KEMERBURGAZ UNIVERSITY
INSTITUTE OF SCIENCE/ELECTRIC AND COMPUTER ENGINEERING DPARTMENT
PHD PROGRAM

PROJECT NAME:

Real Time CPU Scheduling

REQUIREMENT OF ECE 519 Advanced Operating Systems COURSE

PREPEARED BY: AHMED M. ABDALLA MOHAMED

STUDENT NO: 163106002

REAL TIME CPU SCHEDULING

OUTLINES

-> Basic Concepts.

- Schedule.
- Real Time System.
- CPU burst VS I/O burst.
- CPU Scheduler.
- Dispatcher.

-> Scheduling Criteria.

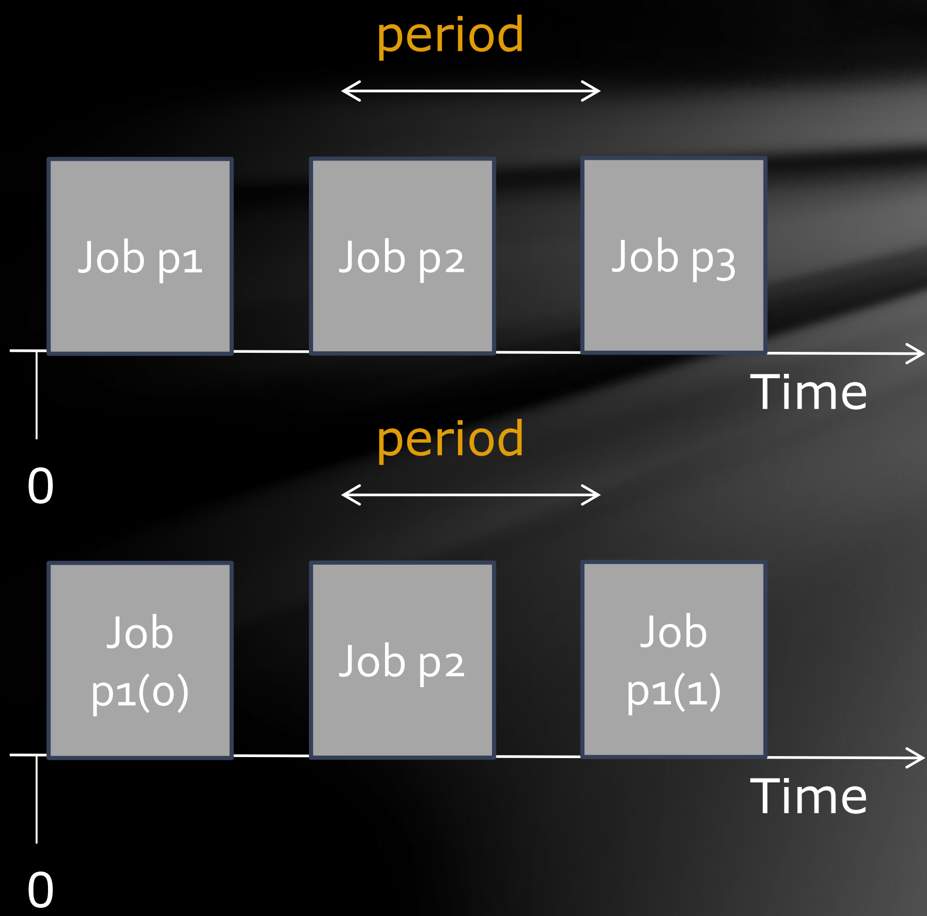
-> Scheduling Algorithms.

- First-Come, First-Served (FCFS) Scheduling.
- Shortest-Job-First (SJF) Scheduling.
- Priority Scheduling.
- Round Robin (RR).

BASIC CONCEPTS

What is a Schedule?

