

PROCESOS —> Instancia de un programa en ejecución más su contexto (PCB)

- Es un concepto manejado por los SO
- El SO los crea y los destruye
- Es un mecanismo esencial para definir y gestionar la ejecución concurrente de los programas bajo el control de los SO
- Nacen mediante un SystemCall
- Contexto -> Conjunto de entidades que definen el estado de un proceso

Componentes del proceso:

- Instrucciones de programa destinadas a ser ejecutadas
- Estado de ejecución
- Memoria de Trabajo (memoria reservada + sus contenidos)
- Otra información para la planificación del SO

Definiciones:

- Espacio de nombres de un programa: Nombre de elementos que un programa puede utilizar (Ej: variables, archivos, etc.)
- Espacio de nombres de un proceso: Espacio en RAM donde están los objetos que pertenecen al procesador.
- Espacio de nombres del procesador: Espacio en memoria que pueden utilizar todos los procesos.
- Espacio de memoria: Direcciones de memoria donde se monta espacio de nombres del procesador.
- Poder de un proceso: Define los recursos que pueden ser accesibles por dicho proceso.

Bloque de Control de Procesos (PCB)

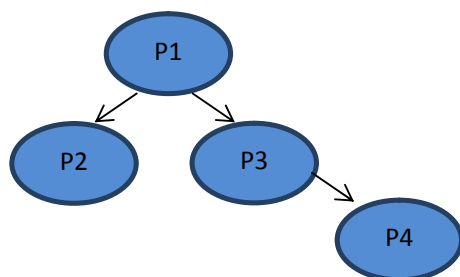
El Bloque de Control de Procesos contiene el contexto de un proceso y todos los datos necesarios para hacer posible la ejecución de ese proceso y satisfacer sus necesidades.

En forma más detallada cada uno de sus campos contiene (no están todos listados, solo los más importantes):

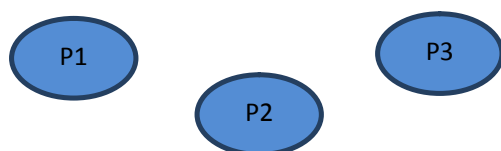
- **Apuntador a proceso anterior:** dirección del BCP anterior (anterior en tiempo pues fue creado antes). El primer BCP tendrá una identificación que lo señale como tal y deberá ser conocida su ubicación por el Planificador de Procesos.
- **Apuntador a proceso posterior:** dirección del BCP posterior (posterior en tiempo, pues fue creado después). El último BCP tendrá un nil (no se descartan encadenamientos circulares, pero por ahora se los presenta como lineales).
- **Identificación de Proceso (ID):** identificación única para este proceso que lo hace inconfundible con otro.
- **Dispositivos:** apuntador a todos los dispositivos a los que tiene acceso el proceso al momento. Esta información puede ser estática si es necesario que el proceso declare antes de comenzar su ejecución los dispositivos a usar, o completamente dinámica si existe la capacidad de obtener y liberar dispositivos a medida que se ejecuta el proceso.
- **Archivos:** ídem Dispositivos pero para los archivos del proceso.
- **Tiempos:** Tiempo de CPU utilizado hasta el momento. Tiempo máximo de CPU permitido a este proceso. Tiempo que le resta de CPU a este proceso. Otros tiempos.
- **Estado:** Ejecutando. Listo. Bloqueado. Wait (En espera). Ocioso.
- **Accounting:** información que servirá para contabilizar los gastos que produce este proceso (números contables, cantidad de procesos de E/S, etc.)

Creación de un proceso → Darle un nombre único y especificar en el PCB objetos y datos

→ **Jerárquica:** Cada proceso es hijo del creador y hereda el entorno de ejecución del padre.



→ **No Jerárquica:** Cada proceso es independiente de su creador (entornos diferentes).



Tipo de recursos de un proceso:

- Variables locales: Solo son accesibles por desde el proceso y solo estos la conocen.
- Variables Globales o Públicas: Son aquellas que también se acceden desde otros Procesos (Compartidas).

