



75-08 Sistemas Operativos
Lic. Ing. Osvaldo Clúa
2009

Facultad de Ingeniería
Universidad de Buenos Aires

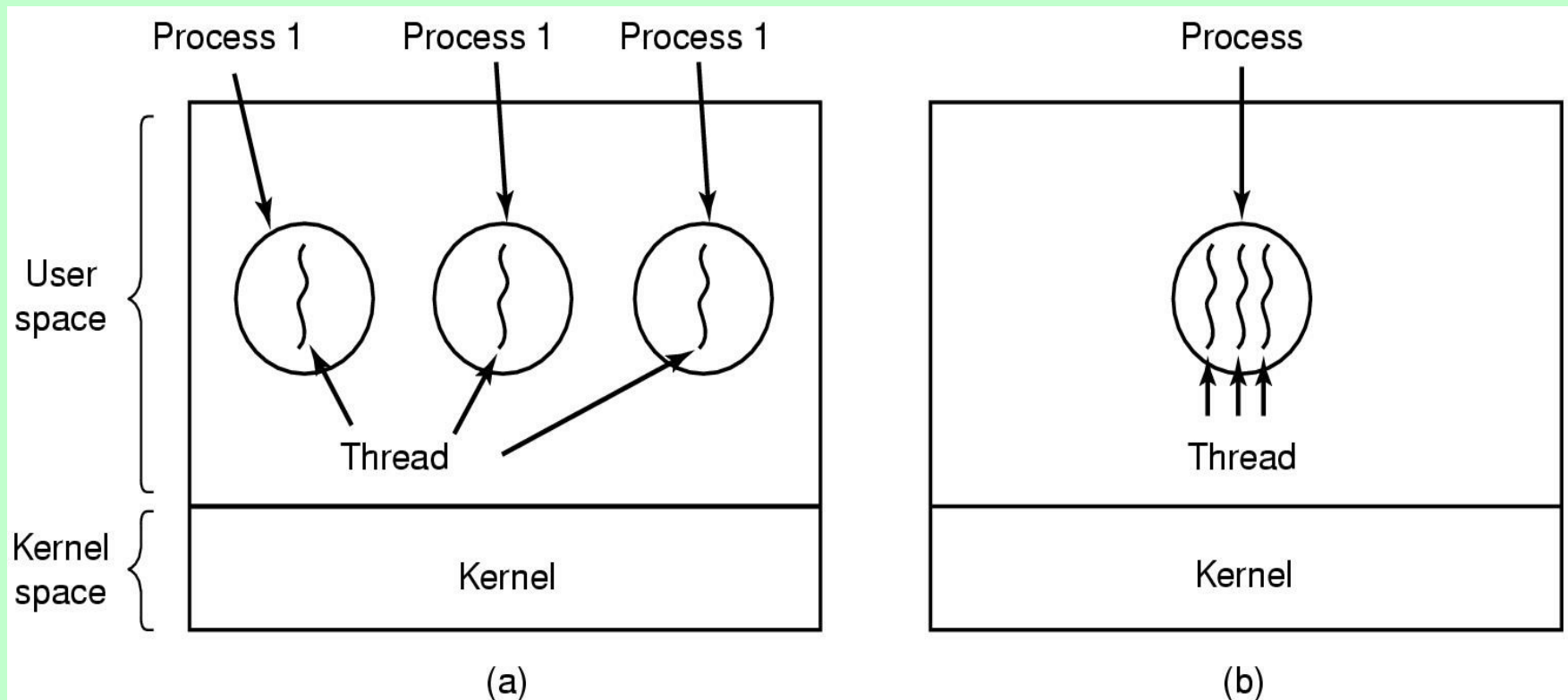
Threads

Hilos de ejecución (Execution Threads)

- Se separa la ejecución del agrupamiento de recursos.
- Los Procesos agrupan los recursos usados.
- Los Threads son hilos de ejecución que comparten el agrupamiento de recursos.
 - Pueden implementarse por multiprogramación o por multiprocesamiento.

