



Proyecto II - Bingo (18%)

Como segundo y último Proyecto se propone desarrollar una aplicación que permita, a un número indeterminado de participantes, jugar al bingo. El juego consiste en un bombo en cuyo interior hay un número determinado de bolas numeradas. Los jugadores juegan con cartones que llevan impresos una cantidad fija de números aleatorios, dentro de un rango determinado. El cantor va sacando bolas del bombo, cantando los números en voz alta. Si un jugador tiene dicho número en su cartón lo tacha continuándose el juego hasta que alguien marque todos los números de su cartón.

Se desea que Ud. implemente una solución en Lenguaje C/C++ bajo una interfaz agradable (pudiera ser incluso en modo texto) en la Distribución Linux de su preferencia

Esta aplicación constará de tres ejecutables correspondiéndose cada uno de ellos con los roles que se pueden dar en este juego, es decir, un narrador que va cantando los números, los distintos participantes del juego y un juez, que comprueba la validez del resultado y determina un ganador.

Para la correcta implementación de la aplicación se hará uso de los distintos recursos del sistema operativo tratados durante las prácticas de la asignatura.

Sala de Juego (Memoria Compartida)

La sala de juego se compone esencialmente de una sala, con aforo limitado a MAX_PARTICIPANTES jugadores, en donde estarán:

- Además de los participantes, el locutor y al menos un juez.
- El bombo, con una cantidad finita (MAX_BOLAS) de bolas, cuyos valores variarán en el intervalo MIN_NUMERO y MAX_NUMERO.
- La pila de tarjetas, con un número fijo de números impresos (NUMEROS). Cada participante podrá jugar simultáneamente con un número máximo de tarjetas, indicado por MAX_TARJETAS_JUGADOR, siendo por tanto $\text{MAX_PARTICIPANTES} * \text{MAX_TARJETAS_JUGADOR}$ el número máximo de tarjetas existentes.

Que la sala tenga un aforo limitado implica que nunca podrá haber más participantes que el máximo número permitido, lo cual no implica que deba llenarse para poder comenzar el juego. Si un participante desea jugar estado el aforo completo, deberá esperar a que finalice el juego actual (alguien cante bingo) y a que haya un hueco para poder entrar a la partida, teniendo siempre prioridad la gente que actualmente ha jugado.

Conociendo estos requisitos, la sala deberá mantener, al menos, esta información:

- Identificadores: array estático con los identificadores de los jugadores.

- Participantes: indicará el número jugadores que hay en la sala.
- Juez: valor del identificador del juez de la partida.
- Tarjetas: array estático de tarjetas. Cada tarjeta deberá al menos contener el array de números de la tarjeta y el identificador del jugador al que pertenece (1 en caso de que no se esté jugando esa tarjeta).
- Ganador: identificador del participante que ha cantado bingo y que por tanto, puede ser ganador de la partida.

Procesos

Como bien sabemos, en un bingo se ven involucrados tres tipos distintos de personajes: el que canta, al menos un juez y los distintos participantes. Cada uno de estos personajes tiene un rol bien definido no coincidente con el rol de cualquier otro personaje.

Cantor o Cantador

El cantor esperará a que pueda comenzar el juego (haya al menos un participante) y esté presente el juez. Mientras tanto, prepara las tarjetas e inicializa las estructuras de datos y valores que puedan ser necesarias. Una vez que se dé por comenzada la partida no podrán entrar nuevos participantes, y el cantor irá sacando números del bombo y avisando a los participantes presentes. La forma de avisar a estos participantes será a través de una cola de mensajes. Como es lógico, no se podrá sacar dos veces la misma bola del bombo.

Cuando alguno de los participantes cante bingo, avisará a uno de los jueces para que vaya a comprobar la validez de la tarjeta (sólo se hayan tachado los números cantados y no queden más números por cantar), confirmando o rechazando la validez del bingo. Si el bingo es cierto, se dará por finalizada la partida y podrá iniciarse una nueva.

