OS-Lab Assignments

Teacher: Dr. Fakhrahmad — TA: Amirhosein Gharaati January 2024

There are multiple assignments related to multi-threading and concurrency that you are required to implement using the \mathbf{Java} language.

To maintain the cleanliness of your Java project, follow these guidelines:

- Use **Git repository** and **Github**.
- Create a main branch for the initial project, and **implement each assignment in a separate** branch.
- Create a **private repository** on GitHub and upload your code there.
- \bullet Grant access to the repository using this email: amirgh1380@gmail.com

Upon completing the assignments, send the GitHub address of your project to my Telegram account: $@AG_1380$

Deadline: To be announced in the Telegram channel

1 Maximum number

Find maximum number in an array of large integers.

- First implement a naïve serial implementation in JAVA
- Then implement parallel form in JAVA.
- Report run time in a table like below:
 Performance = (run time of sequential) / (run time of parallel)
 N: array size

Continue increasing N until you get a significant value of performance. Start N from 100000.

	N1	N2	N3	N4
Serial run time				
Parallel run time				
Performance				

2 The Cigarette Smokers Problem

Assume a cigarette requires three ingredients to make and smoke: tobacco, paper, and matches. There are three smokers around a table, each of whom has an infinite supply of one of the three ingredients — one smoker has an infinite supply of tobacco, another has paper, and the third has matches.

There is also a non-smoking agent who enables the smokers to make their cigarettes by randomly selecting two of the supplies to place on the table. The smoker who has the third supply should remove the two items from the table, using them (along with their own supply) to make a cigarette, which they smoke for a while. Once the smoker has finished his cigarette, the agent places two new random items on the table. This process continues forever.

Write a program in Java that simulates this system, use threads for simulating agent and smokers and use condition.

3 Barbershop problem

A barbershop consists of a waiting room with n chairs, and the barber room containing the barber chair. If there are no customers to be served, the barber goes to sleep.

If a customer enters the barbershop and all chairs are occupied, then the customer leaves the shop. If the barber is busy, but chairs are available, then the customer sits in one of the free chairs and sleeps before his turn.

If the barber is asleep, the customer wakes up the barber and the barber also wakes up asleep customers sitting on the chairs.

Simulate the problem using threads and **monitor** in java. You should have one thread for each customer and the barber. Number of customers is much more than the chairs.

4 Roller Coaster problem

Suppose there are n passengers and one roller coaster car. The passengers repeatedly wait to ride in the car, which can hold maximum C passengers, where C is lower than n. The car can go around the track only when it is full. After it finishes a ride, each passenger waits for a random time before returning to the roller coaster for another ride.

- The car always rides with exactly C passengers
- No passengers will jump off the car while the car is running
- No passengers will jump on the car while the car is running
- No passengers will request another ride before they can get off the car

Suppose the car and each passenger are represented by a thread. Simulate the problem using **condition** in java.