

UT 1 - Programación multiproceso

2º DAM – Curso 2024/2025

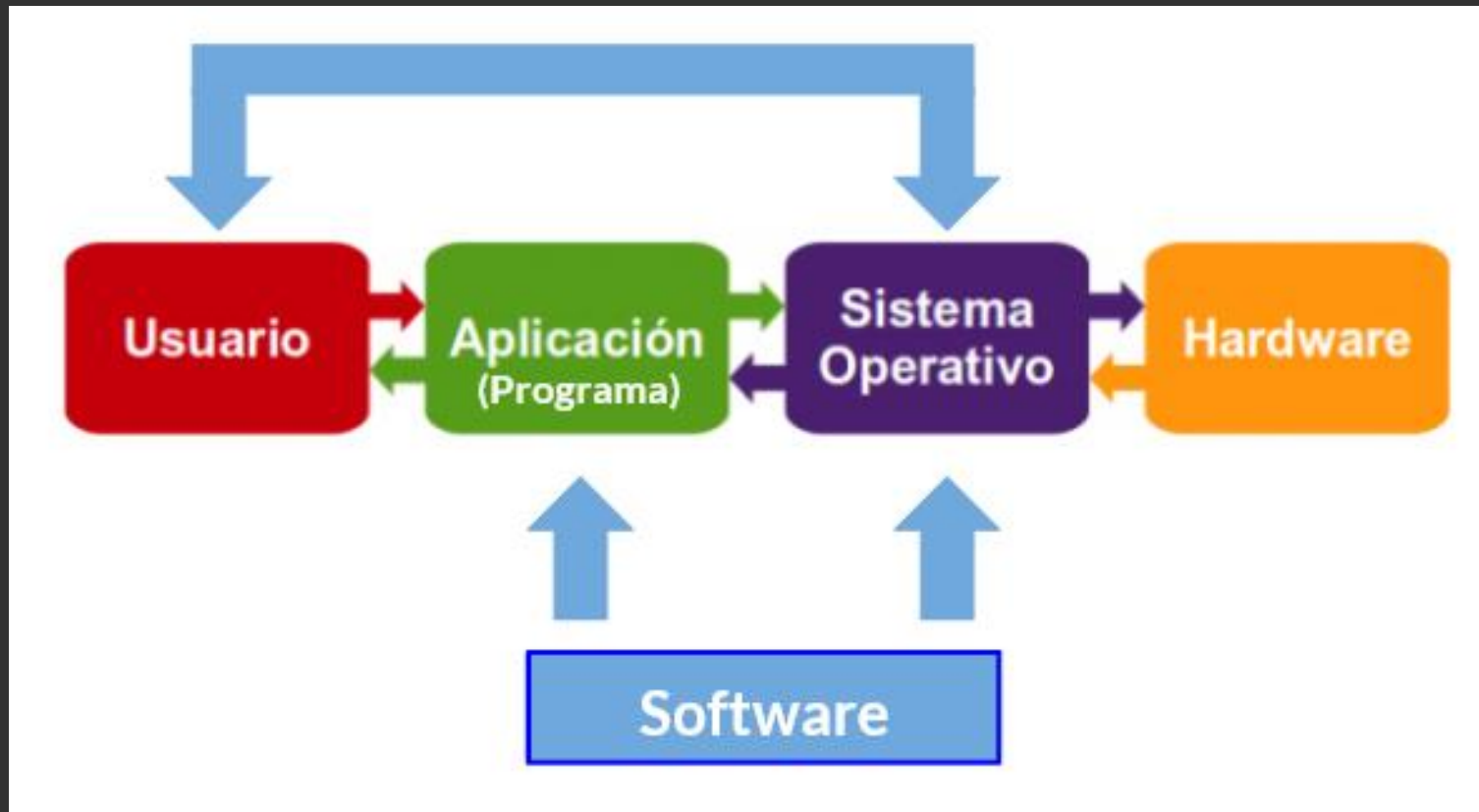
Parte 2 – Gestión de procesos

Andrés Marina Díaz
andresmd@educastur.org

Conceptos básicos

Conceptos básicos

Sistema informático: HW + SW + *User*



Conceptos básicos

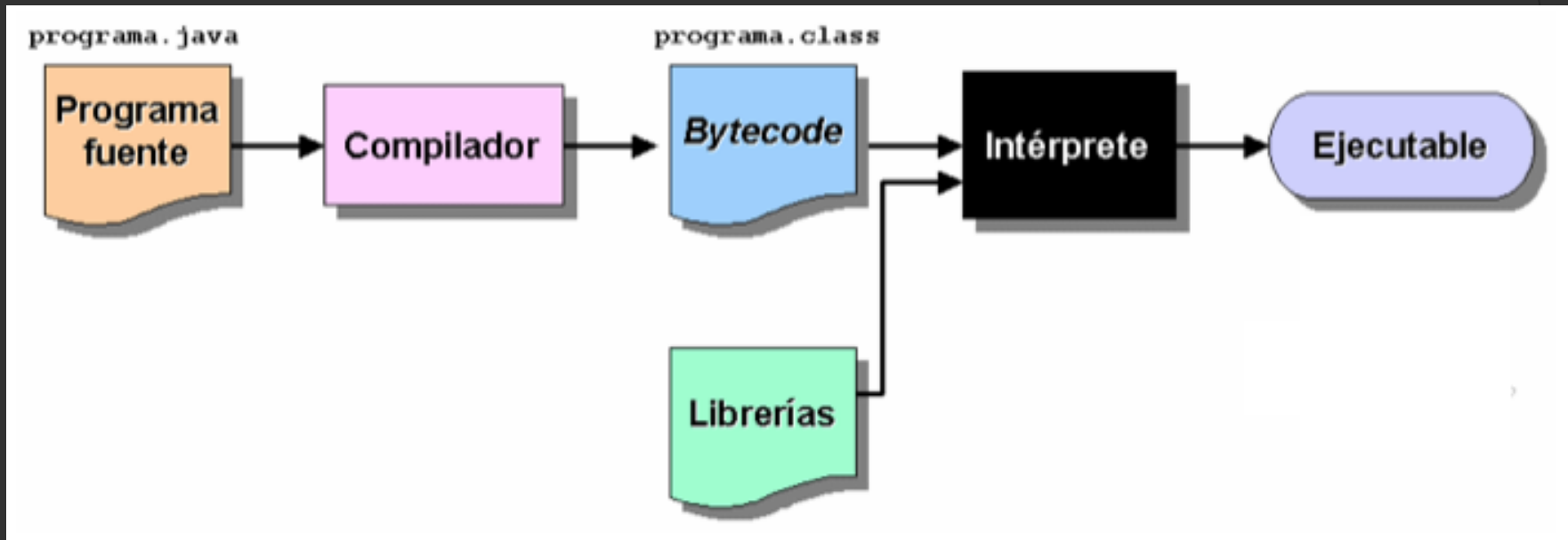
