



Instituto Tecnológico de Buenos Aires

INFORME DEL TP 1

Integrantes:

Hwa Pyoung Kim (62129)

Tomas Ken Kaneko (62297)

Roman Berruti (63533)

Decisiones tomadas

A la hora de empezar el proyecto, discutimos acerca de la estructura de los archivos. Al principio optamos por archivos .c para el master, la vista y el player, pero al contar con varias cosas en el máster (Funciones para la lógica del juego, declaración de las memorias compartidas), se optó por crear un utils.h, utils.c y un sharedMemory.h para poder organizar código y tenerlo más limpio. Dentro de utils creamos la lógica del juego, y dentro de sharedMemory.h definimos la estructura de la sharedMemory.

Para los semáforos, se le decidió poner nombres representativos para poder captar su funcionalidad dentro de los diferentes archivos. Para la sincronización entre el master y player, se decidió nombrar el semáforo A como “semPendingView” y el semáforo B como “semFinishedView”. Estos 2 semáforos se utilizaron para el master y vista, las cuales el máster espera hasta que la vista termine de imprimir el estado del tablero, y recibe un post de parte de la vista para seguir corriendo mientras que la vista recibe un post para imprimir y hace un wait para esperar el siguiente post de parte del máster.

Luego utilizamos los semáforos del C hasta el E con la variable F para sincronizar el master del player. Llamamos al semáforo C como “writerPrivilege”, al D como “masterPlayerMutex”, al E como “masterPlayerMutex” y a la variable F como “playersReadingCount”. El semáforo C se utilizó para evitar la inanición al acceder al estado y frenar las solicitudes de movimientos por parte de los jugadores , para que una vez finalizada su lectura, el master pueda procesar las mismas y pedir a la vista imprimir el estado del tablero. El semáforo D se utilizó como un Mutex para el master para frenar los procesos de solicitudes de movimientos y la modificación del estado de juego. El semáforo E se utilizó para permitir al master procesar procesar, la cual se hace un wait si hay players leyendo del estado y post si todos los players terminaron de leer del estado. Con estos métodos se evita el deadlock del Master, de los players y de la vista.

Instrucciones de compilación y ejecución

Las instrucciones de tanto compilación como la de ejecución se pueden encontrar en el README.md de nuestro repositorio de github: [Link al github](#).

Limitaciones

Nuestro proyecto tiene ciertas limitaciones:

- El nombre de los parámetros de players y vista no puede superar los 19 caracteres
- Intentamos utilizar un jugador más inteligente mediante un getpid y una función findPlayerIndex. Se nos complicó a la hora de crear un findBestMove (función la cual encuentra el mejor movimiento), la cual retornaba movimientos no deseados.

Problemas y soluciones

Durante la implementación de nuestro proyecto, nos encontramos con diferentes problemas.

- Tuvimos problemas con el desarrollo de los semáforos (en especial con el de los jugadores y master), los cuales generaban deadlocks tanto en la vista como en los jugadores. Investigamos más a profundidad el propósito de cada semáforo del A hasta la E, y al ponerle nombres acorde a sus funcionalidades tales, como “writerPrivilege” o “masterPlayerMutex”, pudimos deshacernos de los deadlocks y seguir el orden de los procesos de forma más simple.
- El desarrollo del juego fue también un obstáculo ya que se nos fue difícil poder contemplar todos los casos posibles que se pueden dar dentro del tablero, pero luego de probar y pensar pudimos modularizar funcionalidades y así tener un código más simple.
- Hubo varias trabas más en el proyecto que pudimos resolver mediante las clases virtuales y el foro, con preguntas que hicieron

otros compañeros y nuestro grupo, las cuales aclararon nuestras dudas con respecto a poner en práctica lo aprendido.

Citas de fragmentos de código reutilizados de otras fuentes

Para nuestro proyecto, investigamos el manual de Linux, man, para entender el funcionamiento. También utilizamos estructuras de la utilización de semáforos, creación de memorias compartidas y creación de procesos hijos que se dieron en las clases virtuales y presenciales. En cuanto a la vista, utilizamos la [Documentación de ncurses](#) para mostrar el tablero con los colores respectivos (vimos en los anuncios que se podía utilizar esta librería). Utilizamos ChatGPT para el posicionamiento de los jugadores para que los jugadores no empiecen con una ventaja por sobre el otro.