Tipos de Procesos:

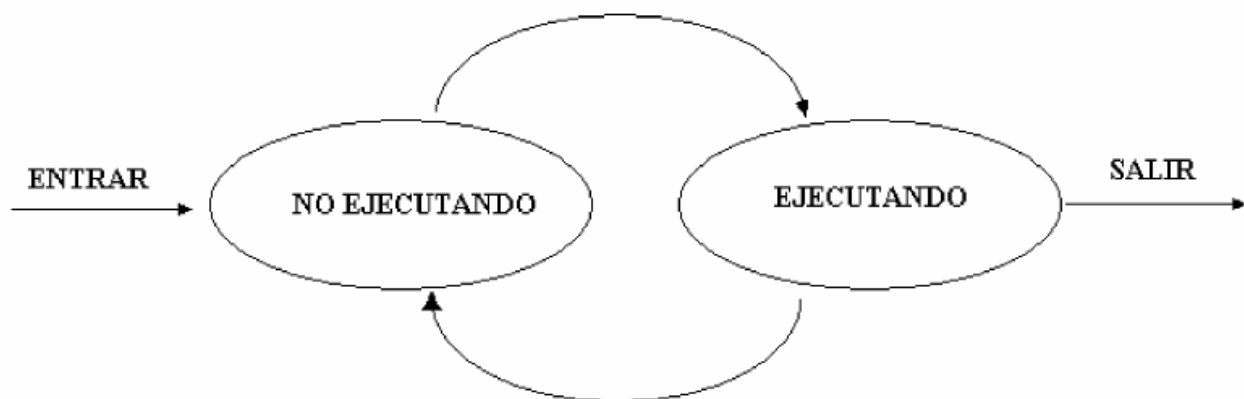
- Independientes o Disjuntos: Son aquellos que sólo tienen variables locales (operan sobre variables disjuntas) o comparten variables globales sin modificarlas.
- Paralelos o Concurrentes: Son aquellos que pueden usar simultáneamente un recurso. Si el recurso es modificado entonces es crítico y se usa la mutua exclusión para sincronizar su uso. También se denominan procesos concurrentes cuando sus ejecuciones se superponen en el tiempo (si la primera operación de uno de ellos es comenzada antes que se complete la última operación del otro).

Clasificación según su código:

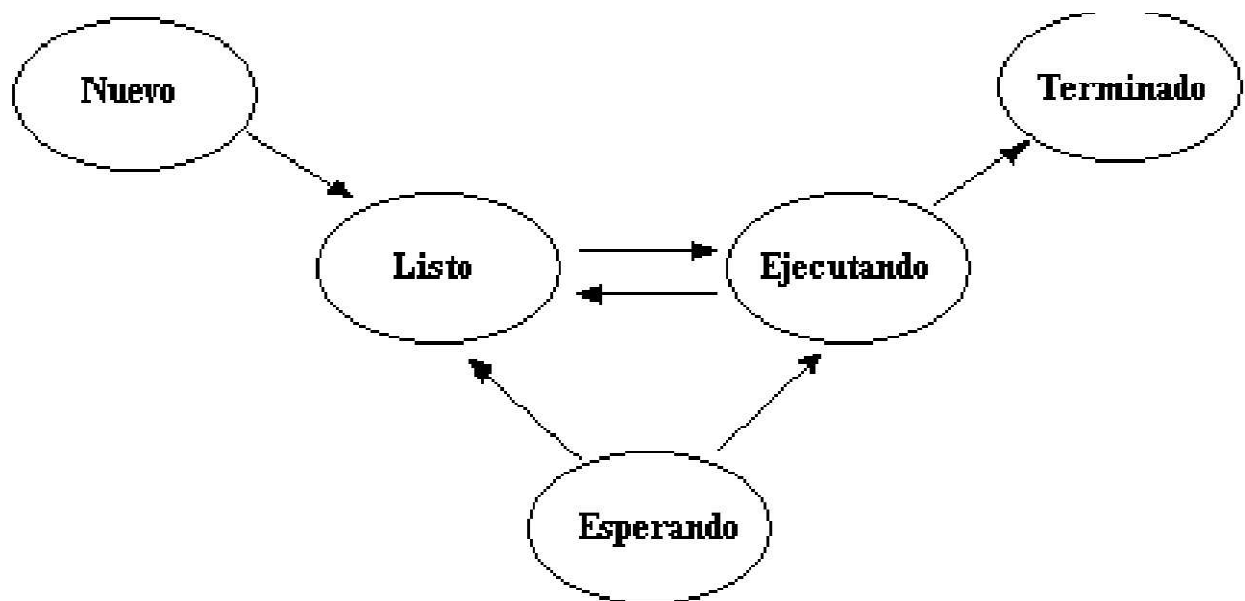
- Reutilizables: Son los que pueden ser ejecutados con diferentes juegos de datos.
- Reentrantes: Sólo tienen código puro y no tienen asociados datos.

