FACULTY OF ENGINEERING AND APPLIED SCIENCE

# SOFE 3950U Operating Systems

# LABORATORY REPORT

Professor: **Amin Avan**

**Experiment**:

LAB REPORT #: **5**

LAB CRN NUMBER and group #:    CRNT:  **74025** | Group: **#6**

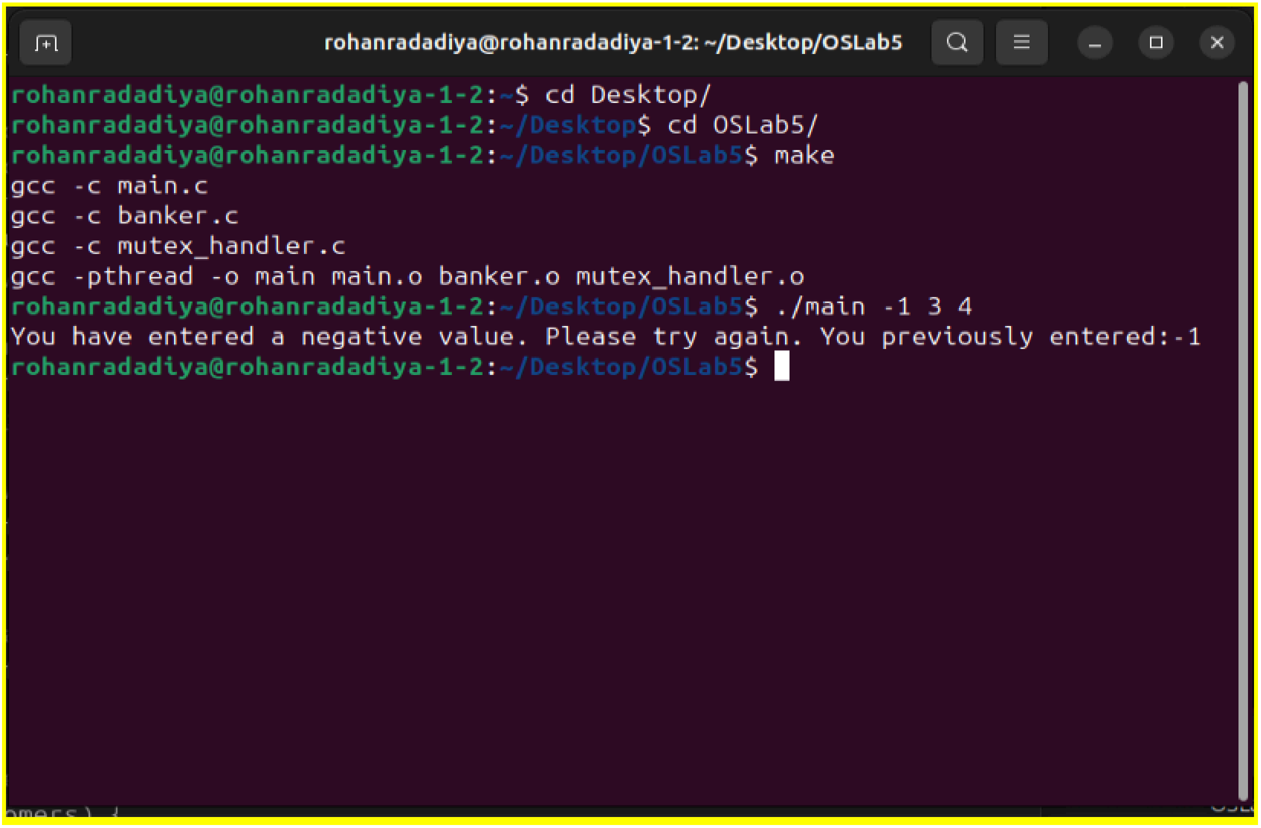
| LAB GROUP MEMBERS | | | | |
|---|---|---|---|---|
| # | Surname | Name | ID | Signature |
| 1 | **Radadiya** | **Rohan** | **100704614** | **R.R** |
| 2 | **Sibi** | **Sabesan** | **100750081** | **S.S.** |
| 3 | **Hussain** | **Syed Airaj** | **100789134** | **A.H** |

