LOVELY PROFESSIONAL UNIVERSITY

Academic Task-3 (Operating System) Compulsory Component

School of Computer Science and Engineering Faculty of Technology And Sciences
Name of the faculty member
Course Code: CSE 316 Course Title: Operating System
Max. Marks:30 Date of Submission: 04/04/2020

Instructions for Assignment Submission

Mention the below in header of the word document

Student Name: gopal sharma

Student ID 11802044 Roll No A05

Email Address: gopalsharma989796@gmail.com

GitHub Link: https://github.com/gopal2044/Os-project

Code: Set5

QUESTION

Ques 5 . CPU schedules N processes which arrive at different time intervals and each process is allocated the CPU for a specific user input time unit, processes are scheduled using a preemptive round robin scheduling algorithm. Each process must be assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to the processes one task has priority 0. The length of a time quantum is T units, where T is the custom time considered as time quantum for processing. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue. Design a scheduler so that the task with priority 0 does not starve for resources and gets the CPU at some time unit to execute. Also compute waiting time, turn around.

Description:

- It is simple, easy to implement, and starvation-free as all processes get fair share of CPU.
- One of the most commonly used technique in CPU scheduling as a core.
- It is preemptive as processes are assigned CPU only for a fixed slice of time at most.
- The disadvantage of it is more overhead of context switching.
- Priority scheduling is one of the most common scheduling algorithms in batch systems. Each process is assigned a priority. Process with the highest priority is to be executed first and so on.
  Processes with the same priority are executed on first come first served basis. Priority can be decided based on memory requirements, time requirements or any other resource requirement.
- `First input the processes with their burst time`

- and priority.
- 2- Sort the processes, burst time and priority
- according to the priority.
- 3- Now simply apply FCFS algorithm.
-

Algorithm:

Assign the process to ready queue.

Step 2: Assign the process to the CPU according to the priority, higher priority process will get the CPU first than lower priority process.

Step 3: If two processes have similar priority then FCFS is used to break the tie.

Step 4: Repeat the step 1 to 3 until ready queue is empty.

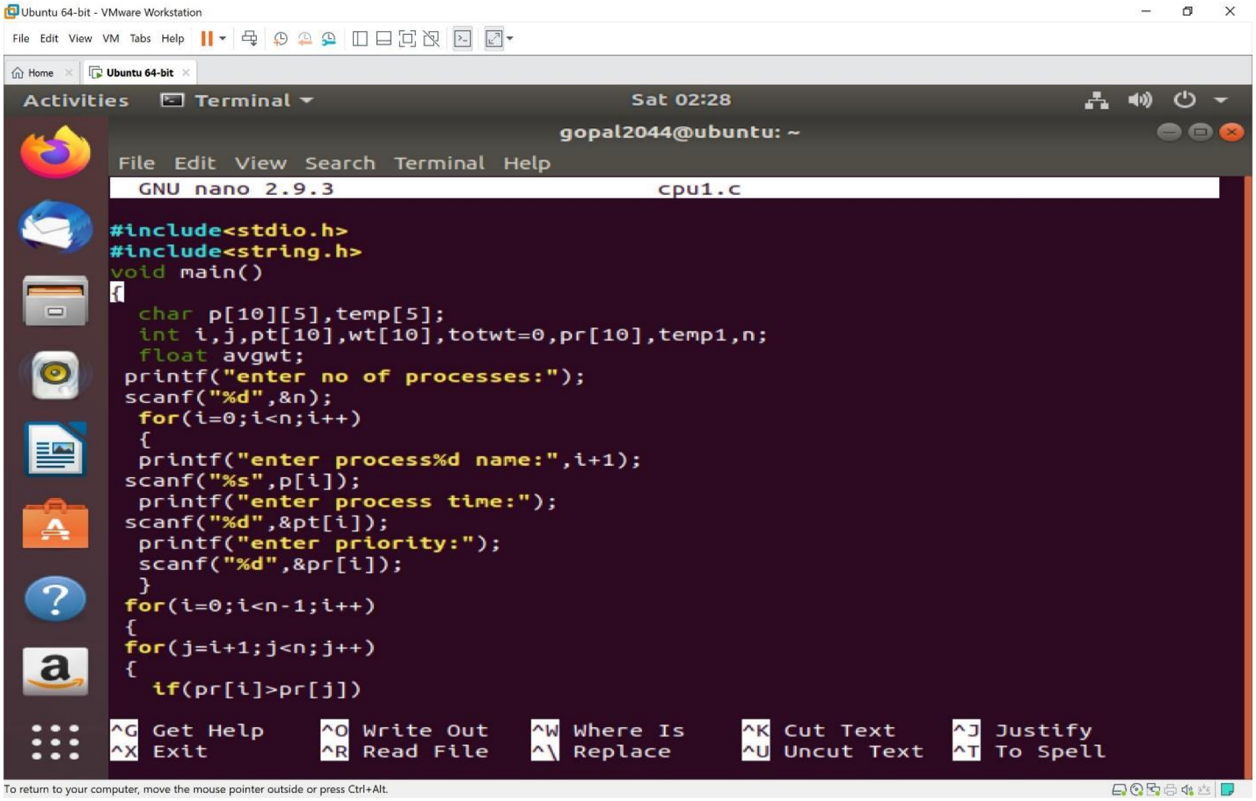
Step 5: Calculate Waiting time and Turnaround time of individual Process.

