PAGE NO :
DATE : 6 / Feb / 19

UNIT – 2

CPU Scheduling

* **CPU Scheduling :-**
Almost all computer resources are require to be schedule before their uses. CPU is one of the primary resource. So, its scheduling is necessary before its uses. The scheduling done by CPU is the basis of multiprogram. Operating system scheduling refered to set of policies and mechanism built into the operating system.

Schedular – A schedular is an operating system module that select the next job to be admitted into the system and the next process to run.

* **Objectives of scheduling –**
1- Fairness
2- Throughput
3- Resources
4- Priorities

PAGE NO :
DATE : / /

* Scheduling Criteria :-
Many criteria have been suggested for comparision of CPU scheduling algorithm. The criteria includes the following -

1- CPU utilization - We have to keep the CPU as busy as possible. It may ranged 0 to 100%. In real time system it should range from 40 to 90% for lightly loaded and heavily loaded system.

2- Throughput - It is the measure of work in terms of number of process completed per unit time for long process. This range may be 1 process per hour. For short transaction, throughput may be 10 process per second.

- Turn around time - It is the sum of time periods spend in waiting to get into memory, waiting in ready queue, executing on the CPU and doing I/O time.

- Waiting time - The CPU scheduling algorithm does not affect the amount of time during, which a process execution waiting time. It is the sum of time periods

spend waiting in ready queue.

- **Responsed time** – It is the amount of time, it takes to start responding not the time taken to output that response.

* Types of Scheduling :-

1- Preemptive scheduling schemes.
2- Non-preemptive scheduling schemes.

CPU scheduling decisions may take place under the following four circumstances –

(1) When a process switch from the running state to the ~~ready~~ waiting state (For example I/O wait).
(2) When a process switch from the running state to the ready state (For example when an interrupt occur).
(3) When a process switch from the waiting state to the ready state (For example completion of I/O).
(4) When a process terminates.

In circumstances (1) and (4) there is no choice in terms of scheduling. When scheduling takes place under (1) and (4)

circumstances, we say the scheduling scheme is non-preemptive.
However, in circumstances (2) and (3) there is a choice in terms of scheduling, process may or may not be selected for its execution. This type of scheduling is called preemptive.

* **Scheduling Algorithm :-**
CPU scheduling deals with the problem of deciding which of the process in the ready queue is to be allocated to the CPU. Now, we describe several of the many CPU scheduling algorithm exist.

(1)- **First Come First Serve (FCFS) -**
FCFS is the simplest of all scheduling algorithm. The key concept of this algorithm is allocated the CPU in the order in which the process arrive.

Ques - Consider the following set of process having their burst time mention in milliseconds. CPU burst time indicates that how much time the process need the CPU.

Process	Burst time (ms)
P_1	20
P_2	7
P_3	5

PAGE NO :
DATE : / /

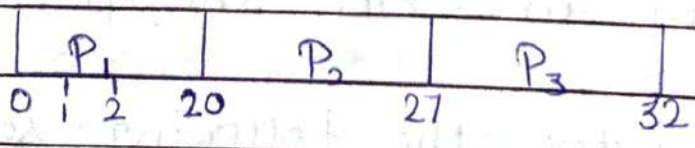
Calculate the average waiting time if the process arrive in the order of –

1- P_1, P_2, P_3 – (0, 1, 2)
2- P_2, P_3, P_1 – 0
3- P_3, P_1, P_2 – 2
4- P_3, P_2, P_1 – 0

Solution:- (1) If process arrive in order P_1, P_2, P_3. Let us assume at 0 ms, 1 ms and 2 ms respectively.

Process	Burst time	Arrival time
P_1	20	0
P_2	7	1
P_3	5	2

Since, FCFS is a non-preemptive algorithm so when a process execution is started, it is not preempt~~ive~~ until its execution is completed.