## GITHUB LINK: **https://github.com/rohanradadiya/OSLAB5**

## Introduction:

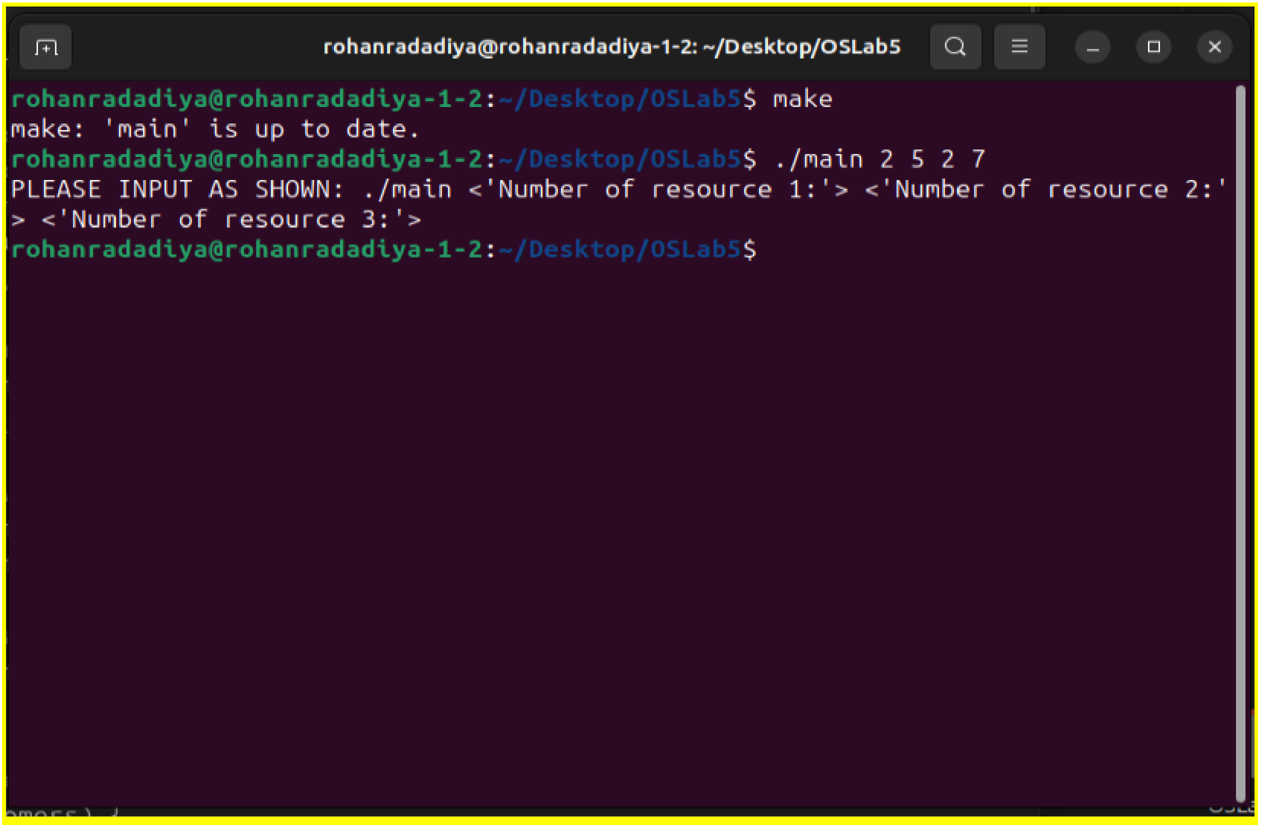
To start off, laboratory 5, discussed in the following lab report, outlines the constructed program and the related inputs/outputs regarding Banker’s algorithm. This lab focused on threads, preventing issues, and avoiding deadlocks, combining them into one task. The general aim was to simulate a bank scenario where customers ask for and give back resources, and the banker ensures the system safety using the banker's algorithm. In addition, threads continuously make requests within certain limits, and mutex locks were used to keep data safe during concurrent operations. The program is started with command line arguments to set the initial resource availability. For example, "./main 10 5 7" sets resources accordingly. Understanding the importance of resource allocation and the reasoning behind processes needing the required resources was a crucial aspect of the practical lab. The practicality and having to use software, hands-on, made it a very useful experience in terms of understanding the overall concepts of the course as a whole.

