JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Lecture-17 By Vivek Dubey

COMPUTER SCIENCE AND ENGINEERING

CS403PC

COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

CS403PC

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Scheme of Operating System

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

S. No.	Course Code	Course Title	L	T	P	Credits
3	CS403PC	Operating Systems	3	0	0	3
6	CS406PC	Operating Systems Lab	0	0	3	1.5

Lecture-17 By Vivek Dubey

UNIT-2:

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- **Process and CPU Scheduling –**
  - Process concepts and scheduling, Operations on Processes, Cooperating Processes, Threads, and Interposes Communication, Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling.
- **System call interface for process management-**
  - fork, exit, wait, waitpid, exec

Lecture-17 By Vivek Dubey

Scheduling Algorithms

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Different Scheduling Algorithms

- *First Come First Serve (FCFS)*
- *Shortest Job First(SJF)*
- *Longest Job First(LJF)*
- *Shortest Remaining Time First(SRTF)*
- *Longest Remaining Time First(LRTF)*
- *Round Robin Scheduling*
- *Priority Based scheduling (Non-Preemptive)*
- *Highest Response Ratio Next (HRRN)*
- *Multilevel Queue Scheduling*
- *Multi level Feedback Queue Scheduling*

Lecture-17 By Vivek Dubey

Round Robin Scheduling

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Scheduling Algorithm

- **The name of this algorithm** *comes from the round-robin principle*, where each person gets an equal share of something in turns.
- **It is mostly used** *for multitasking.*
- In Round-robin scheduling, **each ready task** runs turn by turn only in a cyclic queue *for a limited time slice*.
- **This algorithm also offers** *starvation free execution of processes.*

Lecture-17 By Vivek Dubey

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Characteristics of Round-Robin Scheduling

- **Round robin is** a pre-emptive algorithm
- **The CPU is shifted to** the next process after fixed interval time, *which is called time quantum/time slice.*
- **The process that is preempted is added** to the end of the queue.
- **Round robin is** a hybrid model which is clock-driven.
- **Time slice should be minimum**, which is assigned for a specific task *that needs to be processed*. However, it may differ OS to OS.
- **It is a real time algorithm** which responds to the event *within a specific time limit.*
- **Round robin is** one of the oldest, fairest, and easiest algorithm.
- **Widely used scheduling method** in traditional OS.

