Hilos en los sistemas operativos

Autora
Saray Cubillos García

July 17, 2020

Los hilos tanto internos como externos permiten directamente que el sistema operativo ejecute varias funciones al mismo tiempo haciendo referencia principalmente a la transferencia y el manejo de datos a gran escala.

1 ¿Qué es un hilo?

Un hilo es una unidad básica de utilización de CPU, esta contiene un identificador de hilo, su propio contador de programa, un conjunto de registros, y una pila; que se representa a nivel del sistema operativo con una estructura llamada TCB (thread control block). Los hilos comparten con otros hilos (Se pasan información entre sí) que pertenecen al mismo proceso la sección de código, la sección de datos, entre otras cosas. Si un proceso tiene múltiples hilos, puede realizar más de una tarea a la vez (esto es real cuando se posee más de un CPU).

Ventajas de usar hilos • Respuesta: el tiempo de respuesta se reduce, debido a que el programa puede continuar ejecutándose, aunque parte de él esté bloqueado. • Compartir datos: los hilos comparten la memoria y los recursos del proceso al que pertenecen, en este sentido 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 de múltiples CPUs: permite que hilos de un mismo proceso ejecuten en diferentes CPUs a la vez. En un proceso mono-hilo (hilo único), un proceso ejecuta en una única CPU, independientemente de cuantas tenga disponibles.

A nivel práctico los hilos pueden ser implementados a nivel de usuario o a nivel de kernel. Hilos a nivel de usuario: Estos hilos se gestionan sin soporte del Sistema Operativo, el cual solo reconoce un hilo de ejecución . 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 Sistema Operativo no utiliza hilos a nivel de kernel.

Algunos ejemplos de sistemas ULT son: GNU Portable Threads, FSU Threads, Apple Computer Thread Manager, REALbasics cooperative threads. Hilos a nivel de kernel: el Sistema Operativo es quien crea, planifica y gestiona los hilos. Se reconocen tantos hilos como se hayan creado. Otro de los beneficios consiste en poder planificar diferente a la estrategia del Sistema Operativo. Los hilos a nivel de kernel tienen como gran beneficio poder aprovechar mejor las arquitecturas multiprocesadores, las cuales proporcionan un mejor tiempo de respuesta, debido a que si un hilo se bloquea, los otros se pueden seguir ejecutando. Ventajas de los hilos a nivel kérnel: El proceso de usuario no se tiene que encargar de la planificación de los hilos . Si se cuenta con un procesador con más de un núcleo, el sistema operativo puede planificar los hilos en diferentes núcleos.

2 ¿Qué se puede hablar de la historia de los hilos?

Nos remontamos a 1965 con la máquina de tiempo compartido "Berkeley". En esta época no habían nombrado los hilos, pero hacían parte del proceso de ejecución, los procesos de interacción se daban a través de variables compartidas. Max Smith hizo un prototipo de hilos implementados alrededor de 1970 usando pilas múltiples en un proceso para ayudarse con compilaciones de segundo plano. Sin embargo se cree que el mayor progenitor de hilos es el lenguaje de programación PL/I . El lenguaje fue definido por IBM quien adjuntó un hilo para las "llamadas o invocaciones". No está claro cuándo fue implementado pero se examinó estrechamente mientras "Multics" estaba siendo diseñado se decidió que las tareas llamadas definidas no se ubicaban en el proceso, desde eso no hubo protección entre los hilos de control. Así que "Multics" tomó una dirección diferente y las tareas compartidas fueron removidas del PL/I por IBM.

Luego vino Unix, en los principios de 1970. El proceso trajo consigo unos hilos secuenciales de control con una dirección virtual en el espacio (Incidentalmente, el proceso de Unix se derivó directamente de el proceso diseñado de Multics [Saltzer, 66]). Por lo que "procesos" en el sentido de Unix son maquinas de peso. Desde entonces ellos no pudieron compartir memoria (cada uno tenía su propio espacio de dirección) ellos interactuaron a través de señales. Compartir memoria (además un mecanismo pesado) fue añadido mucho después. Después de un tiempo, Unix comenzó a perder el proceso viejo que podía compartir memoria. La "invención" de hilos: viejo estilo de procesos que comparten el espacio de dirección con un solo proceso de Unix. Ellos también fueron llamados "de poco peso", para contrastar con los proceos de "mucho peso" de Unix. Esta distinción se remonta a finales de 1970 o comienzos de 1980. Hasta el primer "microkernel" (Thoth (precursor de V-kernel y QNX), Amoeba, Chorus, la familia RIG-Accent-Mach,etc) En un dato aparte, los hilos han estado en continuos usos de aplicaciones de telecomunicaciones por un largo tiempo.

3 Tipo de hilos existentes y software

En Linux embebido el hilo es la entidad de planificación, distinguiendo tres clases: 1) Los hilos FIFO (First In First Out) de tiempo real tienen la mayor prioridad y no son apropiativos. 2) Hilos round robin de tiempo real, los cuales son apropiativos. ; tiene un quantum asociado 3) Hilos de tiempo compartido con menor prioridad; así como los hilos round robin tiene un quantum asociado.