**Screenshots:**



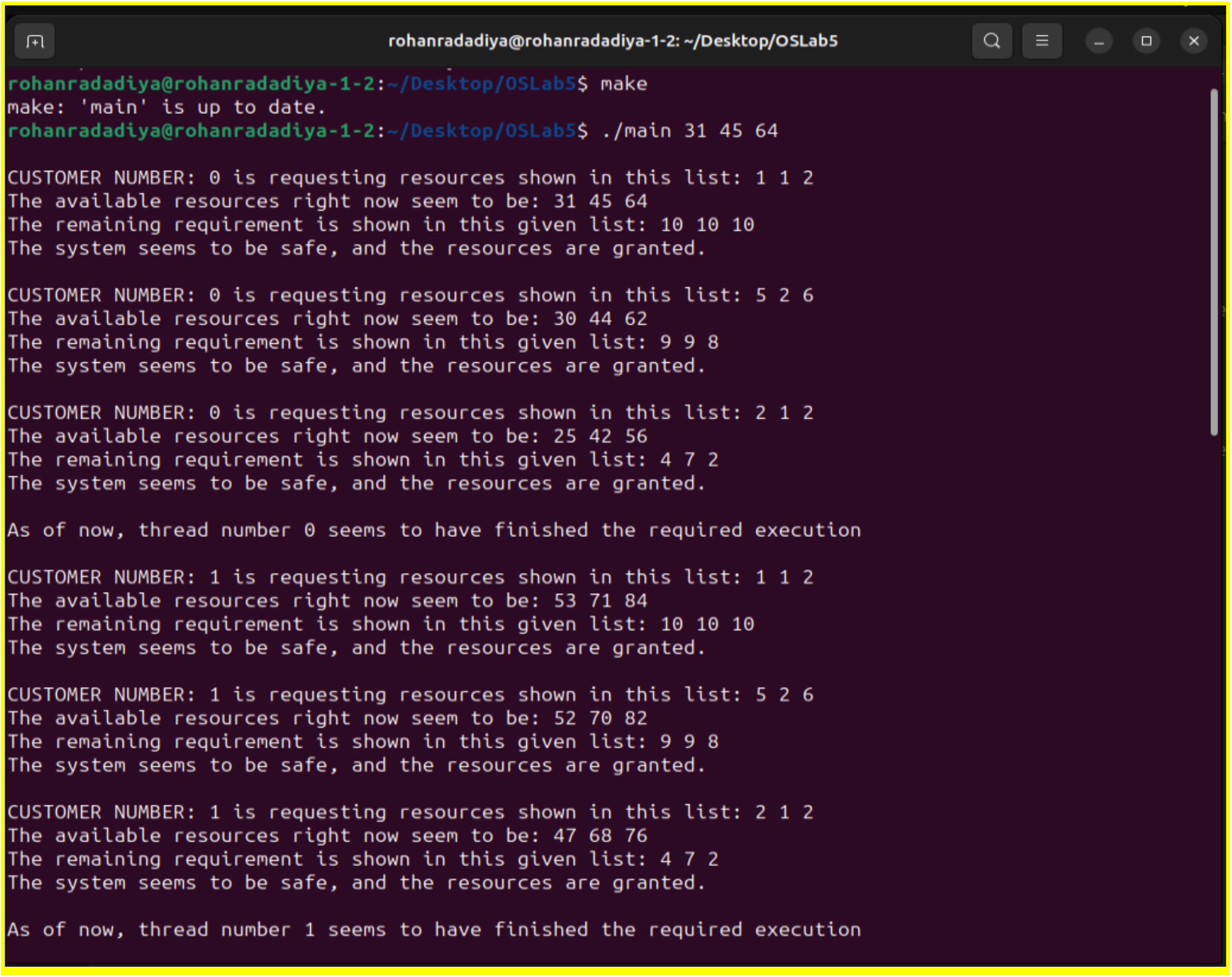
```
rohanradadiya@rohanradadiya-1-2:~$ cd Desktop/
rohanradadiya@rohanradadiya-1-2:~/Desktop$ cd OSLab5/
rohanradadiya@rohanradadiya-1-2:~/Desktop/OSLab5$ make
gcc -c main.c
gcc -c banker.c
gcc -c mutex_handler.c
gcc -pthread -o main main.o banker.o mutex_handler.o
rohanradadiya@rohanradadiya-1-2:~/Desktop/OSLab5$ ./main -1 3 4
You have entered a negative value. Please try again. You previously entered:-1
rohanradadiya@rohanradadiya-1-2:~/Desktop/OSLab5$ 
```

