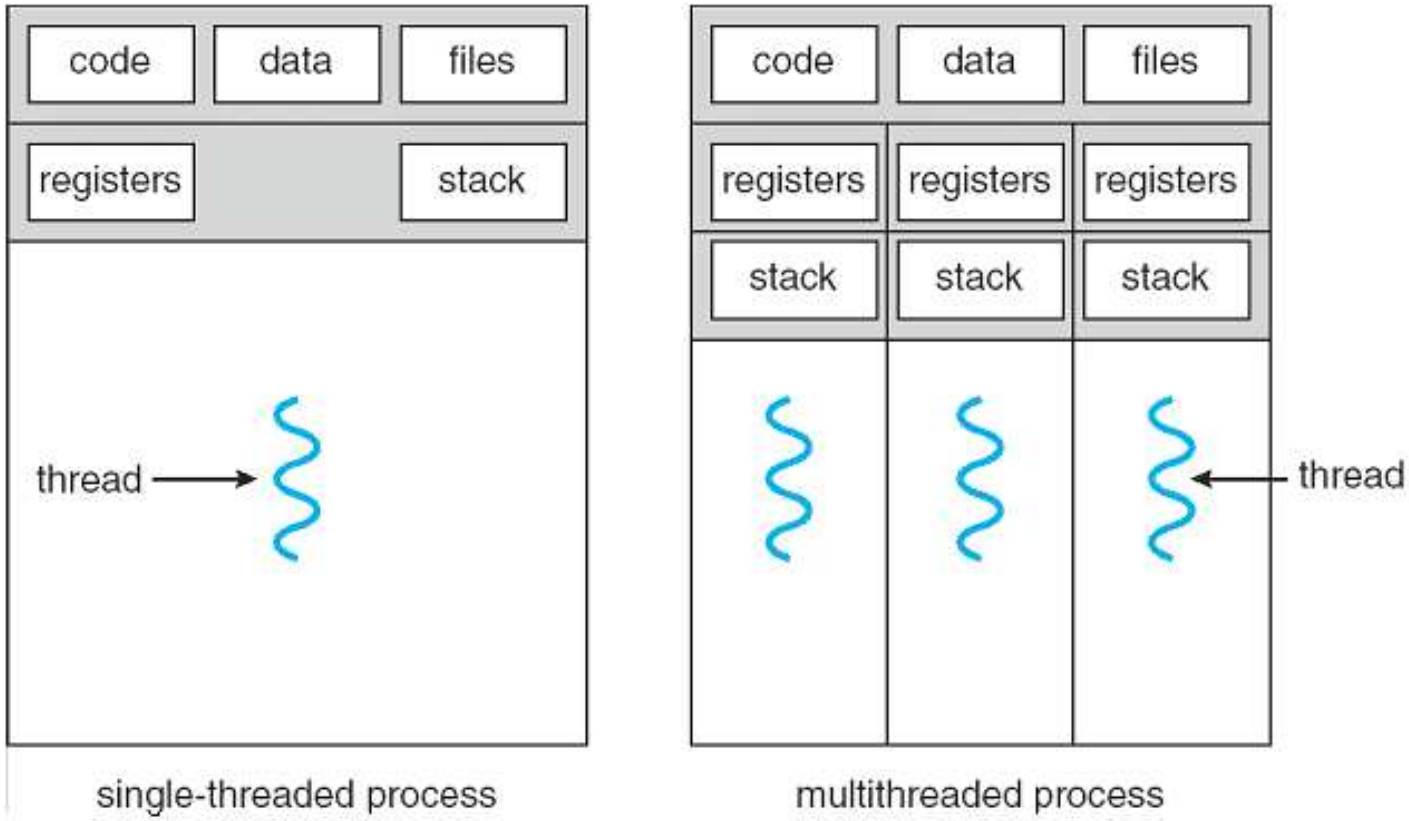
Threads

Hilos - Lightweight process - Procesos ligeros

Concepto



Los threads comparten el espacio de direcciones del proceso

Ganancia

Si un hilo se bloquea para lanzar E/S el SO puede elegir otro hilo del mismo proceso para continuar ejecutando.

