Homework 6 Notes

1. Overview

- Work with semaphores, locks and/or condition variables to solve another resource management problem.
- Another problem inspired by Operating Systems and determining how to allow many independent threads to use the same resources.

2. Trapped on and Island

- Island with:
    - a = # of adults
    - c = # of children.
- Only 1 boat
- Job the program is to get all the people off the island and back to the mainland.
- Program will take two CLI inputs.
    - Both must be integers and strictly larger than 0.
    - First number is how many adults are on the island.
    - Second number is how many children are on the isalnd.
- Program should produce an error if no value is given or an invalid input is given.
- Each adult and each child MUST be implemented as an individual thread.
    - Once a thread is on the mainland and is no longer needed, it may exit.
- The shared resources between threads is the boat.
    - There is only 1 boat available.
        - 2 seats

            - a driver and a passenger
        - The driver may row the boat from the island to the mainland.
- Boat has the following requirements:
    - Exactly two seats
        - 1 driver who rows the boat
        - 1 passenger who rides the boat.
    - 2 children can ride the boat
    - 1 child and 1 adult can ride in the boat
    - 2 adults CANNOT ride in the boat
    - It takes between 1-4 seconds random to row from the island to the mainland or vice versa
    - No threads may row the boat more than 4 times without a break.
        - Sitting in the passenger seat for 1 direction or on the island counts as a break.
- Need to make sure to always return the boat to the island for the next group.
    - If boat is not returned to the island, all remaining threads will be trapped forever.
    - This means that if two threads take the boat to the mainland, you cannot have both exit or the boat will never be returned.
- Once all threads are safely off the island print out the following statistics:'
    1. Number of times the boat traveled to the mainland
    2. Number of times the boat returned to the island
    3. Number of boats with 2 children
    4. Number of boats with 1 child and 1 adult.
    5. Number of boats with only 1 person (child or adult)
    6. Number of times adults were drivers
    7. Number of times children were drivers

- Can use semaphore provided in lecture
- Do NOT use the starvation free mutex code in lecture will not help you.
- You MUST make sure your I/O and any shared data structures are thread safe.
- You MUST ensure the following:
    - All threads make it back to the mainland
    - The boat never contains an incorrect number of people.
- Once you have completed your code you will write your thoughts in a readme file.

Example

- To run with 5 adults and 3 children

```
./bin/island 5 3
```

2.1 Output Style

- When a thread gets in the boat print out what has happened.

    ```
    # Template
    [Thread Name] got into the [position] seat of the boat.

    # Example if a child got into driver seat and an adult got into the passenger
    # seat we print
    Children 1 got into the driver's seat of the boat.
    Adult 4 got into the passenger seat of the boat.
    ```

- When that boat is being used print out what happens.

```
# Template
Boat is traveling from [start] to [stop].
```

```
# Example if the boat was going from the island to the mainpoint we print
Boat is traveling from island to mainland.
```

- Once all threads have made it to the mainland, print out a final summary using the following template.

```
# Numbers don't matter

Summary of Events
Boat traveled to the mainland: 12
Boat returned to the island: 11
Boat with 2 children: 7
Boats with 1 chilld and 1 adult: 3
Boats with only 1 person (child or adult): 4
Times adults were the driver: 4
Times children when the driver: 11
```

2.4 Readme

- Readme include both instructions and reflections on your code.
- Must be stored in the root of file structure.
- Markdown format.
- No min/max of short essay questions.

1. Your name and drexel ID
2. Instructions
3. Short Essay Question 1: What did you use to protect the boat and why?
4. Short Essay Question 2: How did threads decide what position to take in the boat?
5. Short Essay Question 3: How did you reset the boat for the next group?

6. Short Essay Question 4: Why are you certain everyone will get off the island?
7. Short Essay Question 5: What was the most challenging part of this assignment?

2.5 Makefile

- Must provide a makefile to compile code.
- Must have:
  - make - Builds the program
  - make run - Runs the simulation with 7 adults and 9 children
  - make clean - remove compiled code.

2.2 Implementation

- Regular shit

2.3 Citation

- Reference it if you cheat