

• COMP 2240 Assignment 1 Report

c3238805 Ni Zeng

There are 10 files in the assignment 1 java program:

- **A1.java** (Main class of the Assignment 1 Java program)
- **datafile1.txt**
- **datafile2.txt**
- **FileUtil.java** (subclass to read data)
- **Processor.java** (subclass to store Processor's variable)
- **sub_FBV.java** (subclass for FBV algorithm)
- **sub_FCFS.java** (subclass for FCFS algorithm)
- **sub_LTR.java** (subclass for LTR algorithm)
- **sub_SRT.java** (subclass for SRT algorithm)

Version 1 of Assignment 1 java program:

Reviewing all the result in Version 1 of the java program, all algorithms working on datafile2 and only LTR algorithm does not work on datafile1.txt . (Figure1 and Figuer2).

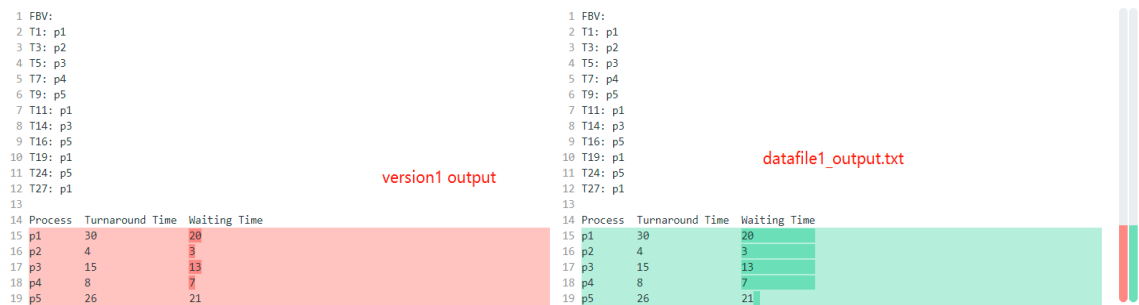


Figure 1. FBV test with datafile1

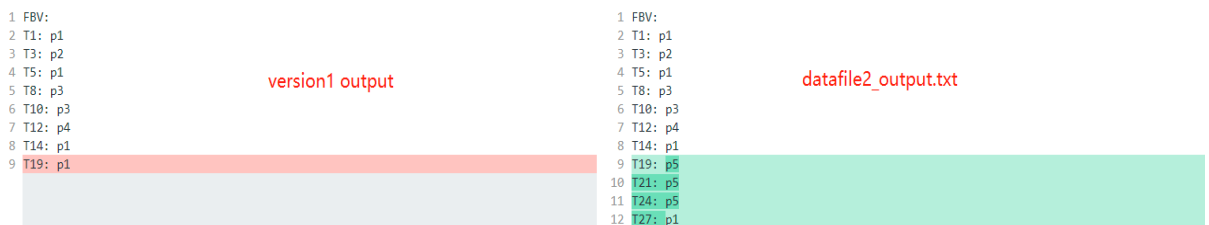


Figure 2 FBV test with datafile2

Version 2 of Assignment 1 java program (latest version):

In order to correct the error causing the mismatched with the give output sample, I use the java debugger to step into each step and check each level queue's stored processor and trace down the location of each processor.

I have then realized I did not check whether the bottom queue has a processor that stays in the bottom queue for more than 32ms each time when a processor finished (I only check the bottom queue when first priority queue run). So, I did added on some code for each priority queue's function and

finally correct the output result that matched both datafile1_output.txt and datafile2_output.txt

Extra datafile is created and test run the java program

After testing with the given datafile1 and 2, I have also created my own datafile to test run the program, following scenario are considering in the created datafile:

Extra processor (more than 5 processor in the database).

If a processor arrive after current time when all previous processor finished execution.

(Considering the idle time during run time).

Variety arrives time and different order of processor's id.

Variety tickets for different processor.

After testing with own created database, version 2 java program can correctly managed all the database and formed correct result output.

Interesting Observations (LRT proportional-share scheduling algorithm):

after a few test run and debugging with version 2 of the java program, I have then noticed the order of the ready queue (when the processor arrived at current Time) in LRT is actually effecting the output result. Base on the calculation that we implement to select the "winner" processor to run next. For example, if we have two different ready queues: P3->P1->P2 and P1->P2->P3 then it will result in different "winner". Because we have been given fixed "random number tickets" to find the winner.