

Sistemas Operativos II

Dennis Josué Valverde Gómez 20190111782

Tarea 1.

Mapa conceptual capítulo 10.1 Multiprocessor and Multicore Scheduling Clasificación de multiprocesadores

Multiprocesador débilmente acoplado o distribuido, cluster.

Cada procesador con su memoria principal y canales de I/O. Procesadores especializados funcionalmente.

Hay un procesador maestro que controla los procesadores especializados.

Multiprocesador fuertemente acoplado.

Conjunto de procesadores que comparten memoria principal y están controlados integralmente por el sistema operativo.

Granularidad o frecuencia de sincronización entre procesos

Categorías de paralelismo

**Fino.** Paralelismo inherente a un único flujo de instrucciones.

Uso más complejo del paralelismo en el uso de hilos.

Instrucciones: < 20

Medio. Procesamiento multitarea en una sola aplicación. Debido a que varios hilos de una aplicación interactúan con tanta frecuencia la planificación puede afectar el rendimiento total.

Instrucciones: 20-200

**Grueso.** Multiprocesamiento de procesos concurrentes en una multiprogramación ambiente.

Instrucciones: 200-2,000

**Muy grueso.** Procesamiento distribuido entre nodos de la red para formar un único entorno informático.

Instrucciones: 2,000-1M

Independiente. No hay sincronización explícita entre procesos, cada uno representa una tarea por separado.

Instrucciones: No aplica.

Problemas de diseño

#### Asignación de procesos.

### Estática

Para cada procesador se mantiene una cola. La desventaja es que un procesador puede estar inactivo mientras otro tiene un backlog.

### Dinámica

Los hilos se mueven de una cola para un procesador a una cola para otro procesador, esto es utilizado por Linux.

# Multiprogramación en procesadores individuales.

En el multiprocesador que trata con granularidad gruesa o independiente cada procesador debería cambiar entre varios procesos para un alto uso y mejor rendimiento. Pero para aplicaciones de grano medio que se ejecutan en varios procesadores ya no es primordial que cada uno esté ocupado, más bien se necesita el mejor rendimiento promedio para las aplicaciones.

#### Despacho de procesos.

En un uniprocesador multiprogramado, el uso de prioridades o algoritmos de planificación sofisticados basados en el uso pasado puede mejorar el rendimiento sobre una estrategia simple de primero en llegar, primero en ser atendido. Cuando consideramos multiprocesadores, estas complejidades pueden ser innecesarias o incluso contraproducentes, y un enfoque más simple puede ser más efectivo con menos sobrecarga. En el caso de la planificación de hilos, surgen nuevos problemas que pueden ser más importantes que las prioridades o los historiales de ejecución.



#### Carga compartida.

Los procesos no se asignan a un procesador específico, se mantiene una cola global de hilos listos y cada procesador cuando está inactivo, selecciona un hilo de la cola.

#### Planificación en grupo

Un conjunto de hilos relacionados se programa para ejecutarse en un conjunto de procesadores al mismo tiempo, de manera uno a uno.

# Asignación de procesador dedicado.

Esto es lo opuesto al enfoque de compartir la carga y proporciona una programación implícita definida por la asignación de hilos a procesadores. Cada programa, durante su ejecución, se le asigna un número de procesadores igual al número de hilos en el programa. Cuando el programa termina, los procesadores regresan al conjunto general para su posible asignación a otro programa.

#### Planificación dinámica:

El número de hilos en un proceso puede modificarse durante el curso de la ejecución.

## **Multicore Thread Scheduling**

Es necesario asignar hilos a núcleos de manera eficiente para aprovechar la caché compartida y reducir el acceso a la memoria externa. El uso cooperativo de recursos permite que varios hilos accedan a las mismas ubicaciones de memoria. En la contención de recursos los hilos compiten por las ubicaciones de memoria caché.

La planificación consciente de la contención busca maximizar la eficacia de la memoria caché compartida y minimizar la necesidad de accesos a la memoria externa.