RAMA UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF ENGINEERING & TECHNOLOGY

BCS-501 Operating System

Lecturer-07

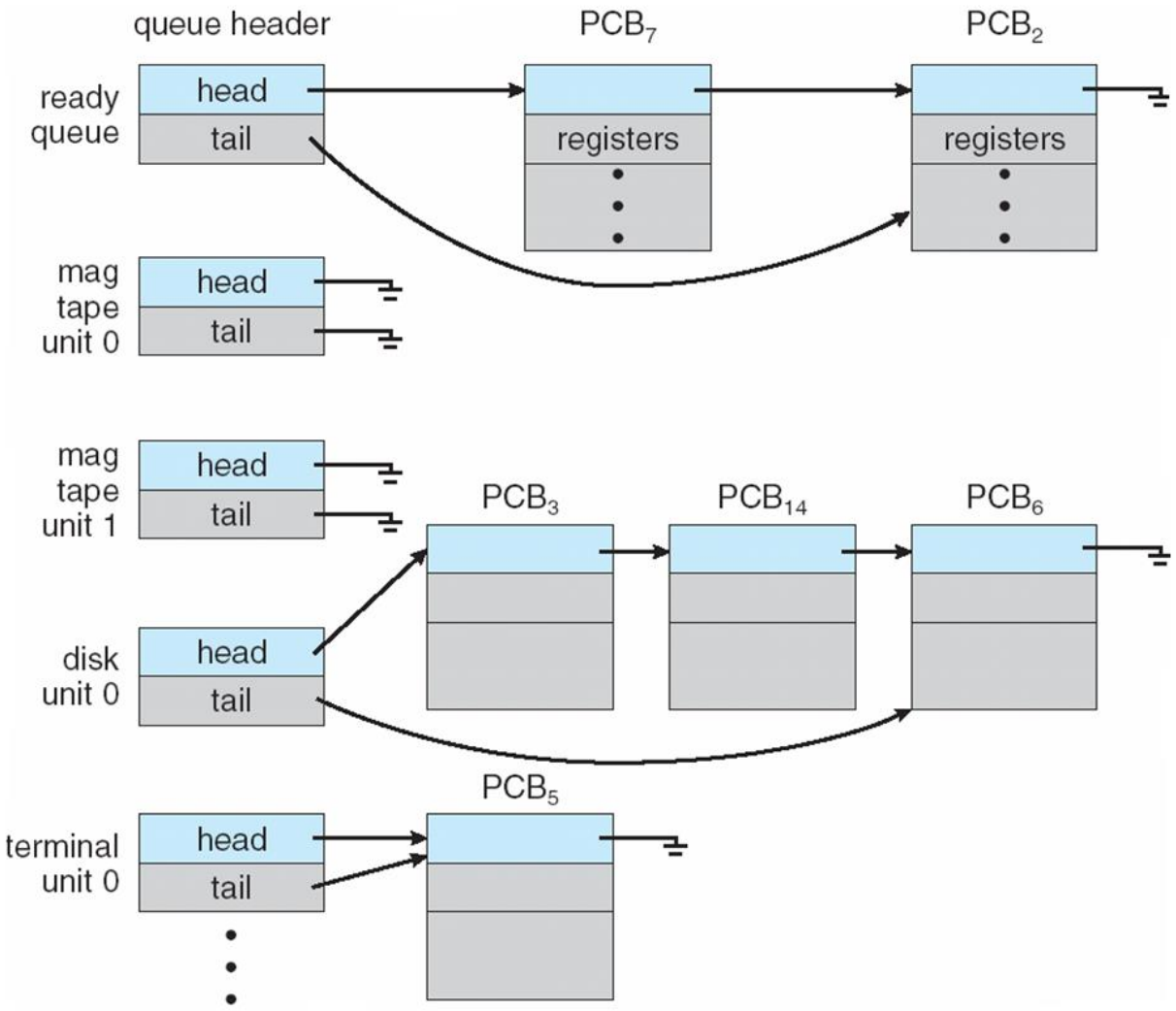
Manisha Verma
Assistant Professor
Computer Science & Engineering

