Report for Project 2

Design and Implementation:

The program can be broken up into three major functions that define the overall behavior of the game. The program starts in main where the supplied arguments are then used at the randomized seed, number of players, and potato chip bag respectively. Then the mutexes are initialized along with the seed and player variables. From there we use two for loops, one to represent each round for the game which will create threads for each round and the second loop creates the total number of threads equivalent to the number of players. The dealer thread gets created first as the dealer is responsible for setting up the game and once it terminates/joins the card playing threads are established and their work is performed. The rest of main is reserved for joining the threads appropriately and destruction of mutex/thread resources.

The dealer function starts by pulling the associated player id from the arguments provided during the thread creation process. This function represents most of the dealer behavior including establishing deck from the initialized deck, shuffling utilizing the random shuffle algorithm for shuffling the deck, pulling the greasy card from the randomized deck, and storing each supplied card for the player function use. There is other functionality including the associated console and output logging, establishing the condition for which the player threads terminate by setting the flag, and setting up the winner variable. The appropriate mutexes are used to isolate the critical sections of resources so data racing can be prevented.

The player function starts by pulling the associated player id from the args argument provided during the thread creation process. The vector that represents each player’s hand is established and the card supplied. A while loop checks to see if the round has ended, which is established once a player’s hand has a card that matches the greasy card established as a global variable. The first if statement determines which player goes next based on current player, locking/unlocking the deck mutex and drawing their card. The next if statement is used to compare the current player’s hand with the greasy card matches establishing the winner and setting up the break condition to exit the while loop. Otherwise, the else statement handles the random discarding of a card from the player’s hand and the chip eating logic for the bag. The next part of the logic establishes who the next player is. The last of the logic is just for unlocking/locking the file/console mutex and writing the appropriate information for logging purposes.

Lastly, to avoid cramming documentation the associated text files of this program have both the linux display output and the log file output provided.