

Hilos

Trabajo de Investigación

Abdy Sánchez
Juan Pablo Vargas Mosquera

Departamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Medellín
Octubre de 2021

Índice

| | |
|--|---|
| 1. ¿Qué son los hilos? | 2 |
| 2. ¿Cómo funcionan los hilos? | 2 |
| 3. Ventajas de usar hilos | 2 |
| 4. Hilos a nivel de usuario y de kernel | 3 |
| 5. ¿Cómo se relacionan los hilos a nivel de kernel y los de usuario? | 3 |
| 6. Bibliografía | 4 |

1. ¿Qué son los hilos?

Un hilo es una unidad básica de utilización de CPU, la cual contiene un id de hilo, su propio program counter, un conjunto de registros, y una pila; que se representa a nivel del sistema operativo con una estructura llamada TCB (thread control block).

2. ¿Cómo funcionan los hilos?

Un hilo dentro de un procesador es la forma en que cada uno de los núcleos del CPU recibirá y procesará la información. Los hilos juegan un papel fundamental en la forma en que esta misma información es procesada. Esta tecnología está implementada en todos los CPU de sobremesa actuales. En Intel lleva el nombre de Hyper-Threading, y en AMD el de Simultaneous MultiThreading (SMT), que no es otra cosa que la implementación de un hilo virtual extra sobre cada uno de los hilos que alimentan a sus núcleos individuales dentro de estos componentes, la forma en que se comportan estos es sencillamente alimentando de instrucciones al CPU de manera cíclica en cada una de sus vueltas de reloj. Cada procesador tendrá que ir resolviendo las instrucciones que son recibidas a través de estos canales y, finalmente, serán resueltas bajo la prioridad que se le asigna mediante programación. Los hilos virtuales son ayudas de software que permiten que los hilos puedan mover información de manera mucho más eficiente hacia cada uno de los núcleos, donde serán administradas de manera cíclica y escalonada, pero con una velocidad mucho más óptima según sus capacidades de frecuencia le permitan. El añadir más núcleos a un CPU es la forma más sencilla de resolver siempre la brecha de rendimiento, pero esto también implica costos mucho más altos. Evidentemente, también de ahí nace la necesidad de separar los CPU por gamas, en las que los más caros, como el Ryzen 9 y el Core i9, son los que más núcleos tienen, y los más baratos, como el Core i3 y el Ryzen 3, los que menos.

3. Ventajas de usar hilos

- Respuesta: el tiempo de respuesta mejora, ya que el programa puede continuar ejecutándose, aunque parte de él esté bloqueado.
- Compartir recursos: los hilos comparten la memoria y los recursos del proceso al que pertenecen, por lo que se puede tener varios hilos de ejecución dentro del mismo espacio de direcciones.
- Economía: Es más fácil la creación, cambio de contexto y gestión de hilos que de procesos.
- Utilización múltiples CPUs: permite que hilos de un mismo proceso ejecuten en diferentes CPUs a la vez. En un proceso mono-hilo, un proceso

ejecuta en una única CPU, independientemente de cuantas tenga disponibles.

4. Hilos a nivel de usuario y de kernel

A nivel práctico los hilos pueden ser implementados a nivel de usuario o a nivel de kernel.

- **Hilos a nivel de usuario:** son implementados en alguna librería. Estos hilos se gestionan sin soporte del SO, el cual solo reconoce un hilo de ejecución.
- **Hilos a nivel de kernel:** el SO es quien crea, planifica y gestiona los hilos. Se reconocen tantos hilos como se hayan creado.

Los hilos a nivel de usuario tienen como beneficio que su cambio de contexto es más sencillo que el cambio de contexto entre hilos de kernel. Además, se pueden implementar aún si el SO no utiliza hilos a nivel de kernel. Los hilos a nivel de kernel tienen como gran beneficio poder aprovechar mejor las arquitecturas multiprocesadores, y que proporcionan un mejor tiempo de respuesta, ya que, si un hilo se bloquea, los otros pueden seguir ejecutando.

5. ¿Cómo se relacionan los hilos a nivel de kernel y los de usuario?

Existen 3 formas para establecer la relación.

- **Modelo Mx1 (Many to one).** El modelo asigna múltiples hilos de usuario a un hilo del kernel, tiene como inconveniente que, si un hilo se bloquea, todo el proceso se bloquea. También, dado que solo un hilo puede acceder al kernel cada vez, no podrán ejecutarse varios hilos en paralelo en múltiples CPUs.

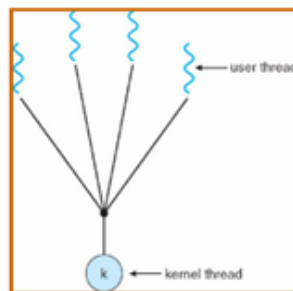


Figura 1: Modelo 1

- **Modelo 1x1 (one to one).** El modelo asigna cada hilo de usuario a un hilo del kernel. Proporciona una mayor concurrencia que el modelo anterior, permitiendo que se ejecute otro hilo si uno se bloqueó. Tiene como inconveniente que cada vez que se crea un hilo a nivel de usuario, se crea un hilo a nivel del kernel, y la cantidad de hilos a nivel del kernel están restringidos en la mayoría de los sistemas.

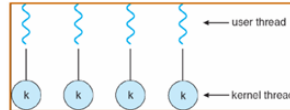


Figura 2: Modelo 2

- **Modelo MxN (many to many).** El modelo multiplexa muchos hilos de usuario sobre un número menor o igual de hilos del kernel. Cada proceso tiene asignado un conjunto de hilos de kernel, independientemente de la cantidad de hilos de usuario que haya creado. No posee ninguno de los inconvenientes de los dos modelos anteriores, ya que saca lo mejor de cada uno. El usuario puede crear tantos hilos como necesite y los hilos de kernel pueden ejecutar en paralelo.

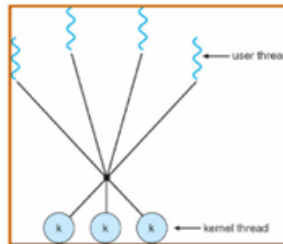


Figura 3: Modelo 3

6. Bibliografía

- Bibliografía: <https://www.fing.edu.uy/tecnoinf/mvd/cursos/so/material/teo/so05-hilos.pdf>
- <https://codigoespaguetti.com/noticias/tecnologia/como-funcionan-los-hilos-o-threads-de-un-procesador-cpu/>