Operating Systems - CS 4328

Project 2

Group Members

- Cameron Peterson - cnp23
- Tyler Cummings - trc119

Design and Implementation

Design

This project is intended to simulate a game called "Pair War", which involves a deck of 52 cards, three players, and a dealer. For each round, the dealer shuffles the deck and deals out a single card from the deck to each player. Each player then draws a card from the deck, resulting in a hand of two cards. If the two cards match, the player with the matched hand wins the round and another round begins. However, if no player has a matched pair, then each player must discard one of their two cards and withdraw another until a match occurs.

Once a round has completed, all of the player's cards are returned to the deck.

Implementation

Each player is represented by an instance of a `Player` struct. Each struct instance contains `PlayerNumber` (1 through 3) and `Hand`, which is a vector representing the player's hand. The deck is also represented as a vector. The dealer isn't explicitly represented, but dealer behaviors are performed in the `shuffle_deck()` and `deal()` functions. `Player` behaviors are executed in the `draw()` and `discard()` functions. A mutex is utilized to ensure fair access to the deck and status condition, viz. `WIN` or `!WIN`.

For each turn within a round, a thread is created that corresponds to a player. Within each thread is a call to the `turn(void* pl)` function, where `pl` is the player whose turn it is. In this function, the mutex locks for the deck (`deck_mutex`) and game status (`status_mutex`) are obtained. Cards will be removed from the player's hand if said hand has a size greater than or equal to 2. Once the hand size is equal to 1, the player will draw a card from the deck. If the two cards in the hand match, the player wins the round. The deck and game status mutexes are released and the thread terminates.

Once a player has won, the `end_round()` function is invoked, which releases the memory allocated to the vectors that housed the deck and player's hands.

Output is handled in two ways, namely console output and file output. The file output will be written to `log.txt`. If `log.txt` already exists, its contents will be overwritten with the new output. Otherwise, `log.txt` will be created.

vector.c and vector.h

We felt that utilizing a vector-like container to house the deck, player hands, and players would simplify the work required to complete this project and improve its overall readability.

Results

The results of five independent runs with five different seed values are contained in the `/Report/` folder.

Instructions

To compile and run the program, perform the following commands:

```
cd OpSys_Project_2/
make
./RUN <SEED VALUE>
```

where `<SEED VALUE>` is any integer (e.g.,-1, 0, 1, 2, 3, etc.).

Upon completion of the run, a file named `log.txt` can be found, which contains output similar to that of the console output.