Esto provoca una ganancia en cuanto al cambio de contexto (Transición de dominio) ya que el vaciado de la cache es mucho menor entre hilos que entre procesos (por ejemplo, UNIX 1,8 milisegundos y en TOPAZ con hilos 0,4 milisegundos)

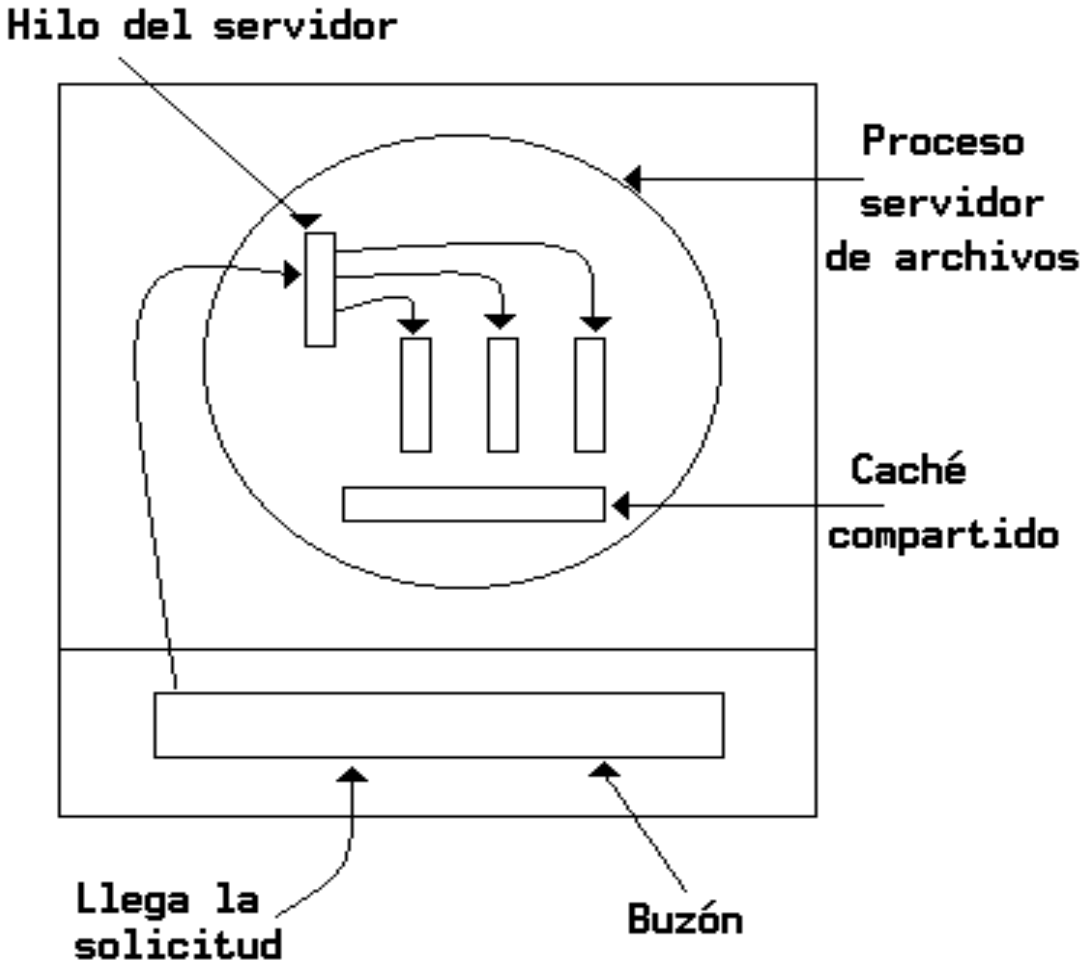
El cambio de contexto de los registros también es menor ya que no es necesario cambiarlos a todos.

