

Ques. 3. Consider a scheduler which schedules the job by considering the arrival time of the processes where arrival time if given as 0 is discarded or displayed as error. The scheduler implements the shortest job first scheduling policy, but checks the queue of the processes after the every process terminates and time taken for checking and arranging the process according to the shortest job is 2 time unit. Compute the waiting time, turnaround time and average waiting time and turnaround time of the processes. Also compute the total time taken by the processor to compute all the jobs.

The inputs for the number of requirements, arrival time and burst time should be provided by the user.

Consider the following units for reference.

Develop a scheduler which submits the processes to the processor in the defined scenario, and compute the scheduler performance by providing the waiting time for process, turnaround time for process and average waiting time and turnaround time.

Name:- YASH ASHISH PARTE

ID:- 11810532

Emailaddress:- yashaparte@gmail.com

1.Description :-

In this problem we use shortest job first ,in this we have given arrival waiting time with processes then we have to choose the arrivale time. If shortest job first is an algorithm in which the process having the smallest execution time is chosen for the next execution. This scheduling method can be preemptive or non-preemptive. It significantly reduces the average waiting time for other processes awaiting execution. The full form of SJF is Shortest Job First.

2.Algorithm:-

- 1.Sort all the processes according to their arrival time.
- 2.Select the process with minimum arrival time as well as minimum burst time.
- 3.After completion of the process, select from the ready queue the process which has the minimum burst time.
- 4.Repeat above processes untill all processes have finished their execution.

3.Complexity:-

The complexity of the SJF is $O(n \log n)$.

4.Code Snippet

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char s;
    int num;
    long length;
    FILE *fi;
    printf("Enter the value of num : ");
    scanf("%d", &num);
    fi = fopen("C:\\Newprogram.txt", "w+");
```

```

if(fi == NULL)
{
printf("error");
exit(1);
}
printf("Enter anything: ");
scanf("%d",&s);
fprintf(fi,"%d",s);
fclose(fi);
fseek(fi, 0, SEEK_END);
length = ftell(fi);
fseek(fi,(length-num), SEEK_SET);
do {
s = fgetc(fd);
putchar(s);
} while (s != EOF);
fclose(fd);
return(0);
}

```

6. Boundary conditions:-

- a. SJF is associated with each jobs as a unit time to complete
- b. useful for batch-type processing where waiting for jobs to complete is not critical.
- c. output offering is improves in it

7. Test cases:

- a) at time=0, burst time=6 so process p2 and p3 are in ready queue then p3 have short burst time will serve first and then p2 will implement.
- b) at time=9, total time will be 16 so process p5,p6,p7,p8 are in ready queue and will be arranged as the shortest burst time. p6 having burst time 3 and shortest then p7 having 4 burst time then likely p5 and p8 will be arranged.
- c) at process p9 and p10 the burst time of p10 is shorter as compared to the p9 so p10 will arranged first
completion time:- p1 CT=6 TAT=CT-AT =6
p2 CT=9 6
p3 CT=7 3
p4 CT=16 7
p5 CT=28 18
p6 CT=19 7
p7 CT=23 9
p8 CT=33 17
p9 CT=42 25
p10CT=35 16.