

30 mayo 2023

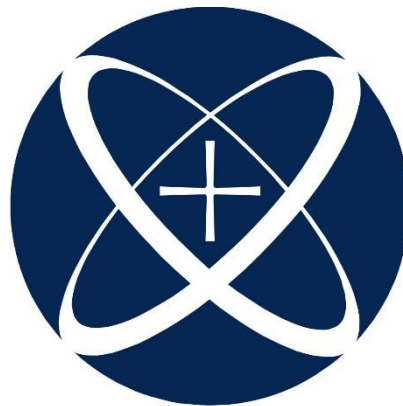
Actividad 5

Estructura de los sistemas de cómputo y
del sistema operativo

Fundamentos de Sistemas Operativos

ESI032B

Mtro. José Luis Elvira



ITESO

Universidad Jesuita
de Guadalajara

Laura Griselda González Camacho

Marco Ricardo Cordero Hernández

ING. EN SISTEMAS COMPUTACIONALES

1. ¿Cuál es la ventaja de crear hilos contra crear procesos?

La principal ventaja es que los hilos comparten recursos de memoria y archivos con el proceso padre, por lo que no se deben duplicar. Por la misma razón de memoria compartida, la comunicación y sincronización son más rápidas entre hilos.

2. ¿Por qué crear hilos es una estrategia para sacar provecho de arquitecturas con múltiples CPUs?, explica cuál es el papel del sistema operativo para que esto sea posible.

La creación de hilos permite la ejecución paralela de tareas independientes entre ellas o que utilizan diferentes recursos, de manera que se utilizan varios core de CPU para ejecutar cada uno una parte de un proceso y tener de manera más rápida los elementos para el siguiente paso de la ejecución del proceso, utilizando todos los recursos posibles. El sistema operativo hace esto posible a través del planificador de kernel, que selecciona un hilo a un core.

3. Considerando las estructuras de datos PCB, ¿qué es necesario agregar para que un proceso pueda tener múltiples hilos?

Se necesita agregar un campo de identificación del hilo, un contexto de ejecución, una cola o lista en el PCB, información del estado del hilo y recursos compartidos (memoria, semáforos, etc.)

4. En todos de los sistemas operativos modernos, por cada thread (hilo) de ejecución es necesario que haya un stack. Explique por qué es necesario que haya un stack por cada hilo y por qué no se puede compartir.

Es necesario que haya un stack independiente por cada hilo, puesto que de esa manera los hilos pueden llamar a funciones haciendo uso de variables locales, sin afectar a otros hilos, causar interferencia o conflictos. De esa manera, también cada hilo puede tener su estado de ejecución independiente junto con su posición actual en código.

5. ¿Cuál es la diferencia de los hilos a nivel kernel con los hilos a nivel usuario?

La diferencia principal es que los hilos a nivel kernel son más eficientes, pero más complejos de utilizar y se requieren más privilegios para su uso, mientras que los hilos a nivel usuario son más fáciles de utilizar y no aprovechan en su totalidad la paralelización del hardware.

6. ¿Por qué el usar hilos a nivel usuario no permite obtener el máximo desempeño en un procesador Multicore?

Esto se debe a que el planificador de kernel realmente sólo toma en cuenta un hilo por proceso de kernel. Mientras no haya múltiples hilos a nivel de sistema operativo, que estén asignados a diferentes cores del procesador, sólo se usará un solo núcleo.

7. En el estudio de los ULT frente a los KLT, se apuntó que una desventaja de los ULT que consiste en que cuando un ULT realiza una llamada al sistema, no se bloquea sólo ese hilo, sino todos los hilos del mismo proceso. ¿por qué?. Explica cuales son los problemas que se presentan en los KLT cuando estos hacen llamadas bloqueantes

Como el sistema operativo no conoce la manera en la que están funcionando los hilos a nivel usuario, se maneja solo por hilos kernel, sin un mapeo todos estarían corriendo sobre

un mismo procesador y al no saber que hilos de usuario se podrían seguir ejecutando sin conflictos en la espera de una entrada o salida bloquea todos.

8. Explica cómo se da el mapeo de los hilos a nivel usuario con los hilos a nivel kernel y como los hilos a nivel kernel con los procesadores y/o núcleos de un sistema.

Mapeo de hilos a nivel de usuario a hilos a nivel kernel:

Cuando se crea un hijo a nivel usuario el sistema operativo asigna uno o más hilos a nivel kernel para ejecutarlo, siendo uno a uno o muchos a uno. El sistema operativo se encarga de la sincronización y coordinación entre los hilos para su correcta ejecución.

Hilos a nivel kernel con procesadores y/o núcleos del sistema:

Una vez que los hilos a nivel usuario se asignan a los hilos a nivel kernel, estos se planifican y ejecutan en los procesadores o núcleos del sistema. La planificación y asignación de hilos a procesadores y/o núcleos se realiza mediante algoritmos de planificación del SO.

9. Menciona un ejemplo diferente al que vimos en las presentaciones de una aplicación que puede ser optimizada usando hilos.

Un programa para procesamiento de imágenes, existiendo distintos filtros dentro de la aplicación este se puede distribuir para que el hilo principal sea el que procese el archivo o imagen, destinando hilos secundarios para procesar los filtros.

10. Define el concepto de afinidad.

La afinidad de un proceso o hilo determina si este se debe mantener corriendo en un mismo procesador en lugar de estarse intercambiando entre múltiples procesadores por cuestiones de balanceo de carga.

11. Busca información en Internet sobre la llamada al sistema clone() en Linux y explica cuáles son los argumentos que esta llamada recibe.

La llamada al sistema clone() crea un proceso hijo. Permiten al proceso hijo compartir elementos de su contexto de ejecución con el proceso padre. Su principal uso es para implementar hilos. En caso de éxito regresa el PID del hijo en el hilo de ejecución del proceso padre. Puede recibir varios parámetros:

Un puntero a una función que será ejecutada por el nuevo proceso o hilo.

Un puntero al inicio del stack del nuevo proceso o hilo.

Banderas que controlan el comportamiento de clone() y determinan si se crea un nuevo proceso o hilo, y los recursos que va a compartir con los procesos hijos.

Un puntero que se pasa como argumento de la función que se pasa, para pasar datos adicionales.

12. En el sistema operativo Linux, un hilo es creado clonando el proceso que realiza la llamada `pthread_create()`, es decir, un hilo es un proceso clonado en el cual se comparte casi todo el espacio en memoria. Por lo tanto indique:

a) ¿Qué segmentos de memoria del proceso que se deben compartir y qué no?

Lo que no se comparte entre hilos son los stack, cada hilo debe tener su stack. Los datos y el código se deben compartir.

b) Aun así es demostrado que crear hilos llega a ser hasta 20 veces más rápido que crear proceso, explique porqué.

Porque comparten contexto y segmentos de memoria y eso ocasiona que se usen menos recursos, por lo que se hace más rápido.

13. ¿Qué aprendiste?

Aprendimos los conceptos fundamentales de los hilos en los sistemas operativos. También analizamos las diferencias entre crear procesos hijos con la función `fork()` y con la función `clone()`. La segunda es la que crea los hilos, y estos comparten contexto de ejecución y segmentos de memoria con su proceso padre lo que los hace más rápido. Con los conocimientos adquiridos y reforzados con esta actividad estamos preparados para comenzar a ver la parte práctica de los hijos.