Participante

El jugador sólo se encargará de ir tachando de sus tarjetas los números cantados por el cantor.

Una vez que tenga todos sus números tachados, avisará al cantor mediante la señal BINGO, poniendo su identificador en la variable correspondiente de la memoria compartida.

El participante tendrá dos modos de juego, manual, en el que se presentarán por pantalla los números de la tarjeta y podrá ir tachándolos, indicando el número mediante teclado, o cantar bingo si cree que lo ha conseguido, escribiendo bingo mediante teclado. En este modo manual el programa participante no realizará ninguna comprobación, y ha de confiar en lo indicado por el usuario, salvo si el número indicado no está en ninguna de sus tarjetas, en cuyo caso mostrará un mensaje de error.

En modo automático se simulará un jugador experto, que tachará los números conforme son cantados, y cantará bingo tan pronto como termine de tachar una tarjeta.

Se han de implementar los dos modos de juego evitando la repetición de código, de forma que este se defina claramente en unas pocas funciones, y sea sencillo añadir nuevos modos de juego a la aplicación.

Juez

Cuando se ha definido la sala de juego, se ha descrito que al menos deberá existir un juez para comprobar la validez de la tarjeta ganadora. Por norma, un juez sólo podrá estar al tanto de un número máximo de jugadores (MAX_JUGADOREZ_JUEZ). En caso de que haya más jugadores, deberá llamarse a más jueces aunque aquel juez que fue invocado en primera instancia, mediante el ejecutable correspondiente, será el único juez capaz de hablar con el cantor para determinar la validez de la tarjeta ganadora. Supongamos que en la sala hay 11 participantes y que cada juez sólo puede tener a su cargo a 5 jugadores. En este caso, deberá haber 3 jueces de los cuales 1 habrá sido invocado mediante el ejecutable del juez y los otros dos serán hijos del primer juez y por tanto relegados a un segundo puesto, no pudiendo hablar directamente con el cantor sino sólo con su padre.

Cuando un jugador cante bingo, el cantor avisará al juez de primera instancia para que compruebe la validez del bingo. El juez de primera instancia elegirá entonces a uno de sus hijos (o él si no hay más jueces) para dicha comprobación. El hijo correspondiente (si lo hay) comprobará las dos condiciones necesarias para certificar el bingo avisando a su padre de la decisión tomada, dejando el identificador de la memoria compartida intacto identificando que efectivamente el proceso con ese identificador es el ganador, o modificándolo a 1 si el proceso no cumple las condiciones necesarias para ser proclamado ganador. El proceso padre será el encargado de hacer llegar la decisión al proceso cantor. Este aviso se realizará mediante una señal.

Semáforos

El acceso a las tarjetas de la memoria compartida, para tachar los números cantados por el cantor, deberá estar protegida mediante un semáforo, de forma que se aseguren las condiciones de concurrencia y exclusión mutua.

Deberá existir también un semáforo n-ario encargado de controlar que no accedan más jugadores a la sala de los permitidos por el aforo.

También se ha de evitar que se modifique la identificación del Ganador hasta que el juez no dictamine sobre una posible comprobación de bingo en curso. Para ello se usará un semáforo adicional que se ha de obtener para cantar bingo, permitiendo así al cantor y al juez atender las peticiones de una en una, declarando ganador al primero que cantó bingo de forma correcta.

Mensajes

Como ya se ha comentado en el párrafo anterior, el cantor deberá avisar a los participantes del número de la bola obtenida aleatoriamente del bombo mediante el envío de un mensaje. El jugador al recibir el mensaje estará recibiendo el número a tachar de su tarjeta, en caso de que éste esté impreso en ella.

Alarmas y Señales

Parece razonable definir un cierto control sobre cuándo empezar la partida de bingo. Este control se realizará por tiempo transcurrido desde que entre el primer jugador a la sala.

