

# OPERATING SYSTEMS

## LAB ASSIGNMENT – 5

**Name:** - PANDUGA VENKATA JAYA SRIKANTH REDDY

**Reg No:-** 21MIS1095

## CPU SCHEDULING ALGORITHMS

### FCFC ALGORITHM:-

#### CODE:-

```
#include <stdio.h>
#include <stdlib.h>
typedef struct Process
{
    int id,at,bt,ct,tat,wt;
} process;

void sorting(int n, process p[])
{
    process temp;
    for (int i=0;i<n;i++)
    {
        for (int j=0; j<n-1;j++)
        {
            if (p[j].at>p[j+1].at)
            {
                temp = p[j];
                p[j] = p[j+1];
                p[j+1] = temp;
            }
        }
    }
}

void fcfs(int n, process p[])
{
    p[0].ct = p[0].bt + p[0].at;
    p[0].tat = p[0].ct - p[0].at;
    p[0].wt = 0;
    for (int i=1;i<n;i++)
    {
        p[i].ct = p[i-1].ct + p[i].bt;
        if (p[i].at > p[i-1].ct)
        {
            p[i].ct += p[i].at-p[i-1].ct;
        }
        p[i].tat = p[i].ct-p[i].at;
        p[i].wt = p[i].tat-p[i].bt;
    }
    float avg_tat = 0.0;
    float avg_wt = 0.0;
    printf("\nID\tAT\tBT\tCT\tTAT\tWT\n");
    for(int i=0;i<n;i++)
    {
```

```

    avg_tat += p[i].tat;
    avg_wt += p[i].wt;
    printf("%d\t%d\t%d\t%d\t%d\t%d\n", p[i].id,p[i].at,p[i].bt,p[i].ct,p[i].tat,p[i].wt);
}
printf("\nAverage turn-around time = %.2f\n", (avg_tat/n));
printf("Average waiting time = %.2f", (avg_wt/n));
}
void print_gantt_chart(process p[], int n)
{
    int i, j;
    printf(" ");
    for(i=0; i<n; i++)
    {
        for(j=0; j<p[i].bt; j++) printf("--");
        printf(" ");
    }
    printf("\n| ");
    for(i=0; i<n; i++)
    {
        for(j=0; j<p[i].bt - 1; j++) printf(" ");
        printf("P%d", p[i].id);
        for(j=0; j<p[i].bt - 1; j++) printf(" ");
        printf("| ");
    }
    printf("\n ");
    for(i=0; i<n; i++)
    {
        for(j=0; j<p[i].bt; j++) printf("--");
        printf(" ");
    }
    printf("\n");
    printf("0");
    for(i=0; i<n; i++)
    {
        for(j=0; j<p[i].bt; j++) printf(" ");
        if(p[i].tat > p[i].ct) printf("\b");
        printf("%d", p[i].ct);
    }
    printf("\n");
}

int main()
{
    int n;
    printf("Enter the total number of processes: ");
    scanf("%d", &n);
    process queue[n];

    printf("Enter the arrival time and burst time of the processes:\n");
    for (int i=0;i<n;i++)
    {
        queue[i].id = i+1;
        printf("For process %d: ",i+1);
        scanf(" %d %d",&queue[i].at,&queue[i].bt);
        queue[i].ct = queue[i].tat = queue[i].wt = 0;
    }
    sorting(n,queue);
    fcfs(n,queue);
    puts("");
    puts("    GANTT CHART    ");
    puts("    *****    ");
    print_gantt_chart(queue, n);
    return 0;
}

