

Sistemas Operativos

Dr. Wenceslao Palma M. <wenceslao.palma@ucv.cl>

En términos simples un thread corresponde a un flujo de control.

En un programa generalmente existe un thread dentro del cual existe un único punto de ejecución. En este caso no necesitamos algún soporte especial para programar un único thread.

Múltiples threads dentro de un programa significa que en cualquier instante existen múltiples puntos de ejecución, cada uno representado por un thread. Esto se conoce como multithread.

MS-DOS soporta un único proceso y un único thread. Algunas versiones de UNIX soportan múltiples procesos pero sólo un thread por proceso.

Linux, Windows, Solaris y OS/2 soportan múltiples procesos y múltiples threads.

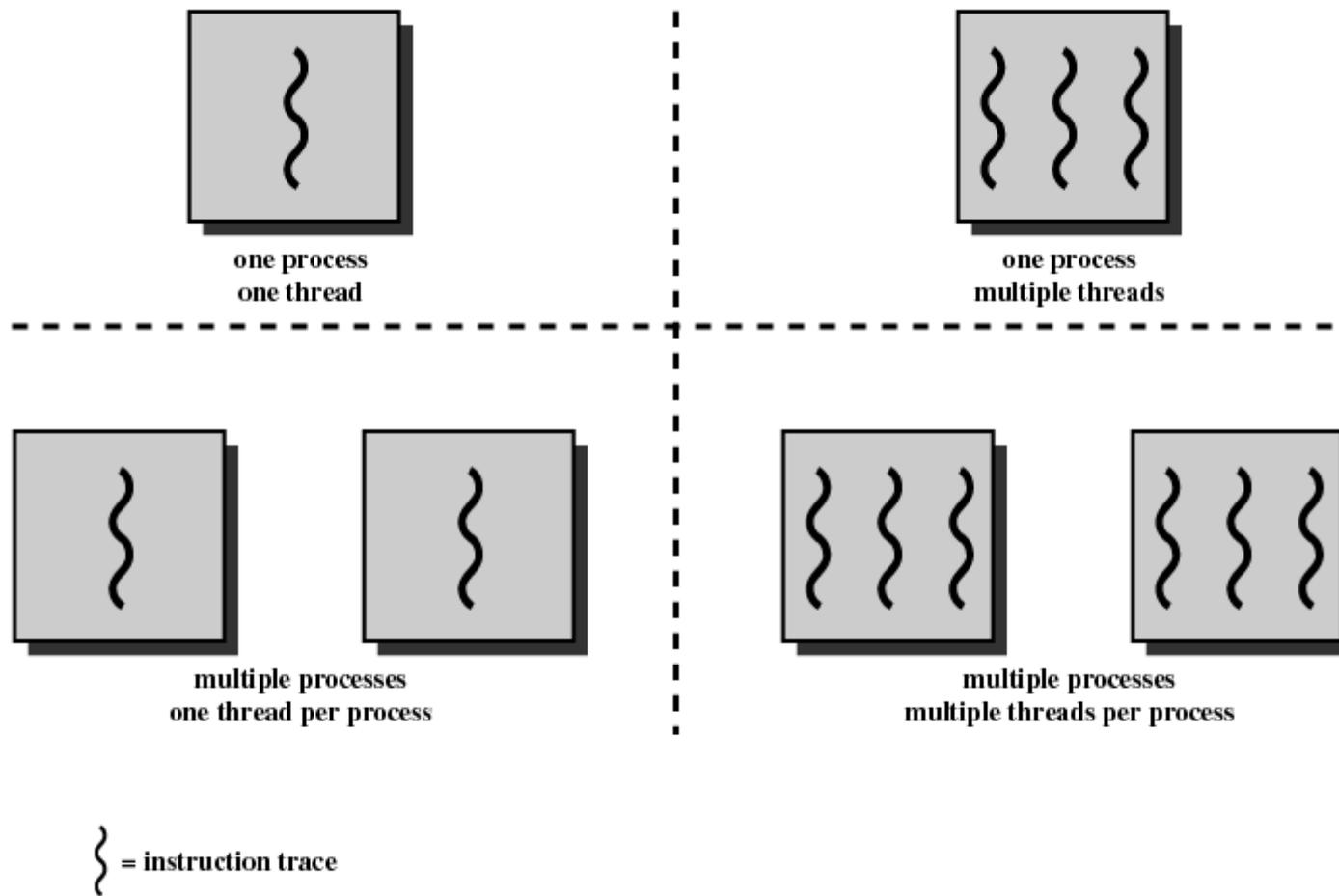


Figure 4.1 Threads and Processes [ANDE97]

En un entorno multithread, un proceso tiene asociado:

- un espacio de direcciones virtuales que contiene la imagen del proceso.
- acceso protegido al procesador, otros procesos, archivos y recursos de E/S.

Un proceso puede contener uno o más threads, cada uno de los cuales contiene:

- estado del thread.
- contexto del procesador.
- pila de ejecución.
- almacenamiento para variables locales.
- acceso a la memoria y recursos del proceso al cual pertenece.

Ventajas de la utilización de threads:

- es mucho más rápido crear/terminar un thread que un nuevo proceso.
- es más rápido cambiar entre dos threads que entre dos procesos.
- la comunicación entre procesos es más eficiente.

Ejemplos.

- Trabajo interactivo y en segundo plano.
- Procesamiento asíncrono. Un thread grabando el buffer de un procesador de texto.
- Aceleración de la ejecución. Lectura+cálculo+escritura.

