Calvin Nichols 0588016

# ***Operating Systems Assignment #3: Blackjack!***

***Problem:***

Create a blackjack server that connects up to 4 clients, forks a dealer program which handles the game of blackjack with each client.

***Solution:***

*Server:*

I used select with the timer flag to allow for asynchronous handling of clients. Within that I allowed clients to connect, check for the password and if it was wrong or they did anything other than simply send the password, they would be disconnected.

To keep track of clients information I made a linked list containing:

- timer
- connected time
- file descriptor
- success

Timer contained the length of time before timeout, so 5 seconds for the password to be sent and 10 seconds for a response to the dealer. Connected time is equal to the time the client connected, this value may be reset if the timer needs to be reset (ex. due to a hit). The file descriptor is the file descriptor it connected on and success is dependant on the password.

*Timer:*

To check the timer I use difftime function to return the difference in seconds, I use this for all my timers including my global timer and when the timer has been exceeded they are disconnected.

***Decisions/Assumptions/Justifications:***

I assumed if any user sent anything unexpected they want to be disconnected. I also decided against using sigalrm as I felt it was unnecessary and didn't provide any real benefit over the strategy I employed. I felt my method worked well because I didn't have to deal with the sigarlm triggering at an inconvenient time.

***Limitations and Bugs:***

I am unsure if I properly implemented all conditions with the cards properly.

I also assume that the client is sending "HIT\0" and "STAND\0" and if the buffer in the client sends anything extra they will be disconnected. I only mention this because other people working in the lab ran into this problem.

I did not use a great method for obtaining the file descriptors in my program. I simply attempted to send a bunch of file descriptors cards and send would return -1 if it was not successful send. This is not optimal for a few reasons, I lose some cards and it is possible I miss a file descriptor since I only check them until 15. This basically guarantees I will never get one connected to a later file descriptor unless a ridiculous amount of clients attempted to connect and failed. I think it is an okay solution for this assignment because it is very simple and should work for almost all situations except the most extreme.

Calvin Nichols 0588016

My documentation and commenting is much worse than my previous two assignments because of time restraints, I would have liked to fix that before submissions.

***Testing:***

I did all testing by hand.

I ran the server with different combinations of clients. I used clients that sent garbage, timed out and acted properly. I tried combinations of multiple clients and single clients. I did almost all testing locally but I did do some across the lab computers to confirm it worked in all cases.

***References:***

linked list
http://www.c.happycodings.com/Data_Structures/code5.html

select_server.c Jason Ernst