Waiting time for process P_1 = 0 ms
Waiting time for process P_2 = 20 – 1 = 19 ms
Waiting time for process P_3 = 27 – 2 = 25 ms

Average waiting time = $\frac{0+19+25}{3}$ = 14.67 ms.

PAGE NO:
DATE: / /

Consider a case if the process P_1, P_2, P_3 arrive at the same time say at 0 ms in the system in the order of P_1, P_2, P_3.

Waiting time for process P_1 = 0 ms
Waiting time for process P_2 = 20 ms
Waiting time for process P_3 = 27 ms

Average waiting time = $\frac{0+20+27}{3} = \frac{47}{3}$

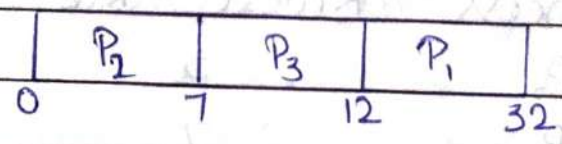
= 15.6 ms

Sol 2:- If process arrive in order P_2, P_3, P_1. Let us assume at 0 ms all process.

Process	Burst time	Arrival time
P_2	7	0
P_3	5	0
P_1	20	0

Since, FCFS is a non-preemptive algorithm so when a process execution is started, it is not preemptive until its execution is completed.

The grant chart will be-



Waiting time for process P_2 = 0 ms
Waiting time for process P_3 = 7 ms
Waiting time for process P_1 = 12 ms

PAGE NO :
DATE : / /

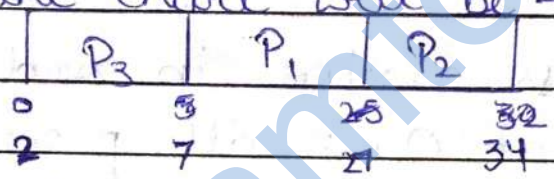
Average waiting time = $\frac{0+7+12}{3} = \frac{19}{3}$

= 6.34 ms.

(3) If process arrive in order P_3, P_1, P_2 and arrival time is 2 ms for all process.

Process	Burst time	Arrival time
P_3	5	2
P_1	20	2
P_2	7	2

The grant chart will be —



Waiting time for process P_3 = 2 ms

Waiting time for process P_1 = 5−2 = 3 ms

Waiting time for process P_2 = 25−2 = 23 ms

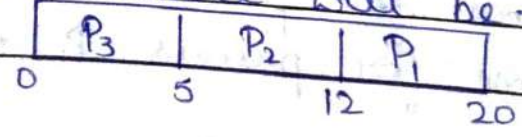
Average waiting time = $\frac{2+3+23}{3} = \frac{28}{3}$

= 9.34 ms.

(4) If process arrive in order P_3, P_2, P_1 and arrival time is 0 ms for all process.

Process	Burst time	Arrival time
P_3	5	0
P_2	7	0
P_1	20	0

The grant chart will be —



Waiting time for process P_3 = 0 ms
Waiting time for process P_2 = 5 ms
Waiting time for process P_1 = 12 ms
Average waiting time = $\frac{0+5+12}{3} = \frac{17}{3} = 5.67$ ms

(2)- **SJF (Shortest Job First) Algorithm -**

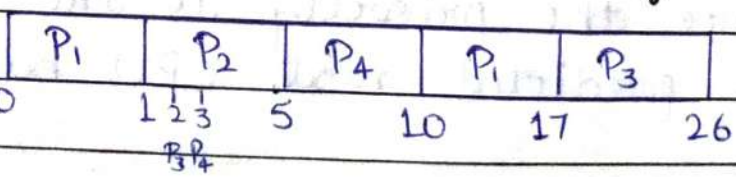
Shortest job first scheduling is a different approach to CPU scheduling. This algorith associate with each process the length of the next CPU burst. When the CPU is available it is assigned to the process that has the smallest next CPU burst. If the two process have same length of CPU burst then FCFS scheduling algorithm is follow.

