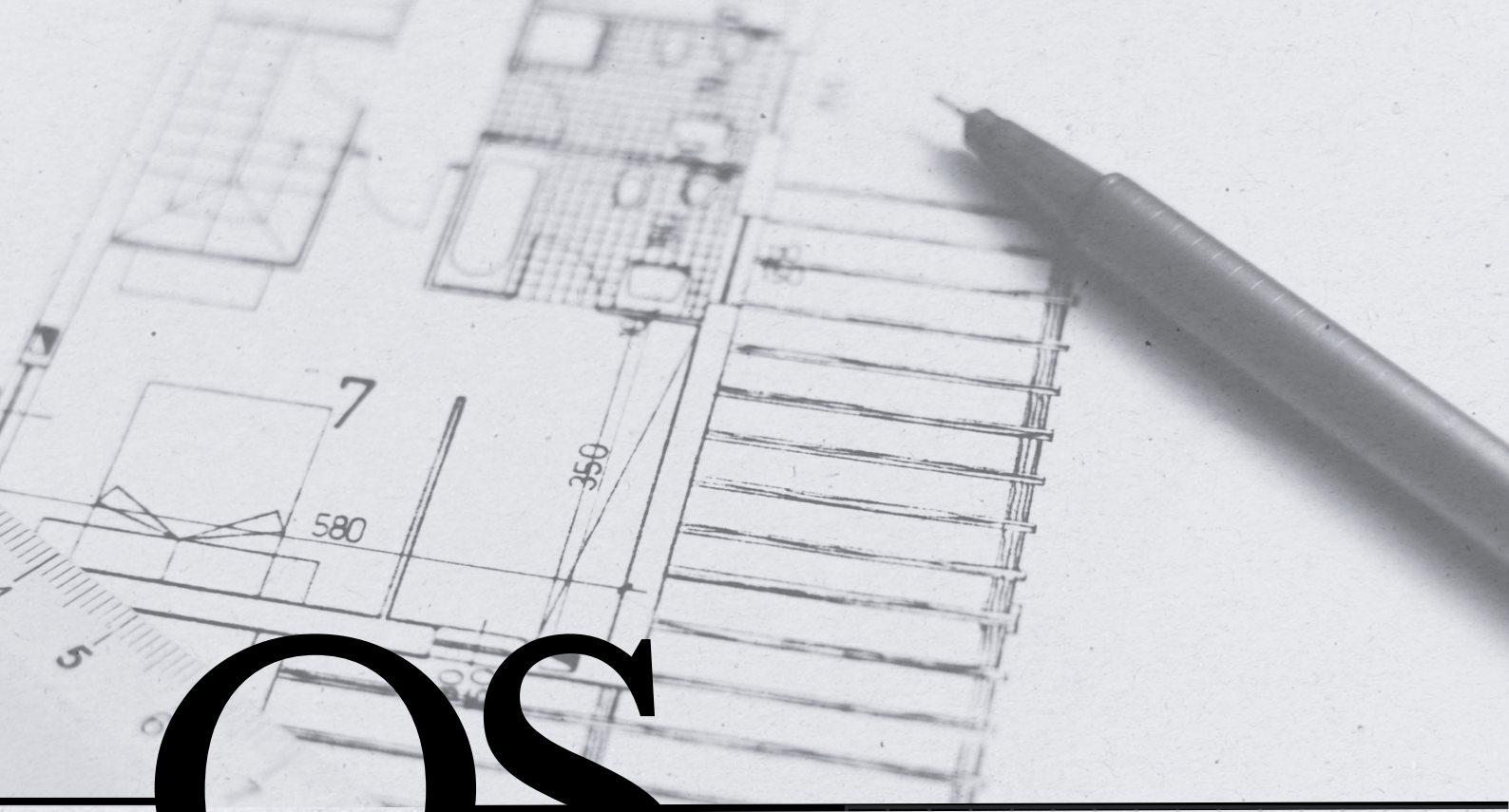


OS

PROYECTO FINAL

Shortest Job Next (SJN)

DESCRIPCIÓN:

Crear una simulación con Interfaz Gráfica de Usuario (GUI) del planificador de proceso Shortest Job Next (SJN), también conocido como Shortest Job First (SJF).

