GESTOR DE PROCESOS

Karim Guevara Puente de la Vega

2017

Objetivos

- Conocer el concepto de proceso, en el que se basa todo el funcionamiento de un sistema informático.
- Describir los diversos mecanismos relacionados con los procesos, incluyendo los de planificación, creación, finalización y los de comunicación

Introducción

- Los primeros sistemas informáticos solo permitían que se ejecutara un programa a la vez.
- Este programa tenia el control completo del Sistema.
- Tenia acceso a todos los recursos.
- Por el contrario los actuales permiten que se carguen en memoria múltiples programas y se ejecuten concurrentemente.

Programa y Proceso

- Programa
  - Es estático. No varía nada
  - No tiene estado
  - Variables sin valores
  - Rutinas sin dirección
  - Condiciones sin evaluar
  - **Es el algoritmo a ejecutar**
- Proceso
  - Es dinámico
  - Tiene estado
  - Las variables tienen valores
  - Rutinas están en alguna dirección
  - Condiciones se pueden evaluar
  - **Es la ejecución del algoritmo**

Procesos e Hilos

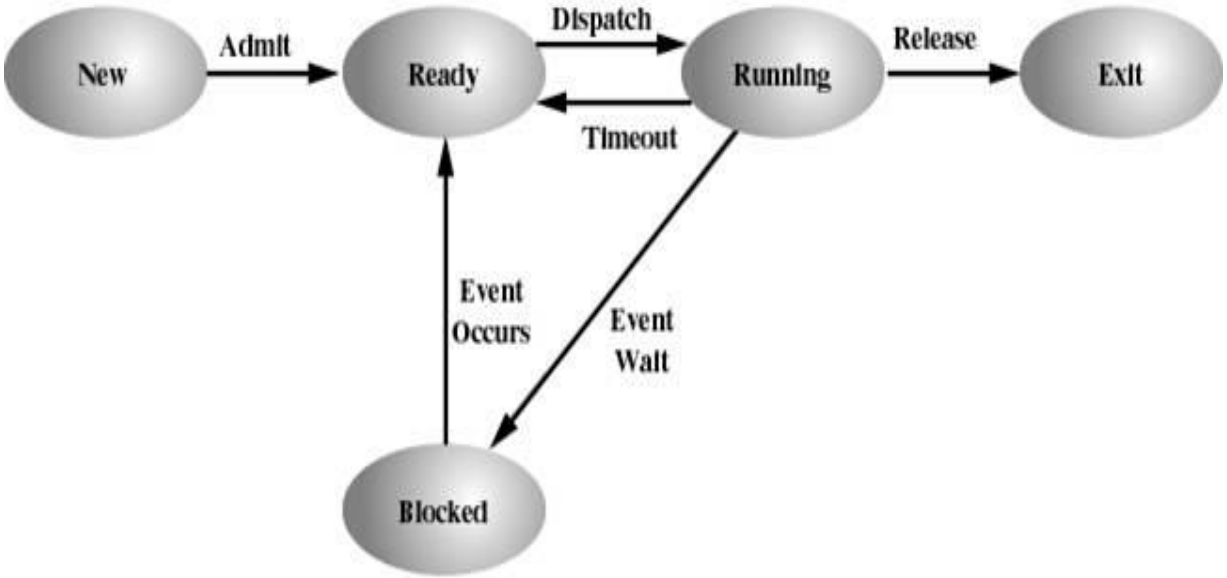
- Un cómputo es una combinación de un proceso (hilo implícito/explícito) y una colección de recursos.
- **Proceso clásico**
  - Un motor de ejecución
- **Proceso moderno**
  - Múltiples motores de ejecución: hilo/proceso ligero
    - Datos privados del hilo: pila
    - Estado del hilo: estructura del SO
    - Contador de programa

Procesos

- Proceso moderno es un entorno en el que se ejecutan los instrucciones basadas en hilos:
  - Espacio de direcciones: 2^{32} (4 Gb)
  - Programa
  - Datos
  - Recursos
  - Identificador de proceso
- Los hilos son los elementos activos de las instrucciones:
  - Ambiente.
  - Datos
  - Identificador de hilo

