**Unidad 2: Procesos y Planificación de Procesos**

Contenido

[Procesos: Definición 1](#_Toc109131983)

[El modelo del proceso 2](#_Toc109131984)

[Creación de un proceso 4](#_Toc109131985)

[El arranque del sistema 4](#_Toc109131986)

[La ejecución, desde un proceso 4](#_Toc109131987)

[Una Petición de Usuario 5](#_Toc109131988)

[El inicio de un trabajo por lotes 5](#_Toc109131989)

[Terminación de procesos 5](#_Toc109131990)

[Salida normal (voluntaria) 5](#_Toc109131991)

[Salida por error (voluntaria) 6](#_Toc109131992)

[Error fatal (involuntaria) 6](#_Toc109131993)

[Eliminado por otro proceso (involuntaria) 6](#_Toc109131994)

[Jerarquías de procesos 6](#_Toc109131995)

[Estados de un proceso 6](#_Toc109131996)

[Implementación de Procesos 8](#_Toc109131997)

* Estados de un proceso. Implementación de los procesos. Estructuras de Datos Asociadas PCB. Cambio de contexto.
* Planificación de Procesos: criterios y objetivos buscados. Administración de la CPU. Conceptos asociados. Quantum o slice. Tiempos de espera. Tiempo de retorno.
* Políticas de planificación apropiativas y no apropiativas. FCFS, SJF, SRTF, Round-Robin, Prioridades, Colas Multinivel

# Procesos: Definición

El **concepto** central de cualquier Sistema Operativo es el de proceso: *una abstracción de un programa en ejecución también llamada tarea.*

No hay un acuerdo universal sobre una definición de proceso, pero sí algunas definiciones aceptadas:

• Un programa que se está ejecutando.

• Una actividad asincrónica.

• El emplazamiento del control de un procedimiento que está siendo ejecutado.

• Aquello que se manifiesta por la existencia en el Sistema Operativo de un bloque

de control de proceso.

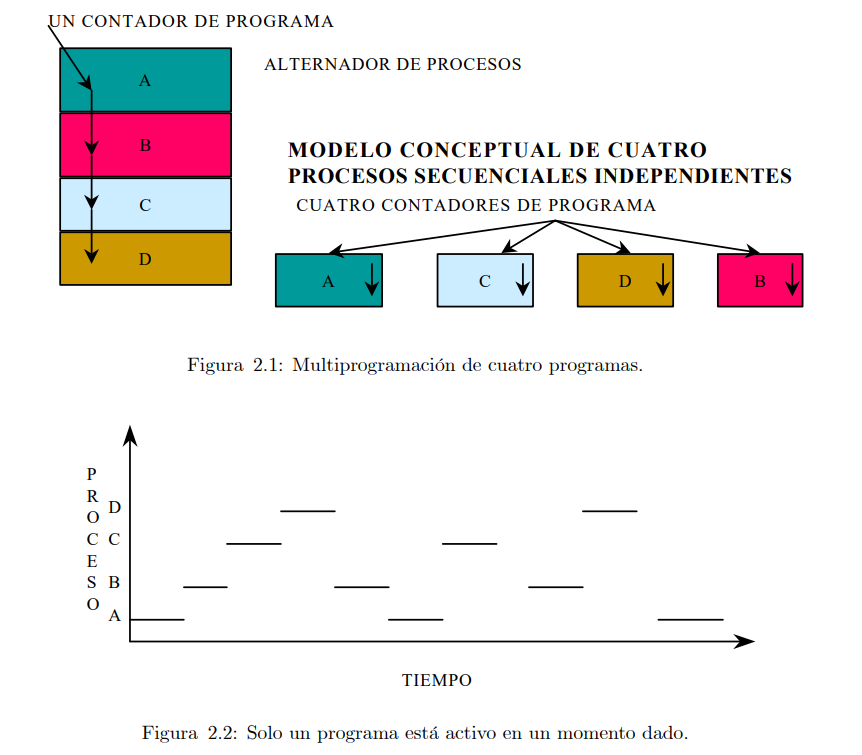
• Aquella entidad a la cual son asignados los procesadores.