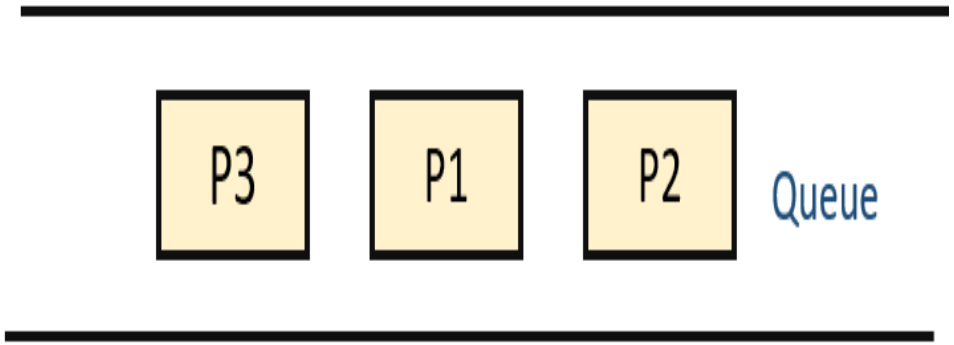
Lecture-17 By Vivek Dubey

Example of Round-robin Scheduling

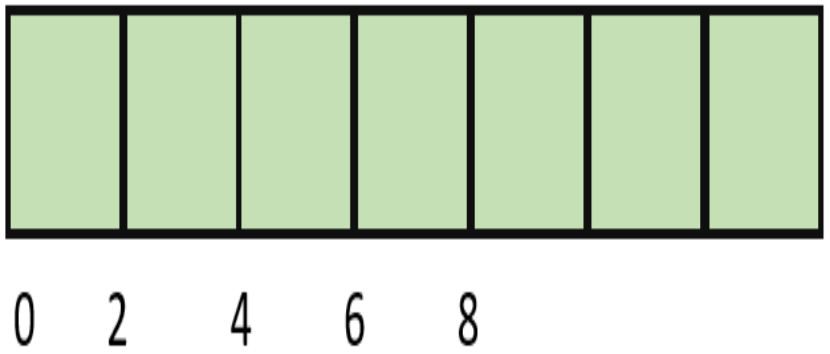
Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Consider this following three processes

Process Queue	Burst time
P1	4
P2	3
P3	5



Time Slice = 2

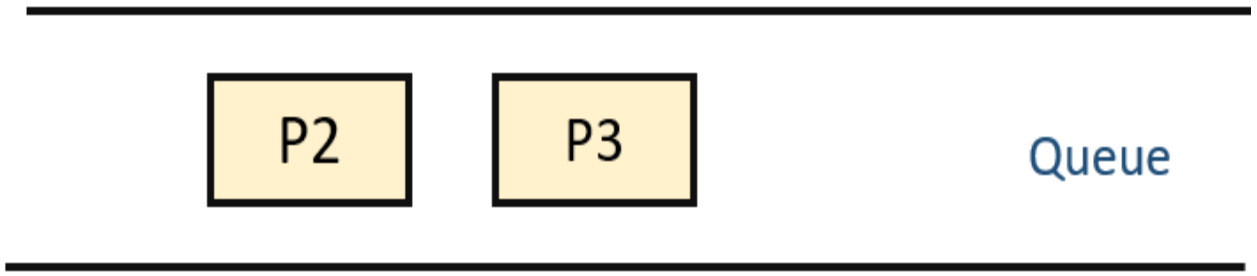


Lecture-17 By Vivek Dubey

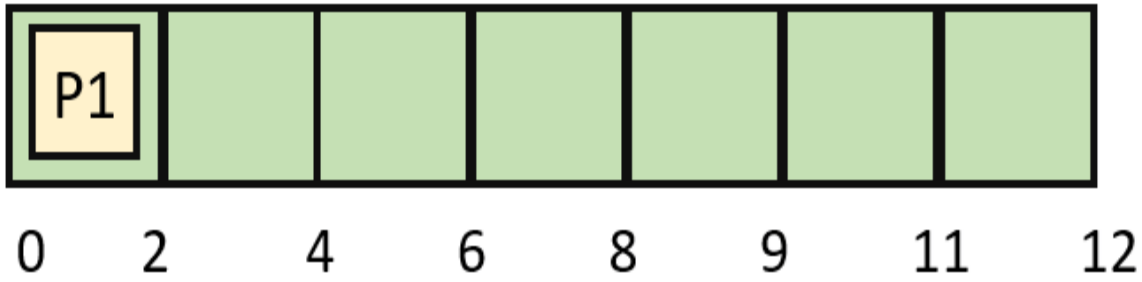
Step 1) The execution begins with process P1, which has burst time 4. Here, every process executes for 2 seconds. P2 and P3 are still in the waiting queue.

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Process Queue	Burst time
P1	4
P2	3
P3	5