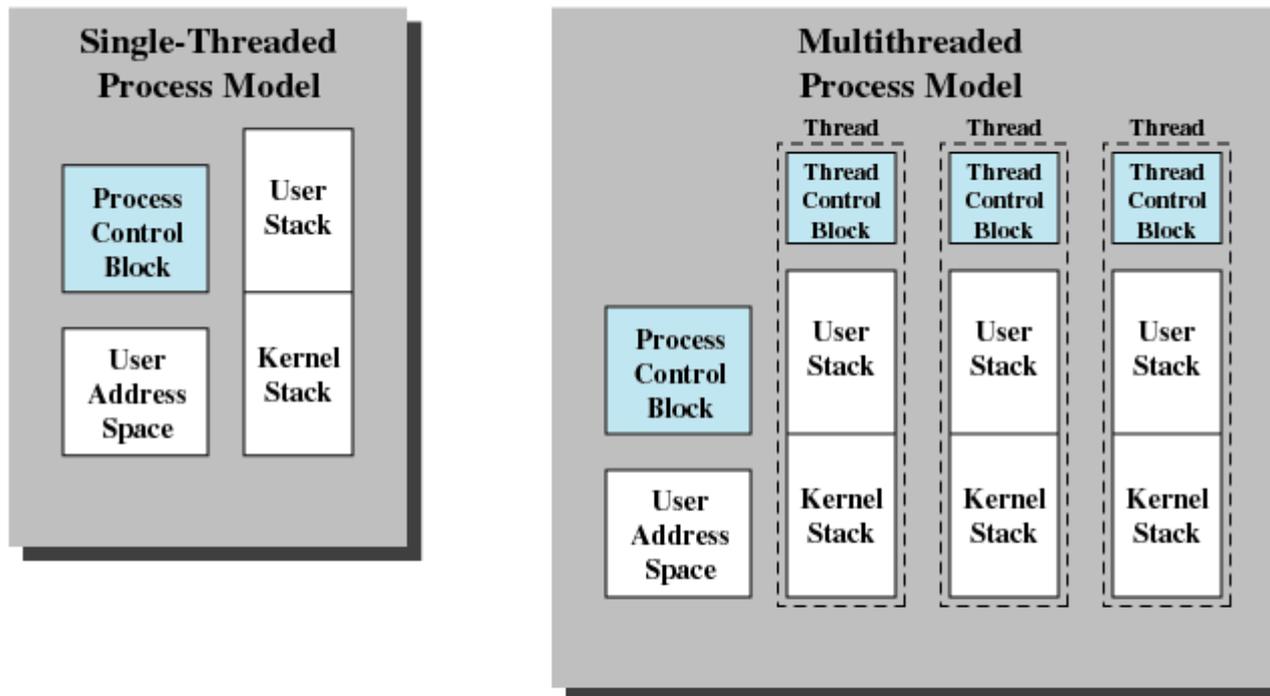
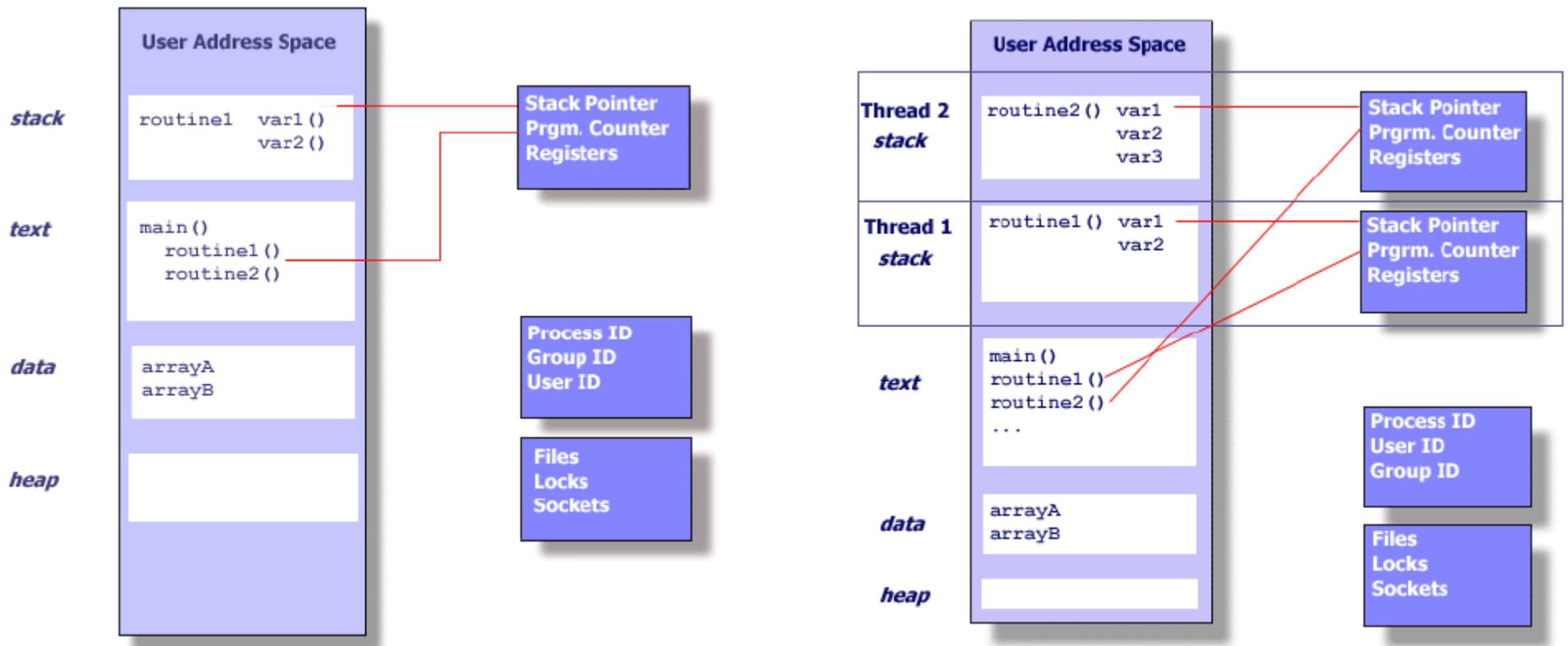


Figure 4.2 Single Threaded and Multithreaded Process Models



Estados de un hilo. running, ready y blocked.