Diagrama de transición o diagramas de estados

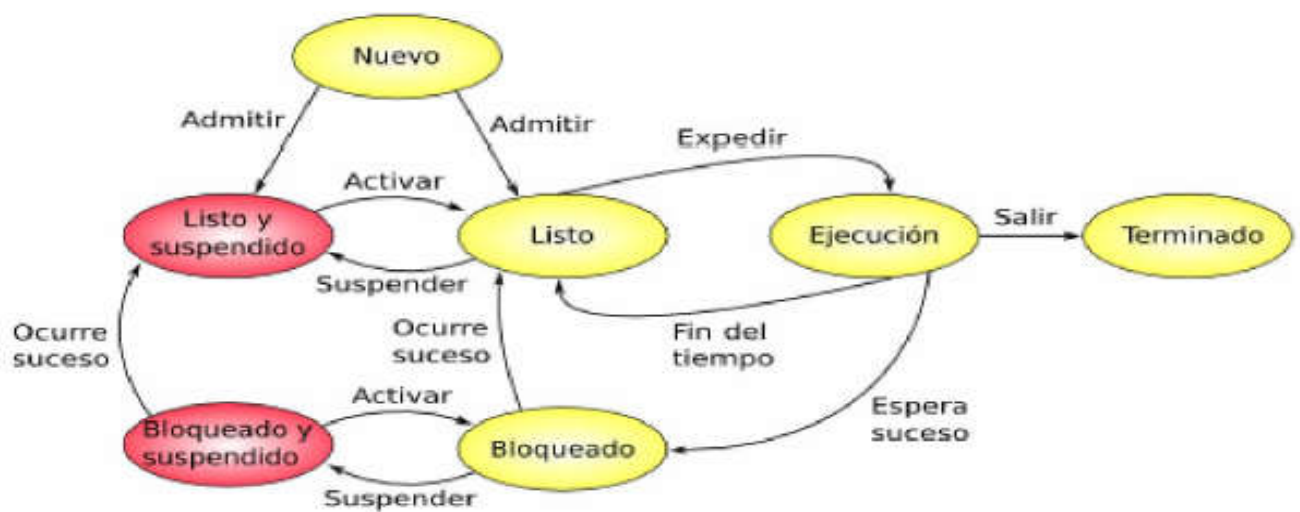
Modelo de dos Estados



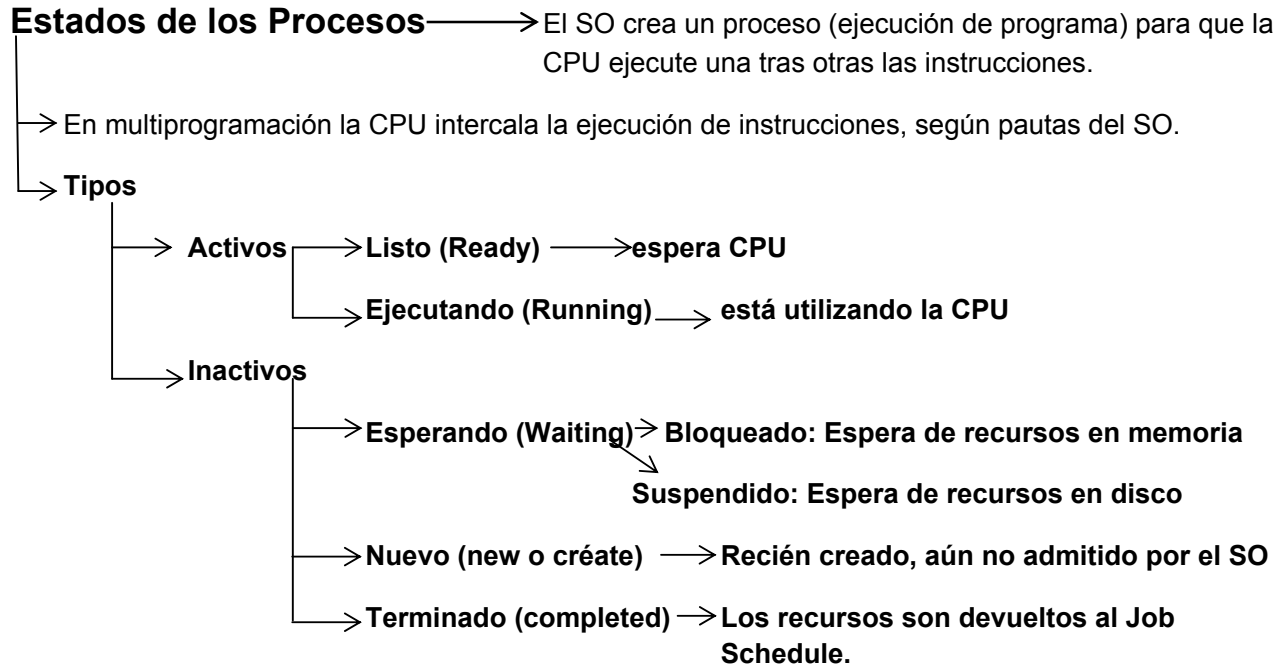
Modelo de cinco Estados



Modelo de cinco Estados más suspendido (utiliza SWAP a disco rígido)



Estados de los Procesos



Fin de un proceso

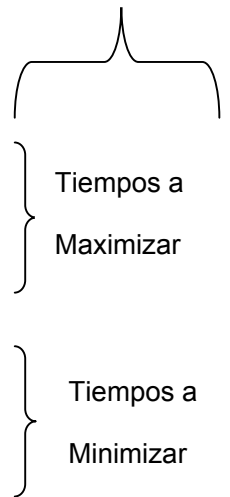
- **Normal:** Proceso completo
- **Anormal:** Ocurre un error y no puede finalizar de forma normal
- **Espera de recursos:** Pasa a bloqueado
- **Desalojo:** Un proceso se está ejecutando (running) y debe pasar al estado listo (ready) porque ingreso un proceso de mayor prioridad. Ej: estamos navegando por internet y un proceso del kernel requiere ser ejecutado en el momento.
- **Cuando un proceso muere desaparece su PCB liberando todos los recursos locales y comunes (estos son tomados por el Job Scheduler, capa 1 del SO). Cuando un proceso finaliza, también lo deben hacer sus hijos.**

POLITICAS DE ASIGNACION

Pautas:

- Uso del procesador al 100%
- Cantidad de trabajos por unidad de tiempo
- Tiempo de Ejecución: Tiempo desde que ingresa un proceso hasta que termina
- Tiempo de Espera: Permanencia en Listos
- Tiempo de Respuesta: Tiempo que tarda en obtenerse un resultado

Mejor
Algoritmo



1) FIFO (First In First Out)

El primer proceso en entrar es el primero en ser atendido. Se respeta el orden de llegada. No hay desalojo, un proceso solo pasa a bloqueado por decisión propia. Si entra a bloqueado debe volver al final de la cola de "listos".

2) Más corto primero (JSF) sin desalojo

Ordena los procesos en cola por el tiempo que van a utilizar el procesador.

3) Más corto primero (JSF) con desalojo

Una variante de la política de asignación de procesos anterior. El desalojo se puede dar por dos causas:

