Name:Akshay S Gadhave

Roll No:221076

SY Comp A3

Assignment 4

AIM: Write a program to show the demonstration of scheduling algorithms. 1)FCFS 2)SJF 3)Priority scheduling

Theory:

Process scheduling in OS is done with the help of scheduler. It schedules several processes in system in different states like ready,running,blocked.
The primary goal behind scheduling the processes is to maximize the throughput and Increase the CPU utilization.

Below Are different terminologies used with process scheduling:

Arrival Time: Time at which the process arrives in the ready queue.
Completion Time: Time at which process completes its execution.
Burst Time: Time required by a process for CPU execution.
Turn Around Time: Time Difference between completion time and arrival time.
Turn Around Time = Completion Time – Arrival Time
Waiting Time: Time Difference between turn around time and burst time.
Waiting Time = Turn Around Time – Burst Time

Different scheduling Algorithms:

1)First come First Serve(FCFS): Simplest scheduling algorithm that schedules according to arrival times of processes. First come first serve scheduling algorithm states that the process that requests the CPU first is allocated the CPU first. It is implemented by using the FIFO queue. When a process enters the ready queue, its PCB is linked onto the tail of the queue. When the CPU is free, it is allocated to the process at the head of the queue. The running process is then removed from the queue. FCFS is a non-preemptive scheduling algorithm.

2)**Shortest Job First(SJF):**Process which have the shortest burst time are scheduled first.If two processes have the same bust time then FCFS is used to break the tie. It is a non-preemptive scheduling algorithm.

3)Priority Based Scheduling: In this scheduling, processes are scheduled according to their priorities, i.e., highest priority process is scheduled first. If priorities of two processes match, then schedule according to arrival time. Here starvation of process is possible.

Other than these there several scheduling algorithms which are used in Operating System according to the need.

CODE:

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
	int n,temp,*arr,*bt,*stbt,*ct,*tat,*find,*p;
	int i=0,j=0;//index varibles for loops
	int k,*wt;
	float avwt=0,avtat=0;//average wt and tat

	int choice;

	do
	{
		printf("1.FCFS 2.SJF 3.exit:Enter your choice:");
		scanf("%d",&choice);

		switch(choice)
		{
			case 1:
```

```
avwt=0,avtat=0;

printf("Enter the total no of processes to schedule:");
scanf("%d",&n);

//Dynamic memory allocation for arrays

arr=(int *)malloc(n*sizeof(int));
bt=(int *)malloc(n*sizeof(int));
stbt=(int *)malloc(n*sizeof(int));
ct=(int *)malloc(n*sizeof(int));
tat=(int *)malloc(n*sizeof(int));
find=(int *)malloc(n*sizeof(int));
wt=(int *)malloc(n*sizeof(int));

printf("\nEnter the arrival time and burst time for the processes:\n");
for(i=0;i<n;++i)
{
        printf("P[%d]:\n",i+1);
        scanf("%d",&arr[i]);
        scanf("%d",&bt[i]);
}

for(i=0;i<n;++i)
{
        find[i]=arr[i];
}
//sorting the arrival time array for FCFS algo
```

```
for(i=0;i<n;++i)
{
	j=i;
	while(j>0 && arr[j-1]>arr[j])
	{
		temp=arr[j];
		arr[j]=arr[j-1];
		arr[j-1]=temp;
		j--;
	}
}

//making changes in sorted burst time array as per arrival times
for(i=0;i<n;++i)
{
	for(j=0;j<n;++j)
	{
		if(find[i]==arr[j])
		{
			stbt[j]=bt[i];
			break;
		}
	}
}



ct[0]=stbt[0];
```

```
    //calculating completion time
    for(i=1;i<n;i++)
    {
      ct[i]=ct[i-1]+stbt[i];
    }
    printf("\nProcess\t\tBurst Time\tWaiting Time\tTurnaround Time\n");

    //calculating tat
    for(i=0;i<n;++i)
    {
            tat[i]=ct[i]-arr[i];
    }

