

FUNCIONES DE UN SISTEMA OPERATIVO



FUNCIONES

Las funciones principales de un sistema operativo son

- Gestión de procesos
- Gestión de memoria
- Gestión de Entrada / Salida
- Gestión de archivos
- Gestión de la seguridad.

GESTIÓN DE PROCESOS

- Un proceso es un programa que está en ejecución. Cada vez que se manda ejecutar un programa se crearía un proceso. El sistema operativo debe realizar una gestión adecuada de los recursos del sistema para la correcta ejecución de los procesos.

PROCESOS

- Cada vez que se ejecuta un programa se crea una estructura de datos llamada **Bloque de control de proceso (BCP)** o **Process Control Block (PCB)**. El sistema operativo identifica a cada proceso con esta estructura, que lo diferencia de cualquier otro proceso y que puede contener, dependiendo del sistema operativo del que se trate, la siguiente información: PID (Process Identificador) número entero único y diferente para cada proceso, estado en que se encuentra, prioridad, recursos asignados, valores de los registros del procesador, propietario, permisos, etcétera.

Procesos

- Dependiendo de si el proceso lo ha lanzado un usuario o es un proceso del sistema operativo, podemos distinguir entre:
- **Procesos del sistema.** Son procesos que generalmente se ejecutan al iniciar el sistema y suelen estar en ejecución para proporcionar servicios a los usuarios del sistema
- **Procesos de usuarios.** Son los procesos que manda ejecutar el usuario que entre en el sistema. Si el sistema es multiusuario, puede haber varios procesos de distintos usuarios ejecutándose a la vez.

Servicios del Sistema

- Los servicios de un sistema operativo son procesos que proporcionan una determinada función a los usuarios.
- Cualquier sistema operativo, para empezar a funcionar y para terminar, necesita los servicios de arranque y parada. Actualmente son procesos sencillos en los que unos ficheros de configuración controlan el inicio y arranque del sistema.

Servicios del Sistema

- **Arranque.** Mediante el arranque se prepara el sistema para que pueda ser explotado y utilizado por los usuarios.
- **Parada.** El servicio de parada debe dejar al sistema de manera que se pueda volver a arrancar sin problemas.

Además de estos servicios, un sistema operativo actual proporciona muchos más servicios

ALGORITMOS DE PLANIFICACIÓN DE PROCESOS

- En sistemas operativos multitarea los procesos aparentemente se ejecutan a la vez, pero si solo hay un único procesador, este se tiene que distribuir entre todos los procesos que se estén ejecutando.
- A la forma en que la CPU se distribuye para ejecutar los procesos se le llama planificación. Existen distintos algoritmos de planificación.

ALGORITMOS DE PLANIFICACIÓN DE PROCESOS

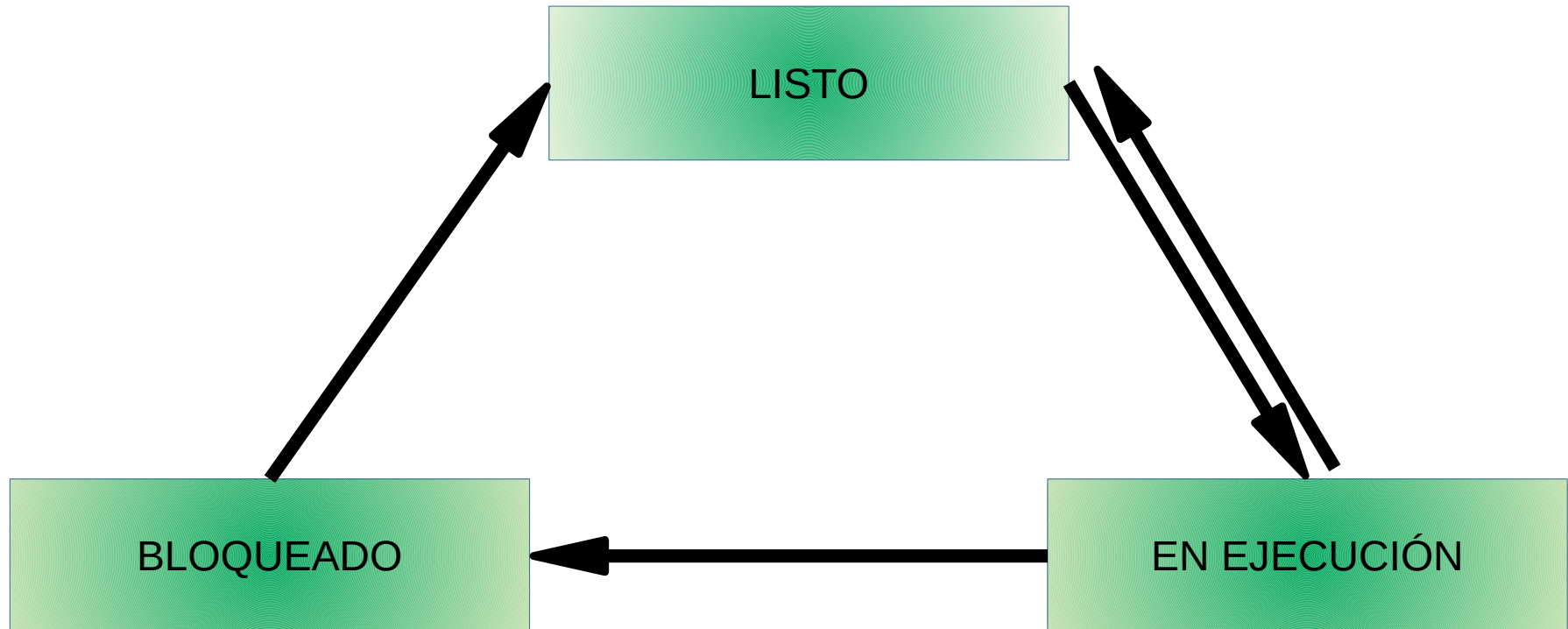
- El **planificador** a corto plazo (dispatcher) decide que proceso de los que están preparados para ejecutarse pasa a utilizar el procesador. El planificador debe intentar minimizar el tiempo de respuesta, maximizar la cantidad de procesos que se ejecuten y evitar que un proceso quede postergado indefinidamente. Debe intentar tratar a los procesos por igual salvo en caso de que haya procesos que tengan mayor prioridad, aunque también debe evitar que procesos muy prioritarios pospongan indefinidamente procesos menos prioritarios, mediante técnicas como el **envejecimiento** (aging).