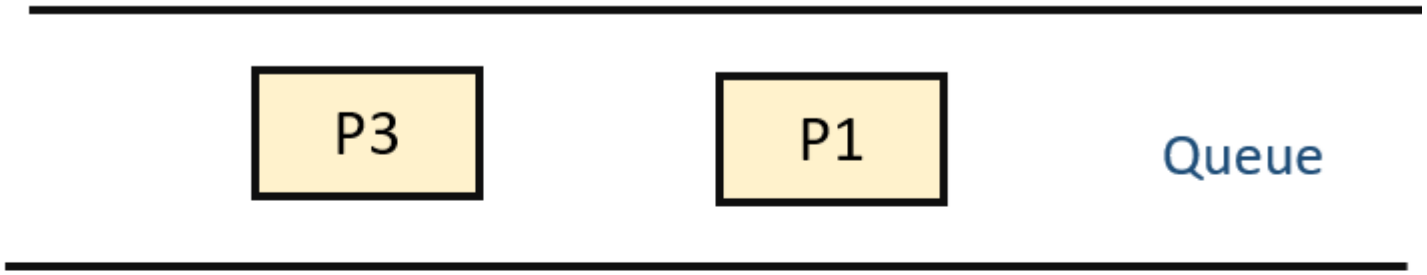
Time Slice = 2



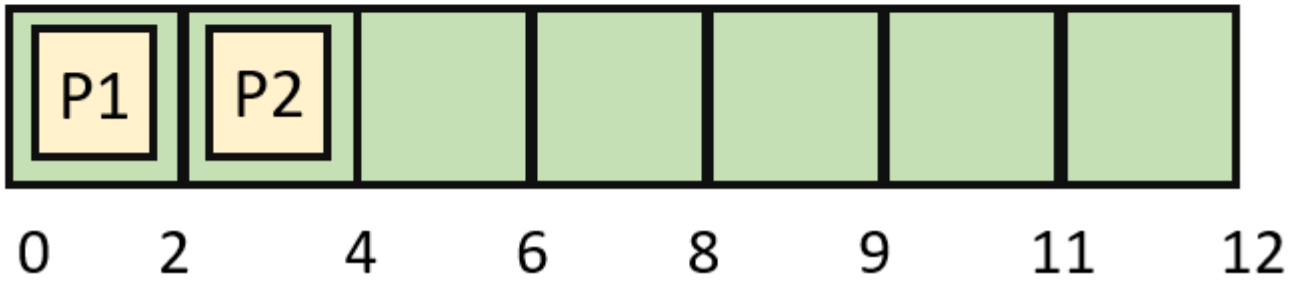
Step 2) At time =2, P1 is added to the end of the Queue and P2 starts executing.

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Process Queue	Burst time
P1	4
P2	3
P3	5



