

CS 5/7343
Operating Systems and System Software
Fall 2020
Program 5

Due date: 12/11/20 (Fri) 11:59pm. Absolutely NO extensions.

This program require you to simulate real-time process scheduling by checking whether the real time scheduling algorithms we talked about in class can schedule a set of processes.

Input

You program should read in the name of a file (via command line parameter). The file has the following format

- The first line have 2 integers. The first one is the number of processes (n). The second one is the time (in seconds) (s) that simulation is run
- Each of the next n lined denote a process (you should show them as $P_0, P_1, \dots P_{p-1}$ respectively). Each line has two numbers, the first is the period of the process, the second is the time (in seconds) for each process to run.

A sample input file will looks something like the following:

```
3 150
60 15
35 10
75 25
```

- Your program should print the scheduling done by the two algorithms for the time period specified, and if the scheduling algorithm fails to schedule them, denote when the failure occurs. You should assume the first instance of all processes arrives at time 0.

Output

Case 1: Rate Monotonic Scheduling

For the above file with rate monotonic scheduling, your output should be something like the following table:

Time	Event	Process Running	Process Waiting
0	A: P1-1, P2-1, P3-1	P2-1 (35, 10)	P1-1 (60, 15), P3-1(75, 25)
10	F: P2-1	P1-1 (60, 15)	P3-1 (75, 25)
25	F: P1-1	P3-1 (75, 25)	

35	A: P2-2	P2-2 (35, 10)	P3-1 (75, 15)
45	F: P2-2	P3-1 (75, 15)	
60	F: P3-1, A: P1-2	P1-2 (60, 15)	
70	A: P2-3	P2-3 (35, 10)	P1-2 (60, 5)
75	A: P3-2	P2-3 (35, 5)	P1-2 (60, 5), P3-2 (75, 25)
80	F: P2-3	P1-2 (60, 5)	P3-2 (75, 25)
85	F: P1-2	P3-2 (75, 25)	
105	A: P2-4	P2-4 (35, 10)	P3-2 (35, 5)
115	F: P2-4	P3-2 (35, 5)	
120	A: P1-3	P1-3 (60, 15)	
135	F: P1-3		
140	A: P2-5	P2-5 (35, 10)	
150	F: P2-5 A: P3-3	P3-3 (75, 25)	

Explanation of the table

The table have four columns:

- The first column represents the time that an event happens. We need to have an entry for the following two events:
 - A new instance of the process arrives
 - The current running process finishes
- The second column represent the events mentioned above. The format is (in order of output)
 - F: <the instance of the process that finishes>
 - A: <the instance(s) of the process(es) that arrives>
 - FAIL (see

For each process we denote it by a symbol P<process number>-<instance>. For example, P1-3 denote the 3rd instance of process P1. (The initial instance is known as instance 1). If more than one processes arrives, the process should be listed in increasing order of the process' period. If there is a tie, then they should be listed with process number.

- The third column shows the process instance that is currently running (or ready to start running), and the fourth column shows the process instances that has arrived but is yet to run.

For each case, you should list the process in the following format: P<process number>-<instance> (<period>, <time LEFT TO RUN>). For example P1-2 (60, 5) denotes instance 2 of process 2, which has a period of 60 and this instance still has 5 second to run.

Notice that when there are multiple process that have the same period and is ready to one, we break ties by select the one with the smaller id (i.e. P1 before P2 etc.)

Now consider the following case:

2 80
10 8
12 9

The output table should look something like the following:

Time	Event	Process Running	Process Waiting
0	A: P1-1, P2-1,	P1-1 (10, 8)	P2-1 (12, 9)
8	F: P1-1	P2-1(12, 9)	
10	A:P1-2	P1-2 (10, 8)	P2-1 (12, 7)
12	A: P2-2, FAIL	P1-2 (10, 8)	P2-1 (12, 7), P2-2 (12, 9)

Here the scheduling failed at time 12 when the second instance of P2 arrives when the first instance hasn't finished. In this case, you should put FAIL as an event, and your simulation should stop here with nothing more being output.

Case 2: Earliest deadline first scheduling

For the first case, the output for earliest deadline first scheduling is as follows:

Time	Event	Process Running	Process Waiting
0	A: P1-1, P2-1, P3-1	P2-1 (35, 10)	P1-1 (60, 15), P3-1(75, 25)
10	F: P2-1	P1-1 (60, 15)	P3-1 (75, 25)
25	F: P1-1	P3-1 (75, 25)	
35	A: P2-2	P2-2 (70, 10)	P3-1 (75, 15)
45	F: P2-2	P3-1 (75, 15)	
60	F: P3-1, A: P1-2	P1-2 (120, 15)	
70	A: P2-3	P2-3 (105, 10)	P1-2 (120, 5)
75	A: P3-2	P2-3 (120, 5)	P1-2 (120, 5), P3-2 (150, 25)
80	F: P2-3	P1-2 (120, 5)	P3-2 (150, 25)
85	F: P1-2	P3-2 (150, 25)	
105	A: P2-4	P2-4 (140, 10)	P3-2 (150, 5)
115	F: P2-4	P3-2 (150, 5)	
120	F: P3-2, A: P1-3	P1-3 (180, 15)	

135	F: P1-3		
140	A: P2-5	P2-5 (175, 10)	
150	F: P2-5, A: P3-3	P3-3 (225, 25)	

The only difference is in the description of the process instance in column 3 and 4. Instead of printing the period of the process, you should print the deadline for that process instance. For example P2-4 (140, 10) denotes the 4th instance of process 2, with the deadline of time 140, and still have 10 seconds to run.

Output to the file

WARNING: I intend to grade this homework automatically – i.e. you write your output to a file, and I will write a program that compare your output to the expected text file. So please make sure you follow EVERY instruction on how the output file should be formatted. Even an extra space, or a forgotten commenting of a debug line will cause your output to be different, and you will receive ZERO credit if that happens.

Consider this program not just a test of your OS knowledge, but a more important test on how careful you are in terms of developing software and following requirements.

Rather than trying to format the table, you should put your output in the following format:

- Each row of your table should be output as 4 lines
- The first line should contain a single number, which is the time
- The second line should contain the information in the second column, without the comma and have a single space character separating the various entries. If FAIL is to be printed, it should be all caps
- The third line should contain the information in the third column, exactly as is.
- The fourth line should contain the information in the fourth column, with the comma removed
- There should be no trailing space (or any characters) for each line.
- When there is a space character in the output, only SINGLE space character is allowed.

Your program should first print (to standard output) the result with Rate Monotonic Scheduling, and then print an empty line with NO characters in it, then print the result with Earliest deadline first scheduling.

There should be NO trailing lines after the result for Earliest deadline first scheduling is printed (you should put an end of line character (“\n”) for your last line).

The output file for the example is provided.

Extra credit

Try to extend the program to test (for Rate Monotonic Scheduling only) whether the scheduling algorithm will be successful for the processes for the input (without limiting to the time specified in the input file).

If you attempt that, you should output, after the output for the base case, an empty line, follow by the next line with either a 1 (yes) or 0 (no) to the question above.