

Documentación

Proyecto 1

Principios de Sistemas Operativos

Prof. Kevin Moraga

Julio Rojas 2015090806

Jose Paulo Yock 2015018215

Introducción:

En este proyecto, se nos pide implementar la biblioteca pthreads, con el nombre de my_thread, además de esto, hay que implementar tres tipos de schedulers (RoundRobin, sorteo, tiempo real).

Utilizando esta biblioteca, se debe crear un tecRoyale, el cuál es un juego basado en Clash Royale pero para terminal. El juego consta de dos pantallas, con la mitad de la arena en cada pantalla. Deben de haber al menos diez guerreros, con stats como tipo, nivel, ataque, vida. Para generar los guerreros de la mano, se debe utilizar un algoritmo genético.

Ambiente de desarrollo:

Para realizar este proyecto se utilizó el lenguaje de programación C. Entre las estructuras de datos más importantes se encuentra un arreglo para almacenar los hilos, un arreglo de booleanos para saber el estado de cada hilo, un arreglo para saber la prioridad de los hilos y un último para conocer la cantidad de tickets que tiene cada hilo.

Entre las funciones se encuentran:

- my_thread_create: La cual recibe como parámetros, función a ejecutar, parámetros necesarios, tickets iniciales y prioridad. Esta función crea un hilo y lo agrega a la cola del scheduler.
- my_thread_end: La cual se encarga de finalizar un hilo.
- my_thread_yield: Esta función activa el scheduler y cede el procesador.
- run_threads: Esta función activa el scheduler y pone a correr los hilos.

- `my_mutex_init`: Esta función inicializa el mutex, recibe como parámetro el puntero a la variable mutex.
- `my_mutex_lock`: Esta función recibe como parámetro el mutex a bloquear y se encarga de bloquear el recurso siguiente en el texto.
- `my_mutex_destroy`: Esta función finaliza el mutex.
- `my_mutex_unlock`: Esta función desbloquea el recurso y lo coloca como disponible.
- `my_mutex_trylock`: Esta función bloquea el recurso, pero si este no está disponible no se bloquea.
- `my_thread_chsched`: Función que se encarga de cambiar el scheduler activo.
- `gameControler`: Hilo del juego el cual se encarga de inicializar el tablero, y es quien se encarga de imprimir en pantalla el tablero.
- `netControler`: Hilo del juego el cual administra los mensajes y ejecuta las funciones encargadas.
- `input`: Hilo que maneja las entradas del usuario para interactuar con la interfaz.

Instrucciones para ejecutar el programa:

Para ejecutar el código se presenta un archivo makefile, por lo que se debe ejecutar:

```
$make all
$make run
```

O también, después de hacer el primer comando, se puede iniciar con:

```
$./tecRoyale <archivo ini>
```

Actividades realizadas por estudiante:

Julio Rojas:

- 24 de septiembre: Se comienza a leer sobre la implementación de los hilos. Horas: 7
- 25 de septiembre: Se hacen pruebas con `setjmp` y `longjmp`. Horas: 4

- 30 de septiembre: Se Hacen pruebas con context. Horas: 4
- 30 de septiembre: Se hacen pruebas con Clone. Horas: 4
- 1 de octubre: Se realiza un código base con context. Horas: 3
- 2 de octubre: Se agregan las primeras funciones de la biblioteca. Horas: 6
- 6 de octubre: Se investiga funcionamiento del scheduler. Horas: 4
- 7 de octubre: Se realiza el scheduler RoundRobin. Horas: 4
- 8 de octubre: Se realizan los schedulers de tiempo real y sorteo. Horas: 7
- 9 de octubre: Se investiga sobre el funcionamiento de los mutex. Horas: 6
- 12 de octubre: Se implementan los mutex. Horas: 5
- 13 de octubre: Se resuelven errores en el código. Horas: 3
- 13 de octubre: Se implementa algoritmo genético del juego. Horas: 3
- 14 y 15 de octubre: Haciendo debug al código buscando un error de memoria. Horas: 10
- 16 de octubre: Se realiza de nuevo el algoritmo genético y se corrige el error. Horas: 2
- 16 de octubre: Se incorpora el código del juego con los hilos propios. Horas: 2

Horas totales: 79

Jose Paulo Yock:

- 28 de septiembre 9:00pm - 10:00pm: Horas: 1
 - Pensar en el flujo del programa.
 - Buscar informacion sobre el manejo de MPI.
- 29 de septiembre 11:00am - 2:40pm: Horas: 2.4
 - Se continua con la investigacion de como funciona la biblioteca MPI para conectar dos computadoras.
 - Se retorna complejo como funciona la conexión.
- 1 de octubre 10:30am - 6:35pm: Horas: 7
 - Se comienza con la conexión de sockets.
 - Se investiga sobre como utilizar select para realizar un read() sin que se quede esperando.