**The image above portrays an error-checking function. The input cannot contain any negative values.**



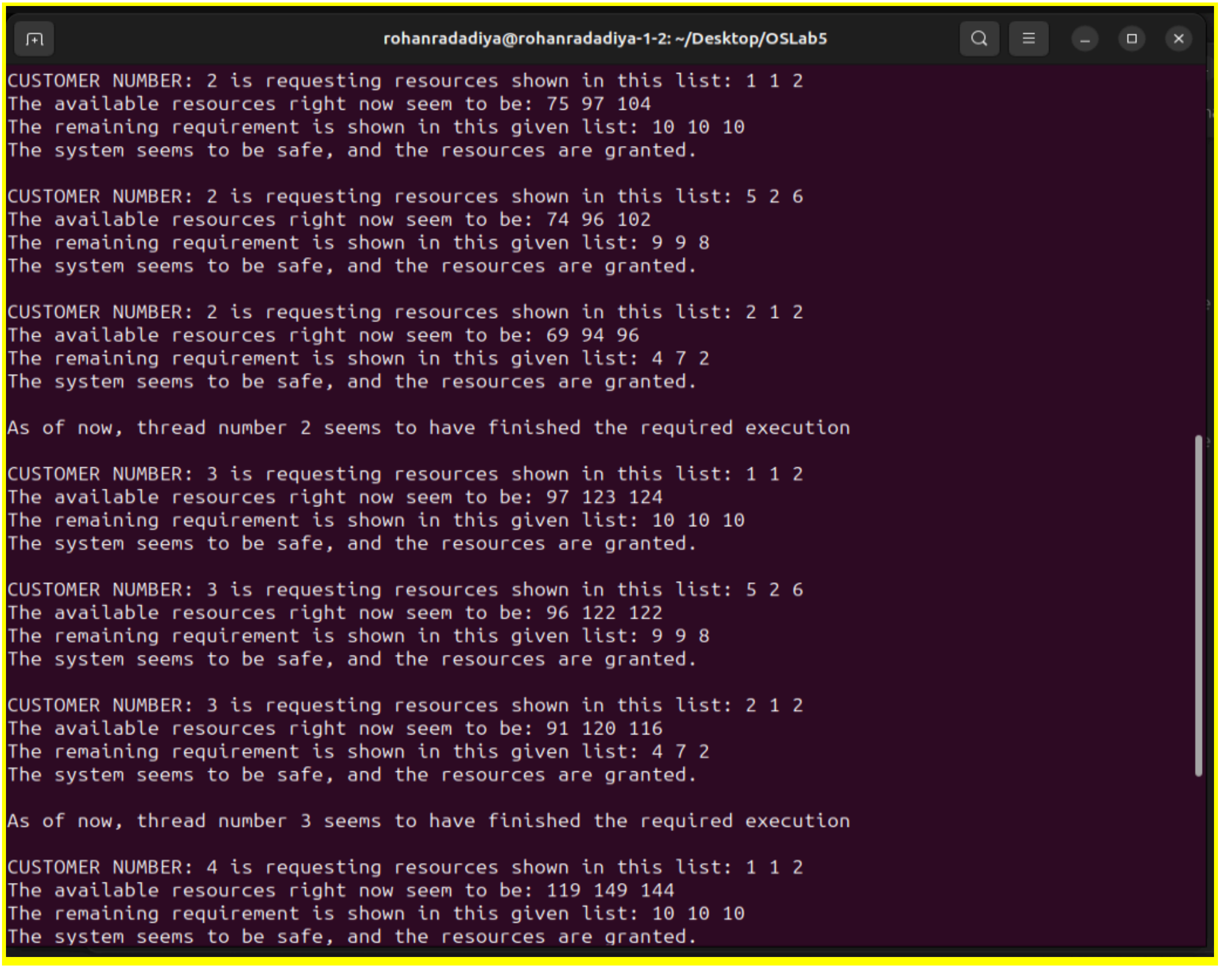
```
rohanradadiya@rohanradadiya-1-2:~/Desktop/OSLab5$ make
make: 'main' is up to date.
rohanradadiya@rohanradadiya-1-2:~/Desktop/OSLab5$ ./main 2 5 2 7
PLEASE INPUT AS SHOWN: ./main <'Number of resource 1:'> <'Number of resource 2:'
> <'Number of resource 3:'>
rohanradadiya@rohanradadiya-1-2:~/Desktop/OSLab5$
```