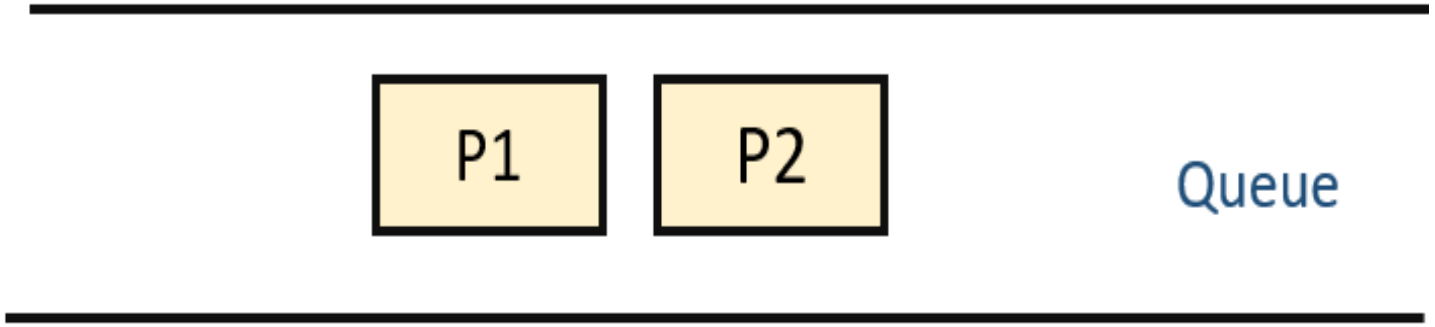
Time Slice = 2



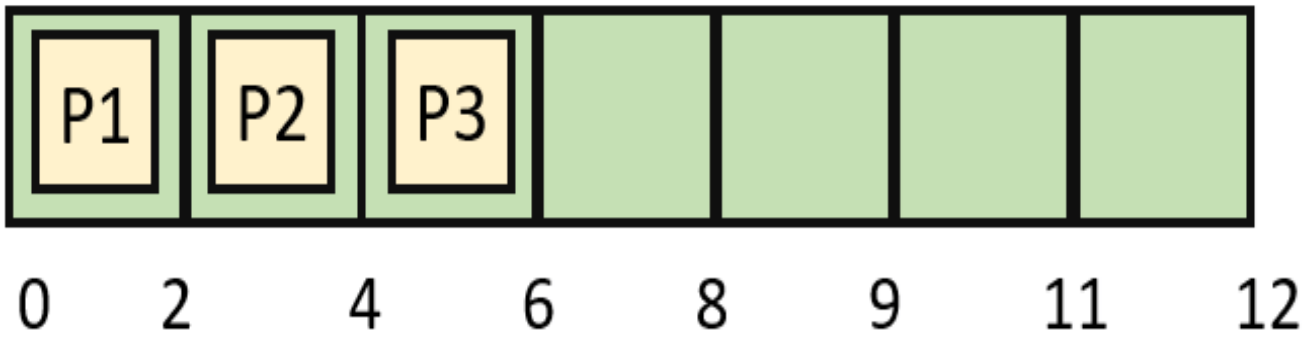
Step 3) At time=4 , P2 is preempted and add at the end of the queue. P3 starts executing.

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Process Queue	Burst time
P1	4
P2	3
P3	5



