

INSTITUTO TECNOLÓGICO DE COSTA RICA

SISTEMAS OPERATIVOS

# Proyecto I

## TEC - Royale

*Jean Marco Rojas*

*2015040717*

*Jonathan Guillén*

*201261579*

*Jose Daniel Peñaranda*

*2013039845*

Profesor  
Sr. K. MORAGA

Junio 01, 2021

# 1 Introduction

Es posible realizar la implementación de la biblioteca de pthreads en el espacio de usuario, esto permite al estudiante comprender cómo se puede programar un subsistema del sistema operativo, sin tener que hacer cambios en el kernel. Para ello se reimplementará pthread y se probará utilizando un programa que utiliza la biblioteca creada por el estudiante. Se realizará la re-implementación de la biblioteca de pthreads, llamada mypthread, de las siguientes funciones:

- my\_thread\_create
- my\_thread\_end
- my\_thread\_yield
- my\_thread\_join
- my\_thread\_detach
- my\_mutex\_init
- my\_mutex\_destroy
- my\_mutex\_lock
- my\_mutex\_unlock
- my\_mutex\_trylock
- my\_thread\_chsched:

# 2 Ambiente de desarrollo:

Cada miembro del equipo trabajara desde su propia máquina para la realización del código necesario para el proyecto, el problema presentado se dividirá en asignaciones y cada miembro seleccionará de cuales se hará cargo distribuyendo equitativamente las cargas de trabajo, esto se hará mediante sprints semanales, que nos permite dividir y observar el trabajo realizado por los integrantes del grupo, cada vez que se dé por terminada una asignación o parte de la misma se debe subir el trabajo realizado a Github de manera tal que permita el acceso a dicha información a los demás miembros del grupo, esto con el fin de tener una integración continua con versiones del trabajo realizado, poder realizar revisiones y comentar sobre el código realizado, además al final de cada sprint se hace una autoevaluación para detectar las debilidades del grupo de trabajo y así saber en qué mejorar. A continuación se detallan las herramientas utilizadas para la realización de esta tarea:

**C:** En el caso de este lenguaje se utilizará el compilador gcc y se programara utilizando Sublime text 3, Clion, VScode y Netbeans 8.2.

**Latex:** Este editor de texto será la herramienta principal para la realización de los documentos del kick off y la documentación final de la tarea.

**GitHub:** Como sistema de control de versiones.

**Google Chrome:** Como navegador web y donde se correrá el servidor, en su versión 89.0.4389.114.

**Ubuntu:** Como sistema operativo donde será desarrollada la tarea, en su versión 20.04.2.0.

### 3 Estructuras de datos usadas y funciones

**mythreads:** tiene las funciones antes mencioandas, igual a la de la biblioteca original de pthreads.

**cruzar\_puente():** Envía a un hilo a cruzar el puente, con el algoritmo de calendarización que se definió.

**cruzar\_puente\_semaforo():** Envía a un hilo a cruzar el puente, haciendo uso de un semaforo.

**createUnit():** Crea un guerrero tomando en cuenta un numero de salud y de ataque definido al momento de crearlo.

**createPoblation():** Crea una lista de guerreros, una poblacion de guerreros, esto para iniciar el programa, ya que crea la primer generacion de guerreros los cuales como son todos los mismo se pasa el tipo de guerrero por parametro.

**insertUnitList():** Inserta un guerrero al principio de una lista de guerreros.

**selectionUnits():** Selecciona a los mejores guerreros de la población actual para que estos puedan crear a la siguiente generación de guerreros.

**crossover():** se hace la mezcla de características, o genes, de cada guerrero para generar uno nuevo.

**mutation():** Mutacion que tiene cada unidad nueva al momento de crearla.

**mating():** Proceso de reproducción de los mejores guerreros seleccionados, los cuales se cruzan para tener los valores de los padres y tener su variacion por la mutacion.

**genetics():** Funcion principal para los algoritmos geneticos, recibe una poblacion y el numero de esta y va seleccionando a los mejores para reproducirlos y tener una nueva generacion.

### 4 Instrucciones para ejecutar el programa

Para ejecutar el programa se debe tener instalado los lenguajes en los que esta realizado este lenguaje, estos serian C. Para compilar el proyecto escribimos en la terminal (ubicada en la carpeta del proyecto) “gcc -Wall main.c MyPThreads.c DataStructures.c -o test” y “./prueba”

