Program description in short

A betting game server accepting clients on tcp sockets, letting the clients pick a number within a published range, and every 15 seconds randomize a winning number, and then letting the clients know if they won or not.

Description of the program

The program must accept both IPv4 and IPv6 connections on TCP port 2222 and handle up to BEGASEP\_NUM\_CLIENTS clients. The code must allow for this define to range from 1 to 64500.

An unique client id must be generated upon receipt of a BEGASEP\_OPEN message, this client id must never collide with another connected client.

If the maximum number of simultaenous clients is reached the server must tear down the connection, else it must respond to the BEGASEP\_OPEN message with a BEGASEP\_ACCEPT message with the allocated client id and the allowed betting number range.

When the client has sent a BEGASEP\_BET message, which must follow no other message than BEGASEP\_ACCEPT and have a betting number within the published range, the server is to include that client in the next betting run.

The client must only send the BEGASEP\_BET message once, any protocol breaches must immediately result in a connection teardown by the server.

A betting run must be performed every 15 seconds but only if there are clients connected.

The betting run must generate a random number within the published betting number range and broadcast the winning number to all clients with

the BEGASEP\_RESULT message, indicating for each client if they won or not.

Each winner should also be printed on a seperate line to stdout, syslog,

or similar.

The connection to each client that betted in this run must then be closed.

The program must be able to run "forever" without interaction.

The following defines and values are to be used:

#define BEGASEP\_NUM\_CLIENTS 40

#define BEGASEP\_NUM\_MIN 0xe0ffff00

#define BEGASEP\_NUM\_MAX 0xe0ffffaa

Protocol specification

The Betting Game Server Protocol, BEGASEP, is a binary protocol

in network byte order. Messages are described below.

BEGASEP common header included in all messages:

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Ver | Len | Type | Client Id |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

version = 4 bytes

len = 4 bytes

type = 1 byte

client id = 2 bytes

**struct BEGASEP\_CommonHeader**

**{**

**int ProtocolVersion**

**int PacketLength**

**int PacketType**

**int ClientId**

**};**

Ver (Version) is always set to 1.

Len (Length) is the length of the entire message in bytes, including header.

Type is the message type (BEGASEP\_\*) as described below.

Client Id is assigned by the server in the BEGASEP\_ACCEPT message

and must thereafter be used in all communication between the server

and client. It must be set to 0 in the BEGASEP\_OPEN message.

Message type BEGASEP\_OPEN = 1

This message has no payload.

It is sent upon connect by the client and may be sent only once.

Client Id must be set to 0.

Message type BEGASEP\_ACCEPT = 2

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Lower end of number range |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Upper end of number range |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

This message is sent by the server in response to a BEGASEP\_OPEN message,

the Client Id is set to a dynamically assigned and unique id

in the range 0..BEGASEP\_NUM\_CLIENTS-1

The Lower and Upper end of betting range fields are set to the

BEGASEP\_NUM\_MIN and BEGASEP\_NUM\_MAX defines.

Message type BEGASEP\_BET = 3

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Betting number |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

This message is sent by the client in response to a BEGASEP\_ACCEPT.

The Betting number is the number the client is betting on and

must be within the number range returned in the BEGASEP\_ACCEPT message.

If it is not, the client is to be discarded.

Message type BEGASEP\_RESULT = 4

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Status | Winning number... |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Win num cont. |

+-+-+-+-+-+-+-+-+

This message is sent by the server to all clients when a winning number

has been generated and is sent to all clients that have sent a

BEGASEP\_BET message.

The Status field is set to 1 if this client had sent the winning number

in its BEGASEP\_BET message, or 0 if it did not match.

Winning number is simply the winning number.

When this message has been delivered to the client the connection

must be torn down and terminated and the client forgotten.

Environment

This program must be written in C (not C++) and be compilable with gcc

or Microsoft Visual Studio C++. Only standard operating system libraries

may be used (such as -lc, -lnls, etc).