

Lab Report No. 07

Lab Report Name: Implementation of FCFS Scheduling algorithm .

Objectives:

- i. What is FCFS Scheduling algorithm.
- ii. How to implementation in C

Theory:

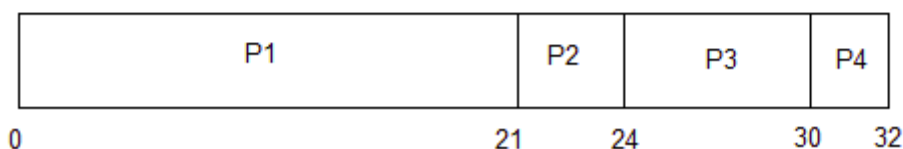
First Come First Served (FCFS) is a Non-Preemptive scheduling algorithm. FIFO (First In First Out) strategy assigns priority to process in the order in which they request the processor. The process that requests the CPU first is allocated the CPU first. This is easily implemented with a FIFO queue for managing the tasks. As the process come in, they are put at the end of the queue. As the CPU finishes each task, it removes it from the start of the queue and heads on to the next task.

Example:

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



C program for implementing FCFS:

```
#include<stdio.h>
void findWaitingTime(int processes[], int n,int bt[], int wt[])
{
    wt[0] = 0;

    for (int i = 1; i < n ; i++)
        wt[i] = bt[i-1] + wt[i-1] ;
}
```

```
}
```

```
void findTurnAroundTime( int processes[], int n,int bt[], int wt[], int tat[])
{
    for (int i = 0; i < n ; i++)
        tat[i] = bt[i] + wt[i];
}
```

```
void findavgTime( int processes[], int n, int bt[])
{
    int wt[n], tat[n], total_wt = 0, total_tat = 0;

    findWaitingTime(processes, n, bt, wt);

    findTurnAroundTime(processes, n, bt, wt, tat);

    printf("\n\nProcesses  Burst time  Waiting time  Turn around time\n");

    for (int i=0; i<n; i++)
    {
        total_wt = total_wt + wt[i];
        total_tat = total_tat + tat[i];
        printf(" %d      %d      %d      %d\n", (i+1), bt[i], wt[i], tat[i]);
    }
    double s=(float)total_wt / (float)n;
    double t=(float)total_tat / (float)n;

    printf("\nAverage waiting time = %f\n",s);
    printf("\n");
    printf("Average turn around time = %f \n",t);
}
```

```
int main()
{
    int n;
    printf("Enter the number of process : ");
    scanf("%d",&n);

    int processes[n],burst_time[n];
    int i=0;
    while(i<n){
        //for(int i =0; i<n; i++){
            printf("Enter process %d bust time : ",i+1);
            processes[i] = i+1;

            scanf("%d",&burst_time[i]);
        i++;
    }
```

```
}  
  
findavgTime(processes, n, burst_time);  
return 0;  
}
```

output:

```
Enter the number of process : 3  
Enter process 1 bust time : 5  
Enter process 2 bust time : 6  
Enter process 3 bust time : 7  
  
Processes    Burst time    Waiting time    Turn around time  
1            5            0              5  
2            6            5              11  
3            7            11             18  
  
Average waiting time = 5.333333  
Average turn around time = 11.333333  
  
Process returned 0 (0x0)   execution time : 7.456 s  
Press any key to continue.
```