Step 6: Calculate Average waiting time and Average Turnaround time

Description (Code):

```
#include<stdio.h>
#include<conio.h>
void main()
{
  char p[10][5],temp[5];
  int i,j,pt[10],wt[10],totwt=0,pr[10],temp1,n;
  float avgwt;
 printf("enter no of processes:");
 scanf("%d",&n);
  for(i=0;i<n;i++)
  {
  printf("enter process%d name:",i+1);
 scanf("%s",p[i]);
  printf("enter process time:");
 scanf("%d",&pt[i]);
  printf("enter priority:");
  scanf("%d",&pr[i]);
  }
 for(i=0;i<n-1;i++)
 {
 for(j=i+1;j<n;j++)
 {
   if(pr[i]>pr[j])
 {
   temp1=pr[i];
 pr[i]=pr[j];
  pr[j]=temp1;
  temp1=pt[i];
  pt[i]=pt[j];
  pt[j]=temp1;
 strcpy(temp,p[i]);
 strcpy(p[i],p[j]);
  strcpy(p[j],temp);
  }
  }
  }
 wt[0]=0;
  for(i=1;i<n;i++)
  {
   wt[i]=wt[i-1]+wt[i-1];
   totwt=totwt+wt[i];
   }
avgwt=(float)totwt/n;
printf("p_name\t p_time\t priority\t w_time\n");
for(i=0;i<n;i++)
{
   printf(" %s\t %d\t %d\t %d\n" ,p[i],pt[i],pr[i],wt[i]);
   }
  printf("total waiting time=%d\n avg waiting time=%f",totwt,avgwt);
  getch;
}
```

