

Proyecto de investigación - Hilos

Stiven Velásquez López

Julio 2020

1 Los hilos en los microprocesadores.

(6) Un hilo de procesamiento se define como el flujo de control de datos de un programa, los cuales permiten administrar las tareas de un microprocesador y de sus distintos núcleos con una mayor eficacia. Gracias a estos es posible dividir en segmentos los procesos o tareas para optimizar los tiempos de espera de cada instrucción en la cola del proceso. Estos segmentos se conocen con el nombre de subprocesos o threads.

Cada hilo de procesamiento abarca un trozo del proceso a ejecutar. Estos trozos de información se van introduciendo al núcleo físico, haciendo más simple el proceso a realizar, evitando así tareas ineficientes causadas por introducir en este caso la tarea completa en el núcleo. Así, la CPU es capaz de procesar una gran cantidad de tareas al mismo tiempo, y de manera simultánea.

2 Historia de los hilos.

(1) La idea de hilo, como un flujo secuencial de control inició en 1965 con el sistema Berkeley Timesharing. (8) En un principio los hilos se conocían comúnmente como “tareas” y este fue el nombre que se adoptó en “OS / 360 Multiprogramming with a Variable Number of Tasks (MVT)” en 1967

Estos procesos interactuaban con variables compartidas, semáforos y medios similares. Max Smith realizó un primer ejemplar de hilos en Multics en 1970; esto lo logró con ayuda de diversas pilas en un solo proceso pesado y así lograr resistir compilaciones de fondo. En el mismo año apareció unix.

La noción de unix de un “proceso” añadía a un hilo de control secuencial un espacio de direcciones virtuales(9). Estos procesos resultaban ser bastante pesados. No se compartía memoria (cada uno tenía su propio espacio de direcciones), su interacción se realizaba a través de tuberías, señales, etc. La memoria compartida se agregó un tiempo después.

Los usuarios de Unix empezaron a perder los procesos en los que se compartía memoria. Para solucionar esto se crearon los subprocesos, que compartían el espacio de direcciones de un único proceso de Unix. Siendo procesos livianos en comparación de los antiguos procesos pesados. (2)

Tiempo después aparecieron los hilos en los sistemas operativos como se conocen hoy en día, permitiendo a una aplicación realizar varias tareas a la vez de manera recurrente, estos hilos de ejecución compartían el espacio de memoria. Todo esto permitió hacer más simple el diseño de las aplicaciones que llevaban a cabo varias funciones simultaneas.

3 ¿Qué tipos de hilos existen?.

(4)

-Subprocesos a nivel de usuario: son subprocesos administrados por el usuario, más rápidos de crear, administrar y se desarrollan mediante una biblioteca de subprocesos a nivel de usuario. Este subproceso es genérico y puede ejecutarse en cualquier sistema operativo, su biblioteca de hilos tiene código para crear y destruir threads, para pasar mensajes y datos entre ellos, para programar ejecución de hilos, guardar y restaurar sus contextos. Estos hilos se gestionan sin soporte del sistema operativo, el cual sólo reconoce un hilo de ejecución. Los hilos a nivel de usuario tienen como ventaja que su cambio de contexto es más simple que el cambio de contexto entre hilos a nivel de núcleo. Además, se pueden implementar aún si el sistema operativo no usa hilos a nivel de núcleo.

-Subprocesos a nivel de núcleo: son subprocesos gestionados por el sistema operativo que actúan sobre el núcleo. Estos subprocesos son más lentos para crear y administrar que los hilos de usuario. Son compatibles con el sistema operativo, conservan información de contexto para el proceso en su conjunto y para los hilos individuales que hacen parte del proceso. Su programación se hace de acuerdo a subprocesos. El sistema operativo es quien crea, planifica y gestiona los hilos. Se reconocen tantos hilos como se hayan creado. Los hilos a nivel de núcleo tienen como ventaja que proveen un mejor tiempo de respuesta, ya que si un hilo se bloquea, los otros se pueden seguir ejecutando.(6)

