

CS 558: Computer Systems Lab (2020)

Assignment Set II

Assignments will be evaluated by the TAs.

- ❖ You should submit report (for answering non-code related questions, if any), complete source codes and executable files (whichever applicable).
- ❖ All codes must be properly documented and good code writing practice should be followed (carry marks).
- ❖ Copying is strictly prohibited. Any case of copying will automatically result in F grade for the whole course, irrespective of your performance in the other parts of the lab.
- ❖ Submission deadline: 8th March 2020
- ❖ Marks distribution: 20, 20, 30, 30.

1. Adding a New System Call to the Kernel

(20 Marks)

When you run a program which calls `open`, `fork`, `read`, `write` (many others may be found at <http://man7.org/linux/man-pages/man2/syscalls.2.html>), you essentially make a system call. System calls help a program to interact with the kernel for performing certain task(s). System calls are required to perform a variety of operations such as: creating processes, building networks, accessing files and IO devices. New system calls can be added as per the system administrator's convenience. In the process of adding a new system call, you may need to modify the kernel source code.

Create your own system call `mysyscall`, which appends the string 'is a student of IITG' with an input name. For example, if the input for your system call is 'Alice', it will print 'Alice is a student of IITG'.

- A) After creating the system call, you have to add it to kernel's log. Note that in the process of modifying the kernel, your system might get corrupted. You are, therefore, advised to use a virtual machine software (virtualbox, vmware, or qemu) to run a guest linux OS. After that, download the latest kernel source code, using which you can start modifying your guest OS.
- B) Also write a sample C program to use the `mysyscall` to test whether your own system call is working properly or not. Once the procedure is completed, generate a patch file. Submit this patch file along with your test C program as your deliverable for this question. The following URL may help you regarding generation of a patch file.

<https://stackoverflow.com/questions/2460558/creating-a-patch-file-from-a-diff-of-2-folders>

2. Car Manufacturing System

(20 Marks)

A car is manufactured at each stop on a conveyor belt in a car factory. A car is constructed from the following parts – chassis, tires, seats, engine (assume that this includes everything under the hood and the

steering wheels), the top cover, and painting. Thus, there are 6 tasks in manufacturing a car. However, there are some **constraints** that are to be followed while making a car. The constraints are:

- Tires, seats or the engine cannot be added until the chassis is placed on the belt.
- The car top cannot be added until tires, seats and the engine are put in.
- The car cannot be painted until the top is put on.

A stop on the conveyor belt at the car factory has four skilled technicians assigned to it- **A, B, C and D**. Each of these four technicians has different skills as mentioned below:

1. **A** is skilled at adding tires and painting the car,
2. **B** is skilled in putting the chassis on the belt,
3. **C** is skilled in attaching the seats,
4. **D** is skilled in adding the engine and putting top cover.

Implement the functionality of all the workers i.e. A, B, C, and D to be able to work on a car, without violating the task order outlined above.

Input to the program: The program will take a number N as input from the command line, which is the number of cars that is to be built at the stop in the factory. Your program should implement the functionality of the workers in the form of threads. And finally, your program should be able to create N cars without violating the conditions.

Output of the program:

For a car, whenever an intermediate task is completed i.e. engine is put on, print a message that the task is completed (with description of the task) along with the technician who has completed the task.

Example:

Correct sequence for building a car would be:

- B put chassis on the belt.
- C attached the seat.
- D put engine.
- A added tires.
- D put top cover.
- A painted the car.

And *incorrect sequence* for building the a would be:

- B put chassis on the belt.
- D put engine.
- A added tires.

- D put top cover.
- C attached the seat.
- A painted the car.

3. Banking System

(30 Marks)

Consider a banking system that can serve N number of customers. The banking system can provide three types of services: draft processing, cheque processing and net-banking. Each customer continuously keeps on requesting one of the three services and the banking system keeps on servicing those requests indefinitely by offering one of the services and providing a voucher number to it. The customer who is requesting the same service gets its request serviced with a unique voucher number and signal the banking system on completion. The banking system then again offer new service and a new voucher number with it.

Following requirements are to be fulfilled in the program:

- I. The banking system should randomly offer one of the services and generate a unique voucher number to it.
- II. For each time a service is offered, maintain a log by writing the service being offered and voucher number in a file on the disk. Also write the customer's information who has availed that service.

Your task is to code this banking system and customers' requests. Synchronisation should be achieved by using a method other than the one used in Q2.

A report also needs to be submitted explaining the way in which solution is developed with its limitation and a way to overcome the limitation.

4. Automated Voting System

(30 Marks)

The city of Dreamland has witnessed the sad news of the sudden demise of their mayor who suffered from cardiac arrest. After a few days of regional grief and condolence, the election commission decided to conduct a by-election. In the by-election, three candidates contested for the position of mayor, viz., X, Y and Z. There are five polling booths in different localities, viz., A, B, C, D and E. To make the election process smooth, fast and hassle-free, the election commission has decided to automate the polling and counting process as follows:

- I. Each booth has an EVM (Electronic Voting Machine) with a controller. A voter cast his/her vote by clicking a button corresponding to the candidate of his/her choice (X or Y or Z).
- II. Each EVM has a controller which is connected to a CCM (Central Counting Machine).
- III. The CCM records the vote count for each candidate.
- IV. Once a voter cast his/her vote, the controller sends the response to the CCM and updates the record file.
- V. Note that all the polling booths operate simultaneously.

- VI. Each booth controller sends a poll completion message at the end of the poll in that booth.
- VII. After receiving poll completion message from each of the polling booths, the CCM displays the final result.

Implement the automated system using concurrency and synchronization with java.

Note: Use a different method for synchronization (other than those used in Q2 & Q3).

[Hint: Input can be provided in batch noting the time, response and booth ID]