



IC-6600
Sistemas Operativos

Profesor:
Ing. Cristian Campos Agüero

Doc. Original: Ing. Allan Rodríguez Dávila, MGP

Comunicación entre procesos

Procesos

- Independientes
 - No afectan o no pueden afectarse por otros procesos
- Cooperativo
 - Pueden afectar o verse afectados por otros procesos que se ejecutan en el sistema

Cooperación entre procesos

- Compartir información
 - Acceso concurrente a información
- Acelerar cálculos
 - Dividir tareas en subtareas de ejecución paralela
- Modularidad
 - Dividir funciones en hilos
- Conveniencia
 - Utilizar los recursos al máximo

Comunicación entre procesos

- IPC
 - Interprocess communication
- Memoria compartida
 - Región de memoria cooperativa
- Paso de mensajes
 - Procesos cooperativos

Comunicación entre procesos

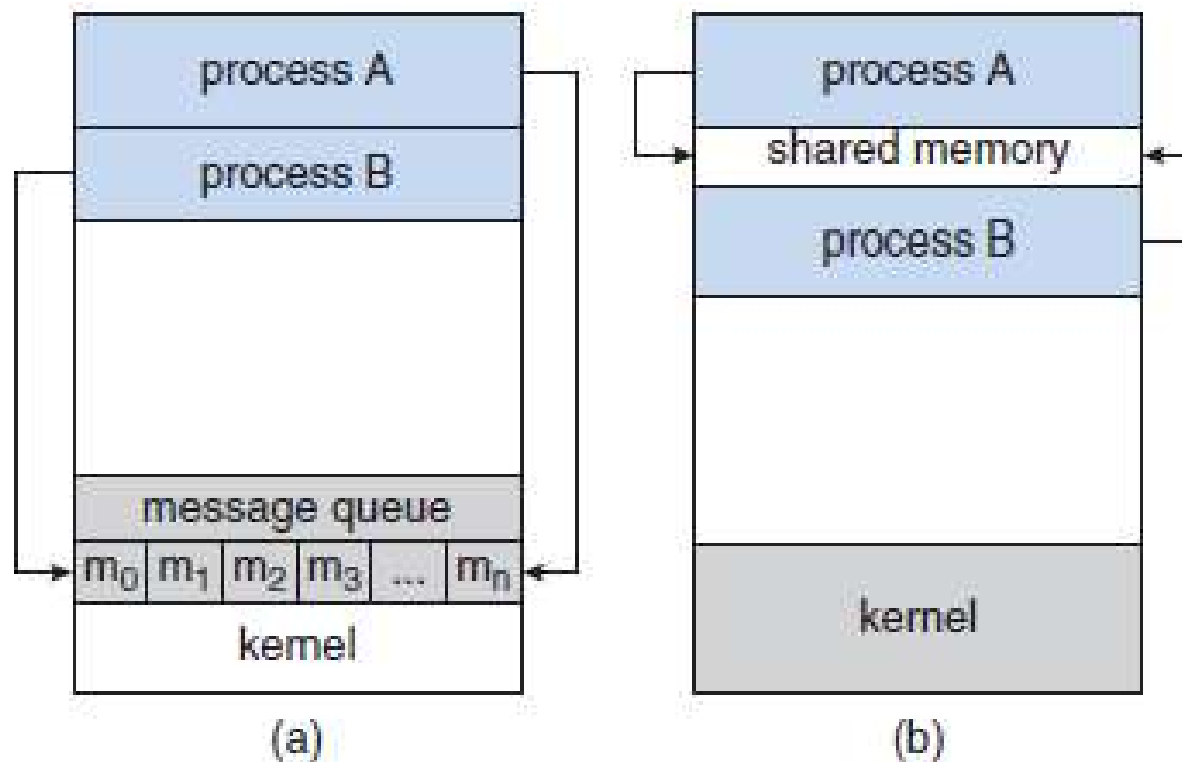


Figure 3.12 Communications models. (a) Message passing. (b) Shared memory.

