Number Guess - Sockets

This assignment gives the programmer experience with using the sockets to communicate between processes. Sockets communicate between the server process and the client process. This assignment will specify what the server process and the child process requirements for the assignment

The program will play a number guessing game between a game background server and a guessing game client. The server starts by randomly generating a number between 1 and 100 inclusive, the client tries to guess the server random number by writing each guess to the server. After each client guess, the server writes back to the client a message that indicates a win, that the guess number is higher or that the guess number is lower. The client gets twelve number guesses.

General Program Requirements

Use global constants for use by the program:

All arrays that require a size declarator should be of size declarator of 1024.

Use the standard file operations for the sockets read(), write(), close()… etc.
Do not use FILE and the f_ file commands for all file operations.

You must use try-catch LineInfo.h throws for all socket API command.

Use C++ (g++) features (iostream, string… etc.).

Build using g++ -std=c++11.

The names of the source programs and associated execution files are:

```
sockclient.cpp
sockclient

sockserver.cpp
sockserver
```

Program Command Line

To run the game:

```
./sockserver &
./sockclient
```

When the client has finished playing the game, the client closes the socket which causes the server to sense the socket dis-connect causing the server to exit.

Number Guess - Sockets

sockserver process

```
create a socket
check for socket creation failure

setup the sockaddr_in structure
  use 8888 for the socket port number

bind the created socket
check for socket bind failure

listen to the socket
  Use a listen queue size of three

accept the socket listen
check if the accept is a failure

create a random number and associated random number string

loop

   read a guess message from the client

   check if message received is a request for random number
   write to the client socket random number string

else // a no guess
   if read size is > 0
      convert read message to int
      make guess no string

      if the random number equals the guess number
         set a result string that states the status win

      else if (random number > guess number)
          set a result string that states the status of the number is greater than the guess no string

      else if (random number < guess number)
          set a result string that states the status of the number is less    than the guess no string

      write to the client socket the status result string

   else if read size is zero
      display the client disconnected
      fflush stdout
      break the loop

loop end
```

Number Guess - Sockets

sock client process

```
create a socket
check for socket creation failure

setup the sockaddr_in structure
  use 8888 for the socket port number

connect to the created socket

check for socket creation failure

display start of game for numbers between 1 and 100

set number of tries to zero

loop (game)

   display current try number
   prompt for guess number and get input

   convert guess number to guess number string

   write to server socket the guess number string
   check for socket write failure

   read the server socket message
   check for socket read failure

   Check if receive message contains Win
      break the loop

   check if the client maximum number of guess tries has been exceeded

   display the maximum number of tries

   write to server to send random number
   check for write error

   read random number message from server
   check for read error

   convert to an int no
   display the random no
   display game over
   break

loop End (game)
```

Number Guess - Sockets

Activities File

After done designing and testing, capture the run results in an activities text file named activities.txt.

This activities.txt file must contain a successful guess number try and an unsuccessful guess number try.

Use the copy and paste technique to a local editor text file for creating the activities.txt file.

Do not use script or terminal session to create the sockguess.txt file.

See the sample run below.

Submittal

Put in a zip folder that you name:

LineInfo.h
sockserver.cpp
sockclient.cpp
activity.txt

Submit the .zip folder to blackboard.

Number Guess - Sockets

```
{babbage:~/courses/3377/assigns/06} ./sockserver &
[1] 7005
{babbage:~/courses/3377/assigns/06} ./sockclient


Try to guess a number between 1 and 100

Current Try Number : 1
Enter number guess :

50

Server reply :

The number is greater than : 50

Current Try Number : 2
Enter number guess :

75

Server reply :

The number is less than : 75

Current Try Number : 3
Enter number guess :

63

Server reply :

The number is less than : 63

Current Try Number : 4
Enter number guess :

56

Server reply :

The number is less than : 56

Current Try Number : 5
Enter number guess :

53

Server reply :

The number is less than : 53

Current Try Number : 6
Enter number guess :

51
```

Number Guess - Sockets

```
Server reply :

The number is greater than : 51

Current Try Number : 7
Enter number guess :

52

Server reply :

You Win !!

The number is : 52

{babbage:~/courses/3377/assigns/06} Client disconnected
```

Number Guess - Sockets

```
{babbage:~/courses/3377/assigns/06} ./sockclient


Try to guess a number between 1 and 100

Current Try Number : 1
Enter number guess :

50
50
The number is less than : 50

Server reply :

The number is less than : 50

Current Try Number : 2
Enter number guess :

25
25
The number is greater than : 25

Server reply :

The number is greater than : 25

Current Try Number : 3
Enter number guess :

26
26
The number is greater than : 26

Server reply :

The number is greater than : 26

Current Try Number : 4
Enter number guess :

27
27
The number is greater than : 27

Server reply :

The number is greater than : 27

Current Try Number : 5
Enter number guess :

28
28
The number is greater than : 28
```

Number Guess - Sockets

```
Server reply :

The number is greater than : 28

Current Try Number : 6
Enter number guess :

29
29
The number is greater than : 29

Server reply :

The number is greater than : 29

Current Try Number : 7
Enter number guess :

30
30
The number is greater than : 30

Server reply :

The number is greater than : 30

Current Try Number : 8
Enter number guess :

31
31
The number is greater than : 31

Server reply :

The number is greater than : 31

Current Try Number : 9
Enter number guess :

32
32
The number is greater than : 32

Server reply :

The number is greater than : 32

Current Try Number : 10
Enter number guess :

33
33
The number is greater than : 33

Server reply :
```

Number Guess - Sockets

```
The number is greater than : 33

Current Try Number : 11
Enter number guess :

34
34
The number is greater than : 34

Server reply :

The number is greater than : 34

Current Try Number : 12
Enter number guess :

35
35
The number is greater than : 35

Server reply :

The number is greater than : 35


The maximum number of tries 12 is done.
Send Random Number
The random number is : 36

Game is over

{babbage:~/courses/3377/assigns/06}
Client disconnected
```