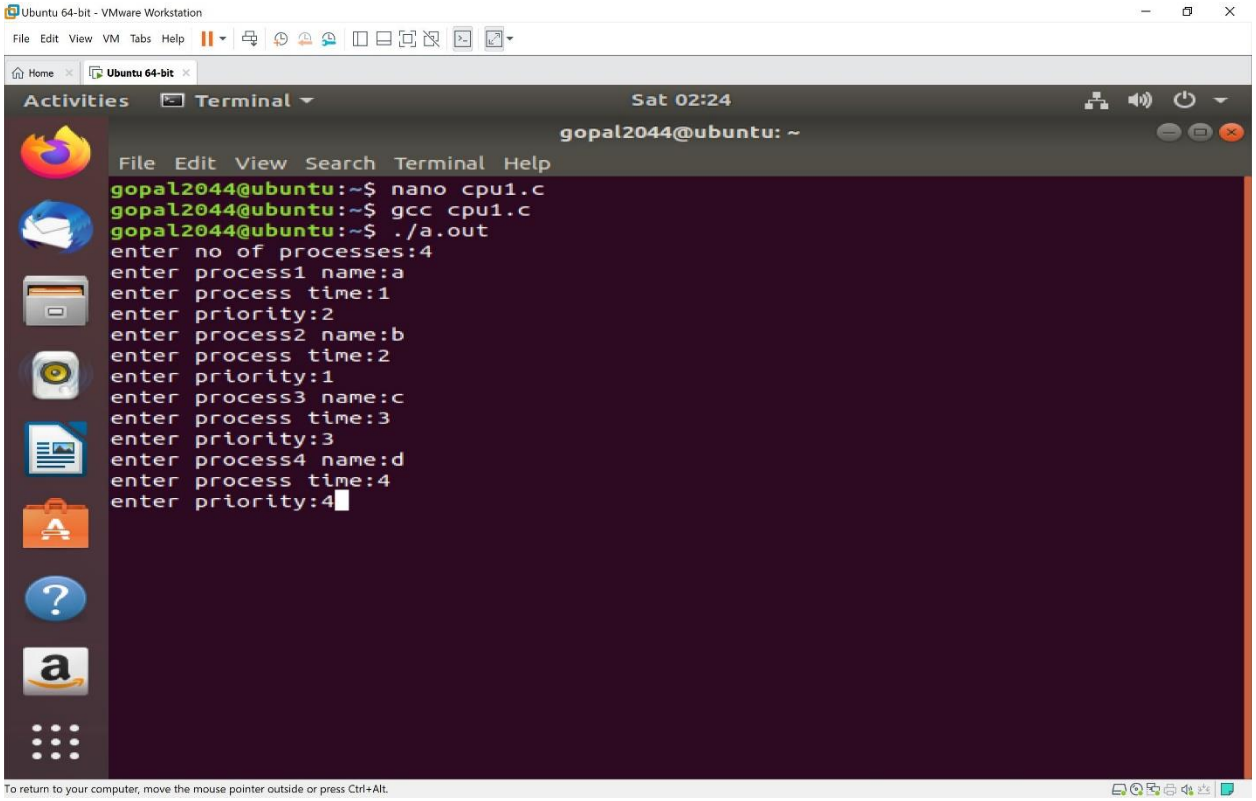
Code snippet


Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Activities
Terminal
Sat 02:28
gopal2044@ubuntu: ~
File Edit View Search Terminal Help
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

