Korakot Santiudommongkol

CSC412 Assignment 6 Report

Mutex Locks

For synchronization I created 5 different mutex locks, one for drawing the grid, one for showing the state, and one for each of the different colors. I locked the mutex for drawgrid and drawstate right before the draw grid and drawstate function call, and unlocked that lock right after the function call is done. As for the locks on the different colors the locks were put in the function **acquire{color}ink** function and the **refill{color}ink** function because when adding to the color ink tank or subtracting from it, you are adding or subtracting from a shared resource, so I put the lock before adding or subtracting and unlock right after the operation.

Coloring the Grid

I used the **TravelerFunc** for the thread function. In the function I given each individual threads a random row, col, direction, and color. Then after getting all the random necessary things for the thread, it would use the function **placeColoronGrid** to place the color in the right spot on the grid. In the **placeColoronGrid** function it will be in a loop until the thread can get the necessary color it needs from the ink tank, if it can get the color it needs the function will color the grid and that will be the end of the function.

Displacement of the Threads

The way my displacements works is there is an int array which holds two numbers, 1 and 3. Since the next direction of the thread has to be perpendicular which means it has to turn left or right. The way to get a left turn or right turn with the enumerator is (currentdirection + 1) % NUM_TRAVELER_DIR to get a left turn and (currentdirection + 3) % NUM_TRAVELER_DIR for a right turn. So how I randomize which turn the thread should make less is to call the random function and module it by 2, so that the result would be either a 0 or 1 which is the index of the int array. For the displacement I used a while loop which loop until the thread reaches the one of the corner, becoming moving to the next spot on the grid, the thread would be check if it is in the corner or not, if it is the end of thread. For the move to the next grid there is a switch statement which shows how it should be moved, the only problem with this is that if the current position is on the edge of the grid and the current direction will move it outside of the grid, then from that point the direction will be changed to the opposite direction so that the thread does not move outside of the grid.

Difficulties of Assignment

The difficulty I faced during this assignment is the displacement portion of the assignment. As I was testing my program for displacement, I faced with segmentation faults, because I did not consider the possibility of the thread going outside of the grid if it is on the edge.

Limitation

I do not think there are really any real limitations. Other than the fact that I did not implement producer threads and you need to continuous refill the ink tank manually which requires you to hold down 3 buttons. The other limitation is the possibility that out of 9 threads that all 9 threads are the same color or that the 9 threads are only 2 colors.