

CURSO PARA LA OBTENCIÓN DEL

DIPLOMA DE INFORMÁTICA MILITAR

(59130)

Teoría de Sistemas Operativos

Capítulo 4. Multithreads y concurrencia

Índice

[1. Conceptos sobre procesos 4](#_Toc146230160)

[1. Se dice que existe una pseudosimultaneidad cuando: 4](#_Toc146230161)

[2. Respecto a multithreading: 4](#_Toc146230162)

[3. Se dispone de un ordenador monoprocesador que permite multiprogramación, para lo cual es preciso: 4](#_Toc146230163)

[4. En relación con los Sistemas Operativos, ¿Qué se entiende por overhead? 4](#_Toc146230164)

[5. Los beneficios de la programación multiproceso pueden ser (marcar la respuesta FALSA): 5](#_Toc146230165)

[6. Un kernel de un buen sistema operativo multinúcleo dispone de programas que deben resolver gestiones de multithreding como (SEÑALE LA FALSA): 5](#_Toc146230166)

[7. La tecnología HyperThreading consiste en 5](#_Toc146230167)

[8. Una tecnología que “engaña” al sistema operativo al hacerle creer que existen dos núcleos cuando en realidad solo existe uno 6](#_Toc146230168)

[9. Aplicando la Ley de Amdahl, supongamos que tenemos una aplicación que es 50 por ciento paralela y 50 por ciento serie. Si ejecutamos esta aplicación en un sistema con dos núcleos de procesamiento, podemos obtener una aceleración de: 6](#_Toc146230169)

[10. Aplicando la Ley de Amdahl, supongamos que tenemos una aplicación que es 50 por ciento paralela y 50 por ciento serie. Si ejecutamos esta aplicación en un sistema con cuatro núcleos de procesamiento, podemos obtener una aceleración de: 6](#_Toc146230170)

[11. El soporte para subprocesos se puede proporcionar a nivel de usuario, o por el kernel 6](#_Toc146230171)

[12. El modelo de gestión de subprocesos del SO de tipo muchos a uno: 6](#_Toc146230172)

[13. El modelo de gestión de subprocesos del SO de tipo uno a uno: 7](#_Toc146230173)

# Preguntas

## Se dice que existe una pseudosimultaneidad cuando:

1. Una CPU con un solo núcleo tiende que compartir su tiempo de proceso entre varias tareas, tal que en un instante de tiempo está ejecutando varios instrucciones de programa a la vez
2. El SO organiza y calcula los hilos de forma óptima tal que varios usuarios (al frente de diferentes procesos) tienen la impresión de estar atendidos a la vez por una CPU mononúcleo
3. Una CPU multinúcleo donde se procesan varios tareas de forma simultánea.
4. Una CPU con varios microprocesadores donde se procesan varios tareas de forma simultánea

## Respecto a multithreading:

1. La concurrencia o multithreading de una aplicación depende exclusivamente de la gestión del SO
2. El efecto de concurrencia o multithreading de una aplicación depende exclusivamente del programador que la crea.
3. La simultaneidad real de una aplicación creada con hilos o tareas concurrentes necesita que la CPU sea multinúcleo.
4. Una CPU multinúcleo siempre producirá mayor velocidad de proceso que una mononúcleo al ejecutar cualquier aplicación

## Se dispone de un ordenador monoprocesador que permite multiprogramación, para lo cual es preciso:

1. Utilizar técnicas de multiplexación en el tiempo virtual
2. Utilizar técnicas de ejecución de programas en tiempo real
3. Utilizar técnicas de ejecución en tiempo compartido
4. Fragmentar de forma lógica la memoria caché en tantas partes como programas en ejecución tengamos

## En relación con los Sistemas Operativos, ¿Qué se entiende por overhead?

1. Porcentaje de tiempo que transcurre entre la cabecera (header) del primer paquete y el siguiente
2. Porcentaje de tiempo en el que la CPU está atendiendo a tareas del propio sistema
3. Tiempo que tarda la cabeza del disco, una vez posicionada en la pista, en acceder al sector que le interesa
4. Porcentaje de tiempo que la CPU gasta en tratar los datos de entrada

## Uno de los beneficios de la programación multiproceso (concurrente) pueden ser (marcar la respuesta FALSA):

1. El uso de subprocesos múltiples en una aplicación interactiva puede permitir que un programa continúe ejecutándose incluso si parte de él está bloqueado o está realizando una operación prolongada, lo que aumenta la capacidad de respuesta para el usuario.
2. Supongamos que en un GUI de una aplicación un usuario, al hacer clic en un botón de una pantalla, se desencadena una operación que requiere mucho tiempo de proceso. Una aplicación de thread único no respondería al usuario que quedaría bloqueado hasta que se haya completado la operación. En una aplicación creada con multithreding también habría que esperar a que ese duradero subproceso termine para avisar al usuario con un Thread que puede seguir introduciendo texto.
3. El beneficio de compartir código y datos permite que una aplicación tenga varios subprocesos de actividad diferentes dentro del mismo espacio de direcciones.
4. la creación de subprocesos consume menos tiempo y memoria que la creación de procesos.

