

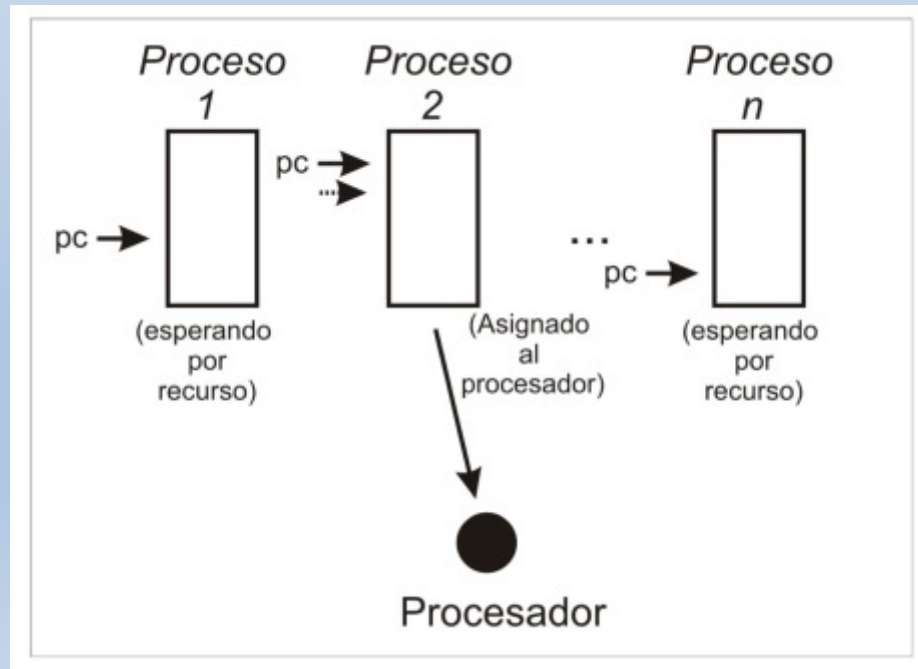
PROCESOS

PROCESO

1. El principal concepto en cualquier sistema operativo es el de **proceso**.
2. Un proceso es un programa en ejecución, incluyendo el valor del contador de programa (PC), los registros y las variables.
3. El recurso procesador es alternado entre los diferentes procesos que existan en el sistema, dando la idea de que ejecutan en paralelo (multiprogramación).

Metadatos de un proceso

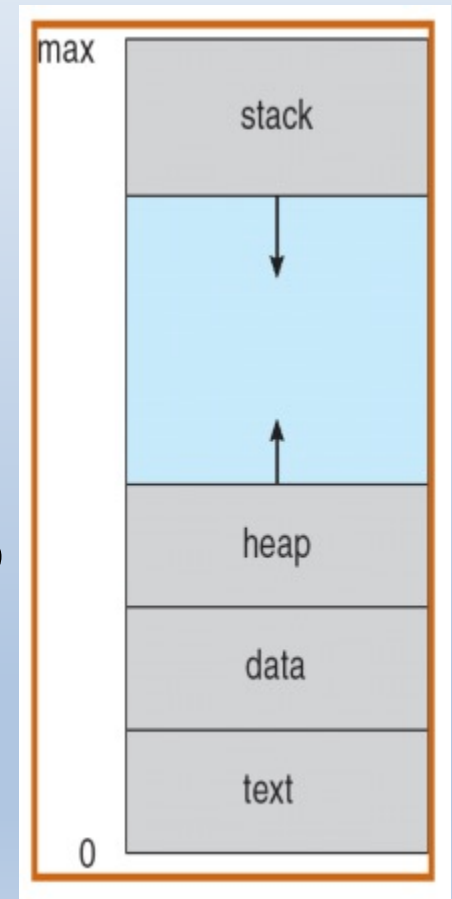
- Cada proceso tiene su Program Counter (**PC**), y avanza cuando el proceso tiene asignado el recurso procesador. A su vez, a cada proceso se le asigna un número que lo identifica entre los demás: identificador de proceso (process id - PID)



Estructura de un proceso

Un proceso en memoria se constituye de varias secciones:

- **Código** (text): Instrucciones del proceso.
- **Datos** (data): Variables globales del proceso.
- **Memoria dinámica** (heap): Memoria dinámica que genera el proceso.
- **Pila** (stack): Utilizado para preservar el estado en la invocación anidada de procedimientos y funciones.