PROGRAMA

Toda la información (tanto código como datos) almacenada en memoria no volátil que resuelve una necesidad concreta para los usuarios.

Conceptos básicos

EJECUTABLE

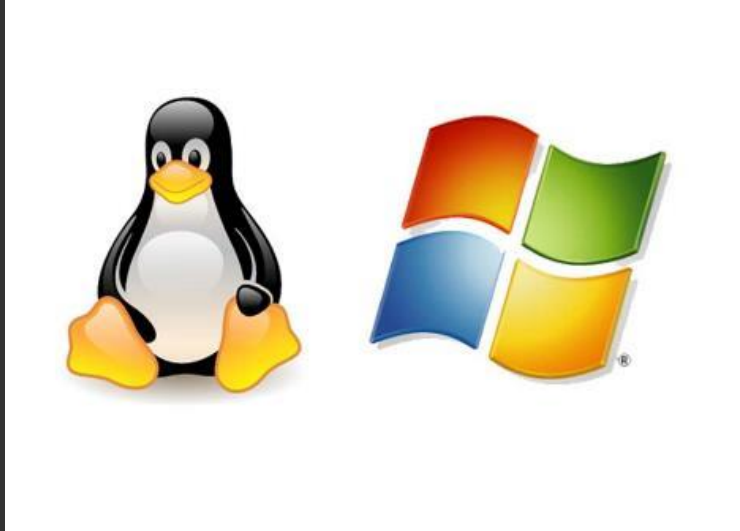
Fichero que contiene el código binario o interpretado que será ejecutado en un ordenador.