Si un proceso es expulsado de la memoria principal, todos los hilos lo estarán ya que todos comparten el espacio de direcciones del proceso.

Existen cuatro operaciones asociadas con el cambio de estado de un hilo:

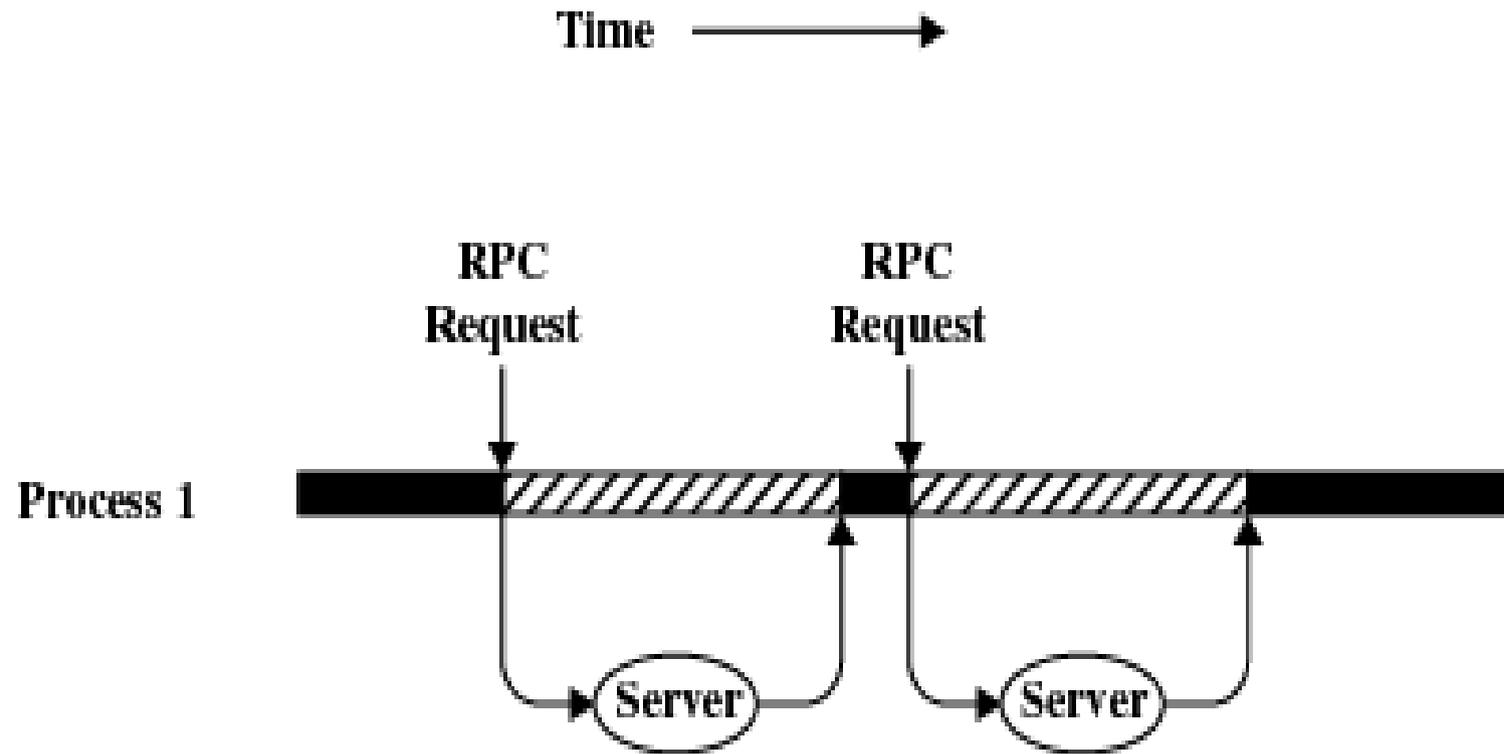
Creación. Una vez creado un thread pasa al estado ready.

