
4: Threads

Sistemas Operativos 1
Ing. Alejandro León Liu



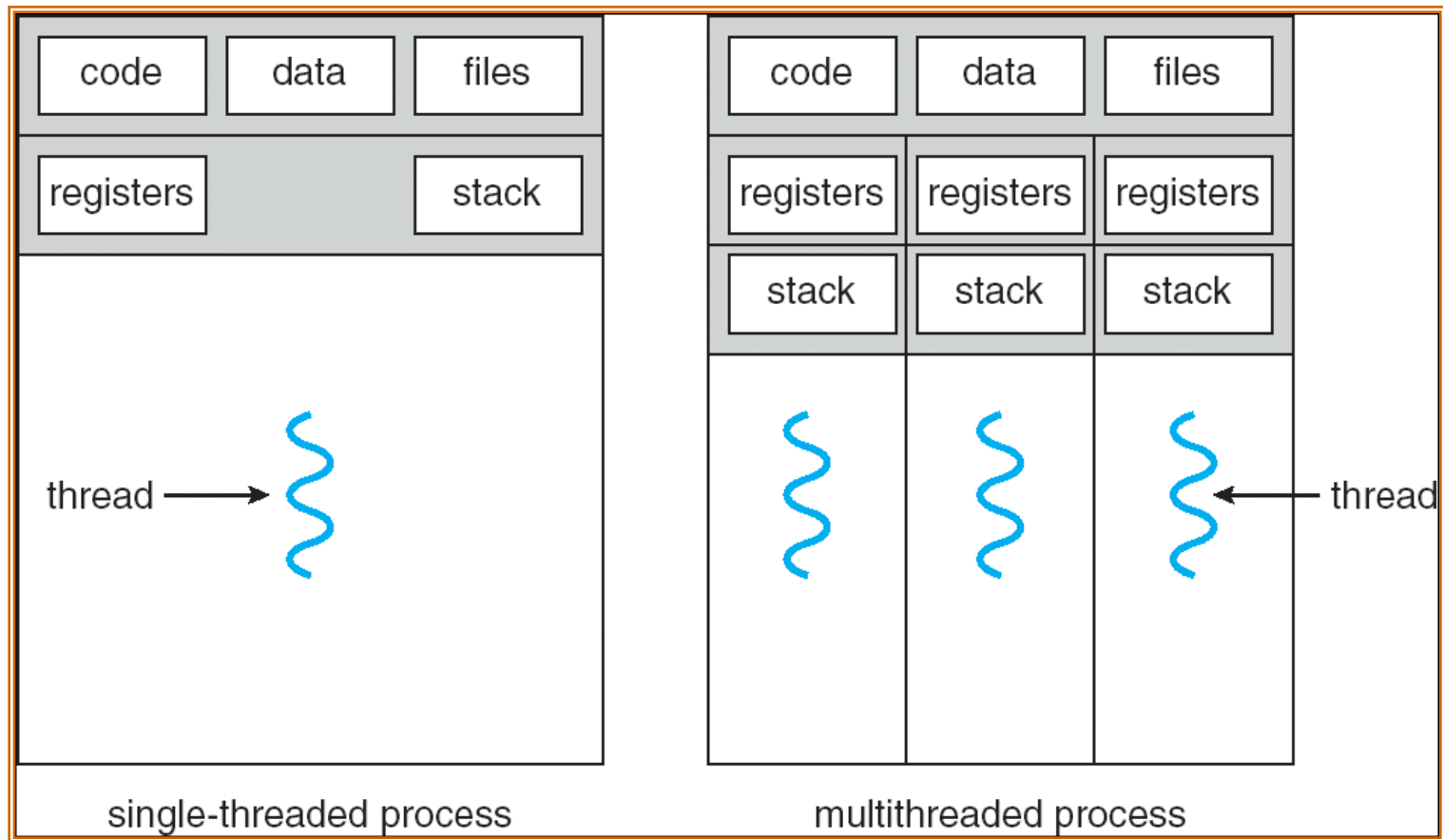
-
- ▶Necesitamos un proceso que realice varias tareas de forma concurrente.
 - ▶ Crear proceso principal y múltiples forks().
 - ▶ Inicialmente, asignar recursos para nuevo proceso
 - Proceso hijo puede durar poco.
 - ▶ Context switch
 - Almacenar estado de proceso
 - Almacenar datos del proceso
 - ▶ Comunicación entre procesos
 - IPC
 - Mensajes
 - Memoria compartida
 - Requieren varias system calls.



THREADS

- ▶ Unidad básica de utilización de CPU.
- ▶ Múltiples threads por proceso
- ▶ Código
 - ▶ Compartido
- ▶ Datos
 - ▶ Compartido
- ▶ Archivos
 - ▶ Compartido
- ▶ Registros
 - ▶ Independiente
- ▶ Pila
 - ▶ Independiente





▶ Beneficios

▶ Interactividad.

- ▶ Ejemplo: Browser: recibe peticiones usuario y trae página.