Memoria compartida

- Región compartida
 - Espacio de direcciones de un proceso
 - Permiso de sistema operativo
- Productor-Consumidor
 - Cliente-Servidor
 - Utilizan un buffer
- Buffer no limitado vs Buffer ilimitado

Paso de mensajes

- Útil en entornos distribuidos
 - Chat
- Send y receive
- Enlace de comunicación
 - Directa o indirecta
 - Síncrona o asíncrona
 - Buffer explícito o automático

Comunicación directa

- **Nombrado:**
 - Método cómo se referencian los procesos
 - `send(P, mensaje) & receive(Q, mensaje)`
- **Automático** entre cada **dos procesos**
 - Exactamente dos procesos
 - Entre cada par existe un único enlace
- Se da **simetría en direccionamiento**
 - Se nombran transmisor y receptor

Comunicación directa

- `send(P, message)` —Send a message to process P.
 - `receive(Q, message)` —Receive a message from process Q.
-
- `send(P, message)` —Send a message to process P.
 - `receive(id, message)` —Receive a message from any process. The variable `id` is set to the name of the process with which communication has taken place.

Comunicación Indirecta

- Los mensajes se envían y reciben en **buzones de correo o puertos**
 - Colocan y eliminan mensajes
- Puede establecer entre dos procesos si tienen **buzón compartido**
- **Enlace puede asociarse con 2 o más**
 - Dos procesos pueden tener varios enlaces

Comunicación Indirecta

- `send(A, mensaje)`—Envía un mensaje al buzón de correo A.
- `receive(A, mensaje)`—Recibe un mensaje del buzón de correo A.

Comunicación con sincronización

- Envío con bloqueo
 - Transmisor bloquea hasta que se reciba
- Envío sin bloqueo
 - Envío sin esperar respuesta
- Recepción con bloqueo
 - Receptor bloquea hasta recibir mensaje
- Recepción sin bloqueo
 - Receptor extrae mensaje válido o nulo

Comunicación según buffer

- Capacidad cero
 - No existen mensajes en espera
- Capacidad limitada
 - Cola longitud n , si se llega al límite se bloquea el transmisor
- Capacidad ilimitada
 - Cola infinita, el transmisor no se bloquea

Comunicación Cliente-Servidor

Trabajo en clases (Sección 3.6)

- Sockets
- Llamadas a procedimientos remotos (RPC)
- Invocación a métodos remotos (RMI)

Hilos

Hilos

- **Unidad básica** de utilización del CPU
 - Propietario de Id, contador, registros y pila
 - Comparte código, datos, archivos, etc.
- **Procesos mono-hilos o multi-hilos**
 - Proceso pesado un solo hilo
- **Kernels multi-hilos**
 - Hilos para tareas específicas: gestionar dispositivos, interrupciones, memoria



Ventajas

- Capacidad de respuesta
 - Programa continúa con una parte bloqueada o realizando operación larga
 - Incrementa respuesta al usuario
- Recursos compartidos
 - Comparten memoria y recursos del proceso padre y código

Ventajas

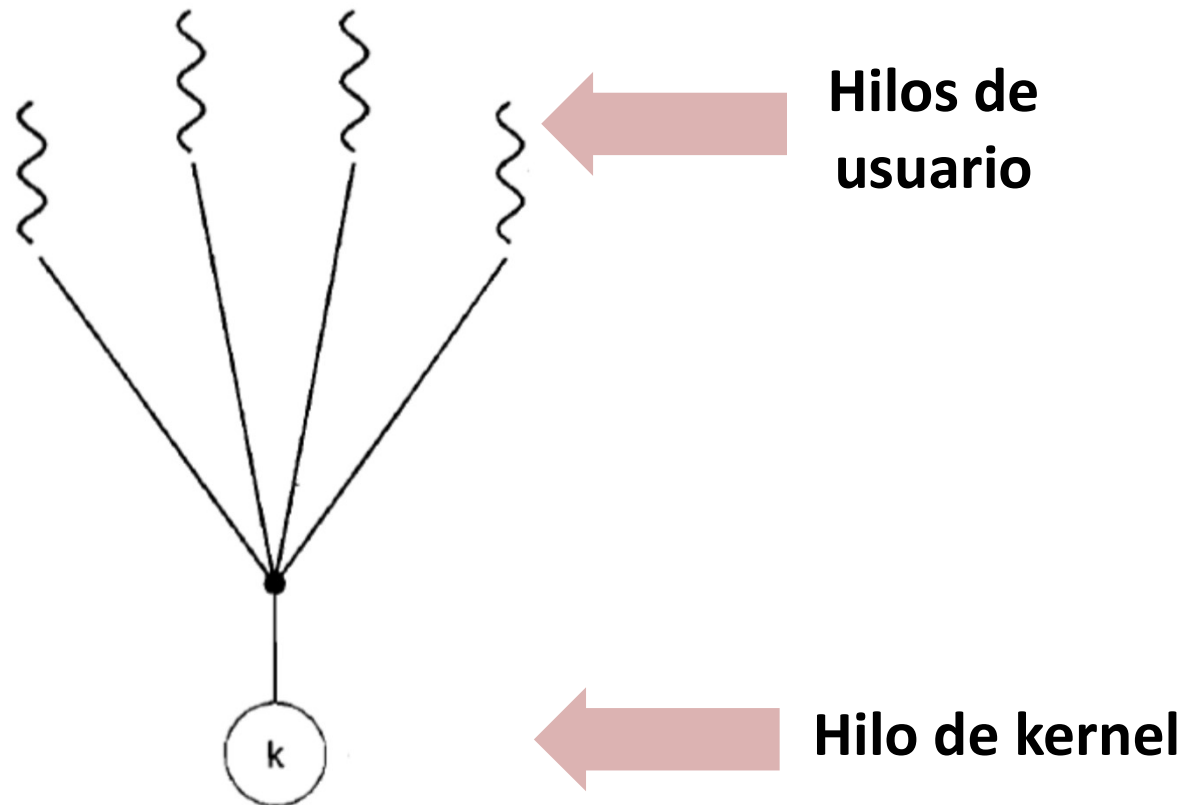
- Economía
 - Comparten recursos, facilidad de creación
 - Crear proceso 30 veces más lento que hilo
- Utilización arquitectura multiprocesador
 - Hilo x cada procesador
 - Ejecución paralela
 - Aumenta la concurrencia