Por: Brayan Mejía Mora
Profesor: Lopez Herrera Juan Luis

UNIVERSIDAD VERACRUZANA
Ciclo 2023 - 2024

Shortest Job First (SJF) Schedule

Brayan Mejía Mora

INTRODUCCION

En este proyecto, exploraremos a fondo el funcionamiento del planificador de procesos "Shortest Job First" (SJF) mediante la creación de una simulación. El propósito principal de esta iniciativa es lograr una comprensión práctica y profunda de los planificadores de procesos, específicamente enfocándonos en un sistema de un solo núcleo.

Objetivos del proyecto:

Comprensión Práctica del SJF:

Explorar cómo el algoritmo SJF determina la planificación de procesos en base a la duración estimada de ejecución.

Simulación Interactiva:

Desarrollar una simulación interactiva que permita visualizar la distribución y planificación de los procesos en un sistema de un solo núcleo.

Desarrollo de Habilidades de Implementación:

Potenciar las habilidades de implementación al llevar a cabo la simulación del algoritmo SJF.

Motivación del Proyecto:

Este proyecto va más allá de la simple tarea académica. La motivación radica en:

- **Comprensión Profunda:** No se trata solo de completar una materia, sino de comprender a fondo la utilidad de los planificadores de procesos.
- **Aplicación Práctica:** La teoría es esencial, pero la verdadera comprensión proviene de la aplicación práctica. La simulación nos permite llevar el conocimiento a la práctica.
- **Inmersión en Algoritmos Operativos:** Al enfocarnos en el SJF, estamos explorando uno de los algoritmos fundamentales utilizados en sistemas operativos

CONTEXTO

Para llevar a cabo la implementación del algoritmo de planificación de trabajos más cortos (SJF), me enfrenté a varios desafíos que requerían una cuidadosa consideración y

programación. Aunque el algoritmo en sí no es inherentemente complejo, la simulación exigía una representación visual clara de la distribución de procesos en un entorno de un solo núcleo. Aquí describo cómo abordé estos aspectos clave del proyecto:

Generación Aleatoria de Procesos:

- Desarrollé un programa interno que simulaba la creación aleatoria de procesos para representar la dinámica de un sistema operativo real. Esto involucró la generación de procesos con duraciones estimadas de ejecución variables.

Representación Visual con Colores:

- Implementé un sistema de colores para visualizar los procesos de manera más intuitiva. Cada proceso se representó con un color único, facilitando la identificación y el seguimiento visual a lo largo de la simulación.

Indicación del Proceso Actual:

- Destaqué visualmente el proceso actual en ejecución para proporcionar una comprensión inmediata de qué tarea estaba siendo procesada en un momento dado. Esto ayudó a seguir la secuencia de ejecución y entender la lógica detrás del planificador SJF.

Consideraciones de un Solo Núcleo:

- Programé el planificador teniendo en cuenta que solo podía haber un núcleo. Esto implicó la gestión cuidadosa de la ejecución de procesos de manera secuencial y la asignación de recursos de manera eficiente.

Parámetros de Ajuste Dinámico:

- Incorporé parámetros ajustables que permitieron la modificación dinámica de los valores de los procesos. Esto facilitó la observación de cómo variaciones en los tiempos de ejecución y las prioridades afectaron la planificación y distribución de procesos.

