MEDAC

Instituto Oficial de Formación Profesional

TEMA 2. PROCESOS

PROGRAMACIÓN DE
SERVICIOS Y PROCESOS
2º DAM

Curso 2021-2022

TEMA 2. PROCESOS
APARTADO 1. OBJETIVOS

- Conocer el concepto de proceso.
- Conocer e identificar todos los estados por los que puede pasar un proceso.
- Conocer diferentes planificadores de procesos.
- Conocer el concepto de cambio de proceso.

- Proceso ➡ Archivo que está en ejecución y bajo control del SO, se crea cuando ejecutamos un programa.

Proceso = programa en ejecución

⚠ Un proceso es una identidad independiente de todo lo demás, aunque se ejecute en un mismo programa.

Independientemente del SO que estemos utilizando, todos son multitarea, lo que nos permite ejecutar varios procesos a la vez.

De esta forma vamos a poder ejecutar las tareas unas tras otras o podremos realizar poco a poco, cada tarea, las tendremos completadas todas al final, pero las podremos realizar de una forma más rápida. Esto es gracias a la gestión de procesos que nos proporciona el propio SO.

Dentro de los procesos, vamos a distinguir según su **modo de ejecución** los siguientes tipos:

- Procesos por lotes: son procesos formados por una serie de tareas que se van a realizar. Al usuario que las ejecuta sólo le interesa el resultado final y no en su ejecución.
  Ejemplo: pasar el antivirus al ordenador.

- Procesos interactivos: en estos procesos hay una interacción del usuario y del propio proceso, que puede pedir al usuario datos para su ejecución.
  Ejemplo: Un procesador de textos.

- Procesos en tiempo real: estos se basan en ciertas tareas en las que el tiempo de respuesta por parte del sistema es crucial.
  Ejemplo: Brazo mecánico en una operación o el sistema de conducción automática de un coche.

Si nos fijamos al origen de la ejecución, distinguimos dos tipos de procesos:

- Procesos en modo kernel: estos procesos los ejecuta el propio núcleo del SO y que el usuario no puede controlar. Realizan las tareas más delicadas, como la gestión de memoria, la gestión de tráfico de red, etc.

- Procesos en modo usuario: son procesos lanzados por el propio usuario, por lo que podemos decir que son los procesos más “normales” o usuales.

  Ejemplo: Un procesador de textos.

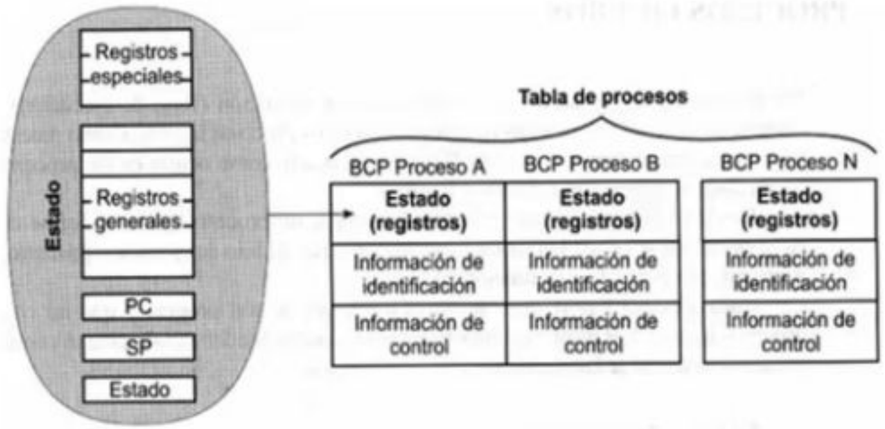
Partes de un proceso

Los procesos están formados por una estructura llamada PCB (Bloque de Control de Proceso). Toda la información que se guarda en el PCB es única y nos permitirá controlar el proceso.

Los PCB están formados por:

- PID o identificador de proceso: es un número único para cada proceso que nos permitirá identificarlo de forma única.
- Estado actual: es el estado actual en el que se encuentra el proceso, puede ser listo, bloqueado, ejecutándose...
- Espacio de direcciones en memoria: esto nos indica dónde comienza la zona de memoria que está reservada para cada proceso.