Modelo multi-hilos

- Hilos de usuario
 - Se gestionan por encima del kernel
 - Sin soporte de kernel
- Hilos de kernel
 - Soportadas por el sistema operativo
- Modelos:
 - Muchos a uno
 - Uno a uno
 - Muchos a muchos

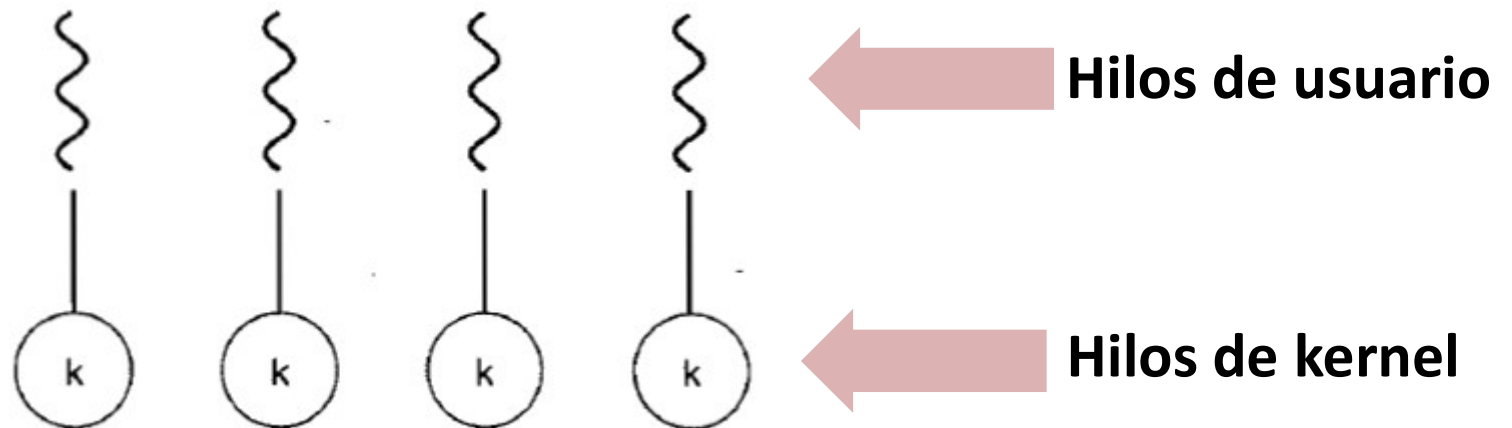
Modelo Muchos a uno

- Múltiples hilos a nivel de usuario a un hilo de kernel



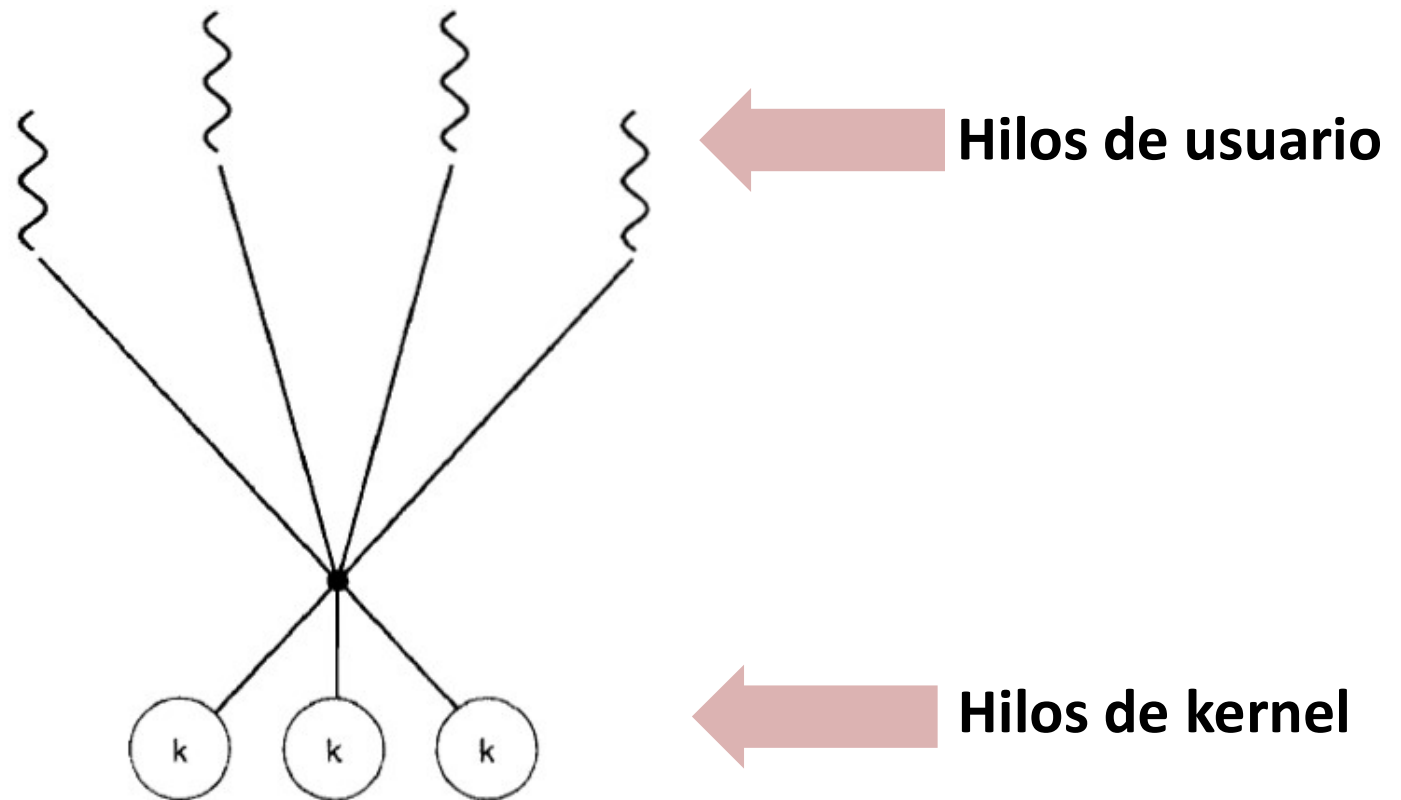
Modelo uno a uno

- Cada hilo de usuario se asigna a un hilo de kernel



Modelo muchos a muchos

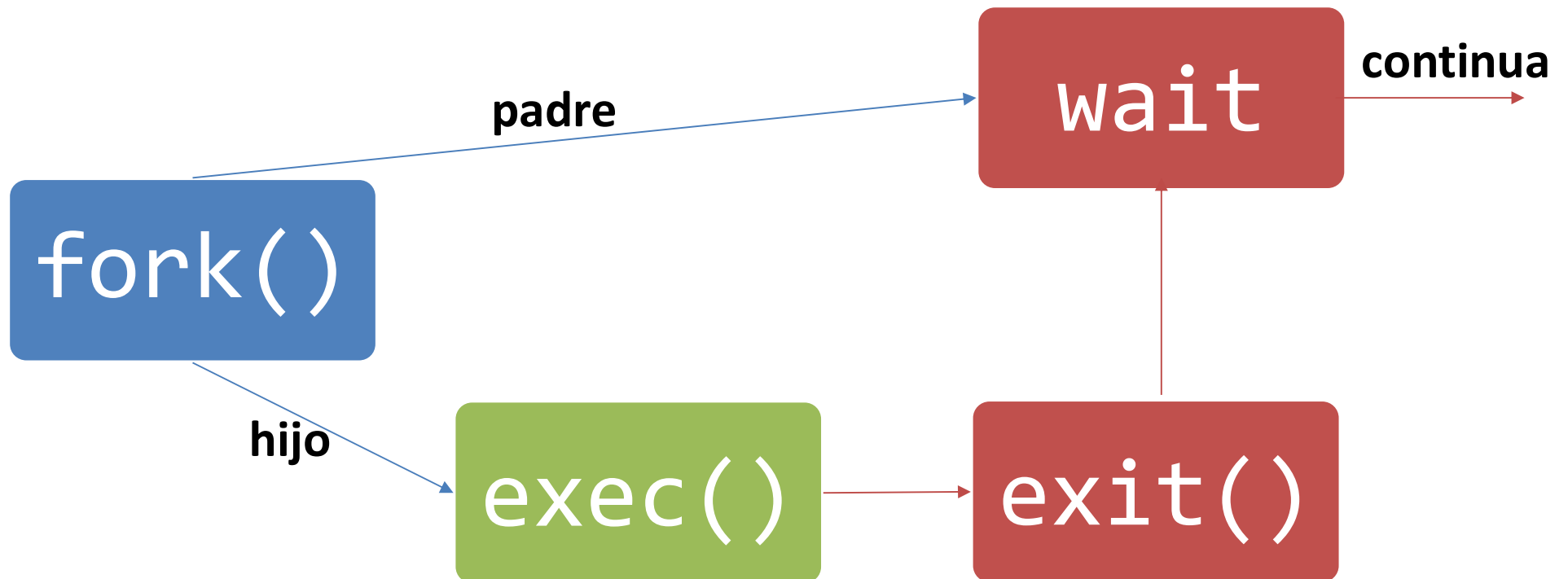
- Multiplexa muchos hilos de usuario sobre hilos de kernel



Fork & Exec

- Cuando un proceso crea otro existen dos posibilidades:
 - El padre continúa ejecutándose concurrentemente con el hijo
 - El padre espera hasta que alguno o todos terminen
- En cuánto al uso de la memoria:
