

Tema 8

Procesos

8.1 Aspectos básicos de los procesos

8.1.1 Concepto de proceso

* Definición informal: un proceso es un programa en ejecución

Un programa ejecutable es un conjunto de instrucciones y datos almacenados en un fichero. Cuando lo que tiene ese programa se carga en la memoria y se pone en ejecución, se convierte en un proceso.

* Definición técnica: un proceso es una entidad formada por los siguientes elementos principales:

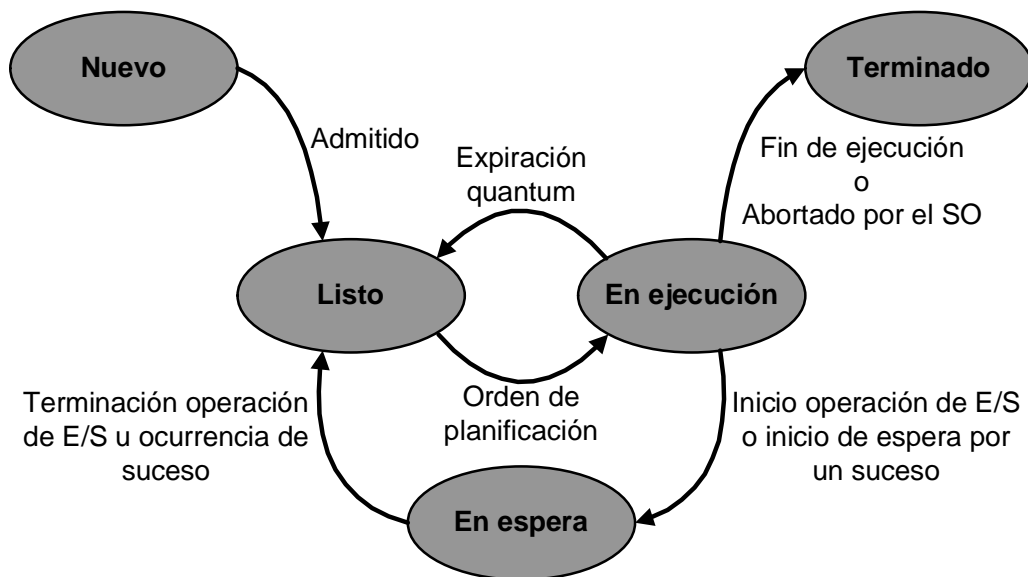
- Una imagen binaria de un programa, cargada total o parcialmente en la memoria física. La imagen binaria está formada por las instrucciones y datos del programa.
- Un área de memoria para almacenar datos temporales, conocida como pila.

La imagen binaria y la pila son el programa en si mismo, pero para que el SO pueda controlar el programa hacen falta una serie de estructuras de datos. Las estructuras fundamentales son:

- La tabla de páginas para traducir las direcciones virtuales generadas por el proceso en las direcciones físicas en la que se encuentra almacenado.
- Una estructura de control, conocida como PCB, para que el sistema operativo pueda controlar su ejecución.

8.1.2 Estados de un proceso

Un proceso pasa por varios estados durante su ejecución. Los estados posibles para un proceso se muestran en la figura siguiente:



En la figura anterior los nodos (nuevo, listo, etc.) representan los estados y los arcos, las acciones o eventos que llevan a un cambio de estado.

* Definición de los estados:

- **Nuevo:** El proceso se acaba de crear, pero aún no ha sido admitido en el grupo de procesos ejecutables por el sistema operativo.

Habitualmente en un sistema operativo multitarea como Windows, nada más que un proceso se crea, éste resulta admitido, pasando al estado listo. Sin embargo, esto no tiene por qué ser siempre así. Por ejemplo, en una situación de sobrecarga temporal del sistema, el SO puede decidir retardar la admisión de los procesos nuevos. Así se alivia la carga del sistema, ya que hasta que un proceso no es admitido, éste no compite por los recursos del sistema.

- **Listo:** El proceso está esperando ser asignado al procesador para su ejecución.