Example:-

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

Calculate average waiting time in
1- Preemptive SJF scheduling
2- Non-preemptive SJF scheduling

Sol 1-

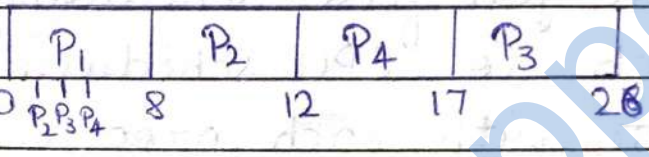


PAGE NO :
DATE : / /

Waiting time for process $P_1 = 0 + (10-1) = 9$ ms
Waiting time for process $P_2 = 1-1 = 0$ ms
Waiting time for process $P_3 = 17-2 = 15$ ms
Waiting time for process $P_4 = 5-3 = 2$ ms
Average waiting time $= \frac{9+2+15}{4} = \frac{26}{4}$

$= 6.5$ ms.

Solution 2-



Waiting time for process $P_1 = 0$ ms
Waiting time for process $P_2 = 8-1 = 7$ ms
Waiting time for process $P_3 = 17-2 = 15$ ms
Waiting time for process $P_4 = 12-3 = 9$ ms
Average waiting time $= \frac{7+15+9}{4} = \frac{31}{4}$

$= 7.75$ ms.

(3)- **Priority Scheduling** -

A priority is associated with each process and the CPU is allocated to the process with the highest priority. Equal priority process are scheduled in FCFS order.

A SJF is simply a priority algorithm where the priority is the inverse of the predicted next CPU burst.

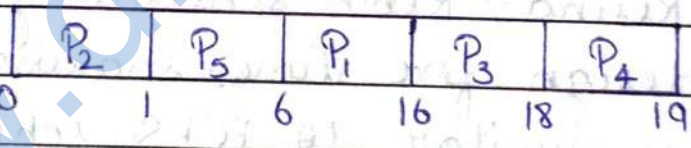
PAGE NO:
DATE: / /

Ex- Consider the following set of process assume to have arrive at time 0 in the order P_1, P_2, P_3, P_4, P_5 with the length of the CPU burst time given in milliseconds.

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

using Priority scheduling, calculate average waiting time.

Solution- Using priority scheduling we would schedule the given process according to the following grant chart -



Waiting time for process P_2 = 0 ms
" " " " P_5 = 1 ms
" " " " P_1 = 6 ms
" " " " P_3 = 16 ms
" " " " P_4 = 18 ms

Average waiting time = $\frac{1+6+16+18}{5} = \frac{41}{5}$

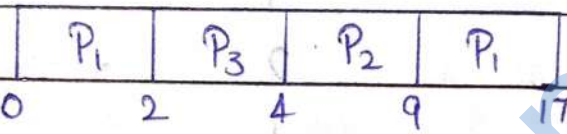
= 8.2 ms.

PAGE NO :
DATE : / /

Ques- Preemptive Priority Scheduling –

Process	Burst time	Priority	Arrival time
P_1	10	3	0
P_2	5	2	1
P_3	2	1	2

olution:- The grant chart will be –



Waiting time for process $P_1 = 0 + (9-2) = 7$ ms

" " " " $P_3 = 2 - 0 = 0$ ms

" " " " $P_2 = 4 - 2 = 2$ ms

Average waiting time = $\frac{0+7+2}{3} = \frac{9}{3} = 3$ ms.

(4)- **Round-Robin scheduling Algorithm (RR):-**
The Round-Robin scheduling algorithm is design for time-sharing system. It is similar to FCFS scheduling algorithm but preemption is added to switch between process.

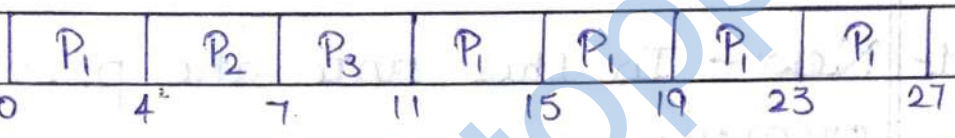
A small unit of time called a time quantum (or time slice) is define where range is kept between 10 to 100 ms.