 - El proceso hijo es un duplicado del padre
 - El proceso hijo carga un nuevo programa

Fork & Exec



fork() & exec()

- En un **hilo** al llamar **fork** hay dos opciones:
 - Duplica todos los hilos
 - Duplica el hilo que invocó la llamada
- Típicamente al llamar **exec** en un hilo se **reemplazan todos los hilos** con el nuevo programa

Cancelación

- Un hilo puede cancelarse
 - Hilo objetivo
 - Ejemplo: página en proceso de carga
- Cancelación asíncrona
 - Otro hilo envía la cancelación inmediata
- Cancelación diferida
 - El hilo objetivo consulta si debe cancelarse.
Puntos de cancelación

Tratamiento de señales

- Asíncrona o síncrona
- Características:
 - Se genera debido a un determinado suceso
 - Se suministra a un proceso
 - Una vez suministrada, debe ser tratada
- Ejemplo:
 - `pthread_kill (pthread_t tid, int signal)`

Tratamiento de señales

- Tratamiento de señal en hilos:
 - Suministrar la señal al hilo que sea aplicable
 - Suministrar la señal a todos los hilos del proceso
 - Suministrar la señal a ciertos hilos del proceso
 - Asignar un hilo para recibir todas la señales

Conjuntos compartidos de hilos

- Cola de hilos
 - Se creen hilos a la espera de “trabajos”
 - Al terminar un “trabajo” vuelve a la cola
 - Más eficiente que crear un hilo
 - Se limita el máximo de hilos
 - Ajuste fijo según el número de procesadores
 - Ajuste dinámico según la carga (patrones de uso)

- Silberschatz & Galvin. Sistemas Operativos . 6ta Edición. Limusa Wiley.
- Stallings. Sistemas Operativos. 4ta Edición. Prentice Hall.
- Tanenbaum. Distributed Operating Systems, Prentice Hall.

TEC | Tecnológico
de Costa Rica