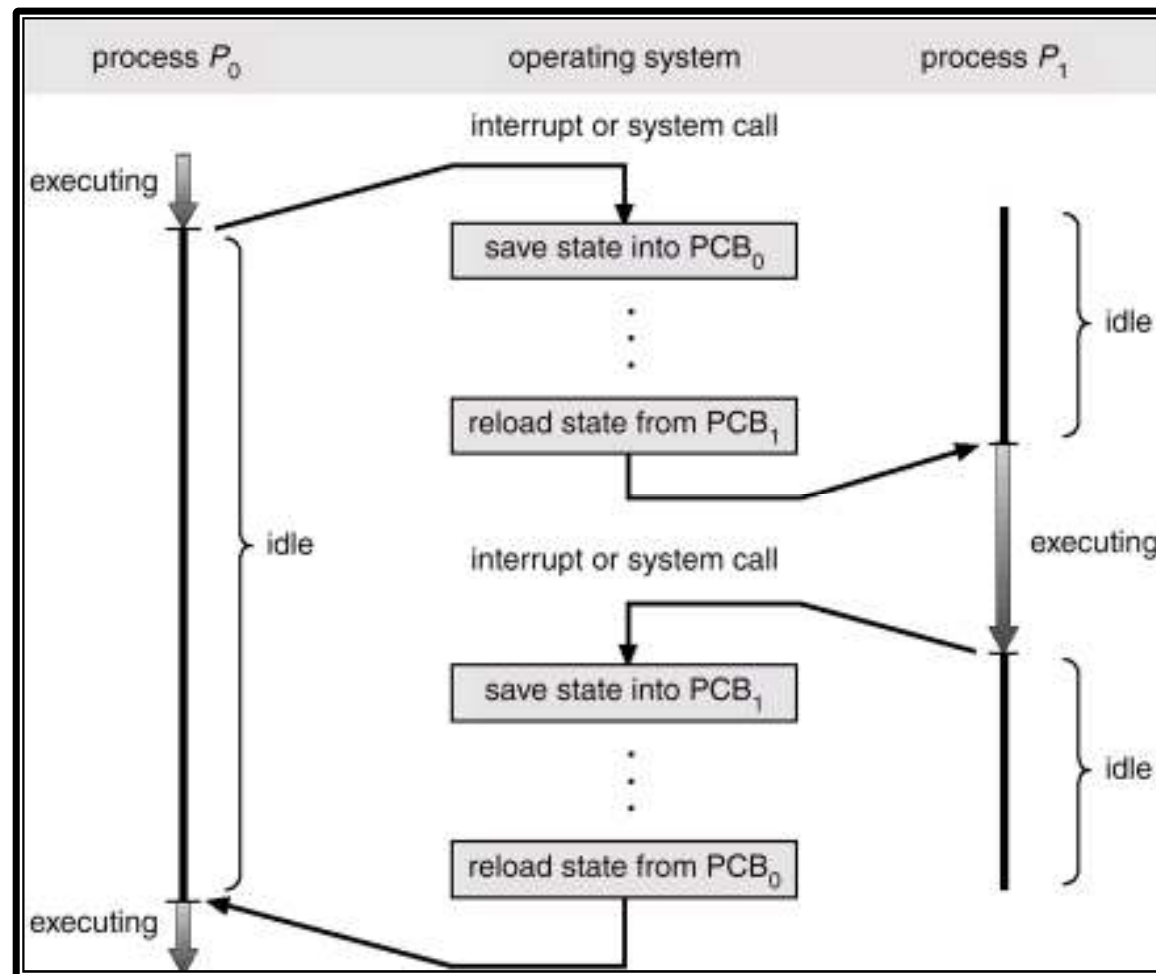
Process Control Block (PCB)

- Information associated with each process.
 - Process state
 - Program counter
 - CPU registers
 - CPU scheduling information
 - Memory-management information
 - Accounting information
 - I/O status information

Process Control Block (PCB) Cont'd

Pointer	Process state
Process number	
Program counter	
CPU registers	
Memory management info	
I/O status information	
Accounting Information	