- Información importante para la planificación: aquí encontraremos toda la información que nos permitirá planificar el proceso, como puede ser el tiempo de quantum, la prioridad,etc.
- Información de cambio de contexto: aquí encontramos toda la información que tendremos que guardar o recuperar cuando ejecutemos un cambio de contexto, valor de los registros de la CPU, contador del programa,etc.
- Recursos utilizados por el proceso: aquí vamos a tener todos los recursosque utiliza el proceso, como puede ser tráfico de red, ficheros,etc.



El **planificador** es el encargado de crear y poner en ejecución a los procesos en el SO, y el **usuario** de intervenir de forma directa.

Existe el ciclo de vida de un proceso para que los procesos funcionen correctamente.

Los procesos cuando se ejecutan pueden atravesar distintos estados desde que se inician hasta que finalizan, teniendo que cumplir ciertos requisitos para pasar de un estado a otro.

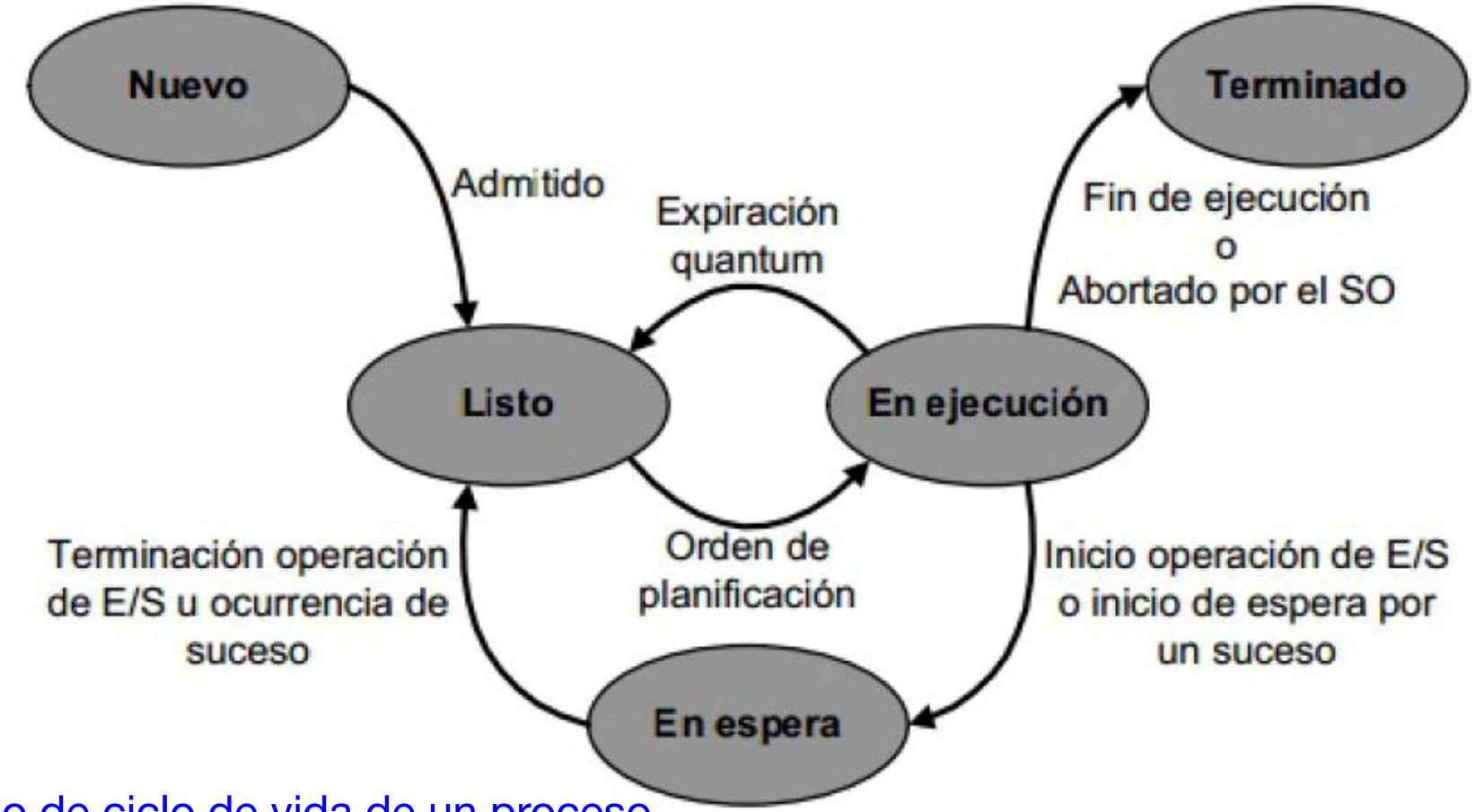
Los estados por los que pasa un proceso son:

- **Nuevo:** el proceso se crea a partir de un fichero ejecutable, aunque todavía no lo hayan admitido en el grupo de procesos ejecutables por el SO.
- **Listo:** ya se ha creado el proceso pero aún no está listo para ejecutarse, porque el planificador del SO no lo ha seleccionado para entrar a ejecución.

