

EXPERIMENT -6 :SCHEDULING ALGORITHM

Scheduling Algorithms in OS:-

- There are various algorithms that are used by the Operating System to schedule the processes on the processor in an efficient way.

The Purpose of a Scheduling Algorithm:-

- Maximum CPU utilization
- Fair allocation of CPU
- Maximum throughput
- Minimum turnaround time
- Minimum waiting time
- Minimum response time

Different Type of algorithms:-

- First Come First Serve
- Round Robin
- Shortest Job First
- Shortest remaining time first
- Priority based scheduling
- Highest Response Ratio Next

1) First Come First Serve:-

It is the simplest algorithm to implement. The process with the minimal arrival time will get the CPU first. The lesser the arrival time, the sooner will the process get the CPU. It is the non-preemptive type of scheduling.

```
// FCFS algorithm assuming each process arrives at t = 0
#include <stdio.h>
#include <unistd.h>

void main() {
    int n, i;
    float avgwt = 0, avgtat = 0;
    printf("Enter No. Of Processes");
    scanf("%d", &n);
    int bt[n], wt[n], tat[n];
    printf("Enter the burst Time (BT) of the processes:");
    for (i = 0; i < n; i++)
    {
        scanf("%d", &bt[i]);
    }
    wt[0] = 0;
    tat[0] = bt[0];
    for (i = 1; i < n; i++)
    {
        wt[i] = wt[i - 1] + bt[i - 1];
        tat[i] = wt[i] + bt[i];
    }
    printf("P-id\tBT\tWT\tTAT");
    printf("\n");
    for (i = 0; i < n; i++)
    {
        printf("%d\t%d\t%d\t%d\t%d\n", i + 1, bt[i], wt[i], tat[i]);
        avgwt += wt[i];
        avgtat += tat[i];
    }
    avgwt /= n;
    avgtat /= n;
    printf("Average Waiting Time(WT):%f", avgwt);
    printf("Average Turn Around Time(TAT):%f", avgtat);
}
```

```
(base) om-college@OM-M-PATEL-MACBOOK-M1-AIR output % ./"FCFS"
Enter the number of processes:5
Enter the burst time of the processes:6 8 9 2
1 4 2 5 0
Process Burst Time      Waiting Time      Turn Around Time
1        6             0                6
2        8             6               14
3        9            14               23
4        2            23               25
5        1            25               26
Average Waiting Time:13.600000Average Turn Around Time:18.799999%
```

2) Shortest Job First:-

The job with the shortest burst time will get the CPU first. The lesser the burst time, the sooner will the process get the CPU. It is the non-preemptive type of scheduling.

```
//write SJF assuming each process arrives at t=0
#include <stdio.h>
#include <unistd.h>

void main()
{
    int n, i, j;
    float avgwt = 0, avgtat = 0;
    printf("Enter the number of processes:");
    scanf("%d", &n);
    int bt[n], wt[n], tat[n], p[n];
    printf("Enter the burst time of the processes:");
    for (i = 0; i < n; i++)
    {
        scanf("%d", &bt[i]);
        p[i] = i + 1;
    }
    for (i = 0; i < n; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (bt[i] > bt[j])
            {
                int temp = bt[i];
                bt[i] = bt[j];
                bt[j] = temp;
                temp = p[i];
                p[i] = p[j];
                p[j] = temp;
            }
        }
    }
    wt[0] = 0;
    tat[0] = bt[0];
    for (i = 1; i < n; i++)
```

```

    {
        wt[i] = wt[i - 1] + bt[i - 1];
        tat[i] = wt[i] + bt[i];
    }
    printf("Process\tBurst Time\tWaiting Time\tTurn Around Time");
    printf("\n");
    for (i = 0; i < n; i++)
    {
        printf("%d\t%d\t%d\t%d\n", p[i], bt[i], wt[i], tat[i]);
        avgwt += wt[i];
        avgtat += tat[i];
    }
    avgwt /= n;
    avgtat /= n;
    printf("\n");
    printf("Average Waiting Time: %.2f\n", avgwt);
    printf("Average Turn Around Time: %.2f\n", avgtat);
}
```

```
(base) om-college@OM-M-PATEL-MACBOOK-M1-AIR ~ % /Users/om/OS-LAB/OS-7\,8/SJF ; exit;
```

```
Enter the number of processes:3
```

```
Enter the burst time of the processes:3 3 4
```

Process	Burst Time	Waiting Time	Turn Around Time
1	3	0	3
2	3	3	6
3	4	6	10

```
Average Waiting Time: 3.00
```

```
Average Turn Around Time: 6.33
```