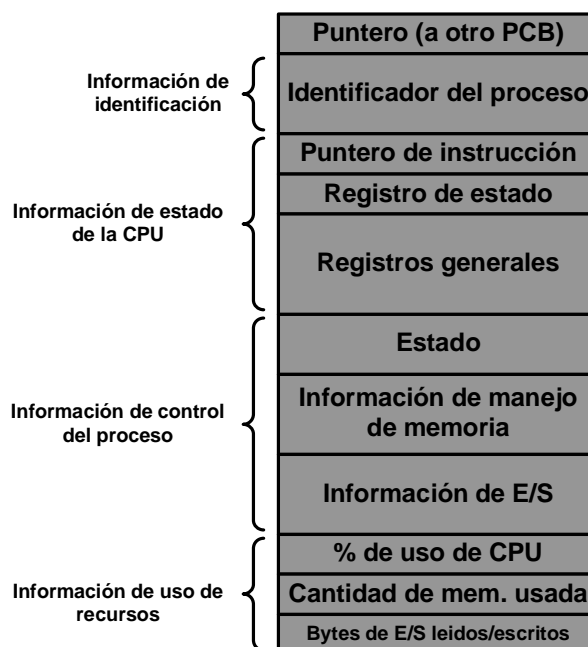
Una CPU clásica (con un solo núcleo) solo se puede dedicar en cada momento a un proceso. Los procesos que están preparados para ejecutarse permanecen en estado listo hasta que se les concede la CPU. Entonces pasan al estado "En ejecución".

- **En ejecución:** El proceso tiene la CPU y ésta ejecuta sus instrucciones.
- **En espera:** El proceso está esperando a que ocurra algún suceso, como por ejemplo la terminación de una operación de E/S.
- **Terminado:** El proceso ha sido sacado del grupo de procesos ejecutables por el sistema operativo. Después de que un proceso es marcado como terminado se liberarán los recursos utilizados por ese proceso, por ejemplo, la memoria.

8.1.3 Bloque de control de proceso (PCB)

- * PCB = Process Control Block
- * Definición: Es una estructura de datos que permite al sistema operativo controlar diferentes aspectos de la ejecución de un proceso.
- * Estructura típica del PCB de un proceso:

El PCB se organiza en un conjunto de campos en los que se almacena información de diversos tipos. Los campos típicamente mantenidos en el PCB de un proceso se muestran en la figura siguiente:



- * Información típica mantenida en el PCB:

Puede clasificarse en cuatro categorías:

- Información de identificación

Esta información está integrada básicamente por el **identificador del proceso** (PID), que es un número que identifica al proceso. Este número es diferente para todos los procesos que se encuentran en ejecución.

- Información de estado de la CPU

Se trata de un conjunto de campos que almacenan el estado de los registros de la CPU cuando el proceso es suspendido.

- Información de control del proceso

Se trata de un conjunto de información que es utilizada por el sistema operativo para controlar diversos aspectos de funcionamiento del proceso. Pertenecen a esta categoría de información los siguientes campos:

- **Estado del proceso**: Listo, en ejecución, etc.
- **Información de manejo de memoria**: Como por ejemplo, la dirección física de memoria en la que se ubica la tabla de páginas del proceso.

– **Información de E/S:** Lista de ficheros abiertos, ventanas utilizadas, etc.

- Información de uso de recursos

Se trata de un conjunto de información relativa a la utilización realizada por el proceso de los recursos del sistema, como por ejemplo, el porcentaje de utilización de la CPU, la cantidad de memoria usada o los bytes de E/S escritos y leídos por el proceso.

8.2 Planificación de procesos

El objetivo de los sistemas multitarea es mantener múltiples programas en ejecución simultáneamente, pero como la CPU sólo puede ejecutar un programa de cada vez, hay que decidir quién se ejecuta en cada momento.

Se denomina planificación (*scheduling*) al mecanismo utilizado por el sistema operativo para determinar qué proceso (entre los presentes en el sistema) debe ejecutarse en cada momento.

8.2.1 Planificación en sistemas de tiempo compartido

- * Los sistemas operativos más importantes del mercado actual (Windows, Linux, Mac OS y todas las versiones de Unix) se consideran sistemas operativos de tiempo compartido.
- * Objetivo prioritario de estos sistemas: Garantizar que el tiempo de respuesta de los programas se mantiene en unos valores admisibles para los usuarios.

Cuando un usuario interacciona con un programa y le da una orden, quiere que el programa responda en un tiempo razonable. Para conseguir esto hay que hacer que el resto de programas que se encuentren en ejecución no monopolicen la CPU. Para ello, hay que ir repartiendo la CPU entre todos los programas, y además muy rápidamente, para que cada programa tenga una fracción del recurso CPU cada muy poco tiempo.

