Name: Abdullah Khan

ment #5

Roll no: 122A1002

Aim: Process Management: Scheduling:

Write a program to implement any two CPUscheduling algorithms like FCFS, SJF, Round Robin etc.

Theory:

CPU scheduling is a process which allows one process to use the CPU while the execution of another process is on hold(in waiting state) due to unavailability of any resource like I/O etc, thereby making full use of CPU. The aim of CPU scheduling is to make the system efficient, fastand fair.

Major scheduling algorithms are:

1. First Come First Serve(FCFS) Scheduling
2. Shortest-Job-First(SJF) Scheduling
3. Priority Scheduling
4. Round Robin(RR) Scheduling

First Come First Serve(FCFS) Scheduling

- Jobs are executed on first come, first serve basis.
- Easy to understand and implement.
- Poor in performance as average wait time is high.

PROCESS	BURST TIME
P1	21
P2	3
P3	6
P4	2

The average waiting time will be = (0 + 21 + 24 + 30)/4 = 18.75 ms

This is the GANTT chart for the above processes

Shortest-Job-First(SJF) Scheduling

- Best approach to minimize waiting time.
- Actual time taken by the process is already known to processor.
- Impossible to implement.

PROCESS	BURST TIME
P1	21
P2	3
P3	6
P4	2

In Shortest Job First Scheduling, the shortest Process is executed first.

Hence the GANTT chart will be following :

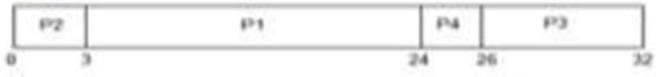
Now, the average waiting time will be = (0 + 2 + 5 + 11)/4 = 4.5 ms

Priority Scheduling

Priority is assigned for each process.
Process with highest priority is executed first and so on.
Processes with same priority are executed in FCFS manner.
Priority can be decided based on memory requirements, time requirements or any other resource requirement.

PROCESS	BURST TIME	PRIORITY
P1	21	2
P2	3	1
P3	6	4
P4	2	3

The GANTT chart for following processes based on Priority scheduling will be.



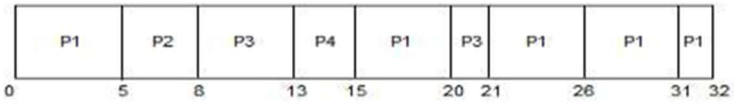
The average waiting time will be, (0 + 3 + 24 + 26)/4 = 13.25 ms

Round Robin(RR) Scheduling

- A fixed time is allotted to each process, called **quantum**, for execution.
- Once a process is executed for given time period that process is preempted and other processexecutes for given time period.
- Context switching is used to save states of preempted processes.

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.

FCFS Code:

```c
#include<stdio.h>
int main()
{
    int n,bt[30],wait_t[30],turn_ar_t[30],av_wt_t=0,avturn_ar_t=0,i,j;
    printf("Please enter the total number of processes(maximum 30):");  // the maximum process that be used to calculate is specified.
    scanf("%d",&n);
    printf("\nEnter The Process Burst Timen");
    for(i=0;i<n;i++)  // burst time for every process will be taken as input
    {
        printf("P[%d]:",i+1);
        scanf("%d",&bt[i]);
    }
    wait_t[0]=0;
    for(i=1;i<n;i++)
    {
        wait_t[i]=0;
        for(j=0;j<i;j++)
            wait_t[i]+=bt[j];
    }
    printf("\nProcess\t\tBurst Time\tWaiting Time\tTurnaround Time");
    for(i=0;i<n;i++)
    {
        turn_ar_t[i]=bt[i]+wait_t[i];
        av_wt_t+=wait_t[i];
        avturn_ar_t+=turn_ar_t[i];
        printf("\nP[%d]\t\t%d\t\t%d\t\t%d",i+1,bt[i],wait_t[i],turn_ar_t[i]);
    }
    av_wt_t/=i;
    avturn_ar_t/=i;  // average calculation is done here
    printf("\nAverage Waiting Time:%d",av_wt_t);
    printf("\nAverage Turnaround Time:%d",avturn_ar_t);
    return 0;
}
```

FCFS Output:

```
siesgst@siesgst-OptiPlex-3020: ~/04
siesgst@siesgst-OptiPlex-3020:~/04$ gcc -o fcfs fcfs.c
siesgst@siesgst-OptiPlex-3020:~/04$ ./fcfs
Please enter the total number of processes(maximum 30):5

Enter The Process Burst TimenP[1]:10
P[2]:5
P[3]:7
P[4]:2
P[5]:8

Process         Burst Time      Waiting Time    Turnaround Time
P[1]            10                      0                               10
P[2]            5                       10                              15
P[3]            7                       15                              22
P[4]            2                       22                              24
P[5]            8                       24                              32
Average Waiting Time:14
Average Turnaround Time:20siesgst@siesgst-OptiPlex-3020:~/04$
siesgst@siesgst-OptiPlex-3020:~/04$
```

SJF Code:

```c
#include <stdio.h>
int main()
{

        int A[100][4];
        int i, j, n, total = 0, index, temp;
        float avg_wt, avg_tat;
        printf("Enter number of process: ");
        scanf("%d", &n);
        printf("Enter Burst Time:\n");

        for (i = 0; i < n; i++) {
                printf("P%d: ", i + 1);
                scanf("%d", &A[i][1]);
                A[i][0] = i + 1;
        }
        for (i = 0; i < n; i++) {
                index = i;
                for (j = i + 1; j < n; j++)
                        if (A[j][1] < A[index][1])
                index = j;
                temp = A[i][1];
                A[i][1] = A[index][1];
                A[index][1] = temp;

                temp = A[i][0];
                A[i][0] = A[index][0];
                A[index][0] = temp;
        }
        A[0][2] = 0;

        for (i = 1; i < n; i++) {
                A[i][2] = 0;
```

```
        for (i = 1; i < n; i++) {
                A[i][2] = 0;
                for (j = 0; j < i; j++)
                        A[i][2] += A[j][1];
                total += A[i][2];
        }
        avg_wt = (float)total / n;
        total = 0;
        printf("P     BT     WT     TAT\n");

        for (i = 0; i < n; i++) {
                A[i][3] = A[i][1] + A[i][2];
                total += A[i][3];
                printf("P%d     %d     %d     %d\n", A[i][0],
                A[i][1], A[i][2], A[i][3]);
        }
        avg_tat = (float)total / n;
        printf("Average Waiting Time= %f", avg_wt);
        printf("\nAverage Turnaround Time= %f", avg_tat);
}
```

SJF Output:

```
siesgst@siesgst-OptiPlex-3020:~/04$ gcc -o sjf sjf.c
siesgst@siesgst-OptiPlex-3020:~/04$ ./sjf
Enter number of process: 5
Enter Burst Time:
P1: 10
P2: 20
P3: 4
P4: 8
P5: 2
P   BT   WT   TAT
P5   2   0   2
P3   4   2   6
P4   8   6   14
P1   10   14   24
P2   20   24   44
Average Waiting Time= 9.200000
Average Turnaround Time= 18.000000siesgst@siesgst-OptiPlex-3020:~/04$
siesgst@siesgst-OptiPlex-3020:~/04$
```

Conclusion: The two algorithms are implemented from the above list.