Estados

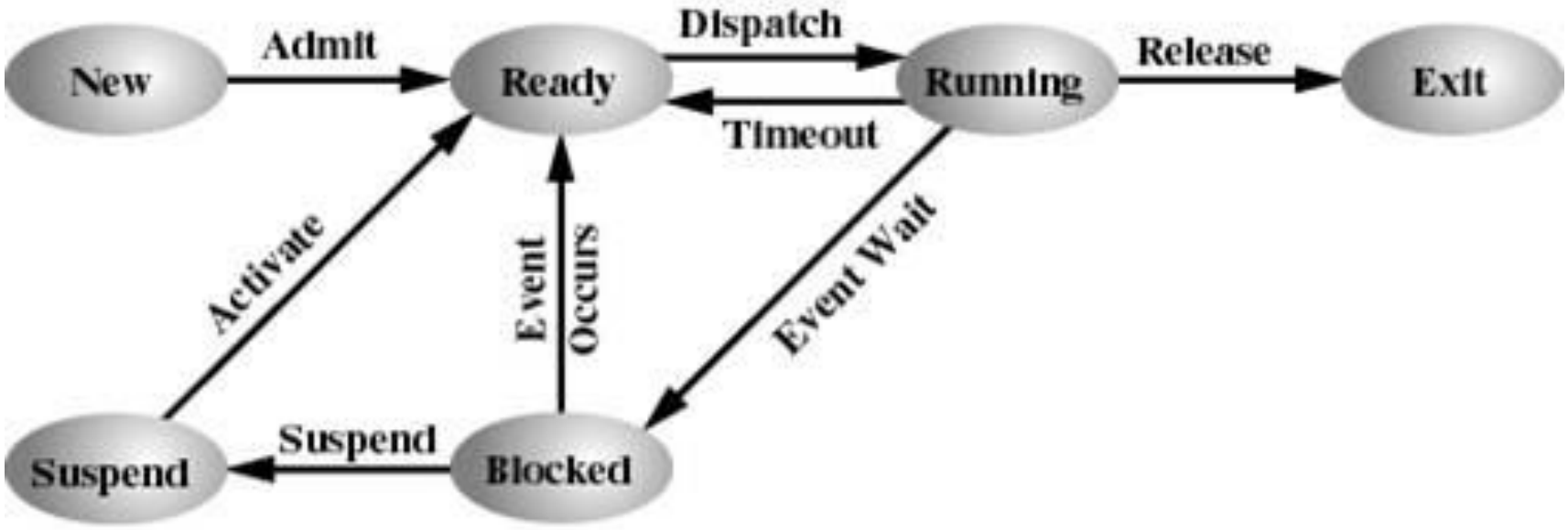
- A medida que se ejecuta va cambiando
- Se define según la actividad actual.
- Diagrama de estados representa los diferentes estados en los que un hilo puede estar en distintos momentos.



Procesos Suspendidos

- El modelo planteado de tres estados (preparado/listo, ejecución y en espera/bloqueado), pero no es suficiente.
- Dado que para que un proceso se ejecute debe ser cargado en memoria completamente.
- Por lo que sería conveniente suspenderlos:
  - El procesador es más rápido que la E/S así que todos los procesos tendrían que esperar la E/S.
  - Intercambiar esos procesos al disco para liberar más memoria.
  - Dos nuevos estados:
    - Bloqueado - suspendido.
    - Listo - suspendido.

Procesos Suspendidos



Razones para suspender procesos

- Intercambio (swapping).
  - El SO necesita liberar suficiente memoria RAM para cargar un nuevo proceso.
- Otra razón del SO.
  - El SO puede suspender un proceso que se sospecha causa un problema.
- Solicitud del usuario.
- Por tiempo.
  - Se ejecuta con cierta frecuencia, entonces mientras no se usa se suspende.
- Solicitud del proceso padre.
  - El padre desea suspenderlo para examinar o modificar el proceso o para coordinar con otros procesos.