CODIGO FUENTE

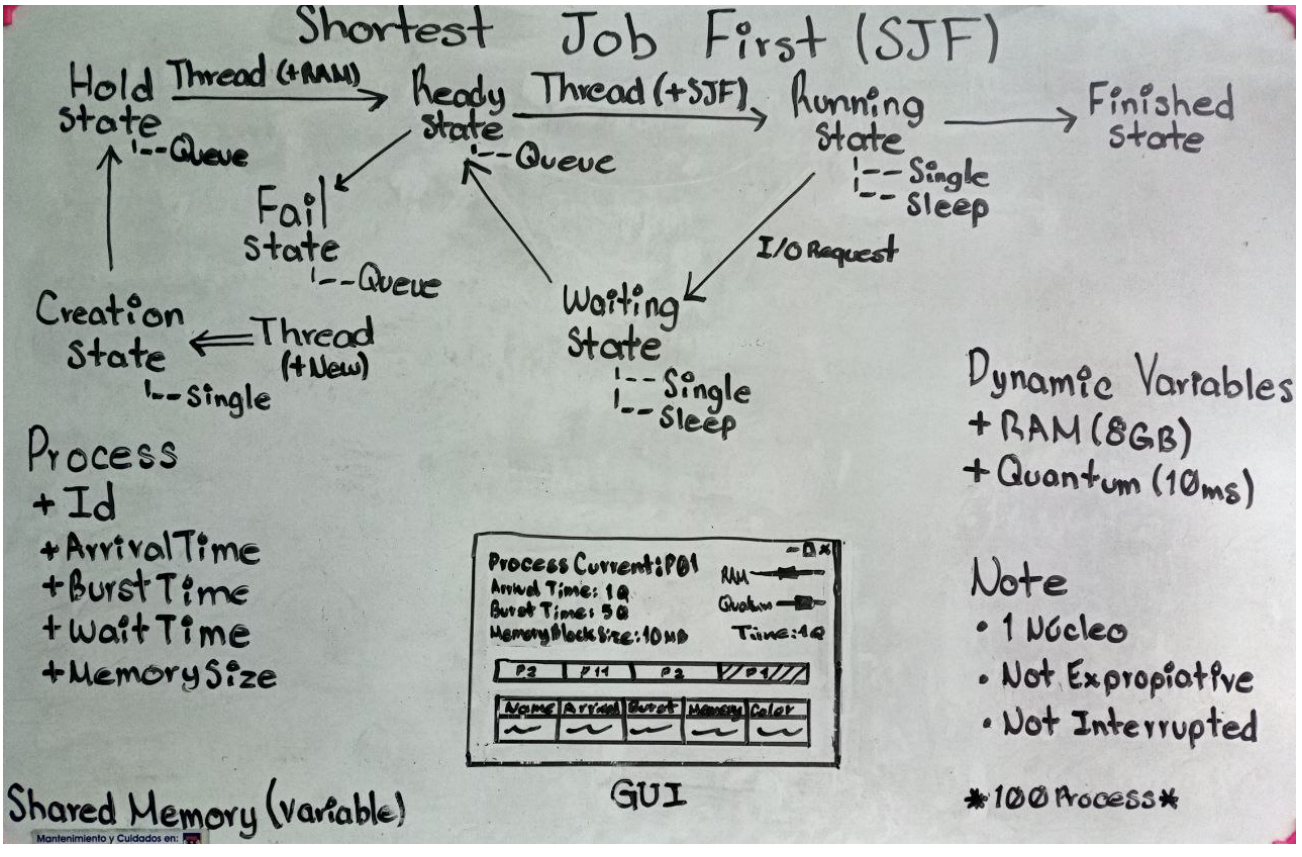


Imagen 1. Idea

Para dar forma a la implementación de la simulación, decidí inicialmente plasmar mis ideas en un pizarrón. Esta elección se basó en la conveniencia y en mi preferencia personal, ya que me permite visualizar de manera más clara el flujo del programa. En este dibujo, esbocé la estructura y el funcionamiento esperado del programa, identificando los estados que cada proceso debía atravesar y señalando los puntos donde se requerirían hilos para simular el funcionamiento simultáneo, emulando así un sistema operativo real.

El dibujo inicial también sirvió como punto de partida para concebir la interfaz visual del programa. Aunque la representación final difirió de la idea original, este proceso me recordó la realidad de que la planificación de proyectos no siempre sigue un camino lineal y que la iteración es una parte fundamental del desarrollo.