Conceptos básicos

EJECUTABLE

- Windows: archivos .exe
- Linux: Tiene activado permiso de ejecución



Conceptos básicos

EJECUTABLE – TIPOS:

- **Binarios:** Conjunto de instrucciones que directamente son ejecutadas por el procesador del ordenador (código compilado). Ej.: C.
- **Interpretados:** Conjunto de instrucciones que son interpretadas antes de ser ejecutadas por el procesador del ordenador (código interpretado). Ej.: JavaScript.

Conceptos básicos

PROCESO

Programa, o instancia de este, en ejecución. Este concepto no se refiere únicamente al código y a los datos, sino que incluye todos los recursos asignados por el SO para su correcta ejecución :

- **Contador de programa.**
- **Imagen de memoria.**
- **Estado del procesador.**

Conceptos básicos

DAEMON (DEMONIO)

Un demonio, servicio o programa residente es un **tipo especial de proceso** informático no interactivo, es decir, que se **ejecuta en segundo plano en vez de ser controlado directamente por el usuario**

Ejemplo:

sshd: Un demonio que permite conexiones seguras a través de SSH (Secure Shell).

Programación concurrente

Programación concurrente

Concurrencia: Coincidencia de varios sucesos al mismo tiempo.

Situaciones:

- Procesos que necesitan comunicarse entre ellos.
- Procesos que necesitan acceder al mismo recurso (fichero,...)

IMPORTANTE: controlar la forma para que no haya errores, resultados incorrectos o inesperados.

Programación concurrente

La programación concurrente permite la posibilidad de tener en ejecución de tener en ejecución al mismo tiempo múltiples tareas. Dichas tareas se pueden ejecutar en:

Programación concurrente

Un único procesador (multiprogramación).

- Si solamente existe un único procesador, solamente un proceso puede estar en un momento determinado en ejecución.
- El sistema operativo se encarga de cambiar el proceso en ejecución después de un período corto de tiempo (del orden de milisegundos) creando en el usuario la percepción de que múltiples programas se están ejecutando al mismo tiempo (**programación concurrente**).
- La programación concurrente no mejora el tiempo de ejecución global de los programas ya que se ejecutan intercambiando unos por otros en el procesador. Sin embargo, permite que varios programas parezca que se ejecuten al mismo tiempo.

Programación concurrente

Varios núcleos en un mismo procesador (multitarea).

- Cada núcleo (*core*) podría estar ejecutando una instrucción diferente al mismo tiempo.
- El sistema operativo se encarga de planificar los trabajos que se ejecutan en cada núcleo y cambiar unos por otros para generar multitarea.
- Todos los *cores* comparten la misma memoria por lo que es posible utilizarlos de forma coordinada (**programación paralela**).
- La programación paralela permite mejorar el rendimiento de un programa ya que permite que se ejecuten varias instrucciones a la vez.

Programación concurrente

Varios ordenadores distribuidos en red.

- Cada uno de los ordenadores tendrá sus propios procesadores y su propia memoria (**programación distribuida**).
- La programación distribuida posibilita la utilización de un gran número de dispositivos de forma paralela, lo que permite alcanzar elevadas mejoras en el rendimiento de la ejecución de programas distribuidos.
- Como cada ordenador posee su propia memoria, imposibilita que los procesos puedan comunicarse fácilmente, teniendo que utilizar otros **esquemas de comunicación más complejos y costosos** a través de la red que los interconecte.