▶ Compartir recursos

- ▶ Código
- ▶ Datos
- ▶ Archivos

▶ Economía

- ▶ Ahorrar asignar recursos
- ▶ Minimizar tiempo context – switch
- ▶ Tiempo para crear procesos en Solaris
 - proceso - 30x;
 - thread - x

▶ Utilización de arquitecturas multiprocesador

- ▶ Concurrencia



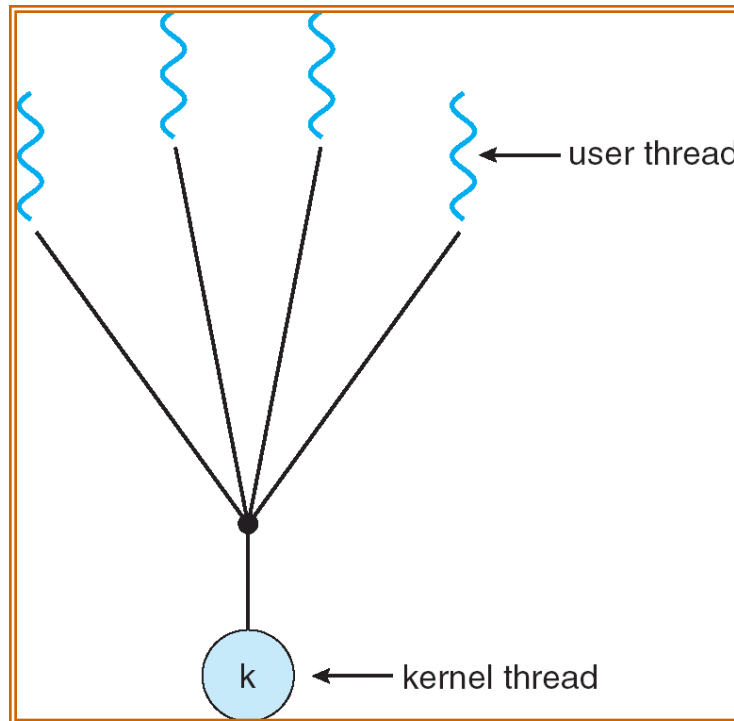
MODELOS

- ▶ User threads
 - ▶ Threads de programas
- ▶ Kernel threads
 - ▶ Ejecución de kernel



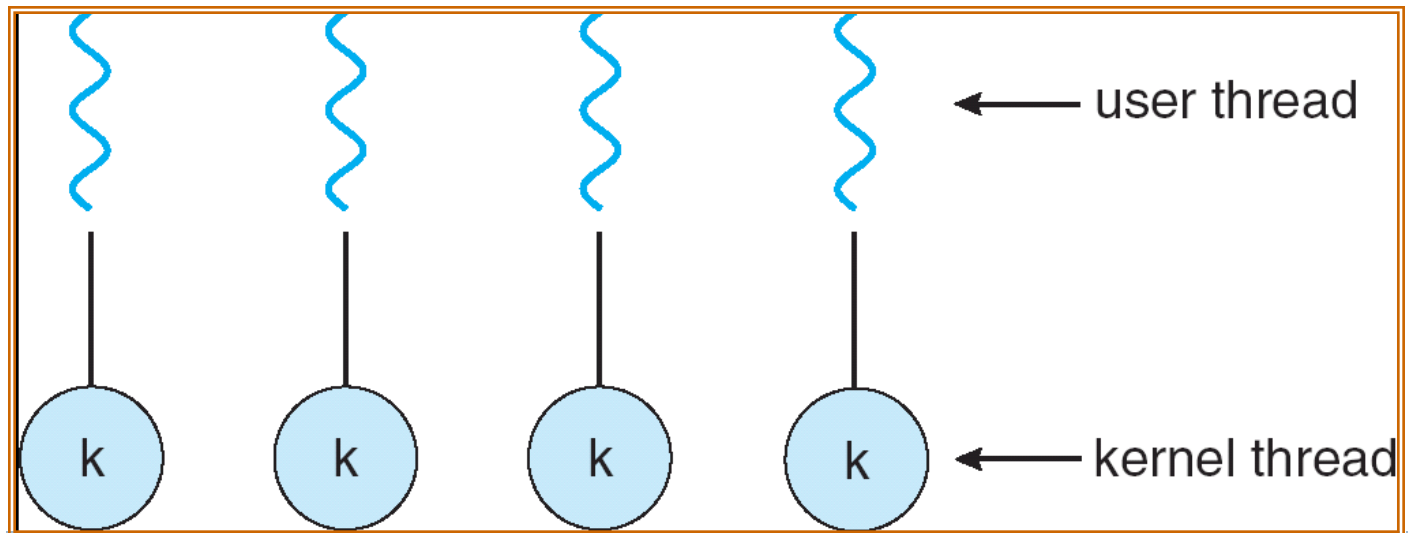
► Muchos a uno

- Muchos user thread por un kernel thread
 - Un user thread puede acceder a kernel thread a la vez
 - Blocking system call: bloquea ejecución de otras user threads

