



Instituto Tecnológico de Costa Rica
Escuela de ingeniería en computación

CURSO:
IC6600 - Principios de Sistemas Operativos GR 20

Proyecto 1

PROFESOR:
MSc. Kevin Moraga

ESTUDIANTES:
Renzo Giuliano Barra Mostajo 2019003700
Joan Alberto Sánchez Chinchilla 20151238670

22 de mayo del 2022

1 Introducción

Se plantea que debe hacerse una simulación de una ciudad usando hilos. Dentro de esta ciudad hay carros, ambulancias, barcos, puentes y plantas nucleares. Adicionalmente se debe trabajar y crear varios scheduler que permita gestionar los threads. Debe haber 3 implementaciones entre las que se incluyen: Round robin, sorteo y de tiempo real. Se menciona que para realizar esto se debe crear una biblioteca de hilos personalizada con todos sus métodos respectivos.

Los métodos mencionados son:

- `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`

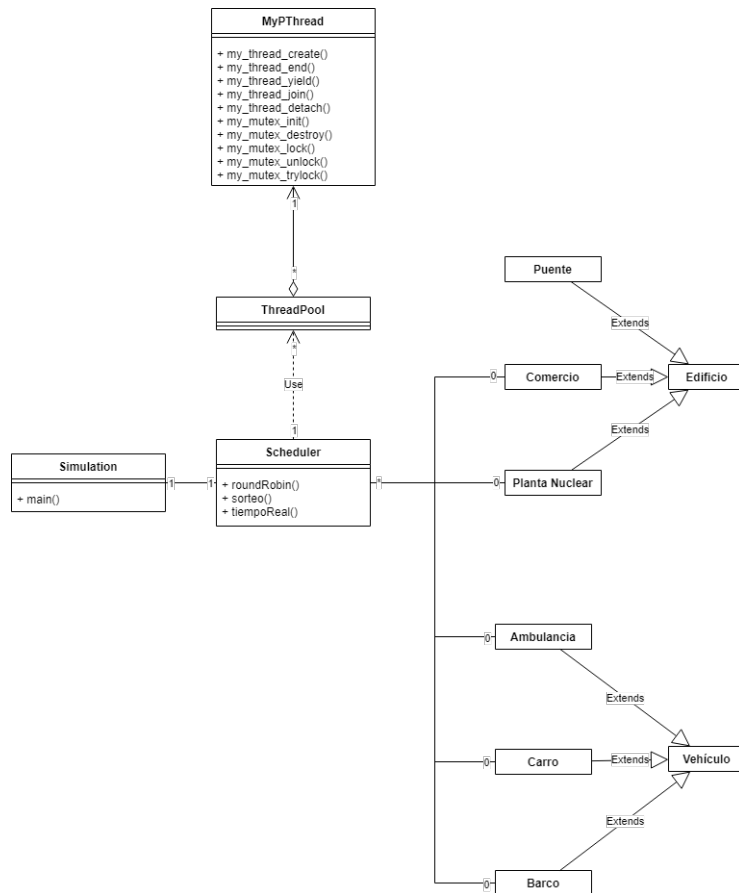
2 Ambiente de desarrollo

Para control de versiones se utilizará GitHub como repositorio, la plataforma donde se trabajará será en Visual Studio Code, en Linux Ubuntu como sistema operativo. Además deberá ser programado utilizando el lenguaje de programación Rust. Esto debido a que es especificado como requerimiento del proyecto y además caso contrario pues no se evalúa.

3 Control de versiones

A continuación se adjunta el link al repositorio que se utilizará para este trabajo:
https://github.com/Edgy190/proyecto1_SO

4 Diagrama UML



5 Estructuras de datos usadas y funciones

- Thread city
 - Simple matrix: Biblioteca de matriz para manejar la estructura básica de la ciudad.
 - Vectores: Para almacenar coordenadas de inicio y fin del camino. Además, guardan la ruta a seguir dados los puntos anteriores.
- Métodos Thread city implementados
 - create_city_matrix: Crea el layout de la ciudad.
 - find_path: Encuentra el camino más corto para mobilizar la ciudad.
 - random_coordinates: Devuelve coordenadas aleatorias que funcionan como punto de inicio y fin del recorrido de un vehículo.
 - add_ship_to_city: Devuelve la unica ruta posible que pueden tener los los barcos dentro de la ciudad partiendo de que navegan a traves del puente 1.

- `add_car_to_city`: Devuelve una ruta posible para un vehiculo entre unas coordenadas de inicio y otras de final.
- `follow_path`: Se encarga de manejar el posible recorrido por un vehiculo o barco dentro de la ciudad. Toma en cuenta distintas acciones dependiendo de la zona en la que se encuentre actualmente, por ejemplo los semaforos, el ceda, el puente elevadizo, etc.

