



## Procesos e Hilos

### Segunda Parte

#### Objetivos

- ❖ Adquirir los conceptos básicos sobre procesos en sistemas operativos.

#### Preguntas

#### Procesos e Hilos

**1. ¿A que se le llama traza de un proceso? ¿y traza combinada?**

**Traza de un proceso** Secuencia de instrucciones que se ejecutan cuando un proceso está corriendo.

**Traza combinada** se denomina a la secuencia de instrucciones que el CPU ejecuta

- 2. Suponga que una computadora tiene tres procesos (P1, P2, P3) para ejecutar, las direcciones de memoria (DM) de las instrucciones son las mostradas en los cuadros, el P2 tiene una instrucción de E/S en la dirección 723A, se estima que la entrada y salida tardara unos 10 ciclos de reloj, el S.O. es de tiempo compartido y asigna un tiempo máximo de 16 ciclos de reloj para cada proceso en ejecución, el CPU trabaja ejecutando 1 instrucción cada 4 ciclos de reloj. ¿Cual es la traza combinada de los tres procesos?**

**Nota :** los procesos se ejecutan en orden P1, P2, P3, P1,P2, P3, P1..., si alguno termina su ejecución, el planificador de procesos ejecuta de igual forma con los procesos que quedan (Round-Robin). Cuando ocurre una E/S, el planificador pasa a otro proceso, y retorna cuando termina la E/S, como máximo cada proceso tiene para ejecutar instrucciones 16 ciclos de reloj (sin contar las E/S).

	DM	
P1	0100	Inicio
	0101	
	0103	
	0105	
	0109	
	010B	
	010D	
		Fin

	DM	
P2	7231	Inicio
	7234	
	7236	
	7238	
	723A	
	723C	
	723D	
		E/S

	DM	
P3	8021	Inicio
	8022	
	8023	
	8024	
	8025	
		fin

--	--	--

	723F	fin
--	------	-----

--	--	--

0100	
0101	
0103	P1
0105	
7231	
7234	
7236	P2
7238	
8021	
8022	
8023	P3
8024	
0109	
010B	P1
010D	FIN P1
723A	P2
723C	P2 E/S
8025	FIN P3
	Espera E/S P2
723D	P2
723F	FIN P2

3. ¿Cuáles son los eventos que producen la creación de procesos? De un ejemplo de cada uno.

**Nuevo proceso de lotes:** El sistema operativo dispone de un flujo de control de lotes de trabajos, habitualmente una cinta un disco. Cuando el sistema operativo está listo para procesar un nuevo trabajo, leerá la siguiente secuencia de mandatos de control de trabajos.

**Sesión interactiva:** Un usuario desde un terminal entra en el sistema.

**Creado por el sistema operativo para crear un servicio:** El sistema operativo puede crear un proceso para realizar una función en representación de un programa de usuario, sin que el usuario tenga que esperar (por ejemplo, un proceso para controlar la impresión).

**Creado por un proceso existente:** Por motivos de modularidad o para explotar el paralelismo, un programa de usuario puede ordenar la creación de un número de procesos.

4. ¿Como se llama al hecho de que un proceso sea creado por otro? ¿Que nombre recibe el proceso creador y el proceso creado?

Cuando un proceso lanza otro, **al primero se le denomina proceso padre**, y **al proceso creado se le denomina proceso hijo**

## **5. De ejemplos de terminación de procesos.**

### **Finalización normal**

El proceso ejecuta una llamada al sistema operativo para indicar que ha completado su ejecución

### **Límite de tiempo excedido**

El proceso ha ejecutado más tiempo del especificado en un límite máximo. Existen varias posibilidades para medir dicho tiempo. Estas incluyen el tiempo total utilizado, el tiempo utilizado únicamente en ejecución, y, en el caso de procesos interactivos, la cantidad de tiempo desde que el usuario realizó la última entrada.

### **Memoria no disponible**

El proceso requiere más memoria de la que el sistema puede proporcionar.