**The image above shows another error-checking function used. The input must contain 3 (three) values of resources, and not more, and not less than three values of inputs.**

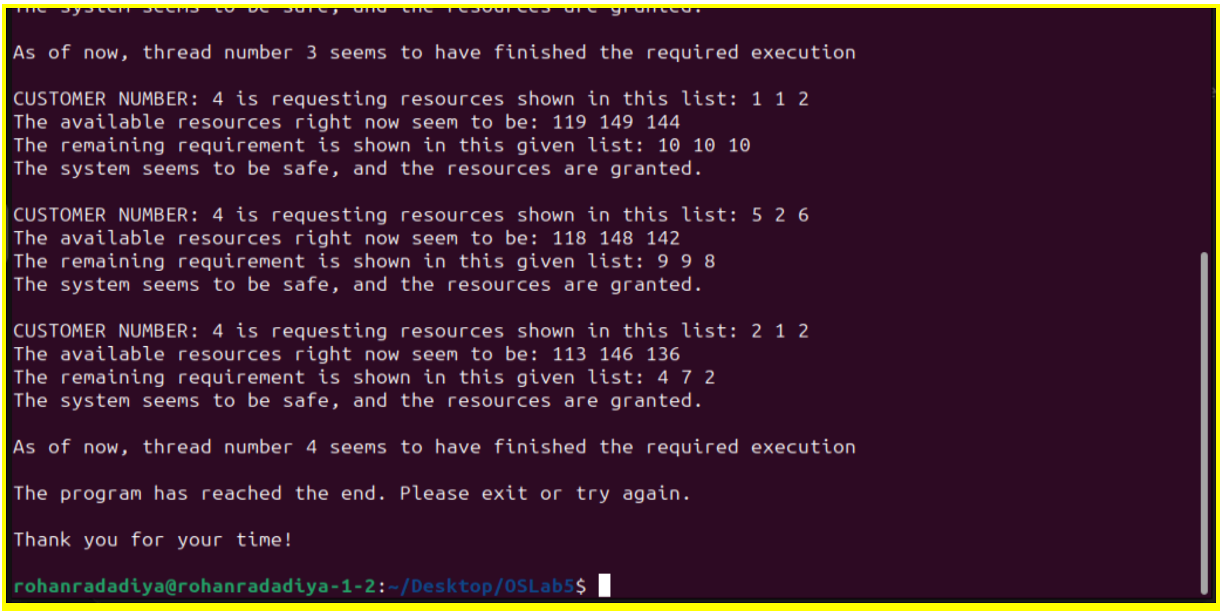


rohanradadiya@rohanradadiya-1-2: ~/Desktop/OSLab5

```
rohanradadiya@rohanradadiya-1-2:~/Desktop/OSLab5$ make
make: 'main' is up to date.
rohanradadiya@rohanradadiya-1-2:~/Desktop/OSLab5$ ./main 31 45 64

CUSTOMER NUMBER: 0 is requesting resources shown in this list: 1 1 2
The available resources right now seem to be: 31 45 64
The remaining requirement is shown in this given list: 10 10 10
The system seems to be safe, and the resources are granted.

CUSTOMER NUMBER: 0 is requesting resources shown in this list: 5 2 6
The available resources right now seem to be: 30 44 62
The remaining requirement is shown in this given list: 9 9 8
The system seems to be safe, and the resources are granted.

CUSTOMER NUMBER: 0 is requesting resources shown in this list: 2 1 2
The available resources right now seem to be: 25 42 56
The remaining requirement is shown in this given list: 4 7 2
The system seems to be safe, and the resources are granted.

As of now, thread number 0 seems to have finished the required execution

CUSTOMER NUMBER: 1 is requesting resources shown in this list: 1 1 2
The available resources right now seem to be: 53 71 84
The remaining requirement is shown in this given list: 10 10 10
The system seems to be safe, and the resources are granted.

CUSTOMER NUMBER: 1 is requesting resources shown in this list: 5 2 6
The available resources right now seem to be: 52 70 82
The remaining requirement is shown in this given list: 9 9 8
The system seems to be safe, and the resources are granted.

CUSTOMER NUMBER: 1 is requesting resources shown in this list: 2 1 2
The available resources right now seem to be: 47 68 76
The remaining requirement is shown in this given list: 4 7 2
The system seems to be safe, and the resources are granted.

As of now, thread number 1 seems to have finished the required execution
```



```
rohanradadiya@rohanradadiya-1-2: ~/Desktop/OSLab5
CUSTOMER NUMBER: 2 is requesting resources shown in this list: 1 1 2
The available resources right now seem to be: 75 97 104
The remaining requirement is shown in this given list: 10 10 10
The system seems to be safe, and the resources are granted.

CUSTOMER NUMBER: 2 is requesting resources shown in this list: 5 2 6
The available resources right now seem to be: 74 96 102
The remaining requirement is shown in this given list: 9 9 8
The system seems to be safe, and the resources are granted.

CUSTOMER NUMBER: 2 is requesting resources shown in this list: 2 1 2
The available resources right now seem to be: 69 94 96
The remaining requirement is shown in this given list: 4 7 2
The system seems to be safe, and the resources are granted.

As of now, thread number 2 seems to have finished the required execution

CUSTOMER NUMBER: 3 is requesting resources shown in this list: 1 1 2
