Trabajo Práctico Final

Sistemas Operativos I

Dzikiewicz, Luis Enrique
Legajo: D-3850/4
luisdzi.87@gmail.com

Ernandorena, Iván
Legajo: E-1115/1
ivan.ernandorena@gmail.com

Güella, Julio
Legajo: G-5061/1
julioguella@hotmail.com

Fecha de entrega: -2017

Docentes

Guido Macchi
Guillermo Grinblat
José Luis Díaz

Diseño del trabajo práctico

El trabajo práctico consta de dos archivos que permiten su funcionamiento:

- `server.erl`: En él se encuentra todo lo relacionado a los servidores. Se tratan las conexiones de los usuarios, creación de juegos, jugadas, y demás comandos.
- `game.erl`: Se encarga de la interfaz y de la parte visual relacionada a el juego de TA-TE-TI, por ejemplo, se encuentran las funciones encargadas de imprimir la ayuda, el tablero, chequear cuando un jugador gana, etc.

Conceptos de diseño

- Al inicializar un servidor, se crea un nuevo proceso donde se llevará una lista con los nombres de los clientes de todo el sistema distribuido llamado `list_of_client`, que se registra globalmente con el nombre de `clients_pid` para poder acceder a este proceso desde cualquier nodo. Se realiza lo mismo, para un proceso registrado globalmente como `games_pid` para llevar los nombres de las partidas en curso. Además tanto los clientes como las partidas en curso han sido registradas globalmente. También, se da inicio y registro local a los procesos de `pbalance` y `pstat`. Luego de iniciar estos procesos claves para el almacenamiento de información y funcionamiento del sistema se pasa a la función `dispatcher` con el Socket donde el nodo recibe conexiones entrantes.
- Con respecto al balanceo, la función `pstat` se encarga de enviarle a la función `pbalance` de todos los nodos el estado de carga de él mismo. Luego, `pbalance` recibe el estado de carga de todos los nodos y se queda con el nodo de menor carga, así cuando un proceso desea saber que nodo es el de menor carga, `pbalance` le envía este nodo.
- Un juego consta de sus jugadores, observadores, el tablero de juego, y de quien es el turno actual. Los jugadores dentro de un juego estan almacenados de la forma {`N,Jugador`}, donde `N` es un 1 o un 2 correspondiente al turno y `Jugador` es el nombre registrado globalmente (un átomo). Por diseño, el usuario que crea el juego con el comando `NEW` va a ser el jugador número 1 y el turno se establece aleatoriamente.
- Cuando se realiza un cambio en un juego, el servidor envía automáticamente a cada jugador y observador de ese juego el nuevo tablero indicando quien fue el jugador que realizó esa jugada. Se obvió el comando `UPD` por esto, ya que esta alternativa parece más clara y eficiente para los usuarios.

Inicializar el sistema

- Antes de inicializar la consola de Erlang, ejecutamos el comando `$epmd -daemon` por si el comando `$erl` arranca automaticamente el epmd (Erlang Port Mapper Daemonepmd). Es para evitar el error "register/listen error: econnrefused".
- Antes de abrir la terminal de Erlang con `$erl` debemos conocer la IP del equipo en la red. Una de las formas de obtenerla es con el comando `$ipconfig`. Una vez tenemos la IP abrimos la terminal de Erlang de la siguiente manera: `erl -name nodo@IP`, donde `nodo` es el nombre que le queremos dar al mismo y `IP` la que se obtuvo anteriormente.
- Compilamos el servidor con `1>c(server).` El servidor se arranca con la función `start\2` del módulo `server`. Esta función toma como parametros un nombre para el nodo (puede ser el mismo con el que se abrio el shell), un puerto disponible y una lista con el nombre de los demás nodos de la forma `nodo@IP` que en principio puede ser vacia ya que no hay otros nodos en el sistema.
- Si se quiere conectar otro servidor que esté en otra PC al sistema distribuido se debe: Conectar al server TCP mediante `gen_tcp:connect(Address,Port,[Options])`, donde `Address` es la dirección IP donde se encuentra

el otro servidor, `Port` es el puerto con el que se inicializó el servidor y `[Options]` puede ser la lista vacía. Si se requiere algun tipo especial de conexion ver las opciones aqui.

Luego, se deben sincronizar ambos nodos (al sincrnoziar un nodo con otro que ya está sincronizado con otros, automáticamente se sincroniza con todos los demás) con `net_kernel:connect_node(Node)`, donde `Node` es el nombre que se le dio al nodo. Si se quiere saber el nombre de un nodo se puede utilizar la función `node()`. Además, luego de la sincronización, se puede saber con que nodos estoy conectado con `nodes()`.

- Si tenemos dos servidores en la misma PC, basta conectar ambos servidores con la función `net_kernel:connect_node(Node)`.

Como jugar

- Una vez que el sistema distribuido, con uno o mas nodos, está corriendo, se puede conectar a este desde otra terminal a traves del protocolo `telnet`. Para conectarse se debe conocer la IP de uno de los nodos y el puerto donde trabaja. `telnet IP Puerto` basta para conectarse al sistema.

- Lo primero que se debe hacer es registrarse con un nombre de usuario (`CON 'nombre'`). Luego de esto se puede comenzar viendo los juegos disponibles con `LSG` o creando un nuevo juego `NEW 'nombre juego'`. Para ver todos los comandos disponibles una vez conectado está disponible la ayuda con `HELP`.

- Al estar permitido participar en más de un juego a la vez. Cada vez que se muestra un tablero de un juego, en la parte superior se muestra el nombre del juego y quienes son sus jugadores junto con que simbolo tiene asignado cada uno. Además se indica quien fue el último en realizar una jugada, sobre todo para que los observadores puedan seguir mejor el progreso de una partida y si un jugador está jugando varias partidas simultaneamente pueda saber con exactitud a que juego pertenece el tablero que se le muestra.