Operating Systems
Term Project
Report

Erencan Güneş
160316023 / İÖ

Hüseyin Güney
160311061 / İÖ

Enviroments: Java

We used java language in the project. There are 4 classes in total. Main class is used to create and start threads. Student class contains the code run by student threads. At the same time, Librarian class contains the code run by library threads. In addition, Library class coded to simulate a library.

The task of the student threads is to read 6 books and complete their life, and the task of the Librarian thread is to receive the read books (in a way, put them back on the shelf) and make the other students' books available to them. This problem creates a synchronization problem in the operation of the program. To prevent this, we used the ReentrantLock class, one of the java's lock libraries. This class has two separate methods we use, lock() and unlock().lock() and unlock() methods are written between only one thread. It means that it can be run. In other words, we have blocked unwanted accesses in this way.

Methods of the Student class:

```
public void takeBook(int book) throws InterruptedException {
    library.books[book].lock();
    library.b[book] = 0;
    limit[book] = 0;
    library.isReturnedBook[book] = false;
    readBook(book);
  }
```

Here, the student who bought the book states that he got the value of the book in the library by setting it to 0. At the same time, he states that he does not want to return it for now. Then he goes to the stage of reading the book with the readBook() method.

```
public void readBook(int book) throws InterruptedException {

    timeForRead = r.nextInt(2000);
    System.out.println(name + " Reading book " +book + " in "+ timeForRead + " milliseconds");
    Thread.sleep(timeForRead);

    deliverBook(book);
  }
```

Here, the student reads the book for a random period of time and switches to the deliverBook() method, which will indicate that it wants to return the book.

```
public void deliverBook(int book){
    library.isReturnedBook[book] = true;
```

```
        library.returnedBook[book] = 1;
        library.books[book].unlock();
        System.out.println(name + " deliver book "+ book);
    }
```

If we think about it here, the student who goes to the library, wants to return the book, and the Library class, which is used as a common area by Librarian and Student threads, also notifies the return request.

Methods of the Librarian class:

```
public void receiveBook() throws InterruptedException {
        while (true) {
            Thread.sleep(r.nextInt(2000));
            random = r.nextInt(6);
            library.books[random].lock();
            if (library.returnedBook[random] == 1 && library.isReturnedBook[random] ) {
                    library.b[random] = 1;
                    library.isReturnedBook[random] = false;
                System.out.println(name + " received book " + random );

            }
            library.books[random].unlock();

           if(!library.isAllStudentRead.contains(false)) {
              System.out.println(name+" 's work is done");
              break;
           }
        }
    }
```

Here, the librarian checks the variables of the Library class, which is used as a common field. If the changes made in the deliverBook() method above are suitable for the controls here, the librarian takes delivery of the book and declares that "this book can now be used by other students." By doing this, we finish the task if all students have read all the books.