

Name: Revanth Atmakuri
Student id: 1684293
CCID: atmakuri
Assignment #4, Winter 2022

Objectives:

From my point of view, this assignment helped me to gain experience in multithreaded programming and an in-depth knowledge in Deadlocks and managing shared resources.

Design Overview:

- I have two structs, one is to store the job information and the other is to store all the resources details.
- I have only one mutex to lock/unlock the shared data(That will be modified by the jobs)
- I have a separate integer array to store the status of each job, initially all the status of the jobs are set to idle and as the job runs the status would be updated. So, value 1 would indicate it's in wait state, value 2 would indicate it's in running state, and value 3 would indicate it's in IDLE state.
- For acquiring the resources either for reading/writing that thread requires to lock the data first. So because of this I am able to achieve concurrency.
- all the arrays for jobs and resources are based on index mapping.
- Since, C doesn't have an inbuilt data structure for dynamic arrays. So, I had to pre-initialize the length of the arrays.
- I tried to have some helper functions.
- I tried to have most of the data to be global variables so I don't have to pass them to every thread, the variables that require the modification of their contents by a thread I have a lock for those variables.
- To know each and every step in the program I had a print statement under if condition. So to see progress step by step in the code you can set debug value to 1 by default it is set to 0.

Project Status

Project works as expected.

Assumptions

- Assuming the .a4w22 program will be invoked with inputfile, monitor time and NITER values.
- Assuming the maximum number of jobs are at most 25.
- Assuming all the lines in the input file are at most 100 characters.
- Assuming the maximum number of resources would be at most 10
- Assuming all the time printed to the screen is in millisecs.
- Assuming there will be only one resource line in the inputfile
- Assuming before the start of the job/threads the state of the job is IDLE.

- Assuming the input file will be in the same directory while testing the program.
- Assuming other threads can print to the screen while the monitor thread is printing.

Testing and Results

- I created a custom input file and tested it both on my mac and lab-machine and it works as expected.
- I also tested it with different monitor and NITER values and the output looks as expected.