**Explorando los Procesos y Algoritmos de Planificación de CPU**



**Integrantes:**

Fabricio Alejandro Villegas Zeballos

**Gestión: 2025**

**1. Introducción**

La planificación de la CPU es un componente esencial del sistema operativo, ya que determina el orden en que los procesos acceden al procesador. Esta investigación tiene como objetivo comprender en profundidad los algoritmos de planificación más utilizados, su funcionamiento, sus ventajas y limitaciones, así como los estados que atraviesan los procesos. Además, se busca implementar dichos algoritmos en una aplicación desarrollada en C# utilizando WPF, para facilitar la visualización y comprensión de estos conceptos. La dinámica grupal fomenta la investigación colaborativa, el análisis crítico y el desarrollo de habilidades técnicas clave en el ámbito de los sistemas operativos y la programación basada en hilos.

**📘 2. Conceptos básicos sobre procesos y planificación de la CPU**

Un proceso es una instancia de un programa en ejecución. Los procesos requieren recursos del sistema como memoria, tiempo de CPU y dispositivos de E/S para completarse. Los sistemas operativos modernos utilizan mecanismos de planificación de procesos para asignar eficientemente el tiempo de CPU entre múltiples procesos activos.

La planificación de CPU es la tarea del sistema operativo que decide qué proceso será ejecutado por la CPU en un momento dado. Este proceso es necesario en sistemas multitarea para maximizar el uso de la CPU, mejorar la respuesta del sistema y asegurar una ejecución justa entre procesos.

**🔄 3. Estados de un proceso**

Durante su ciclo de vida, un proceso puede pasar por los siguientes estados:

* Nuevo (New): El proceso ha sido creado.
* Listo (Ready): Esperando a ser asignado a la CPU.
* Ejecución (Running): Actualmente ejecutándose en la CPU.
* Esperando (Waiting): Esperando a que ocurra un evento, como la finalización de una operación de E/S.
* Terminado (Terminated): El proceso ha finalizado su ejecución.

La planificación de la CPU debe gestionar estos estados de manera eficiente, especialmente en situaciones donde múltiples procesos compiten por el mismo recurso.

**⚙️ 4. Algoritmos de planificación de CPU**

A continuación, se describen algunos de los algoritmos de planificación más relevantes:

4.1 FCFS (First Come, First Served)

* Descripción: Atiende los procesos por orden de llegada.
* Ventajas: Simple de implementar.
* Desventajas: Puede generar largos tiempos de espera (problema de convoy).
* Adecuado para: Sistemas donde el orden de llegada es prioritario.

4.2 SJF (Shortest Job First)

* Descripción: El proceso con el menor tiempo de ejecución es atendido primero.
* Ventajas: Tiempo de espera promedio mínimo.
* Desventajas: Puede causar inanición de procesos largos.
* Adecuado para: Sistemas con trabajos previsibles y controlados.

4.3 Round Robin

* Descripción: Cada proceso recibe un pequeño intervalo de tiempo llamado *quantum*.
* Ventajas: Equidad entre procesos, ideal para sistemas interactivos.
* Desventajas: Si el *quantum* es muy corto, hay sobrecarga; si es muy largo, se comporta como FCFS.
* **Adecuado para: Sistemas en tiempo compartido.**

**📊 5. Comparación: Ventajas y Desventajas**

| Algoritmo | Ventajas | Desventajas |
| --- | --- | --- |
| FCFS | Fácil de implementar | Tiempo de espera elevado en algunos casos |
| SJF | Menor tiempo de espera promedio | Posible inanición de procesos largos |
| Round Robin | Justo y eficiente en tiempo compartido | Complejidad en la selección del *quantum* |

**🧪 6. Ejemplo de implementación (pseudocódigo simplificado)**

text

CopiarEditar

RoundRobin(processes, quantum)

queue ← processes

while queue no está vacía:

proceso ← queue.dequeue()

if proceso.tiempoRestante > quantum:

ejecutar proceso durante quantum

proceso.tiempoRestante -= quantum

queue.enqueue(proceso)

else:

ejecutar proceso durante tiempoRestante

proceso.tiempoRestante = 0

Este algoritmo simula la planificación Round Robin, donde cada proceso recibe una cantidad limitada de tiempo (quantum) antes de que el siguiente proceso sea ejecutado.

**🧾 7. Conclusiones**

La planificación de la CPU es un área fundamental dentro del estudio de los sistemas operativos, especialmente en ambientes donde se ejecutan múltiples procesos simultáneamente. Comprender los diferentes algoritmos y sus implicaciones permite a los futuros desarrolladores tomar decisiones informadas al diseñar sistemas eficientes. La implementación de estos algoritmos en lenguaje C# mediante WPF no solo permite poner en práctica los conocimientos adquiridos, sino también visualizar de forma clara el comportamiento de cada algoritmo. Esta experiencia aporta tanto habilidades técnicas como competencias para el trabajo colaborativo**.**

**📚 8. Referencias bibliográficas (APA 7)**

* Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Fundamentals of Operating Systems* (9.ª ed.). Wiley.
* Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (4.ª ed.). Pearson.
* Arpaci-Dusseau, R. H., & Arpaci-Dusseau, A. C. (2018). *Operating Systems: Three Easy Pieces*. Arpaci-Dusseau Books.
* Microsoft Learn. (2024). *C# Programming Guide*. <https://learn.microsoft.com/en-us/dotnet/csharp/>