• La unidad despachable.

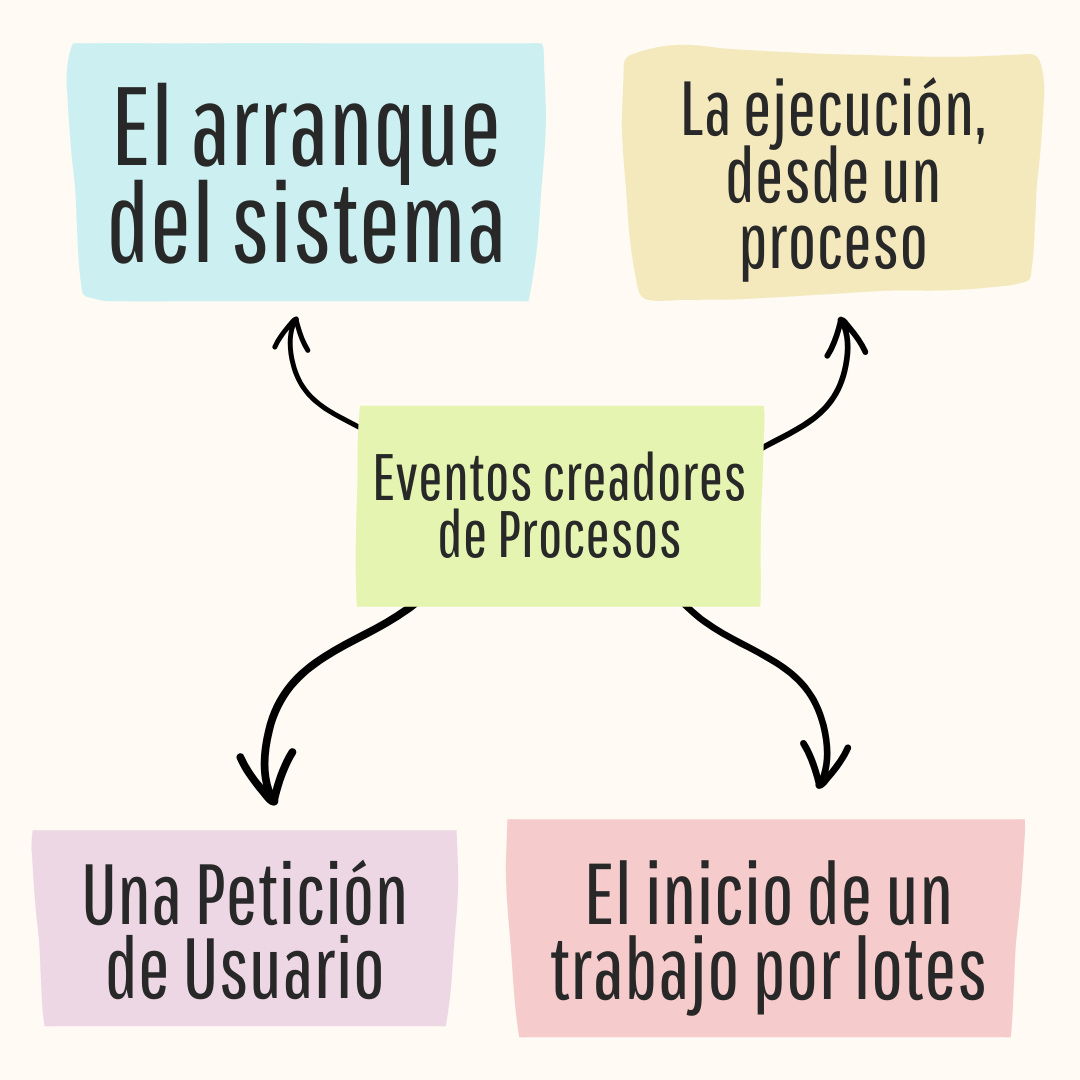
En sistemas de multiprogramación la cpu alterna de programa en programa, en un esquema de seudoparalelismo, es decir que la cpu ejecuta en cierto instante un solo programa, intercambiando muy rápidamente entre uno y otro.

