EXPERIMENT NO. 3

AIM: Write a program to implement Round Robin scheduling algorithm for process management.

RESOURCES REQUIRED:

H/W Requirements: P-IV and above, Ram 128 MB, Printer, Internet Connection.

S/W Requirements: Python compiler.

THEORY:

Round Robin is a CPU scheduling algorithm where each process is assigned a fixed time slot in a cyclic way.

It is simple, easy to implement, and starvation-free as all processes get fair share of CPU.

One of the most commonly used technique in CPU scheduling as a core.

It is pre-emptive as processes are assigned CPU only for a fixed slice of time at most.

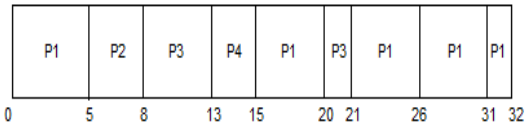
The disadvantage of it is more overhead of context switching.

Steps to find waiting times of all processes:

1. Create an array **rem_bt[]** to keep track of remaining burst time of processes. This array is initially a copy of bt[] (burst times array).
2. Create another array **wt[]** to store waiting times of process. Initialize this array as 0.
3. Initialize time: t = 0.
4. 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. rem_bt[i] -= quantum
    b. Else
        i. t = t + rem_bt[i]
        ii. wt[i] = t – bt[i]
        iii. rem_bt[i] = 0; // This process is over.

PROCESS	BURST TIME
P1	21
P2	3
P3	6
P4	2

The GANTT chart for round robin scheduling will be,



The average waiting time will be, 11 ms.

CONCLUSION: Hence, we have implemented a program on Round Robin scheduling algorithm on process management.

CODE:

```
from prettytable import PrettyTable
def rr():
    pid,at,bt,tt,wt = [],[],[],[],[]
    print()
    z = int(input("Enter number of Process: "))
    ct = [0]*z
    quantum = int(input("Enter the quantum time: "))
    print()

    for i in range(0,z):
        pid.append(int(input("Enter Proccess id:")))
        print()
        at.append(int(input("Enter arrival time:")))
```

```
    print()
    bt.append(int(input("Enter burst time:")))
    print()
for i in range(0,z):
    min = [pid[i],at[i],bt[i]]
    j = i-1
    while(j>=0 and at[j]>min[1]):
        at[j+1],pid[j+1],bt[j+1] = at[j],pid[j],bt[j]
        j = j-1
    pid[j+1],at[j+1],bt[j+1] = min[0],min[1],min[2]

rem_bt = bt.copy()
tot = 0
while(True):
    status = True
    for x in range(0,z):
        if rem_bt[x] > 0:
            status = False

            if (rem_bt[x]-quantum)>0:
                rem_bt[x] -= quantum
                tot += quantum
            else:
                tot += rem_bt[x]
                ct[x] = tot+1
                rem_bt[x] = 0
    if status:
```

```
            break
    tt.append(ct[0]-at[0])
    wt.append(tt[0]-bt[0])
    for i in range(1,z):
        tt.append(ct[i]-at[i])
        wt.append(tt[i]-bt[i])
    x = PrettyTable()
    x.field_names = ["Process id","Arrival Time","Burst Time","Completion Time","Turnaround Time","Waiting Time"]


    for a,b,c,d,e,f in zip(pid,at,bt,ct,tt,wt):
        x.add_row([a,b,c,d,e,f])


    print(x)


    print("Total turnaround time: "+str(sum(tt))+"\nTotal waiting time: "+str(sum(wt)))
    print("Average turnaround time: "+str(sum(tt)/z)+"\nAverage waiting time: "+str(sum(wt)/z))


if __name__ == "__main__":
    print("55_Adnan_Shaikh")
    rr()
```

OUTPUT:

```
Command Prompt

C:\Users\adnan\OneDrive\Desktop\College\Sem 4\OS\Scheduling algorithm>python roundrobin.py
55_Adnan_Shaikh

Enter number of Process: 4
Enter the quantum time: 3

Enter Proccess id:1

Enter arrival time:1

Enter burst time:10

Enter Proccess id:2

Enter arrival time:2

Enter burst time:4

Enter Proccess id:3

Enter arrival time:3

Enter burst time:5

Enter Proccess id:4

Enter arrival time:4

Enter burst time:3

+------------+--------------+------------+-----------------+-----------------+--------------+
| Process id | Arrival Time | Burst Time | Completion Time | Turnaround Time | Waiting Time |
+------------+--------------+------------+-----------------+-----------------+--------------+
|     1      |      1       |     10     |        23       |        22       |      12      |
|     2      |      2       |     4      |        17       |        15       |      11      |
|     3      |      3       |     5      |        19       |        16       |      11      |
|     4      |      4       |     3      |        13       |        9        |      6       |
+------------+--------------+------------+-----------------+-----------------+--------------+
Total turnaround time: 62
Total waiting time: 40
Average turnaround time: 15.5
Average waiting time: 10.0
```