```

## OUTPUT:-

```
ubuntu@Ubuntu:~/os/lab5/21MIS1095$ gcc fcf_21MIS1095.c
gcc: error: fcf_21MIS1095.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.
ubuntu@Ubuntu:~/os/lab5/21MIS1095$ gcc fcfs_21MIS1095.c
ubuntu@Ubuntu:~/os/lab5/21MIS1095$ ./a.out
Enter the total number of processes: 5
Enter the arrival time and burst time of the processes:
For process 1: 3 4
For process 2: 5 3
For process 3: 0 2
For process 4: 5 1
For process 5: 4 3

ID      AT      BT      CT      TAT      WT
3        0        2        2        2        0
1        3        4        7        4        0
5        4        3       10        6        3
2        5        3       13        8        5
4        5        1       14        9        8

Average turn-around time = 5.80
Average waiting time = 3.20
      GANTT CHART
      *****
      -----
      | P3 |  P1  |  P5  |  P2  |P4|
      -----
      0   2       7       10      13  14
ubuntu@Ubuntu:~/os/lab5/21MIS1095$ gcc sjf_21MIS1095.c
ubuntu@Ubuntu:~/os/lab5/21MIS1095$ ./a.out
```

## SJF SCHEDULING ALGORITHM

### CODE :-

```
#include <stdio.h>
#include <limits.h>

typedef struct Process {
    int id,at,bt,ct,tat,wt,status;
} process;

void sorting(int n, process p[]) {
    process temp;
    for (int i=0;i<n;i++) {
        for (int j=0; j<n-1;j++) {
            if (p[j].at>p[j+1].at) {
                temp = p[j];
                p[j] = p[j+1];
                p[j+1] = temp;
            }
        }
    }
}

int overhead=0;

void sjf(int n, int total_bt, process p[]) {
    int exec, min_bt;
    float avg_wt=0.0, avg_tat=0.0;
    for (int time = p[0].at; time<total_bt+overhead;) {
        int min_bt = INT_MAX;
        int exec = -1;
        for (int i=0;i<n;i++) {
            if (p[i].at <= time && p[i].status != 1 && p[i].bt < min_bt) {
                exec = i;
                min_bt=p[i].bt;
            }
        }
        if (exec == -1) {
            time++;
            overhead++;
            continue;
        }
    }
}
```

```

        time += p[exec].bt;
        p[exec].ct = time;
        p[exec].tat = p[exec].ct - p[exec].at ;
        p[exec].wt = p[exec].tat - p[exec].bt;
        p[exec].status = 1;
        avg_tat += p[exec].tat;
        avg_wt += p[exec].wt;
    }

    printf("\nID\tAT\tBT\tCT\tTAT\tWT\n");
    for(int i=0;i<n;i++) {
        printf("%d\t%d\t%d\t%d\t%d\t%d\n", p[i].id,p[i].at,p[i].bt,p[i].ct,p[i].tat,p[i].wt);
    }
    printf("\nAverage turn-around time = %.2f\n", (avg_tat/n));
    printf("Average waiting time = %.2f", (avg_wt/n));
}

int main() {
    int n, total_bt=0;
    printf("Enter the total number of processes: ");
    scanf(" %d", &n);
    process queue[n];

    printf("Enter the arrival time and burst time of the processes:\n");
    for (int i=0;i<n;i++) {
        queue[i].id = i+1;
        queue[i].status = 0;
        printf("For process %d: ",i+1);
        scanf(" %d %d",&queue[i].at,&queue[i].bt);
        total_bt += queue[i].bt;
        queue[i].ct = queue[i].tat = queue[i].wt = 0;
    }

    sorting(n,queue); // sorting the processes on the basis of arrival time
    sjf(n,total_bt,queue);
    return 0;
}

```

## Output:

```
| P3 | P1 | P3 | P2 | P4 |
|-----|
0  2  7  10  13  14
ubuntu@Ubuntu:~/os/lab5/21MIS1095$ gcc sjf_21MIS1095.c
ubuntu@Ubuntu:~/os/lab5/21MIS1095$ ./a.out
Enter the total number of processes: 5
Enter the arrival time and burst time of the processes:
For process 1: 3 1
For process 2: 1 4
For process 3: 4 2
For process 4: 0 6
For process 5: 2 3

ID    AT    BT    CT    TAT    WT
4      0     6     6     6     0
2      1     4    16    15    11
5      2     3    12    10     7
1      3     1     7     4     3
3      4     2     9     5     3

Average turn-around time = 8.00
Average waiting time = 4.80ubuntu@Ubuntu:~/os/lab5/21MIS1095$ gcc priority_21MIS10
ubuntu@Ubuntu:~/os/lab5/21MIS1095$ ./a.out
Enter the total number of processes: 5
Enter the burst time and priority of 5 processes:
```