Process

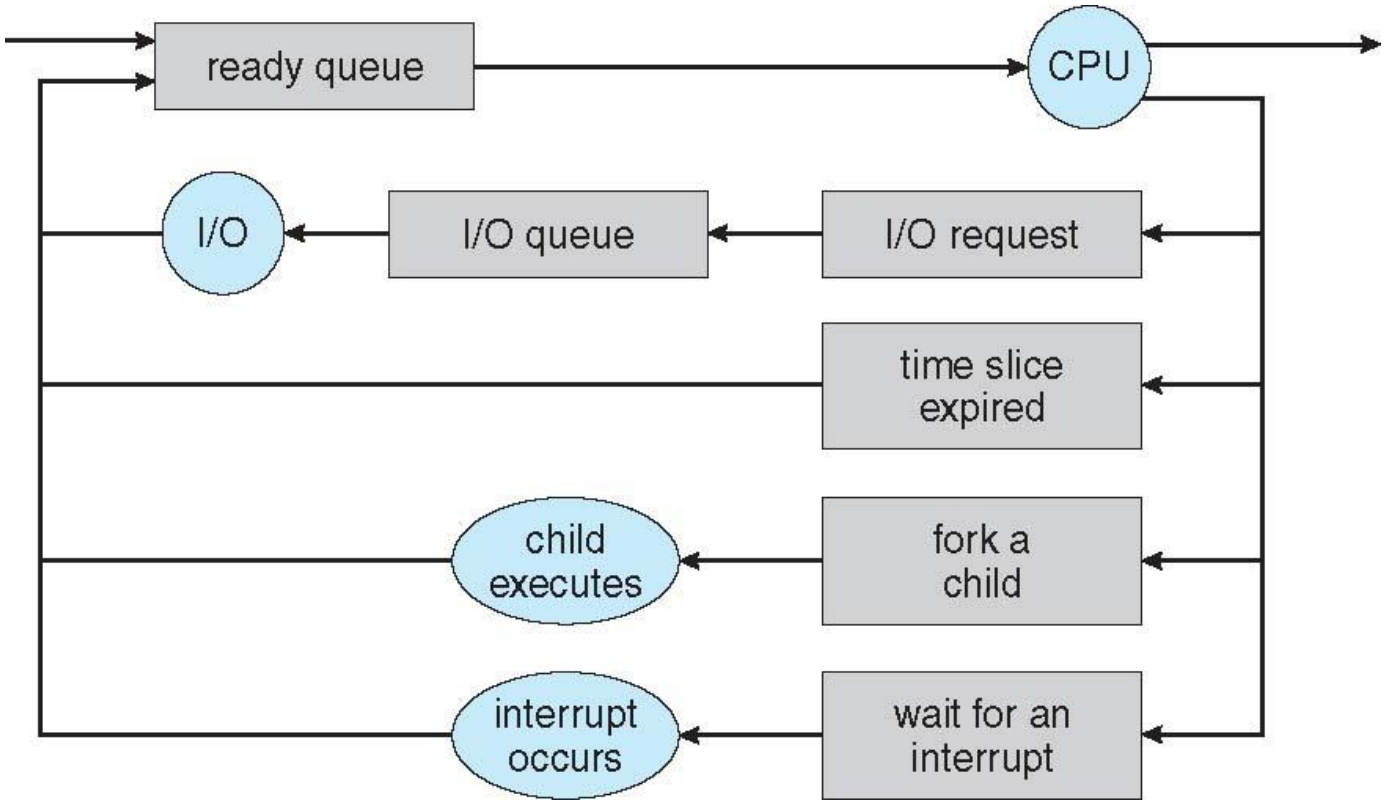
- Process Scheduling
- Representation of Process Scheduling
- Schedulers
- Addition of Medium Term Scheduling
- Process Creation
- Context Switch

Process Scheduling

- Maximize CPU use, quickly switch processes onto CPU for time sharing.
- Process scheduler selects among available processes for next execution on CPU
- Maintains scheduling queues of processes:-
  - 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


queue header
PCB7
PCB2
ready queue
head
tail
registers
registers
mag tape unit 0
head
tail
mag tape unit 1
head
tail
PCB3
PCB14
PCB6
disk unit 0
head
tail
PCB5
terminal unit 0
head
tail

Representation of Process Scheduling



Schedulers

Short-term scheduler (or CPU scheduler) – selects which process should be executed next and allocates CPU
- Sometimes the only scheduler in a system
- Short-term scheduler is invoked frequently (milliseconds) ⇒ (must be fast)

Long-term scheduler (or job scheduler) – selects which processes should be brought into the ready queue
- Long-term scheduler is invoked 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

Long-term scheduler strives for good *process mix*

Long-term scheduler (or job scheduler) – selects which processes should be brought into the ready queue