ESTADOS DE LOS PROCESOS

- **Listos**, en espera o preparados: el proceso está preparado para ejecutarse, es decir, está en espera de que el proceso que se está ejecutando deje libre la CPU.
- **Bloqueados**: el proceso está esperando un recurso que está siendo utilizado por otro proceso en ese momento.
- **En ejecución**: el proceso está ejecutando sus instrucciones en ese momento, es decir, está ocupando la CPU.

ESTADOS DE UN PROCESO

Estados en los que puede estar un proceso y sus transiciones



TRANSICIONES DE ESTADO DE LOS PROCESOS

- Cuando un proceso pasa de un estado a otro, se produce una transición de estado. Podemos tener cuatro transiciones.
 - De ejecución a bloqueo
 - De ejecución a listo
 - De listo a en ejecución
 - De bloqueado a listo

TRANSICIONES DE ESTADO DE LOS PROCESOS

De ejecución a bloqueo

- Al iniciar una operación de Entrada / Salida

De ejecución a listo

- Por ejemplo, en un sistema que utilice el tiempo compartido, cuando el proceso que ocupa al CPU lleva demasiado tiempo ejecutándose continuamente, el sistema operativo decide que otro proceso ocupe la CPU, pasando el proceso que ocupaba la CPU a estado listo.

TRANSICIONES DE ESTADO DE LOS PROCESOS

De listo a en ejecución

- Cuando lo requiere el planificador de la CPU

De bloqueado a listo

- Se dispone del recurso por el que se había bloqueado el proceso. Por ejemplo, termina la racionan de E/S

De las cuatro transiciones de estado posibles, la única iniciada por el proceso de usuario es el bloqueo, las otras tres son iniciadas por entidades externas al proceso.

TIPOS PRINCIPALES DE PLANIFICACIONES

- **Expulsiva**
 - Un proceso puede ser desalojado de la CPU sin que haya finalizado, y se queda en la cola de procesos en espera.
- **No Expulsiva**
 - Un proceso no puede ser desalojado de la CPU hasta que termina o se bloquea porque necesita un recurso no disponible.

A la operación de desalojar un proceso de la CPU para que otro empiece a ejecutarse, se le llama cambio de contexto. Esta operación debe realizarse con la mayor rapidez posible y en ella se debe almacenar la información sobre el estado de la ejecución del proceso, para que cuando vuelva a estar en ejecución, se reinicie desde el mismo punto y con los mismos valores en que se desalojó.

- Existen una serie de algoritmos de planificación que le indican al planificador qué proceso se debe elegir para pasar a ejecutarse de los que están en espera. Estos algoritmos deben maximizar la utilización de la CPU, evitando que esté libre en algún momento, y minimizar el tiempo de respuesta de cada proceso.

En cada algoritmo sabremos para cada proceso:

Tiempo de entrada o de llegada al sistema (T), es el momento en que el proceso entra en el sistema.

Tiempo de ejecución (T_x), es el tiempo que el proceso necesita para su ejecución total.

Nos interesa obtener para cada proceso los siguientes datos:

- **Tiempo de respuesta o de retorno (T_R)**, es el tiempo que pasa desde que el proceso llega al sistema hasta que se obtienen los resultados.
- **Tiempo de espera (T_E)**, es el tiempo que el proceso pasa dentro del sistema en espera

$$T_E = T_R - T_X$$

Además también queremos saber los tiempos medios tanto de respuesta como de espera, para poder saber si para ciertos datos de entrada el algoritmo es más o menos óptimo, en general.