

**Universidad de Guadalajara**



**Centro Universitario de Ciencias Exactas e Ingeniería**

División de Tecnologías para la Integración Ciber-humana

**Materia:** Sistemas Operativos

**Profesor:** Violeta Del Rocío Becerra Velázquez

**Alumno:** Denice Estefania Rico Morones

**Código:** 219421171

**Carrera:** Ingeniería en Computación

**Sección:** D04

**Título:** Algoritmos de Planificación

**Fecha:** 22/08/24

**Contenido**

Hilos POSIX .....	3
Preguntas .....	4
Conclusión.....	6
Bibliografía .....	6

## Hilos POSIX

Los hilos POSIX también conocidos como Hilos Threads son una norma estándar que define una API para crear y gestionar hilos en sistemas operativos compatibles con POSIX como Unix y Linux; estos permiten poder controlar diversos procesos o flujos de trabajo a la vez (multiprogramación).

En esta misma biblioteca se asocian los procesos a nivel de núcleo; en el caso de Linux este no diferencia entre Hilos y Procesos; al tener múltiples hilos de nivel de usuario se asocian con procesos Linux y comparten el mismo ID de grupo.

Al tener un sistema multihilo podemos definir distintos hilos concurrentes en un solo proceso esto se puede lograr utilizando la misma cantidad de hilos de nivel de usuario como hilos de nivel de núcleo.

Existen 2 categorías en cuanto a la implementación de hilos:

- Hilos a nivel de núcleo o kernel: Esta implementación es mucho más rápida que una a nivel de usuario, el kernel por sí mismo no sabe que existen hilos.
- Hilos a nivel de usuario: estos son creados por el kernel y destruidos por el mismo. Normalmente se obtienen mejores resultados con estos.

El sistema operativo al desconocer la existencia de hilos de nivel de usuario estos se crean por medio de una biblioteca de hilos que se ejecutarán en un espacio de usuario de un proceso, estos hilos de nivel de usuarios son muy eficientes los cuales no requieren de un cambio de contexto esto para seguir ejecutando o cambiar de uno a otro. Si un hilo está a nivel de núcleo son reconocidos por el núcleo y múltiples hilos de un mismo proceso se ejecutan al mismo tiempo en un multiprocesador y si un hilo llega a ser bloqueado este no afecta al proceso completo.

Dentro de la biblioteca Pthreads existen las siguientes acciones:

- Crear, separar y unir los hilos.
- Obtener los atributos de los hilos.
- Manejar la exclusión mutua esto para que el acceso a algunas partes del código sea exclusivo.
- Sincronización entre hilos.

Funciones:

## Algoritmos de planificación

- Para crear un nuevo hilo tenemos la siguiente sintaxis:

### **pthread\_create()**

En esta función requerimos especificar un identificador del hilo, los atributos y la función que se ejecutará en el hilo.

- Unir hilos:

### **pthread\_join()**

gracias a esta función podemos sincronizar los hilos ya que éste permite a que otro hilo termine su ejecución, además podemos recuperar su valor de retorno.

- Identificar hilos:

### **pthread\_self()**

este devuelve el identificador del hilo.

- Terminar hilos:

### **pthread\_exit()**

esta función permite terminar la ejecución de un hilo de manera normal.

Los hilos también proporcionan concurrencia a nivel de proceso en comparación a cualquier otro proceso clásico esto nos ayuda a no requerir del uso de llamadas al sistema de una manera explícita.

## Preguntas

1. Describa en qué consisten los algoritmos de planificación Apropiativos.

Un algoritmo de planificación apropiativo son los cuales donde los procesos son obligados a salir por una razón o motivo como pudiera ser por el tiempo, después del tiempo fijo lo saca (esto lo llamamos RR Quantum), esto

es porque se forman como van llegando y el tiempo se divide de igual manera entre todos los procesos.

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

Un proceso por prioridades es de planificación apropiativa y son los más flexibles que hay y los que más se adaptan. Existen los **estáticos y dinámicos**.

**Los estáticos** no cambian, el sistema determina la prioridad más alta que puede estar asociada al número más alto esto ayuda a ordenarlos en base a una prioridad si va a ser primero ultimo.

**La dinámica** la prioridad sí puede cambiar haciendo posible que un proceso puede ejecutarse.

3. ¿En qué consiste el Algoritmo de Planificación de colas múltiples?

Estoy consiste en dividir los procesos en diferentes colas siguiendo ciertos criterios como la prioridad o el tipo de proceso, cada cual la puede tener su propio algoritmo de planificación.

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

**Bloqueado y suspendido** es cuando un proceso está esperando algún evento y gracias a esto no puede continuar su ejecución, hay un momento de que el proceso ha sido interrumpido se guarda su estado para poder ser reactivado más tarde.

**Listo y suspendido** es cuando un proceso está listo para ejecutarse, pero no tiene memoria asignada en ese momento.

5. ¿Cuál es el tiempo de Respuesta?

El tiempo de respuesta transcurre desde que un usuario envía una solicitud hasta que se recibe la primera respuesta del sistema.

6. Describa el algoritmo de planificación Apropiativo RR.

Este es un tipo de planificación apropiativa donde se asigna un tiempo de CPU fijo a lo cual llamamos quantum esto en cada proceso en una cola de listos.

7. ¿Qué es el Quantum?

## Algoritmos de planificación

Este es el tiempo máximo que un proceso puede utilizar la memoria o CPU antes de ser interrumpido.

## Conclusión

En conclusión puedo decir que es muy importante conocer todos estos asuntos y temas ya que son claves para conocer el funcionamiento de los distintos algoritmos que existen en cuanto a la ejecución de procesos, es importante saber identificarlos para saber con cuál nos conviene trabajar ya que algunos son más eficientes que otros ya que algunos tienen o aplican cierta prioridad.

## Bibliografía

- Campos Monge, E. M., & Campos Monge, M. (2023). *Sistemas operativos, Sistemas informáticos y Lenguajes de Programación*. España: RA-MA.
- Lezcano Brito, M. G. (2017). *Fundamentos de Sistemas Operativos Entornos de Trabajo*. Bogotá: Ediciones Universidad Cooperativa de Colombia.
- Llaven, D. S. (2015). *Sistemas operativos. Panomarama para ingeniería en computación e informática*. México: Patria.
- Rodriguez Garcia, J., Díaz Martín, J. C., & Irala Veloso, I. (s.f.). *PROGRAMACIÓN CONCURRENTE BASADA EN HILOS POSIX*. Escuela Politécnica. Departamento de Informatica. Universidad de Extremadura.
- Stallings, W. (2005). *Sistemas operativos. Aspectos internos y principios de diseño*. Madrid: Pearson Education.
- Wikipedia. (23 de Abril de 2024). *pthread*. Obtenido de Wikipedia: <https://es.wikipedia.org/wiki/Pthreads>