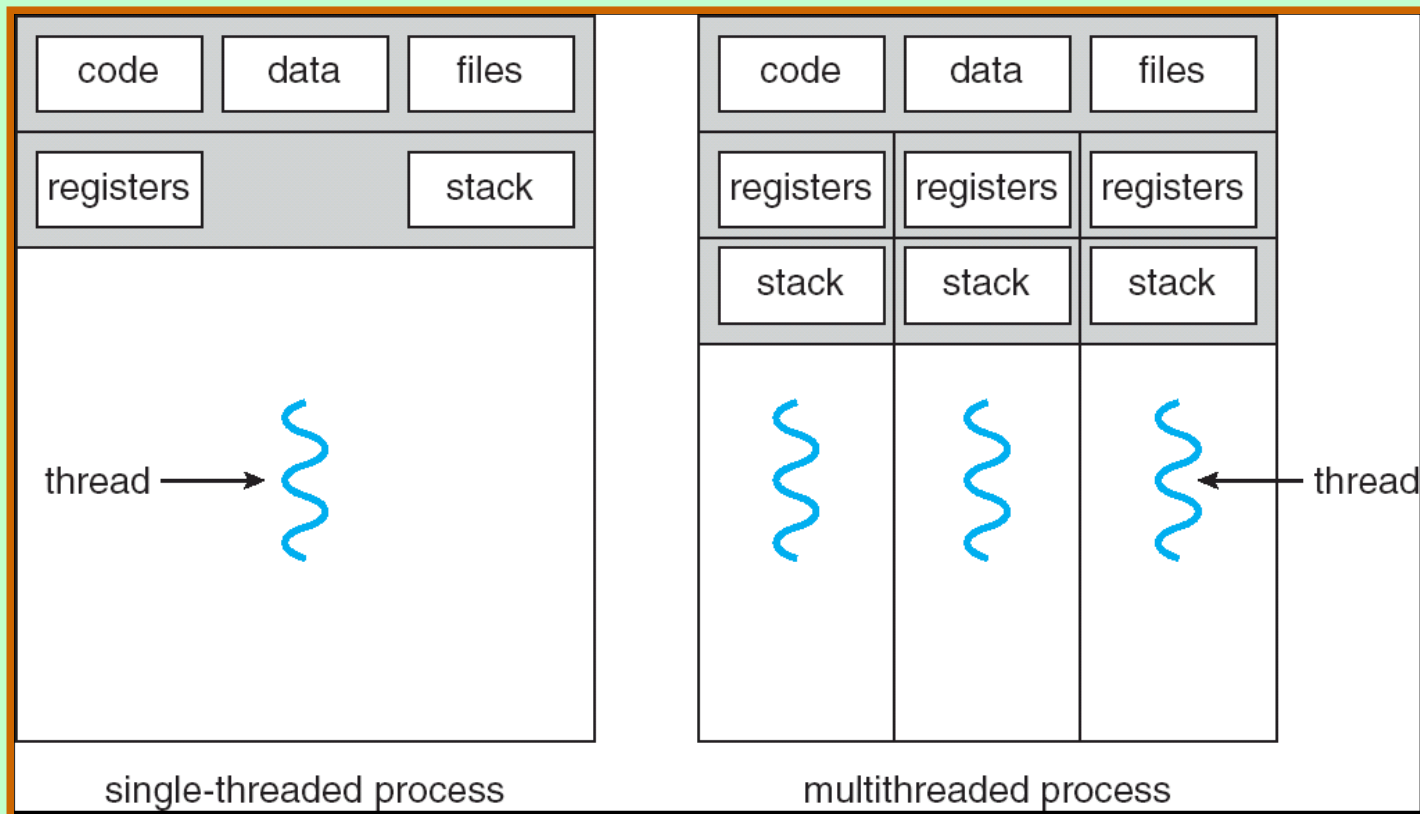
Threads vs Procesos

- Los Threads comparten los recursos del proceso, entre ellos, el espacio de memoria.



Threads vs Procesos (2)

- Cada thread mantiene su propia información de estado.



(Thread Control Block) TCB

- Hay información que pasa del PCB al (o los) TCB.

Per process items

Address space
Global variables
Open files
Child processes
Pending alarms
Signals and signal handlers
Accounting information

Per thread items

Program counter
Registers
Stack
State

Aplicaciones de Multithreading

- Varias aplicaciones que concurren sobre los mismo datos como:
 - Un server que lanza un thread por cada pedido.
 - Un procesador de texto concurrente con su corrector y su armador de pagina.
 - El manejo de Interfaces Gráficas.
- Sin embargo

Threads vs. eventos

- Algunos de los sistemas hechos con threads podrían hacerse con eventos.
- Hay una discusión con campos:
 - A favor de los threads.
 - A favor de los eventos.
- Lo que sí es malo es mezclar ambas prácticas (ej: Java threads sobre Windows en GUI).

