

Objectives:

The primary objective of this assignment was to dive deeper in how threads work, by working with multithreaded tasks that share resources. This also makes us learn about semaphores and mutexes, while also having to deal with deadlock prevention.

Design Overview:

The project was designed with modularity and organization in mind. All the tasks and resources are encapsulated into structs which makes it easier to ensure you are accessing the appropriate arguments regarding a specific task or resource. The project was also designed with many functions to complete individual tasks, abstracting a lot of the code.

Project Status:

The project is working up to specification with the only assumption being that the input file must have either a comment or empty line after the last line that would need to be processed. If an empty line or comment is not present then the last line to be processed would not be processed. Another assumption made is that there will be sufficient resources presented for each task, so there is at least 2 resources if a task requires 2 of such resource.

Testing and Results:

The main method of testing involved debugging and various print statements. As well the example given in the assignment description was used, along with a mixture of personal test cases, such as multi resources and two tasks sharing the same resource.

Acknowledgements:

CMPUT 379 Lecture slides

APUE 3rd edition

OSC 10th edition

Geeksforgeeks

Linux manual page - <https://man7.org/linux/man-pages/index.html>