CPU Switch From Process to Process



PROCESS SCHEDULING

Process Scheduling

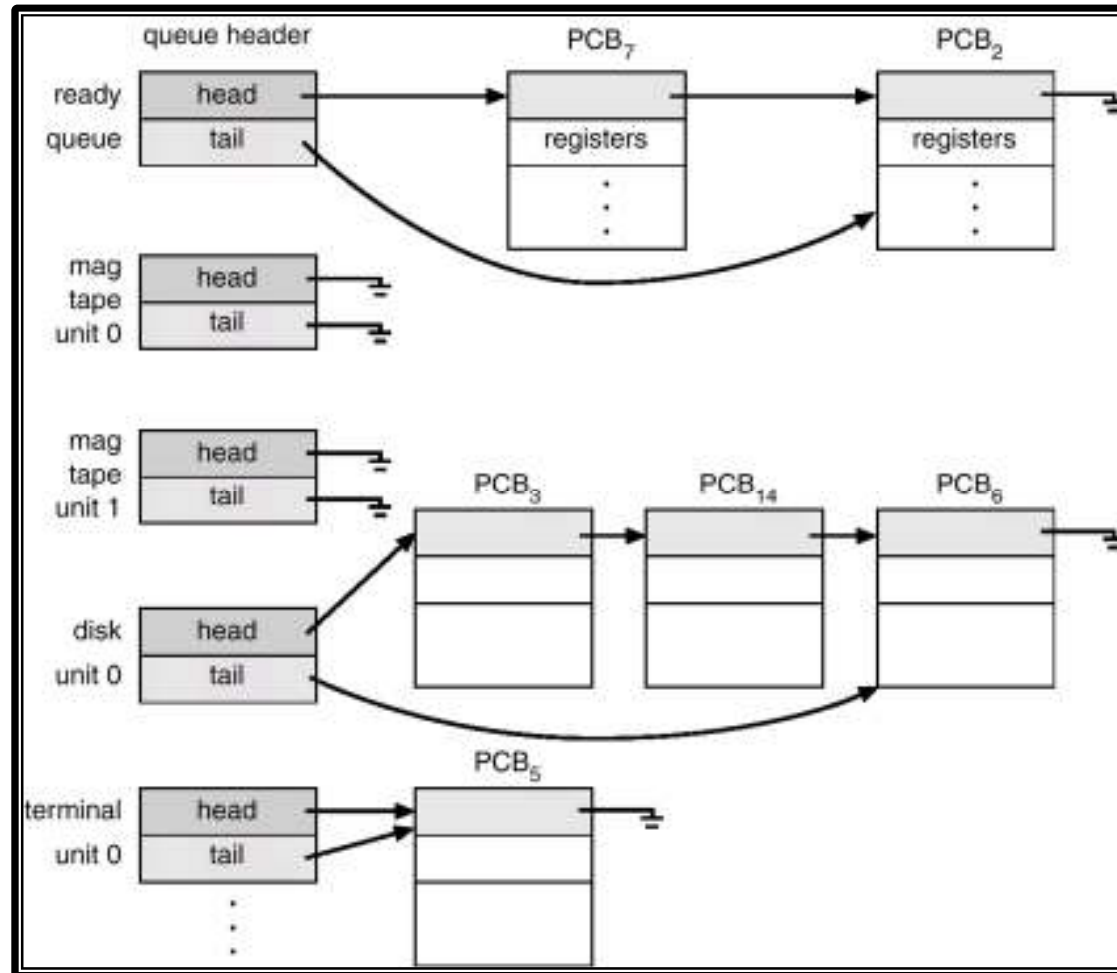
- **Multi-Programming systems**
 - Some processes executing at all times
 - Maximize CPU utilization
- **Time-Sharing systems**
 - Switch the CPU among processes frequently
 - Users can interact with a program while it's executing
 - Virtually run processes at the same time