Elementos por hilos y por proceso

Elemento	**PROCESO**	**THREAD**
Espacio de direcciones	Le pertenece	Lo comparten
Stack/Pila	Le pertenece	Cada hilo tiene propio
Registros	Le pertenecen	Cada hilo tiene su propio conjunto
Archivos abiertos	Le pertenecen	Los comparten
Reloj	Le pertenece	Lo comparten
Variables globales	Le pertenecen	Las comparten
Hijos	Procesos hijos	Hilos hijos
Estado	Tiene propio	Cada hilo tiene el suyo
Protección entre	Existe	Usualmente no existe ya que cooperan entre sí
Señales	Le pertenecen	Las comparten

Estructuras de implementación de hilos (1)

Estructura Servidor-Trabajador

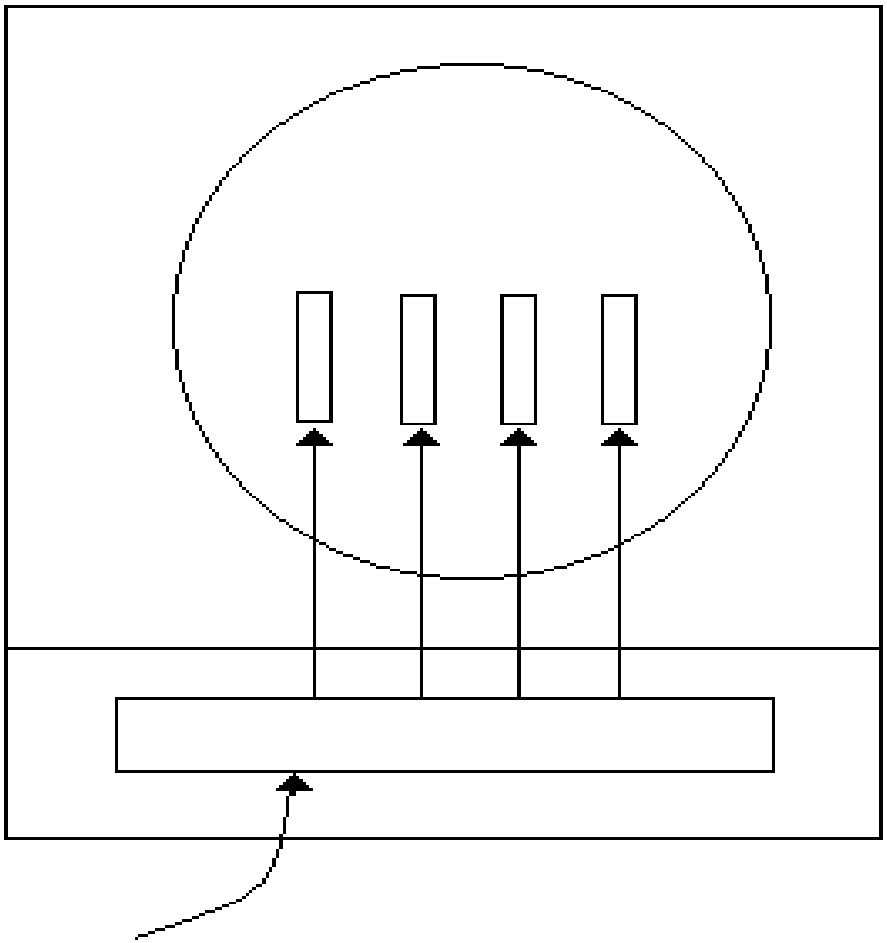


Un hilo hace las funciones de servidor despachando los pedidos y los otros hilos de trabajadores

Uso: servidor de archivos

Estructuras de implementación de hilos (2)