Multiprocesos y multihilado: Varios procesos en varios hilos. Comportamiento de los hilos: Cuando un hilo necesita un recurso que otro hilo está ejecutando, este hilo debe esperar que el recurso se "desbloquee", mientras sigue con otros procesos. Desventajas de la multitarea y ejecución en el hardware: Múltiples hilos pueden interferir entre ellos, en especial cuando comparten recursos como las cachés. El multihilo soportado por hardware también está limitado en muchos procesadores. Aspectos íntimos de la ejecución multihilo. Hay una serie de aspectos a la hora de implementar un sistema de gestión de hilos

muy críticos como puede ser la sincronización entre hilos y que en algunas ocasiones dependen del hardware que se esté usando o bien de la plataforma de desarrollo (que versión de la maquina virtual Java se está usando por ejemplo) o incluso del mismo código que los desarrolladores de software crean. Los Sistemas Operativos en general implementan hilos de dos maneras: · Multihilo apropiativo: Permite al sistema operativo determinar cuándo debe haber un cambio de contexto. · Multihilo cooperativo: Depende del propio hilo abandonar el control cuando llega a un punto de detención, lo que puede traer problemas cuando el hilo espera la disponibilidad de algún recurso. El soporte de hardware para multihilo desde hace poco se encuentra disponible. Esta característica fue introducida por Intel en el Pentium 4, bajo el nombre de HyperThreading.

Hilos de Sistemas operativos M.1 Un proceso define un espacio de dirección y la propiedad sobre los recursos es dinámica. Dentro de este mismo proceso pueden crearse y ejecutarse múltiples hilos.(Implementación de Windows NT, MACH.) 1:M Un hilo es capaz de migrar de un ambiente de proceso a otro. Esto permite que un hilo se pueda mover fácilmente entre distintos sistemas.. Objeto de hilo de Windows: Existe una asignación de prioridad, se hereda del padre. Cada proceso corre con unas garantías de seguridad, estas se cargan con un "token" y los hijos heredan esa seguridad.

4 ¿Cómo se hace la implementación de hilos a nivel de hardware?

Dependiendo de la plataforma, a veces los hilos de usuario inclusive utilizan multitarea cooperativa para pasar el control dentro de un mismo proceso. Cualquier llamada al sistema bloqueante (como recibir datos de un archivo para utilizarlos de forma inmediata) interrumpirá la ejecución de todos los hilos de ese proceso, debido a que el control de ejecución es entregado al sistema operativo quien en este caso no sabe nada sobre hilos.

Hay tres patrones en los que se diferencian en general los modelos de hilos; puede emplearse más de uno de estos patrones en distintas áreas de cada aplicación, e incluso pueden anidarse (se podría tener una línea de ensamblado dentro de la cual uno de los pasos sea un equipo de trabajo):

Jefe/trabajador: Un hilo tiene una tarea distinta a la de los demás: el hilo jefe genera o recopila tareas para realizar, las separa y entrega a los hilos trabajadores. Este es el proceso más común que implementan servidores (modelo clásico del servidor Web Apache) y para aplicaciones gráficas (GUI), en que hay una parte del programa (el hilo jefe) esperando a que ocurran eventos de índole externa. El jefe realiza poco trabajo se limita a administrar y los hilos trabajadores realizan su operación, posiblemente notifican al jefe de su trabajo, y terminan su ejecución. Equipo de trabajo al iniciar la porción miltihilos del proceso, se crean muchos hilos idénticos, que realizarán las mismas tareas sobre diferentes datos. Este modelo es frecuentemente utilizado para cálculos matemáticos (criptografía,render,álgebra lineal). Puede combinarse con un estilo jefe/trabajador para irle dando al usuario una previsualización del resultado del cálculo, debido a que éste se irá ensamblando progresivamente, porción por porción. Su diferencia más notoria es sobre

el patrón jefe/trabajador consiste en que el trabajo a realizar por cada uno de los hilos se plantea desde el comienzo, esto significa, que el paso de división de trabajo no es un hilo más, sino que prepara los datos para que éstos sean ejecutados en paralelo. Estos datos no son resultado de eventos independientes (como en el anterior caso), si no partes de un único caso. Como resultado, resulta natural que en este modelo las consecuencias generadas por los distintos hilos son agregados o totalizados al terminar su procesamiento. Los hilos no acaban, sino que se sincronizan y después continúan la ejecución lineal. Línea de ensamblado Si una tarea larga puede dividirse en pasos sobre bloques de la información en total para procesar, cada hilo puede enfocarse a hacer un solo paso y pasarle los datos a otro hilo mientras va terminando. Una de las grandes ventajas de este modelo es que ayuda a mantener rutinas sencillas de comprender, y permite que el procesamiento de datos continúe, incluso si parte del programa está bloqueado esperando E/S. Un punto de importancia para tener en cuenta en una línea de ensamblado es que, si bien los hilos trabajan de manera secuencial, pueden estar ejecutándose de forma paralela sobre bloques consecutivos de información, eventos, etc...

References

REFERENCIAS BIBLIOGRAFICAS http://ocw.uc3m.es/ingenieria-informatica/sistemas-operativos/material-de-clase-1/mt$_t$2$_l$5.*pdf* $Parra, L.(2012).MICROPROCESADORES.http$: $//www.aliat.org.mx/BibliotecasDigitales/sistemas/Microprocesadores.pdfRodriguezGarcia, E.(2017, ju$ $qués onyenqués edi f erencian.ElEspañol.https : //www.elespanol.com/omicrono/tecnologia/20170707/nuc$ $hilos-procesador-di f erencian/229478224_0.htmlCarreteroPérez, J., Mateos, A.C., Daniel, J., Sánchez, G.$ $Lección5 : HilosyProcesosfiIntroducciónyconceptosbásicos.Andrés, M., Ramos, H., Nacional, U., Distanc$