Vidyalankar Polytechnic	DEPARTMENT OF COMPUTER ENGINEERING

Subject: Operating System	Subject Code:22516
Semester:5th Semester	Course: Computer Engineering
Laboratory No:V118	Name of Subject Teacher: Natasha Brahme
Name of Student: Siddharth Shah	Roll Id: 22203A0041

Experiment No:	14
Title of Experiment	Implement scheduling alogrithms.

- **Program Code :**

1. **Consider the processes P1,P2,P3,P4 given in the below table, arrives for execution in the same order, with arrival time 0 and given burst time,find using the fcfs scheduling algorithm.**
   1.Turn around time for each process
   2.Waiting time for each process
   3.Average turn around time
   4.average waiting time

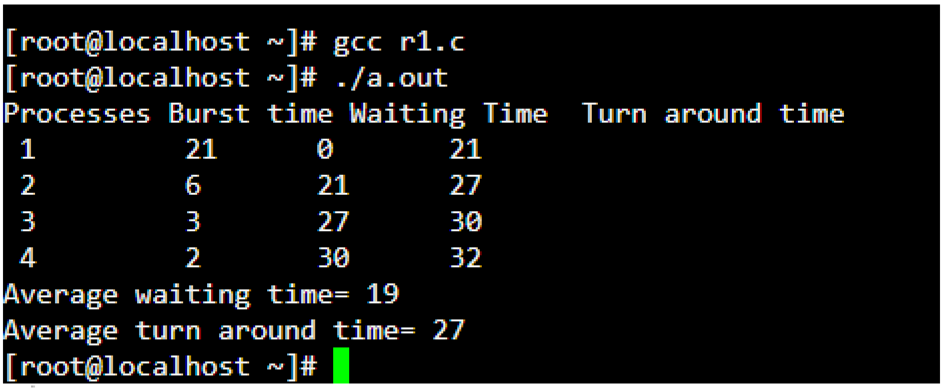
Process	Burst Time
P1	21
P2	6
P3	3
P4	2

Ans :

Code :

```
#include <stdio.h>
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);
printf("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];
printf(" %d",(i+1));
printf("\t   %d",bt[i]);
printf("\t   %d",wt[i]);
printf("\t   %d\n",tat[i]);
}
int s=(float)total_wt/(float)n;
int t=(float)total_tat/(float)n;
printf("Average waiting time= %d",s);
printf("\n");
printf("Average turn around time= %d\n",t);
}

int main(){
int processes[]={1,2,3,4};
int n=sizeof processes/sizeof processes[0];
int burst_time[]={21,6,3,2};
findavgTime(processes,n,burst_time);
return 0;



}
```

Output :



```
[root@localhost ~]# gcc r1.c
[root@localhost ~]# ./a.out
Processes Burst time Waiting Time  Turn around time
 1         21      0       21
 2         6       21      27
 3         3       27      30
 4         2       30      32
Average waiting time= 19
Average turn around time= 27
[root@localhost ~]#
```

- **Practical related questions :**

1. **Compare SJF,Priority and RR with respect to turnaround time and average waiting time.**

Ans :

Priority Scheduling algorithm	**RR Algorithm**	**SJF algorithm**
Generally minimizes the turnaround time because it processes the shortest jobs first.	Turnaround time can be higher in RR because every process gets a fair share of CPU time, but it might have to wait multiple cycles before getting enough time to complete.	Turnaround time depends heavily on the priority of the jobs.
SJF often results in the minimum average waiting time since shorter jobs do not have to wait behind longer jobs.	RR aims for fairness but typically has a higher average waiting time compared to SJF.	Can be efficient if high-priority tasks are important, but low-priority processes may experience longer waiting times.

2. **State the conditions for preemptive and non-preemptive scheduling algorithm.**

Ans :

Conditions for Pre-emptive scheduling algorithm :

1. Higher Priority Arrival: If a process with a higher priority than the currently running process arrives, it can preempt the current process.
2. Arrival of a Shorter Job: In algorithms like Shortest Remaining Time First (SRTF), a newly arrived process with a shorter remaining execution time than the current process can preempt the running process.
3. Time Quantum Expiration: In Round Robin (RR), the currently running process is preempted when its allocated time quantum expires, and the CPU is given to the next process in the queue.
4. System Calls or Interrupts: Certain interrupts or system calls (e.g., I/O operations) may cause the currently running process to be preempted.

Conditions for non-preemptive scheduling algorithm :

1. Completion of the Process: A running process will not be interrupted until it has completed its execution.
2. Voluntary Release: If a process voluntarily enters a waiting state (e.g., to perform I/O operations), the CPU is then given to another ready process.
3. Process Termination: The CPU is only available to other processes after the currently running process terminates.

3. **Give the reason of problems arises in FCFS.**

Ans :

1. **Long Waiting Time for Short Processes (Convoy Effect)**:

In FCFS, if a long process arrives before shorter ones, the shorter processes must wait until the long one finishes. This leads to high average waiting times, especially when there’s a mix of short and long processes.

2. **No Prioritization**:

FCFS does not consider the priority of tasks. All processes are treated equally, regardless of their urgency or importance. This means that critical tasks might have to wait behind less important ones, which is not always ideal for time-sensitive applications.

3. **Poor Response Time**:

   The response time in FCFS can be poor for processes that arrive after others, as they have to wait in line even if the earlier processes are long-running.

4. **Not Suitable for Time-Sharing Systems**:

   FCFS is not well-suited for time-sharing systems where the goal is to ensure that each process gets a fair share of the CPU. Since FCFS runs processes until completion before moving to the next one, it can lead to one process monopolizing the CPU.

4. Write a formula for turnaround Time

Ans :

Waiting time = Starting time – Arrival time

5. Write a formula for average waiting time.

Ans :

Turnaround time = Ending time – Arrival time

- **Exercise**

1. The jobs are scheduled for execution as follows :
   Solve the problem by using FCFS and pre-emptive SJF
   Find average waiting time using gantt chart

Process	Arrival time	Burst time
P1	0	10
P2	1	4
P3	2	14
P4	3	8

Draw gantt chart for above mention example.

Ans :

Gantt chart for FCFS scheduling algorithm :

P1	P2	P3	P4

Gantt chart for Pre-emptive SJF scheduling algorithm :

P1	P2	P4	P1	P3

FCFS scheduling algorithm :

Waiting time of P1 = 0

Waiting time of P2 = 9

Waiting time of P3 = 12

Waiting time of P4= 25

Average waiting time = (0+9+12+25)/4 = 11.5 msec.

SJF pre-emptive scheduling algorithm :

Waiting time of P1 = 11

Waiting time of P2 = 0

Waiting time of P3 = 20

Waiting time of P4= 2

Average waiting time = (11+0+20+2)/4 =8.25 msec.

2. Calculate average waiting time using RR algorithm for the following set of processes with the length of the CPU burst time given in millisecond. (Time quantum 20 ms)

Process	Burst Time
P1	12
P2	45
P3	78
P4	90

Ans :

Waiting time of P1 = 0 ms

Waiting time of P2 = 52 ms

Waiting time of P3 = 105 ms

Waiting time of P4 = 123 ms

Average waiting time = (0+52+105+123)/4 = 70 ms