- * Esquema de funcionamiento: A cada proceso en ejecución se le asigna un *quantum*, que representa el tiempo máximo que puede estar ocupando la CPU. Entonces un proceso abandona la CPU, o bien cuando se bloquea por una operación de E/S (pasando al estado “en espera”), o bien cuando expira su *quantum* (pasando al estado “listo”).)

8.2.2 Colas de planificación

- * Son unas estructuras de datos que organizan los PCBs de los procesos que se encuentran cargados en el sistema en función de su estado.
- * El SO planifica los procesos en función de la información mantenida en estas colas.
- * Estas estructuras se forman enlazando los PCBs de los procesos mediante punteros.

* Existen dos tipos de colas:

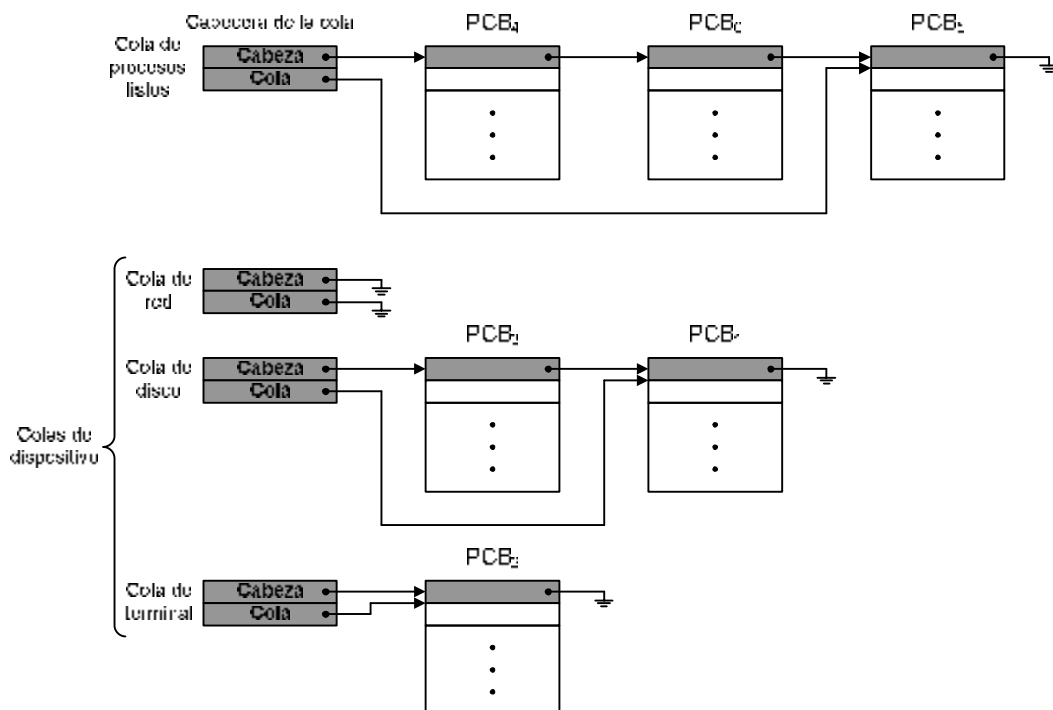
- **Cola de procesos listos:** Contiene a los procesos que se encuentran en el estado “listo”.

Debe indicarse una vez más que estos procesos son los que están preparados para ser asignados a la CPU.

- **Cola de dispositivo:** Contiene los procesos que están esperando por un determinado dispositivo. Estos procesos se encuentran en el estado “En espera”. Cada dispositivo tiene una cola asignada.

Hay muchos dispositivos, como por ejemplo el disco, que son intensivamente utilizados por muchos procesos. Los procesos deben esperar ordenadamente para poder utilizar este recurso.

* En la figura siguiente se muestran estos dos tipos de colas:



Las colas se forman enlazando mediante punteros los PCBs de los procesos. El primer campo del PCB es un puntero que se usa para formar estas colas.

La cola tiene una cabecera que contiene dos punteros (llamados cabeza y cola) que se usan para apuntar al primer y último proceso de la cola.

8.2.3 Concepto de cambio de contexto (*context switch*)

- * Es el hecho de abandonar la ejecución de un proceso y poner en marcha otro proceso.
- * El cambio de contexto requiere salvar el estado que tienen los registros de la CPU justo antes de que ésta abandone el proceso que se saca de ejecución. Así, después, se podrá reanudar la ejecución de este proceso, justo en el punto en el que se suspendió su ejecución. El estado de los registros de la CPU se salva en el PCB del proceso.

