Ex. No.: 6d) Date 20/3 /25

## ROUND ROBIN SCHEDULING

Aim:

To implement the Round Robin (RR) scheduling technique

## Algorithm:

1. Declare the structure and its elements.

Get number of processes and Time quantum as input from the user.

3. Read the process name, arrival time and burst time

4. Create an array rem\_bt[] to keep track of remaining burst time of processes which is initially copy of bt[] (burst times array)

5. Create another array wt[] to store waiting times of processes. Initialize this array as 0. 6.

Initialize time: t = 0

Keep traversing the all processes while all processes are not done. Do following for i'th process if it is not done yet.

a- If rem bt[i] > quantum

(i) t = t + quantum

(ii) bt\_rem[i] -= quantum;

b- Else // Last cycle for this process

(i) t = t + bt rem[i];

(ii) wt[i] = t - bt[i]

(iii) bt rem[i] = 0; // This process is over

- Calculate the waiting time and turnaround time for each process.
- Calculate the average waiting time and average turnaround time.
- 10. Display the results.

Program Code: # include cstdia. A > int main () " suit q, n; Rrintf ("Enter the number of process; "); scanf (" Tod", & n); to Cn) tot Cn), ve int 61 (n) at (n), wt (n), let (n), rest (n), et (n), confiestes; float lotal tat = 0; lotal\_ wt = 6; Bor (int i=0; i=n; i++) Rrint ("Process ") od Burst Dimi.", i+1);
scanf ("" od", & bt [i]);
frint ("Process "od Arrived Jime:", i+1); Scanf ("") d", & at [i]);

```
fruit ("Enter the time quantine");
 scanf (" "/od", (a);
while (comp on)
 int done =1;
 for (int i=0; i < n; i+1)
(if (nt (i)>0 && at (i) e=t)
for(int i=0; i cn; i++)
 of (ret [i] o && at [i] == tim ) {
  xfci)= qi
clas &
    t+ = rt (i);
    rt EiJ=0,
    ct [i]=ti
    tat [i] = et [i] - at [i];
     wt [i] = tot [i] - bt[i];
     Lota - let += let [i];
     lotal _wt += wt (i);
     comp ++;
   4
3
of (done) the time ++;
```

float any total total tot (n;

float any wt = total wt (n;

fruitf (" Parens Burst Ini Aminal Die aumoround Die Waiting Dim (n");

for (int i = 0; i < n; i + 1)

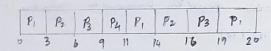
Fruitf (" Yod "I. d "I. d "I. d "I. d "I. d", i + 1;

bt [i], at [i], tat [i], cut [i];

Printf (" Amage Jum Around time = "1.2f", any tat);

Printf (" Amage Waiting Jum = "1.2f", any tut);

when o;



91 8/2 8/2 8/1 8/4 8/3 8/2 8/1

may white dance to it is no wet

Proces	AT (ms)	OT(ms)	c7 (ms)	TAT = ET - AT (ms)	WT = 747 - BT
1	0	17	20	26	13
2	1	5	16	15	10
3	2	6	19	17	1.]
4	3	2	11	8	6

## Sample Output:

inter lotal Number of	Processes.	#		
Miter Details of Free	ess[1]	五年 東京		1
erival Time 0 unst Time 4		Property and F	11.52	1 4
nten Details of Productive 1 Prival lime: 1 Parst Time: 7	es-[z]			
nter Details of Proto orival Time2 urst Time5	estil Variati		The site	, i
hter Details of Proce				
orival lime 3 unat Time of the ball	176. 6 %	When a the	+ 11 to 11	40
nter Time Quantum.				
recess 10	Hurst Time	Turnaround Time	withing for	
oress[1]	4	alar L		
roces[J]	<i>\$</i>	16 18	11	
rocess[4])	·	21	14	
verage Waiting Time:	11,500000 17,000000			

					STATE OF THE PARTY
Enter the	no of prous	: 4	Enter the	time quan	tum: 3
5 6 2 Procus	Ania	tine	2 3 Berrot Em	waiting Jim	Then Area to
			ecros am	J	
4	3		2	6	,
2	1		5	10	15
3	2		ь	11	17
1	0		7	13	20

Any waiting time is: 10 ms

Result:

Thus, the implementation of Round Robin cov scheduling has been executed succesfully.

8th