## PRIORITY SCHEDULING ALGORITHM

### CODE:-

```
#include <stdio.h>
typedef struct Process
{
    int id,bt,ct,tat,wt,prio;
} process;

void sort(process p[],int n)
{
    for (int i=0;i<n;i++)
        for (int j=i+1;j<n;j++)
            if (p[i].prio>p[j].prio)
            {
                process temp=p[i];
                p[i]=p[j];
                p[j]=temp;
            }
}

void pbs(int n, process p[])
{
    p[0].ct = p[0].bt;
    p[0].tat = p[0].ct;
    p[0].wt = 0;
    for (int i=1;i<n;i++)
    {
        p[i].ct = p[i-1].ct + p[i].bt;
        p[i].tat = p[i].ct;
        p[i].wt = p[i].tat-p[i].bt;
    }
    float avg_tat = 0.0;
    float avg_wt = 0.0;
    printf("\nID\tBT\tCT\tTAT\tWT\n");
    for(int i=0;i<n;i++)
    {
        avg_tat += p[i].tat;
        avg_wt += p[i].wt;
        printf("%d\t%d\t%d\t%d\t%d\n", p[i].id,p[i].bt,p[i].ct,p[i].tat,p[i].wt);
    }
    printf("\nAverage turn-around time = %.2f\n", (avg_tat/n));
    printf("Average waiting time = %.2f", (avg_wt/n));
}

void print_gantt_chart(process p[], int n)
```

```

{
    printf( "\n\n");
    int i, j;
    printf(" ");
    for(i=0; i<n; i++)
    {
        for(j=0; j<p[i].bt; j++) printf("--");
        printf(" ");
    }
    printf("\n|");
    for(i=0; i<n; i++)
    {
        for(j=0; j<p[i].bt - 1; j++) printf(" ");
        printf("P%d", p[i].id);
        for(j=0; j<p[i].bt - 1; j++) printf(" ");
        printf("|");
    }
    printf("\n ");
    for(i=0; i<n; i++) {
        for(j=0; j<p[i].bt; j++) printf("--");
        printf(" ");
    }
    printf("\n");
    printf("0");
    for(i=0; i<n; i++)
    {
        for(j=0; j<p[i].bt; j++) printf(" ");
        if(p[i].tat > 9) printf("\b");
        printf("%d", p[i].tat);
    }
    printf("\n");
}

int main()
{
    int n;
    printf("Enter the total number of processes: ");
    scanf(" %d", &n);
    process queue[n];
    printf("Enter the burst time and priority of %d processes:\n",n);
    for (int i=0;i<n;i++)
    {
        queue[i].id = i+1;
        scanf(" %d %d",&queue[i].bt,&queue[i].prio);
        queue[i].ct = queue[i].tat = queue[i].wt = 0;
    }
    sort(queue,n);
    pbs(n,queue);
    print_gantt_chart(queue, n);
    return 0;
}

```



}

**Output:**

```
Average turn-around time = 13.40
Average waiting time = 4.80ubuntu@Ubuntu:~/os/lab5/21MIS1095$ gcc priority_21MIS1095.c
ubuntu@Ubuntu:~/os/lab5/21MIS1095$ ./a.out
Enter the total number of processes: 5
Enter the burst time and priority of 5 processes:
8 4
4 1
6 3
2 2
5 5

ID      BT      CT      TAT      WT
2       4       4       4       0
4       2       6       6       4
3       6      12      12       6
1       8      20      20      12
5       5      25      25      20

Average turn-around time = 13.40
Average waiting time = 8.40

-----
|  P2  | P4  |   P3   |   P1   |   P5   |
-----
0      4   6      12      20      25
ubuntu@Ubuntu:~/os/lab5/21MIS1095$ gcc roundrobin_21MIS1095.c
```