Creación de un Proceso

- Asignar un espacio de memoria
- Seleccionar un BCP libre
- Rellenar el BCP con información de identificación
- Cargar en el segmento de texto el código mas las rutinas de sistema, y en el segmento de datos los datos iniciales.
- Crear en el segmento de pila la pila inicial del proceso

Descriptor de procesos – Bloque de Control de Procesos (PCB)

- Estructura de datos, en la que el SO almacena toda la información para gestionar los procesos

Campo	Descripción
Nombre interno del proceso	Un entero o índice de una tabla, usado por el SO
* Estado	Del hilo base
Propietario	Identificación del propietario
Estadísticas de ejecución	Suma de tiempo, tiempo de inicio, etc.
Hilo	Hilos asociados al proceso
Procesos relacionados	Referencia a procesos padre/hijo/hermanos
Procesos hijos	Procesos hijos del proceso
Espacio de direcciones	Descripción del mismo y sus enlaces
Recursos	Recursos poseídos por el proceso
* Pila	Posición de la base de la pila del hilo en la memoria principal

Descriptor de procesos – Bloque de Control de Procesos (PCB)

Campo	Descripción
Contador del programa	Dirección de la siguiente instrucción
Registros de la CPU	Acumuladores, registros, etc.
Información de planificación CPU	prioridad
Información de gestión de memoria	
Información del estado de E/S	Lista de dispositivos

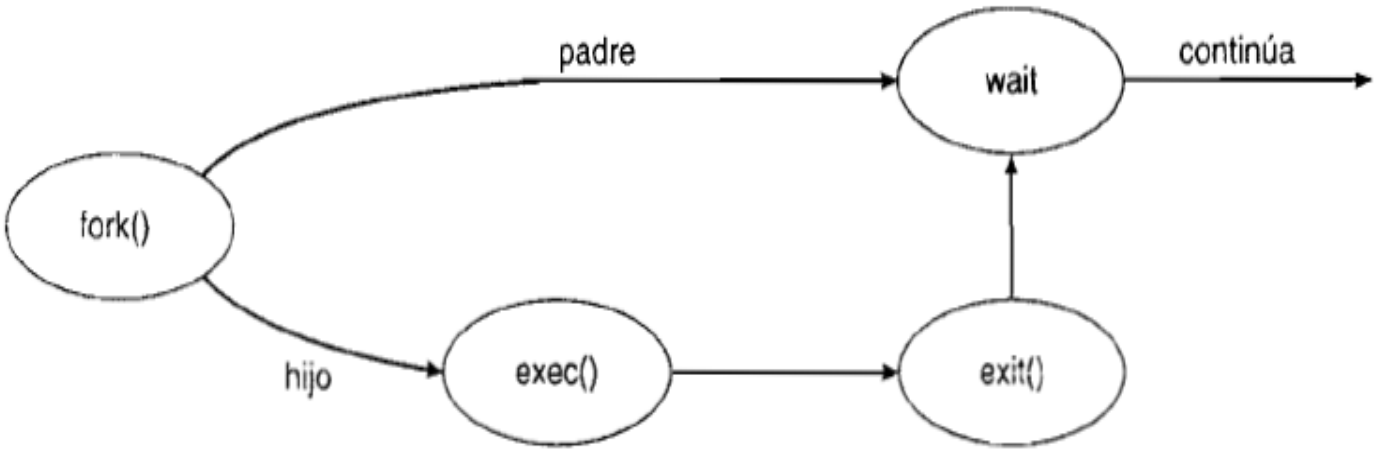
Creación de procesos

- Creación de Procesos.
  - Un proceso puede crear otros varios procesos.
  - Usa llamada al Sistema.
  - Proceso creador -- **Padre.**
  - Nuevos -- **Hijo.**
  - Los procesos se identifican mediante un identificador **PID.**