Además, incorporar conscientemente limitaciones al sistema simulado con el propósito de explorar y comprender mejor diversas situaciones, por en ende el sistema simulado tiene un solo núcleo y 8 MB de RAM. Aunque permití cierta variabilidad en la asignación de memoria en relación con los procesos, es crucial destacar que la gestión directa de la memoria corresponde al planificador de memoria. Por razones de simplicidad, opté por implementar un planificador de memoria tipo FIFO.

El dibujo también arrojó luz sobre la naturaleza no expropiativa del planificador de procesos. Al no ser expropiativo, un proceso mantiene la CPU hasta que completa su ejecución, lo que implica que si realiza alguna operación de entrada o salida, permanece

en espera con la CPU, aumentando así la ráfaga de tiempo. Esta observación resaltó la importancia de comprender las características del algoritmo de planificación elegido.

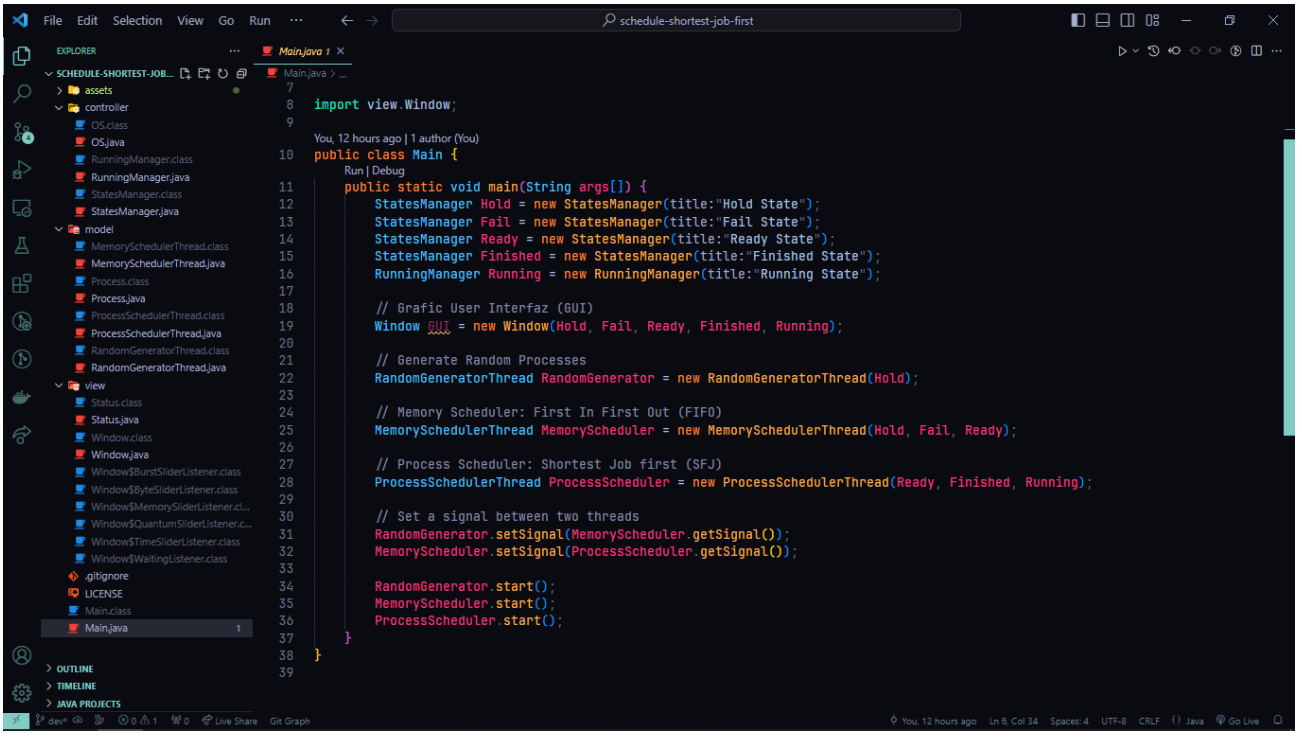


Imagen 2. Código

Para la realización de este proyecto, mi enfoque inicial consistió en identificar los puntos críticos donde sería necesario emplear hilos para garantizar un funcionamiento simultáneo y eficiente del sistema de simulación. Estos puntos clave incluyeron la generación aleatoria de procesos, la transferencia de procesos entre la cola de retención y la cola de procesos listos con memoria asignada, y la ejecución de procesos en la CPU utilizando el planificador de procesos "Shortest Job First" (SJF).