## ROUND ROBIN SCHEDULING ALGORITHM

### CODE:

```
#include<stdio.h>
int at=0,bt[100],rt[100],temp[100];
float wait_time=0,turn_time=0;

void main()
{
    int c,j,n,time,r,flag=0,time_q,ltr,i,wt=0;
    printf("Enter no.of process:");
    scanf("%d",&n);
    r=n;

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

        printf("Enter burst time of p%d: \t",c+1);
        scanf("%d",&bt[c]);
        rt[c]=bt[c];
        temp[c]=bt[c];
        printf("\n");
    }
    printf("Enter time quantum:\t");
    scanf("%d",&time_q);
    printf("\n\n\tprocess\tAT\tTAT\tWT\torder\n\n");

    for(time=0,c=0; r!=0;)
    {
        if(rt[c]<=time_q && rt[c]>0)
        {
            time=time+rt[c];
            rt[c]=0;
            flag=1;
        }
        else if (rt[c]>0)
        {
            rt[c]=rt[c]-time_q;
            time=time+time_q;
        }
    }
```

```

if(rt[c]==0 && flag==1)
{
    wt=0;
    wt = time-at-bt[c];
    r--;
    printf("\tP%d\t%d\t%d\t%d\t%d\n",c+1,at,time-at,wt,c+1);
    ltt=time-at;
    wait_time=wait_time+time-at-bt[c];
    turn_time=turn_time+time-at;
    flag=0;
}
if( c == n-1)
    c=0;
else if(at<=time)
    c++;
else
    c=0;
}
j=0;
printf("\n\n\n");
printf("Gantt Chart ");
printf("\n\n\n");
printf("\t");
for (int i=at;i<=time;i++){
    printf("--");

}
    printf("--");
printf("\n");
printf("\t");
for(i=at; i<time;)
{
    if(bt[j]>=time_q)
    {
        printf("P%d  |\t",j+1);
        i+=time_q;
        bt[j]=bt[j]-time_q;
    }
    else if(bt[j]>0)
    {
        printf("P%d |\t",j+1);
        i+=bt[j];
        bt[j]=0;
    }
    j++;
}

```

```

    if(j>=n)
    {
        j=0;
    }
}
printf("\n");

j=0;
printf("\t");
for (int i=at;i<=time;i++){
    printf("--");
}
    printf("--");
    printf("\n");
    printf("\t ");
for(i=at; i<time;)
{

    if(temp[j]>=time_q)
    {

        printf(" ");
        printf(" %d\t",i+time_q);
        i+=time_q;
        temp[j]=temp[j]-time_q;

    }

    else if(temp[j]>0)
    {
        printf(" ");
        printf("%d\t",i+temp[j]);
        i+=temp[j];
        temp[j]=0;

    }
    j++;
    if(j>=n)
    {
        j=0;

    }
}
printf("\n\n\n");
printf("\nAverage_waiting_time=%.2f\n",wait_time/n);
printf("Average_turn_around_time=%.2f\n",turn_time/n);

```

```
    printf("\n\n");  
}
```

## Output:

```
0      4      6      12      20      25  
ubuntu@Ubuntu:~/os/lab5/21MIS1095$ gcc roundrobin_21MIS1095.c  
ubuntu@Ubuntu:~/os/lab5/21MIS1095$ ./a.out  
Enter no.of process:4  
Enter burst time of p1:      21  
  
Enter burst time of p2:      3  
  
Enter burst time of p3:      6  
  
Enter burst time of p4:      2  
  
Enter time quantum:      5  
  
process AT      TAT      WT      order  
P2      0      8      5      2  
P4      0      15     13      4  
P3      0      21     15      3  
P1      0      32     11      1  
  
Gantt Chart  
  
-----  
P1 | P2 | P3 | P4 | P1 | P3 | P1 | P1 | P1 |  
-----  
5 8 13 15 20 21 26 31 32  
  
Average_waiting_time=11.00  
Average_turn_around_time=19.00  
  
ubuntu@Ubuntu:~/os/lab5/21MIS1095$
```