Creación de procesos

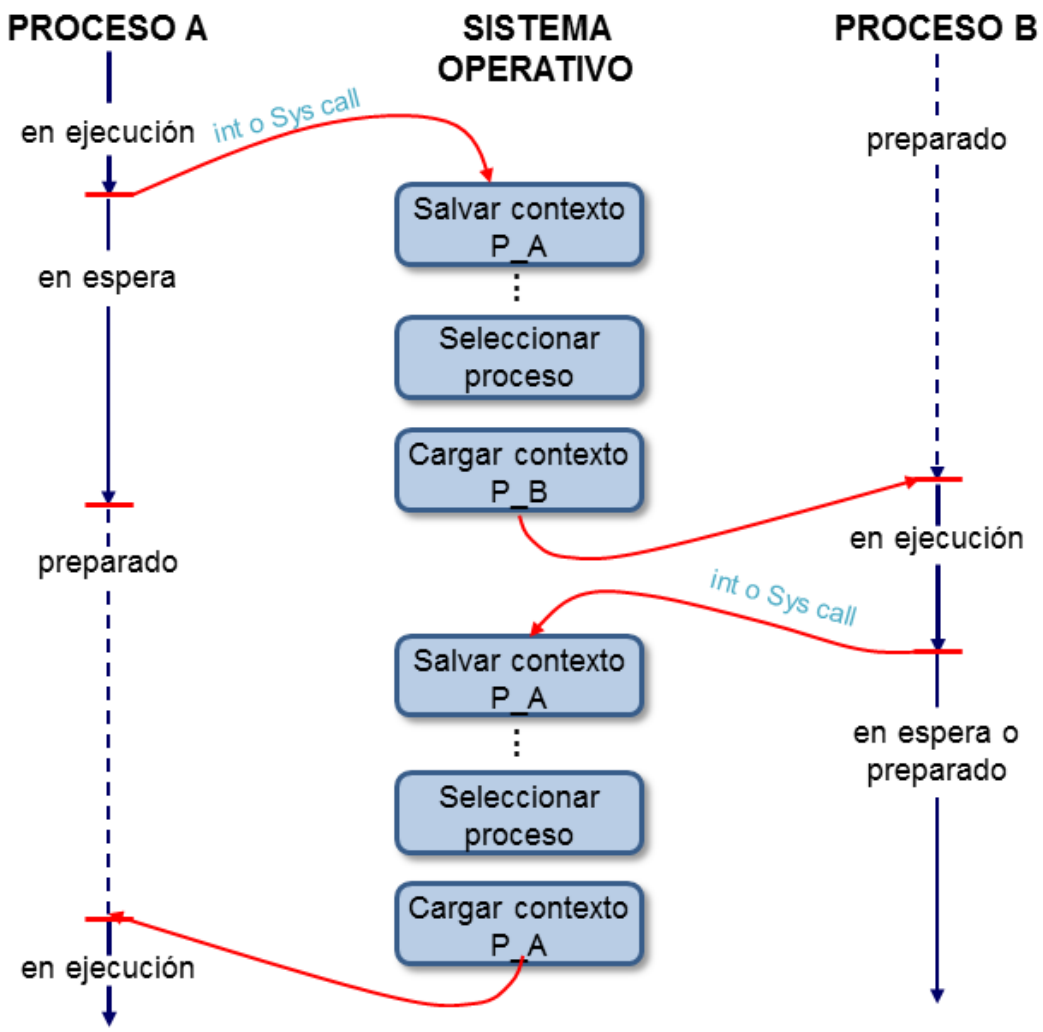
```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
pid.t pid;

    /*bifurca un proceso hijo */
    pid =fork();

    if (pid <0) {/* se produce un error */

      fprintf(strderr, "Fork Failed");
      exit(-1);
    }
    else if (pid ==0) {/* proceso hijo /
        execlp("/bin/ls/","ls", NULL);
    }
    else {/* proceso padre*/
      /* el padre espera a que el proceso hijo se complete */
      wait(NULL);
      printf("Hijo completado")
    }
}
```

Representación gráfica



Cambio de contexto

- Cuando el SO obtiene el control de procesador
  - el proceso/hilo hace una llamada al sistema
  - se produce una interrupción
- Sólo puede hacerse con una o varias instrucciones privilegiadas



Terminación de procesos

- Cuando ejecuta su ultima instrucción -: EXIT().
- Un proceso puede causar la terminación de otro.
- Un proceso termina la ejecución de uno de sus hijos por:
  - El hijo a extendido el uso de algunos de los recursos.
  - La tarea asignada al hijo ya no es necesaria.
  - El padre abandona el sistema.

