[Premium Students Assignment]

Date
Page

Q.1. Five batch jobs A to E arrive at same time. They have estimated running times 10, 2, 6, 8, 4 minutes. Their priorities are 3, 2, 5, 4, 1 respectively with 5 being highest priority for each of the following algorithm determine mean process turnaround time - Ignore process swapping overhead. Round robin (q=3), Priority the scheduling, FCFS, SJF.

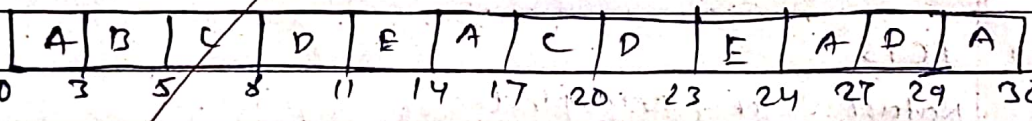
Sol^n -

Job	Arrival Time	Burst time	Priority
A	0	10	3
B	0	2	2
C	0	6	5
D	0	8	4
E	0	4	1

(i) Round Robin

: Nature = Preemptive

Gnatt chart :



Jobs	AT	BT	Exit Time	TAT = ET - AT	WA = TAT - BT
A	0	10	30	30-0 = 30	30-10 = 20
B	0	2	5	5-0 = 5	5-2 = 3
C	0	6	8	8-0 = 8	8-6 = 2
D	0	8	29	29-0 = 29	29-8 = 21
E	0	4	24	24-0 = 24	24-4 = 20

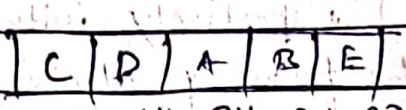
Avg TAT = $\frac{30+8+5+29+24}{5} = \frac{91}{5} = 19.2$ ms

Avg waiting time: $\frac{66}{5} = 13.2$ ms

(ii) Priority scheduling

Nature : Non-preemptive

Gnatt chart :



Jobs	BT	AT	ET	TAT	WT
A	10	0	24	24	14
B	2	0	26	26	24
C	6	0	6	6	0
D	8	0	14	14	6
E	4	0	30	30	26

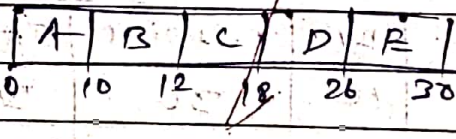
Avg TAT = $\frac{24+26+6+14+30}{5} = \frac{100}{5} = 20ms$

Avg Waiting time = $\frac{70}{5} = 14ms$

(iii) FCFS

Nature : Non-preemptive

Gnatt chart :



Jobs	AT	BT	ET	TAT	WT
A	0	10	10	10	0
B	0	2	12	12	10
C	0	6	18	18	12
D	0	8	26	26	18
E	0	4	30	30	26

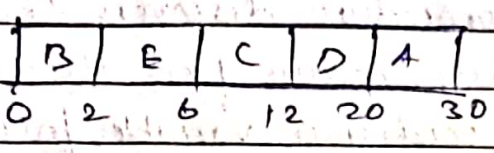
Avg TAT = $\frac{10+12+18+26+30}{5} = \frac{96}{5} = 19.2ms$

Avg Waiting time = $\frac{0+10+12+18+26}{5} = \frac{66}{5} = 13.2ms$

(iv) SJF

Nature: Non-preemptive

Gnatt chart:



Jobs	AT	BT	ET	TAT	WT
A	0	10	30	30	20
B	0	2	2	2	0
C	0	6	12	12	6
D	0	8	20	20	12
E	0	4	6	6	2

$$\text{Avg TAT} = \frac{30+20+12+6+2}{5} = \frac{70}{5} = 14\text{ms}$$

$$\text{Avg WT} = \frac{20+0+6+12+2}{5} = \frac{40}{5} = 8\text{ms}$$

Qno 2. What is moniter? Give the implementation of bounded buffer producer-consumer problem using moniter.

Ans- Monitor: Moniters were developed in the 1970s to make it easier to avoid race condition and implement mutual exclusion.

A moniter is a collection of procedures, variable and data structure grouped together in a single module or package.

=> Process can call the monitor procedures but cannot access the internal data structure.

=> Monitor have an important property that makes them useful for achieving mutual exclusion.

The solution uses condition variable along with two operations on them, wait and signal. when a monitor procedure discovers that it cannot continue (e.g. the procedure find the buffer full), it does wait on some condition variable full.

