**Experiment 2 : FCFS Scheduling**

**Code :**

#include<stdio.h>

int main() {

int n,awt=0,atat=0,i,j;

printf("FCFS Scheduling Algorithm\nEnter Total Number Of Process : ");

scanf("%d",&n);

printf("Enter Process Burst Time : \n");

int bt[n],wt[n],tu[n];

for(i=0;i<n;i++) {

printf("P[%d] : ",i+1);

scanf("%d",&bt[i]);

}

wt[0]=0;

for(i=1;i<n;i++) {

wt[i]=0;

for(j=0;j<i;j++) {

wt[i]=wt[i]+bt[j];

}

}

printf("\nProcess BT\tWT\tTAT");

for(i=0;i<n;i++) {

tu[i]=bt[i]+wt[i];

awt=awt+wt[i];

atat=atat+tu[i];

printf("\nP[%d]\t%d\t%d\t%d",i+1,bt[i],wt[i],tu[i]);

}

awt=awt/i;

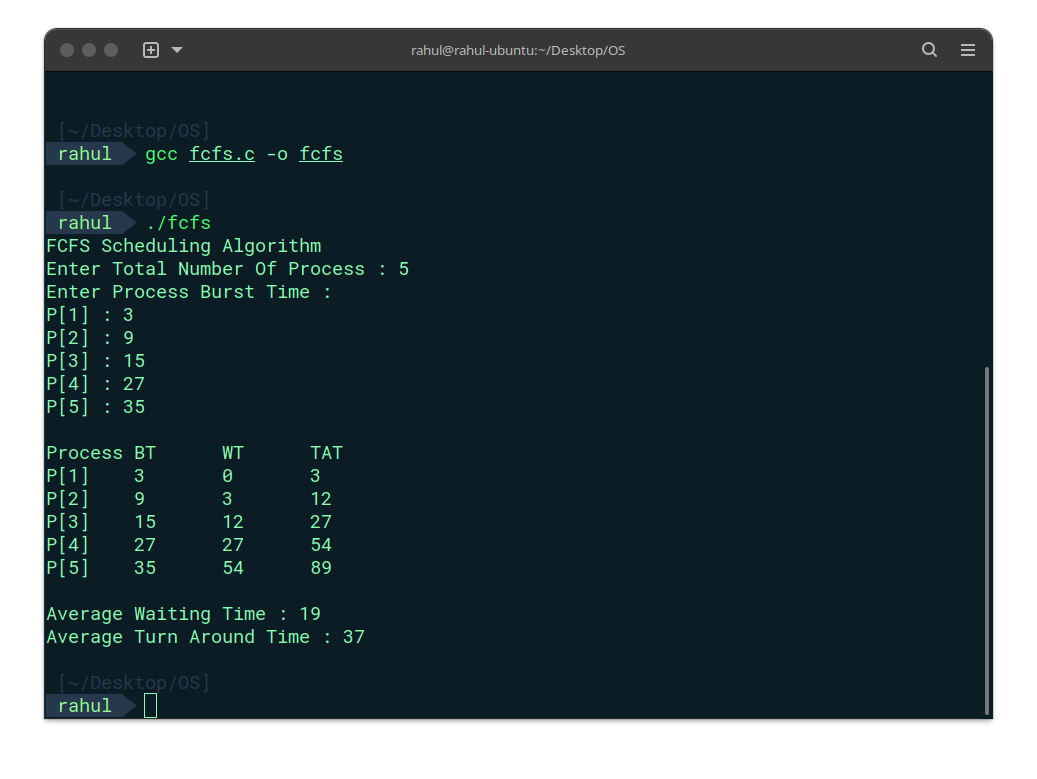
atat=atat/i;

printf("\n\nAverage Waiting Time : %d",awt);

printf("\nAverage Turn Around Time : %d\n",atat);

return 0;

}

**Output :**

**Experiment 2b: SJF Scheduling**

**Code :**

#include<stdio.h>

struct p {

int process;

int burst;