Hilos

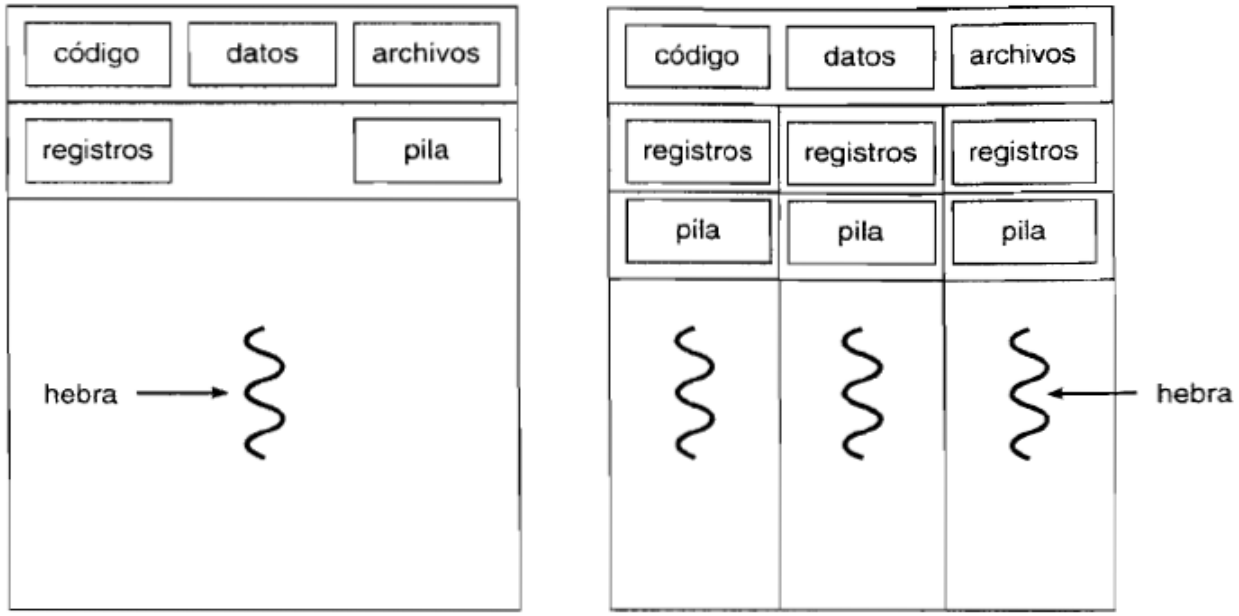
- Unidad básica de utilización de la CPU.
- Conformado por:
  - Un ID de hebra.
  - Un contador de Programa.
  - Conjunto de Registros.
  - Una Pila.
- Comparte con otras hebras.
  - Sección de código.
  - Sección de datos.
  - Otros, como archivos abiertos y señales.

Usos de los hilos

- Los hilos permiten la combinación del paralelismo con la ejecución secuencial y el bloqueo de las llamadas al sistema.
  - Trabajo interactivo y en segundo plano
    - Hojas de cálculo
  - Procesamiento asíncrono
    - Procesador de textos
    - Aceleración de la ejecución
    - Trabajo en lotes
  - Estructuración modular de los programas
    - Funciones diferentes de una aplicación

