Project 1: Queue Processing

Due date: Sunday June 18 11:00 PM EST.

### You may discuss any of the assignments with your classmates and tutors (or anyone else) but all work for all assignments must be entirely your own . Any sharing or copying of assignments will be considered cheating (this includes posting of partial or complete solutions on any public forum). If you get significant help from anyone, you should acknowledge it in your submission (and your grade will be proportional to the part that you completed on your own). You are responsible for every line in your program: you need to know what it does and why. You should not use any data structures and features of Java that have not been covered in class (or the prerequisite class). If you have doubts whether or not you are allowed to use certain structures, just ask your instructor.

In this project you will simulate the operating system’s selection of processes to send to the CPU. The operating system will select the next process from the of awaiting processes. Each process will require 1 or more the resources A, B and C. Some processes will require only B for example, while another might require A and B, yet another B and C. If the resource is available, the process can be started. If one or more of the resources are unavailable, then the process must wait one cycle. A process that is started will only use a resource for one cycle. A process can only start if all the previous processes have been started. Here is a chart describing a possible scenario:

Starting process list with resources in ( ): P1(A);P2(B); P3(B,C);P4(C);P5(A,B,C); P6(B,C) ;P7(A);P8(A);P9(B);P10(C)

|  |  |  |  |
| --- | --- | --- | --- |
| Cycle | Processes Running | Comment |  |
| 1 | P1, P2 | P3 must wait – Resource B in use Notice P4 can not start ahead of P3 though its resource is available |  |
| 2 | P3 |  |  |
| 3 | P4 | P4 must wait – Resource C used by P3 |  |
| 4 | P5 | P5 must wait – Resource C used by P4 |  |
| 5 | P6,P7 | P6 must wait – Resource C used by P5. P7 can run at same time as P6 |  |
| 6 | P8,P9,P10 | P8,P9,P10 all can run together as no resources are shared. |  |

Total number of cycles needed: 6

There are 2 parts to the assignment, both parts have the same output of the number of cycles , and final length of the queue.

**Part A : Read a one line from the Input file where the line has the format shown here (and above):**

P1(A);P2(B); P3(B,C);P4(C);P5(A,B,C); P6(B,C) ;P7(A);P8(A);P9(B);P10(C)

For each input string from the input file, assign the processes to a list, then execute the list and determine the number of cycles to completely execute the processes. In our example the answer is 6. You should have one output line for each record of the file.

**Part B:**

Randomly generate a list of 20 processes. Start executing processes as before. Randomly select 1,2 or 3 resources (A,B,C) for each process. But at the end of each cycle (regardless of how many processes were run), add 2 more process to the end of the list with 1,2,3 random resources. Output the number of cycles needed to empty the list of processes, but if the list does not empty by cycle 1000, then output the number of processes left (length of the list). Output the length of the list of processes every 100th cycle to watch its growth:

Length of processes at cycle 100: 104

Length of processes at cycle 200: 107

Length of processes at cycle 300: 63

Length of processes at cycle 400: 139

**Number are samples only , your numbers should be different.**

The goal of the exercise to understand how to simulate the operating system’s selection of processes to run.

# Objectives

The goal of this programming project is for you to master (or at least get practice on) the following tasks:

* Read input files
* Work with singly linked list
* Utilize random numbers

**Start early!** This project may not seem like much coding, but debugging always takes time. Analyze and plan now so questions are not being asked a day before the due date.

## Working on This Assignment

You should start right away! First be sure you can replicate the results provided here. Then create some of you own test files. Next run the other provided input files. Finally due to random generation part of the project – do some tracing to show interim results.

**Grading Criteria – 20 points**

1. (3) Input files are read properly

2. (5) Linked list accurately represents the processes and is correctly processed in the program

3. (5) Part A runs properly and has the correct result outputted from the provided list

4. (7) Part B runs properly and has the correct result of the simulation

**Submission:**

A java class that does Part A (using the books jar file) and a java class that does Part B.