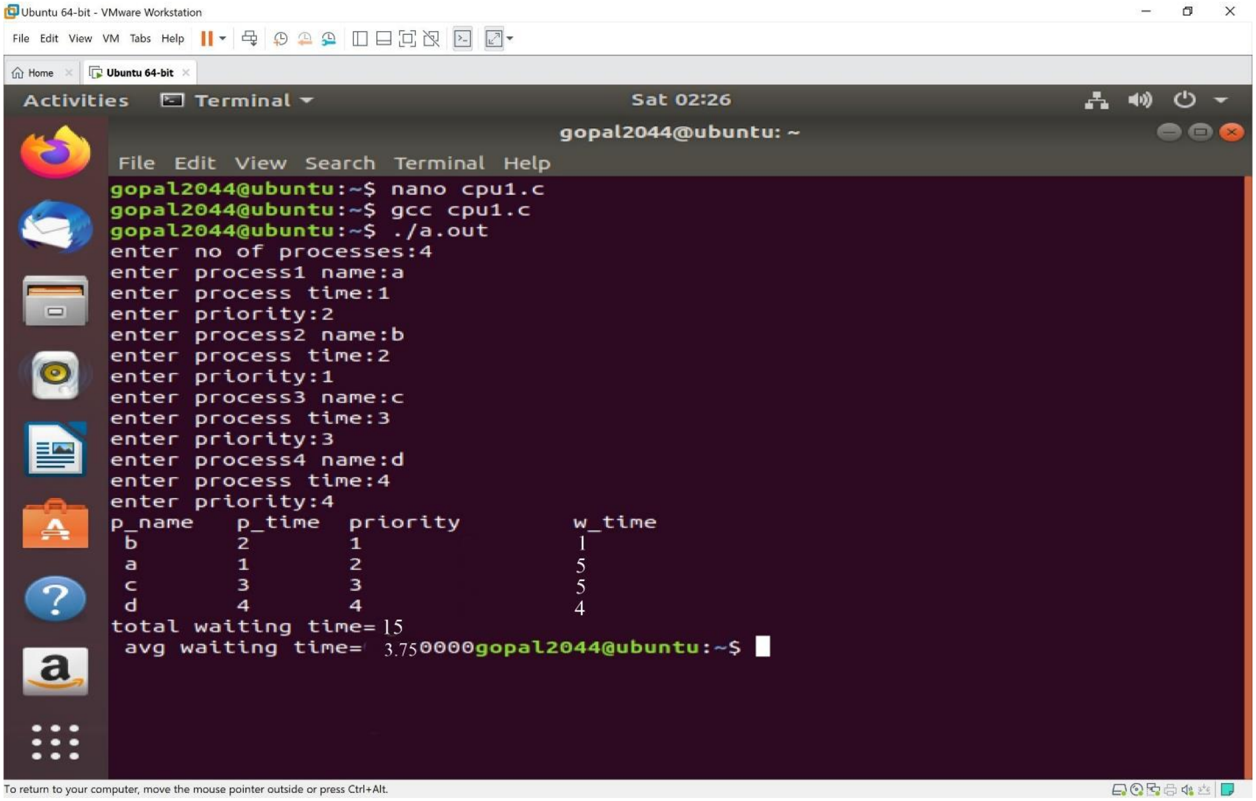
```
  GNU nano 2.9.3                      cpu1.c

#include<stdio.h>
#include<string.h>
void main()
{
  char p[10][5],temp[5];
  int i,j,pt[10],wt[10],totwt=0,pr[10],temp1,n;
  float avgwt;
 printf("enter no of processes:");
 scanf("%d",&n);
  for(i=0;i<n;i++)
  {
  printf("enter process%d name:",i+1);
 scanf("%s",p[i]);
  printf("enter process time:");
 scanf("%d",&pt[i]);
  printf("enter priority:");
  scanf("%d",&pr[i]);
  }
 for(i=0;i<n-1;i++)
 {
 for(j=i+1;j<n;j++)
 {
   if(pr[i]>pr[j])

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```


Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Activities
Terminal
Sat 02:24
gopal2044@ubuntu: ~
File Edit View Search Terminal Help
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

```
gopal2044@ubuntu:~$ nano cpu1.c
gopal2044@ubuntu:~$ gcc cpu1.c
gopal2044@ubuntu:~$ ./a.out
enter no of processes:4
enter process1 name:a
enter process time:1
enter priority:2
enter process2 name:b
enter process time:2
enter priority:1
enter process3 name:c
enter process time:3
enter priority:3
enter process4 name:d
enter process time:4
enter priority:4
```


Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Activities
Terminal
Sat 02:26
gopal2044@ubuntu: ~
File Edit View Search Terminal Help
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

```
gopal2044@ubuntu:~$ nano cpu1.c
gopal2044@ubuntu:~$ gcc cpu1.c
gopal2044@ubuntu:~$ ./a.out
enter no of processes:4
enter process1 name:a
enter process time:1
enter priority:2
enter process2 name:b
enter process time:2
enter priority:1
enter process3 name:c
enter process time:3
enter priority:3
enter process4 name:d
enter process time:4
enter priority:4
p_name   p_time   priority         w_time
 b        2        1                1
 a        1        2                5
 c        3        3                5
 d        4        4                4
total waiting time= 15
 avg waiting time=  3.750000gopal2044@ubuntu:~$ 
```

Description:

8.Have you made minimum 5 revisions of solution on GitHub?

GitHub Link:

- I have uploaded the project on the GitHub. And I have also done the revisions of the project on the GitHub as my project is going on.
- I have made more than 6 revisions on that project and also uploaded the documented description file with it.