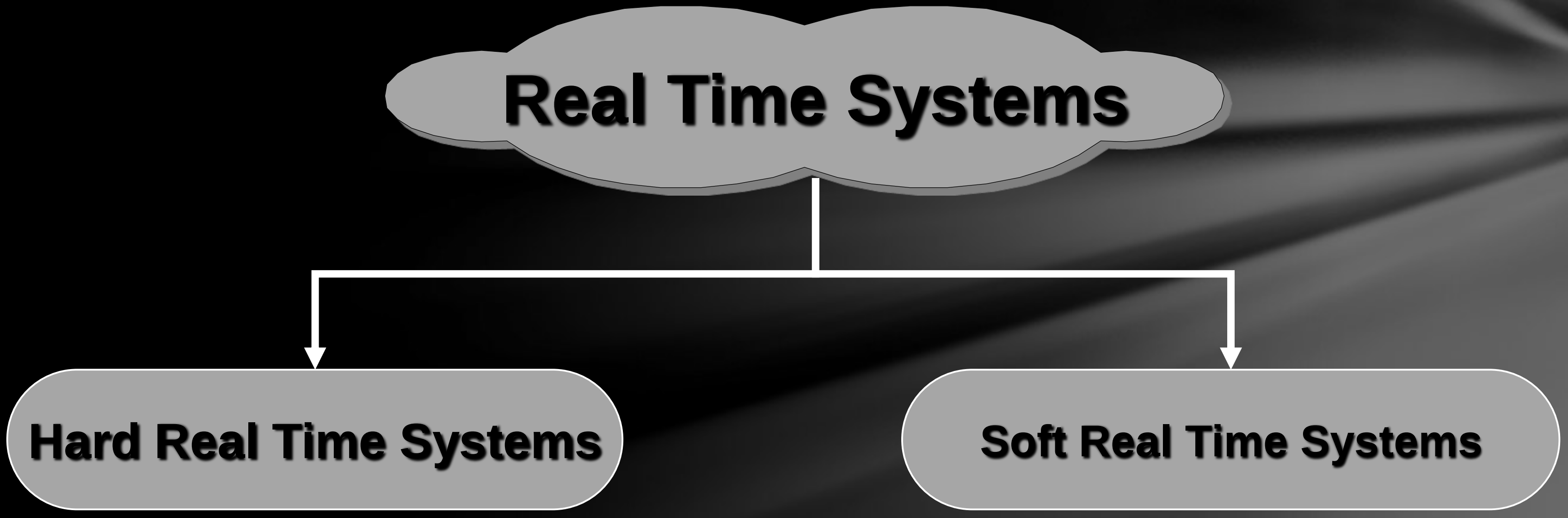
Times at which process execute.



BASIC CONCEPTS (CONT..)

Real Time System:

Systems that performs critical tasks.



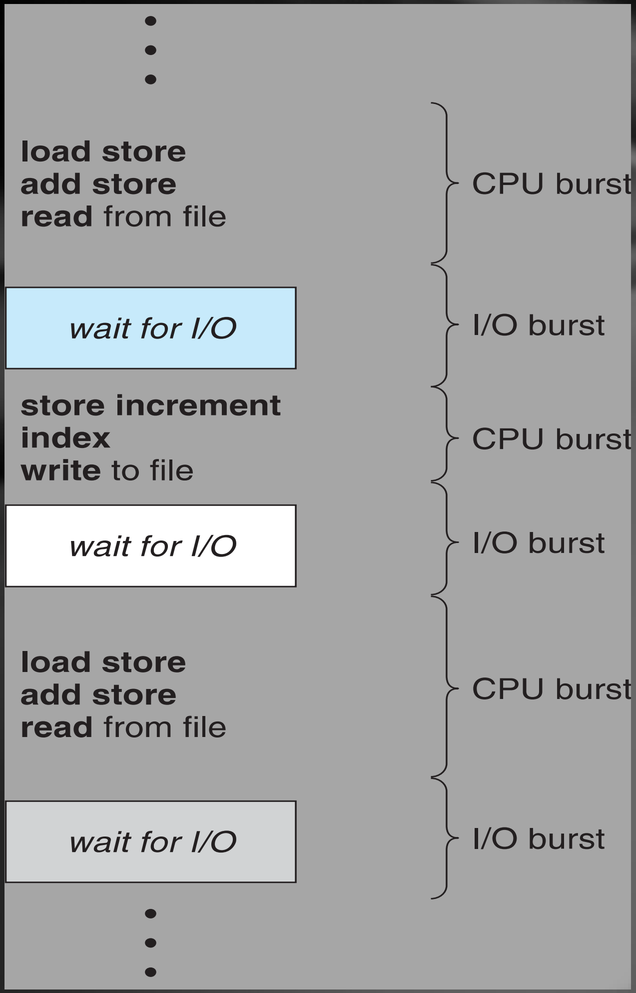
- Hard real time systems: critical tasks must be performed on time.
- Soft real time systems: critical tasks gets priority over other tasks.

