Proyecto 1: Scheduling Ships

Sebastián Alba Vives Nicol Otárola Porras Noel Castro Acosta Karina Martínez Guerrero Kevin Lobo Juárez
Área Académica de Ingeniería en Computadores
Instituto Tecnológico de Costa Rica

***Abstract*—This project focuses on the development of the CEthreads library, which reimplements the functionalities of Pthreads, including thread creation, management, and mutex handling. The objective is to provide efficient thread management for scheduling tasks according to various algorithms such as Round Robin (RR), Shortest Job First (SJF), Priority, First-Come-First-Served (FCFS), and Real-Time Scheduling. These scheduling techniques are applied to simulate the transit of ships through a canal, where each ship is represented as a separate thread.**

The project simulates a channel where ships, categorized as Normal, Fishing, or Patrol, transit from one ocean to another. The canal allows passage in only one direction at a time, and scheduling algorithms are used to determine the order in which the ships cross. The simulation includes flow control mechanisms such as Equity, Sign-Based, and a free-flow mode referred to as "Tico."

A prototype hardware component is integrated to visually represent the ships waiting to enter the canal, the direction of the flow, and the type of ship currently transiting. This is achieved using LED indicators and other electronic components. The hardware representation ensures that the traffic flow through the canal is displayed, and no collisions occur between ships moving in opposite directions.

Through this project, we aim to enhance the understanding of thread management and scheduling in operating systems while implementing a tangible hardware model that interacts with the scheduling algorithms.

***Palabras clave*—Canal de barcos, CEthreads, hilos, scheduling.**

I. Introducción

El enfoque del proyecto es la creación de la biblioteca CEthreads, la cual replica funciones importantes de Pthreads, como la creación y control de hilos y el uso de mutex. También, implica desarrollar un sistema de gestión y programación de barcos utilizando hilos y algoritmos de planificación. El sistema recrea el paso de embarcaciones por un canal que solo permite el tráfico en una sola dirección a la vez. Diversos algoritmos de programación de hilos han sido utilizados, como Round Robin (RR), Shortest Job First (SJF), Prioridad, First-Come-First-Served (FCFS) y Real-Time Scheduling, para mejorar la gestión del tráfico en el canal.

El proyecto utiliza conceptos de sistemas operativos como la creación y gestión de hilos, la sincronización de recursos con mutex y varios algoritmos de planificación de procesos. Se requiere comprender la forma en que los sistemas operativos distribuyen y administran los recursos entre procesos o hilos para garantizar una ejecución eficiente y sin problemas.

Además, cabe mencionar que el objetivo del proyecto es crear un simulador que pueda controlar el tráfico de barcos en un canal haciendo uso de hilos. Los hilos simbolizan distintas clases de embarcaciones (convencionales, de pesca y de patrulla) que navegan por un canal en una sola dirección a la vez. Para programar este tráfico, se utilizan diversos algoritmos de programación. Asimismo, un elemento de hardware exhibe de manera visual la situación del canal y las embarcaciones a través de luces LED, garantizando la prevención de colisiones.

Por último, se requiere que el documento describa cómo se llevó a cabo la integración de la biblioteca CEthreads, la simulación de tráfico marítimo y la aplicación de los algoritmos de planificación en el proyecto.

II. Desarrollo

A. Ambiente de desarrollo

El proyecto fue desarrollado en un ambiente Linux, se recomienda ejecutarlo en Ubuntu. Es necesario que el ambiente de ejecución tenga instalado un compilador de C, como GCC, además la librería GNU Make para poder hacer uso del makefile para compilar el programa y ejecutarlo.