### **Violaciones de frontera**

El proceso trata de acceder a una posición de memoria a la cual no tiene acceso permitido.

### **Error de protección**

El proceso trata de usar un recurso, por ejemplo, un fichero, al que no tiene permitido acceder, o trata de utilizarlo de una forma no apropiada, por ejemplo, escribiendo en un fichero de sólo lectura.

### **Error aritmético**

El proceso trata de realizar una operación de cálculo no permitida, tal como una división por 0, o trata de almacenar números mayores de los que la representación hardware puede codificar.

### **Límite de tiempo**

El proceso ha esperado más tiempo que el especificado en un valor máximo para que se cumpla un determinado evento.

### **Fallo de E/S**

Se ha producido un error durante una operación de entrada o salida, por ejemplo la imposibilidad de encontrar un fichero, fallo en la lectura o escritura después de un límite máximo de intentos (cuando, por ejemplo, se encuentra un área defectuosa en una cinta), o una operación inválida (la lectura de una impresora en línea).

### **Instrucción no válida**

El proceso intenta ejecutar una instrucción inexistente (habitualmente el resultado de un salto a un área de datos y el intento de ejecutar dichos datos).

### **Instrucción privilegiada**

El proceso intenta utilizar una instrucción reservada al sistema operativo.

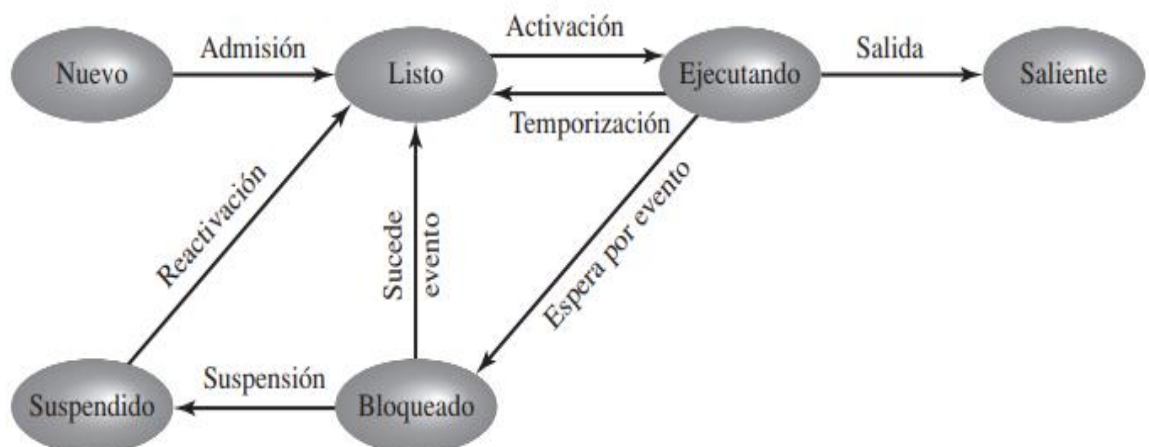
### **Uso inapropiado de datos**

Una porción de datos es de tipo erróneo o no se encuentra inicializada.

### **Intervención del operador por el sistema operativo**

Por alguna razón, el operador o el sistema operativo ha finalizado el proceso (por ejemplo, se ha dado una condición de interbloqueo).

### **Terminación del proceso padre**



**10. ¿Cuales son las estructuras de control mas importantes del SO? ¿Para que se utilizan cada una de ellas?**

**Tablas de memoria:** se usan para mantener un registro tanto de la memoria principal (real) como de la secundaria (virtual). Estas deben incluir la siguiente información:

- Las reservas de memoria principal por parte de los procesos
- Las reservas de memoria secundaria por parte de los procesos.
- Todos los atributos de protección que restringe el uso de la memoria principal y virtual, de forma que los procesos puedan acceder a ciertas áreas de memoria compartida.
- La información necesaria para manejar la memoria virtual.