- `My_PThread`

- `pub struct MyPThread`: Definición de variables de contexto, arreglos numéricos, y números para mantener control sobre los hilos.
- `pub signal_context: ucontext_t`
- `pub threads: (vec![number_threads], ucontext_t)`
- `pub current_thread: ucontext_t`
- `pub exit_context: ucontext_t`
- `pub priority: vec![number_threads]`
- `pub priority_aux: vec![number_threads]`
- `pub tickets: vec![number_threads]`
- `pub dead_threads: vec![number_threads]`
- `pub curr_context: u32`
- `pub init: u32`
- `pub active_threads: u32`
- `pub active_threads_aux: u32`
- `pub total_tickets: u32`
- `pub active_scheduler: u32`

- Métodos `My_PThread` implementados

- `my_thread_create`: Crea un hilo, se le pasa una función, parámetros, prioridad y tickets disponibles para el scheduler.
- `my_thread_end`: Interrumpe el hilo, agrega el hilo a `dead_threads`, le quita los tickets y los hilos auxiliares activos.
- `my_thread_yield`: Interrumpe al hilo para escoger un scheduler.
- `my_thread_chsched`: Cambia el scheduler de un hilo.
- `roundRobin`: Algoritmo de scheduler del Round Robin.
- `sorteo`: Algoritmo de scheduler del Sorteo.
- `real`: Algoritmo de scheduler del tiempo real.
- `interrupt`: Interrumpe el proceso de un hilo.

6 Instrucciones para ejecutar el programa

Para la ejecución del programa deberá clonar el repositorio de github o bien contar con el archivo .zip que contiene los archivos ejecutables del programa. Luego, acceda al directorio donde se encuentran dichos archivos y abra una nueva terminal. Navegue hasta la carpeta thread_ville con el comando 'cd' de linux y ejecute el siguiente comando:

```
~/Documents/proyecto1_S0/thread_ville$ cargo run
```

Acto seguido deberá ver en consola la siguiente salida del programa donde se ejemplifica un recorrido por un carro dentro de la ciudad.

```
Inicio del camino: [5, 4]
Fin del camino: [0, 4]
Camino seguido: [[5, 4], [4, 4], [3, 4], [2, 4], [1, 4], [0, 4]]

0 1 0 0 2 1
0 1 0 0 2 0
0 0 0 1 2 0
" "
0 1 0 1 2 0
" "
0 1 0 0 2 0
0 0 0 1 2 0
0 1 0 0 0 1
```

Nota: Debe tener instalado el lenguaje de programación rust en sus sistema operativo, así como también 'cargo' que debería instalarse por defecto junto a rust.

7 Actividades realizadas por el estudiante

Cronograma de actividades de proyecto			
Fecha	Reporte de actividades	Estudiantes	Tiempo invertido en horas individual
13-Apr-22	Inicio del proyecto, elaboración del kick-off. Investigación, planteamiento del problema y estrategia para su solución	Renzo Barra, Joan Sánchez	4
22-Apr-22	Entrega del kick off considerando tiempo invertido desde la actividad anterior en investigación	Renzo Barra, Joan Sánchez	36
9-May-22	Distribución de la programación, Renzo los hilos, scheduler y mutex. Joan la ciudad y sus componentes	Renzo Barra, Joan Sánchez	1
13-May-22	Reporte de tiempo invertido en programación para cada parte desde la actividad anterior, involucró mucha investigación	Renzo Barra, Joan Sánchez	12
17-May-22	Reporte de avances notorios en la ciudad (8 horas), problemas con la biblioteca de threads, schedulers y mutex	Renzo Barra, Joan Sánchez	9
21-May-22	Se concluyen los schedulers sin probar debido a problemas con los hilos. Se implementan algunas funciones de hilos, quedan pendiente el mutex	Renzo Barra, Joan Sánchez	8
22-May-22	Conclusión de programación de ambas partes, intento de conectar ambas partes. Concluye la elaboración de la documentación	Renzo Barra, Joan Sánchez	12
Total			164
Nota: La sección de horas muestra la cantidad de horas invertidas individualmente según un acumulativo entre la actividad anterior y el reporte de la actividad			

8 Autoevaluación

Donde se evalúa en una escala [1] Muy Malo, [2] Malo, [3] Regular, [4] Bueno y [5] Muy bueno, de los siguientes rubros y para cada estudiantes miembro del proyecto:

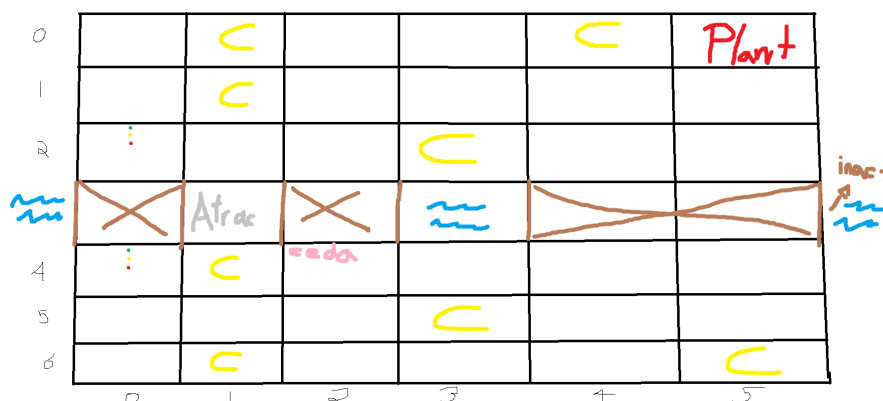
- Renzo Barra
 - Aprendizaje de RR: [5]
 - Aprendizaje de Tiempo Real: [2]
 - Aprendizaje de Lottery: [5]
 - Aprendizaje de pthreads: [3]
- Joan Sánchez
 - Aprendizaje de RR: [5]
 - Aprendizaje de Tiempo Real: [3]
 - Aprendizaje de Lottery: [5]
 - Aprendizaje de pthreads: [3]