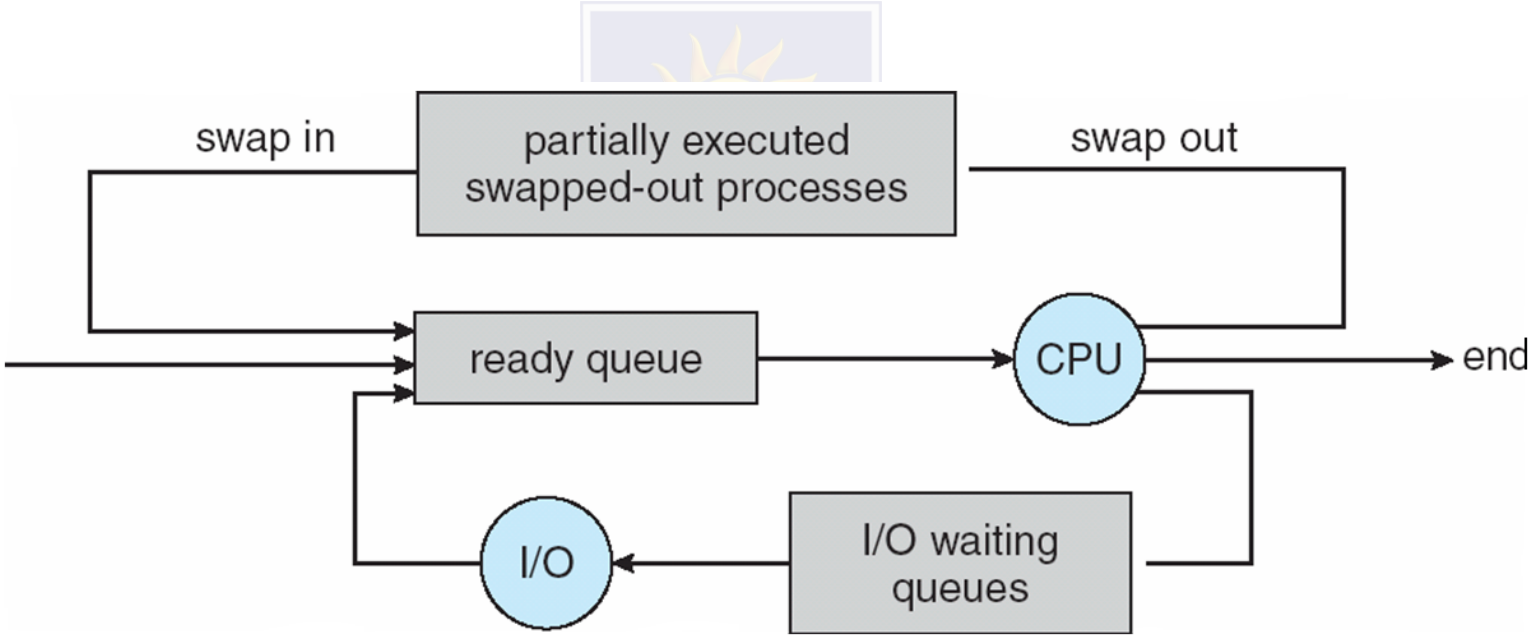
Short-term scheduler (or CPU scheduler) – selects which process should be executed next and allocates CPU

- 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

Addition of Medium Term Scheduling

Medium-term scheduler can be added if degree of multiple programming needs to decrease

Remove process from memory, store on disk, bring back in from disk to continue execution: swapping



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

Process Creation

•Parent process creates 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.
- •Parent and child share no resources.

•Execution

- •Parent and children execute concurrently.
- •Parent waits until children terminate.

•Address space

- •Child duplicate of parent.
- •Child has a program loaded into it.

•UNIX examples

- •fork system call creates new process
- •execute system call used after a fork to replace the process' memory space with a new program

MCQ

Resource sharing means……
A. Parent and children share all resources.
B. Children share subset of parent's resources
C. All sharable
D. None

Short-term scheduler (or CPU scheduler) –selects which process should be executed……
A. next and allocates CPU
B. previous and allocates CPU
C. Last process and allocates CPU
D. None

Context switch….

A. When CPU switches to another process
B. the system must save the state of the old process
C. load the saved state for the new process
D. All of these

UNIX uses…...

A. fork system call creates new process
B. Terminate process
C. memory space with a new program
D. Loading with process