CS4328: Project #2

Due on November, 24 at 11:55PM

Mina Guirguis

Colten Murray

Implementation and Design:

In order to implement the matching dice game, a thread was allocated for each player and dealer using pthreads. The matching dice game requires 4 players and 1 dealer so 5 threads are needed in total. The first 4 player threads will run the player function for rolling the dice rollDice(void* arg) and the dealer thread runs the function for dealing the game, deal(void* arg). The argument passed in both of these functions is the thread ID. The threads are declared together in a pthread_t array named handle. From there the first four threads are initialized using a for loop that runs the pthread_create function for each thread. It is important that the dealer thread is initialized after this, because the player threads need to be set up first in order to ensure that the dealer thread is initialized properly.

When the player threads are initialized, a pthread_cond_wait is triggered for each player to wait for their turn to be signaled. Once the player threads are set up, the dealer thread is initialized causing the dealer to send the turn signal to the first player. After that, the dealer waits until it receives the game_won signal. When the first player receives the turn signal it unlocks and rolls the dice for their turn. The sum is calculated and then compared to its partners last roll. Once the players turn is complete they send the turn signal to the next player. This continues in a while loop until one of the comparisons match then the game_won signal is sent to the dealer and each player thread exits.

Test Runs:

Individual Run 1 (seed = 5):

This seed resulted in a short run of only 3 turns in which team A and C won in the quickest way possible. This run shows the slight advantage team A and C receives from getting to compare sums first.

Individual Run 2 (seed = 573):

This seed resulted in a much longer run of 12 turns in which team B and D win.

Individual Run 3 (seed = 657456):

This seed resulted in an even longer run of 23 turn is which team A and C win.

Individual Run 4 (seed = 78201568471):

This seed resulted in a rather short run of only 6 turns. This run shows two things, the size of the seed does not necessarily correlate with the length of the run and that comparisons are not only made at the C and D turns. It is possible for A and B to declare the winner if their roll matches their partners from the last batch of rolls.

Individual Run 5 (seed = 78201568472):

Although this run results in a similar length of run 4, there is no indication of any pattern to the generated numbers of each run. Although the seeds are only one number apart, they seem to have generated completely different numbers.

Compile and Run:

The project is compiled on the Texas State CS servers using the command:

“g++ -O3 –pthread project2.cpp –o project2”

The project is ran using the command:

“./project2 seed”

seed is where the int argument is placed that sets the seed for the run. For example:

“./project2 573”