Procesos e Hilos

- Un proceso pueden ser
  - Tradicional: Monohilo.
  - Multihilo



proceso de una sola hebra

proceso multihebra

Ventajas

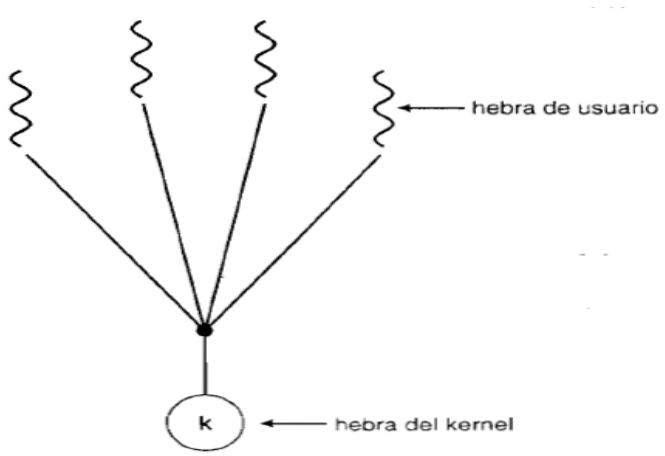
- Capacidad de respuesta
- Compartición de recursos.
- Economía.
  - La asignación de memoria y recursos en procesos es costosa.
  - Solaris crear un proceso es 30 veces mas lento que crear una hilo.
- Utilización sobre arquitecturas multiprocesador.

Modelos Multihilo

- Los hilos pueden proporcionarse en dos niveles
  - Nivel de usuario
  - Nivel de kernel
    - La mayoría de SO las soportan
  - Debe existir una relación entre los hilos de usuario y los de kernel.
    - Muchos a uno
    - Uno a uno
    - Muchos a muchos

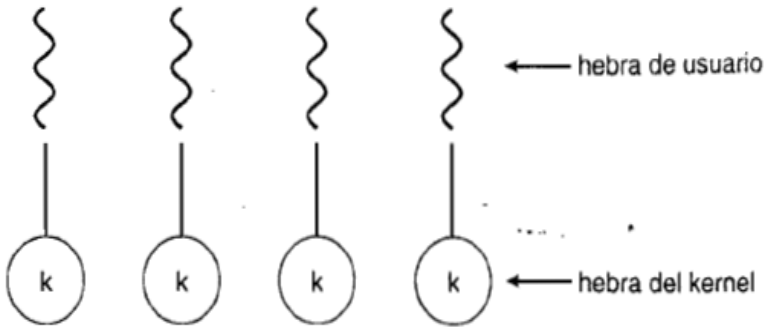
Muchos a Uno

- La gestión de hilos se hace en el espacio de usuario.
- El proceso completo se bloquea si un hilo realiza una llamada bloqueante.
- Dado que solo una hilo puede acceder al kernel, no podrán ejecutarse en paralelo sobre múltiples procesadores.
- P.e.
  - Sistema de hebras Green - Solaris
  - GNU Portable Threads



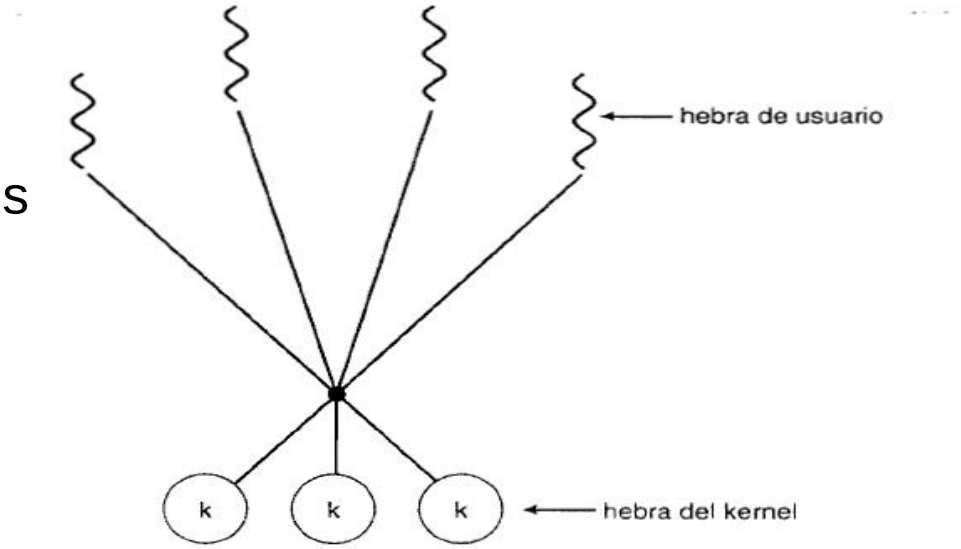
Modelo uno a uno

- Proporciona una mayor concurrencia
  - Se ejecute otro hilo mientras que otro realiza una llamada bloqueante.
- Ejecución de múltiples hilos en paralelo.
- Inconveniente.
  - Al crear una hilo de usuario se debe crear hilo de kernel, esto puede repercutir en el rendimiento
    - Mayoría de implementaciones se restringe el numero de hilos.
- P.e.
  - Sistemas Linux
  - Sistemas Windows(95, 98, NT, 2000 y XP)