BASIC CONCEPTS (CONT..)

CPU burst VS I/O burst

A CPU bursts when it is executing instructions

An I/O system bursts when it services requests to fetch information.



BASIC CONCEPTS (CONT..)

CPU Scheduler

- **Short-term scheduler** selects from among the processes in ready queue, and allocates the CPU to one of them
  - Queue may be ordered in various ways

- CPU scheduling **decisions** may take place when a process:
  1. Switches from running to waiting state
  2. Switches from running to ready state
  3. Switches from waiting to ready
  4. Terminates

- Scheduling under 1 and 4 is **nonpreemptive**

- All other scheduling is **preemptive**

BASIC CONCEPTS (CONT..)

Dispatcher

- Dispatcher module gives control of the CPU to the process selected by the short-term scheduler, this involves:

-> switching context

- > switching to user mode

- > jumping to the proper location in the user program to resume that program

- **Dispatch latency** – time it takes for the dispatcher to stop one process and start another running

SCHEDULING CRITERIA

- What do we want to achieve from scheduling (One of the following):

-> Max **CPU utilization** – keep the CPU as busy as possible

-> Max **Throughput** – of processes that complete their execution per time unit

-> Min **Turnaround time** – amount of time to execute a particular process

-> Min **Waiting time** – amount of time a process has been waiting in the ready queue

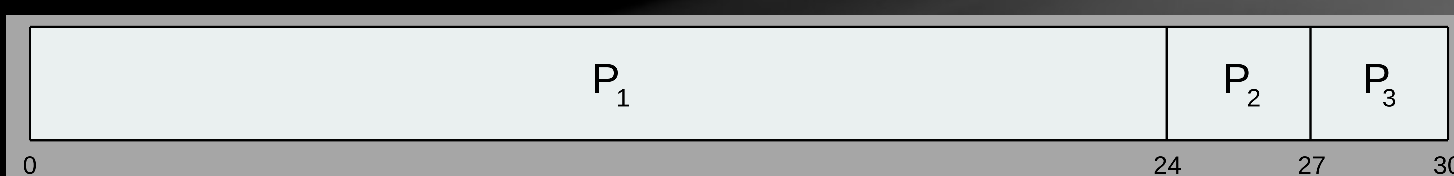
SCHEDULING ALGORITHMS

First-Come, First-Served (FCFS) Scheduling:

Example 1:- Suppose that the processes arrive in the order: P_1 , P_2 , P_3

Process	Burst Time
P_1	24
P_2	3
P_3	3

- The Gantt Chart for the schedule is:



- Waiting time for $P_1 = 0$; $P_2 = 24$; $P_3 = 27$
- Average waiting time: $(0 + 24 + 27)/3 = 17$

SCHEDULING ALGORITHMS (CONT..)

(FCFS) Example 2:-

Suppose that the processes arrive in the order: P_2 , P_3 , P_1

- The Gantt chart for the schedule is:

- Waiting time for P_1 = 6; P_2 = 0; P_3 = 3
- Average waiting time: (6 + 0 + 3)/3 = 3
- Much better than previous case why ?

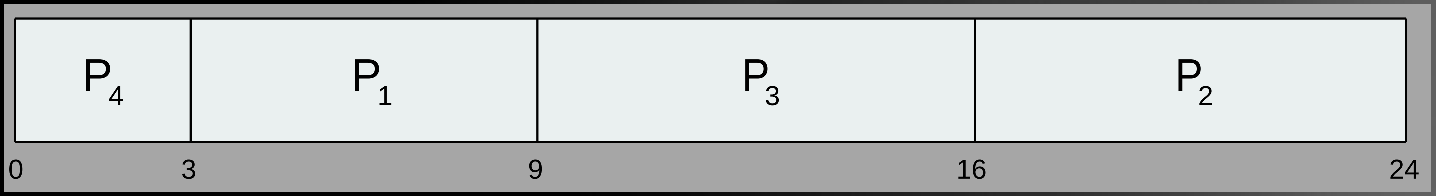
SCHEDULING ALGORITHMS (CONT..)