};

int main() {

int n;

printf("SJF Scheduling Algorithm\nEnter the no of processes:");

scanf("%d",&n);

int wt[n],bt[n],tat[n],ct[n],atat=0,awt=0,temp1=0,temp2=0;

struct p arr[n];

printf("Enter the burst time for the processes:\n");

for(int i=0;i<n;i++) {

printf("P[%d] : ",i+1);

arr[i].process=i+1;

scanf("%d",&bt[i]);

arr[i].burst = bt[i];

}

for(int i=0;i<n-1;i++) {

for(int j=i+1;j<n;j++) {

if(arr[i].burst>arr[j].burst) {

temp1 = arr[i].process;

temp2 = arr[i].burst;

arr[i].process = arr[j].process;

arr[i].burst = arr[j].burst;

arr[j].process = temp1;

arr[j].burst = temp2;

}

}

}

wt[0]=0;

tat[0]=arr[0].burst;

atat = arr[0].burst;

for(int i=1;i<n;i++) {

wt[i] = wt[i-1] + arr[i-1].burst;

tat[i] = tat[i-1] + arr[i].burst;

awt = awt + wt[i];

atat = atat + tat[i];

}

atat=atat/n;

awt=awt/n;

printf("Process\tBT\tWT\tTAT\n");

for(int i=0;i<n;i++) {

printf("P%d\t%d\t%d\t%d\n",arr[i].process,arr[i].burst,wt[i],tat[i]);

}

printf("Average Waiting Time : %d\n",awt);

printf("Average Turn Around Time : %d\n",atat);

return 0;

}

**Output :**

1. Start.
2. Initialize variables: n, wt[n], bt[n], tat[n], ct[n], atat=0, awt=0, temp1=0, temp2=0.
3. Display "SJF Scheduling Algorithm".
4. Display "Enter the no of processes:".
5. Read the value of n.
6. Create a structure called p with two members process and burst.
7. Create an array of structures called arr with size n.
8. Display "Enter the burst time for the processes:".
9. For i=0 to n-1, do steps 10 to 12.
10. Display "P[i+1] : ".
11. Read the value of bt[i] and assign it to arr[i].burst.
12. Set arr[i].process to i+1.
13. Sort the array arr in ascending order of burst time using Selection Sort algorithm.
14. Set wt[0] to 0, tat[0] to arr[0].burst, and atat to arr[0].burst.
15. For i=1 to n-1, do steps 16 to 19.
16. Set wt[i] to wt[i-1] + arr[i-1].burst.
17. Set tat[i] to tat[i-1] + arr[i].burst.
18. Add wt[i] to awt and add tat[i] to atat.
19. End for loop.
20. Set atat to atat/n and awt to awt/n.
21. Display "Process BT WT TAT".
22. For i=0 to n-1, do step 23.
23. Display "P[arr[i].process] arr[i].burst wt[i] tat[i]".
24. Display "Average Waiting Time : awt".
25. Display "Average Turn Around Time : atat".
26. End.

**Experiment 2c: Priority Scheduling**

**Code :**

#include<stdio.h>

struct p {

int process;

int burst;

int priority;

};

int main() {

int n;

printf("Priority Scheduling Algorithm\nEnter the no of processes:");

scanf("%d",&n);

int wt[n],bt[n],tat[n],ct[n],atat=0,awt=0,temp1=0,temp2=0,temp3=0;

struct p arr[n];

printf("Enter the burst time & priority for the processes:\n");

for(int i=0;i<n;i++) {

printf("P[%d] : ",i+1);

arr[i].process=i+1;

scanf("%d",&bt[i]);

arr[i].burst = bt[i];

printf("Priority : ");

scanf("%d",&arr[i].priority);

}

for(int i=0;i<n-1;i++) {

for(int j=i+1;j<n;j++) {

if(arr[i].priority>arr[j].priority) {

temp1 = arr[i].process;

temp2 = arr[i].burst;

temp3 = arr[i].priority;

arr[i].process = arr[j].process;

arr[i].burst = arr[j].burst;

arr[i].priority = arr[j].priority;

arr[j].process = temp1;

arr[j].burst = temp2;

arr[j].priority =temp3;

