

**Department of Information Technology**

**Report on Course based project**

**Title of the project:**

Designing an applet implementing selected algorithms

**Batch Number: 02**

**Student roll numbers:**

17071A12G2-P.SAI SRUJANA

17071A12G4-REETHU

17071A12G5-SAI KUMAR

17071A12G6-S.DEEPIKA

17071A12G7-SRAVYA

**Under the guidance of : V.LAVANYA MAM**

**Designation: Professor in VNRVJIET**

**ABSTRACT:**

This project is a combination of applets and scheduling algorithms. An applet is a Java program that runs in a Web browser. An applet can be a fully functional Java application because it has the entire Java API at its disposal. A Process Scheduler schedules different processes to be assigned to the CPU based on particular scheduling algorithms. By following the below mentioned steps we completed this project.

1. Code for FCFS algorithm
2. Code for SJF algorithm
3. Code for Round Robin
4. Code for Priority
5. Code an applet with buttons implementing waiting time, average time, turnaround time, average turnaround time of selected algorithms.

These algorithms are either non-preemptive or preemptive. Non-preemptive algorithms are designed so that once a process enters the running state, it cannot be preempted until it completes its allotted time, whereas the preemptive scheduling is based on priority where a scheduler may preempt a low priority running process anytime when a high priority process enters into a ready state.

**CREATION OF BUTTON:**

Button b = new Button (“Click me!”);

**BUTTONS AND GUI COMPONENTS:**

* An applet can contain various graphical user interface components that allow a user to interact with the applet.
  + - Some components are for user input
    - Others display information
    - Some components can do both.

**EVENT CLASSES:**

The classes that represent events are at the core of Java’s event handling mechanism. The event handling must begin with the event classes.

**BUTTONS:**

ACTIONEVENT CLASS: This class is defined in java.awt.event package. The ActionEvent is generated when button is clicked or the item of a list is double clicked.

ACTIONLISTENER CLASS: It is notified whenever you click on the button or menu item. It is notified against ActionEvent.

ACTIONPERFORMED CLASS: It is a class that is responsible in handling all action events such as when the user clicks on a component.

GETACTIONCOMMAND () : The action command identifies the button. When using two or more buttons within the same application, the action command gives you an easy way to determine which button was pressed.

**PROBLEM DEFINITION:**

An applet is a Java program that can be embedded into a web page. It runs inside the web browser and works at client side. An applet is embedded in an HTML page using the APPLET or OBJECT tag and hosted on a web server.

Using these applets our aim is to design an interface with buttons implementing the waiting time, average time, turnaround time, average turnaround time of selected algorithms. This program take input from the user and schedules the process according to the user requirement.

The selection of algorithm can be done using the applet feature – buttons. The algorithms used are

* FCFS
* SJF
* Round Robin
* Priority

**DESIGN/MODULE DESCRIPTION**:

Each algorithm is stored in particular folder. In the command prompt the part for this folder will be specified and applet program is executed.

This program consists of codes and various algorithms. After setting the path the applet part is compiled and executed.

**CODE:**

**FCFS**

import java.util.\*;

class Fcfs

{

public static void runfcfs()

{

int atime[]=new int[20];

int btime[]=new int[20];

int wtime[]=new int[20];

int ctime[]=new int[20];

int ttime[]=new int[20];

int totalwt=0,totaltt=0;

float avgwt=0,avgtt=0;

Scanner sn = new Scanner(System.in);

System.out.print("\nEnter the number of processes : ");

int n = sn.nextInt();

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

System.out.println();

System.out.print("Enter the arrival time of process "+(i+1)+" : ");

atime[i]=sn.nextInt();

System.out.print("Enter the burst time of process "+(i+1)+" : ");

btime[i]=sn.nextInt();

}

ctime[0]=btime[0];

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

{

ctime[i]=ctime[i-1]+btime[i];

}

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

ttime[i]=ctime[i]-atime[i];

wtime[i]=ttime[i]-btime[i];

totalwt=totalwt+wtime[i];

totaltt=totaltt+ttime[i];