Process Scheduling Queues

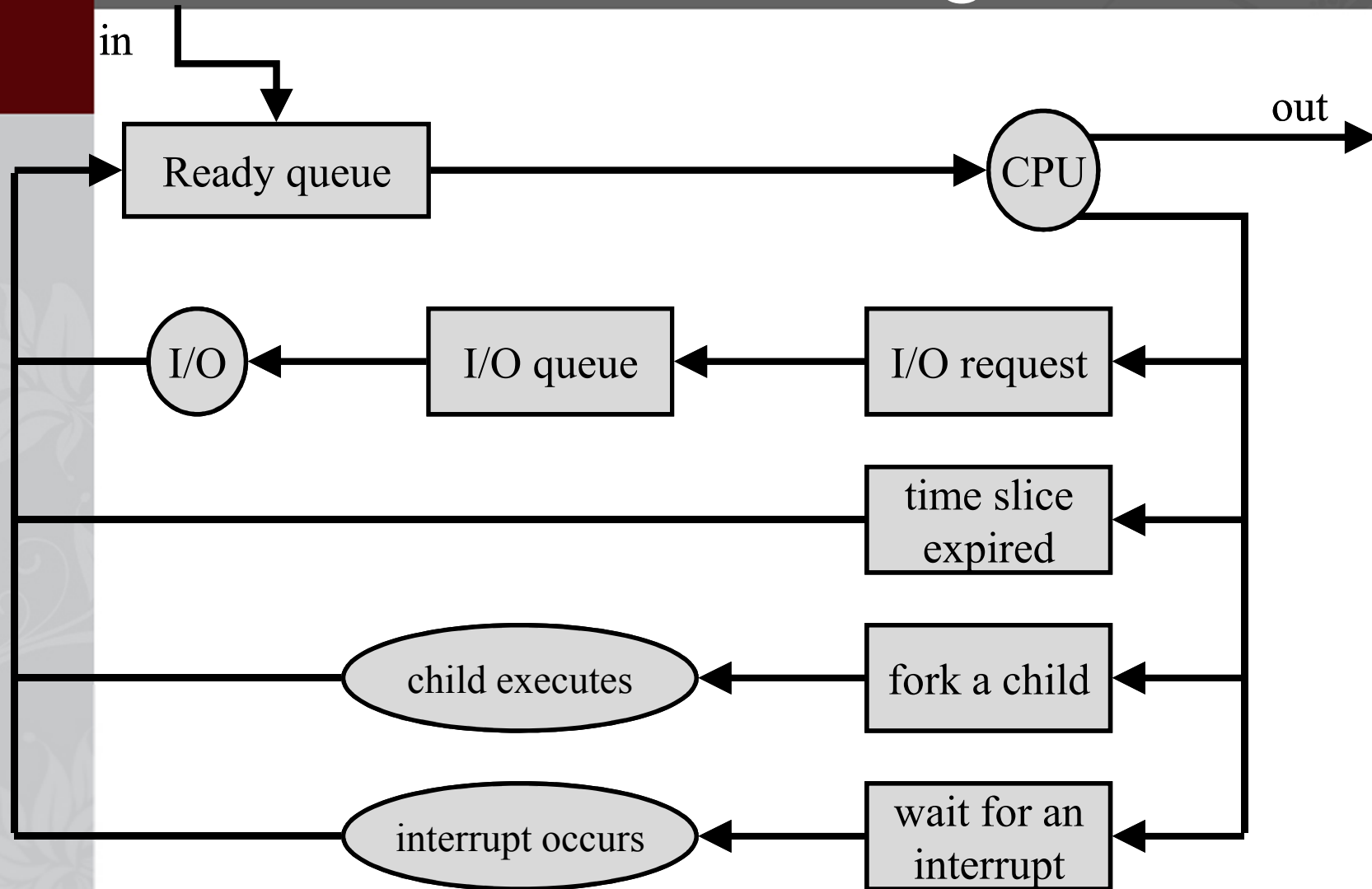
- **Job queue**
 - set of all processes in the system.
- **Ready queue**
 - set of all processes residing in main memory which is ready and waiting to execute.
- **Device queues**
 - set of processes waiting for an I/O device.