Cuando un jugador entra en la sala, se unirá al array de semáforos y a la memoria compartida y escribirá su identificador en el array de participantes de la partida siempre y cuando no se supere el aforo. Una vez hecho esto, se iniciará un cronómetro que, llegado al tiempo especificado

(TIEMPO_INICIO) se iniciará la partida, inicializándose todas las estructuras que puedan ser necesarias.

Todas aquellas señales que no sean necesarias, deben estar enmascaradas, y por tanto no permitidas, de tal forma que en cada momento estarán activas solo las señales que la aplicación tenga previsto recibir.

Aclaratorias

- El programa subastador (./cantor) se instanciará solo una vez, es decir, sólo existirá un proceso cantor, que será el encargado de la gestión de la memoria compartida (sala).

La finalización de este proceso se deberá a una de las siguientes razones:

- Todos los postores abandonan la sala
- Se produce un error grave en la ejecución de alguno de los programas.
- El programa participante (./participante) se instanciará tantas veces como jugadores vaya a haber en la sala, hasta que se cumpla el número máximo de clientes o hasta que, por tiempo, se inicie el juego.

El postor deberá encontrarse en un bucle infinito realizando las acciones correspondientes hasta que el usuario decida abandonar la sala. Para ello, pulsará CTRLC, liberando así todos los recursos que él haya reservado. Este programa deberá ir notificando por pantalla todas las acciones que realice sobre la sala, así como los avisos enviados al proceso cantor.

- El programa juez (./juez) se instanciará una única vez y creará tantos jueces como sea necesario, según el número de participantes que haya en la sala y la variable correspondiente definida. Deberá notificar por pantalla todas las acciones que realice, ya sean señales recibidas, enviadas o modificaciones de las variables de la memoria compartida.
- El código deberá estar organizado de la siguiente manera (Recomendación):
 - cantor.c, participante.c y juez.c: serán los ficheros fuentes de únicamente los programas principales correspondientes.
 - memoria.c y memoria.h: código fuente y fichero de cabeceras de las funciones que tratan directamente con la gestión de la sala. (API de la memoria compartida)
 - senales.c y senales.h: código fuente y fichero de cabeceras de las funciones de manejo y gestión de las señales. (API de señales)
 - mensajes.c y mensajes.h: código fuente y fichero de cabeceras necesarios para la gestión y uso de las colas de mensajes. (API de mensajes)
 - semaforos.c y semaforos.h: lo referente a semáforos (API de semáforos).
 - bingo.c y bingo.h: gestión y lógica de la aplicación en sí. Se encargará de llamar a las funciones de las apis anteriores al igual que de cumplimentar los requisitos del juego.
 - defines.h: definición de todos aquellos defines que sean necesarios.

Nota Final

Se deberá asegurar, en todo momento, la integridad de todas las estructuras que se usen, el buen uso de los recursos necesarios y asegurar en todo momento la perfecta concurrencia y sincronización de los procesos. Las esperas han de ser no ocupadas, evitando al máximo el consumo de tiempo de CPU.

Todos los accesos concurrentes a la memoria compartida deben estar convenientemente protegidos mediante el uso de semáforos. Además será necesario controlar todos los retornos de las llamadas a las funciones del sistema, modularizar y comentar el código y liberar todos los recursos que hayan sido utilizados, devolviéndose al sistema operativo a la situación inicial que tenía antes de ejecutar las aplicaciones correspondientes.

Informe Explicativo

Adicionalmente se debe entregar un informe que describa el análisis, desarrollo y funcionamiento del programa, el mismo debe contener los siguientes puntos:

- Marco Teórico
- Algoritmos Utilizados. Descomposición Modular
- Descripción de la Estructura de Datos Utilizada
- Especificaciones de Entrada
- Especificaciones de Salida
- Diccionario de Variables.
- Tabla de Funciones Definidas
- Diagramas UML (Si Aplica)
- Consideraciones.
- Restricciones.
- Conclusiones

Fecha de Entrega: Lunes 09-Marzo-2015 / Hora: 09:00am

Lugar: Laboratorio de Base de Datos

Integrantes: Dos (02) personas / Grupos Asignados