}

avgwt=(float)totalwt/n;

avgtt=(float)totaltt/n;

System.out.println("\nArrival\_time\tBurst\_time\tCompletion\_time\tTurnaround\_time\tWait\_time");

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

{ System.out.println(atime[i]+"\t\t"+btime[i]+"\t\t"+ctime[i]+"\t\t"+ttime[i]+"\t\t"+wtime[i]);

}

System.out.println("\nTotal turnaround time: "+totaltt+"ms \nAverage turnaround time: "+avgtt+"ms");

System.out.println("\nTotal wait time: "+totalwt+"ms \nAverage wait time: "+avgwt+"ms");

}

}

**SJF**

import java.util.Scanner;

class Sjf {

public static void runsjf()

{

int process[] = new int[10];

int ptime[] = new int[10];

int wtime[] = new int[10];

int ttime[] = new int[10];

int temp, n, totalwt=0,totaltt=0;

float avgwt=0,avgtt=0;

Scanner get = new Scanner(System.in);

System.out.println("Enter Number of Processes:");

n = get.nextInt();

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

{

System.out.println("Enter Process "+(i+1)+" ID: ");

process[i] = get.nextInt();

System.out.println("Enter Process "+(i+1)+" Burst Time: ");

ptime[i] = get.nextInt();

}

for(int i=0;i<n-1;i++)

{

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

{

if(ptime[i]>ptime[j])

{

temp = ptime[i];

ptime[i] = ptime[j];

ptime[j] = temp;

temp = process[i];

process[i] = process[j];

process[j] = temp;

}

}

}

wtime[0] = 0;

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

{

wtime[i] = wtime[i-1]+ptime[i-1];

totalwt = totalwt + wtime[i];

ttime[i] = wtime[i] + ptime[i];

totaltt = totaltt + ttime[i];

}

avgwt = (float)totalwt/n;

avgtt = (float)totaltt/n;

System.out.println("P\_ID P\_TIME T\_TIME W\_TIME");

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

{

System.out.println(process[i]+"\t"+ptime[i]+"\t"+ttime[i]+"\t"+wtime[i]);

}

System.out.println("Total Waiting Time: "+totalwt+"ms");

System.out.println("Average Waiting Time: "+avgwt+"ms");

System.out.println("Total Waiting Time: "+totaltt+"ms");

System.out.println("Average Waiting Time: "+avgtt+"ms");

}

}

**Round Robin**

import java.util.\*;

class Roundrobin{

public static void runroundrobin(){

Scanner sn = new Scanner(System.in);

int i,j,k,q,sum=0;

System.out.println("Enter number of process:");

int n=sn.nextInt();

int bt[]=new int[n];

int wt[]=new int[n];

int tat[]=new int[n];

int a[]=new int[n];

System.out.println("Enter brust Time:");

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

System.out.println("Enter brust Time for "+(i+1));

bt[i]=sn.nextInt();

}

System.out.println("Enter Time quantum:");

q=sn.nextInt();

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

a[i]=bt[i];

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

wt[i]=0;

do{

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

if(bt[i]>q){

bt[i]-=q;

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

if((j!=i)&&(bt[j]!=0))

wt[j]+=q;

}

}

else{

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

if((j!=i)&&(bt[j]!=0))

wt[j]+=bt[i];

}

bt[i]=0;

}

}

sum=0;

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

sum=sum+bt[k];

}

while(sum!=0);

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

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

System.out.println("process\t\tBT\tWT\tTAT");

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

System.out.println("process"+(i+1)+"\t"+a[i]+"\t"+wt[i]+"\t"+tat[i]);

}

float avg\_wt=0;

float avg\_tat=0;

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

avg\_wt+=wt[j];

}

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

avg\_tat+=tat[j];

}

System.out.println(" Total waiting time: "+(avg\_wt)+"ms\n total turn around time: "+(avg\_tat)+"ms");

System.out.println("average waiting time: "+(avg\_wt/n)+"ms\n Average turn around time: "+(avg\_tat/n)+"ms");

