

## Unidad 3: Procesos (parte 1)

### ¿Que es un proceso?

*Un proceso es un programa en ejecución. Cada proceso tiene asociado un espacio de direcciones, una lista de ubicaciones de memoria que va desde algún mínimo hasta cierto valor máximo, donde el proceso puede leer y escribir información. El espacio de direcciones contiene el programa ejecutable, los datos del programa y su pila. También hay asociado a cada proceso un conjunto de recursos, que comúnmente incluye registros (el contador de programa y el apuntador de pila), una lista de archivos abiertos, alarmas pendientes, listas de procesos relacionados y toda la demás información necesaria para ejecutar el programa. En esencia, un proceso es un recipiente que guarda toda la información necesaria para ejecutar un programa. Las llamadas al sistema de administración de procesos clave son las que se encargan de la creación y la terminación de los procesos.*

### ¿Que es una llamada al sistema?

Una llamada al sistema es un método utilizado por los procesos para comunicarse con el núcleo del sistema. En los sistemas operativos modernos, esto es necesario cuando un proceso necesita transmitir o leer información del hardware, de otros procesos o del propio kernel. De este modo, la llamada es el punto de enlace entre el usuario y el kernel.

Antes de que la llamada al sistema termine de procesarse y se transmitan o reciban los datos correspondientes, el kernel toma el control del programa o proceso. La ejecución se interrumpe durante este período. Una vez realizada la acción solicitada por una llamada al sistema, el núcleo renuncia al control y el código continúa desde el punto en el que se inició previamente la llamada al sistema.

Por ejemplo, un proceso llamado intérprete de comandos o shell lee comandos de una terminal. El usuario acaba de escribir un comando, solicitando la compilación de un programa. El shell debe entonces crear un

proceso para ejecutar el compilador. Cuando ese proceso ha terminado la compilación, ejecuta una llamada al sistema para terminarse a sí mismo.

Si un proceso puede crear uno o más procesos aparte (conocidos como procesos hijos) y estos procesos a su vez pueden crear procesos hijos, realizando una estructura de árbol de procesos. Los procesos relacionados que cooperan para realizar un cierto trabajo a menudo necesitan comunicarse entre sí y sincronizar sus actividades. A esta comunicación se le conoce como comunicación entre procesos.

Hay otras llamadas al sistema de procesos disponibles para solicitar más memoria (o liberar la memoria sin utilizar), esperar a que termine un proceso hijo y superponer su programa con uno distinto.

## Procesos en primer plano y segundo plano

En nuestro sistema operativo podemos tener procesos en primer plano (foreground) o en segundo plano (background).

Un proceso ejecutándose en primer plano lanzado desde un terminal monopoliza dicho terminal, por lo que en principio, no podremos ejecutar ningún otro proceso a la vez. Este hecho transportado al entorno de interfaz gráfica nos indica que cualquier proceso que requiera o utilice la intervención humana para su funcionamiento será considerado como proceso en primer plano.

Por el contrario un programa en segundo plano una vez iniciado, deja de bloquear el terminal desde el que se inició, y nos vuelve a dejar ingresar comandos. En el entorno de interfaz gráfica nosotros no tenemos una representación de los procesos en segundo plano y generalmente les llamamos a estos procesos como servicios o demonios. Al carecer de interacción humana, estos procesos se comunican con los usuarios a través de los registros de actividad (logs) para mostrar los errores que puedan tener en su ejecución. Puede consultar los servicios que posee su entorno, accediendo al administrador de tareas.

## Creación de un proceso

Hay cuatro métodos que provocan la creación de procesos:

1. El arranque del sistema.
2. La ejecución, desde un proceso, de una llamada al sistema para creación de procesos.
3. Una petición de usuario para crear un proceso.
4. El inicio de un trabajo por lotes.

En Windows una llamada a una función de Win32 (CreateProcess) maneja la creación de procesos y carga el programa correcto en el nuevo proceso. Esta llamada tiene 10 parámetros, que incluyen el programa a ejecutar, los parámetros de la línea de comandos para introducir datos a ese programa, varios atributos de seguridad, bits que controlan si los archivos abiertos se heredan, información de prioridad, una especificación de la ventana que se va a crear para el proceso (si se va a crear una) y un apuntador a una estructura en donde se devuelve al proceso que hizo la llamada la información acerca del proceso recién creado. Además de CreateProcess, Win32 tiene cerca de 100 funciones más para administrar y sincronizar procesos y temas relacionados.

## Cierre de un proceso

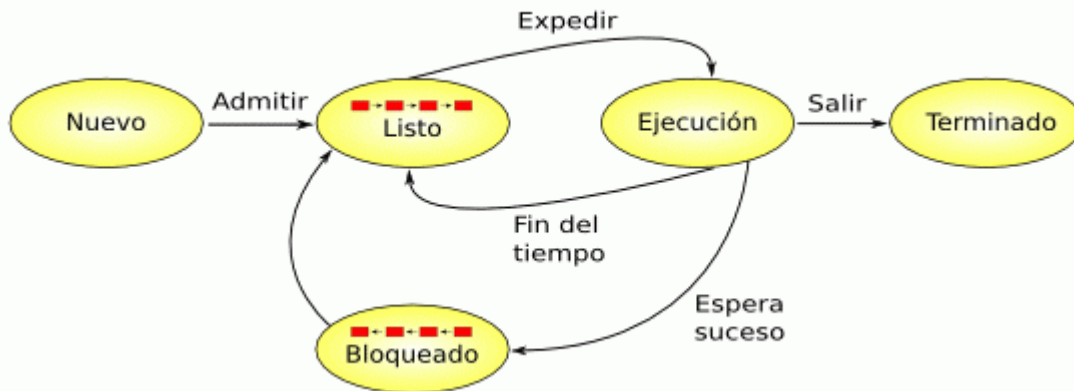
Una vez que se crea un proceso, empieza a ejecutarse y realiza el trabajo al que está destinado, este terminará debido a una de las siguientes condiciones:

1. Salida normal.
2. Salida por error.
3. Error fatal.
4. Eliminado por otro proceso.

En Windows una llamada a una función de Win32 (ExitProcess) maneja el cierre correcto de los procesos cuando estos envían su señal de cierre.

## Estados de un proceso

En esta etapa podemos definir cuál es el camino que sigue un proceso desde que se crea hasta que concluye su tarea asignada.



Como vemos en la imagen anterior, podemos identificar que tenemos el estado de **Nuevo** (el proceso ha iniciado y se le asignan sus características básicas entre ellas el espacio de direcciones que va a utilizar), el estado de **Listo** (en este estado el proceso se encuentra en una cola de espera para acceder a la CPU), el estado de **Ejecución** (el proceso se encuentra dentro de la CPU ejecutando las instrucciones que contiene, incrementando a su vez su contador de programa), el estado **Bloqueado** (en este punto, el proceso espera para continuar, que se libere algún dispositivo de E/S o que ocurra algún evento para continuar) y el estado **Terminado** (el proceso ya ha concluido sus tareas y está listo para liberar sus recursos)

## Implementación de los procesos

Para implementar el modelo de procesos, el sistema operativo mantiene una tabla llamada tabla de procesos, con sólo una entrada por cada proceso (PCB). 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.

## Bibliografía

Tanenbaum, A. S. (2003). *Sistemas operativos modernos*. Pearson Educación.