
PROYECTO DE INVESTIGACIÓN
HILOS

UNIVERSIDAD DE ANTIOQUIA

JUAN GUILLERMO QUEVEDO
CIFUENTES

FACULTAD DE INGENIERÍA

INFORMATICA II

2020

Actualmente los microprocesadores tienen varios núcleos, pero para que sirva esto, en pocas palabras entre más núcleos tenga un procesador puede hacer mas tareas al mismo tiempo, ya que se puede dividir las tareas entre los núcleos del procesador, veamos el procesador como una oficina y cada núcleo seria un trabajador, al cual se le asignan tareas, estas tareas se les asignan cuando terminan la que están ejecutando, entonces entre más núcleos hallan para una misma cantidad de tareas más rápido las terminaran. ¿Pero si quisiéramos ser más eficientes en los trabajos que tenemos que podemos hacer?, pues podemos usar hilos, son por así decirlo la fila de tareas que tienen los núcleos, además que cada núcleo puede tener dos de estos hilos (en la mayoría de casos) lo que hacen es que se puedan alternar las dos filas de tareas cada que exista un tiempo de espera en la fila principal, volvamos al ejemplo de la oficina, los hilos seria como un porta papeles en el cual cada trabajador tiene sus tareas en orden, los hilos seria estos porta papeles y se podría dar a cada trabajador 2 de estos, entonces el trabajador cada que tenga que hacer una pausa de las tareas del porta papeles principal, porque tenga que esperar a la memoria por ejemplo, ejecutara las instrucciones del otro portapapeles. De esta manera también tendremos mas capacidad de procesamiento en nuestros microprocesadores.[2][4]

La historia de los hilos empieza incluso antes de que existieran los procesadores multinúcleo y se puede decir que los hilos son los precedentes de estos últimos, esta tecnología fue traída por primera vez por Intel para los computadores personales, pero en otras aplicaciones ya tenia tiempo de estar en investigación. Intel la patento con el nombre de “Hyper-Threading”, la utilizo por primera vez en sus procesadores Xeon desde 2002, tres años antes de lanzar su procesador Pentium D que es su primer procesador dual Core, que consistía en dos Pentium 4 (que también contaban con la tecnología Hyper-Threading) dentro del mismo encapsulado.[3][1]

Existen dos tipos de hilos los hilos a nivel de usuario (“user-level thread” en inglés) y los hilos a nivel del kernel (“kernel-level thread” en inglés), el primero es completamente controlado por el usuario y el sistema operativo no es consiente de estos, las ventajas de estos es que no tienes que tener permisos del procesador ya que no lo tocas con estos hilos, además permite al usuario administrar los procesos para que sean personalizados y pueda optimizar la aplicación según su necesidad, pero su desventaja es que sufren del los

bloqueos del sistema operativo lo que bloque los tanto el hilo en uso como los demás entonces no te deja cambiar durante este bloqueo. En los hilos a nivel de kernel es el núcleo el que administra sus hilos, entonces el procesador puede decidir cuando es mas conveniente cambiar de hilo, además que si se bloquea un hilo el núcleo puede cambiar de hilo para continuar en la tarea. Aunque existen modelos híbridos “Los modelos de hilos híbridos (N: M) intentan combinar la eficiencia de los hilos de nivel de usuario con la funcionalidad de los hilos de nivel de núcleo. En un modelo híbrido, los subprocesos de nivel de usuario N se asignan a subprocesos de nivel de kernel M. En la práctica, esto ofrece un mayor grado de flexibilidad. El bloqueo de llamadas al sistema, por ejemplo, no implica necesariamente un bloqueo de subprocesos listos a nivel de usuario.” [5], pero esto no siempre es posible porque se necesita una compatibilidad compleja entre el usuario y el sistema.[5]

La implementación a nivel de hardware de los hilos dependerá del fabricante ya que dependiendo de las tecnologías con las que cuente será como fabricara su procesador con multihilo, pero para dar un ejemplo y ya que antes se ha hablado de Intel veamos como ellos solucionaron este problema. En los procesadores de Intel antes de el Hyper-Threading es que el núcleo cuando necesitaba información de la memoria necesitaba esperar a que el chipset que controlaba la memoria le pasara los datos que se necesitaban procesar, una vez que estos datos llegaban al núcleo este ultimo los procesaba más rápido de lo que el chipset le entregaba los siguientes datos generando tiempos muertos. Intel soluciono este problema duplicando este chipset, pero una versión mejorada, entonces cuando un chipset estaba esperando tener los datos de la memoria el otro estaba volcando la información en el procesador. Esta es la manera en la que Intel creo los hilos, pero los programas tienen que estar preparados para aprovechar los hilos, si no, no habría diferencia entre tener uno o dos hilos por núcleo.[3]

Entonces para aprovechar los hilos se tiene que implementar a nivel de software, pero para hacer esto hay que tener en cuenta con que sistema operativo y en cual lenguaje de programación se va a hacer, ya que de esto depende las librerías que haya que usar, la síntesis y los la forma de implementarlo. Ya que hay lenguajes los cuales están mas adaptados para usar hilos como java o Delphi, pero la mayoría necesita librerías dependiendo del sistema operativo para hacer uso de los hilos. Por ejemplo en c++ cuenta con la librería

“pthread”, para que el usuario pueda hacer uso de los hilos y poder crear programas que aprovechen mejor las capacidades del hardware disponible.[6]

Los hilos son una herramienta muy útil, que se utiliza hasta sin saberlos, pero también le permite a los usuarios crear programas mas eficientes, que aprovechen mejor los recursos. El uso de estos te puede permitir que tus programas den una mejor experiencia y aunque usarlos es opcional es muy recomendable hacer uso de ellos, cuando se tenga la posibilidad.

Referencias

- [1] Intel Corporation. Introduction to hyper-threading technology, 2010. Accedido el 16-07-2020 a <https://software.intel.com/content/www/xl/es/develop/articles/introduction-to-hyper-threading-technology.html?wapkw=Hyper-Threading>.
- [2] Elías Rodríguez García. Núcleos e hilos en un procesador: qué son y en qué se diferencian. *El Español*, Julio 2017. recuperado de https://www.elespanol.com/omicrono/tecnologia/20170707/nucleos-hilos-procesador-diferencian/229478224_0.html.
- [3] Wilmer Garzón-Alfonso. Micro arquitecturas de alto rendimiento.
- [4] Miguel Rebollo Pedruelo. El procesador. 2011.
- [5] Marc Rittinghaus. System call aggregation for a hybrid thread model, 2010.
- [6] Fco Javier Ceballos Sierra. *Programación orientada a objetos con C++*, volume 3. Grupo Editorial RA-MA, 2007.