}

}

**Priority**

import java.util.\*;

class Priority

{

public static void runpriority()

{

int p[]=new int[20];

int btime[]=new int[20];

int wtime[]=new int[20];

int ctime[]=new int[20];

int ttime[]=new int[20];

int totalwt=0,totaltt=0,temp;

float avgwt=0,avgtt=0;

Scanner sn = new Scanner(System.in);

System.out.print("\nEnter the number of processes : ");

int n = sn.nextInt();

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

System.out.println();

System.out.print("Enter the priority of process "+(i+1)+" : ");

p[i]=sn.nextInt();

System.out.print("Enter the burst time of process "+(i+1)+" : ");

btime[i]=sn.nextInt();

}

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

{

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

{

if(p[i]>p[j])

{

temp=p[i];

p[i]=p[j];

p[j]=temp;

temp=btime[i];

btime[i]=btime[j];

btime[j]=temp;

}

}

}

ctime[0]=btime[0];

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

{

ctime[i]=ctime[i-1]+btime[i];

}

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

ttime[i]=ctime[i];

wtime[i]=ttime[i]-btime[i];

totalwt=totalwt+wtime[i];

totaltt=totaltt+ttime[i];

}

avgwt=(float)totalwt/n;

avgtt=(float)totaltt/n;

System.out.println("\nPriority\tBurst\_time\tCompletion\_time\tTurnaround\_time\tWait\_time");

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

{ System.out.println(p[i]+"\t\t"+btime[i]+"\t\t"+ctime[i]+"\t\t"+ttime[i]+"\t\t"+wtime[i]);

}

System.out.println("\nTotal turnaround time: "+totaltt+"\nAverage wait time: "+avgtt+"ms");

System.out.println("\nTotal wait time: "+totalwt+"\nAverage wait time: "+avgwt+"ms");

}

}

**Scheduler Applet**

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

/\*

<applet code=SchedulerApplet height=500 width=400>

</applet>

\*/

public class SchedulerApplet extends Applet implements ActionListener {

public static final String FCFS = "FCFS";

public static final String SJF = "SJF";

public static final String ROUND\_ROBIN = "ROUND ROBIN";

public static final String PRIORITY = "PRIORITY";

String msg = " ";

Button b1, b2, b3;

public void init() {

Button b1 = new Button(FCFS);

Button b2 = new Button(SJF);

Button b3 = new Button(ROUND\_ROBIN);

Button b4 = new Button(PRIORITY);

add(b1);

add(b2);

add(b3);

add(b4);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

}

public void actionPerformed(ActionEvent ae) {

System.out.println(ae.getActionCommand());

switch (ae.getActionCommand()) {

case FCFS:

Fcfs.runfcfs();

break;

case SJF:

Sjf.runsjf();

break;

case ROUND\_ROBIN:

Roundrobin.runroundrobin();

break;

case PRIORITY:

Priority.runpriority();

break;

}

repaint();

}

public void paint(Graphics g) {

g.drawString(msg, 20, 100);

}

}

**RESULT:**

As soon as one executes the program, the applet displays buttons consisting of FCFS, SJF, ROUND ROBIN AND PRIORITY. The user has to click on one of the buttons. As per the selection the program is further executed. Let us suppose that user has selected FCFS.

Output:

Enter the number of processes: 4

Enter the arrival time of process 1: 0

Enter the arrival time of process 2: 1

Enter the arrival time of process 3: 2

Enter the arrival time of process 4: 3

Enter the burst time of process 1: 8

Enter the burst time of process 2: 4

Enter the burst time of process 3: 9

Enter the burst time of process 4: 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Arrival time | Burst time | Completion time | Turnaround time | Waiting time |
| 0 | 8 | 8 | 8 | 0 |
| 1 | 4 | 12 | 11 | 7 |
| 2 | 9 | 21 | 19 | 10 |
| 3 | 5 | 26 | 23 | 18 |

Total turnaround time: 61ms

Average turnaround time: 15.25ms

Total wait time: 35ms

Average wait time: 8.75ms