**Tablas de E/S:** se usan para gestionar los dispositivos de E/S y los canales del computador.

**Tablas de ficheros:** estas tablas otorgan información sobre la existencia de ficheros, su posición en almacenamiento secundario, su estado actual, y otros atributos. La mayoría de, o prácticamente toda, esta información se puede gestionar por el sistema de ficheros.

**Tablas de procesos:** se implementan para gestionar los procesos. Estas tablas deben tener información sobre dónde están localizados los procesos, y debe conocer los atributos de los procesos que quiere gestionar (por ejemplo, identificador de proceso y estado del mismo).

## 11. ¿Como esta formada la “información del estado del proceso”?

### Registros visibles por el usuario

Un registro visible por el usuario es aquel al que se puede hacer referencia por medio del lenguaje máquina que ejecuta el procesador cuando está en modo usuario.

Normalmente, existen de 8 a 32 de estos registros, aunque determinadas implementaciones RISC tienen más de 100.

### Registros de estado y control

Hay una gran variedad de registros del procesador que se utilizan para el control de operaciones. Estos incluyen:

- Contador de programa: contiene la dirección de la siguiente instrucción a ejecutar.
- Códigos de condición: resultan de la operación lógica o aritmética más reciente (por ejemplo, signo, cero, acarreo, igual, desbordamiento).
- Información de estado: incluyen los flags de interrupciones habilitadas/deshabilitadas, modo ejecución.

## 12. ¿Que realiza el SO cuando crea un proceso?

crear un proceso procederá de la siguiente manera:

1. **Asignar un identificador de proceso** único al proceso. En este instante, se añade una nueva entrada a la tabla primaria de procesos, que contiene una entrada por proceso.
2. **Reservar espacio para proceso.** Esto incluye todos los elementos de la imagen del proceso. Para ello, el sistema operativo debe conocer cuánta memoria se requiere para el espacio de direcciones privado (programas y datos) y para la pila de usuario. Estos valores se pueden asignar por defecto basándonos en el tipo de proceso, o pueden fijarse en base a la solicitud de creación del trabajo remitido por el usuario. Si un proceso es creado por otro proceso, el proceso padre puede pasar los parámetros requeridos por el sistema operativo como parte de la solicitud de la creación de proceso. Si existe una parte del espacio direcciones compartido por este nuevo proceso, se fijan los enlaces apropiados. Por último, se debe reservar el espacio para el bloque de control de proceso (BCP).
3. **Inicialización del bloque de control de proceso.** La parte de identificación de proceso del BCP contiene el identificador del proceso, así como otros posibles identificadores, tal como el indicador del proceso padre. En la información de estado de proceso del BCP, habitualmente se inicializa con la mayoría de entradas a 0, excepto el contador de programa (fijado en el punto entrada del programa) y los punteros de pila de sistema (fijados para definir los límites de la pila del proceso). La parte de información de control de procesos se inicializa en base a los valores por omisión, considerando también los atributos que han sido solicitados para este proceso. Por ejemplo, el estado del proceso se puede inicializar normalmente a Listo o Listo/Suspendido. La prioridad se puede fijar, por defecto, a la prioridad más baja, a menos que una solicitud explícita la eleve a una prioridad mayor. Inicialmente, el

proceso no debe poseer ningún recurso (dispositivos de E/S, ficheros) a menos que exista una indicación explícita de ello o que haya sido heredados del padre.

**4. Establecer los enlaces apropiados.** Por ejemplo, si el sistema operativo mantiene cada cola del planificador como una lista enlazada, el nuevo proceso debe situarse en la cola de Listos o en la cola de Listos/Suspendidos.

**5. Creación o expansión de otras estructuras de datos.** Por ejemplo, el sistema operativo puede mantener un registro de auditoría por cada proceso que se puede utilizar posteriormente a efectos de facturación y/o de análisis de rendimiento del sistema.