III. Atributos

A. Estrategias para el trabajo individual y en equipo (planificación, ejecución y evaluación)

- **Planificación:** A nivel de planificación podemos decir que se discutieron y definieron los objetivos desde el inicio, asegurando que cada miembro del equipo pudiera aportar a la realización de esta tarea según las habilidades y conocimientos de cada uno. Podemos decir que se utilizo un enfoque inclusivo, ya todos los miembros participaron en la toma de decisiones y se estuvo abierto a escuchar las opiniones. Se pensó en que la división de tareas maximizara el trabajo concurrente, sin embargo en algunos casos esto no era posible, ya que una tarea dependía de que otra estuviera completa o semifuncional.
- **Ejecución:** Se insto a la participación equitativa de todos los miembros, ya que tuvimos un espacio para tener reuniones y escuchar asertivamente el uno al otro, con el fin de compartir ideas y resolver problemas. Durante las fases de desarrollo, nos reunimos para intercambiar propuestas sobre el manejo de los hilos y los barcos y el diseño de la pantalla.
- **Evaluación:** Aparte de las sesiones durante el desarrollo, hicimos sesiones de retroalimentación y revisión, donde se revisaron los avances y se ajustaron diversos puntos tanto en la parte técnica del proyecto como en los roles.

Encargado	Objetivo	Metas
Kevin y Noel	Implementación de la biblioteca de hilos	• Implementar desde cero una biblioteca que maneje la creación, finalización y sincronización de hilos • Implementar la función para manejo de candados (mutex)
Karina	Implementación de los algoritmos de calendarización	• Implementar los diferentes algoritmos de calendarización (RR, Prioridad, SJF, FCFS, Tiempo real) • Integrar con la biblioteca de hilos
Nicole	Simulación de los barcos	• Implementar la lógica de los barcos como hilos, incluida la creación y configuración del comportamiento (tipo) • Implementar el paso de los barcos por el canal • Manejar la generación de barcos, simulando la espera en cola y su cruce por el canal • Sincronizar el acceso al canal usando mutex
Kevin	Implementación del flujo del canal	• Implementar la lógica del control del flujo en el canal, según los algoritmos de flujo (Equidad, Letrero, Tico). • Implementar el correcto cambio de sentido del canal • Manejar la sincronización de los barcos (hilos) para garantizar que los barcos no crucen al mismo tiempo en sentidos opuestos.
Sebastián	Implementación del hardware	• Diseñar el circuito para que los LEDs representen el estado del canal, el sentido del flujo y los barcos cruzando el canal. • Integrar el hardware con la lógica del proyecto en C

Fig. 1. División de tareas y metas para cada sección.

B. Indicar la planificación del trabajo mediante la identificación de roles, metas y reglas

C. Indicar cuales acciones promueven la colaboración entre los miembros del equipo durante el desarrollo del proyecto.

Para la organización del equipo se creo un grupo de Whatsapp, para poder tener una buena comunicación con los miembros del grupo. Además, el uso de herramientas de control de versiones como git permitió que todos pudiéramos ver y contribuir a los avances del proyecto en tiempo real. Esto ayudo a que hubiese una colaboración con aportes funcionales.

D. Indicar como se ejecutan las estrategias planificadas para el logro de los objetivos

La estrategia principal fue iterativa, se desarrollaron primero las funcionalidades básicas, como la creación de hilos, la calendarización con un solo algoritmo y se realizaron mejoras progresivas. Entonces, primero, nos aseguramos que las bases funcionaran bien antes de seguir implementando las mejoras y el resto de funcionalidades. Este enfoque permitió identificar y corregir errores a medida que surgían, garantizando que los objetivos se cumplieran de forma gradual.

E. Indicar la evaluación para la el desempeño del trabajo individual y en equipo

Contribuciones Individuales (Sebastian)

- Diseño del hardware: 10
- Implementación de comunicación serial: 10
- Acoplamiento de hilos con el hardware: 10

Contribuciones Individuales (Nicol)

- Diseño de los barcos: 10
- Implementación del paso por el canal: 10
- Sincronización de barcos: 10

Contribuciones Individuales (Noel)

- Implementación de terminación de hilos: 10
- Pruebas de funcionamiento de hilos

Contribuciones Individuales (Karina)

- Algoritmos de calendarización SJF, Prioridad: 10
- Depuración de pruebas finales del proyecto: 10
- Acoplamiento de hilos con el hardware: 10

Contribuciones Individuales (Kevin)

- Implementación de creación y eliminación de mutex: 10
- Implementación de creación de hilos: 10
- Implementación de algoritmos de calendarización RR y Tiempo real: 10

Desempeño en el Trabajo en Equipo (Todos)

- Colaboración en la Revisión: 10
- Depuración en Conjunto: 10
- Comunicación y Propuestas: 10
- Gestión del Tiempo: 10
- Revisión de Código: 10
- Documentación del Código: 10

F. Indicar la evaluación para las estrategias utilizadas de equidad e inclusión

Estas estrategias se evaluaron observando como cada miembro del equipo pudo participar en las decisiones del proyecto y aportar sus ideas. Se revisaron las responsabilidades distribuidas para asegurarse de que todos tuvieran una carga de trabajo justa, y se alentó la participación activa de todos en las discusiones importantes.