Bloqueo. Cuando un thread espera por un suceso, se bloquea (cambio de contexto) y el procesador puede pasar a ejecutar un thread en estado ready.

Desbloqueo. el thread pasa al estado ready.

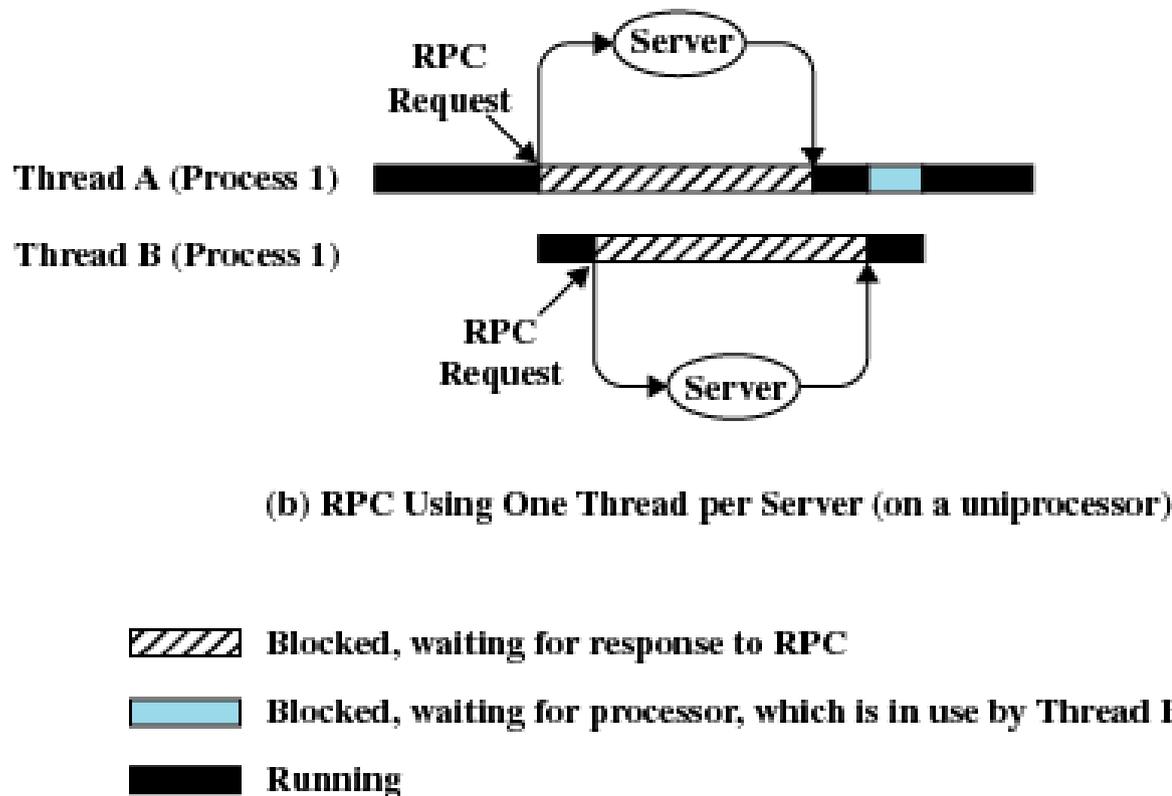
Término. Cuando un thread termina se liberan su contexto y pilas.

Llamada RPC usando un thread



(a) RPC Using Single Thread

Llamada RPC usando un múltiples threads



(b) RPC Using One Thread per Server (on a uniprocessor)

Figure 4.3 Remote Procedure Call (RPC) Using Threads