### 13. ¿Cual es la diferencia entre un proceso y un hilo?

Las diferencias entre un proceso y un hilo son las siguientes:

1. Lleva mucho menos tiempo crear un nuevo hilo en un proceso existente que crear un proceso totalmente nuevo. Los estudios realizados por los que desarrollaron el sistema operativo Mach muestran que la creación de un hilo es diez veces más rápida que la creación de un proceso en UNIX [TEVA87].
2. Lleva menos tiempo finalizar un hilo que un proceso.
3. Lleva menos tiempo cambiar entre dos hilos dentro del mismo proceso
4. Los hilos mejoran la eficiencia de la comunicación entre diferentes programas que están ejecutando. En la mayor parte de los sistemas operativos, la comunicación entre procesos independientes requiere la intervención del núcleo para proporcionar protección y los mecanismos necesarios de comunicación. Sin embargo, ya que los hilos dentro de un mismo proceso comparten memoria y archivos, se pueden comunicar entre ellos sin necesidad de invocar al núcleo.

### 14. ¿Que son “unidad propietaria de recursos” y “unidad de ejecución”?

**Propiedad de recursos.** Un proceso incluye un espacio de direcciones virtuales para el manejo de la imagen del proceso (la imagen de un proceso es la colección de programa, datos, pila y atributos definidos en el bloque de control del proceso) De vez en cuando a un proceso se le puede asignar control o propiedad de recursos tales como la memoria principal, canales E/S, dispositivos E/S y archivos. El sistema operativo realiza la función de protección para evitar interferencias no deseadas entre procesos en relación con los recursos.

**Planificación/ejecución.** La ejecución de un proceso sigue una ruta de ejecución (traza) a través de uno o más programas. Esta ejecución puede estar intercalada con ese u otros procesos. De esta manera, un proceso tiene un estado de ejecución (Ejecutando, Listo, etc.) y una prioridad de activación y ésta es la entidad que se planifica y activa por el sistema operativo.

La diferencia entre estos dos es que **la unidad de propiedad de recursos** se suele denominar **proceso** o tarea, mientras que **la unidad que se activa o se ejecuta** se suele denominar **hilo** o proceso ligero.

### 15. ¿Qué significa multihilo? ¿Cuáles son las diferencias entre modelo multihilo y monohilo?

Multihilo se refiere a la capacidad de un sistema operativo de dar soporte a múltiples hilos de ejecución en un solo proceso, es decir que, hilos de diferentes procesos se puedan ejecutar en paralelo.

Un proceso multihilo puede lograr la concurrencia sin la sobrecarga del uso de múltiples procesos. Los hilos del mismo proceso pueden intercambiar información a través de su

espacio de direcciones común y tienen acceso a los recursos compartidos del proceso. Los hilos de diferentes procesos pueden intercambiar información a través del uso de memoria compartida. Las diferencias entre modelo multihilo y modelo monohilo es que cuando manejamos varios hilos tenemos un bloque de control por hilo y un bloque que controla todos esos hilos en cambio cuando manejamos un solo hilo tenemos un solo bloque de control.

#### 16. ¿Un hilo puede estar en estado suspendido? Explique.

Estados de los hilos. Igual que con los procesos, los principales estados de los hilos son: Ejecutando, Listo y Bloqueado. Generalmente, no tiene sentido aplicar estados de suspensión a un hilo, ya que dichos estados son conceptos de nivel de proceso. En particular, si se expulsa un proceso, todos sus hilos se deben expulsar porque comparten el espacio de direcciones del proceso.

**En resumen: Un hilo no puede estar en estado suspendido puesto a que un hilo no posee recursos, es solo una unidad de ejecución.** Un hilo está en el mismo estado en el que se encuentra su proceso padre.

#### 17. ¿Qué tipos de hilos existen? Detalle cada uno (a nivel de núcleo y a nivel de usuario)

Los tipos de hilos que existen son dos: A **nivel de usuario**, y a **nivel de núcleo**, y un **híbrido entre los dos**.