Shortest-Job-First (SJF) Scheduling:

EXAMPLE:-

Process	Arrival Time	Burst Time
P_1	0.0	6
P_2	2.0	8
P_3	4.0	7
P_4	5.0	3

- SJF scheduling chart



- Average waiting time = (3 + 16 + 9 + 0) / 4 = 7

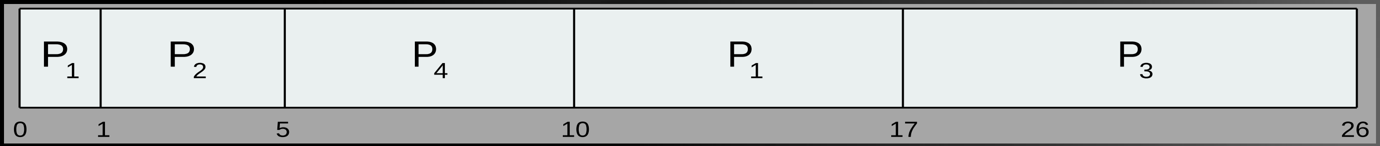
SCHEDULING ALGORITHMS (CONT..)

Example of Shortest-remaining-time-first:

- Now we add the concepts of **varying arrival times** and **preemption** to the analysis.

Process	Arrival Time	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

- *Preemptive* SJF Gantt Chart



- Average waiting time = [(10-1)+(1-1)+(17-2)+(5-3))]/4 = 26/4 = 6.5 msec

SCHEDULING ALGORITHMS (CONT..)

Priority Scheduling:

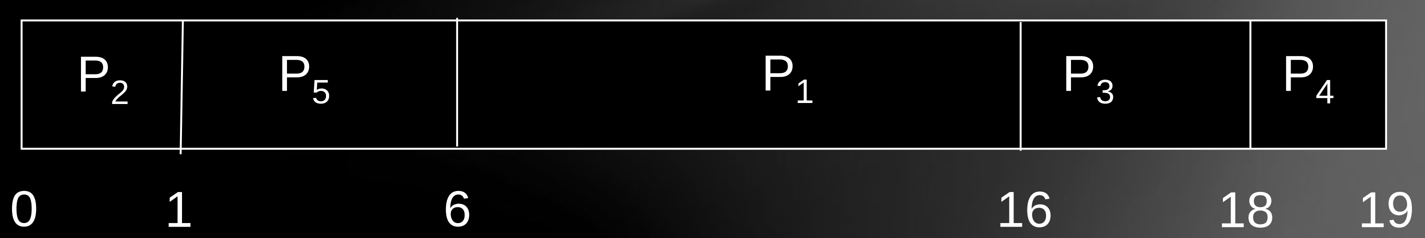
- A priority number (integer) is associated with each process
- The CPU is allocated to the process with the **highest priority** (smallest integer ≡ highest priority)
  - Preemptive
  - Nonpreemptive
- Problem ≡ **Starvation** – low priority processes may never execute
- Solution ≡ **Aging** – as time progresses increase the priority of the process

SCHEDULING ALGORITHMS (CONT..)

Example of Priority Scheduling:

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

- Priority scheduling Gantt Chart



- Average waiting time = 8.2 msec

SCHEDULING ALGORITHMS (CONT..)