}

}

}

wt[0]=0;

tat[0]=arr[0].burst;

atat = arr[0].burst;

for(int i=1;i<n;i++) {

wt[i] = wt[i-1] + arr[i-1].burst;

tat[i] = tat[i-1] + arr[i].burst;

awt = awt + wt[i];

atat = atat + tat[i];

}

atat=atat/n;

awt=awt/n;

printf("\tPriorityProcess\tBT\tWT\tTAT\n");

for(int i=0;i<n;i++) {

printf("\t%d  \tP%d\t%d\t%d\t%d\n",arr[i].priority,arr[i].process,arr[i].burst,wt[i],tat[i]);

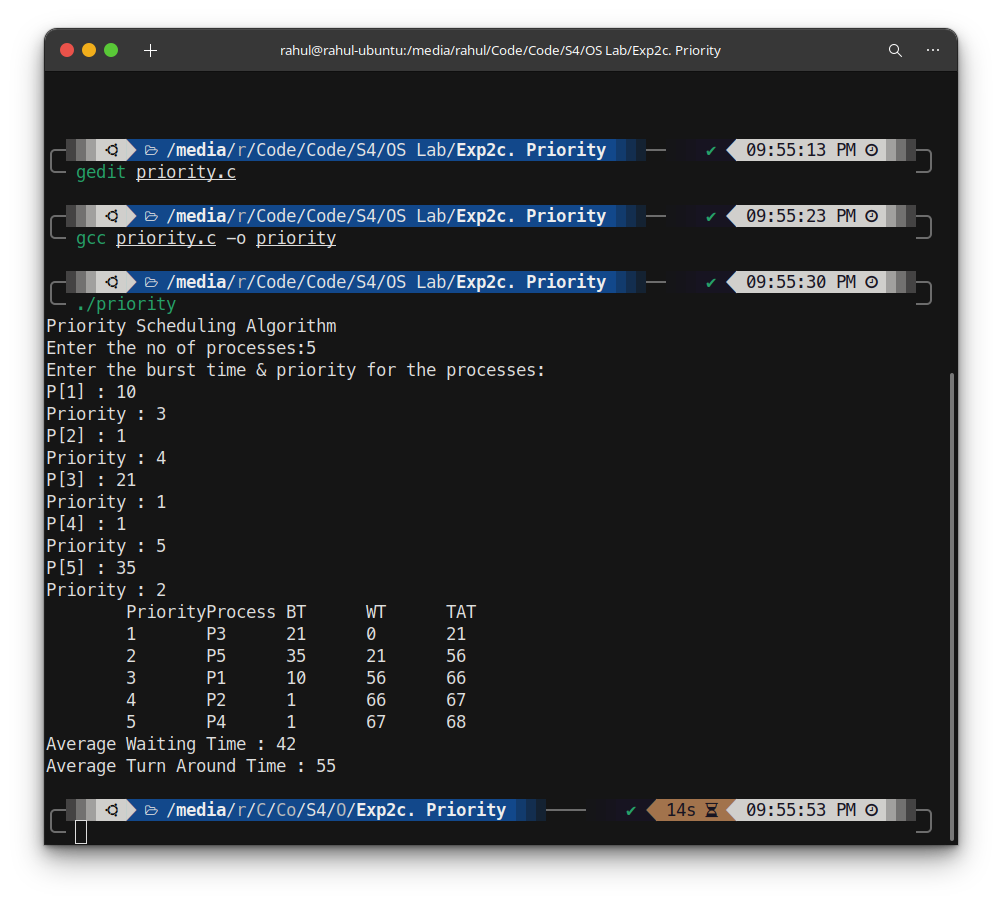
}

printf("Average Waiting Time : %d\n",awt);

printf("Average Turn Around Time : %d\n",atat);

return 0;

}

**Output :**