

BIM306 Operating Systems

Project-3

Singleton Pattern

Due: May 14, 2019

Background

The Singleton pattern ensures a class has only one instance, and provides a global point of access to it. One of the pattern implementation is double-checked locking

(<http://ceng.eskisehir.edu.tr/abilge/BIM492/icerik/05%20-%20Singleton%20Pattern.pdf>).

Implementation of the pattern in object-oriented programming is given:

```
public class Singleton {
    private volatile static Singleton uniqueInstance;

    private Singleton() {}

    public static Singleton getInstance() {
        if (uniqueInstance == null) {
            synchronized (Singleton.class) {
                if (uniqueInstance == null) {
                    uniqueInstance = new Singleton();
                }
            }
        }
        return uniqueInstance;
    }
}
```

Check for an instance and if there isn't one, enter a synchronized block.

Note we only synchronize the first time through!

Once in the block, check again and if still null, create an instance.

* The volatile keyword ensures that multiple threads handle the uniqueInstance variable correctly when it is being initialized to the Singleton instance.

! If performance is an issue in use of getInstance() method then this method can drastically reduce the overhead.

Project Definition

In the project, first, you are asked to implement getInstance method respect to the double checked locking using with C programming language (POSIX library) in to the given template code. Also, you should use semaphore. This will not be same as object-oriented version naturally, but the idea is the same.

Second, you should explain your code how to implement step by step.