

Ex. no: 5a
30/03/2022

Implementation of FCFS (First Come First Serve) CPU Scheduling Algorithm

Aim

To schedule snapshot of processes queued according to FCFS Scheduling.

process scheduling

- CPU scheduling is used in multiprogrammed operating systems.
- By switching CPU among processes, efficiency of the system can be improved.
- Some scheduling algorithms are FCFS, SJF, priority, Round-Robin etc..
- Gantt chart provides a way of visualizing CPU scheduling and enables to understand better.

First come First serve.

- process that comes first is processed first
- FCFS scheduling is non-preemptive.
- Not efficient as it results in long average waiting time.
- can result in starvation, if processes at beginning of the queue have long bursts.

Algorithm:

- Define an array of structure process with members pid, btime, wtime & ttime.
- Get length of the ready queue i.e. number of process.

- obtain $btime$ for each process.
- The $wtime$ for first process is 0.

compute $wtime$ and $ttime$ for each process as

$$a. wtime_{i+1} = wtime_i + btime_i$$

$$b. ttime_i = wtime_i + btime_i$$

- compute average waiting time $awat$ and average turnaround time $atur$.
- Display the $btime$, $ttime$, $wtime$ for each process.
- Display GANTT chart for the above scheduling.
- Display $awat$ time and $atur$.
- stop.

IMPLEMENTATION OF FCFS CPU SCHEDULING ALGORITHM

PROGRAM:

```

#include <stdio.h>
struct process
{
    int pid;
    int btime;
    int wtime;
    int ttime;
} p[10];
int main()
{
    int i,j,k,n,ttur,twat;
    float awat,atur;
    printf("Enter no. of process : ");
    scanf("%d", &n);
    for(i=0; i<n; i++)
    {
        printf("Burst time for process P%d (in ms) : ",(i+1));
        scanf("%d", &p[i].btime);
        p[i].pid = i+1;
    }
    p[0].wtime = 0;
    for(i=0; i<n; i++)
    {

```

```

        p[i+1].wtime = p[i].wtime + p[i].btime;
        p[i].ttime = p[i].wtime + p[i].btime;
    }
    ttur = twat = 0;
    for(i=0; i<n; i++)
    {
        ttur += p[i].ttime;
        twat += p[i].wtime;
    }
    awat = (float)twat / n;
    atur = (float)ttur / n;
    printf("\n FCFS Scheduling\n\n");
    for(i=0; i<28; i++)
        printf("-");
    printf("\nProcess B-Time T-Time W-Time\n");
    for(i=0; i<28; i++)
        printf("-");
    for(i=0; i<n; i++)
        printf("\n P %d\t%4d\t%3d\t%2d",p[i].pid,p[i].btime,p[i].ttime,p[i].wtime);

    printf("\n");
    for(i=0; i<28; i++)
        printf("-");

    printf("\n\nAverage waiting time: %5.2fms", awat);
    printf("\n\nAverage turn around time : %5.2fms\n", atur);
    printf("\n\nGANTT Chart\n");
    printf("-");
    for(i=0; i<(p[n-1].ttime + 2*n); i++)

```

```

    for(i=0; i<28; i++)
        printf("-");
    for(i=0; i<n; i++)
        printf("\n P%-4d\t%-4d\t%-3d\t%-2d", p[i].pid, p[i].btime, p[i].ttime, p[i].wtime);
    printf("\n");
    for(i=0; i<28; i++)
        printf("-");
    printf("\n\nAverage waiting time: %5.2fms", awat);
    printf("\n\nAverage turn around time : %5.2fms\n", atur);
    printf("\n\n GANTT Chart\n");
    printf("-");
    for(i=0; i<(p[n-1].ttime + 2*n); i++)
        printf("-");
    printf("\n");
    for(i=0; i<n; i++)
    {
        k=p[i].btime/2;
        for(j=0; j<k; j++)
            printf("-");
        printf("P%d", p[i].pid);
        for(j=k+1; j<p[i].btime; j++)
            printf(" ");
        printf("|");
    }
    printf("\n~");
    for(i=0; i<(p[n-1].ttime + 2*n); i++)
        printf("-");

```



```

        printf("-");
    printf("\n");
    printf("|");
    for(i=0; i<n; i++)
    {
        k = p[i].btime/2;
        for(j=0; j<k; j++)
            printf(" ");
        printf("P%d",p[i].pid);
        for(j=k+1; j<p[i].btime; j++)
            printf(" ");
        printf("|");
    }
    printf("\n");
    printf("-");
    for(i=0; i<(p[n-1].ttime + 2*n); i++)
        printf("-");
    printf("\n");
    printf("0");
    for(i=0; i<n; i++)
    {
        for(j=0; j<p[i].btime; j++)
            printf(" ");
        printf("%2d",p[i].ttime);
    }
}

```

Result

Thus waiting time and turnaround time for processes based on FCFS scheduling was computed and the average waiting time was determined.

A red checkmark and a stylized red signature or mark.