- Se crea un programa el cual funciona como chat para pasar información de una pc a otra.
 - Pensar de como va a ser el manejo de mensajes.
 - No se envían los mensajes, pero si se realiza la conexión.
- 1 de octubre 9:00am - 10:30pm: Horas: 1.3
 - Se descubre el error el cual se presenta en el buffer de stdin, por lo que se prueba la conexión con mensajes estáticos.
- 2 de octubre 2:00pm - 4:00pm: Horas: 2 -Diseñar el juego, como se manejan los hilos y el paso de mensajes.
- 5 de octubre 12:00pm - 9:00pm: Horas: 9
 - Iniciar con el ciclo del juego, hilos.
 - Se investiga sobre como leer sin esperar al enter.
- 6 de octubre 3:00pm - 7:30pm: Horas: 4.3
 - Inicio con el manejo de la baraja.
 - Selección de carta.
 - Estructura para crear nuevo hilo.
- 7 de octubre 1:30pm - 3:10pm: Horas: 2
 - Reestructurar el tablero.
 - Continuar con el manejo de las unidades.
- 7 de octubre 4:00pm - 5:30pm: Horas: 1.3
 - Continuar con el manejo de los hilos.
- 8 de octubre 3:15pm - 6:00pm: Horas: 2.4
 - Continuar con el manejo de hilos.
 - Hilos creados, se continua con el movimiento.
 - Se completan los movimientos.
 - Se presenta un error con el movimiento cuando el guerrero se encuentra después de las torres.
- 8 de octubre 7:00pm - 9:00pm: Horas: 2
 - Se arregla el movimiento.
 - Se comienza como la creación del guerrero por red.
- 12 de octubre 7:00pm - 11:00pm: Horas: 4
 - Se crean hilos en la otra pc.
 - Se corrigen errores en el movimiento.
 - Se implementa el ataque.
- 14 de octubre 2:00pm - 7:30pm: Horas: 5.3
 - Ocurre un error con la muestra de las cartas cuando se implementa el algoritmo genético, se rastrea el error a esa función así que se deja comentada para que sea corregido.

- 15 de octubre 6:30pm - 9:30pm: Horas: 3
 - Se trabaja en la finalización del juego.
 - Se modifica el manejo de las torres para finalizar el juego.
 - Se ataja la finalización en la otra pc.
 - Se muestra el mensaje de perdedor o ganador.
- 16 de octubre 1:00pm - 4:35pm: Horas: 3.3
 - Se trabaja en el archivo ini del juego.
 - Se implementa este archivo para inicializar los guerreros utilizados por el algoritmo genético.

Horas totales: 51

Comentarios finales (estado del programa):

El proyecto se logra finalizar casi en su totalidad, con excepciones en algunos casos. - No se implementan los métodos: `my_thread_join` y `detach`, principalmente por que no son necesarios para la implementación del juego.

- Falta de implementar los semáforos en el paso de pantallas.
- Falta el personaje tipo bomba.

Conclusiones y Recomendaciones del proyecto.

En este proyecto, la parte de la implementación de los hilos fue muy pesada, por lo que se recomienda empezar por ahí. El tema es complicado para encontrar ejemplos sencillos, por lo que se recomienda tener paciencia, leer mucho, ver muchos ejemplos y tratar de entender todos estos.

Bibliografía

- Multitasking using `setjmp`, 1. (2017). Multitasking using `setjmp`, `longjmp`. Stackoverflow.com. 17 de octubre 2017, obtenido de <https://stackoverflow.com/questions/2560792/multitasking-using-setjmp-longjmp>
- `mfiber`: Simple User Level Thread Library.. (2017). LID: Lost In Droid. 17 de octubre 2017, obtenido de <https://machiry.wordpress.com/2012/01/02/mfiber-simple-user-land-thread-library/>
- Implementing a Thread Library on Linux (evanjones.ca). (2017). Evan-jones.ca. 17 de octubre 2017, obtenido de <http://www.evanjones.ca/software/threading.html>

- Implementing a Thread Library on Linux (evanjones.ca). (2017). Evan-jones.ca. 17 de octubre 2017, obtenido de <http://www.evanjones.ca/software/threading.html>
- INI file. (2017, October 11). In Wikipedia, The Free Encyclopedia. 17 de octubre 2017, obtenido de https://en.wikipedia.org/w/index.php?title=INI_file&oldid=804815470
- Manual de linux:

```
$man longjmp  
$man context  
$man sched.h
```

- GitHub:
 - <https://github.com/brianwatling/libfiber>
 - <https://gist.github.com/DanGe42/7148946>