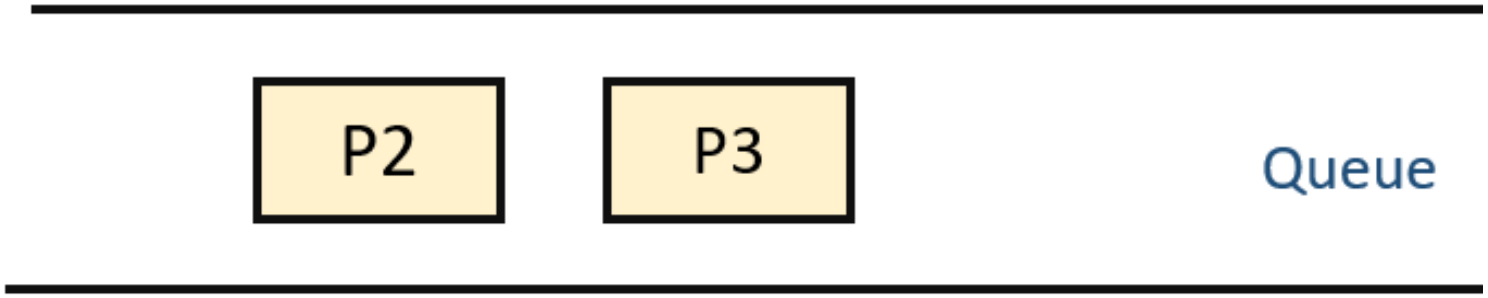
Time Slice = 2



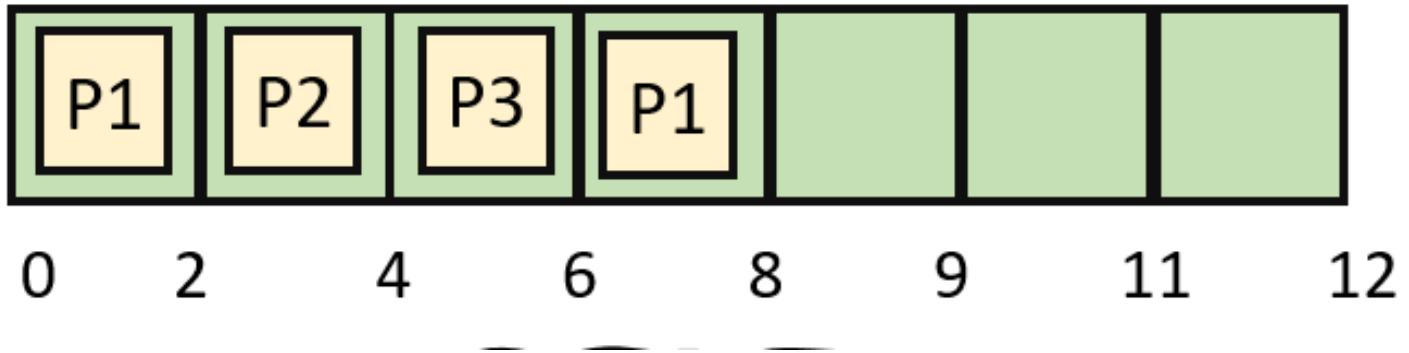
Step 4) At time=6, P3 is preempted and add at the end of the queue. P1 starts executing.

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Process Queue	Burst time
P1	4
P2	3
P3	5



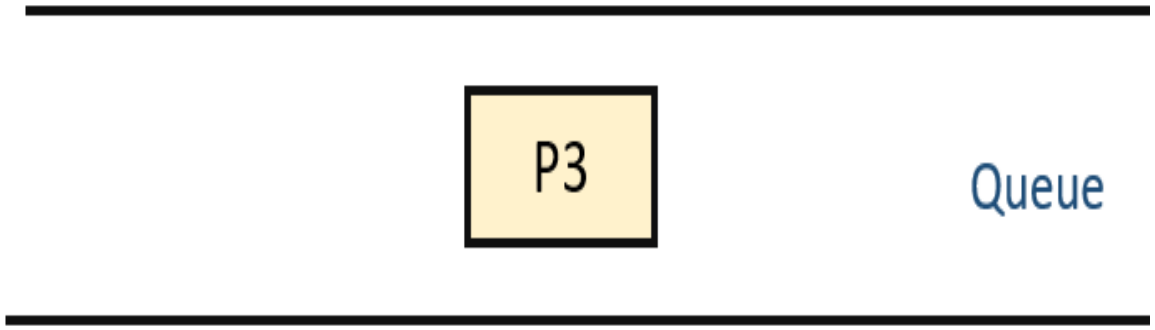
Time Slice = 2



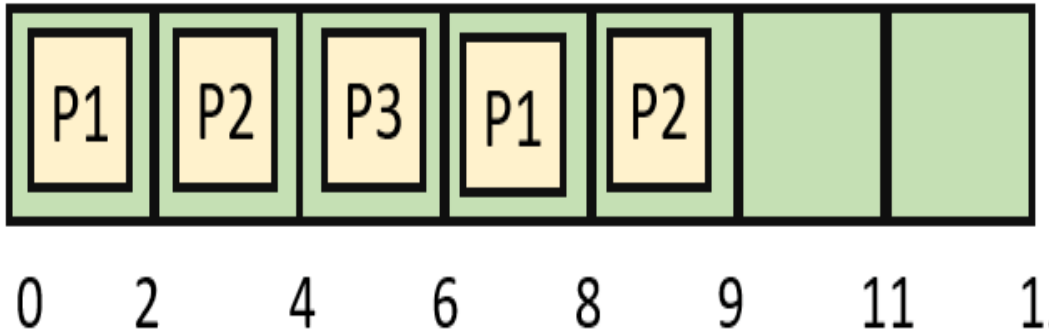
Step 5) At time=8 , P1 has a burst time of 4. It has completed execution. P3 starts execution

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Process Queue	Burst time
P1	4
P2	3
P3	5