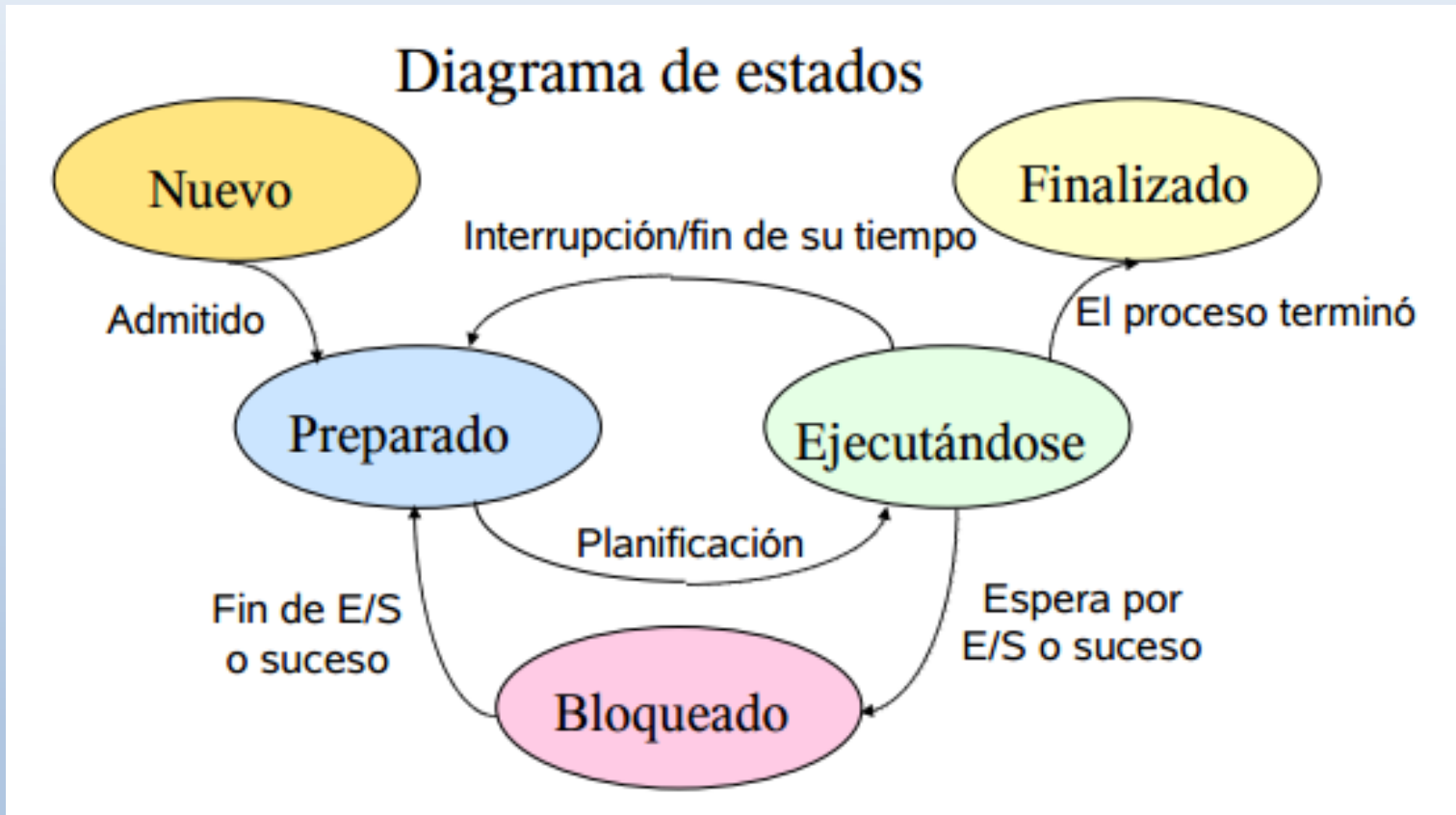


Estados de un proceso

Los estados de un proceso son:

