Operating Systems: Fall 2016

BSCS V (A&B)

Semester Project-2 **Total Marks: 60**

Date Assigned: 29 Nov 2016 Due Date and Time: 11 Dec 2016 [11:00 pm]

Objective: Implement a 'Priority Based' Round Robin Scheduling Algorithm using C++ on Linux.

Processes will be scheduled using a round-robin scheduling algorithm with priorities. Each process has a CPU burst, then an I/O burst followed by another CPU burst before it terminates. Each process is assigned a numerical priority, with a higher number indicating a lower relative priority. In addition, the system also has an idle task (which consumes no CPU resources and is identified as Pidle). This task is scheduled whenever the system has no other available processes to run. The length of a time quantum will be provided by the user from the command line and can only be 4 or 7 units. Make sure other values are not accepted by your program. Your scheduler will work for exactly 5 processes.

Scheduling Rules:

- Every process first enters the ready queue. Highest priority process will always be at the queue head.
- The process at the head of the queue will run on the CPU for the time quantum set by the user.
- Every running process will join the ready queue when the quantum expires.
- Only after the expiry of the quantum we make a scheduling decision and the highest propriety process gets the CPU.
- No running process is preempted until its time quantum is expired even if a high propriety process enters the system during this time.
- Processes proceed for I/O only after CPU burst 1 is complete. Processes join the ready queue after I/O completion and terminate when CPU burst 2 is complete.
- For processes arriving at the same time with equal priorities, schedule them in the order they are listed in the input file.
- Use FCFS to break any other ties that you may encounter during scheduling.

Input data:

The input data will be read from a text file (provided to you) in csv format. The first line will have column names followed by the process data (separated by commas). Note that this is just a sample data and we will test your program for correctness with our own data. Your program should work for 5 processes. Make sure user gets appropriate messages if number of processes is out of range. The sample data in the input file is as follows:

Round Robin with Priority Scheduling					
P ID	ARIVAL TIME	CPU1	/IO/	CPU2	PRIORITY
P1	0	6	15	8	2
P2	3	4	10	11	4
P3	6	8	12	6	5
P4	8	15	9	11	1
P5	11	13	18	7	3

Sample Output:

Your program should generate the following output on the screen (in form of a table)

```
[prompt#] ./file 4
Time quantum set by user: 4 units
Input file read successfully.
Output table shown below:
PID     AT      FT      TAT     WT
P1      0
P2      3
P3      6
P4      8
P5      11
Average waiting time:
Average turnaround time:
CPU utilization rate (Percent):
```

Missing values are to be calculated and shown above in the output. Make sure these are formatted properly on the screen. Remember the text in blue font above is the output which must be displayed exactly in the given format. No extra stuff should be printed.

PID: Process ID (should be capital P1, P2 etc...)
AT: Arrival time
FT: Finished at time: The time when a process exits the system after completion.
WT: Waiting time   TAT: Turnaround time

```
[prompt#] ./file 5
Incorrect time quantum. Enter correct value.
```

Important Instructions:

Use any appropriate data structures for this project. The code must be properly commented and this carries marks too. Group details (Names, R Nos, Section etc) should appear on top of the code in comments. Also, mention the exact contribution of each group member to the project in these comments.
User must get appropriate messages on the screen if data values entered as input are out of range or incorrect. Do test your program thoroughly and make sure it gives correct output before submitting. If your program does not compile, you do not get credit. You will not be allowed to debug your program during the demo. We will not entertain excuses like "it was working before but we don't know what happened to it now!!!".

Demos and an extensive viva will be conducted to grade the project. Schedule will be given later. We will test your program with different test data. Do not hard code any values.

Caution: Write and test all code in Linux environment using C++ only. DO NOT WRITE CODE IN WINDOWS. Groups giving excuses that their code runs in windows and does not run in Linux because of any reason will get a zero. Test thoroughly before submitting. We will compile using g++.

Submission: Only the program .cpp file is to be emailed well before the deadline to avoid unforeseen delays. Email a copy to yourself to be sure you get it. Do not send multiple emails.

Your Group details are given in this document below, check your group number and rename your program file according to it before submitting.

Submission filename: GroupX.cpp (where X is your Group number)

Send emails to submit_work@yahoo.com.
The email subject should be "Group X: OS Project-2". Five marks will be deducted if the email subject/file names are not named exactly as specified.

