ITESO

Universidad Jesuita de Guadalajara

Axel Roberto Orozco Hernández
Edgar Alan Torres Tovar

Fundamentos de Sistemas Operativos
Profesor: Leonardo Sandoval. Gonzalez

Actividad 5

1.- ¿Cuál es la ventaja de crear hilos contra crear procesos?
La principal ventja de crear hilos en lugar d eprocesos es que los hilos comparten el mismo espacio de direcciones, lo que permite comunicación y sincronizacion mas eficiente. Esto se traduce en un menor uso de recursos del sistema y timepos de creacion y cambio de contexto mas rapidos, ya que no es necesario duplicat el espacio de memoria.

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.
Crear hilos permite que multiples tareas de un proceso se ejecuten en paralelo en distintas CPUs o nucleos, mejorando significativamente el desempeño en arquitecturas multicore. El sistema operativo es responsable de gestionar la asignacion de hilos alos CPUs mediante un planificador que distribuye los hilos de forma eficiente, maximizando el uso de los recursos de procesamiento.

3.- Considerando las estructuras de datos PCB, ¿qué es necesario agregar para que un proceso pueda tener múltiples hilos?
Para que un proceso pueda tener multiples hilos, es necesario incluir una estructura que mantenga la informacion de los hilos, como el Thread Control Block (TBC). Esto incluye registros de contexto, stack indivifual para cada hilo y punteros que enlacen los TBCs al PCB del proceso principal.

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 que no se puede compartir.
Cada hilo necesita su propio stack porque el stack almacena datos variables locales, direcciones de retorno y registros de funcion especificos de la ejecucion del hilo. Compartir un stack entre hilos llevaria a una sobreescritura de datos y resultados imprecindibles, ya que los hilos pueden estar ejecutando diferentes funciones o partes del codigo al mismo timepo.

5.- ¿Cuál es la diferencia de los hilos a nivel kernel con los hilos a nivel usuario?
Los hilos a nivel kernel son gestionados directamente por el sistema operativo, lo que permite un mejor manejo de multiples nucleos y programacion simultanea.

6.- ¿Por qué el usar hilos a nivel usuario no permite obtener el máximo desempeño en un procesador Multicore?
Los hilos a nivel usuario no permiten obtener el maximo desempeño en un procesador multicore porque el sistema operativo no tiene visibilidad sobre ellos y, por tanto, no puede programarlos de manera efectiva en diferentes nucleos.

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
El ULT, cuando un hilo realiza una llamada al sistema bloqueante, todo el proceso se bloquea porque el jernel no distingue entre los hilos del proceso. En cambio, en KLT, el kernel puede gestionar los hilos de forma independiente, ede modo que cuendo un hilo realiza una llamada bloqueante, solo ese hilo se bloquea. Sin embargo, los KLT pueden experimentar sobrecarga adicional debido al manejo del kernel y al cambio de contexto mas costoso en comparacion con los ULT.

8.- Explica cómo se da el mapeo de los hilos a nivel usuario con los hilos a nivel kérnel y como los hilos a nivel kérnel con los procesadores y/o núcleos de un sistema.
Existen diferentes:

- Modelo Uno a Uno: directamente a un hilo kernel.
- Modelo de muchos a uno: varios hilos a un solo hilo kernel.
- Modelo de muchos a muchos: mejor gestion y aprovechamiento del paralelismo.

9.- Menciona un ejemplo diferente al que vimos en las presentaciones de una aplicación que puede ser optimizada usando hilos.
Un servidor web puede ser optimizado usando hilos para manejar multiples solicitudes de clientes simultaneamente.

10.- Define el concepto de afinidad.
Preferencia de un proceso o hilo de ejecutarse en un procesador o nucleo especifico.

11.-Busca información en Internet sobre la llamada al sistema clone() en Linux y explica cuáles son los argumentos que esta llamada recibe.
Clone es usada para crear un nuevo hilo.
Hay varios argumentos:

- ‘fn’: puntero a la funcion que ejecutara el nuevo proceso.
- ‘ child_stack ‘: puntero al stack del nuevo hilo.
- ‘ flags ‘: define que recursos compartir entre el padre y el hijo, como memoria, archivo descriptor, espacio de direcciones, etc.
- ‘ arg ‘: argumento pasado a la funcion ‘ fn ‘.

12.- En el sistema operativo Línux, 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?

Se comparten los segmentos de codigo, datos y heap, asi como descriptores de archivo, variables globales y el espacio de direcciones.
b) Aun así es demostrado que crear hilos llega a ser hasta 20 veces más rápido que crear proceso, explique porqué.
La creacion de hilos es mas rapida porque no requiere duplicar el espacio de memoria completo, solo se asigna un nuevo stack y regsitros de contexto.

13.- ¿Qué aprendiste?
Sobre la importancia de los hilos y procesos en los sistemas operativos, y como lo shilos permiten aprovechar las arquitecturas multicore. Tambien las diferencias clave entre hilos a nivel usuario y a nivel kernel, y la relevancia del sistema operativo en la gestion de recursos.

Bibliografias:

- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10th ed.) Wiley.
- Lover, R. (2013). Linux System Programming. O'Reilly Media.
- Stallings, W. (2018). Operating Systems: Internals and Design Principles (9th ed.). Pearson.