Example- Consider the following set of process that arrive at time 0 ms

Process	Burst time
P_1	20
P_2	3
P_3	4

If we use time quantum of 4 ms then calculate the average waiting time using RR scheduling.

Sol:- The grant chart will be—



Waiting time for process $P_1 = 0 + (11-4) = 7$

" " " " $P_2 = 4$

" " " " $P_3 = 7$

Average waiting time $= \frac{7+4+7}{3} = \frac{18}{3} = 6$ ms.

★ **Process :-**

A process is an instance of a program in execution. A process in the smallest unit of work individually schedulable by an operating system. Such a system consist of a collection of process. Operating system process execute system code and user process execute user code. All these process execute concurrently.

Process is not same as program. A program is a passive entity while a process is an

active entity.

Imp:
* **Process States** – When a process comes in execution it change its state. The state of a process is defined in part by the current activity of that process. Each process may be one of the following state during the time of its execution.

1- **New** – In this state the process is being created.

2- **Ready** – In this state the process is waiting to be assign to a processor through a short term scheduler.

3- **Waiting** – In this state the process is waiting or blocked for some event to occur such as input – output completion or reception office signal.

4- **Running** – In this state instructions are being executed.

5- **Terminated** – In this state the process has finished the execution.

The state transition diagram corresponding to these states is represented in figure

below —

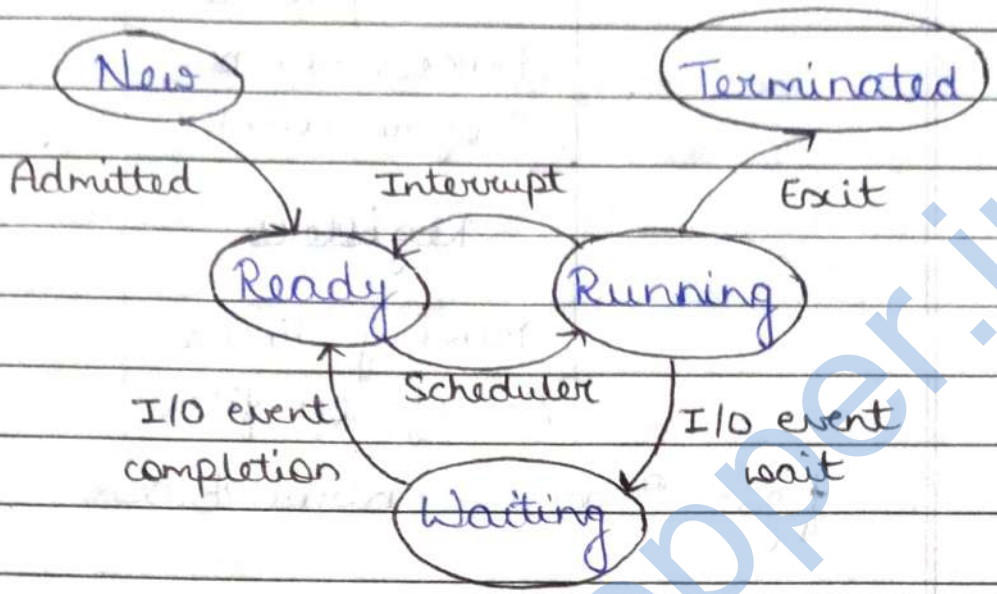


Fig:- Process states Transition Diagram

★ Process Control:-

Process control block — In operating system, each process is represented by a process control block or a task control block.