## Un kernel de un buen sistema operativo multinúcleo dispone de programas que deben resolver gestiones de multithreding como (SEÑALE LA FALSA):

1. Identificar en las aplicaciones a ejecutar los hilos programados que podrían ser ejecutadas en paralelo en los núcleos individuales de la CPU
2. Resolver la dependencia de una tarea respecto a los datos de otra asegurándose de que las tareas están sincronizadas para adaptarse a la dependencia de los datos
3. Balancear la carga computacional de diferentes tareas concurrentes para optimizar el tiempo de ejecución de la aplicación.
4. Cuando un programa se ejecuta en paralelo en múltiples núcleos, son posibles muchas rutas de ejecución diferentes. Probar y depurar tales programas concurrentes es inherentemente más fácil que probar y depurar aplicaciones de un solo subproceso.

## La tecnología HyperThreading consiste en

1. ‘simular’ dos procesadores (o núcleos físicos) en uno solo lógico, dividiendo la carga de trabajo entre ambos y por tanto mejorando la velocidad de procesamiento.
2. Es una lógica que Apple ya había implementado desde hace tiempo que consiste en duplicar algunos elementos dentro del procesador como los registros o las memorias caché de tercer nivel
3. Podemos ver procesadores de 2, 4 , 6 u 8 núcleos físicos de Intel, que gracias a la tecnología HyperThreading, pueden simular el doble de núcleos de procesamiento.
4. Los procesadores Intel Core i3 poseen solo dos núcleos lógicos pero se comporta como si tuviese 4 núcleos físicos gracias a esta tecnología.

## Una tecnología que “engaña” al sistema operativo al hacerle creer que existen dos núcleos cuando en realidad solo existe uno

1. Es el HyperThreading de las máquinas Macintosh
2. Intel fue pionera con su tecnología HyperThreading,
3. Es la llamada CMT (Cluster Based Multithreading) de los procesadores AMD
4. Es la llamada SMT (Simultaneous Multi-Threading) de Intel

## Aplicando la Ley de Amdahl, supongamos que tenemos una aplicación que es 50 por ciento paralela y 50 por ciento serie. Si ejecutamos esta aplicación en un sistema con dos núcleos de procesamiento, podemos obtener una aceleración de:

1. 1,6 veces
2. 1,3 veces
3. 1,2 veces
4. 2,6 veces

## El soporte para subprocesos se puede proporcionar a nivel de usuario, o por el kernel

1. Subprocesos (hilos) a nivel kernel son administrados directamente por la CPU
2. Subprocesos (hilos) a nivel usuario son administrados directamente por el kernel,
3. Linux y macOS, admiten subprocesos del kernel.
4. Prácticamente todos los sistemas operativos contemporáneos admiten subprocesos del kernel, excepto Windows

## El modelo de gestión de subprocesos del SO de tipo muchos a uno:

1. Asigna muchos subprocesos a nivel de usuario (compilador) a un subproceso del kernel y la gestión de subprocesos la realiza la biblioteca de subprocesos en el espacio del SO.
2. Todo el proceso se bloqueará si un subproceso realiza una llamada al sistema de bloqueo.
3. Como cualquier subproceso puede acceder al kernel, múltiples subprocesos pueden ejecutarse en paralelo en un sistemas multinúcleo.
4. Es el mejor modelo para aprovechar los múltiples núcleos de una CPU multicore.

## El modelo de gestión de subprocesos del SO de tipo uno a uno:

1. Asigna cada subproceso de usuario a un subproceso de la CPU
2. Permite que varios subprocesos se ejecuten en paralelo en multiprocesadores siempre que la creación de un subproceso de usuario se corresponda con la creación del subproceso del kernel correspondiente
3. Linux, junto con la familia de sistemas operativos Windows, no implementan el modelo uno a uno.
4. Multiplexa muchos threads a nivel de usuario en un número mayor de threads del kernel.

# Ejercicios

## Comandos sobre información de CPU

* Pruebe a ejecutar los comandos de Linux que se describen al final del capítulo relacionados con la información de la CPU.
* Haga una breve explicación e interpretación de los principales resultados obtenidos
* Añada otros comandos con variaciones de los anteriores (utilizando otros parámetros)
* Añada otros comandos con relación a la información de procesos y cores, como:

sudo apt-get install cpufrequtils

cpufreq-info

sudo apt-get install cpuid

cpuid

Si se tiene un Intel core i3 i5 i7

sudo apt-get install i7z

sudo i7z

sudo apt-get install inxi

inxi -C

sudo apt-get install hardinfo

hardinfo

## ¿Cómo sabemos si tenemos activado el Hyper Threading?

El Hyper Threading es una característica que hay que activar desde la BIOS pero para entrar, hay que reiniciar el sistema y eso es algo que no siempre podemos hacer cuando queramos.

Con comandos de Linux también podemos saber si está activo el Hyper Threading sin necesidad de rebotar el sistema. Hay varios métodos para ello:

* Con el comando *lscpu*:

[lscpu |grep -i thread

* Con el comando *dmidecode*:

dmidecode -t processor | grep Count

* Revisando el archivo /sys/devices/system/cpu/smt/active:

cat /sys/devices/system/cpu/smt/active

## ¿Cómo contar el número de Threads que está utilizando un proceso de Linux?

Una vez que tenemos el Hyperthreading activado, podemos comprobar si los procesos están utilizando multithreading, si están programados para ello.

En Linux, encontraremos esta información en /proc/PID/status:

cat /proc/13248/status |grep -i thread