

Universidad de Guadalajara  
CUCEI (Ciencias Exactas e Ingenierías)  
Departamento de ciencias computacionales  
Materia: Sistemas operativos  
Profesor: Violeta del Rocío Becerra Velazquez

**Gómez Rubio Alexia**

Código:219551644

Carrera: Ingeniería en computación

Sección: D04

Tarea 4

Tema: Funcionamiento de un sistema operativo

Fecha: 22 Septiembre 2024

## Tabla de contenido

¿Qué son los hilos?.....	3
Características .....	3
Elementos que comparten los hilos .....	3
Elementos que no comparten los hilos .....	3
Los estados de un hilo .....	4
Funciones importantes.....	4
Pthread_create.....	4
Pthread_attr_get.....	4
Pthread_exit.....	4
Pthread_join.....	5
Pthread_cancel.....	5
Preguntas.....	6
Conclusión .....	8
Referencias bibliográficas .....	8

## ¿Qué son los hilos?

Un hilo es la unidad más pequeña de procesamiento que puede ser programada por el sistema operativo. Los hilos permiten ejecutar múltiples tareas en paralelo dentro de un proceso. En lugar de crear varios procesos independientes, los hilos permiten crear tareas concurrentes que comparten el mismo espacio de memoria y recursos dentro de un proceso, son la unidad mínima de procesamiento cuya ejecución puede ser programada por un sistema operativo, también se le llama proceso ligero.

En los procesos de POSIX de forma convencional: existen un hilo el cual ejecuta una sola función, este hilo tiene la capacidad de crear más hilos para así poder ejecutar otras que van a estar en el espacio de dirección de un proceso. Todos estos hilos se encontrarán en un solo nivel, esto quiere decir que son “hermanos” a diferencia de otros procesos la cuales están relacionadas como “padres-hijos”. El proceso de un hilo comparte las variables y los recursos que son globales del proceso.

### Características

- Permiten la ejecución concurrente de instrucciones asociadas a diferentes funciones dentro de un mismo proceso.
- Todos los hilos de un proceso comparten el mismo espacio de direcciones de memoria, archivos abiertos, variables globales, entre otros. Sin embargo, cada hilo tiene su propio conjunto de registros, contador de programa, y una pila de ejecución.
- Al crear varios hilos, es posible ejecutar tareas concurrentemente o en paralelo (dependiendo de la cantidad de núcleos del CPU).
- Las variables compartidas entre hilos pueden generar condiciones de carrera, por lo que se utilizan mecanismos de sincronización como semáforos, mutex y barreras.

### Elementos que comparten los hilos

- Código
- Variables de memoria globales
- Archivos o dispositivos (entrada y salida estándar) que tuviera abiertos el hilo padre

### Elementos que no comparten los hilos

- Contador de programa
- Registros del CPU
- Pila
- Estado del hilo

### Los estados de un hilo

- Listo: El hilo ya puede ser ejecutado, pero está esperando a un procesador.
- Ejecutándose: El hilo se está ejecutando; en sistemas multiprocesador pueden estar ejecutándose varios de forma simultánea.
- Bloqueado: El hilo no puede ejecutarse porque está esperando algo, por ejemplo, variables de condición o una operación de E/S para completarse.
- Terminado: El hilo termina cuando se llamó a la función `pthread_exit` o fue cancelado.

Es importante tener en cuenta que no hay sincronización cuando un hilo crea a otro hilo, es decir, el hilo creado con `pthread_create` puede comenzar a ejecutarse e incluso acabar antes de que se regrese de la función.

### Funciones importantes

#### `pthread_create`

Esta función es capaz de crear de forma inmediata el hilo, el cual estará en el estado de preparación, con los atributos `attr` dentro del proceso que es invocado y después devolverá a través de un thread su identificación. Si en algún caso el valor de `attr` es nulo los atributos se pasarán por defecto. En el caso de producirse un fallo, esta función retornara un número que será distinto de cero, esto indicara que se ha producido un error. En lo consiguiente el hilo ejecutara el código de rutina por `start_routine` a los que se pasan los argumentos referenciados por `arg`.

#### `pthread_attr_get`

Cada hilo estos asociados con un conjunto de atributos. Los atributos están representados por la siguiente variable `pthread_attr_t`.

Los atributos controlan lo siguiente:

- Vinculación (`detach_state`)
- Tamaño de la pila (`stacksize`)
- Dirección de la pila (`stackaddr`)
- Ámbito de contienda (`scope`)
- Herencia de la planificación (`inheritsched`)
- Política de distribución (`schedpolicy`)
- Parámetros de planificación (`schedparam`)

#### `pthread_exit`

Esta función es la que termina la ejecución del hilo ya invocado, con la función `value_ptr` es capaz de devolver los datos que están referenciados en ella, al

hilo el cual está en esperando su finalización. La finalización de la ejecución del proceso es terminada por el ultimo hilo. En caso de que el hilo sea sincronizado (joinable) este identificador del hilo retornara un valor el cual es capaz de ser examinado por otro hilo mediante la invocación `pthread_join` a través del parámetro de estado.

