

OPERATING SYSTEMS

Lab-6: FCFS AND SJF

Name – Vinay Santosh Menon
Registration Number - 20BAI1103

TASK:

To write a C code for demonstrating FCFS and SJF with
and without Arrival Time.

FCFS Without Arrival Time:

Code:

```
#include <stdio.h>
int waitingtime(int proc_id[], int n,
int burst_time[], int wait_time[])
{
    wait_time[0] = 0;
    for (int i = 1; i < n ; i++ )
        wait_time[i] = burst_time[i-1] + wait_time[i-1] ;
    return 0;
}
int turnaroundtime( int proc_id[], int n,
int burst_time[], int wait_time[], int tat[])
{
    int i;
    for ( i = 0; i < n ; i++)
        tat[i] = burst_time[i] + wait_time[i];
    return 0;
}
int avgtime( int proc_id[], int n, int burst_time[])
{
    int wait_time[100], tat[100], total_wt = 0, total_tat = 0;
    int i;
    waitingtime(proc_id, n, burst_time, wait_time);
    turnaroundtime(proc_id, n, burst_time, wait_time, tat);
    printf("Processes| Burst_time| Waiting_time| Turn_around \n");
    for ( i=0; i<n; i++)
    {
        total_wt = total_wt + wait_time[i];
        total_tat = total_tat + tat[i];
        printf(" %d\t    %d\t\t %d    \t    %d\n", i+1, burst_time[i],
wait_time[i], tat[i]);
    }
    printf("Average waiting time = %f\n", (float)total_wt / (float)n);
    printf("Average turn around time = %f\n", (float)total_tat / (float)n);
    return 0;
}
int main()
{
    int proc_id[100];
    int burst_time[100];
    int n;
    printf("no of processes : ");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        printf("Process id: ");
```

```

        scanf("%d",&proc_id[i]);
        printf("burst time: ");
        scanf("%d",&burst_time[i]);
    }
    avgtime(proc_id, n, burst_time);
    return 0;
}

```

Example:

```

PS C:\Users\vinay\OneDrive\Documents\code\OS> cd "c:\Users\vinay\OneDrive\Documents\code\OS\" ; if ($?) { gcc fcfs_witho
.\fcfs_withoutArrivalTime }
no of processes : 5
Process id: 3
burst time: 12
Process id: 4
burst time: 10
Process id: 2
burst time: 5
Process id: 1
burst time: 34
Process id: 5
burst time: 6
Processes| Burst_time| Waiting_time| Turn_around
1         12         0         12
2         10        12         22
3          5        22         27
4         34        27         61
5          6        61         67
Average waiting time = 24.400000
Average turn around time = 37.800000

```

FCFS with Arrival Time:

Code:

```

#include <stdio.h>

void swap(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

void sortArrivalTime(int ArrTime[], int BurstTime[], int PID[], int n)
{
    for (int i = 0; i < n; i++)
        for (int j = i + 1; j < n; j++)
        {
            if (ArrTime[j] < ArrTime[i])
            {

```

```

        swap(&ArrTime[i], &ArrTime[j]);
        swap(&BurstTime[i], &BurstTime[j]);
        swap(&PID[i], &PID[j]);
    }
    else if (ArrTime[j] == ArrTime[i] && PID[j] < PID[i])
    {
        swap(&ArrTime[i], &ArrTime[j]);
        swap(&BurstTime[i], &BurstTime[j]);
        swap(&PID[i], &PID[j]);
    }
}

void findWaitingTime(int processes[], int n, int bt[], int at[], int wt[])
{
    wt[0] = at[0];
    for (int i = 1; i < n; i++)
    {
        wt[i] = (at[i - 1] + bt[i - 1] + wt[i - 1]) - at[i];
        wt[i] = (wt[i] > 0) ? wt[i] : 0;
    }
}

void findTurnAroundTime(int processes[], int n, int bt[], int at[], int wt[],
int tat[])
{
    for (int i = 0; i < n; i++)
        tat[i] = bt[i] + wt[i];
}

void printAvgTime(int processes[], int n, int bt[], int at[], int wt[], int
tat[], int total_wt, int total_tat)
{
    printf("PID\tArrival Time\tBurst Time\tWaiting Time\tTurn Around Time\n");
    for (int i = 0; i < n; i++)
    {
        total_wt = total_wt + wt[i];
        total_tat = total_tat + tat[i];
        printf("%d\t\t", processes[i]);
        printf("%d\t\t", at[i]);
        printf("%d\t\t", bt[i]);
        printf("%d\t\t", wt[i]);
        printf("%d\n", tat[i]);
    }
    float s = (float)total_wt / (float)n;
    float t = (float)total_tat / (float)n;
    printf("Average waiting time = %f", s);
    printf("\n");
    printf("Average turn around time = %f", t);
}

void findavgTime(int processes[], int n, int bt[], int at[])

```

```

{
    int wt[50], tat[50], total_wt = 0, total_tat = 0;
    findWaitingTime(processes, n, bt, at, wt);
    findTurnAroundTime(processes, n, bt, at, wt, tat);

    printAvgTime(processes, n, bt, at, wt, tat, total_wt, total_tat);
}
int main()
{
    int processes[] = {1, 2, 3, 4, 5};
    int n = sizeof processes / sizeof processes[0];
    int arrival_time[] = {3, 5, 0, 5, 4};
    int burst_time[] = {4, 3, 2, 1, 3};
    printf("Enter number of processes: ");
    scanf("%d", &n);
    printf("PID ArrTime BurstTime\n");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", i + 1);
        processes[i] = i + 1;
        scanf("%d %d", &arrival_time[i], &burst_time[i]);
    }
    sortArrivalTime(arrival_time, burst_time, processes, n);
    findavgTime(processes, n, burst_time, arrival_time);
    return 0;
}

