



Centro Universitario de Ciencias Exactas e Ingenierías.

Departamento para la integración ciber humana.

Sistemas operativos.

Becerra Velázquez Violeta Rocío.

**Yáñez Salazar Saul Emanuel.**

**220656484**

Ingeniería en computación.

D04.

Algoritmos de planificación.

Martes 25 de febrero del 2025.

## **Hilos POSIX.**

Los hilos POSIX, también conocidos como Pthreads, son una implementación de hilos de ejecución basada en el estándar POSIX.1c (IEEE Std 1003.1c-1995), que proporciona una API estándar para la creación y gestión de hilos en sistemas operativos compatibles con POSIX, como Linux, macOS y varios sistemas Unix. Estos hilos permiten la ejecución concurrente dentro de un mismo proceso, lo que facilita la paralelización de tareas y el aprovechamiento de múltiples núcleos en los procesadores modernos.

Una de las principales ventajas de los hilos POSIX es que los hilos dentro de un mismo proceso comparten el mismo espacio de direcciones, lo que permite una comunicación eficiente entre ellos sin la necesidad de mecanismos de comunicación interprocesos como tuberías o memoria compartida tradicional. Sin embargo, esta misma característica introduce la necesidad de sincronización adecuada para evitar problemas como condiciones de carrera o inconsistencias en los datos.

El estándar Pthreads proporciona una serie de funciones para la gestión de hilos, entre las cuales destacan `pthread_create`, que permite la creación de un nuevo hilo dentro del proceso; `pthread_exit`, que permite la terminación explícita de un hilo; `pthread_join`, que bloquea la ejecución del hilo que lo llama hasta que el hilo especificado haya terminado; y `pthread_self`, que devuelve el identificador del hilo actual. Además, existen mecanismos de sincronización como los mutexes, que se utilizan para garantizar acceso exclusivo a secciones críticas del código, y las variables de condición, que permiten una sincronización más avanzada entre hilos.

El uso de hilos POSIX es común en aplicaciones de alto rendimiento que requieren ejecutar múltiples tareas en paralelo, como servidores web, motores de bases de datos y aplicaciones científicas o de simulación. Sin embargo, el uso incorrecto de los hilos puede llevar a problemas complejos como condiciones de carrera, interbloqueos y cambios de contexto innecesarios que pueden afectar el rendimiento del sistema. Por esta razón, es fundamental aplicar buenas prácticas de programación concurrente, como evitar el uso excesivo de hilos, minimizar las regiones críticas y utilizar estructuras de datos diseñadas específicamente para la concurrencia. (Universidad Tecnica Federica Santa Maria. , s.f.) (Contreras., 2012)

### **Describe en qué consisten los algoritmos de planificación Apropiativos.**

Los algoritmos de planificación apropiativos son aquellos en los que el sistema operativo puede interrumpir la ejecución de un proceso en cualquier momento para asignar la CPU a otro proceso con mayor prioridad o según la estrategia definida. A diferencia de los algoritmos no apropiativos, en los que un proceso

mantiene la CPU hasta que finaliza o entra en estado de espera, los apropiativos permiten una mayor flexibilidad y optimización en el uso de los recursos del sistema.

**¿En qué consiste el Algoritmo de Planificación por Prioridades y como se clasifica?**

El algoritmo de planificación por prioridades asigna la CPU a los procesos según su **nivel de prioridad**, en lugar de hacerlo en orden de llegada o duración estimada. Cada proceso tiene un valor de prioridad, y el sistema siempre selecciona para ejecución el proceso con la mayor prioridad.

**¿En qué consiste el Algoritmo de Planificación de colas múltiples?**

El algoritmo de planificación de colas múltiples divide los procesos en diferentes colas según sus características y prioridades. Cada cola tiene su propio algoritmo de planificación, y el sistema define reglas para la asignación de CPU entre ellas.

**Diferencia entre el estado “Bloqueado y Suspendido” y “Listo y Suspendido”.**

<b>Criterio</b>	<b>Bloqueado y suspendido</b>	<b>Listo y suspendido.</b>
<b>¿Está esperando un recurso?</b>	<b>Si, entrada/salida o un evento.</b>	<b>No (solo espera RAM)</b>
<b>¿Puede ejecutarse de inmediato?</b>	<b>No, primero debe desbloquearse</b>	<b>No, primero debe cargarse en RAM.</b>
<b>¿Dónde está almacenado?</b>	<b>Memoria secundaria.</b>	<b>Memoria secundaria.</b>
<b>Ejemplo típico</b>	<b>Espera de lectura de disco y además no hay RAM.</b>	<b>Esta en la cola de listos, pero no hay espacio en RAM.</b>

**¿Cuál es el tiempo de respuesta?**

El tiempo de respuesta es el intervalo que transcurre desde que un proceso ingresa en la cola de listos hasta que comienza su ejecución por primera vez.

**Describe el algoritmo de planificación apropiativo RR.**

Se inicia tomando un valor de tiempo que pasaran todos los procesos, y en base a su tiempo de llegada, se forman. El quantum es una unidad de tiempo el cual nos indica cuanto tiempo debe estar el proceso en la CPU, en caso de que no termine, lo vuelve a formar en la cola de procesos hasta que su tiempo de ejecución sea 0.

**¿Qué es el quantum?**

## *Sistemas operativos.*

Es una medida de tiempo el cual se define en base a ciertas características del proceso, el cual nos permite asignar un tiempo justo a todos los procesos pendientes en la cola de procesos.

### **Conclusiones.**

Este trabajo nos da unas bases de como los procesos tienen que lidiar con la equidad y finites de los recursos de una computadora, lo cual es una parte critica de la computadora. Ya que ciertas tareas demandan más recursos que otras (incluyendo tiempo) es necesario para encontrar un equilibrio entre la rapidez y la eficiencia.

Su calificación final para este examen es 7.50/10.00.

No se permiten más intentos

Volver al curso

Programa 2 Procesamiento por lotes con  
multiprogramación)

Ir a...

Actividad 6 (Tarea 4) ►

*Ilustración 1. Resultados finales del Quiz.*

## **Bibliografía**

Contreras., A. M. (Junio de 2012). *Thread POSIX*. Obtenido de UNAM:

<http://lcomp89.fi->

[b.unam.mx/assets/ProgramacionConMultitareas/PracticaHilosPOSIX.pdf](http://b.unam.mx/assets/ProgramacionConMultitareas/PracticaHilosPOSIX.pdf)

Universidad Tecnica Federica Santa Maria. . (s.f.). *POSIX Threads*. Obtenido de

Universidad Tecnica Federica Santa Maria. :

[http://profesores.elo.utfsm.cl/~agv/elo330/2s07/lectures/POSIX\\_Threads.html](http://profesores.elo.utfsm.cl/~agv/elo330/2s07/lectures/POSIX_Threads.html)

