

MEMORIA PROYECTO FINAL - SOPER

- Grupo: 2213
- Pareja: 09

Los distintos archivos se encuentran en la carpeta de “proyecto” dentro del archivo .zip y esta memoria, junto con el fichero de información pedido de la pareja y la documentación generada con Doxygen (carpetas “html” y “latex”), se encuentra en la carpeta “doc”. En la carpeta “src” se encuentra todos los ficheros .c y .h necesarios para el compilado y funcionamiento correctos de la práctica (se puede realizar make desde “src” para compilarlo, pero también desde “proyecto”). En la carpeta “target” se encuentran los distintos ejecutables del proyecto (se puede realizar “./make.sh” para compilarlo desde esa carpeta).

Para que se lleve a cabo el buen funcionamiento de la simulación, primero se debe ejecutar, o bien el monitor o bien el simulador, para que el monitor empiece a imprimir por pantalla el estado del mapa tras la inicialización y las modificaciones de este realizadas en el simulador.

En cuanto a los ficheros .c y .h, hemos creado “nave.c” y “jefe.c” para sus correspondientes tipos de procesos en la simulación. Además, hemos creado “manejadores.c” y “manejadores.h” con varias funciones necesarias en el proyecto como localizar naves enemigas, ver cuando hay un ganador de la partida... y los manejadores utilizados para la simulación. También se ha modificado el código del archivo del monitor para que realice su función.

Diseño:

En la página siguiente se encuentra el esquema de procesos y elementos compartidos utilizados en nuestro proyecto.

Básicamente, toda la simulación gira en torno al simulador, el proceso que inicializa los recursos compartidos y crea los procesos jefes. Estos, a su vez, crearán los procesos naves. (Flechas)

El monitor es un proceso apartado, pero que comparte con el simulador el mapa.

El mapa se encuentra en memoria compartida. Esta la inicializa y crea el proceso simulador, y tras esto, el monitor la abre y utiliza para imprimir por pantalla su estado. Además, las naves la abren también ya que necesitan saber su información en cierto momento de la simulación. (Líneas verdes)

Para el envío de instrucciones entre los distintos procesos tenemos varios recursos. Primero, para enviar las instrucciones entre el simulador y los jefes, y a su vez, los jefes con sus respectivas naves utilizamos tuberías, que se crean en el proceso padre

correspondiente (almacenándose en un array los descriptores de escritura en su proceso padre y en los hijos se efectúa un dup2 cambiando STDIN_FILENO por el descriptor de lectura de la tubería). Segundo, creamos una cola de mensajes en el proceso simulador que será la encargada de gestionar el envío de mensajes entre las naves y el simulador, que contendrán la instrucción indicada e información de las naves para la gestión de las acciones. (Líneas rojas y moradas)

En el diseño posterior no los hemos incluido por temas de espacio, pero hay varios semáforos que se encargan de la sincronización entre los procesos en el envío de instrucciones y mensajes, además de uno encargado de que el monitor espere al simulador.

