1. Design and implement a Schedular using a non-preemptive Shortest Job First algorithm.

Shortest Job First (Non Preemptive) - algorithm

1. BEGIN

2. Get number of processes (numberOfProcesses) as input from user.

3. P[i]=array of processes.

4. while(input){

5. get burst time for each process

6. }

7. sort\_increasing\_order(P) //according to burst time

8. Calculate Waiting time , average waiting time and average turn around time for each process in

P.

9. Display Gantt Chart.

10. END

/\*

\* SJF.cpp

\*

\* Created on: 29-Aug-2012

\* Author: Lokesh Chandra Basu

\* Enrol. no: 10114026

\* Branch: CSE III yr

\*

\*/

**#include**<iostream>

**using** **namespace** std;

**struct** process{

**int** pId ;

**int** burstTime ;

};

**int** **main**(){

**int** numberOfProcesses;

//number of process

cout << "\nNUMBER OF PROCESS: ";

cin >> numberOfProcesses;

**struct** process p[numberOfProcesses];

**struct** process temp;

**int** i;

**int** j;

**int** waitingTime = 0;

**float** totalWaitingTime = 0;

//burst time

**for**(i=0; i<numberOfProcesses; i++){

p[i].pId = i+1 ;

cout << "================================\n";

cout << "PROCESS " << i+1 << "\n";

cout << "\tBURST TIME: ";

cin >> p[i].burstTime;

totalWaitingTime += p[i].burstTime;

}

//sort

**for**(i=0; i<numberOfProcesses; i++){

**for**(j=0; j<numberOfProcesses; j++){

**if**(p[j].burstTime>p[i].burstTime){

temp = p[i];

p[i] = p[j] ;

p[j] = temp;

}

}

}

//display

cout << "\n========================================================\n";

cout << " PROCESS\t BURST TIME\t WAITING TIME\n\n";

**for**(i=0;i<numberOfProcesses;i++){

cout <<"\tP" << p[i].pId << "\t\t" << p[i].burstTime;

cout << "\t\t" << waitingTime << "\n";

waitingTime = waitingTime + p[i].burstTime ;

}

cout << "========================================================\n";

//average waiting time

cout << "AVERAGE WAITING TIME = ";

cout << totalWaitingTime/numberOfProcesses << "\n\n" ;

**return** 0;

}

**OUTPUT:**

NUMBER OF PROCESS: 4

================================

PROCESS 1

BURST TIME: 2

================================

PROCESS 2

BURST TIME: 1

================================

PROCESS 3

BURST TIME: 5

================================

PROCESS 4

BURST TIME: 3

========================================================

PROCESS BURST TIME WAITING TIME

P2 1 0

P1 2 1

P4 3 3

P3 5 6

========================================================

AVERAGE WAITING TIME = 2.75