Round Robin (RR):

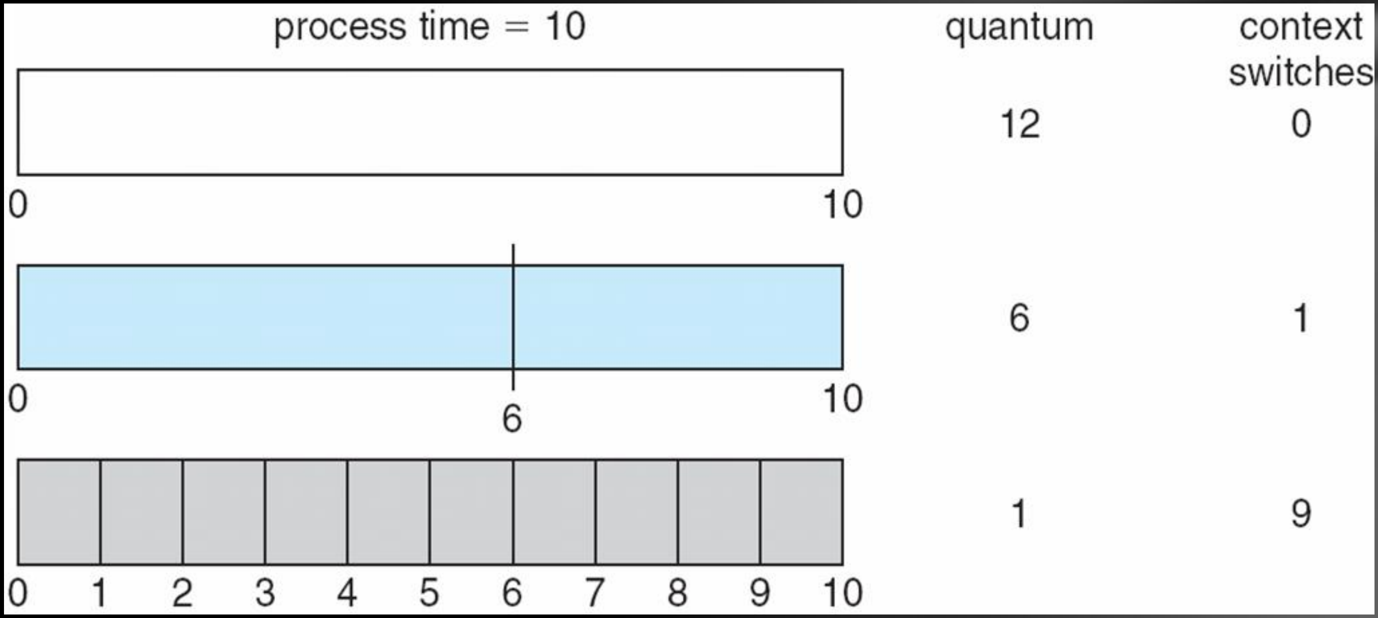
- Each process gets a small unit of CPU time (**time quantum** q)
  - After this time has elapsed, the process is preempted and added to the end of the ready queue.
- If there are n processes in the ready queue and the time quantum is q,
  - each process gets $1/n$ of the CPU
  - No process waits more than $(n-1)q$ time units.
- Timer interrupts every quantum to schedule next process

  Performance

  - *If* q is large $\Rightarrow$ FIFO
  - q small $\Rightarrow$ q must be large with respect to context switch time, otherwise overhead is too high
- Context switch < 10 micro sec

SCHEDULING ALGORITHMS (CONT..)

Time Quantum and Context Switch Time:

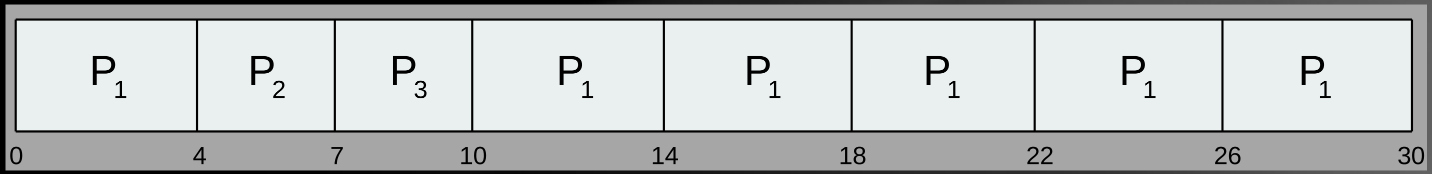


SCHEDULING ALGORITHMS (CONT..)

Example of RR with Time Quantum = 4

Process	Burst Time
P_1	24
P_2	3
P_3	3

- The Gantt chart is:



- Better ***response*** than SJF

THANK YOU FOR YOUR LISTENING