G. Indicar la evaluación para las acciones de colaboración entre los miembros del equipo

La colaboración se evaluó revisando como los miembros del equipo resolvıan problemas en conjunto, compartıan ideas y se apoyaban mutuamente. Las sesiones de revisión de código permitieron a todos compartir sus conocimientos y buscar soluciones a problemas complejos.

IV. Diseño

A. Diagrama de arquitectura

En la figura 2, se observa el diagrama de arquitectura para el sistema de Scheduling Ships, que se compone de dos capas principales, la capa de aplicación y la capa de software.

B. Diagrama de componentes

La figura 3 muestra el diagrama de componentes

C. Diagrama de secuencia

La figura 4 muestra el diagrama de secuencia, donde se describe primero la inicialización del canal, la llegada de los barcos al canal y a continuación la calendarizacion de cada barco para que pueda pasar por el canal, el programa termina cuando los barcos pasaron por el canal.

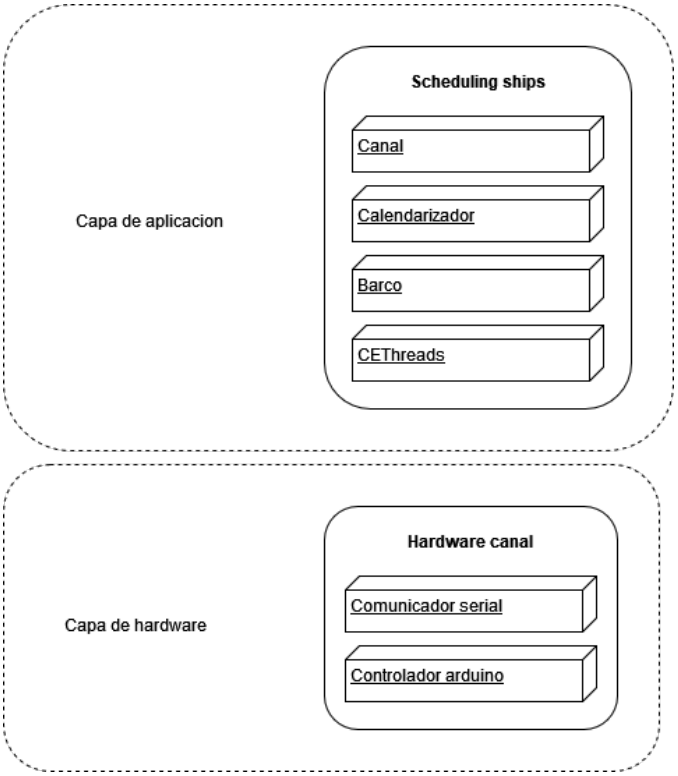

Fig. 2. Diagrama de arquitectura.

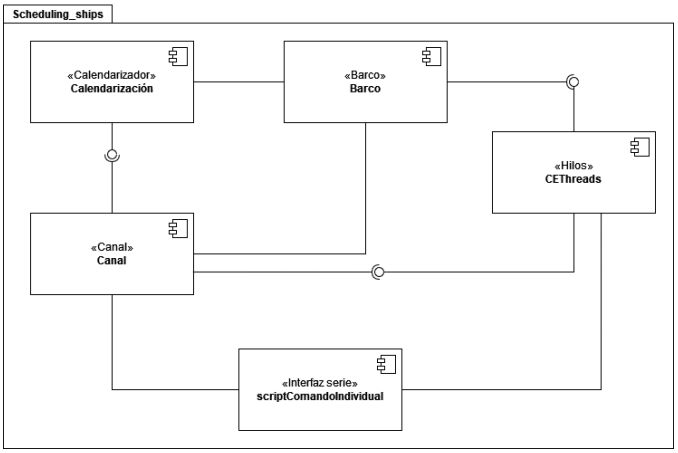

Fig. 3. Diagrama de componentes.