- **Nuevo** (new): Cuando el proceso es creado.
- **Ejecutando** (running): El proceso tiene asignado un procesador y está ejecutando sus instrucciones.
- **Bloqueado** (waiting): El proceso está esperando por un evento (que se complete un pedido de E/S o una señal).
- **Listo** (ready): El proceso está listo para ejecutar, solo necesita del recurso procesador.
- **Finalizado** (terminated): El proceso finalizó su ejecución.

Estados de un proceso



Transiciones entre estados

- **Nuevo \Rightarrow Listo**

Al crearse un proceso pasa inmediatamente al estado listo.

- **Listo \Rightarrow Ejecutando**

En el estado de listo, el proceso solo espera para que se le asigne un procesador para ejecutar (tener en cuenta que puede existir más de un procesador en el sistema). Al liberarse un procesador el planificador (scheduler) selecciona el próximo proceso, según algún criterio definido, a ejecutar.

Transiciones entre estados

- **Ejecutando \Rightarrow Listo**

Ante una interrupción que se genere, el proceso puede perder el recurso procesador y pasar al estado de listo. El planificador será el encargado de seleccionar el próximo proceso a ejecutar.

- **Ejecutando \Rightarrow Bloqueado**

A medida que el proceso ejecuta instrucciones realiza pedidos en distintos componentes (ej.: genera un pedido de E/S). Teniendo en cuenta que el pedido puede demorar y, además, si está en un sistema multiprogramado, el proceso es puesto en una cola de espera hasta que se complete su pedido. De esta forma, se logra utilizar en forma más eficiente el procesador.

Transiciones entre estados

- **Bloqueado \Rightarrow Listo**

Una vez que ocurre el evento que el proceso estaba esperando en la cola de espera, el proceso es puesto nuevamente en la cola de procesos listos.

- **Ejecutando \Rightarrow Terminado**

Cuando el proceso ejecuta su última instrucción pasa al estado terminado. El sistema libera las estructuras que representan al proceso.

PLANIFICACIÓN DE PROCESOS

Un **proceso** es un programa en ejecución. Existen 3 estados en los que puede encontrarse un proceso, estos son: "Listo", "Bloqueado" y "En ejecución".

Para el control de los mismos internamente son almacenados en una lista, cada uno de los nodos guarda información de un proceso.

Los sistemas operativos cuentan con un componente llamado **planificador**, que se encarga de decidir cuál de los procesos hará uso del procesador. La toma de esta decisión, así como el tiempo de ejecución del proceso, estará dada por un algoritmo, denominado **Algoritmo de Planificación**.

Objetivos de la Planificación de procesos

La Planificación de procesos tiene como principales objetivos la equidad, la eficacia, el tiempo de respuesta, el tiempo de regreso y el rendimiento.

- **Equidad:** Todos los procesos deben ser atendidos.
- **Eficacia:** El procesador debe estar ocupado el 100% del tiempo. (CPU Utilization)
- **Tiempo de respuesta:** Tiempo desde que un proceso se carga hasta que da su primera respuesta (Response time).
- **Tiempo de regreso:** Tiempo desde que un proceso se carga hasta que finaliza su ejecución (Turnaround time).
- **Rendimiento:** Maximizar el número de tareas que se procesan por cada hora/segundo (Throughput), por unidad de tiempo.
- **Espera:** es la suma de los tiempos que un proceso estuvo en la cola de procesos listos (Waiting time).