Muchos a Muchos

- Multiplexa varios hilos de usuario sobre un número menor o igual de hilos de kernel.
- Pueden crear tantos hilos de usuario como sean necesarias y las de kernel pueden ejecutarse en paralelo
- P.e.
  - IRIX
  - HP-UX
  - Solaris en versiones anteriores



Aspectos del Diseño de un Paquete de Hilos

- Son un conjunto de primitivas relacionadas con los hilos (ej.: llamadas a biblioteca) disponibles para los usuarios que se llama un “paquete de hilos”.
- Respecto del manejo de los hilos se tienen
  - Hilos estáticos
  - Hilos dinámicos.

Aspectos del Diseño de un Paquete de Hilos

- **En un diseño estático**:
  - Se elige el número de hilos al escribir el programa o durante su compilación.
  - Cada uno de ellos tiene asociada una pila fija.
  - Se logra simplicidad pero también inflexibilidad.

Aspectos del Diseño de un Paquete de Hilos

- **En un diseño dinámico:**
  - Se permite la creación y destrucción de los hilos durante la ejecución.
  - La llamada para la creación de hilos determina:
    - El programa principal del hilo.
    - Un tamaño de pila.
    - Una prioridad de planificación, etc.
  - La llamada generalmente regresa un identificador de hilo:
    - Se usará en las posteriores llamadas relacionadas al hilo.
  - Un proceso:
    - Se inicia con un solo hilo.
    - Puede crear el número necesario de hilos.

Implantación de un Paquete de Hilos

- Un paquete de hilos se puede implantar en el espacio:
  - Del usuario.
  - Del núcleo.

Hilos a nivel de usuario

- Implantación del paquete de hilos en el espacio del usuario:
  - El núcleo no sabe de su existencia.
  - El núcleo maneja procesos con un único hilo.
  - No requiere soporte de hilos por parte del S. O.
  - Los hilos se ejecutan en un sistema de tiempo de ejecución:
    - Es un grupo de procedimientos que manejan los hilos.

+ La conmutación de hilos es más rápida
+ Cada proceso puede tener su propia planificación
+ Portabilidad entre SO diferentes
– El bloqueo de un hilo genera el bloqueo del resto de hilos de una tarea
– Dos hilos de una misma tarea no pueden ejecutarse en procesadores diferentes

Hilos a nivel de núcleo

- Implantación del paquete de hilos en el espacio del núcleo:
  - No se necesita un sistema de tiempo de ejecución.
  - Para cada proceso el núcleo tiene una tabla con una entrada por cada hilo que contiene: Los registros, estados, prioridades y demás información relativa al hilo.
  - Todas las llamadas que pueden bloquear un hilo se implantan como llamadas al sistema.
  - Cuando un hilo se bloquea, el núcleo puede ejecutar:
    - Otro hilo listo del mismo proceso.
    - Un hilo de otro proceso

+ Si un hilo se bloquea no afecta a el resto de hilos de la tarea
– La conmutación de un hilo a otro se hace por medio de interrupciones (mayor sobrecarga)

Biblioteca de Hilos

- Proporciona al programador una API para gestionar.
- Dos formas de implementar una biblioteca.
  - En el espacio de usuario sin ningún soporte de kernel.
    - Invocar a una función de la biblioteca = realizar una llamada a una función local.
  - En el Kernel. Soportada por el Sistema Operativo.
    - Se encuentran en el especio del kernel
    - Al llamar a una función = se da una llamada al Sistema
- Principales Bibliotecas
  - POSIX Pthreads, Win32, Java Pthreads