

Proyecto 2 sistemas operativos: Scheduling ships

1st Juan Pablo Carrillo Salazar
Ingeniería en computadores
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
juanpcarrillo@estudiantec.cr

2nd Josue Cubero Montero
Ingeniería en computadores
Instituto Tecnológico de Costa Rica
San José, Costa Rica
jdcubero@estudiantec.cr

3rd Jose Pablo Fuentes
Ingeniería en computadores
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
jp.1398@estudiantec.cr

4th Dagoberto Rojas
Ingeniería en computadores
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
krojas96@estudiantec.cr

Abstract—La idea fundamental del presente proyecto es realizar una implementación de los hilos en espacio de usuario de tal manera que sea lo más eficiente posible de acuerdo con la funcionalidad solicitada. Donde la abstracción de todo el ambiente que se debe de tener para controlar cualquier evento se vuelve esencial para un correcto funcionamiento del programa. Para esto se realizó el diseño e implementación de un proyecto que simule el proceso de creación de procesos en un sistema operativo y que se encargue de organizarlos y manejarlos mediante el uso de algoritmos de calendarización. A partir de la realización del proyecto se profundizó en el área de paralelismos en los sistemas operativos, ayudando este a comprender y poner en práctica los conceptos básicos relacionados con este tema.

Index Terms—Linux, Proceses, Schedulers, CPU, Threads

I. INTRODUCCIÓN

Para que un sistema operativo tenga un manejo eficiente de paralelismo, se debe de tener especial cuidado en la creación y manipulación de procesos. En este proyecto se busca implementar los hilos en espacio de usuario definiendo la prioridad en su ejecución en base a una serie de algoritmos de tal manera que sea lo más eficiente posible. Para lograr un buen funcionamiento del programa, debemos de controlar cualquier evento considerando cual sería el modo más óptimo para hacerlo. Para hacer esto posible debemos de hacer uso de hilos y definir su óptima ejecución en base a un calendarizador, quien realiza este proceso en base a una serie de algoritmos tales como **RR**, **Prioridad**, **SJF**, **FCFS** y **Tiempo real**. Estos algoritmos son quienes definen -dentro del calendarizador- el orden de la cola de listos y el orden en el que como estos irán siendo ejecutados.

En el caso del manejo de hilos, se tuvo que reimplementar la biblioteca Pthreads llamándola CETHreads y haciendo uso de directivas de C. Dicha reimplementación incluye las funciones:

- CETHread_create
- CETHread_end
- CETHread_join
- CEMutex_init
- CEMutex_destroy
- CEMutex_unlock

En el contexto del problema a resolver dentro de este proyecto, se utilizaron las siguientes abstracciones:

- Cada hilo sería considerado como un **barco**.
- El procesador sería considerado como un **canal** sobre el cual viajan los barcos.

Si consideramos la idea que hay detrás de la ejecución de un proceso, sabemos que este siempre tendrá un tiempo de ejecución en base a lo que sea que este haga. Para lograr que los distintos hilos tuvieran tiempos de ejecución variables dentro de la abstracción utilizada, cada barco debía de tener una serie de parámetros que sirvieran para definir dicho tiempo de ejecución. En este caso el tiempo de ejecución de cada hilo/barco sería definido en base a su velocidad, la cual estaría definida según el tipo de barco:

- Normales: Su velocidad es baja, lo cual implica alto tiempo de ejecución.
- Pesqueros: Su velocidad es media, lo cual implica un tiempo de ejecución entre el tiempo de ejecución de los barcos normales y los patrulla.
- Patrulla: Su velocidad es alta, lo cual implica que su tiempo de ejecución sería el más rápido respecto al de los demás barcos.

Para identificar hasta qué punto cada barco/hilo se logra ejecutar o avanzar a través del canal antes de que este sea detenido por el calendarizador, se dividió el procesador/canal en etapas. Cada una de estas etapas funcionarían para conocer cuán avanzado está la ejecución de cada hilo/barco. Finalmente, para lograr que estas dos abstracciones, tanto para los hilos como para el procesador, pudieran ser mostradas ante un usuario, se hizo uso de dos elementos:

- Hardware: Este debe de servir para darle al usuario una noción de que es lo que está pasando con la ejecución de cada hilo/barco.
- Interfaz gráfica: Este recurso serviría para lo mismo que el anterior pero haciendo uso únicamente de software.

debíamos de "dividir" en etapas a el procesador para que así, pudiésemos llevar un control conocido de cuánto tiempo iba

poder ejecutarse cada hilo en base a su duracion y cuanto de su trabajo iba a poder lograrse en dicho tiempo.