La gestión de colas y la creación de hilos no fueron los únicos desafíos; también fue esencial incorporar semáforos para controlar las secciones críticas del código durante la transición de procesos entre diversos estados (Hold, Fail, Ready, Running, End, Finished). Destaco la necesidad de orquestar el inicio de los hilos de manera cuidadosa, activándolos solo cuando era realmente necesario, es decir, cuando al menos había un proceso en la cola correspondiente.

Con el objetivo de finalizar el proyecto en un plazo de tiempo eficiente, logré que el programa funcionara desde la línea de comandos en un solo día. Sin embargo, la complejidad surgió al implementar la interfaz gráfica, lo que implicó reorganizar archivos y códigos. En este proceso, opté por adoptar un diseño de modelo-vista-controlador para crear la ventana gráfica, lo que permitió una mejor organización y modularidad en el código.

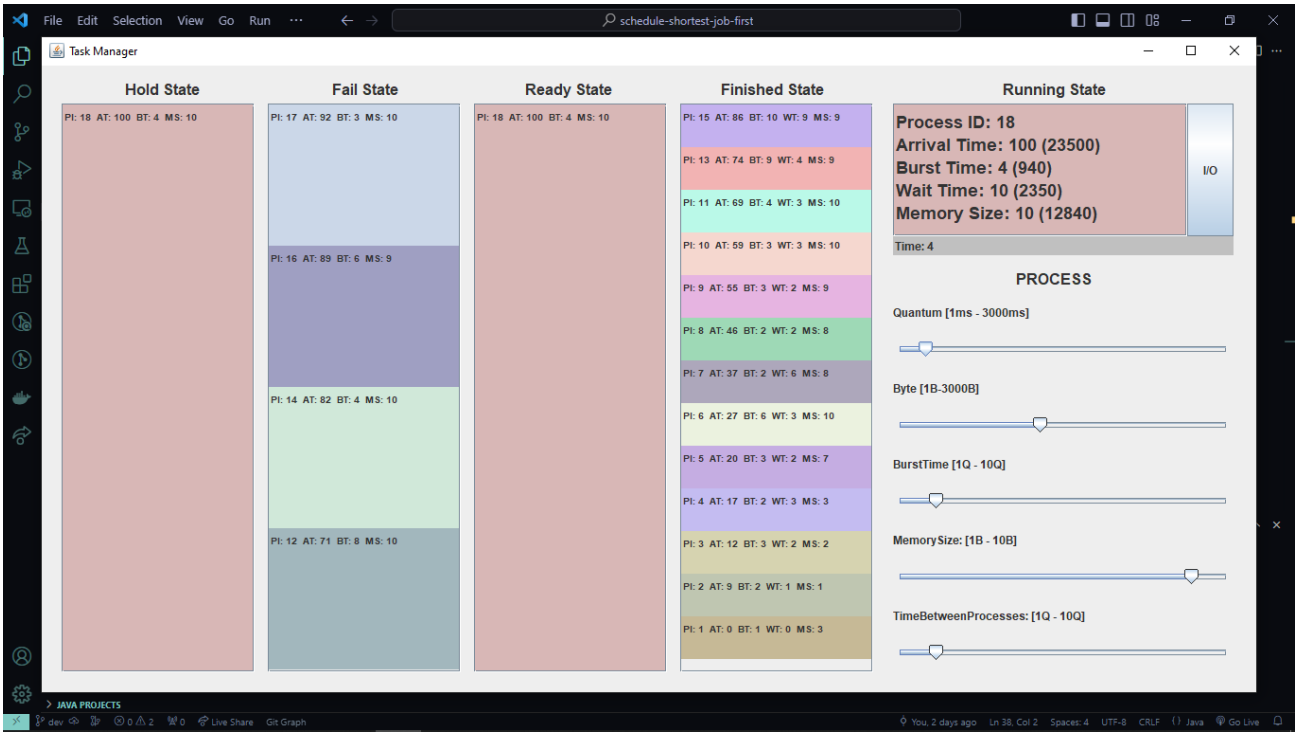
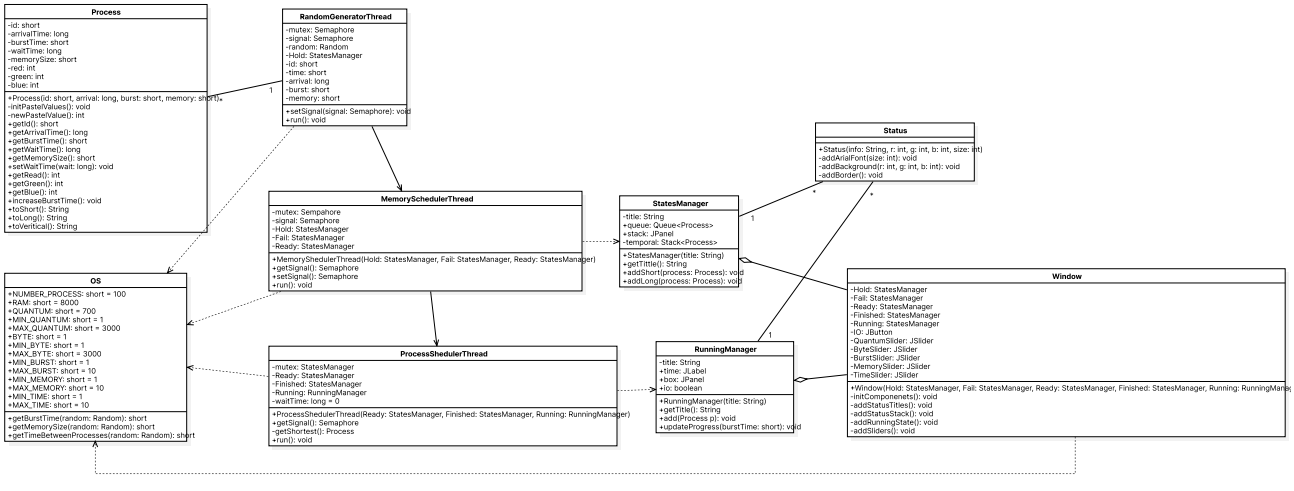