Estructura en Equipo

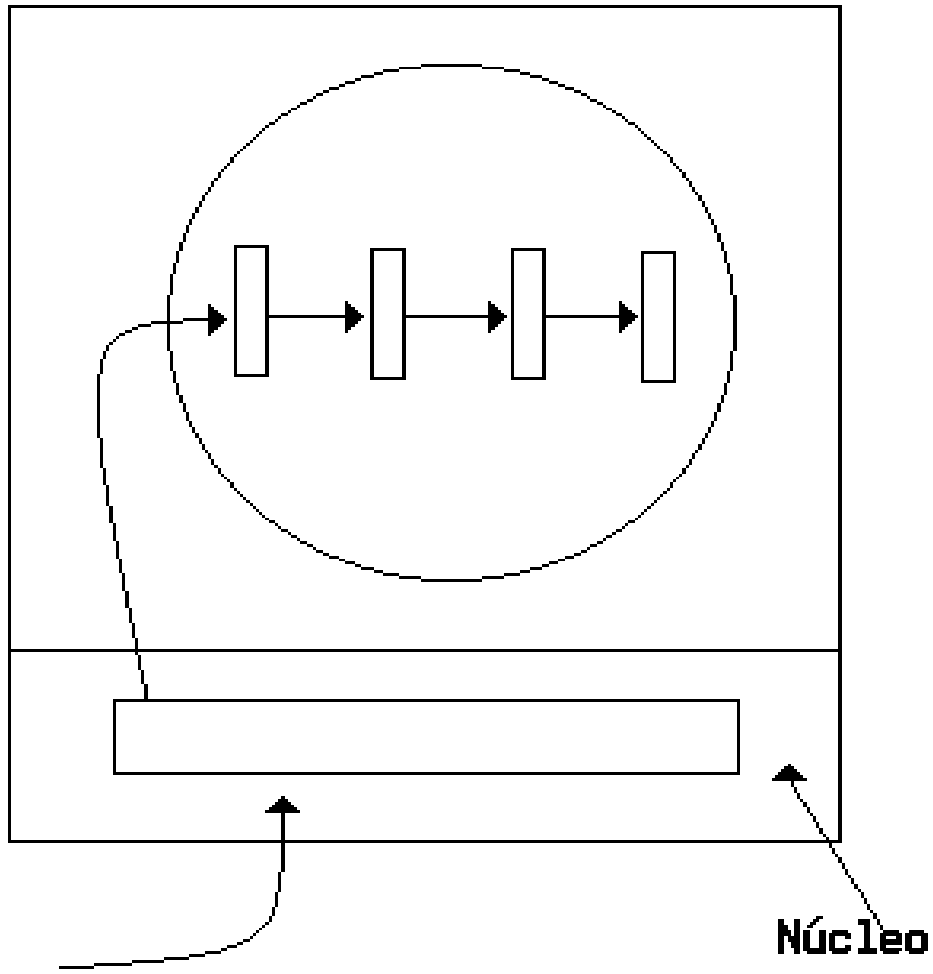


Todos los hilos son iguales y toman los pedidos del buzón de entrada

Uso: servidor de archivos

Estructuras de implementación de hilos (3)

