CS 4328 / CS 5305

Operating Systems

Programming Assignment # 1

Assigned: 9/11/20

Due: Sunday, 10/4/20 at 11:55 PM

This project is going to take a good amount of work and time, so please start early. You would not be able to finish if you start a few days before the due date. Late submissions would incur a penalty of 20% per day for up to 3 days, then they will not be accepted. Leave the last few days for documentation, further testing and formatting the results.

1. Overview

In this project, we are going to build a simulation of "Pair War". Pair War is a simple card game with one dealer, 3 players, and a single deck of cards.

The game is composed of three rounds. At the beginning of each round, the dealer shuffles the cards, deals the first three cards (one card per player) and waits for the round to finish, before repeating the same process for the next round.

A winner in a round is the first player to get a "pair".

In each round, a different player is given the privilege to start. Initially, the dealer shuffles the deck of cards and hands each player a single card in a round robin fashion (say, starting from player 1 for round 1). Once the dealer is done handling the cards, the dealer places the remaining deck of cards on the table and the first round begins. Each player (starting from player 1), draws a card from the deck and compares it to the card he/she has. If they are a pair, the player shows the hand, declares him/herself a winner and the round ends. Otherwise, the player will discard one card (at random) by placing it at the bottom of the deck of cards on the table, and the next player proceeds to draw.

Once a round ends, the dealer shuffles the deck again and hands a card to each player. In the second round, the second player starts drawing a card from the deck. In the third round, the third player starts drawing a card from the deck.

2. Implementation

This project is to be implemented in any programming language using **POSIX threads**.

You can review:

https://computing.llnl.gov/tutorials/pthreads/ (for a tutorial on the POSIX thread library)

https://en.wikipedia.org/wiki/POSIX_Threads (for more info on this subject)

The main function in your program should create a thread for the *dealer* and 3 threads for the *players*. Notice that we want to keep the threads synchronized and to protect any shared

objects.

Also, when a player wins, he/she needs to inform the other players so they can exit the round. Each thread should print a message when it finishes (e.g., “Player 1 wins and exits”, “Player 2 exits”, etc.).

The main program takes a seed as an argument for the random number generation (which will be used in shuffling and discarding cards).

3. The Output

a) Log file

The dealer and the 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.

The log file should look something like this:

PLAYER 1: hand 5

PLAYER 1: draws 7

PLAYER 1: discards 7

PLAYER 1: hand 5

DECK: <contents of the deck, separated by spaces (e.g., 1 2 3)>

…

The final messages for a round should look something like this:

PLAYER 2: hand 3

PLAYER 2: draws 3

PLAYER 2: hand 3 3

PLAYER 2: wins

PLAYER 2: exits round

PLAYER 1: exits round

PLAYER 3: exits round

DEALER: shuffle

b) Screen

The output of the program to the screen (not in the log file), *for each round*, should indicate for each player: the hand, the status (win: yes/no), and the remaining deck of cards. Example:

PLAYER 1:

HAND card1 card2

WIN yes

PLAYER 2:

HAND card1

WIN no

PLAYER 3:

HAND card1

WIN no

DECK: <contents of the deck, separated by spaces (e.g., 1 2 3)>

Notice that the hand of the winner should show two cards (the winning pair) and the hands of the other two players should show only one card. A single run of the program should have 3 rounds with 3 winners.

4. Submission details

Submissions shall be done using the **Assignments** tool on the Canvas website for this class. Please submit a single zip file including all your files.

Name your file **program1_xxxxx.zip** where xxxxx is your TX State NetID.

Submissions shall include:

- the program's source code, and
- a report containing:
    - a brief overview of the design and implementation,
    - the screen results of 5 independent runs of the program with different seeds, and
    - instructions on how to compile and run the program on the CS Linux servers.

You need to write your program in any language that supports POSIX threads.

You can develop it on your own Windows machines (check "POSIX Threads for Windows" on the above wiki page), *however* it is your responsibility to ensure it runs on the CS Linux servers *with a command line input* – nothing graphical (GUI-based) or IDE. Please indicate clearly how to compile and run your program.

5. Grade breakdown

70% of the grade is on the program itself, divided as follows:

20% Developing the correct design and data structures,

20% The correct use of POSIX threads, and

30% That the program compiles and runs as expected.

30% of the grade is on the report (equally divided as outlined above).

Have fun with the project!