- **En ejecución:** siguiendo un algoritmo de planificación, el planificador del SO, elige al proceso para que se ejecute. El proceso se ejecutará de forma continuada hasta cumplir su tiempo máximo de ejecución o hasta que el planificador lo saque de este estado, pasará de “Nuevo” a “Listo”. Si el proceso necesita algún recurso, lo puede pedir mediante interrupciones.

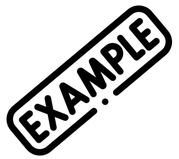
- **Terminado:** el proceso finaliza su ejecución y libera todos los recursos que estaba acaparando, esperando que el SO lo destruya.

DIAGRAMA DE ESTADOS DE UN PROCESO



Vídeo - Ejemplo de ciclo de vida de un proceso

El SO es el encargado principal de toda la gestión de los procesos, que normalmente sigue las órdenes del usuario.



Cuando un usuario da la orden de abrir un programa, realmente es el SO el encargado de crear y poner en ejecución el proceso que corresponde al programa en sí.

Cuando el procesador pasa a **ejecutar un nuevo proceso**, quien debe guardar todo el contexto que tiene el proceso actual y quien debe restaurar el contexto que tenía el proceso que el planificador ha decidido ejecutar es el SO.

Es también el SO el responsable de controlar toda la **comunicación y sincronización** entre los procesos, además de su eliminación y la terminación

Como ya sabemos los SO actuales nos permiten trabajar con multiprogramación, por lo que admiten la ejecución de varios procesos en memoria para maximizar el uso del procesador. Todo esto se debe a que los procesos se van intercambiando con los que están en ejecución, para que finalmente todos usen el procesador de igual forma, consiguiendo una ejecución concurrente de los procesos.

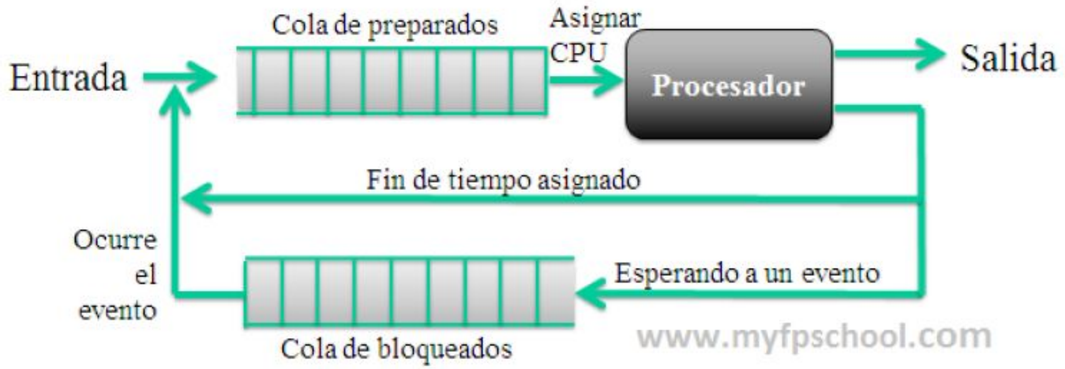
Para que esto ocurra, el SO organiza los procesos en varias colas:

- Una cola de procesos que van a contener todos los procesos que hay en el sistema.

- Una cola de procesos preparados, que tendrá todos los procesos en estado “Listo” los cuales esperan a que el planificador los seleccione para ejecutarse.
- Varias colas que contienen los procesos que están en estado “Bloqueado” y que esperan alguna operación de E/S. Tendremos una cola de estas por cada dispositivo que permita una operación de E/S.

