

Universidad Nacional
Escuela de Informática
Curso: Sistemas Distribuidos

Primer proyecto

El objetivo de este proyecto es poner en práctica el esquema de subscripción-publicación mediante la creación de la simulación de un juego de BINGO. Para ello será necesario crear tres tipos de procesos: el proceso jugador, el proceso difusor y el proceso emisor. La simulación podrá involucrar un proceso emisor, un proceso difusor y muchos procesos jugadores. Toda la programación se debe realizar en Lua y utilizando la librería *zmq* para comunicar los diferentes procesos.

El proceso jugador

El proceso jugador debe iniciar seleccionando en forma aleatoria los números de su cartón y almacenándolos posiblemente en una matriz. El cartón consiste de cinco filas y cinco columnas. Tenga en cuenta que los números de la primera columna van del 1 al 15, la segunda del 16 al 30, la tercera del 31 al 45, la cuarta del 46 al 60, y la quinta del 61 al 75. Sin embargo, la posición central del bingo queda vacía. Se debe verificar que no se repitan números en la misma columna.

Posteriormente el proceso jugador se debe subscribir al proceso difusor y esperar a que le sean enviados los números jugados. Cada vez que llegue un mensaje con un número jugado el proceso debe verificar si existe en su cartón y de ser así lo debe marcar. Posteriormente debe imprimir el cartón en la consola mostrando el estado actual del mismo.

Si un proceso logra completar (marcar) todos los números de su cartón debe imprimir en la pantalla el mensaje “BINGO !!!” y enviar un mensaje al proceso difusor para informarle al resto de procesos jugadores que el juego terminó.

El proceso emisor (locutor)

El proceso emisor o locutor simplemente genera números aleatorios entre 1 y 75 que va enviado periódicamente al proceso difusor. Note que los mensajes deben ser enviados con cierto intervalo de tiempo que permita que todos los procesos jugadores puedan recibirlos y procesarlos. Además, que debe tener especial cuidado de no volver a generar números repetidos, para ello debe llevar una lista de los números que ya fueron “cantados”.

Cada vez que el proceso emisor genere un nuevo número debe imprimir en la consola de lista de todos los números que ha “cantado” previamente.

El proceso difusor

El proceso difusor debe recibir los mensajes provenientes del proceso emisor y difundirlos entre los diferentes procesos jugadores que se estén ejecutando. Este debe ser el primer proceso que se ponga a correr, luego los procesos jugadores y por último el proceso emisor.

Note que adicionalmente existe un mensaje especial que se debe procesar y es el mensaje de “BINGO”. Sin embargo, usted puede codificarlo mediante cualquier otro número (p.ej. 0 ó -1), así el proceso difusor simplemente lo debe pasar y los procesos jugadores saben cómo interpretarlo.

Consideraciones generales

- Toda la programación debe ser realizada en lenguaje Lua y únicamente utilizando la librería *zmq*.
- Debe realizar y documentar pruebas de los programas, utilizando al menos tres procesos jugadores. Sin embargo, solo es necesario mostrar el estado final de los cartones y el proceso emisor; posiblemente mediante capturas de pantalla.
- Se deberá generar una documentación formal, en formato pdf, en donde se describan las diferentes etapas del desarrollo del proyecto, las decisiones de diseño que se tomaron, los mecanismos de programación utilizados, y los resultados de las diferentes pruebas al programa. Dicha documentación deberá incluir al menos las siguientes secciones:
 - Introducción
 - Descripción del problema (este enunciado)
 - Definición de estructuras de datos
 - Descripción detallada y explicación de los componentes principales del programa:
 - Mecanismo de creación y comunicación de procesos
 - Pruebas de ejecución
 - Conclusiones
- Este primer proyecto debe realizarse en forma individual.
- No se permite la copia de código entre estudiantes, tampoco es permitido utilizar librerías adicionales (desarrolladas por terceros).
- Se deberá crear un archivo zip que incluya: el código fuente de todos los programas desarrollados, binarios generados, y documentación en formato pdf.