PROYECTO PARA COMPUTACIÓN 2

# 

# SERVIDOR DE JUEGOS

**INTEGRANTES:**

* ***Matias Figueredo***
* ***José Remaggi***

## Descripción del Server

Se quiere hacer un servidor de juegos. En el cual cada jugador acceda al mismo a través de un **login**. Una vez adentro el servidor lo pone en espera hasta que aparezca un competidor para el juego en al que está intentando jugar (en un principio será sólo 2 juegos par o impar mayor o menor adentro o afuera, y adivinar el número. El servidor almacenará en una base de datos la información de la partida. La conexión se hará a través de **sockets** para IPv4 e IPv6 (**unexpect**). Se va a crear un hilo por usuario y un proceso por cada tablero.

El servidor va a ser el tablero. Se usarán pipes para la comunicación de los jugadores.

Va a haber una bitácora que vaya registrando y mostrando a los jugadores quiénes se conectan, quiénes ganan partidos y quiénes se desconectan.

En la bitácora se usará memoria compartida como forma de IPC

Hilos:

1 por jugador

1 para movimientos

1 contador de movimientos de la partida por jugador

1 para login

1 para bitácora

Procesos:

1 por tablero(partida)

Cosas a compartir:

puntajes de los jugadores

nombre del jugador

Funcionalidades:

* observar partidas: n usuarios pueden meterse a una partida en curso para mirar. El sistema debe mostrar quienes están jugando y el usuario selecciona de entre esas opciones. Esta opción la ofrece si no hay más tableros disponibles para iniciar partidas. El usuario puede dejar de observar o comentar en cualquier momento durante la observación

Jugar partida

* el usuario ingresa al sistema
* el sistema pide ingresar que juego va a jugar
* el usuario ingresa juego elegido
* El sistema busca partida sin finalizar del juego elegido. Si hay ingresa al usuario en esa partida, y avisa al resto de los jugadores que la partida se reanudó
* Si no el sistema busca tableros de ese juego disponibles
  + si hay tablero disponible suma al jugador a ese tablero
  + si no hay tableros disponibles el sistema le pregunta al usuario con qué cantidad de jugadores desea jugar
  + el usuario ingresa cantidad de jugadores n
  + el sistema crea un tablero para n jugadores y suma al jugador al tablero
  + si no hay mas cupos de tablero para crear el sistema ofrece al usuario observar una partida

Mientras se está jugando una partida se puede:

* Abandonar la partida
* Enviar mensaje a los jugadores en el tablero

Ideas de posibles funcionalidades:

* Marcar a la gente que suele abandonar los partidos
* chat
* ver tabla de posiciones(cantidad de partidas jugadas y ganadas, porcentaje de abandono). Esto debería poder hacerse cuando se termina de jugar una partida

Diferencia entre abandonar una partida y dejarla sin terminar. Abandonarla implica que el jugador pierde y la partida sigue. Pero en el caso de que una partida no haya terminado y siga habiendo jugadores que no la abandonaron estos pueden volver a conectarse para seguir jugando.

Debe haber un archivo de configuración que incluya cantidad máxima de jugadores por partida, cantidad máxima de tableros máxima.

**Base de Datos**

* Usar base de datos relacional. Tabla para usuarios, juegos, tableros, mensajes, puntajes

**Juegos:**

**Par o impar, mayor o menor, adentro o afuera:** Se puede jugar de a n jugadores. Juego de cartas. Se juega con cartas francesas sin comodines(52). 4 pintas de 13 cartas. Cada jugador tiene un mazo de cartas propio. El tablero es compartido y los puntajes son compartidos.

* Son 3 fases.
  + Primera fase se pregunta si es par o impar. Se saca una carta
  + Segunda fase: se elige si es mayor o menor. Se saca una carta.
  + Tercera fase: se pregunta si está adentro o afuera del intervalo formado por las dos cartas anteriores. Se saca una carta
* Puntaje: cada jugador que adivine suma 1 punto por ronda
* Después de cada ronda se revisa si hay empate de puntajes. Entre los que empaten en primera posición se juega otra ronda mas, y asi hasta romper el empate.
* Si en la primer partida 3 jugadores empatan en primera posición los 3 ganan 3 puntos y juegan otra vez. Quien gane esa partida gana 3 puntos más, y así sucesivamente si hubiera más empates.

**Adivinar numero:** juego de prueba de funcionalidades. Entre n jugadores se tira un dado por jugador y se suman sus valores. Los jugadores deben decidir qué número es. El que adivine gana 3 puntos. Si hay empate (varias personas eligen el mismo número) se juega otra ronda y quien gane gana 3 puntos. El juego no te avisa si tu numero esta fuera de rango de la suma de los dados, pero si constata que sea un entero positivo

## ESTRUCTURA DEL CÓDIGO

Por cada archivo ***.C*** que hagamos tiene que haber un archivo ***.H*** que se incluya. De esa manera el código queda ordenado

Ej:

puertos.c puertos.h

|  |  |
| --- | --- |
| #include<stdio.h>  #include<stdlib.h>  #include<string.h>  #include"puertos.h"  char\* puerto\_itoa(unsigned short puerto){    unsigned short L,H;  char \*cpuerto;  cpuerto=(char\*) malloc(sizeof (char) \* 8);  puerto&=0x000000FFFF;  L=puerto&0x000000FF;  H=puerto&0x0000FF00;  H>>=8;  snprintf(cpuerto, 8, "%d,%d",H,L);  return cpuerto;  }  unsigned short puerto\_atoi(char \*cpuerto){    unsigned short puerto,L,H;  char \*aux,cpuertoAux[8];    strncpy(cpuertoAux,cpuerto,sizeof cpuertoAux);  aux=strstr(cpuertoAux,",");  if(aux!=NULL){  \*aux=0;  }else{  return 0;  }  L=atoi(cpuerto);  H=atoi(aux+1);  L<<=8;  puerto=H+L;  return puerto;  } | #ifndef PUERTOS\_H\_  #define PUERTOS\_H\_    unsigned short puerto\_atoi(char \*cpuerto);  char\* puerto\_itoa(unsigned short puerto);  #endif |

La llamada en el main o el archivo ***.C*** que lo use, se hace incluyendo el archivo ***.H*** así #include "puertos.h"

## 

## LINKS

<http://stackoverflow.com/questions/1618240/how-to-support-both-ipv4-and-ipv6-connections>