V. Instrucciones de uso

El proyecto se debe ejecutar en un ambiente Linux, para poder iniciar se debe descargar el código fuente del repositorio de Github: Scheduling ships, este repositorio contiene la versión final del código. Dentro del repositorio se encuentra un Makefile, para compilar la versión sin utilizar el hardware, debe ejecutar el comando:

```
make simulacion
make run
```

Para ejecutar la versión con conexión al hardware, ejecutar el comando:

```
make hardware
```

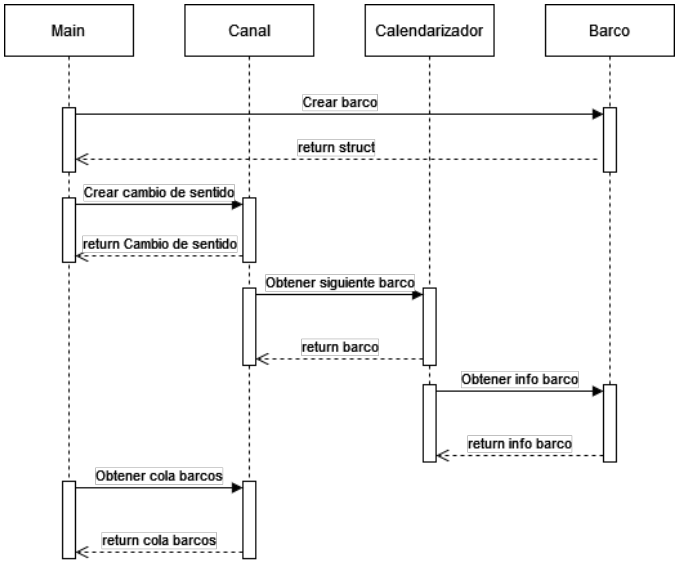

Fig. 4. Diagrama de secuencia para el recorrido de los barcos.

```
make runHardware
```

VI. Conclusiones

La implementación de la biblioteca CEthreads permitió gestionar de manera eficiente múltiples hilos en un entorno de simulación, destacando la importancia de una buena administración del paralelismo para optimizar el uso de los recursos del sistema. Los distintos algoritmos de calendarización mostraron comportamientos diferenciados en cuanto al control del tráfico de barcos, resaltando el impacto de la elección del algoritmo adecuado según las características del canal y la prioridad de los barcos. El desarrollo de un sistema de control de tráfico de barcos basado en hilos demostró que es posible simular un entorno de trafico de barcos, con diferentes características y prioridades, se mueven de manera fluida sin colisiones, garantizando tanto el flujo como la seguridad en el tránsito del canal. Esta simulación evidencia la utilidad de los hilos en el manejo de procesos en sistemas concurrentes y la posibilidad de implementar soluciones eficientes en entornos de tiempo real. Finalmente, la integración del prototipo de hardware permitió representar visualmente el comportamiento del sistema, lo que facilitó la validación de los algoritmos de control de flujo. La representación física del letrero y de las colas de barcos ofreció una comprensión más tangible de los retos que implica la sincronización y la calendarización en sistemas distribuidos, especialmente en casos de tiempo real y prioridades múltiples.

References

[1] G, Ippolito.(2004). POSIX thread (pthread) libraries. https://www.cs.cmu.edu/afs/cs/academic/class/15492-f07/www/pthreads.html.

[2] IMB. (2020). ”pthread.h file,” IBM Documentation. https://www.ibm.com/docs/es/aix/7.2?topic=files-pthreadh-file.

[3] M. Kumar, “Program for Shortest Job First (or SJF) CPU Scheduling — Set 1 (Non- preemptive),” GeeksforGeeks, Feb. 24, 2017. https://www.geeksforgeeks.org/program-for-shortest-job-first-or-sjf-cpu-scheduling-set-1-non-preemptive/

[4] S. Chhabra, “Program for Round Robin Scheduling for the same Arrival time,” GeeksforGeeks, Feb. 28, 2017. https://www.geeksforgeeks.org/program-for-round-robin-scheduling-for-the-same-arrival-time/