## 5 Actividades realizadas

- 1 de mayo 2021 (Daniel, Jean Marco, Jonathan): Buscar información sobre el funcionamiento de pthreads en c (6 horas).
- 2 de mayo 2021 (Jean Marco): Creación de sockets para jugar en diferentes pantallas (terminales) (4 horas).
- 2 de mayo 2021 (Jonathan): Buscar información sobre el funcionamiento de algoritmos genéticos (4 horas).
- 3 de mayo 2021 (Jonathan): Buscar información sobre el funcionamiento de algoritmos genéticos en c (6 horas).
- 15 de mayo 2021 (Jean Marco): Creación de funciones de mypthread (4 horas).
- 16 de mayo 2021 (Jonathan): Creación de funciones de los guerreros y el manejo de los mismos(4 horas).
- 16 de mayo 2021 (Jean Marco): Creación de funciones de mypthread (2 horas).
- 20 de mayo 2021 (Jonathan): Creación de funciones de algoritmos genéticos de selección y cálculo fitness con los guerreros (6 horas).
- 24 de mayo 2021 (Jonathan): Creación de funciones de algoritmos genéticos de cruce y mutación con los guerreros (6 horas).
- 28 de mayo 2021 (Jonathan): Creación de funciones de algoritmos genéticos de reproducción de los guerreros (4 horas).
- 29 de mayo 2021 (Jean Marco): Creación de semáforos (1 hora).
- 30 de mayo 2021 (Jonathan): Creación de funciones de algoritmos genéticos general (6 horas).
- 30 de mayo 2021 (Jean Marco): Creación de algoritmos RR y Lottery (6 horas).
- 02 de junio 2021 (Jean Marco): Escribir la documentación (2 horas).
- 01 de junio 2021 (Jonathan): Afinación y corrección de errores. (2 horas).

## 6 Autoevaluación

En el transcurso del proyecto se logró la realización de la biblioteca mypthread, a excepción de dos funciones del mutex, “my\_mutex\_destroy” y “my\_mutex\_trylock”, los algoritmos de calendarización funcionan por separado pero no se pudieron integrar, con la librería de mypthread, a excepción del algoritmo de tiempo real,

del todo no se pudo crear y usa el por defecto de la biblioteca mypthread, se muestra en diferentes pantallas usando un sistema de sockets y el canvas con el juego funciona casi completo. Se logró realizar la implementación de los algoritmos genéticos para la generación de nuevos guerreros, cada vez que los que estén en juego hayan muerto.

## 6.1 Evaluación

- mypthread: Completado
- Algoritmo RR: Completado
- Algoritmo RR: Semi completado
- Algoritmo tiempo real: No completado
- Cambio de contexto.: No completado
- Múltiples pantallas: Completado
- Canvas: semi completado
- Documentación: Completado
- Kick-off: Completado

## 6.2 Autoevaluación

Jean Marco:

- [5] Aprendizaje de pthreads.
- [5] Aprendizaje de Round Robin.
- [2] Aprendizaje de Tiempo Real.
- [3] Aprendizaje de Cambio de contexto.
- [5] Aprendizaje de Sorteo.

Jonathan:

- [4] Aprendizaje de pthreads.
- [5] Aprendizaje de Round Robin.
- [3] Aprendizaje de Tiempo Real.
- [3] Aprendizaje de Cambio de contexto.
- [5] Aprendizaje de Sorteo.

## 7 Lecciones Aprendidas y Recomendaciones

- Investigar y leer la documentación de pthread, a su vez de ver tutoriales en youtube.
- Funcionamiento de los sockets a profundidad.

- Investigar sobre cómo crear su propia biblioteca.
- Organizarse desde el inicio, por la magnitud del proyecto.
- Investigar sobre los algoritmos genéticos y varias de sus implementaciones.

## 8 Bibliografía

- Multithreading in C - GeeksforGeeks. (2014). Retrieved 3 June 2021, from <https://www.geeksforgeeks.org/multithreading-c-2/>
- Ippolito, G. (2021). Linux Tutorial: POSIX Threads. Retrieved 3 June 2021, from <https://www.cs.cmu.edu/afs/cs/academic/class/15492-f07/www/pthreads.html>
- Sockets en C de Unix/Linux. (2021). Retrieved 3 June 2021, from [http://www.chuidiang.org/clinix/sockets/sockets\\_simp.php](http://www.chuidiang.org/clinix/sockets/sockets_simp.php)
- pthread.h. (2021). Retrieved 3 June 2021, from <https://pubs.opengroup.org/onlinepubs/7908799/xsh/ptl>
- Genetic algorithm - Wikipedia. (2011). Retrieved 3 June 2021, from [https://en.wikipedia.org/wiki/Genetic\\_algorithm](https://en.wikipedia.org/wiki/Genetic_algorithm)
- Hackenberger, B. (2019). Genetics without genes: application of genetic algorithms in medicine. Croatian Medical Journal, 60(2), 177-180. doi: 10.3325/cmj.2019.60.177
- Introduction to Optimization with Genetic Algorithm. (2018). Retrieved 3 June 2021, from <https://towardsdatascience.com/introduction-to-optimization-with-genetic-algorithm-2f5001d9964b>