

Ex. No: 5

Date: 25-09-2023

SIMULATION OF CPU SCHEDULING ALGORITHMS

Aim:

To implement the process scheduling using First Come, First Served (FCFS), Round Robin (RR), Shortest Job First (SJB) and Priority Scheduling Algorithms.

(i) First Come First Serve (FCFS):

Program:

```
#include <stdio.h>
int main(){
    int n, bt[20], wt[20], tat[20], avwt = 0, avtat = 0, i, j;
    printf("Enter total number of processes(maximum 20):");
    scanf("%d", &n);
    printf("\nEnter Process Burst Time\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] += bt[j];
    }
    printf("\nProcess\t\tBurst Time\t\tWaiting Time\t\tTurnaround Time");
    for (i = 0; i < n; i++){
        tat[i] = bt[i] + wt[i]; // Turnaround Time = Burst time-
        waiting Time
        avwt += wt[i];
        avtat += tat[i];
        printf("\nP[%d]\t\t%d\t\t%d\t\t%d", i + 1, bt[i], wt[i],
        tat[i]);
    }
    avwt /= i;
    avtat /= i;
    printf("\n\nAverage Waiting Time:%d", avwt);
    printf("\nAverage Turnaround Time:%d", avtat);
    return 0;
}
```

Output:

```
viswa@desktop:~/2022242001$ ./FCFS
Enter total number of processes(maximum 20):4

Enter Process Burst Time
P[1]:10
P[2]:8
P[3]:4
P[4]:5

Process          Burst Time      Waiting Time    Turnaround Time
P[1]              10              0              10
P[2]              8              10             18
P[3]              4              18             22
P[4]              5              22             27

Average Waiting Time:12
Average Turnaround Time:19
```

(ii) Round Robin:

Program:

```
#include <stdio.h>
int main(){
    int i, limit, total = 0, x, counter = 0, time_quantum;
    int wait_time = 0, turnaround_time = 0, arrival_time[10],
    burst_time[10], temp[10];
    float average_wait_time, average_turnaround_time;
    printf("\nEnter Total Number of Processes: ");
    scanf("%d", &limit);
    x = limit;
    for (i = 0; i < limit; i++){
        printf("\nEnter Details of Process[%d]", i + 1);
        printf("\nArrival Time:");
        scanf("%d", &arrival_time[i]);
        printf("\nBurst Time:");
        scanf("%d", &burst_time[i]);
        temp[i] = burst_time[i];
    }
    printf("\nEnter Time Quantum: ");
    scanf("%d", &time_quantum);
    printf("\nProcess ID\t\tBurst Time\t\tTurnaround Time\t\tWaiting Time: ");
    for (total = 0, i = 0; x != 0;){
        if (temp[i] <= time_quantum && temp[i] > 0){
            total = total + temp[i];
            temp[i] = 0;
            counter = 1;
        }
        else if (temp[i] > 0){
            temp[i] = temp[i] - time_quantum;
            total = total + time_quantum;
        }
        if (temp[i] == 0 && counter == 1){
            x--;
        }
    }
}
```

```

        printf("\n\nProcess[%d]\t\t%d\t\t %d\t\t\t %d", i +
1, burst_time[i], total - arrival_time[i], total -
arrival_time[i] - burst_time[i]);
        wait_time = wait_time + total - arrival_time[i] -
burst_time[i];
        turnaround_time = turnaround_time + total -
arrival_time[i];
        counter = 0;
    }
    if (i == limit - 1){
        i = 0;
    }
    else if (arrival_time[i + 1] <= total){
        i++;
    }
    else{
        i = 0;
    }
}
average_wait_time = wait_time * 1.0 / limit;
average_turnaround_time = turnaround_time * 1.0 / limit;
printf("\nAverage Waiting Time:\t%f", average_wait_time);
printf("\n\nAverage Turnaround Time:\t%f\n",
average_turnaround_time);
return 0;
}

```

Output:

```

viswa@desktop: ~/2022242001
Enter Total Number of Processes: 4
Enter Details of Process[1]
Arrival Time:0
Burst Time:10
Enter Details of Process[2]
Arrival Time:1
Burst Time:8
Enter Details of Process[3]
Arrival Time:1
Burst Time:4
Enter Details of Process[4]
Arrival Time:2
Burst Time:5
Enter Time Quantum: 3
Process ID      Burst Time      Turnaround Time      Waiting Time:
Process[3]      4              18                  14
Process[4]      5              19                  14
Process[2]      8              25                  17
Process[1]      10             27                  17
Average Waiting Time:  15.500000
Average Turnaround Time:  22.250000
viswa@desktop: ~/2022242001$ 