Late projects get zero credit. Do not copy code from other sources. Severe penalties will be applied to groups who copy code/part of code from any other sources and submit it as their own. Before submitting, make sure your project is not plagiarized.

Post project related queries on course discussion forum.

Start early…

Grading Criteria:

1. File names, comments in code, mentioning specific contribution of group members, other submission criteria (file name, mail subject etc).
2. Whether program compiles successfully without a single error or warning. Errors mean zero credit.
3. Error checking related to number of processes, time quantum and other conditions. Proper messages must be displayed on the screen.
4. Correct processing of input, input file format to be exactly as described.
5. Correctness of all calculations and required information. Test your program thoroughly.
6. Only the required information is to be displayed on the screen as output. No extra stuff needed. See examples above. Do not produce or write unwanted/unnecessary output.
8. Your program will be tested with different data to ascertain correctness. If you hard code anything, you will lose maximum credit.

All group members must read the project document thoroughly to avoid surprises.

Good Luck!

OS Group Details

No:	Members	Section
1	Muhammad Haseeb Mohsin 140746 Nasir Iqbal 120635 Salman Mushahid 130829	A
2	Hamna Kaleem 140751 Hassaan Ghalib 140694 M Usman Ghani 140681 Muhammad Asim 140720	B
3	Zeeshan Malik 130926 Touseeq Razzaq 130911 Zohaib Altaf 140726	B
4	Hafiz Zain Ul Abidin 130876 Ammar Nadeem 140706 Abdul Zahoor 140729	B
5	Qasim Sheikh 140709 Eeman Rasheed 140742 Waleed Bashir 140778	A
6	Suniya Waqar 140743 Khushbakhat Afzaal 140722 Maham Zehra 140674	A
7	Syed Daniyal Shah 140734 Muhammad Muheet 140707 Maha Bhatti 140700	A
8	Rabbia Sana 140710 Saika Latif 140973 Saad Muhammad 140671	A
9	Owais Saeed 130915 Sheikh Shahzeb 130864 Abubakar Ikram 140765	B
10	Sania Ahmed 140750) Zainab Inaya 140733) Ali Jawad Naqvi 140735)	B

11	Rimmel Tariq Syed Basit Abbas Moiz Karim	120648 120616 130844	**B**
12	Muhammad Bilal Tamoor Tanvir Shahzaman Marwat	140666 130888 130852	**B**
13	Mohsin Ather Amna Pervaiz Butt Hammad Hussain	140669 140687 140672	**B**
14	Hamza Bin Riaz Adil Hussain Hassan Zaheer	140708 140762 140697	**B**
15	Muhammad Sohaib Shoaib Ismail Amna Bibi	140745 140760 140736	**A**
16	Muhammad Haris Abdullah Mazoor Nouman Amjad	140740 140782 140779	**A**
17	Raffay Rafique Rana Sohaib Hassan Hassan Naveed	140721 140787 140691	**B**
18	Alveena Khan Asma Hameed Hamza Javed	140675 140690 140696	**B**
19	Mian Salman Hassan Ahmad Hasham Muhammad Haseeb	140684 140718 140724	**B**
20	Muhammad Abubakkar Muhammad Faisal Siddique Maham Mushtaq	140685 140703 140682	**A**
21	Wasiullah Waqar Abdullah Kayani Usman Malik	140222 140717 140714	**B**
22	Ayesha Zaheer Ayesha Hanif Maham Ihsan	130867 130879 130808	**A**

23	Memoona Imran 120614 Sidra Bibi 130896 Mubashir Iqbal 140786	**B**
24	Bilal Tahir 140701 Bariq Ullah 140773 Mamoon Shahid 140716	**A**
25	Faizan Salamat 140686 Inam Elahi 140728 Danial Ahmed Dar 140725	**A**
26	Danial Fakhar 130930 Saeed Latif 130912 Raza Ali 130861	**B**
27	Ali Shahbaz 140689 Hammad Ashraf 140758 Hassan Zafar 140677	**A**
28	Muhammad Huzaifa 140695 Saqib Majeed 140752 Danish Zafar 140665	**A**
29	Syed Abdullah Shah 140785 Usman Ali 140619 Hamza Ahsan 140704	**A**
30	Sharjeel Waheed 130826 Inam Ur Rehman 130871 Saqib Sharif 130874	**A**