⇒
```
monitor 'Procedur consumer
    condition full, empty;
    integer count;

    procedure insert (item: integer);
    begin
        if count=N then wait(full)
        insert_item(item)
        count := count + 1;
        if count = 1 then signal(empty)
    ends

    function remove :: integer;
    begin
        if count=0 then wait(empty);
        remove = remove - item
        count := count -1;
        if count = N-1 then signal(full)
    ends

    count := 0;
end monitor;
```

```
procedure consumer;
begin
    while true do
        begin
            item = Producer(consumer_remover);
            consumer_item(item)
        end
end
```

⇒ This action causes the calling process to block. It also allows another process that had been previously prohibited from entering the monitor to enter now.

⇒ The other process, for example the consumer, can wake up its sleeping partner by doing a signal on the condition

Qno 3. Explain the concept of spooling.

ans- Spooling stands for simultaneous peripheral operations Online. It is a computer operating system process that helps in managing input-output operations more efficiently

⇒ It primarily deals with managing data between peripheral devices (such as printers, disk, drivers and tape drives) and CPU of a computer system.

(i) Buffering: Spooling uses buffers to store data temporarily. When an application sends data to peripheral device for processing (such as printing a document), the data is not send directly. Instead it's first store in a buffer.

(iii) Overlapping operations : spooling allows overlapping of I/O operations. While one operation is being performed (e.g. printing) the CPU can simultaneously work on other tasks or process. This overlapping enhances system efficiency and reduces idle time for CPU.

(iii) Device independence : spooling enable device independence meaning the CPU does not need to know the specifics of each peripheral device. It sends data to a spooler which then handle the task of sending the data to the appropriate device.

(iv) Print spooling : one of the most common application of spooling is in print management, when multiple users want to print document simultaneously, spooling queue the print jobs allowing them to be processed in the order they were recieved.

(v) Error handling : spooling system often include error handling mechanism. If a peripheral device encountered an error while processing data, the spooler can handle the error gracefully without disrupting other system processes.

Qno 4. What are the requirements that a solution to the critical section problem must satisfy.

Sol"- A solution to the critical section problem in operating system must satisfy several requirement to ensure proper synchronization and mutual exclusion among concurrent processes accessing shared resources. These requirements are -

(i) Mutual Exclusion : Only one process can execute in its critical section at any given time. This ensure that conflict-ing operations do not occurs concurrently, preventing data corruption or inconsistent result.

(ii) Progress : If no process is executing in its critical section and some process wish to enter their critical section then only those processes that are not executing in their remainder section can participate in deciding which will enter its critical section next.

(iii) Bounded waiting : There exist a bounded on the no. of times other process are allowed to enter their critical section after a process has made a request to enter its critical section and before that request is granted.

(iv) No Assumption about speed or No. of processors : The solution should not make assumptions about the relative speed of processor or the no. of processors available in the systems. It should work correctly regardless of these factors.

(v) Independence of processes : The solution should work correctly even if processes execute in arbitary order and timing, without depending on specific timing or sequence of execution.

Ques. Draw four gnatt chart illustrating the execution of these processes using FCFS, SJF, priority and round robin (quantum = 2) scheduling.

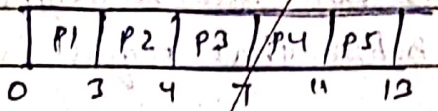
Soln -

Process	Priority	Burst Time	Arrival time
P1	10	3	0
P2	1	1	0
P3	2	3	0
P4	1	4	0
P5	5	2	0

(i) FCFS :

Nature - Non preemptive

Gnatt chart -

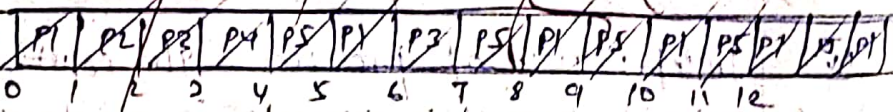


(ii) Round Robin :

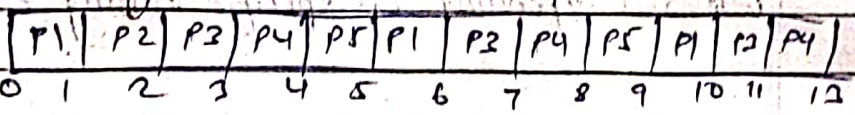
Nature - Preemptive

Gnatt chart -

Ready queue -

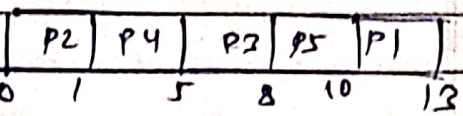


Gnatt chart -



(iii) SJF :

Nature - Non-preemptive



(iv) priority scheduling :

Nature : Non-preemptive (small no. = High Priority)