#### `pthread_join`

Esta función es la que suspende la ejecución del hilo y hace su invocación, el cual está identificado por el valor `tid` finaliza, tanto por la invocación de la función `pthread_exit` o por ya estar cancelado. Si el estado no es nulo, el valor que será devuelto por el hilo se almacenará en la dirección que estará indicada por el estado. El hilo cual se espera su terminación debe de estar en un estado (joinable state), cuando un hilo se encuentra en este estado de terminación, no liberara sus recursos hasta otro hilo espere por él. Esta espera por la terminación de un hilo a otro generara un error.

#### `pthread_cancel`

Es un mecanismo de cancelación por el cual un hilo puede pedir la terminación de otro que este en ejecución. El cual dependiendo de la configuración del hilo al que se le pide su cancelación las cuales pueden aceptar la solicitud de cancelación o rechazarla. Además de esta forma que existe de cancelación hay dos formas diferentes que son: forma asincrónica, forma diferida. Cuando un hilo acepta la cancelación seria como si invocara la siguiente función `pthread_exit`.

## Preguntas

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

Los algoritmos de planificación apropiativos, son también conocidos como expulsivos porque nos permiten la expulsión de procesos para ejecutar un nuevo proceso, poniendo en la cola al anterior.

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

Los procesos de mayor prioridad se ejecutan antes que aquellos con menor prioridad (la prioridad puede estar basada en factores como la importancia del proceso o el tiempo que dura el proceso).

Clasificación:

Estáticos, la prioridad de un proceso se asigna cuando comienza y no cambia.

Dinámicos, la prioridad del proceso puede cambiar durante su ejecución, aumentando su prioridad y evitar problemas de inanición.

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

Este algoritmo divide la cola de procesos listos en diversas colas, los procesos se pueden asignar de forma permanente a una de las colas de planificación, su objetivo es diferenciar entre distintos tipos de trabajos, y no permite el movimiento de los procesos entre las distintas colas.

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

Los procesos que estén en bloqueados y suspendidos, no pueden regresar a memoria principal a menos que primero pasen por bloqueados, después a listos y puedan continuar con su ejecución.

Los procesos que estén en listo y suspendidos, pueden regresar en automáticamente a listos y continuar con su proceso de ejecución (también pueden pasar a bloqueados, y después a listos, y continuar con su ejecución).

### **5. ¿Cuál es el tiempo de Respuesta?**

Es el tiempo que el proceso espera en la cola de listos, para poder empezar a ser ejecutado.

**6. Describa el algoritmo de planificación Apropiativo RR.**

Este algoritmo asigna a cada proceso un tiempo equitativo tratando a todos los procesos por igual y con la misma prioridad. Este algoritmo es circular, porque siempre regresa al primer proceso una vez terminando con el últimos.

**7. ¿Qué es el Quantum?**

El período de tiempo durante el cual se permite que un proceso se ejecute en un sistema multitarea preventivo generalmente se denomina porción de tiempo o quantum. El tiempo mínimo que se permitirá a un proceso el uso del procesador.

## Conclusión

Con la investigación que realicé sobre los hilos pude comprender mejor el tema, la verdad lo desconocía por completo, pero ahora entiendo que los hilos son una herramienta poderosa para mejorar el rendimiento de aplicaciones al permitir la ejecución concurrente o paralela de tareas. A través de los hilos, es posible dividir un proceso en múltiples subunidades que comparten el mismo espacio de memoria, lo que facilita la comunicación entre ellos.

## Referencias bibliográficas

*Algoritmos de planificación de procesos*. (n.d.). Jmoral.es. Retrieved September 16, 2024, from <http://jmoral.es/blog/planificacion-procesos>

Hilo, •. (n.d.). *A continuación se describen algunos conceptos importantes*: Unam.Mx. Retrieved September 18, 2024, from <http://lcomp89.fi-b.unam.mx/assets/ProgramacionConMultitareas/PracticaHilosPOSIX.pdf>

Guaman-QuincheSeguir, R. (n.d.). *Hilos con Posix*. SlideShare. Retrieved September 20, 2024, from <https://es.slideshare.net/slideshow/hilos-con-posix/251991106>

(N.d.-b). Vpnunlimited.com. Retrieved September 20, 2024, from <https://www.vpnunlimited.com/es/help/cybersecurity/round-robin>