

CM25D008

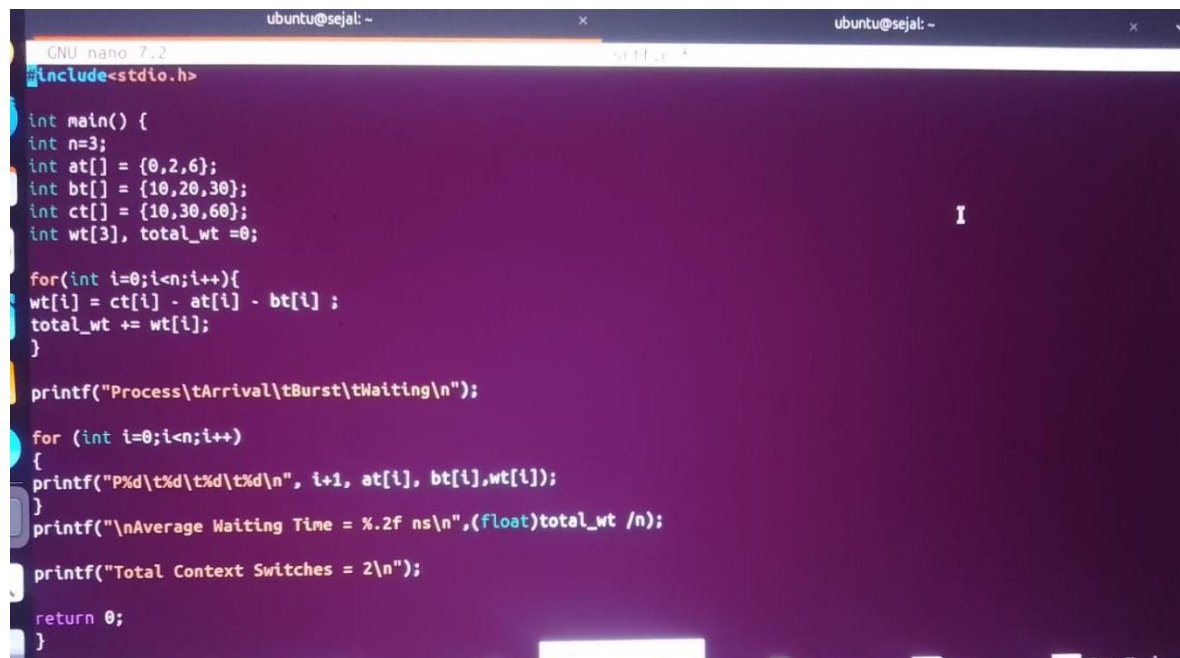
PRACTICAL NO :- 04

Output :-

In an operating system, three CPU-intensive processes are ready for execution which require 10 ns, 20 ns, and 30 ns and arrive at times 0 ns, 2 ns, and 6 ns, respectively. Write a program to calculate:

- The total number of context switches needed if the operating system implements Shortest Job First (Preemptive) scheduling algorithm.
- The average waiting time of the processes before getting the CPU

Code:-

A screenshot of a terminal window showing a C program in the nano editor. The program calculates the average waiting time and total context switches for three processes using the Shortest Job First (SJF) scheduling algorithm. The processes have arrival times at[] = {0, 2, 6}, burst times bt[] = {10, 20, 30}, and completion times ct[] = {10, 30, 60}. The waiting times wt[] are calculated as {0, 8, 24}. The program outputs the process details and the final results: Average Waiting Time = 10.67 ns and Total Context Switches = 2.

```
GNU nano 7.2
#include<stdio.h>

int main() {
int n=3;
int at[] = {0,2,6};
int bt[] = {10,20,30};
int ct[] = {10,30,60};
int wt[3], total_wt =0;

for(int i=0;i<n;i++){
wt[i] = ct[i] - at[i] - bt[i] ;
total_wt += wt[i];
}

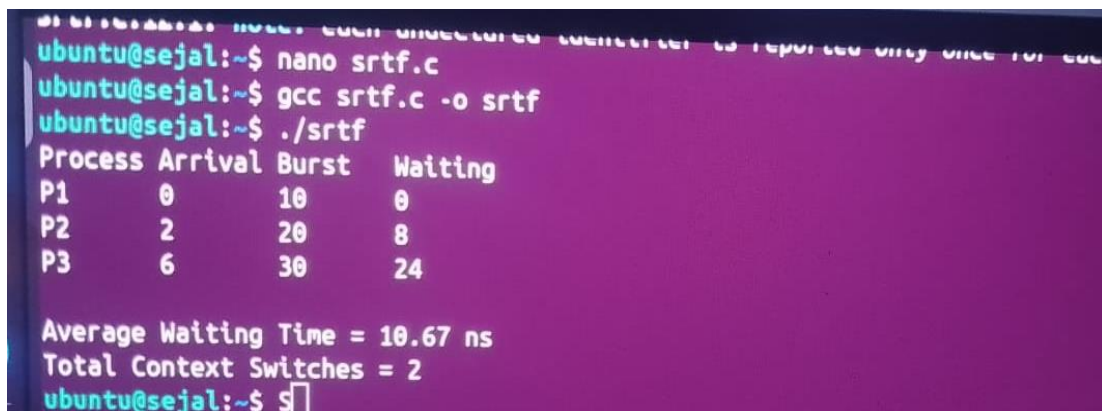
printf("Process\tArrival\tBurst\tWaiting\n");

for (int i=0;i<n;i++)
{
printf("P%d\t%d\t%d\t%d\n", i+1, at[i], bt[i],wt[i]);
}
printf("\nAverage Waiting Time = %.2f ns\n",(float)total_wt /n);

printf("Total Context Switches = 2\n");

return 0;
}
```

Output:-

A screenshot of a terminal window showing the execution of the program. The user runs 'nano srtf.c', 'gcc srtf.c -o srtf', and './srtf'. The output displays a table of process details and the calculated average waiting time and total context switches.

```
ubuntu@sejal:~$ nano srtf.c
ubuntu@sejal:~$ gcc srtf.c -o srtf
ubuntu@sejal:~$ ./srtf
Process Arrival Burst   Waiting
P1      0       10       0
P2      2       20       8
P3      6       30      24

Average Waiting Time = 10.67 ns
Total Context Switches = 2
ubuntu@sejal:~$
```