    //calculaing waiting time
    for(i=0;i<n;++i)
            wt[i]=tat[i]-stbt[i];

    for(i=0;i<n;++i)
    {
            avtat+=tat[i];
            avwt+=wt[i];
            printf("\nP[%d]\t\t%d\t\t%d\t\t%d",i+1,bt[i],wt[i],tat[i]);
    }

    printf("\n\n");

avwt/=n;
    avtat/=n;
```

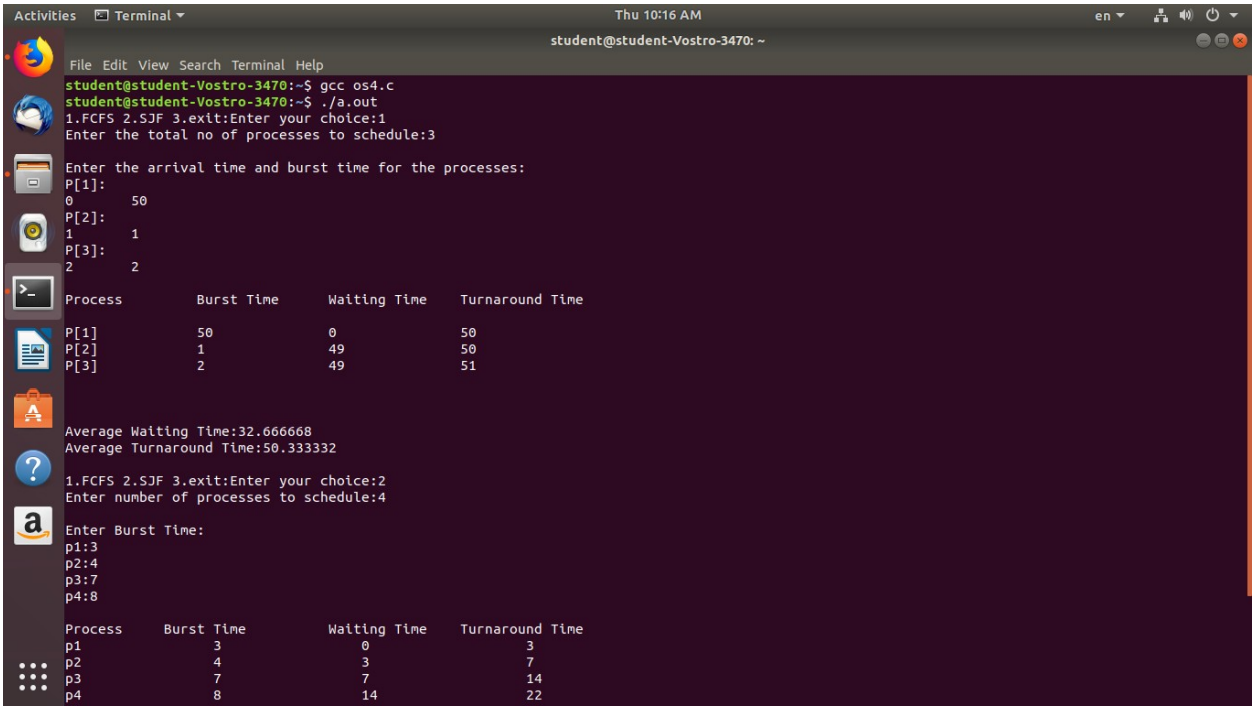
```
printf("\n\nAverage Waiting Time:%f",avwt);
printf("\nAverage Turnaround Time:%f\n\n",avtat);

//free the memory aloccated using malloc
free(arr);
free(bt);
free(stbt);
free(wt);
free(find);
free(tat);

break;

//FCFS algorithm ends here



case 2:
        avwt=0,avtat=0;

        printf("Enter number of processes to schedule:");
        scanf("%d",&n);