8.2.4 Concepto de swapping (intercambio)

- * Se trata de un mecanismo que permite sacar procesos de ejecución, salvándolos en el disco, para luego volver a ponerlos en ejecución cuando sea requerido.
- * El objetivo del “swapping” es aliviar al sistema, cuando su carga de trabajo es demasiado alta, suspendiendo temporalmente en el disco unidades de trabajo (procesos). Cuando la carga del sistema baja, se ponen de nuevo en ejecución los procesos temporalmente suspendidos. Al final se conseguirá mejorar el rendimiento global del sistema multitarea.

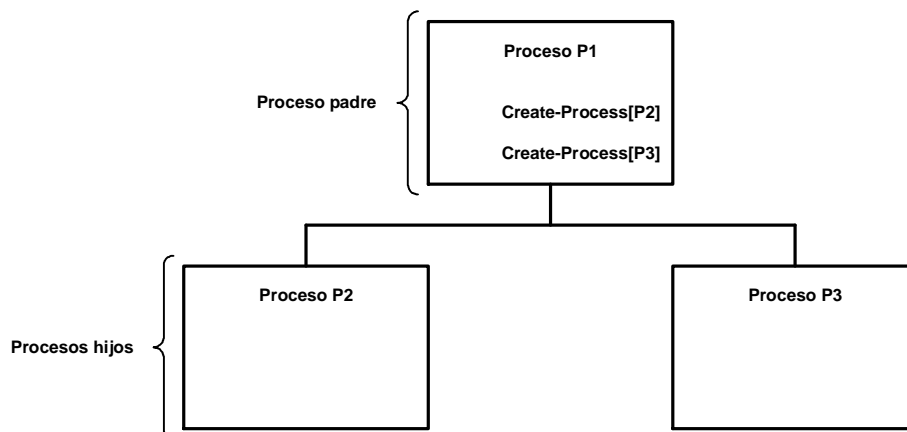
Debe observarse la clara diferencia existente entre los mecanismos del “cambio de contexto” y del swapping.

8.3 Operaciones sobre procesos

Los procesos tienen que poder ser creados y eliminados dinámicamente en el sistema. Debido a ello, el sistema debe proporcionar facilidades para llevar a cabo estas acciones con los procesos. Las funcionalidades básicas se indican a continuación.

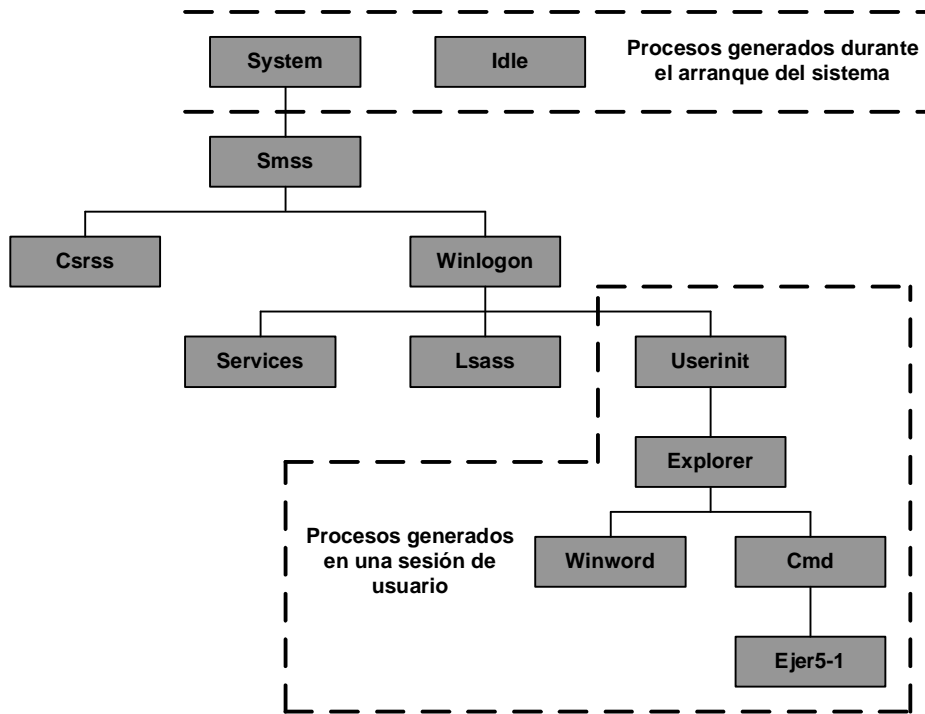
8.3.1 Creación de procesos

- * Todo sistema operativo debe proporcionar un servicio Create-Process, que será utilizado por un proceso para crear otro proceso.
- * Al proceso que solicita el servicio Create-Process se le denomina proceso padre, y al proceso que es creado mediante este servicio, proceso hijo.
- * Ejemplo:



- * Este mecanismo de generación de procesos tiene como consecuencia que las relaciones de parentesco entre los procesos existentes en un sistema tenga estructura de árbol.

* Ejemplo de árbol de procesos típico de una plataforma Windows



Siempre debe haber un proceso raíz, que será creado durante el arranque del sistema. En el caso de Windows este proceso se llama System.

También se crea durante el arranque el proceso idle, que es un proceso que no hace nada. Este proceso es el que se ejecuta cuando la CPU no tiene ningún otro proceso para ejecutar.

A partir de System se van generando otra serie de procesos (Smss, Csrss, Winlogon, etc.) que llevan a cabo labores vitales en el sistema y que por tanto permanecen en ejecución mientras el sistema esté en funcionamiento.

Winlogon controla el inicio de las sesiones de usuario. Cuando un usuario se autentica en el sistema, Winlogon genera el proceso Userinit para cargar el perfil de usuario y poner en marcha el proceso Explorer que proporciona la interfaz con usuario. A partir del Explorer se irán generando nuevos procesos según sea requerido por el usuario.

8.3.2 Terminación de procesos

- * Un proceso puede terminar por sí mismo, o bien puede ser terminado por otro proceso, que generalmente sólo puede ser su proceso padre.
- * Un proceso termina por sí mismo llamando a un servicio del sistema, denominado normalmente Exit o Exit-Process.

En los programas sencillos que llevamos a cabo en las prácticas, normalmente no hacemos ninguna llamada al sistema para terminar el proceso, pero ésta es siempre insertada por el sistema de desarrollo. Debe tenerse en cuenta que en la imagen binaria de un programa hay bastante más código que el que explícitamente escribe el usuario.

- * Un proceso puede terminar la ejecución de un proceso hijo llamando a un servicio del sistema, conocido normalmente como Abort o Terminate-Process.

El que un proceso haga que termine otro proceso es una situación extraordinaria, normalmente ligada a la ocurrencia de errores. Cuando las cosas van bien, los procesos terminan por sí mismos.

8.4 Cooperación entre procesos

En la mayoría de las ocasiones los procesos son entidades totalmente aisladas: llevan a cabo su trabajo sin tener que comunicarse con otros procesos o programas. Todos los programas realizados en las prácticas de la asignatura son así. Son programas muy simples que no necesitan comunicarse con otros programas. Sin embargo las cosas en la realidad no son tan sencillas. En muchas ocasiones, los programas o procesos necesitan intercambiar información entre sí. Pongamos dos ejemplos:

- 1) En una plataforma Windows, el intercambio de información a través del portapapeles.*
- 2) Chatear a través de la red. Hay dos procesos (dos navegadores) que intercambian información.*

Son dos casos totalmente diferentes, pero son dos ejemplos claros de programas que cooperan entre sí.

La cooperación entre procesos requiere que estos se comuniquen. A continuación se indican los mecanismos básicos de comunicación:

* Memoria compartida

- Se basa en que los procesos que desean comunicarse compartan una misma región de memoria física. Para llevar a cabo la comunicación, uno escribe y otro lee de la región de memoria compartida.
- Los procesos utilizan servicios del sistema operativo para compartir la región.

* Paso de mensajes

- Los procesos utilizan una pareja de servicios del sistema operativo para comunicarse. Estos servicios son conocidos habitualmente como *Send* y *Receive*.
- Para llevar a cabo la comunicación un proceso ejecuta la función *Send* y el otro *Receive*, intercambiando de esta forma un bloque de información que recibe el nombre de mensaje.

NOTA:

Los párrafos escritos sobre fondo gris y recuadrados mediante línea discontinua contienen información complementaria al resto del contenido de estos apuntes. De cara al examen, los conocimientos fundamentales que el alumno debe adquirir son los que se encuentran fuera de estos recuadros.