**Nivel de usuario:** En un entorno ULT puro, la aplicación gestiona todo el trabajo de los hilos y el núcleo no es consciente de la existencia de los mismos. Cualquier aplicación puede programarse para ser multihilo.

Por defecto, una aplicación comienza con un solo hilo y ejecutando en ese hilo. Esta aplicación y su hilo se localizan en un solo proceso gestionado por el núcleo. En cualquier momento que la aplicación esté ejecutando (el proceso está en estado Ejecutando), la aplicación puede crear un nuevo hilo a ejecutar dentro del mismo proceso.

**El uso de ULT en lugar de KLT, presenta las siguientes ventajas:**

1. **El cambio de hilo no requiere privilegios de modo núcleo** porque todas las estructuras de datos de gestión de hilos están en el espacio de direcciones de usuario de un solo proceso. Por consiguiente, el proceso no cambia a modo núcleo para realizar la gestión de hilos. Esto ahorra la sobrecarga de dos cambios de modo (usuario a núcleo; núcleo a usuario).
2. **La planificación puede especificarse por parte de la aplicación.** Una aplicación se puede beneficiar de un simple algoritmo de planificación cíclico, mientras que otra se podría beneficiar de un algoritmo de planificación basado en prioridades. El algoritmo de planificación se puede hacer a medida sin tocar el planificador del sistema operativo.
3. **Los ULT pueden ejecutar en cualquier sistema operativo.** No se necesita ningún cambio en el nuevo núcleo para dar soporte a los ULT. La biblioteca de los hilos es un conjunto de utilidades a nivel de aplicación que comparten todas las aplicaciones.

**Hay dos desventajas de los ULT en comparación con los KLT:**

1. **En un sistema operativo típico muchas llamadas al sistema son bloqueantes.** Como resultado, cuando un ULT realiza una llamada al sistema, no sólo se bloquea ese hilo, sino que se bloquean todos los hilos del proceso.
2. **En una estrategia pura ULT, una aplicación multihilo no puede sacar ventaja del multiproceso.** El núcleo asigna el proceso a un solo procesador al mismo tiempo. Por consiguiente, en un determinado momento sólo puede ejecutar un hilo del proceso. En efecto, tenemos multiprogramación a nivel de aplicación con un solo proceso. Aunque esta multiprogramación puede dar lugar a una mejora significativa de la velocidad de la aplicación, hay aplicaciones que se podrían beneficiar de la habilidad de ejecutar porciones de código de forma concurrente.



**Nivel de núcleo:** En un entorno KLT puro, el núcleo gestiona todo el trabajo de gestión de hilos. No hay código de gestión de hilos en la aplicación, solamente API para acceder a las utilidades de hilos del núcleo. Windows es un ejemplo de este enfoque.

La principal desventaja del enfoque KLT en comparación con el enfoque ULT es que la transferencia de control de un hilo a otro del mismo proceso requiere un cambio de modo al núcleo.

Las dos medidas para la transferencia de control de un hilo a otro son las siguientes: **Crear un Proceso Nulo**, el tiempo para crear, planificar, ejecutar, y completar un proceso/hilo que llama al **procedimiento nulo** (es decir, la sobrecarga de crear un proceso/hilo); y Señalizar-Esperar, el tiempo que le lleva a un proceso/hilo señalar a un proceso/hilo que está esperando y a continuación esperar una condición (es decir, la sobrecarga de sincronizar dos procesos/hilos).

De esta forma, mientras que hay una ganancia significativa entre el uso de multihilos KLT en comparación con procesos de un solo hilo, hay una ganancia significativa adicional por el uso de ULT. Sin embargo, depende de la naturaleza de la aplicación involucrada que nos podamos beneficiar o no de la ganancia adicional. Si la mayor parte de los cambios de hilo en una aplicación requieren acceso al modo núcleo, el esquema basado en ULT no sería tan superior al esquema basado en KLT.