Formas de construcción de un server

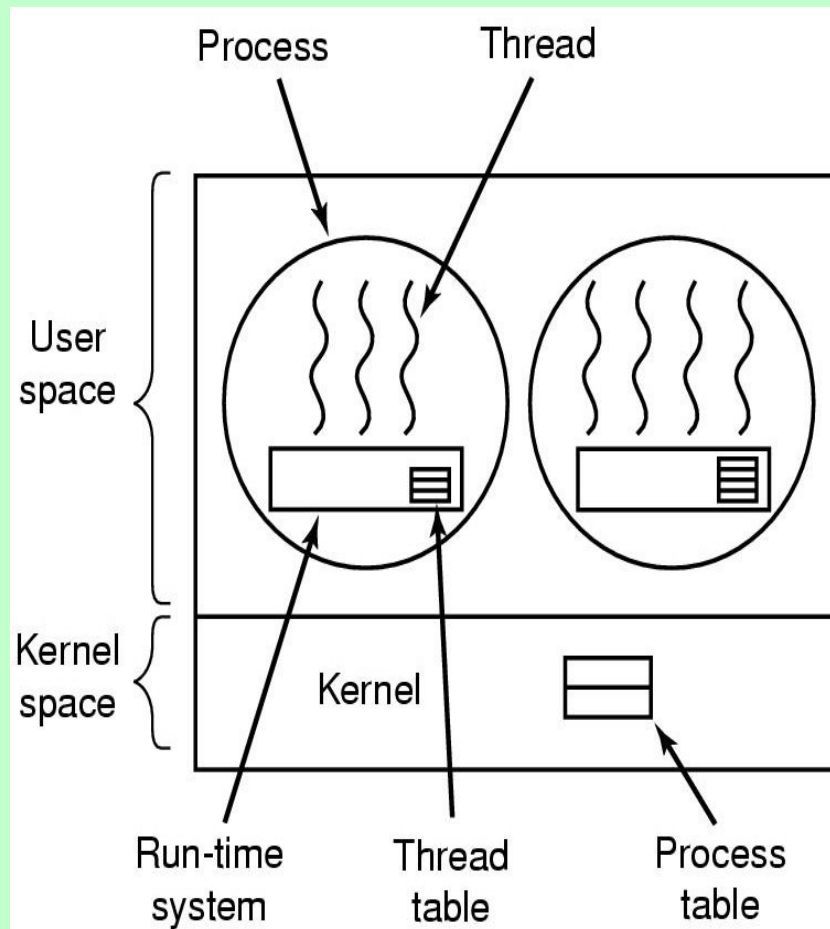
Model	Characteristics
Threads	Parallelism, blocking system calls
Single-threaded process	No parallelism, blocking system calls
Finite-state machine	Parallelism, nonblocking system calls, interrupts

- Tres modelos de arquitecturas de un server.
- Hay que ser muy cuidadoso con la elección.
 - Algo mas sobre **Finite State Machines**

Implementación de Threading

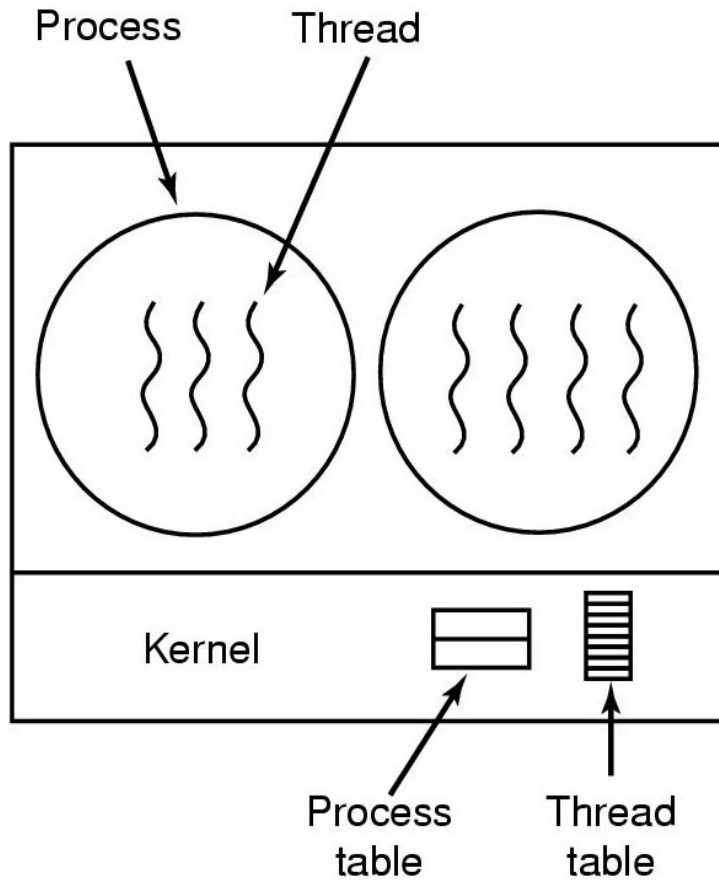
- A nivel del kernel
 - El scheduler planifica threads.
 - Si el próximo está en otro proceso, hay un *process context switch*; si está en el mismo, no lo hay.
- A nivel de usuario
 - La biblioteca de threads permite al usuario multiplexar su time slice.
- En un nivel intermedio.

