

Los hilos en los microprocesadores

Camilo Alvarez Muñoz
camilo.alvarezm@udea.edu.co

16 de julio de 2020

Nuestras computadoras forman parte fundamental de nuestra vida cotidiana, son necesarias para realizar una gran cantidad de tareas diferentes, y conforme nuestra sociedad ha cambiado, también han cambiados nuestras necesidades, los procesadores han tenido que evolucionar para ser capaces de cumplir cualquier tipo de tarea o necesidad que nos sea requerida. Los procesadores son los cerebros de nuestros computadores, son aquellos que permiten al computador procesar y manejar los datos e instrucciones que dejan a los usuarios hacer muchas cosas, como editar fotos, leer un documento, ejecutar un programa o jugar un videojuego.

Un procesador se puede descomponer en subprocesadores, o conocidos comúnmente como núcleos (Cores), estos núcleos son unas unidades de procesamiento que nos permiten ejecutar y procesar las instrucciones que requiera el computador para realizar cierta tarea (Como utilizar un programa, escribir o escuchar un vídeo). Anteriormente se usaban procesadores con solo un núcleo, pero debido a la cantidad de tareas y necesidades que han surgido con el tiempo, se vio la necesidad de crear procesadores como muchos más núcleos para poder procesar distintos programas simultáneamente y así aumentar por mucho el rendimiento [García-Doncel, 2019]. Los núcleos reciben las tareas a realizar pero una parte fundamental y básica del procesador son los hilos que en realidad son los principales encargados de esa sensación de simultaneidad al momento de realizar bastantes tareas. Los hilos son secuencias o flujos de instrucciones que son necesarias para realizar un proceso. Para hacer una gran tarea, el procesador divide todo en tareas más pequeñas (hilos) que contienen pequeños trozos de instrucción y que son pasados al núcleo para ser ejecutados de forma alternada, al hacer esto simula la sensación de estar realizando tareas al mismo tiempo. Trabajar con hilos permite mejorar los tiempos de espera entre tareas, tener una forma más eficiente y rápida de realizar distintos procesos a la vez y un mejor manejo de recursos [García, 2017].

El concepto de hilos data su origen en la década de 1960, cuando conceptos como el Berkeley Timesharing System planteaban uso de los sistemas al mismo tiempo que distintos procesos se estaban ejecutando. En un principio los hilos eran conocidos por nombres como “tareas” o “procesos”, estos procesos tenían formas

de trabajar similares a las actuales. En 1970 sistemas como Unix desarrollaron fuertemente los hilos al introducir conceptos como el control por espacios de direcciones virtuales. El planteamiento hecho por Unix llevó al descontento a sus usuarios debido a que los procesos no compartían memoria, esto conllevó a que se solidificara el concepto de los hilos, procesos que permiten la multitarea y que compartían la misma memoria trabajando de forma simultánea y dinámica. Los principales desarrolladores como AMD o Intel se han encargado en los últimos 20 años a impulsar el concepto de multihilos y multinúcleos hasta el punto de ahora brindarnos los procesadores tan eficientes que usamos ahora [O'Sullivan, 2005].

Existen dos tipos de hilos, los hilos manejados a nivel de usuario (ULT), y los hilos manejados a nivel de Kernel (KLT). Los hilos a nivel de usuario son aquellos en el que la gestión de hilos es realizada por la aplicación, entonces el Kernel en ningún momento tiene noción de la existencia de estos hilos, se usan unas librerías o bibliotecas que nos permiten utilizar muchas funciones destinadas a la planificación y control de hilos, como crear hilos, destruirlos, crear preferencias, mandar mensajes y modificar datos, entre otros. Este tipo de hilos se pueden ejecutar en cualquier sistema operativo y no necesita los privilegios del Kernel, los ULT son mas fáciles y rápidos de usar pero también tienden a tener algunas desventajas como el no poder hacer uso de multihilos [Hacha, 2017].

La implementación de hilos por software depende de varios factores como el lenguaje utilizado y el procesador en uso. El tipo de procesador influye en la cantidad de hilos disponibles a usar (en las principales marcas suelen ser en general 2 por núcleo). Distintos lenguajes como Java, C y C++ tienen sus propias librerías destinadas para hilos, y también se usan librerías propias del sistema.

Los hilos a nivel de Kernel son aquellos únicamente manejados por el sistema operativo con llamadas al sistema, entonces él es el encargado de gestionar cosas como la distribución de los procesos en los núcleos, la cantidad de hilos usados, los privilegios y demás opciones. Este manejo de hilos presenta algunas ventajas como la posibilidad de tener múltiples hilos de un mismo proceso en distintos núcleos y la re planificación de hilos bloqueados [Hacha, 2017].

La forma en la que los hilos son implementados por hardware depende mucho del fabricante del procesador, distintos procesadores suelen contar con solamente un núcleo, y ahora la gran mayoría cuenta con hasta más de 4 núcleos. Estos procesadores que tienen más núcleos son los que permiten a nuestro computador realizar muchas tareas simultáneamente, entre mas núcleos tiene el procesador mayor cantidad de aplicaciones podremos usar. Al momento de hacer un uso de un programa en el computador, el procesador se encarga de crear un sistema o código multihilos, el procesador se encarga de distribuir los recursos del sistema a los di-

ferentes hilos, distribuye también los tiempos y órdenes en los que los hilos serán tratados, y brinda diferentes estados y funciones para un correcto control de los hilos. Las tácticas de manejo de hilos cada vez han sido más desarrolladas gracias a los principales distribuidores de procesadores como lo son AMD e Intel.

Referencias

[García, 2017] García, E. R. (2017). Núcleos e hilos en un procesador: qué son y en qué se diferencian.

[García-Doncel, 2019] García-Doncel, J. G. (2019). Núcleos e hilos de un procesador.

[Hacha, 2017] Hacha, F. T. (2017). Gestión de hilos de ejecución.

[O'Sullivan, 2005] O'Sullivan, B. (2005). The history of threads. [Sitio: Serpentine].