**GANDAKI COLLEGE OF ENGINEERING AND SCIENCE**

LAB 2: IMPLEMENT SJF SCHEDULING ALGORITHM

**SUBMITTED BY: SUBMITTED TO: -**

**Shivaji Chalise PRATIKSHYA SHRESTHA**

**BE2019SE685 Teacher**

**OBJECTIVE:** TO IMPLEMENT SJF SCHEDULING ALGORITHM

**THEORY:** Shortest Job First (SJF) is an algorithm in which the process having the smallest execution time is chosen for the next execution. This scheduling method can be preemptive or non-preemptive. It significantly reduces the average waiting time for other processes awaiting execution.

**SOURCE CODE:**

SJF (Non-Preemptive)

#include<stdio.h>

int main()

{

int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;

float avg\_wt,avg\_tat;

printf("Enter number of process:");

scanf("%d",&n);

printf("\nEnter Burst Time:\n");

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

{

printf("p%d:",i+1);

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

p[i]=i+1;

}

//sorting of burst times

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

{

pos=i;

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

{

if(bt[j]<bt[pos])

pos=j;

}

temp=bt[i];

bt[i]=bt[pos];

bt[pos]=temp;

temp=p[i];

p[i]=p[pos];

p[pos]=temp;

}

wt[0]=0;

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

{

wt[i]=0;

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

wt[i]+=bt[j];

total+=wt[i];

}

avg\_wt=(float)total/n;

total=0;

printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time");

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

{

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

total+=tat[i];

printf("\np%d\t\t %d\t\t %d\t\t\t%d",p[i],bt[i],wt[i],tat[i]);

}

avg\_tat=(float)total/n;

printf("\n\nAverage Waiting Time=%f",avg\_wt);

printf("\nAverage Turnaround Time=%f\n",avg\_tat);

}

**Output:**

Enter number of process:3

Enter Burst Time:

p1:2

p2:1

p3:5

Process Burst Time Waiting Time Turnaround Time

p2 1 0 1

p1 2 1 3

p3 5 3 8

Average Waiting Time=1.333333

Average Turnaround Time=4.00000

SJF (Premptive)

#include <iostream>

using namespace std;

void findWaitingTime(int processes[], int n, int bt[], int wt[]) {

wt[0] = 0;

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

wt[i] = bt[i - 1] + wt[i - 1];

}

void findTurnAroundTime(int processes[], int n, int bt[], int wt[], int tat[]) {

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

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

}

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

int wt[n], tat[n], total\_wt = 0, total\_tat = 0;

findWaitingTime(processes, n, bt, wt);

findTurnAroundTime(processes, n, bt, wt, tat);

cout << "Processes "

<< " Burst time "

<< " Waiting time "

<< " Turn around time\n";

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

total\_wt = total\_wt + wt[i];

total\_tat = total\_tat + tat[i];

cout << " " << i + 1 << "\t\t" << bt[i] << "\t " << wt[i] << "\t\t "

<< tat[i] << endl;

}

cout << "Average waiting time = " << (float)total\_wt / (float)n;

cout << "\nAverage turn around time = " << (float)total\_tat / (float)n;

}

int main() {

int processes[] = {1, 2, 3};

int n = sizeof processes / sizeof processes[0];

int burst\_time[] = {10, 5, 8};

findavgTime(processes, n, burst\_time);

return 0;

}

**Output:**

Enter the Total Number of Processes: 3

Enter Details of 3 Processes:

Enter Arrival Time: 1

Enter Burst Time: 2

Enter Arrival Time: 3

Enter Burst Time: 1

Enter Arrival Time: 2

Enter Burst Time: 2

Average Waiting Time: 0.666667

Average Turnaround Time: 2.333333

**CONCLUSION:** Finally we understood and learned to implement SJF Scheduling Algorithm.