```

(iii) Shortest Job First (SJB) :

Program :

```
#include <stdio.h>
int main(){
    int A[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", &A[i][1]);
        A[i][0] = i + 1;
    }
    for (i = 0; i < n; i++){
        index = i;
        for (j = i + 1; j < n; j++)
            if (A[j][1] < A[index][1])
                index = j;
        temp = A[i][1];
        A[i][1] = A[index][1];
        A[index][1] = temp;
        temp = A[i][0];
        A[i][0] = A[index][0];
        A[index][0] = temp;
    }
    A[0][2] = 0;
    for (i = 1; i < n; i++){
        A[i][2] = 0;
        for (j = 0; j < i; j++)
            A[i][2] += A[j][1];
        total += A[i][2];
    }
    avg_wt = (float)total / n;
    total = 0;
    printf("P      BT      WT      TAT\n");
    for (i = 0; i < n; i++){
        A[i][3] = A[i][1] + A[i][2];
        total += A[i][3];
        printf("P%d      %d      %d      %d\n", A[i][0],
            A[i][1], A[i][2], A[i][3]);
    }
    avg_tat = (float)total / n;
    printf("Average Waiting Time= %f", avg_wt);
    printf("\nAverage Turnaround Time= %f", avg_tat);
}
```

Output:

```
viswa@desktop:~/2022242001$ gcc SJF.c -o SJF
viswa@desktop:~/2022242001$ ./SJF
Enter number of process: 4
Enter Burst Time:
P1: 10
P2: 8
P3: 4
P4: 5
P          BT          WT          TAT
P3          4           0           4
P4          5           4           9
P2          8           9          17
P1         10          17          27
Average Waiting Time= 7.500000
Average Turnaround Time= 14.250000
viswa@desktop:~/2022242001$
```

(iv) Priority Scheduling:

Program:

```
#include <stdio.h>
void swap(int *a, int *b){
    int temp = *a;
    *a = *b;
    *b = temp;
}
int main(){
    int n;
    printf("Enter Number of Processes: ");
    scanf("%d", &n);
    int b[n], p[n], index[n];
    float avg_wt = 0, avg_tat = 0;
    for (int i = 0; i < n; i++){
        printf("Enter Burst Time and Priority Value for Process %d: ", i + 1);
        scanf("%d %d", &b[i], &p[i]);
        index[i] = i + 1;
    }
    for (int i = 0; i < n; i++){
        int a = p[i], m = i;
        for (int j = i; j < n; j++){
            if (p[j] > a){
                a = p[j];
                m = j;
            }
        }
        swap(&p[i], &p[m]);
        swap(&b[i], &b[m]);
        swap(&index[i], &index[m]);
    }
    int t = 0;
    printf("Order of process Execution is\n");
    for (int i = 0; i < n; i++){
        printf("P%d is executed from %d to %d\n", index[i], t, t
+ b[i]);
        t += b[i];
    }
}
```

```

        printf("\n");
        printf("Process Id      Burst Time      Wait Time      TurnAround
Time\n");
        int wait_time = 0;
        for (int i = 0; i < n; i++){
            printf("P%d          %d          %d          %d\n",
index[i], b[i], wait_time, wait_time + b[i]);
            avg_wt += wait_time;
            wait_time += b[i];
            avg_tat += wait_time + b[i];
        }
        avg_wt /= n;
        avg_tat /= n;
        printf("\n\nAverage Waiting Time: % f\n", avg_wt);
        printf("\nAvergae Turnaround Time: % f \n", avg_tat);
        return 0;
    }
}

```

Output:

```

viswa@desktop: ~/2022242001
viswa@desktop:~/2022242001$ gcc priority.c -o priority
viswa@desktop:~/2022242001$ ./priority
Enter Number of Processes: 4
Enter Burst Time and Priority Value for Process 1: 10
3
Enter Burst Time and Priority Value for Process 2: 8
2
Enter Burst Time and Priority Value for Process 3: 4
4
Enter Burst Time and Priority Value for Process 4: 5
1
Order of process Execution is
P3 is executed from 0 to 4
P1 is executed from 4 to 14
P2 is executed from 14 to 22
P4 is executed from 22 to 27

Process Id      Burst Time      Wait Time      TurnAround Time
P3              4              0              4
P1              10             4             14
P2              8              14            22
P4              5              22            27

Average Waiting Time   : 10.000000
Avergae Turnaround Time : 23.500000
viswa@desktop:~/2022242001$ 

```

Result:

Thus, the process scheduling was implemented using FCFS, Round Robin, SJB and Priority Scheduling Algorithms.