

Saurabh Kumar Singh(16100051)

FCFS Scheduling

ALGORITHM::

Mode:- Non-Preemptive

Criteria:- Arrival Time

- 1- Input the processes along with their burst time (bTime).
- 2- Find waiting time (wTime) for all processes.
- 3- As first process that comes need not to wait so
waiting time for process 1 will be 0 i.e. $wTime[0] = 0$.
- 4- Find waiting time for all other processes i.e. for
process i ->
 $wTime[i] = bTime[i-1] + wTime[i-1]$.
- 5- Find turnaround time = waiting_time + burst_time
for all processes.
- 6- Find average waiting time =
 $total_waiting_time / no_of_processes$.
- 7- Similarly, find average turnaround time =
 $total_turn_around_time / no_of_processes$.

CODE::

```
#include<stdio.h>
void main()
{
    float process[100],aTime[100],bTime[100],abTime[100],wTime[100],tat_time[100];
    int n = 0,i = 0 ;
    float aw_time = 0, atat_time = 0;
    printf("*** FCFS Scheduling Algorithm Using Arrival Time ***\n");
    printf("\nEnter the number of process : ");
    scanf("%d",&n);

    printf("Enter the Arrival time and Burst time.\n\n");
    printf("\tArrival_Time Burst_Time\n");
    for(i = 0 ; i < n ; i++){
        process[i]=i+1;
        printf("P%d : \t", i+1);
        scanf("%f\t%f",&aTime[i],&bTime[i]);
    }
    printf("\n\nProcess\tA_Time\tB_Time\n");
    for(i = 0 ; i < n ; i++){
        printf("P[%d]\t%.2f\t%.2f\n",i,aTime[i],bTime[i]);
    }
    wTime[0] = 0;
    tat_time[0] = bTime[0];
    abTime[0] = bTime[0]+aTime[0];
    for( i = 1 ; i < n ; i++){
        abTime[i] = abTime[i-1] + bTime[i];
        tat_time[i] = abTime[i] - aTime[i];
        wTime[i] = tat_time[i] - bTime[i];
    }
    for(i = 0 ; i < n ; i++){
        aw_time = aw_time + wTime[i];
        atat_time = atat_time + tat_time[i];
    }
    printf("\tA_time\tB_time\tC_time\tTat_time  W_time\n");
    for(i = 0 ; i < n ; i++){
        printf("P[%d]\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\n",i,aTime[i],bTime[i],abTime[i],tat_time[i],wTime[i]);
    }
    printf("\nAverage waiting time : %f",aw_time/n);
    printf("\nAverage turn around time : %f",atat_time/n);
}
```

INPUT::

Enter the number of process : 5

Enter the Arrival time and Burst time.

	A_Time	B_Time
P1 :	0	4
P2 :	1	3
P3 :	2	1
P4 :	3	2
P5 :	4	5

OUTPUT::

Process	Arrival_Time	Burst_Time
---------	--------------	------------

P[0]	0.00	4.00
P[1]	1.00	3.00
P[2]	2.00	1.00
P[3]	3.00	2.00
P[4]	4.00	5.00

	A_time	B_time	C_time	Tat_time	W_time
P[0]	0.00	4.00	4.00	4.00	0.00
P[1]	1.00	3.00	7.00	6.00	3.00
P[2]	2.00	1.00	8.00	6.00	5.00
P[3]	3.00	2.00	10.00	7.00	5.00
P[4]	4.00	5.00	15.00	11.00	6.00

Average waiting time : 3.800000

Average turnaround time : 6.800000