II. AMBIENTE DE DESARROLLO

Para el desarrollo de cada módulo, algoritmo o proceso se uso el lenguaje de programación C, los cuales eran compilados y ejecutados en la última versión estable de Ubuntu, la 21.10 (Impish Indri). El compilador utilizado fue GCC, una serie de compiladores de software libre distribuido por Free Software Foundation.

Para la ejecución de este proyecto se requiere descargar e instalar una biblioteca gráfica, necesaria para visualizar el comportamiento de la aplicación. El proyecto usa "raylib" y puede ser descargada e instalada siguiendo los pasos desde su repositorio oficial <https://github.com/raysan5/raylib>.

Los requisitos mínimos para la ejecución del programa son:

- Procesador de doble núcleo de 2 GHz o superior
- 2 GB de RAM
- 25 GB de espacio libre en el disco duro (Para poder instalar Ubuntu sin problemas)
- Tarjeta gráfica VGA

III. ATRIBUTOS

Esta sección describe los atributos del Instituto Tecnológico de Costa Rica y cómo estos se emplearon durante el desarrollo del proyecto. [1]

A. Aprendizaje continuo

De acuerdo con la definición del Instituto Tecnológico de Costa Rica, el aprendizaje continuo es: [1]

Capacidad para reconocer las necesidades propias de aprendizaje y la habilidad de vincularse en un proceso de aprendizaje independiente durante toda la vida, en un contexto de amplio cambio tecnológico.

Durante este proyecto se trabajó en el aprendizaje continuo de las diferentes maneras:

- Comprensión de problemas complejos sobre analogías de calendarización de procesos en el espacio de usuario.
- Desarrollo de algoritmos de calendarización en bajo nivel, basados en descripciones de alto nivel.
- Aprendizaje sobre el manejo y creación de bibliotecas estáticas en lenguaje de programación C.

Se encontraron, además, varios retos importantes en esta categoría, tales como la búsqueda de una biblioteca mínima para el manejo de hilos en C, de manera que se pudiera implementar solo los métodos necesarios para obtener los resultados deseados.

B. Herramientas de Ingeniería

Nuevamente, la definición del Instituto Tecnológico de Costa Rica en este atributo se define como:

Capacidad para crear, seleccionar, aplicar, y adaptar apropiadamente técnicas, recursos y herramientas modernas de ingeniería y de tecnologías de información, incluyendo predicción y modelado de

TABLE I
TABLA GENERAL DE ACTIVIDADES POR ESTUDIANTE

Actividad	Responsable	Horas invertidas
Diseño de la solución	Todos	3
Desarrollo de CETHreads	Dagoberto	3
Implementación algoritmo de prioridad	Jose Pablo	3
Implementación algoritmo fcfs	Josue	3
Implementación algoritmo sjf	Dagoberto	3
Implementación algoritmo RR	Juan Pablo	3
Creación de Barcos	Josue y Jose Pablo	4
Manejo de archivos	Josue	2
Algoritmos para el tránsito	todos	3
Conexión C-Arduino	todos	4
Documentación	todos	4
Prueba de funcionalidades	todos	3
Total de tiempo invertido		38

problemas de complejos de ingeniería, con la comprensión de las limitaciones asociadas.

Durante este proyecto, se implementaron las siguientes herramientas de ingeniería:

- Diagramas UML (secuencia, arquitectura)
- Desarrollo ágil, priorizando entregables mínimos funcionales de inicio a fin.
- Diseño de esquemáticos electrónicos, para una propuesta de hardware
- Desarrollo iterativo, añadiendo elementos en cada entregable
- Desarrollo usando branching en git
- Uso de git como control de versiones.

En este atributo, se aprendió y hubo familiaridad con nuevas estrategias de desarrollo presentadas por los diferentes compañeros, llevando a una mejor comprensión y manejo de las mismas, por ejemplo git.

IV. DISEÑO

A. Diagrama de secuencia

V. INSTRUCCIONES DE USO

VI. TABLA DE ACTIVIDADES

VII. CONCLUSIONES

1)

REFERENCES

- [1] Instituto Tecnológico de Costa Rica 2022. Atributos TEC. [online]. Disponible en: <https://www.tec.ac.cr/atributos-tec> [Accesado 31 Octubre 2022].

VIII. ANEXOS