4 Implementación de hilos a nivel de hardware.

Los subprocesos de hardware se pueden considerar como núcleos de CPU, sin embargo cada núcleo puede ejecutar diversos subprocesos. La mayoría de las CPU aclaran los subprocesos que se pueden ejecutar en cada núcleo. (5) Un “hilo de hardware” es una CPU física o núcleo. Así que, si una CPU de 4 núcleos acepta 4 subprocesos de hardware al mismo tiempo, la CPU está haciendo 4 cosas al mismo tiempo. Un hilo de hardware tiene la facilidad de ejecutar muchos hilos de software. En los sistemas operativos de la actualidad estos procesos se realizan dividiendo el tiempo: cada uno de los subprocesos tiene unos milisegundos para ejecutarse en la CPU, ya que el sistema operativo cambia de manera rápida entre los subprocesos; pudiese parecer que la CPU está realizando más de una tarea a la vez, pero en realidad un núcleo ejecuta sólo un subproceso de hardware que cambia entre muchos subprocesos de software.

5 Implementación de hilos a nivel de software

.(2) Los subprocesos de software son abstracciones del hardware para que se pueda dar el procesamiento múltiple. Estos hilos son de ejecución, gestionados por el sistema operativo. Si se tienen que realizar muchos subprocesos pero no se cuenta con suficientes recursos, los subprocesos de software son una manera de ejecutar todas y cada una de las tareas en paralelo mediante la asignación de recursos durante un tiempo limitado, para que parezca que todos los subprocesos se ejecutan en paralelo. Por ejemplo Windows puede iniciar procesos, cada uno de ellos tiene uno o más hilos. Estas tareas son gestionadas, ejecutadas y realizadas por el sistema operativo. Algunos lenguajes de programación tienen particularidades de diseño creadas para facilitar a los programadores enfrentar con hilos de ejecución (como java o Delphi). Otros lenguajes desconocen la existencia de hilos de ejecución y para crearlos se deben llamar librerías especiales que dependen del sistema operativo, en este caso tenemos a C y a C++. Cabe destacar que cada núcleo en un microprocesador es una unidad central de proceso separada e independiente, y entre más núcleos tenga un procesador se admitirá realizar una mayor cantidad de tareas simultáneas, cada unidad central hace una tarea específica. Lo que lleva a afirmar que en el hardware del procesador, entre mayor cantidad de núcleos haya, será mayor el rendimiento del equipo. Casi siempre se encontraran dos hilos por cada núcleo en un procesador. Y así el procesador con sus núcleos ejecutan las tareas y los hilos le acompañan en la distribución para lograr un mejor rendimiento.

6 Ejemplo de interrupción usando lenguaje c++.

Se realiza una implementación de Threads en c++, para ello se crea un subproceso para mostrar una lista grande de números en una ventana QDialog, esto se hace como un flujo de ejecución separado, ya que si no se utilizan los hilos el proceso del programa puede ser muy demorado mientras la aplicación logra procesar toda la información y pasar luego a mostrar los números en la pantalla. Se adjunta a la tarea la respectiva implementación.

References

- [1] Kleiman, S., Shah, D. and Smaalders, B., 1996. Programming With Threads. Englewood Cliffs, N.J.: Prentice-Hall.
- [2] Butenhof, D., 2008. Programming With POSIX Threads. Boston: Addison-Wesley.
- [3] Northrup, C., 1996. Programming With UNIX Threads. New York: John Wiley Sons.
- [4] Birrell, A., 1989. An Introduction To Programming With Threads. Palo Alto, Calif.: Digital Systems Research Center.

- [5] Bibliotecadigital.exactas.uba.ar. 2020. [online] Available at: <https://bibliotecadigital.exactas.uba.ar/download/tesis/tesisn6347GonzalezMarquez.pdf>; [Accessed 14 July 2020].
- [6] Ieeexplore.ieee.org. 2020. Performance Of Multi-Process And Multi-Thread Processing On Multi-Core SMT Processors - IEEE Conference Publication. [online] Available at: <https://ieeexplore.ieee.org/document/5650174>; [Accessed 15 July 2020].
- [7] Exabyteinformatica.com. 2020. [online] Available at: <https://www.exabyteinformatica.com/uoc/Informatica/Arquitecturasdecomputadoresavanzadas/Arquitecturasdecomputadoresavanzadas>; [Accessed 16 July 2020].
- [8] Dijkstra, EW, Procesos secuenciales cooperantes , en Lenguajes de programación , Genuys, F. (ed.), Academic Press, 1965.
- [9] Saltzer, JH, Control de tráfico en un sistema informático multiplexado , MAC-TR-30 (Tesis Sc.D.), julio de 1966.