It is a data-structure that physically represent a process in the memory of a computer system. It contains many pieces of information associated with a specific process that include the following —

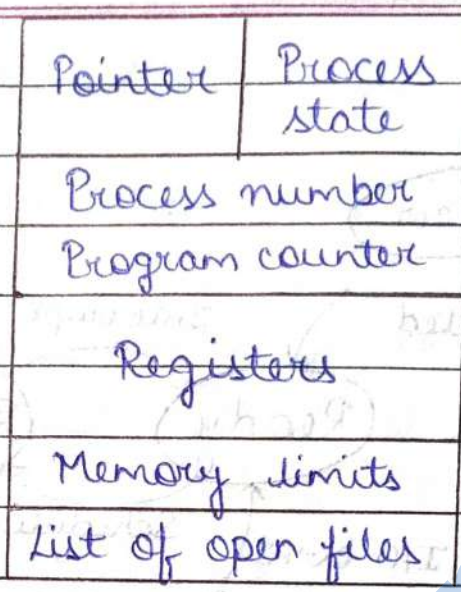


fig:- Process Control Block

1- Process state — The state may be new, ready, waiting, running or terminated.

2- Program counter — The counter indicates the address of the next instructions to be executed for this process.

3- CPU registers — The registers vary in numbers, size and type depending upon the computer architecture. They include accumulators, index registers, flags, stack pointers and other general purpose registers and any condition called information.

4- CPU scheduling information — This information include a process priority, pointers to scheduling queue and any

other scheduling parameters.

5- Memory management information — This information include the value of base and limit registers, the page of segmentable use by the operating system.

6- Input - output information — This information include the list of input - output device allocated to this process, a list of open files and so on.

* **Critical section :-**

The critical section in a code segment where the shared variable can be access. Atomic action is required in a critical section. Only one process can execute in its critical section at a time, all other process have to wait to execute in their critical sections.

The critical section is given as follow -

```
do
{
Entry section
Critical section
Exit section
Remainder section
```

PAGE NO :
DATE : / /

} while (TRUE);

In the above diagram the entry section enters the entry into the CS. It aquire the resources needed for execution by the process. The exit section handles the exit from the critical section. It release the resources and also inform the other process that CS is free.

The CS problem need a solution to synchronize the different process. The solution to the CS problem must satisfy the following conditions :-

1- **Mutual exclusion** - It implies that only one process can be inside the CS. at any time. If any other process require the CS, they must wait until it is free.

2- **Progress** - Progress means that if a process is not using the CS then it should not stop any other process from accessing it. In other words, any process can enter a CS if it is free.

3- **Bounded waiting** - Bounded waiting

means that each process must have a limited waiting time, it should not wait endlessly to access the CS.

* **Process Synchronization :-**
Process Synchronization means sharing system resources by processes in such a way that concurrent access to shared data is handled thereby minimizing the chance of inconsistent data.

Process synchronization was introduced to handle problems that arose while multiple process executions. Some of the problems are critical section problem, mutex locks.

Ex- What be happen if at the same time, same process use the same resource?

```
P()
{
R(a)
a = a + 1;
W(a)
}
```

a = 10, Process P_1, P_2

Systematic manner	Distributed manner
$P_1 = 11$	$P_1 = 10$
$P_2 = 12$	$P_2 = 10$

PAGE NO :
DATE : 7 / Mar/2019

* **Race Condition :-** Race condition is a special condition that may occur during of same data at same time.

Race condition is an undesirable that occurs when a device or system attempt to perform two or more operations at same time but because of the nature of the device or system the operations must be done in the proper sequence to be done correctly.

When order of execution can change the result now that is nothing, it's a race condition.

* **Semaphores :-** A semaphor is an integer variable int s that a part from utilization is access only through two standard atomic operations - wait and signal.

1. **Wait** - It is an atomic operation. It reduces the value by 1.

2. **Signal** - It is an atomic operation. It increases the value by 1.

PAGE NO:
DATE: / /

P_1	P_1, P_2, P_3	
do {	int s = 1;	
entry section	wait (s)	signal (s)
// critical section	{	{
exit section	while (s <= 0);	s = s + 1;
// remainder section	s = s - 1;	}
} while(T)	}	