Time Slice = 2

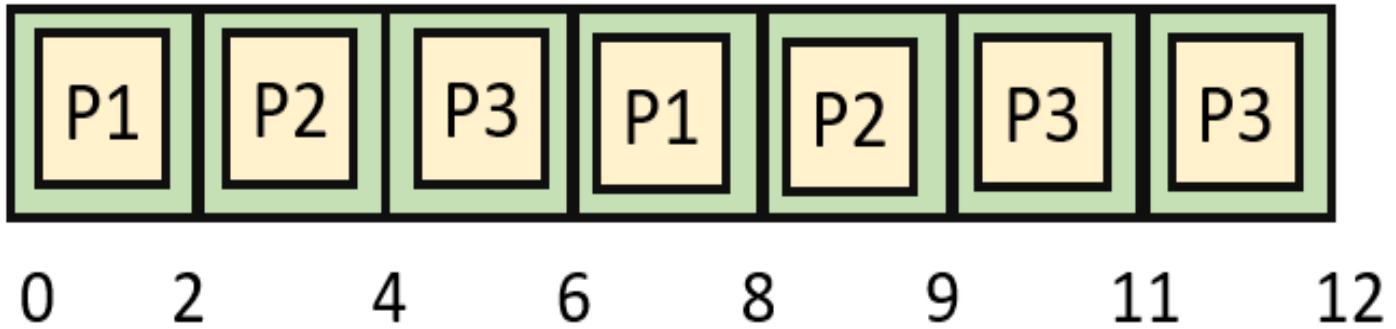


Step 6) P2 has a burst time of 3. It has already executed for 2 interval. At time=9, P2 completes execution. Then, P3 starts execution till it completes.

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Process Queue	Burst time
P1	4
P2	3
P3	5