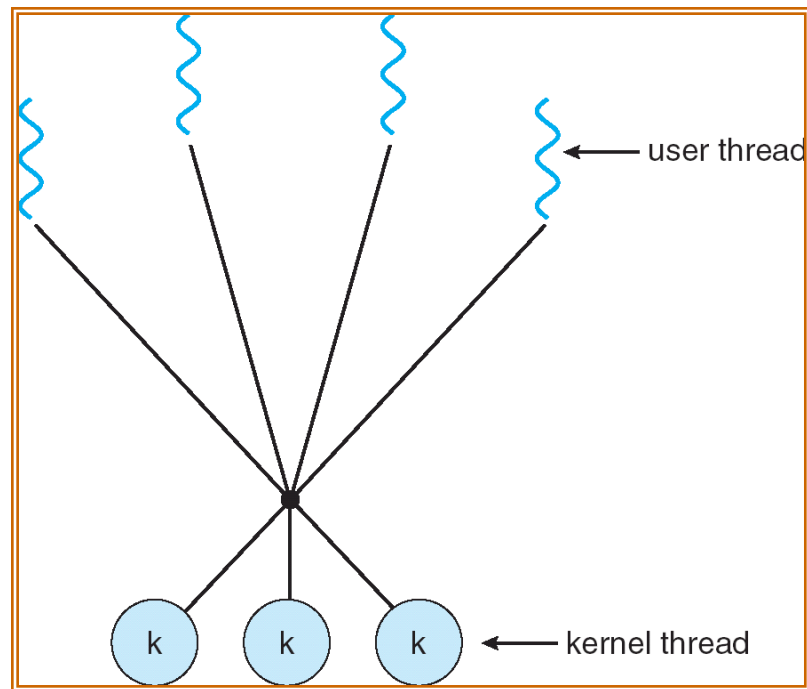


► Uno a uno

- Permite concurrencia real
- Overhead al crear kernel threads
- Limitar número de kernel threads
- Ejemplo: linux, windows Solaris



- ▶ Muchos a muchos
 - ▶ Menor número de kernel threads
 - ▶ Difícil implementación



THREAD LIBRARY

- ▶ API para crear y manejar threads
- ▶ Implementación
 - ▶ Modo usuario
 - ▶ Modo kernel
- ▶ Pthreads
 - ▶ Estándar POSIX
 - ▶ Especificación de comportamiento de threads
 - ▶ No especifica la implementación
 - ▶ Utilizado por Solaris, Linux, Mac OS X
 - ▶ Implementado en modo usuario o kernel
 - ▶ Ver 4thrd-posix.c



▶ Win32 threads

- ▶ Implementado en modo kernel
- ▶ Ver 4thrd-win32.c

▶ Java threads

- ▶ Heredar clase Thread
- ▶ Implementar interfaz Runnable
- ▶ JVM corre sobre S.O.
 - ▶ Win32 en Windows
 - ▶ Pthreads en UNIX y Linux
- ▶ Ver 4Driver.java



ISSUES

- ▶ **Fork():** Crear proceso hijo (duplicado)
 - ▶ ¿Duplicar todas las threads de un proceso?
 - ▶ Existen las dos versiones
- ▶ **Exec():** Reemplazar código de un proceso
 - ▶ Reemplazar todas las threads
- ▶ **Datos específicos por thread**
 - ▶ No es un problema en Java si se hereda la clase Thread



▶ Cancelación

▶ Ejemplo:

- ▶ Botón stop en browser

▶ Cancelación asincrónica

- ▶ Una thread termina a otra de forma inmediata
- ▶ Problema con recursos asignados, como archivos

▶ Cancelación diferida

- ▶ Thread revisa periódicamente si debe terminar.
- ▶ Terminarse de forma ordenada



▶ Señales

▶ Ejemplo:

- ▶ Cambiar de tamaño una ventana.
- ▶ Ctrl + Alt + Delete

▶ Proceso:

- ▶ Evento genera señal
- ▶ La señal es entregada a un proceso
- ▶ El proceso maneja la señal (handle)
 - Default signal handler
 - Sobreescribir signal handler



-
- ▶ ¿Si existen varias threads en un proceso, quién recibe la señal?
 - ▶ Entregar a thread que aplique
 - ▶ Entregar a todas las threads
 - ▶ Entregar a algunas threads
 - Threads especifican qué señales desea recibir
 - ▶ Asignar un thread que recibe señales
 - ▶ Señales deben ser manejadas una vez



▶ Thread pools

- ▶ Ejemplo: Web server.

- ▶ Problemas

- ▶ Varias threads. Overhead para crear. Duración corta.
 - ▶ Muchas threads pueden sobrecargar el sistema.

- ▶ Solución

- ▶ Crear varias threads al inicio de proceso
 - ▶ Threads esperan a que se les asigne trabajo

- ▶ Beneficios

- ▶ Atender una solicitud con thread existente es más rápido que crear un thread nuevo
 - ▶ Limita el número de threads. Garantiza no sobrecargar el sistema

- ▶ Implementado por varios thread APIs



LINUX

► Task

- Flujo de control de un programa
- Procesos y threads
- Clone()
 - Compartir programa

flag	descripción
CLONE_FS	Información de File system es compartida
CLONE_VM	Memoria compartida
CLONE_SIGHAND	Compartir signals
CLONE_FILES	Compartir archivos abiertos