- Porque llegó un programa calificado como más corto que el que está en ejecución
- Porque llegó un programa cuyo tiempo es menor que el tiempo remanente.

4) Administración por prioridades

En la cola de ready los procesos con mayor prioridad van a tener mayor prioridad ante el submódulo Dispatcher.

5) Round Robin

Esta administración consiste en dar a cada proceso la misma cantidad o cuota de uso del procesador. Todos los procesos se les da un quantum, que es una medida del tiempo que

La asignación se comporta como una manecilla que recorre el segmento circular (que representa la ráfaga asignada) y que al pasar al próximo proceso genera una interrupción por reloj. Si el próximo proceso no se encuentra en estado de listo, se pasa al siguiente y así sucesivamente.



Representa el trabajo en conjunto de diferentes tipos de políticas de asignación.



HILO DE EJECUCIÓN (Threads)

Un hilo de ejecución, en sistemas operativos, es una característica que permite a una aplicación realizar varias tareas concurrentemente. Son sub-procesos que comparten el PCB del creador.

Es comparable con los procesos de estructura jerárquica pero a diferencia de los procesos no tienen un contexto independiente.

La principal diferencia entre un hilo y un proceso, es que estos últimos tienen un PCB propio. Los hilos comparten el PCB del proceso que los creo.

Por ejemplo, en un programa de hoja de cálculo un hilo puede estar visualizando los menús y leer la entrada del usuario mientras que otro hilo ejecuta las órdenes y actualiza la hoja de cálculo. Esta medida suele aumentar la velocidad que se percibe en la aplicación, permitiendo que el programa pida la orden siguiente antes de terminar la anterior.