

Proyecto de investigación - Hilos

David Esteban Monsalve Pulgarin

17 de julio de 2020

1. - ¿Qué es un hilo en el contexto de los micro-procesadores?

En sistemas operativos un hilo es un proceso ligero, subproceso o secuencia de tareas encadenadas muy pequeñas que puede ser ejecutada por un sistema operativo. La destrucción de los hilos antiguos por los nuevos surge de una característica que no permitía a una aplicación realizar varias tareas a la vez (concurrentemente). Los distintos hilos de ejecución comparten una serie de recursos tales como el espacio de memoria, los archivos abiertos, la situación de autenticación, etc. Esta técnica permite simplificar el diseño de una aplicación que debe llevar a cabo distintas funciones simultáneamente.

Se puede definir como el flujo de control de programa. Ayudan de forma directa a la manera en la que un procesador administra sus tareas. La función de los hilos se podría decir que hace que los ‘tiempos de espera’ entre procesos se aprovechen mejor. Los hilos son básicamente una tarea que puede ser ejecutada en paralelo con otra tarea; teniendo en cuenta lo que es propio de cada hilo es el contador de programa, la pila de ejecución y el estado de la CPU.

2. ¿Se puede hablar de la historia de los hilos?

Los hilos surgen de la necesidad de optimizar la realización de las tareas de un procesador pues antes los computadores solo contaban con un núcleo y un solo hilo entonces es simple como si fueras al supermercado, estas en la caja y se te olvida un producto el cajero (Procesador) no puede detener el resto de la fila porque se te olvidó algo, el debe seguir atendiendo a las demás personas de tal forma que la fila no se detenga cosa que cuando tu ya estes listo para pagar el finaliza la tarea y vuelve a ejecutar tu tarea y de esta manera actúan los hilos en un procesador. Aunque un núcleo solamente pueda realizar una tarea al mismo tiempo, se pueden usar los hilos para hacer creer al usuario (y al propio ordenador) que sí se puede hacer más de una cosa al mismo tiempo. ¿Y como es eso? Es muy simple: en vez de realizar una tarea por completo, divides la tarea en porciones (cada hilo se encarga de un aspecto concreto del programa), de modo que vas alternando entre porciones de tareas para que parezca que ambas se ejecutan al mismo tiempo.

Hilos una aparición temprana bajo el nombre de "tareas.^{en} OS / 360 multi-programación con un número variable de Tareas (MVT) en 1967. Saltzer (1966) Créditos Victor A. Vyssotsky con el término "hilo". Los programadores de proceso de muchos sistemas operativos modernos soportan directamente tanto tiempo en rodajas y multiprocesador roscado, y el núcleo del sistema operativo permite a los programadores para manipular los hilos mediante la exposición de la funcionalidad requerida por el sistema de llamada de interfaz. Algunas implementaciones de roscado se denominan hilos del núcleo , mientras que los procesos de peso ligero (LWP) son un tipo específico de hilo del núcleo que comparten el mismo estado y la información.

3. ¿Que tipo de hilos existen?

TIPOS DE HILOS Los Sistemas Operativos generalmente implementan hilos de dos maneras:

- Multihilo apropiativo: Permite al sistema operativo determinar cuando debe haber un cambio de contexto. La desventaja de esto es que el sistema puede hacer un cambio de contexto en un momento inadecuado, causando un fenomeno conocido como inversion de prioridades y otros problemas.
- Multihilo cooperativo: Depende del mismo hilo abandonar el control cuando llega a un punto de detencion, lo cual puede traer problemas cuando el hilo espera la disponibilidad de un recurso.

4. Implementacion de hilos

Los hilos se distinguen de los tradicionales procesos en que los procesos son –generalmente– independientes, llevan bastante información de estados, e interactúan solo a través de mecanismos de comunicación dados por el sistema. Por otra parte, muchos hilos generalmente comparten otros recursos de forma directa. En muchos de los sistemas operativos que dan facilidades a los hilos, es más rápido cambiar de un hilo a otro dentro del mismo proceso, que cambiar de un proceso a otro. Este fenómeno se debe a que los hilos comparten datos y espacios de direcciones, mientras que los procesos, al ser independientes, no lo hacen.

Hay dos grandes categorías en la implementación de hilos:

4.1. Hilos a nivel de usuario (UTL)

En una aplicación UTL pura, todo el trabajo de gestión de hilos lo realiza la aplicación y el núcleo o kernel no es consciente de la existencia de hilos. Es posible programar una aplicación como multihilo mediante una biblioteca de hilos. La misma contiene el código para crear y destruir hilos, intercambiar

mensajes y datos entre hilos, para planificar la ejecución de hilos y para salvar y restaurar el contexto de los hilos. Todas las operaciones descritas se llevan a cabo en el espacio de usuario de un mismo proceso. El kernel continua planificando el proceso como una unidad y asignándole un único estado (Listo, bloqueado, etc.).

4.1.1. Ventajas de los ULT

- * El intercambio de los hilos no necesita los privilegios del modo kernel, porque todas las estructuras de datos están en el espacio de direcciones de usuario de un mismo proceso. Por lo tanto, el proceso no debe cambiar a modo kernel para gestionar hilos. Se evita la sobrecarga de cambio de modo y con esto el sobrecoste u overhead.

- * Se puede realizar una planificación específica. Dependiendo de que aplicación sea, se puede decidir por una u otra planificación según sus ventajas.

- * Los ULT pueden ejecutar en cualquier sistema operativo. La biblioteca de hilos es un conjunto compartido.

4.1.2. Desventajas de los ULT

- * En la mayoría de los sistemas operativos las llamadas al sistema (System calls) son bloqueantes. Cuando un hilo realiza una llamada al sistema, se bloquea el mismo y también el resto de los hilos del proceso. En una estrategia ULT pura, una aplicación multihilo no puede aprovechar las ventajas de los multiprocesadores. El núcleo asigna un solo proceso a un solo procesador, ya que como el núcleo no interviene, ve al conjunto de hilos como un solo proceso.

- * Una solución al bloqueo mediante a llamadas al sistema es usando la técnica de jacketing, que es convertir una llamada bloqueante en no bloqueante.

4.2. Hilos a nivel de nucleo (KLT).

En una aplicacion KLT pura, todo el trabajo de gestion de hilos lo realiza el nucleo. En el area de la aplicacion no hay codigo de gestion de hilos, unicamente un API para la gestion de hilos en el nucleo. El sistema es quien provee la creacion, planificacion y administracion de los threads. El sistema reconoce tantos hilos de ejecucion como threads se hayan creado

Algunos lenguajes de programación tienen características de diseño expresamente creadas para permitir a los programadores lidiar con hilos de ejecución (como Java o Delphi). Otros (la mayoría) desconocen la existencia de hilos de ejecución y estos deben ser creados mediante llamadas de biblioteca especiales que dependen del sistema operativo en el que estos lenguajes están siendo utilizados (como es el caso del C y del C++).

5. Cibergrafias

https://www.youtube.com/watch?v=A1JjaAU_tS0

[https://es.wikipedia.org/wiki/Hilo_\(informC3A1tica\)](https://es.wikipedia.org/wiki/Hilo_(informC3A1tica))

<http://blog.drk.com.ar/2015/jugando-con-threads-en-c11>

[https://es.qwe.wiki/wiki/Thread_\(computing\)](https://es.qwe.wiki/wiki/Thread_(computing))