* Process migration between the various queues

Ready Queue And Various I/O Device Queues



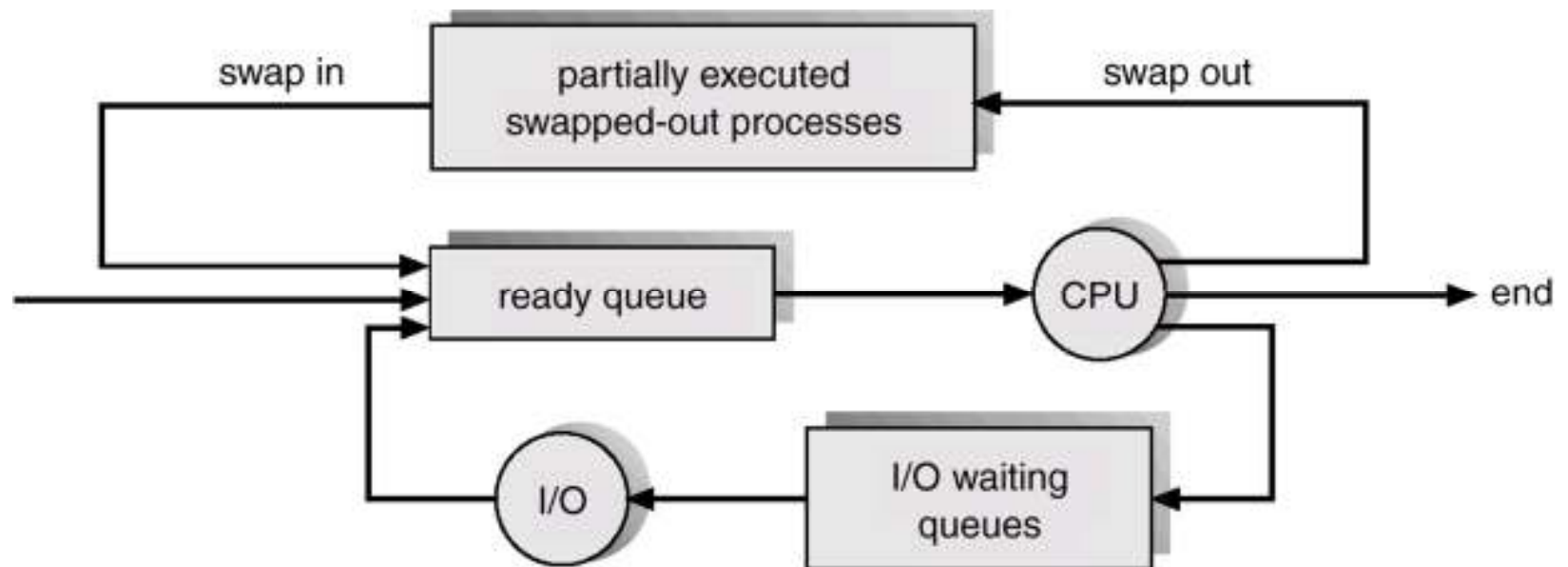
Representation of Process Scheduling



Schedulers

- **Long-term scheduler (or job scheduler)**
 - Selects which processes should be brought into the ready queue.
 - Executes infrequently
 - May be absent in some O/S
- **Short-term scheduler (or CPU scheduler)**
 - Selects which process should be executed next and allocates CPU.
 - Executes frequently

Addition of Medium Term Scheduling



Schedulers Cont'd

- 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.
- Proper System performance
 - Mix of CPU & I/O bound processes
- Improper system performance
 - All processes are I/O bound
 - All processes are CPU bound

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.