Gestión de procesos

Gestión de procesos

- En nuestro equipo se están ejecutando, al mismo tiempo, **muchos procesos**.
- Nuestros SO son **multitarea**.



Gestión de procesos

Tipos de procesos:

- **Por lotes:** Formados por una serie de tareas, de las que el usuario sólo está interesado en el resultado final. No hay interacción/control/supervisión directa. Ej.: imprimir.
- **Interactivos:** Aquellas tareas en las que el proceso interactúa continuamente con el usuario y actúa de acuerdo con las acciones que éste realiza, o a los datos que suministra. Ej.: procesador textos.
- **Tiempo real:** Tareas en las que es crítico el tiempo de respuesta del sistema. Ej.: ordenador coche.

Gestión de procesos

Funcionamiento (1/4):

- En nuestros equipos ejecutamos distintos procesos: por lotes e interactivos.
- Microprocesador > capaz de ejecutar miles de millones de instrucciones básicas en un segundo (es decir, muchas tareas).
- SO > encargado de decidir qué proceso puede entrar a ejecutarse o debe esperar.

Gestión de procesos

Funcionamiento (2/4):

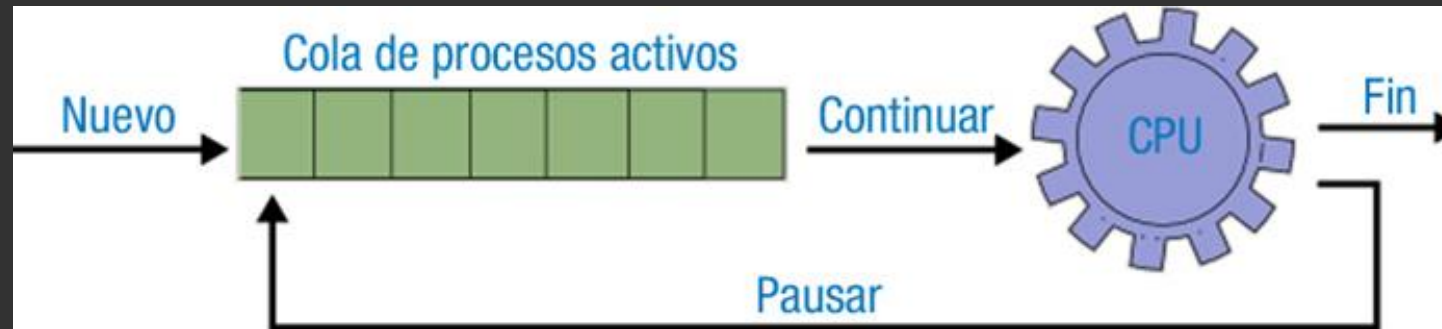
Los SO tienen que repartir el uso del microprocesador entre los distintos procesos:

- ¿Qué le sucede a un proceso cuando no se está ejecutando?
- ¿Por qué el equipo hace otras cosas mientras que un proceso queda a la espera de datos?



Gestión de procesos

Posible solución:



- *Procesos van avanzando posiciones en la cola de procesos activos hasta que les toca el turno.*
- *SO concede el uso de la CPU, a cada proceso durante un tiempo determinado y equitativo > quantum.*

