CS 3413

Assignment 1

Due Date: January 13th, 2020 at 12:30 am

ASSIGNMENT IS TO BE COMPLETED INDIVIDUALLY BY ALL STUDENTS!

In the early days of computers, operating systems serviced users by scheduling jobs in the order in which they arrived (started). This is a First-Come First-Serve (FCFS) algorithm.

The downside of First-Come First-Serve (FCFS) is that you are at the mercy of the order in which jobs arrive. A large job at the beginning has everyone waiting and becoming irritated. A solution to this is to try and share the processor with all of the jobs that are wanting to execute. This is called Round-Robin (RR). In this algorithm as jobs arrive they are added to the run list. Jobs in the run list are then each given a fixed period of time to run on the processor. If the job is not finished its total execution when the time is up, then it goes back into the run list to wait for another turn to continue execution. If two jobs arrive at the same time, then the job with the shortest duration goes first in the run list.

To properly implement RR you also need pre-emption – the concept of periodically stopping a job to take away the CPU and decide who is the next eligible process to run. This is needed to prevent large jobs that acquire the CPU and start computing to only hold the CPU beyond their allotted time. To implement pre-emption, you only execute a job for up to the fixed period and then you add it back into the run list.

For example if we have 1 CPU and a fixed period of 2:

User	Process	Arrival	Duration
Jim	A	2	5
Mary	B	2	3
Sue	D	5	5
Mary	C	6	2

This would result in:

Time	Job
2	B
3	B
4	A
5	A
6	B
7	D
8	D
9	C
10	C
11	A

12	A
13	D
14	D
15	A
16	D
17	IDLE

Summary

Jim	16
Mary	11
Sue	17

Write a C program that will read in job requests and print out the corresponding job schedule according to a RR algorithm as above. The input format is each line entry contains a job separated by a tab. The first line of input is ignored as the header. The output format is two tables. First table is the running time and the job currently executing (tab separated). The second table is a summary with the user name (in the order in which they first appear in the input) and the time when their **last** job is completed.

The period is to be specified as a command line parameter when running your program. For example:

```
./a.out 2
```

Would run your program with a period of 2.

NOTE: THE INPUT AND OUTPUT OF YOUR PROGRAM IS SPECIFIED SUCH THAT THE MARKER CAN AUTO TEST YOUR SUBMISSIONS. PLEASE FOLLOW THE SPECIFICATIONS! INPUT WILL BE READ VIA `stdin` (EG. KEYBOARD). OUTPUT SHOULD BE PRINTED TO `stdout` (EG. MONITOR).

YOU CAN FIND THE SAMPLE INPUT AND OUTPUT FILES WITH THE ASSIGNMENT ON D2L. BE SURE YOUR PROGRAM READS/PRINTS THE PROPER INPUT/OUTPUT FORMATTING!