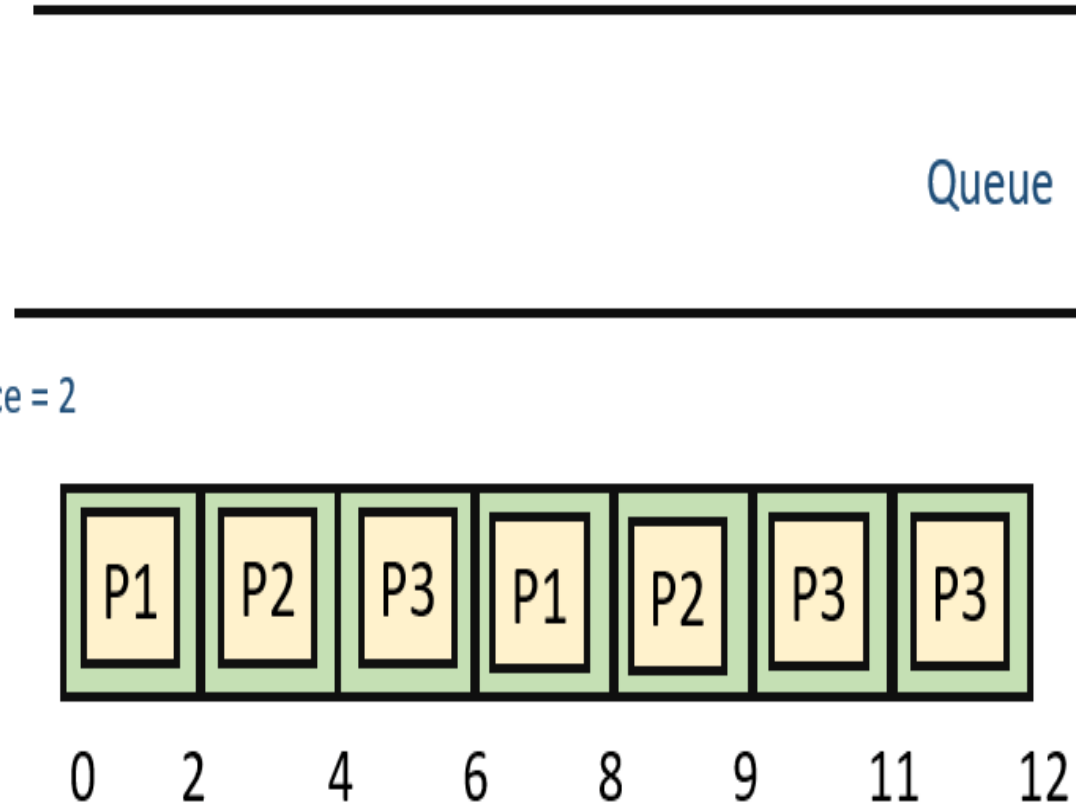
Time Slice = 2



Step 7) Let's calculate the average waiting time for above example.

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- **Wait time**
  - P1= 0+ 4= 4
  - P2= 2+4= 6
  - P3= 4+3= 7
- **Average Waiting Time**
  - =(4+6+7)/3
  - = 17/3
  - =5.6



Advantage and Disadvantages of Round-robin Scheduling

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Advantages

1. It can be actually implementable in the system *because it is not depending on the burst time.*
2. It doesn't suffer from the problem *of starvation or convoy effect*.
3. All the jobs get ***a fare allocation of CPU***.

Convoy Effect is **phenomenon associated with the First Come First Serve (FCFS) algorithm, in which the whole Operating System slows down due to few slow processes.**

Disadvantages

1. The higher the time quantum, ***the higher the response time in the system***.
2. The lower the time quantum, ***the higher the context switching overhead in the system.***
3. Deciding a perfect time quantum *is really a very difficult task in the system*.

Lecture-17 By Vivek Dubey

Coding Part-1

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

```
#include<stdio.h>
void main()
{
    int i,j,n,bu[10],wa[10],tat[10],t,ct[10],max;
    float awt=0,att=0,temp=0;
    printf("Enter the no of processes -- ");
    scanf("%d",&n);
```

Coding Part-2

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)


```
for(i=0;i<n;i++)
{
    printf("\nEnter Burst Time for process %d -- ", i+1);
    scanf("%d",&bu[i]);
    ct[i]=bu[i];
}

printf("\nEnter the size of time slice -- ");
scanf("%d",&t);
```

Coding Part-3

- max=bu[0];

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- for(i=1;i<n;i++)
  - if(max<bu[i])
    - max=bu[i];

Coding Part-4

```
for(j=0;j<(max/t)+1;j++)
    for(i=0;i<n;i++)
        if(bu[i]!=0)
        {
            if(bu[i]<=t)
            {
                tat[i]=temp+bu[i];
                temp=temp+bu[i];
                bu[i]=0;
            }
            else
            {
                bu[i]=bu[i]-t;
                temp=temp+t;
            }
```

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Lecture-17 By Vivek Dubey

Coding Part-5

```
for(i=0;i<n;i++)
{
    wa[i] =  tat[i]  -  ct[i];
    att   =  att     +  tat[i];
    awt   =  awt     +  wa[i];
}
```

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Coding Part-6

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)


```
printf("\nPROCESS\t BURST TIME \t WAITING TIME\tTURNAROUND TIME\n");
for(i=0;i<n;i++)
        printf("%d \t %d \t\t %d \t\t %d \n",i+1,ct[i],wa[i],tat[i]);
printf("\nThe Average Turnaround time is -- %f",att/n);
printf("\nThe Average Waiting time is -- %f ",awt/n);
}
```

Lecture-17 By Vivek Dubey

Output

Lecture-17 By Vivek Dubey

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Enter the no of processes -- 3

Enter Burst Time for process 1 -- 4

Enter Burst Time for process 2 -- 3

Enter Burst Time for process 3 -- 5

Enter the size of time slice -- 2

PROCESS	BURST TIME	WAITING TIME	TURNAROUND TIME
1	4	4	8
2	3	6	9
3	5	7	12

The Average Turnaround time is -- 9.666667
The Average Waiting time is -- 5.666667