CS4328: Project #2

Due on April, 26 at 11:55PM

Mina Guirguis

Please note that this project is copyrighted material. *Late submissions would incur a penalty of 10% per day for up to 2 days, then they will not be accepted. Leave the last few days for documentation, and further testing. Please read the description carefully and see the TA or the instructor (hopefully early) if you have any questions. You may discuss this project with other students. However, you must write your code and your report on your own.*

1 Overview

In this project, we are going to build the game “MatchingCards”. MatchingCards is a simple card game played with 6 players and a single deck of cards. A game is composed of 6 rounds. In each round, one of the players will act as the dealer and will be responsible for shuffling the deck of cards, keeping a single target card to be matched, distributing the cards to the players and waiting for the round to end. In round 1, player 1 will be the dealer, in round 2, player 2 will be the dealer and so on.

In the first round, player 1 will start by shuffling the deck of cards, drawing one card to his/herself (called the target card) and giving each player one card. Player 1 will then place the remaining deck of cards on the table and signals that the round begins. In this round, players in a round-robin fashion and starting with player 2 will draw a card from the deck and compares his/her hand to the target card. If none of the cards in his/her hand match the target card, the player will discard one card at random by placing it at the end of the deck of cards and the next player proceeds. If there is a match (e.g., two kings, two queens, two jacks or any two numbers of the same value), then the player will show their hand, declaring his/herself as the winner and the round ends. At the end of the first round, player 1 will signal the following round to start (i.e., player 2 will be the dealer). A game ends when the 6 rounds are completed.

2 Implementation

This project is to implemented in C/C++ using POSIX threads. You can check:

`https://hpc-tutorials.llnl.gov/posix/`

for a tutorial on the POSIX thread library. The main function should create 6 threads – one for the players. Notice that we want to keep the threads synchronized and to protect any shared objects (e.g., deck of cards, log file, printing to the console). At the end of every round, each thread must print a message if they won or lost this round (e.g., “Player 2 lost round 4” or “Player 3 won round 1”, etc.). The main program takes a seed as an argument for the random number generation (which will be used in shuffling and in discarding cards).

3 The Output

All players will write into a log file each action they take. The log file should be able to describe exactly what is happening at each step of the game. The log file should look something like this:

PLAYER 2: hand 5
PLAYER 2: draws 7
PLAYER 2: discards 7 at random
PLAYER 2: hand 5
DECK: contents of the deck, separated by spaces (e.g., 1 2 3)

```
PLAYER 3: hand J
...
```

The final messages for a round (say round 3) should look something like:

```
...
PLAYER 2: hand 3
PLAYER 2: draws 6
PLAYER 2: hand (3,6) <> Target card is 6
PLAYER 2: wins round 3
PLAYER 6: lost round 3
PLAYER 5: lost round 3
PLAYER 1: lost round 3
PLAYER 4: lost round 3
PLAYER 3: Round ends
...
```

The output of the program to the screen (not in the log file) should indicate the target card at the beginning of the round, the hand of each player when they draw a card and the deck of cards:

```
PLAYER 1: Target Card 6
PLAYER 2: hand 5,7
PLAYER 2: WIN (yes/no)
DECK contents of the deck, separated by spaces (e.g., 1 2 3) ...
```

4 Submission

Submission will be done on Canvas. Submissions will include the code, a report containing a brief overview of the design and implementation, the results of 5 independent runs of the program with different seeds, and instructions on to how to compile and run the game.