Contamos con 4 eventos que van a provocar la creación de un proceso:

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



Necesitamos un planificador de procesos para gestionar todas las colas de procesos que tenemos, este se va a encargar de seleccionar los procesos que van a ejecutarse en el procesador, imponiendo un orden de ejecución entre los procesos de colas.

Principalmente, existen dos tipos de planificación posibles:

- Planificación de procesos a corto plazo.
- Planificación de procesos a largo plazo.

PLANIFICACIÓN DE PROCESOS A CORTO PLAZO

Estos planificadores cogen el proceso de la cola de procesos *preparados* que va a pasar a *ejecución*. Normalmente se invocan frecuentemente y tardan milisegundos, que es cuando se produce un cambio de estado del proceso en ejecución, por lo que tienen que ser muy rápidos en la toma de decisiones.

Esto conlleva a que los algoritmos de este tipo de planificadores sean muy sencillos, por ejemplo:

- Planificación sin desalojo o cooperativa: en esta planificación solo se cambia el proceso en ejecución si se ha bloqueado o terminado.
- Planificación apropiativa: aparte de los casos que cubre la planificación cooperativa, este planificador cambiará el proceso en ejecución si en cualquier momento otro proceso con mayor prioridad puede ejecutarse.
- Tiempo compartido: estos planificadores, cada cierto tiempo (quantum), quitan el proceso que estaba en ejecución y seleccionan otro proceso para que se ejecute. En este caso, todas las prioridades son consideradas iguales.

PLANIFICACIÓN DE PROCESOS A LARGO PLAZO

Estos planificadores no son los encargados de crear las estructuras que componen los procesos, sólo se encargará de decidir, a través de un algoritmo, qué proceso se puede ejecutar en qué momento. Las estructuras de los procesos solo se crean una vez, que es cuando el proceso se crea.

Los planificadores también son conocidos como → SCHEDULER

¿Quién se encarga de planificar los procesos que se van a ejecutar? ➤ El planificador de procesos.

El planificador de procesos debe asegurar el máximo aprovechamiento del sistema, por lo que debe dar servicio a todos los procesos que estén en ejecución en un momento dado y no dejar a ninguno de ellos fuera sin ejecutarse.

Para ello el planificador puede usar distintos algoritmos de planificación:

- **Primero en llegar (FIFO):** es uno de los planificadores más simples y hace que se vayan ejecutando los procesos según se creen, sin ningún otro criterio.
- **Prioridad al más corto:** este planificador ordena los procesos según el tiempo de ejecución que necesiten e irá pasando a ejecución priorizando el que menos tiempo necesite.

- **Prioridad al más largo:** este planificador ordena los procesos según el tiempo de ejecución que necesiten e irá pasando a ejecución priorizando el que más tiempo necesite.

En las dos planificaciones anteriores se produce lo que se conoce como inanición, ya que si hay un proceso que necesita mucho tiempo de ejecución y siempre entran los que necesiten menos, nunca se ejecutará. Y lo mismo pasa en el caso contrario.

- **Round Robin:** este planificador tiene un tiempo de ejecución llamado quantum, que es el tiempo que dejará de ejecutarse a cada proceso, sacándolos de ejecución una vez pasado el quantum de tiempo. De esta manera todos los procesos se ejecutarán poco a poco.
- **Planificación por prioridad:** esta planificación utiliza la prioridad de ejecución que tienen los procesos, ejecutando antes el más prioritario. También puede producir inanición a procesos con poca prioridad.
- **Planificación de colas múltiples:** este tipo de planificador es una mezcla de todos los anteriores, su objetivo es diferenciar entre distintos tipos de trabajos, para ello dividen la cola de procesos preparados en varias colas, una por cada tipo de trabajo, y no permiten el movimiento de los procesos entre las distintas colas. Cada cola puede usar su propia política.