Threads en Espacio de Usuario



- Cuando un thread se bloquea, puede continuar otro.
- El time slice impide que el usuario bloquee al sistema.
- **LinuxThreads**
 - **Pthreads** es una API.
- Perl threads

Kernel Threads

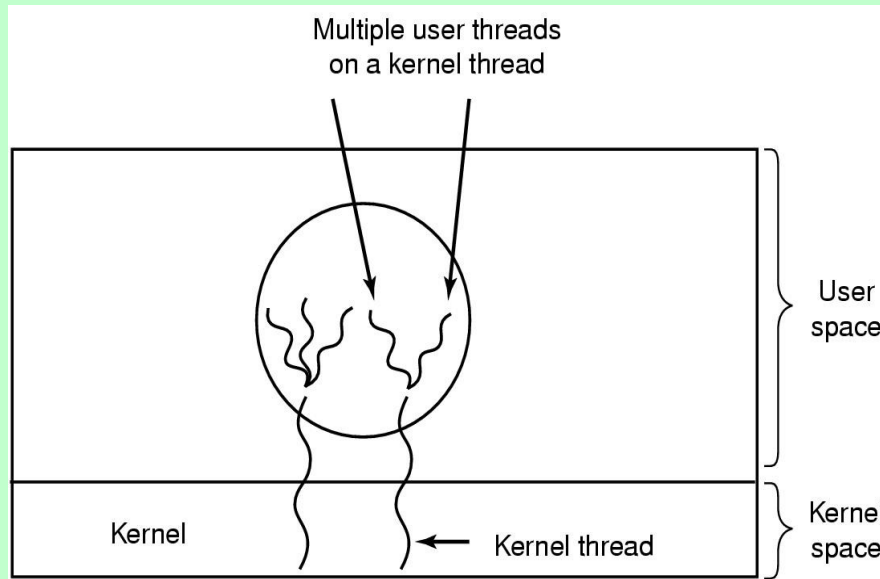


- El Kernel planifica directamente threads.
- Si un thread se bloquea, se planifica otro.
- Native **Posix Threads**
 - Otra implementación de Pthreads.

¿Qué versión de Pthreads hay en mi linux?

```
$ getconf GNU_LIBPTHREAD_VERSION
```

Implementaciones Híbridas



Solaris lwp (LightWeight Process)

- El Kernel vé algunos Threads (o algunos grupos de threads).
- Permite decidir cuales planifica el Kernel y cuales la biblioteca.

Otros sistemas

- Anderson propuso **Scheduler Activations**
 - Ante un bloqueo de thread, el scheduler activa a la biblioteca de usuario.
- Para la implementación de **corutinas** se han propuestos mecanismos conocidos como **fibras**.
 - Antes se hacían con `longjmp (3)` y `setjmp (3)`.
 - Ahora se usan las rutinas de `makecontext (3)`.

Llegan los Objetos

- ¿Que pasa entonces con?
 - Procesos - Unidad de encapsulamiento.
 - Threads - Unidad de Ejecución
 - Un thread “pasa” de un objeto a otro.
 - Un thread por objeto (Cliente servidor).
 - Mas de un thread en un mismo objeto.
- para entender mejor esto debemos entender la Link-Edición.