OPERATIONS ON PROCESSES

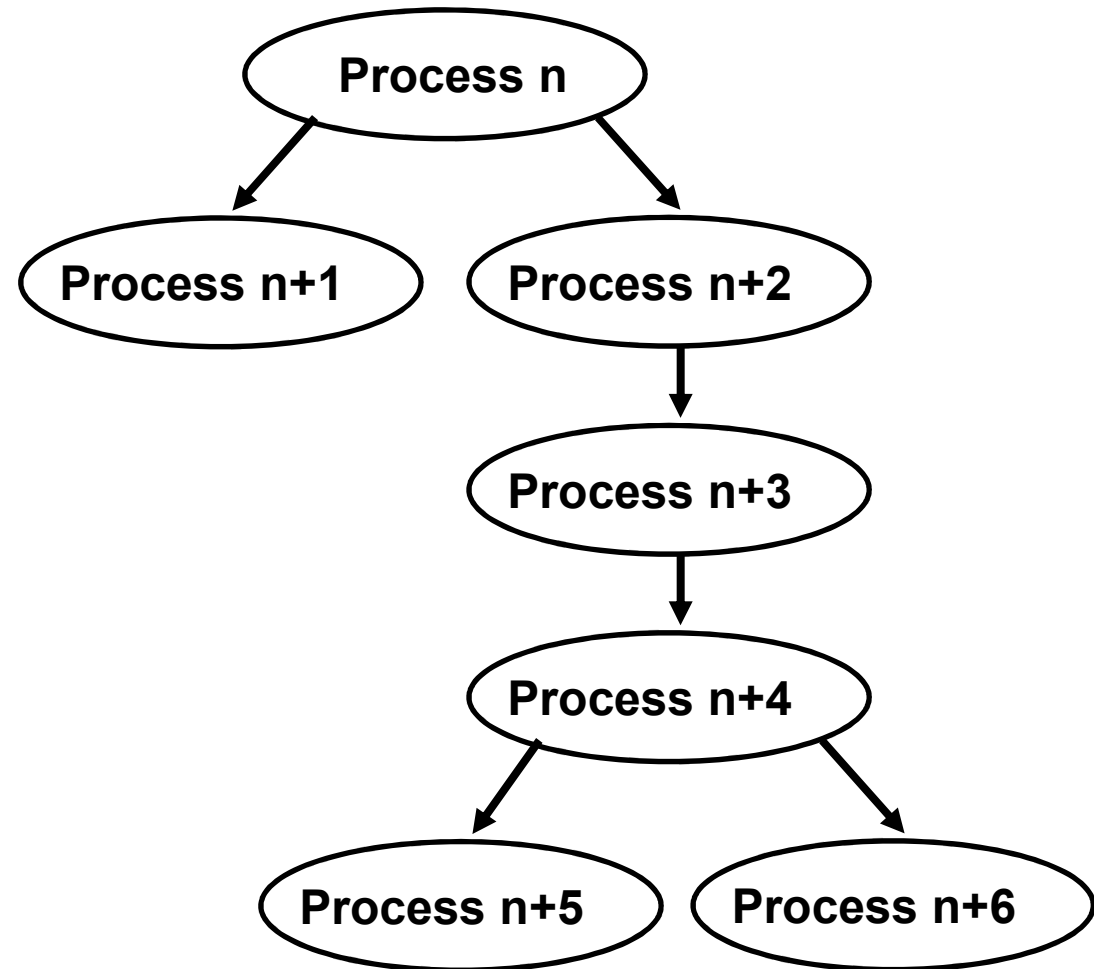
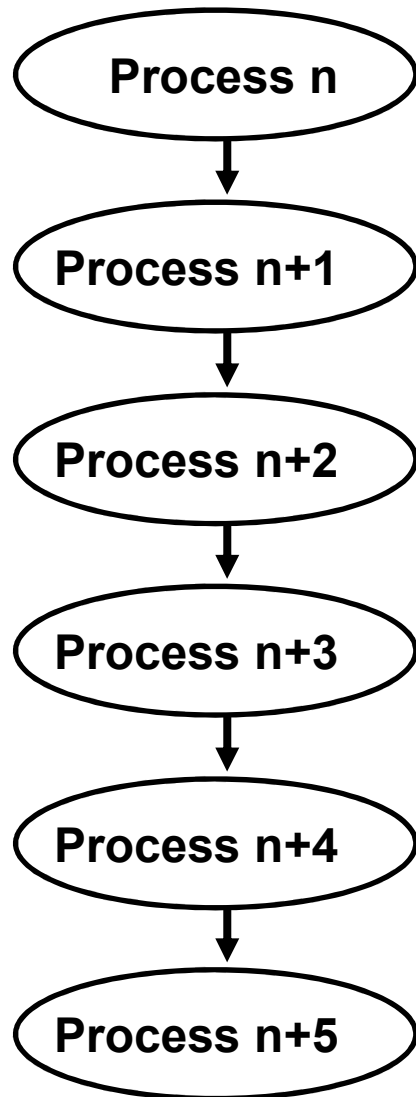
Operations on Processes

- Process Creation
- Process Termination

* Process resources

- CPU time
- Memory
- Files
- I/O devices

Process Creation



Process Creation Cont'd

- Parent process create children processes, which, in turn create other processes, forming a tree of processes.
- Resource sharing
 - Parent and children share all resources.
 - Children share subset of parent's resources.
- Execution
 - Parent and children execute concurrently.
 - Parent waits until children terminate.

Process Creation Cont'd

- **Address space**
 - Child duplicate of parent.
 - Child has a program loaded into it.
- **UNIX examples**
 - **fork** system call creates new process
 - **exec** system call used after a **fork** to replace the process' memory space with a new program.

Process Termination

- Process executes last statement and asks the operating system to decide it (**exit**).
 - Output data from child to parent.
 - Process' resources are de-allocated by operating system.
- Parent may terminate execution of children processes (**abort**).
 - Child has exceeded allocated resources.
 - Task assigned to child is no longer required.
 - Parent is exiting.
 - Operating system does not allow child to continue if its parent terminates.
 - Cascading termination.

Cooperating Processes

- *Independent* process cannot affect or be affected by the execution of another process.
- *Cooperating* process can affect or be affected by the execution of another process
- Advantages of process cooperation
 - Information sharing
 - Computation speed-up
 - Modularity
 - Convenience

INTER-PROCESS COMMUNICATION

Inter-process Communication

- Message Passing
- Shared Memory
- * Synchronization

Message Passing

- **Messages**
 - Fixed size
 - Variable size
- **Communication**
 - Direct:
 - `send(P2, message), receive(P1, message)`
 - Indirect:
 - `send(ID, message), receive(ID, messages)`

Synchronization

- Message passing may be either blocking or non-blocking.
- **Blocking** is considered **synchronous**
- **Non-blocking** is considered **asynchronous**
- **send** and **receive** primitives may be either blocking or non-blocking.