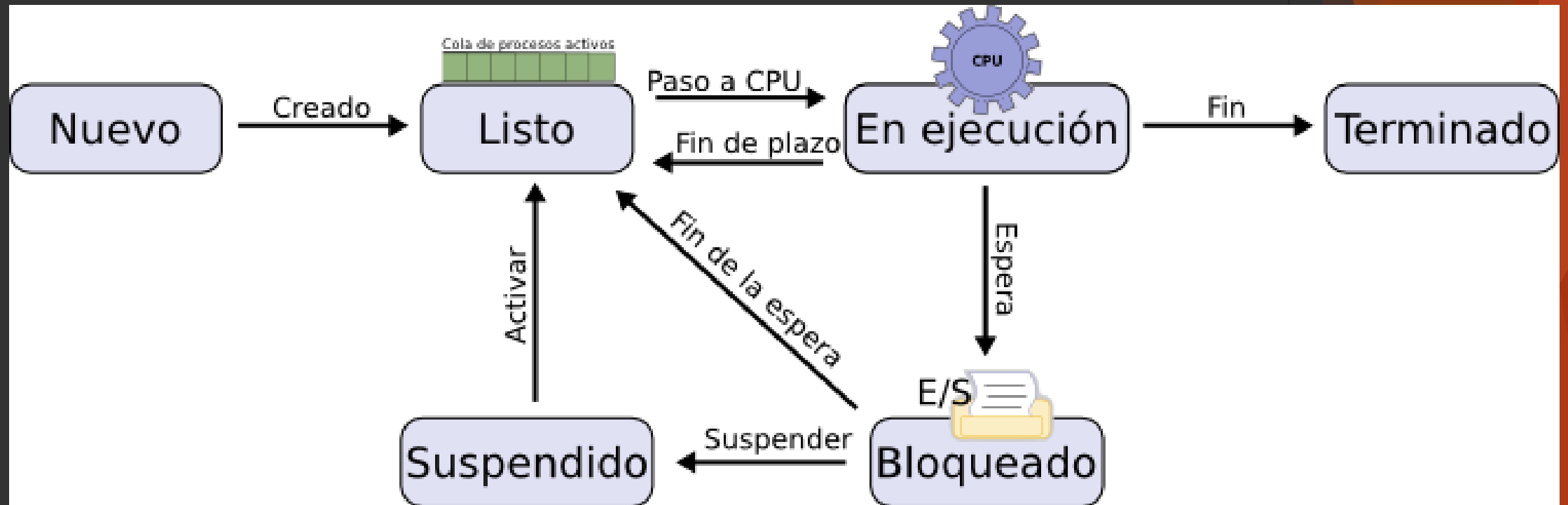
Gestión de procesos

Problemas:

- Cuando un proceso: necesita datos de un archivo, necesita una entrada de datos del usuario, tiene que imprimir o grabar datos,... > **proceso bloqueado (en espera)**.
- Cuando la memoria RAM del equipo está llena, algunos procesos deben pasar a disco para dejar espacio > **proceso suspendido**.
- Procesos “críticos” (no pueden esperar) > procesos SO.

Gestión de procesos

Estados de un proceso



Gestión de procesos

Ciclo vida proceso

- Conjunto de estados por los que transita.
- Estados básicos:
 - ❑ Activo: en ejecución o listo.
 - ❑ En espera: bloqueado o suspendido.

Recuerda: los procesos no pueden pasar por ellos mismos de listos a ejecución, es el SO el que decide.

Gestión de procesos

BCP / PCB (bloque de control de proceso/*process control block*)

Estructura de datos que contiene toda la información que requiere el SO para gestionar un proceso:

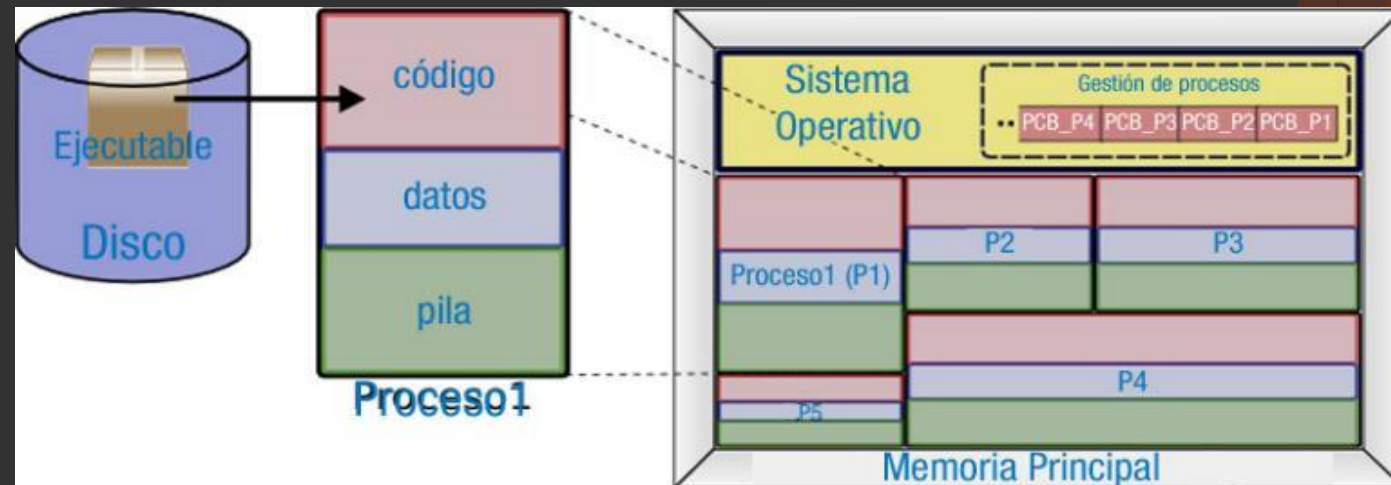
- ❑ PID: identificador del proceso.
- ❑ Estado actual.
- ❑ Espacio de direcciones de memoria.
- ❑ Información para la planificación.
- ❑ ...