The available resources right now seem to be: 97 123 124
The remaining requirement is shown in this given list: 10 10 10
The system seems to be safe, and the resources are granted.

CUSTOMER NUMBER: 3 is requesting resources shown in this list: 5 2 6
The available resources right now seem to be: 96 122 122
The remaining requirement is shown in this given list: 9 9 8
The system seems to be safe, and the resources are granted.

CUSTOMER NUMBER: 3 is requesting resources shown in this list: 2 1 2
The available resources right now seem to be: 91 120 116
The remaining requirement is shown in this given list: 4 7 2
The system seems to be safe, and the resources are granted.

As of now, thread number 3 seems to have finished the required execution

CUSTOMER NUMBER: 4 is requesting resources shown in this list: 1 1 2
The available resources right now seem to be: 119 149 144
The remaining requirement is shown in this given list: 10 10 10
The system seems to be safe, and the resources are granted.
```



```
As of now, thread number 3 seems to have finished the required execution

CUSTOMER NUMBER: 4 is requesting resources shown in this list: 1 1 2
The available resources right now seem to be: 119 149 144
The remaining requirement is shown in this given list: 10 10 10
The system seems to be safe, and the resources are granted.

CUSTOMER NUMBER: 4 is requesting resources shown in this list: 5 2 6
The available resources right now seem to be: 118 148 142
The remaining requirement is shown in this given list: 9 9 8
The system seems to be safe, and the resources are granted.

CUSTOMER NUMBER: 4 is requesting resources shown in this list: 2 1 2
The available resources right now seem to be: 113 146 136
The remaining requirement is shown in this given list: 4 7 2
The system seems to be safe, and the resources are granted.

As of now, thread number 4 seems to have finished the required execution

The program has reached the end. Please exit or try again.

Thank you for your time!

rohanradadiya@rohanradadiya-1-2:~/Desktop/OSLab5$
```

**The images above shows the output when the user enters "./main 31 45 64". As seen, the correct output is portrayed, and all of the required information is displayed back to the user.**

## **Conclusion:**

All in all, the laboratory experiment discussed in this lab report was very helpful to the group in order to understand multithreading, concurrency management, and resource allocation. Creating the allocation with mutex locks for data integrity, demonstrated various synchronization techniques. Furthermore, we also learned about software configuration practices through using parameters within the program through CLI arguments . In conclusion, Lab 5 helped us to understand the importance of resource allocation and ensuring that the processes have the required resources that they need. If that is not the case, there may be interruptions, forcing the overall program to become less efficient than what was required.