Actividad 1: Algoritmo Round robin

Proceso	Tiempo de llegada	Ciclos de CPU
P1	1	4
P2	1	3
P3	2	7
P4	1	1

Quantum= 2

Actividad 2 Algoritmo Round robin

Proceso	Tiempo de llegada	Ciclos de CPU
A	5	6
B	2	4
C	0	3
D	3	7

Quantum= 2

Actividad 3 Algoritmo Round robin

Proceso	Tiempo de llegada	Ciclos de CPU
A	1	8
B	3	5
C	4	9
D	7	4
E	8	1

Quantum= 3

Actividad 4 Algoritmo Round robin

Proceso	Tiempo de llegada	Ciclos de CPU
P1	0	5
P2	4	2
P3	1	7
P4	2	1
P5	3	8

Quantum= 3

Actividad 5 Algoritmo Round robin

Proceso	Tiempo de llegada	Ciclos de CPU
A	0	3
B	2	6
C	4	4
D	7	5
E	8	2

Quantum= 1

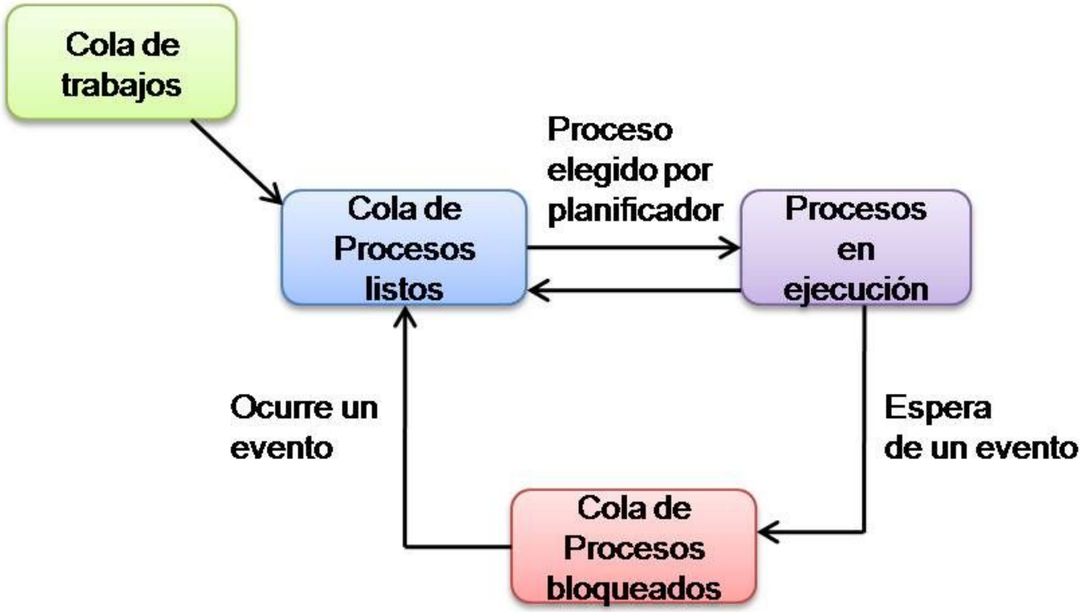
TEMA 2. PROCESOS

APARTADO 9. CAMBIO DE CONTEXTO

Algo que ocurre muy frecuentemente es que el planificador de procesos saque un proceso del estado “Ejecutándose” y lo ponga en el estado “Listo”, en este caso el SO debe ser capaz de guardar el contexto del proceso actual y restaurar el contexto del proceso que el planificador ha elegido ejecutar.

Lo cual significa que deben guardarse todos los datos de la situación del proceso que se estaba ejecutando.

Este cambio de contexto se produce también cuando hay una interrupción, es decir, cuando un proceso que se estaba ejecutando necesita un recurso y produce la interrupción para que se le proporcione.



TEMA 2. PROCESOS

APARTADO 9. CAMBIO DE CONTEXTO

Cada vez que se produzca un cambio de contexto, el SO debe guardar:

- Estado en el que se encontraba el proceso.
- Estado del procesador, debiendo guardar todos los valores que tenían los distintos registros del procesador en el momento del cambio de contexto.
- Información de gestión de memoria, debiendo guardarse el espacio de memoria que tenía reservado el proceso.
- Contador de programa, que es el encargado de indicar por dónde va ejecutándose el proceso, lo cual nos permitirá, más adelante, seguir ejecutando la instrucción del proceso desde donde se quedó.
- Puntero de pila, que se encarga de apuntar a la parte superior de la pila de ejecución del proceso.

Se puede decir que el cambio de contexto es “tiempo no aprovechado” ya que el procesador no hace un “trabajo útil” durante ese tiempo.

Una vez el proceso vuelva a entrar en estado “Ejecutándose”, todos los datos que han sido salvador se deberán volver a restaurar para que el proceso pueda seguir ejecutándose como si nada hubiese pasado.

MEDAC

Instituto Oficial de Formación Profesional

- Muchas gracias -