# El modelo del proceso

El modelo de procesos posee las siguientes características:

* Todo el software ejecutable, inclusive el Sistema Operativo, se organiza en varios procesos secuenciales o procesos.
* Un proceso incluye al programa en ejecución y a los valores activos del contador, registros y variables del mismo.
* Conceptualmente cada proceso tiene su propia CPU virtual.
* Si la CPU alterna entre los procesos, la velocidad a la que ejecuta un proceso no será uniforme, por lo que es necesario aclarar lo siguiente:
  + Que los procesos no deben programarse con hipótesis implícitas acerca del tiempo.
  + Que normalmente la mayoría de los procesos no son afectados por la multiprogramación subyacente de la CPU o las velocidades relativas de procesos distintos.
* Un proceso es una actividad de un cierto tipo, que tiene un programa, entrada, salida y estado.
* Un solo procesador puede ser compartido entre varios procesos con cierto “algoritmo de planificación”, el cual determina cuándo detener el trabajo en un proceso y dar servicio a otro distinto

# Creación de un proceso

Hay cuatro eventos principales que provocan la creación de procesos:

## El arranque del sistema

Generalmente, cuando se arranca un sistema operativo se crean varios procesos. Algunos de ellos son procesos en primer plano; es decir, procesos que interactúan con los usuarios (humanos) y realizan trabajo para ellos. Otros son procesos en segundo plano, que no están asociados con usuarios específicos sino con una función específica.

Los procesos que permanecen en segundo plano para manejar ciertas actividades como correo electrónico, páginas Web, noticias, impresiones, etcétera, se conocen como demonios (daemons).

## La ejecución, desde un proceso

A menudo, un proceso en ejecución emitirá llamadas al sistema para crear uno o más procesos nuevos, para que le ayuden a realizar su trabajo. En especial, es útil crear procesos cuando el trabajo a realizar se puede formular fácilmente en términos de varios procesos interactivos relacionados entre sí, pero independientes en los demás aspectos.

En un multiprocesador, al permitir que cada proceso se ejecute en una CPU distinta también se puede hacer que el trabajo se realice con mayor rapidez.

## Una Petición de Usuario

En los sistemas interactivos, los usuarios pueden iniciar un programa escribiendo un comando o haciendo (doble) clic en un icono. Cualquiera de las dos acciones inicia un proceso y ejecuta el programa seleccionado.

En Microsoft Windows, cuando se inicia un proceso no tiene una ventana, pero puede crear una (o más) y la mayoría lo hace. Mediante el ratón, el usuario puede seleccionar una ventana e interactuar con el proceso, por ejemplo, para proveer datos cuando sea necesario.

## El inicio de un trabajo por lotes

La última situación en la que se crean los procesos se aplica sólo a los mainframes grandes (super PCs). Aquí los usuarios pueden enviar trabajos de procesamiento por lotes al sistema (posiblemente en forma remota). Cuando el sistema operativo decide que tiene los recursos para ejecutar otro trabajo, crea un proceso y ejecuta el siguiente trabajo de la cola de entrada.

Técnicamente, en todos estos casos, para crear un proceso hay que hacer que un proceso existente ejecute una llamada al sistema de creación de proceso. Esta llamada al sistema indica al sistema operativo que cree un proceso y le indica, directa o indirectamente, cuál programa debe ejecutarlo.

# Terminación de procesos

## Salida normal (voluntaria)

La mayoría de los procesos terminan debido a que han concluido su trabajo. Cuando un compilador ha compilado el programa que recibe, ejecuta una llamada al sistema para indicar al sistema operativo que ha terminado.

Los procesadores de palabras, navegadores de Internet y programas similares siempre tienen un icono o elemento de menú en el que el usuario puede hacer clic para indicar al proceso que elimine todos los archivos temporales que tenga abiertos y después termine.

## Salida por error (voluntaria)

E proceso descubre un error. Por ejemplo, si un usuario escribe el comando para compilar el programa foo.c y no existe dicho archivo, el compilador simplemente termina.

Los procesos interactivos orientados a pantalla por lo general no terminan cuando reciben parámetros incorrectos. En vez de ello, aparece un cuadro de diálogo y se le pide al usuario que intente de nuevo.