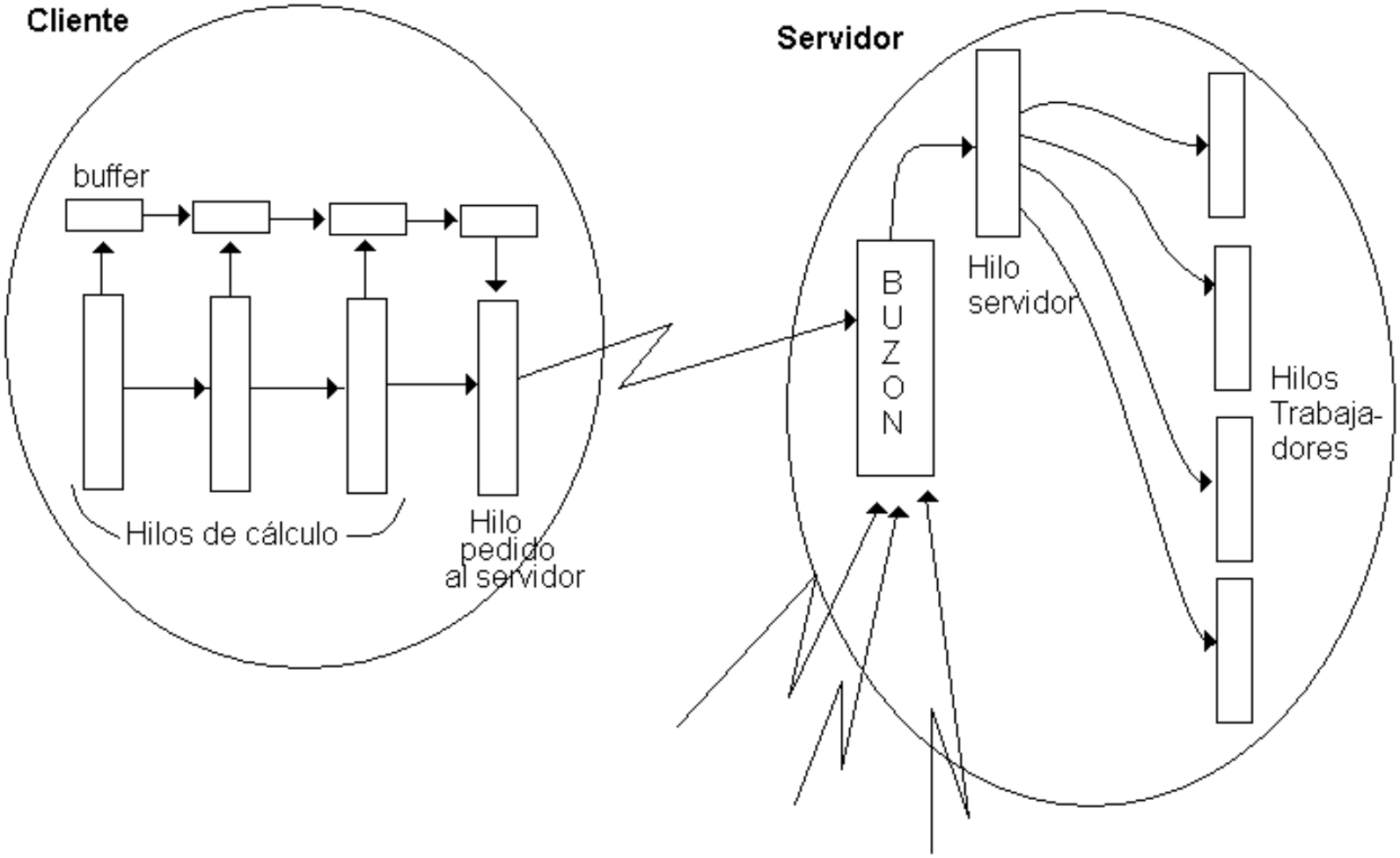
Estructura Pipeline



Lo procesado por un hilo es input del siguiente hilo

Uso: problemas de tipo Productor-Consumidor

Un ejemplo de implementación de threads