        bt=(int *)malloc(n*sizeof(int));
        stbt=(int *)malloc(n*sizeof(int));
        ct=(int *)malloc(n*sizeof(int));
        tat=(int *)malloc(n*sizeof(int));
        wt=(int *)malloc(n*sizeof(int));
```

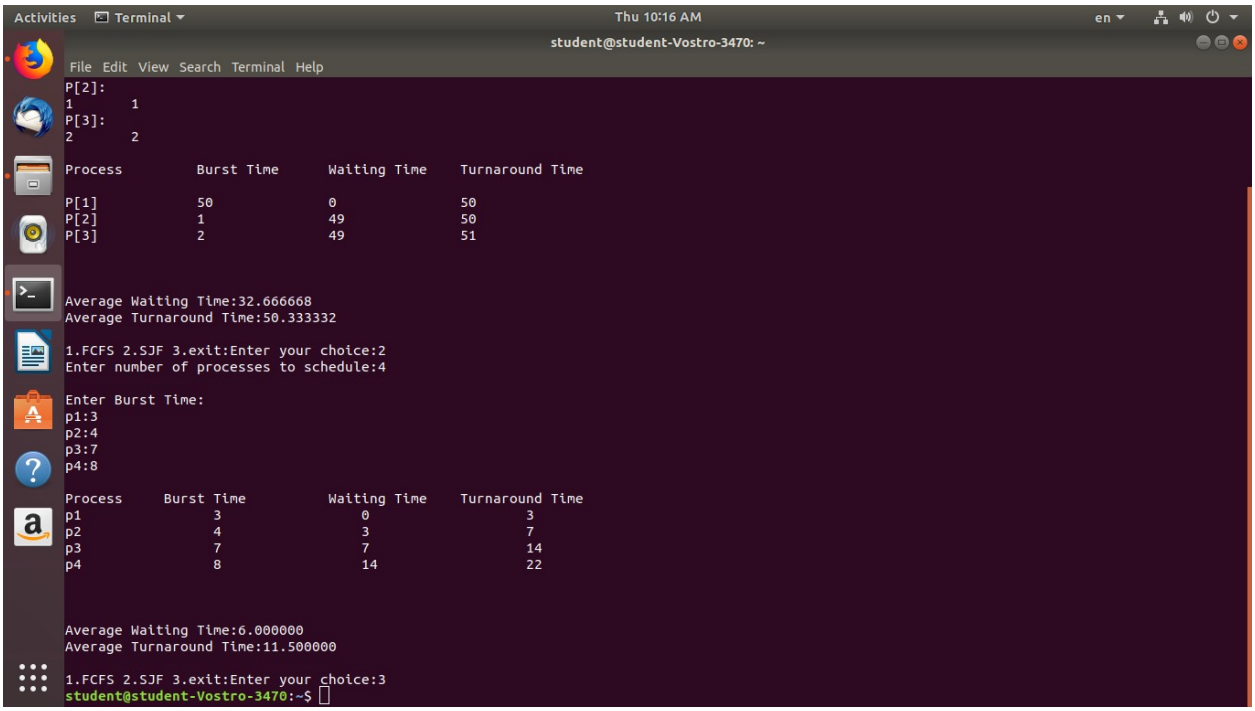
```
p=(int *)malloc(n*sizeof(int));

printf("\nEnter Burst Time:\n");
for(i=0;i<n;i++)
{
printf("p%d:",i+1);
scanf("%d",&bt[i]);
p[i]=i+1;          //contains process number
}

//sorting the burst time for SJF
for(i=0;i<n;++i)
{
        j=i;
        while(j>0 && bt[j-1]>bt[j])
        {
                temp=bt[j];
                bt[j]=bt[j-1];
                bt[j-1]=temp;
                j--;
        }
}


wt[0]=0;          //waiting time for first process will be zero

//calculating waiting time
for(i=1;i<n;i++)
```

```
{
    wt[i]=0;
    for(j=0;j<i;j++)
    wt[i]+=bt[j];
        avwt+=wt[i];
}

for(i=0;i<n;++i)
{
    tat[i]=bt[i]+wt[i];
    avtat+=tat[i];
}

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


for(i=0;i<n;++i)
{
     printf("\np%d\t\t  %d\t\t    %d\t\t\t%d",p[i],bt[i],wt[i],tat[i]);
}

printf("\n\n");

avwt/=n;
    avtat/=n;
    printf("\n\nAverage Waiting Time:%f",avwt);
    printf("\nAverage Turnaround Time:%f\n\n",avtat);
```

```
                    //free the memory aloccated using malloc
                    free(arr);
                    free(bt);
                    free(bt);
                    free(wt);
                    free(p);
                    free(tat);

            //SJF algorithm ends here
                    break;

                case 3:
                    return 0;

                default:
                    printf("\nPlz enter a valid choice\n\n");
                    break;

            }


        }while(choice!=3);


        return 0;
}
```

Output Screenshots:




Activities
Terminal
Thu 10:16 AM
en
student@student-Vostro-3470: ~
File Edit View Search Terminal Help
P[2]:
1 1
P[3]:
2 2
Process Burst Time Waiting Time Turnaround Time
P[1] 50 0 50
P[2] 1 49 50
P[3] 2 49 51
Average Waiting Time:32.666668
Average Turnaround Time:50.333332
1.FCFS 2.SJF 3.exit:Enter your choice:2
Enter number of processes to schedule:4
Enter Burst Time:
p1:3
p2:4
p3:7
p4:8
Process Burst Time Waiting Time Turnaround Time
p1 3 0 3
p2 4 3 7
p3 7 7 14
p4 8 14 22
Average Waiting Time:6.000000
Average Turnaround Time:11.500000
1.FCFS 2.SJF 3.exit:Enter your choice:3
student@student-Vostro-3470:~$