Imagen 2. Window



Imange 4. ModeloUML

CONCLUSION

Al finalizar este proyecto, he tenido la valiosa oportunidad de sumergirme en el mundo práctico de los sistemas operativos mediante la implementación del algoritmo "Shortest Job First" (SJF). Esta experiencia no solo ha sido un ejercicio técnico, sino una verdadera exploración que va más allá de la teoría académica, permitiéndome comprender de manera más profunda el funcionamiento interno de los sistemas operativos.

La implementación activa del algoritmo SJF me ha brindado una apreciación más rica de la planificación de procesos. Enfrentándome a los desafíos de la simulación, he podido observar cómo el sistema operativo toma decisiones en tiempo real, considerando la duración estimada de ejecución de cada proceso y optimizando la secuencia de ejecución.

Este proyecto no solo ha sido un logro personal, sino también un paso significativo hacia la comprensión integral de los principios que sustentan la informática moderna. La interacción con el SJF ha proporcionado una visión valiosa sobre cómo los sistemas operativos gestionan eficientemente los recursos, minimizando el tiempo de espera y optimizando la ejecución de procesos.

LISTA DE VIDEOS

https://youtube.com/playlist?list=PLO15Alfd4DNjxrvFxSlYQe1H6n-OEu235&feature=shared

REFERENCIAS

Jenny's Lectures CS IT. (2019, 22 marzo). Shortest Job First(SJF) scheduling algorithm with example | Operating system [Vídeo]. YouTube. https://www.youtube.com/watch?v=pYO-FAg-TpQ