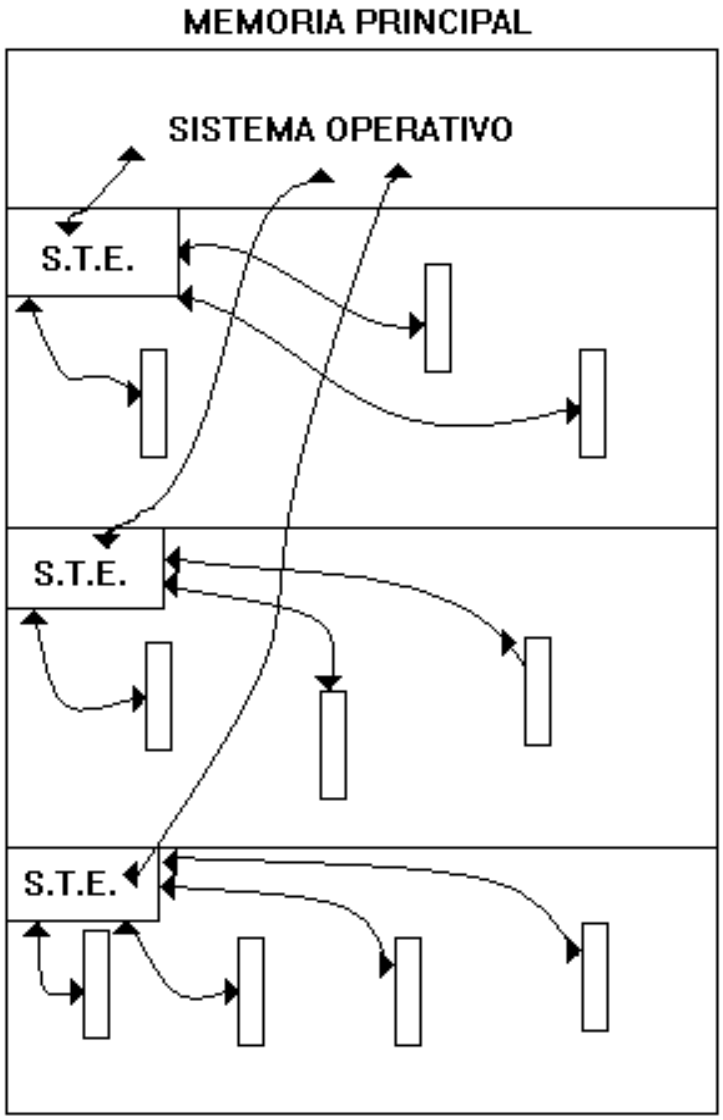


Reconocimiento de Threads

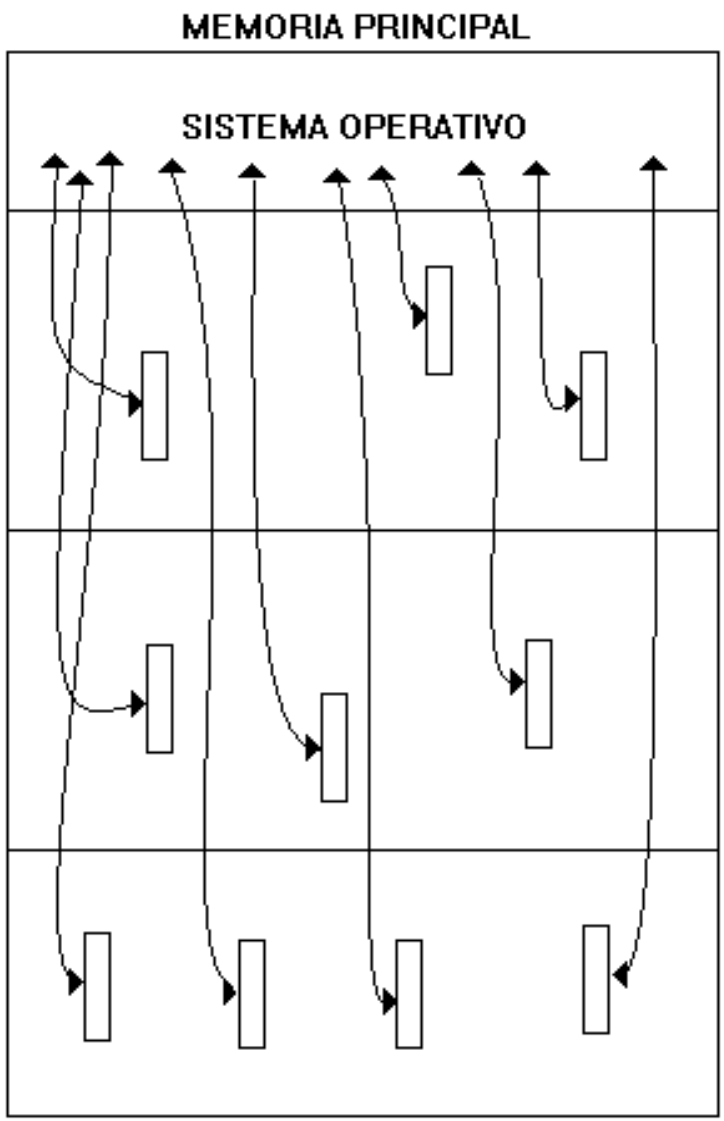
Pero no todos los sistemas operativos reconocen la existencia de threads.