```

Output:

```

PS C:\Users\vinay\OneDrive\Documents\code\OS> cd "c:\Users\vinay\OneDrive\Documents\code\
Time }
Enter number of processes: 5
PID ArrTime BurstTime
1 3 4
2 5 3
3 0 2
4 5 1
5 4 3
PID      Arrival Time      Burst Time      Waiting Time      Turn Around Time
3         0         2         0         2
1         3         4         0         4
5         4         3         3         6
2         5         3         5         8
4         5         1         8         9
Average waiting time = 3.200000
Average turn around time = 5.800000
PS C:\Users\vinay\OneDrive\Documents\code\OS>

```

SJF without Arrival Time:

Code:

```
#include <stdio.h>
int main()
{
    int Burst_time[100][4];
    int i, j, n, total = 0, index, temp;
    float avg_wt, avg_tat;
    printf("Enter number of process: ");
    scanf("%d", &n);
    printf("Enter Burst Time:\n");
    for (i = 0; i < n; i++)
    {
        printf("P%d: ", i + 1);
        scanf("%d", &Burst_time[i][1]);
        Burst_time[i][0] = i + 1;
    }
    for (i = 0; i < n; i++)
    {
        index = i;
        for (j = i + 1; j < n; j++)
            if (Burst_time[j][1] < Burst_time[index][1])
                index = j;
        temp = Burst_time[i][1];
        Burst_time[i][1] = Burst_time[index][1];
        Burst_time[index][1] = temp;

        temp = Burst_time[i][0];
        Burst_time[i][0] = Burst_time[index][0];
        Burst_time[index][0] = temp;
    }
    Burst_time[0][2] = 0;
    for (i = 1; i < n; i++)
    {
        Burst_time[i][2] = 0;
        for (j = 0; j < i; j++)
            Burst_time[i][2] += Burst_time[j][1];
        total += Burst_time[i][2];
    }
    avg_wt = (float)total / n;
    total = 0;
    printf("Processes| Burst_time| Waiting_time| Turn_around \n");
    for (i = 0; i < n; i++)
    {
        Burst_time[i][3] = Burst_time[i][1] + Burst_time[i][2];
        total += Burst_time[i][3];
        printf(" %d\t\t %d\t\t\t %d\t\t\t %d\n", Burst_time[i][0],
```

```

        Burst_time[i][1], Burst_time[i][2], Burst_time[i][3]);
    }
    avg_tat = (float)total / n;
    printf("Average Waiting Time= %f", avg_wt);
    printf("\nAverage Turnaround Time= %f", avg_tat);
}

```

Output:

```

PS C:\Users\vinay\OneDrive\Documents\code\OS> cd "c:\Users\vinay\OneDrive\Documents\code\OS\" ; if ($?) { g
outArrival }
Enter number of process: 5
Enter Burst Time:
P1: 23
P2: 2
P3: 14
P4: 22
P5: 24
Processes| Burst_time| Waiting_time| Turn_around
2         2         0         2
3         14        2        16
4         22        16        38
1         23        38        61
5         24        61        85
Average Waiting Time= 23.400000
Average Turnaround Time= 40.400002
PS C:\Users\vinay\OneDrive\Documents\code\OS>

```

SJF with Arrival Time:

Code:

```

#include<stdio.h>
int main()
{
    int time,bt[10],at[10],sum_bt=0,smallest,rem,n,i,rt[10];
    int sumt=0,sumw=0;
    printf("no of processes : ");
    scanf("%d",&n);rem=n;
    for(i=0;i<n;i++)
    {
        printf("arrival time for process P%d : ",i+1);
        scanf("%d",&at[i]);
        printf("burst time for process P%d : ",i+1);
        scanf("%d",&bt[i]);rt[i]=bt[i];
    }
}

```

```

    rt[9]=bt[9]=9999;
    printf("P_id\t|Arrival_time    |Burst_time    |Waiting_time    |Turn_Around
_Time\n");
    for(time=0;rem!=0;time++)
    {
        smallest=9;
        for(i=0;i<n;i++)
        {
            if(at[i]<=time && rt[i]>0 && rt[i]<rt[smallest])
                smallest=i;
        }
        rt[smallest]--;
        if(rt[smallest]==0)
        {
            rem--;
            printf("P[%d]\t|\t%d\t|\t%d\t|\t%d\t|\t%d\t\n",smallest+1,at[smallest],bt[smallest],time+1-bt[smallest]-at[smallest],time+1-at[smallest]);
            sumt+=time+1-bt[smallest]-at[smallest];
            sumw+=time+1-at[smallest];
        }
    }
    printf("\naverage waiting time = %f",sumw*1.0/n);
    printf("\naverage turnaround time = %f",sumt*1.0/n);
    return 0;
}

```

Output:

```

PS C:\Users\vinay\OneDrive\Documents\code\OS> cd "c:\Users\vinay\OneDrive\Documents\code\OS\" ; if ($?) { g
no of processes : 5
arrival time for process P1 : 3
burst time for process P1 : 1
arrival time for process P2 : 1
burst time for process P2 : 4
arrival time for process P3 : 4
burst time for process P3 : 2
arrival time for process P4 : 0
burst time for process P4 : 6
arrival time for process P5 : 2
burst time for process P5 : 3
P_id |Arrival_time |Burst_time |Waiting_time |Turn_Around_Time
P[1] |3 |1 |0 |1
P[2] |1 |4 |1 |5
P[3] |4 |2 |2 |4
P[5] |2 |3 |6 |9
P[4] |0 |6 |10 |16

average waiting time = 7.000000
average turnaround time = 3.800000
PS C:\Users\vinay\OneDrive\Documents\code\OS>

```