CPU Bursts

Bursts: ráfagas de CPU.

Los procesos suelen tener ráfagas de CPU (CPU-burst) y ráfagas de espera asociados a operaciones de Entrada/Salida (I/O burst).

¿Quién elige y asigna los procesos a ejecutar?

El despachador es el módulo del SO que da el control de la CPU al proceso seleccionado por el scheduler (planificador). Esto implica:

- Cambio de contexto: salvar los registros de la cpu en el PCB del proceso saliente, cargar los registros con los datos del PCB entrante.
- Saltar a la instrucción adecuada en la que había quedado el proceso, restaurar el PC (Program counter).

Apropiativa o no apropiativa.

La **planificación no apropiativa** (no preemptive) es aquélla en la que, cuando a un proceso le toca su turno de ejecución, es decir, no se le puede arrebatarse el uso de la CPU, hasta que el proceso no lo determina no se podrá ejecutar otro proceso. Este esquema **tiene sus problemas**, puesto que si el proceso contiene **ciclos infinitos**, el resto de los procesos pueden quedar aplazados indefinidamente. Otro caso puede ser el de los procesos largos que penalizarían a los cortos si entran en primer lugar.

La **planificación apropiativa** (preemptive) supone que el sistema operativo **puede arrebatarse el uso de la CPU** a un proceso que esté ejecutándose. En la planificación apropiativa existe un reloj que lanza interrupciones periódicas en las cuales el planificador toma el control y se decide si el mismo proceso seguirá ejecutándose o se le da su turno a otro proceso.

Algoritmos de Planificación

○ **Primero en llegar primero en ser servido**

Conocido como FCFS (**First Come First Served**) FIFO. Este algoritmo emplea una cola de procesos, asignando un lugar a cada proceso por el orden de llegada. Cuando el proceso llega es puesto en su lugar en la cola después del que llegó antes que él y se pone en estado de listo. Cuando un proceso comienza a ejecutarse no se interrumpe su ejecución hasta que termina de hacerlo.

○ **Prioridad al más corto**

Su nombre es SJF (**Shortest Job First**). El proceso que se encuentra en ejecución cambiará de estado voluntariamente, o sea, no tendrá un tiempo de ejecución determinado para el proceso. A cada proceso se le asigna el tiempo que usará cuando vuelva a estar en ejecución, y se irá ejecutando el que tenga un menor tiempo asignado. Si se da el caso de que dos procesos tengan igual valor en ese aspecto emplea el algoritmo FCFS.

Algoritmos de Planificación

○ Round Robin

A cada proceso se le asigna un tiempo determinado para su ejecución, el mismo tiempo para todos. En caso de que un proceso no pueda ser ejecutado completamente en ese tiempo se continuará su ejecución después de que todos los procesos restantes sean ejecutados durante el tiempo establecido. Este es un algoritmo basado en FCFS que trata la cola de procesos que se encuentran en estado de listos como una cola circular.

○ Planificación por prioridad

En este tipo de planificación a cada proceso se le asigna una prioridad siguiendo un criterio determinado, y de acuerdo con esa prioridad será el orden en que se atiende cada proceso.

Algoritmos de Planificación

○ Planificación garantizada

Para realizar esta planificación el sistema tiene en cuenta el número de usuarios que deben ser atendidos. Para un número "n" de usuarios se asignará a cada uno un tiempo de ejecución igual a $1/n$.

○ Planificación de Colas Múltiples

El nombre se deriva de MQS (**Multilevel Queue Schedulling**). En este algoritmo la cola de procesos que se encuentran en estado de listos es dividida en un número determinado de colas más pequeñas. Los procesos son clasificados mediante un criterio para determinar en qué cola será colocado cada uno cuando quede en estado de listo. Cada cola puede manejar un algoritmo de planificación diferente a las demás.

Algoritmos de Planificación

Ejemplos de planificación de procesos y planificador online:

<http://cpuburst.com/ganttcharts.html>