9 Lecciones aprendidas

- Realizar una ciudad para que funcione sobre hilos
- Teoría de hilos y su implementación.
- Teoría de mutex
- Teoría de schedulers y la implementación de diversos algoritmos
- Conocimientos y uso de Rust

Durante el desarrollo del kick-off el enfoque fue la teoría de cómo funcionan las cosas del proyecto, como los hilos, el mutex, los schedulers y la formulación de una ciudad que funcionaría con estos conceptos. Gran parte del aprendizaje respecto a rust se hizo por medio de la implementación y traducción de código fuente en C para hilos de otras bibliotecas ya implementadas. Con esto nos referimos al funcionamiento de métodos y estructura de PThread para crear My_PThread. Para los hilos se utilizó la biblioteca `ucontext_t` con el fin de darle a los hilos un contexto y mantenerlos por así decirlo a la vista para su manipulación. Los schedulers se implementaron basados en su teoría y algoritmos de los mismos encontrados en internet y recursos en pdfs. Para esto se utilizan funciones para obtener y setear el contexto de los hilos cuando ingresan al scheduler que les corresponda, adicionalmente para el scheduler de sorteo se utiliza `rand` con el fin de cumplir aleatoriedad en dicho sorteo.

Solución propuesta para la ciudad para su algoritmo:



10 Bibliografía

bhaargav006, “BHAARGAV006/user-thread-library: User-level Thread Library which mimics the interface of the pthread library and runs in user space,” GitHub. [Online]. Available: <https://github.com/bhaargav006/User-Thread-Library>. [Accessed: 22-May-2022].

Comprendamos, “El concepto de planificador para tareas en tiempo real Concurrentes,” El concepto de planificador para tareas en tiempo real concurrentes Comprendamos. [Online]. Available: <https://www.comprendamos.org/alephzero/57/planificador.html>. [Accessed: 22-May-2022].

Implementing a thread library on linux (evanjones.ca). [Online]. Available: <https://www.evanjones.ca/software/threading.html>. [Accessed: 22-May-2022].

L. Williams, “Round Robin scheduling algorithm with example,” Guru99, 09-Apr-2022. [Online]. Available: <https://www.guru99.com/round-robin-scheduling-example.html>. [Accessed: 22-May-2022].

“Lottery scheduling,” MyCareerwise. [Online]. Available: <https://mycareerwise.com/content/lottery-scheduling/> content/exam/gate/computer-science. [Accessed: 22-May-2022].

“Program for round robin scheduling: Set 1,” GeeksforGeeks, 04-May-2022. [Online]. Available: <https://www.geeksforgeeks.org/program-round-robin-scheduling-set-1/>. [Accessed: 22-May-2022].

R. Bikram, "Lottery scheduling in operating system," Bench Partner, 19-Feb-2022. [Online]. Available: <https://benchpartner.com/lottery-scheduling-in-operating-system>. [Accessed: 22-May-2022].

R. Bikram, "Mutexes in operating system," Bench Partner, 06-Feb-2022. [Online]. Available: <https://benchpartner.com/mutexes-in-operating-system>. [Accessed: 22-May-2022].

R. Bikram, “Round robin scheduling,” Bench Partner, 01-Feb-2022. [Online]. Available: <https://benchpartner.com/round-robin-scheduling>. [Accessed: 22-May-2022].

R. Bikram, “Threads implementation,” Bench Partner, 01-Feb-2022. [Online]. Available: <https://benchpartner.com/threads-implementation>. [Accessed: 22-May-2022].

“Round robin scheduling,” Studytonight.com. [Online]. Available: <https://www.studytonight.com/operating-system/round-robin-scheduling>. [Accessed: 22-May-2022].

“Scheduling Para Sistemas de tiempo real - conceptos de sistemas operativos,” 1Library.Co. [Online]. Available: <https://1library.co/article/scheduling-sistemas-tiempo-real-conceptos-sistemas-operativos.q29dv6ez>. [Accessed: 22-May-2022].

“`[-][src]struct libc::ucontext_t,`” `libc::ucontext_t` - Rust. [Online]. Available: https://docs.rs/libc/0.2.58/libc/src/struct.ucontext_t.html. [Accessed: 22-May-2022].