## Error fatal (involuntaria)

La tercera razón de terminación es un error fatal producido por el proceso, a menudo debido a un error en el programa. Algunos ejemplos incluyen el ejecutar una instrucción ilegal, hacer referencia a una parte de memoria no existente o la división entre cero.

## Eliminado por otro proceso (involuntaria)

Un proceso puede ejecutar una llamada al sistema que indique al sistema operativo que elimine otros procesos. El proceso eliminador debe tener la autorización necesaria para realizar la eliminación. En algunos sistemas, cuando un proceso termina (ya sea en forma voluntaria o forzosa) todos los procesos que creó se eliminan de inmediato también.

# Jerarquías de procesos

En algunos sistemas, cuando un proceso crea otro, el proceso padre y el proceso hijo continúan asociados en ciertas formas. El proceso hijo puede crear por sí mismo más procesos, formando una jerarquía de procesos. Observe que, a diferencia de las plantas y los animales que utilizan la reproducción sexual, un proceso sólo tiene un padre (pero cero, uno, dos o más hijos).

**Windows** no tiene un concepto de una jerarquía de procesos. Todos los procesos son iguales. La única sugerencia de una jerarquía de procesos es que, cuando se crea un proceso, el padre recibe un indicador especial un token (llamado manejador) que puede utilizar para controlar al hijo. Sin embargo, tiene la libertad de pasar este indicador a otros procesos, con lo cual invalida la jerarquía.

# Estados de un proceso

Durante su existencia un proceso pasa por una serie de estados discretos, siendo varias las circunstancias que pueden hacer que el mismo cambie de estado. Debido a ello se puede establecer una “Lista de Listos” para los procesos “listos” y una “Lista de Bloqueados” para los “bloqueados”.

Son 3 en total:

* **En ejecución** (en realidad está usando la CPU en ese instante).
* **Listo** (ejecutable; se detuvo temporalmente para dejar que se ejecute otro proceso).
* **Bloqueado** (no puede ejecutarse sino hasta que ocurra cierto evento externo).

La “Lista de Listos” se mantiene en orden prioritario y la “Lista de Bloqueados” está desordenada, ya que los procesos se desbloquean en el orden en que tienen lugar los eventos que están esperando. Al admitirse un trabajo en el sistema se crea un proceso equivalente y es insertado en la última parte de la “Lista de Listos”. La asignación de la cpu al primer proceso de la “Lista de Listos” se denomina “Despacho”, que es ejecutado por una entidad del Sistema Operativo llamada “Despachador”.

El “Bloqueo” es la única transición de estado iniciada por el propio proceso del usuario, puesto que las otras transiciones son iniciadas por entidades ajenas al proceso.

La manifestación de un proceso en un Sistema Operativo es un “Bloque de Control de Proceso” (PCB) con información que incluye:

* Estado actual del proceso.
* Identificación única del proceso.
* Prioridad del proceso.
* Apuntadores para localizar la memoria del proceso.
* Apuntadores para asignar recursos.
* Área para preservar registros.

Cuando el Sistema Operativo cambia la atención de la CPU entre los procesos, utiliza las áreas de preservación del PCB (Bloque de Control del Proceso) para mantener la información que necesita para reiniciar el proceso cuando consiga de nuevo la CPU.

Los sistemas que administran los procesos deben poder crear, destruir, suspender, reanudar, cambiar la prioridad, bloquear, despertar y despachar un proceso.



# Implementación de Procesos

Para implementar el modelo de procesos, el sistema operativo mantiene una tabla (un arreglo de estructuras) llamada **tabla de procesos**, con sólo una entrada por cada proceso (algunos autores llamana estas entradas bloques de control de procesos). Esta entrada contiene información importante acerca del estado del proceso, incluyendo su contador de programa, apuntador de pila, asignación de memoria, estado de sus archivos abiertos, información de contabilidad y planificación, y todo lo demás que debe guardarse acerca del proceso cuando éste cambia del estado en ejecución a listo o bloqueado, de manera que se pueda reiniciar posteriormente como si nunca se hubiera detenido.