Los que los reconocen los implementan en el núcleo - Implementación en el kernel (ej: Windows XP/2000, OS/2, Solaris, MACH, CHORUS, Linux, Tru64 UNIX, Mac OS X)

Para aquellos que no los reconocen se pueden implementar threads a nivel del espacio de direcciones del proceso - Implementación en el espacio del usuario (ej: UNIX)



Implementación en el espacio del usuario



Implementación en el núcleo

Implementación en el espacio del usuario

Dentro del espacio del usuario existe un Sistema de Tiempo de Ejecución (STE) que se encarga de administrar los hilos del proceso.

El STE se encarga de interceptar las llamadas bloqueantes de los hilos (por ej: lanzamiento de E/S) y también realiza el intercambio del procesador entre los hilos (planificación - scheduling)

Implementación en el espacio del usuario
Ventajas y desventajas

- Si un hilo se bloquea para E/S bloquearía a todo el proceso, para solucionarlo se implementan Jackets (implica reescribir parte de las rutinas de la biblioteca de llamadas al sistema)

- Cada proceso puede tener una planificación de threads diferente

- El reloj es único al proceso y por ende una interrupción por reloj bloquea a TODO el proceso y sus threads

- No puede aprovechar la ganancia del multithreading si se implementa en un sistema multiprocesador

Implementación en el espacio del usuario

Ventajas y desventajas (continuación)

- Una interrupción por falla de página (el thread necesita cargar más código en memoria) bloquea a todo el proceso

- El intercambio de contexto entre hilos es de una magnitud menor que en la implementación a nivel del kernel

- Todas las llamadas al sistema pasan por el STE

- Es más escalable que la implementación a nivel del kernel ya que toda la información de administración reside en el espacio del usuario

Implementación en el núcleo

El sistema operativo reconoce los threads.

Ventajas y Desventajas:

- El intercambio de contexto tiene un mayor costo que en la implementación a nivel del usuario

- Requiere de mayor espacio en el núcleo para las tablas y la pila de los hilos => No es tan escalable

- Todas las llamadas al sistema las maneja el núcleo y por lo tanto tiene un mayor costo para él

- La planificación es uniforme para todos los hilos

- Se puede planificar por reloj (quantums)

- No es necesario el uso de jackets

Problemas generales de los hilos

Las variables globales

Pertenecen al proceso y por lo tanto si un hilo las usa y luego pierde el control esa variable puede corromperse por el uso de otro hilo.

Soluciones:

- Prohibir su uso o

- diseñar variables globales propias de cada hilo (algunos lenguajes no lo permiten) o

- reescribir los procedimientos de biblioteca para poder manejarlas

Problemas generales de los hilos

Las Señales

Dos hilos pueden desear capturar una señal de teclado para realizar acciones distintas.

Ya es difícil su manejo en ambientes monothreading y en multithreading el tema se complica aún más.

Solución:

- Dedicar un solo hilo para realizar esta función