Gestión de procesos

Componentes del SO para la Gestión de procesos:

Cargador: encargado de crear los procesos:

- Carga el proceso en memoria principal (RAM).
- Crea una estructura de información llamada BCP / PCB.



Gestión de procesos

Componentes del SO para la Gestión de procesos:

- **Planificador:** encargado de decidir el proceso que debe ejecutarse.



Algoritmo de planificación

Gestión de procesos

Criterios de planificación

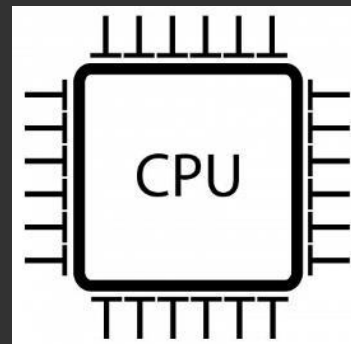
- Justicia.
- Eficiencia.
- Rendimiento.
- Tiempo de retorno.
- Tiempo de espera.
- Tiempo de respuesta.

Objetivo: optimizar ejecución procesos.

Gestión de procesos

Políticas de planificación:

- **Expulsoras / expropiativas:** Proceso puede perder la posesión del procesador sin solicitarlo.
- **No expulsoras / no expropiativas:** Proceso NO puede perder la posesión del procesador sin solicitarlo.



Gestión de procesos

Algoritmos planificación

➤ No expulsores / no expropiativos:

- ☐ Algoritmo FIFO / FCFS
- ☐ Algoritmo SJF
- ☐ Algoritmo por prioridades
- ☐ ...

➤ Expulsores / expropiativos:

- ☐ Algoritmo por prioridades
- ☐ Algoritmo Round-Robin
- ☐ ...

Gestión de procesos

Algoritmo FIFO / FCFS

- First In First Out / First Come First Served > primero en entrar – llegar, primero en salir - servir.
- **Ventajas:**
 - ❑ Fácil de implementar (simple).
 - ❑ Eficiente si hay pocos procesos.
- **Inconvenientes:**
 - ❑ Mucho volumen de procesos > mucha cola.
 - ❑ No eficiente con procesos interactivos.

Gestión de procesos

Algoritmo FIFO / FCFS [1]

Proceso	Tiempo de ejecución	Tiempo de llegada	Tiempo de comienzo	Tiempo de finalización	Tiempo de retorno	Tiempo de espera
A	8	0	0	8	8	0
B	4	1	9	12	$12 - 1 = 11$	$11 - 4 = 7$
C	9	2	13	21	$21 - 2 = 19$	$19 - 9 = 10$
D	5	3	21	26	$26 - 3 = 23$	$23 - 5 = 18$
E	2	4	26	28	$28 - 4 = 24$	$24 - 2 = 22$



Gestión de procesos

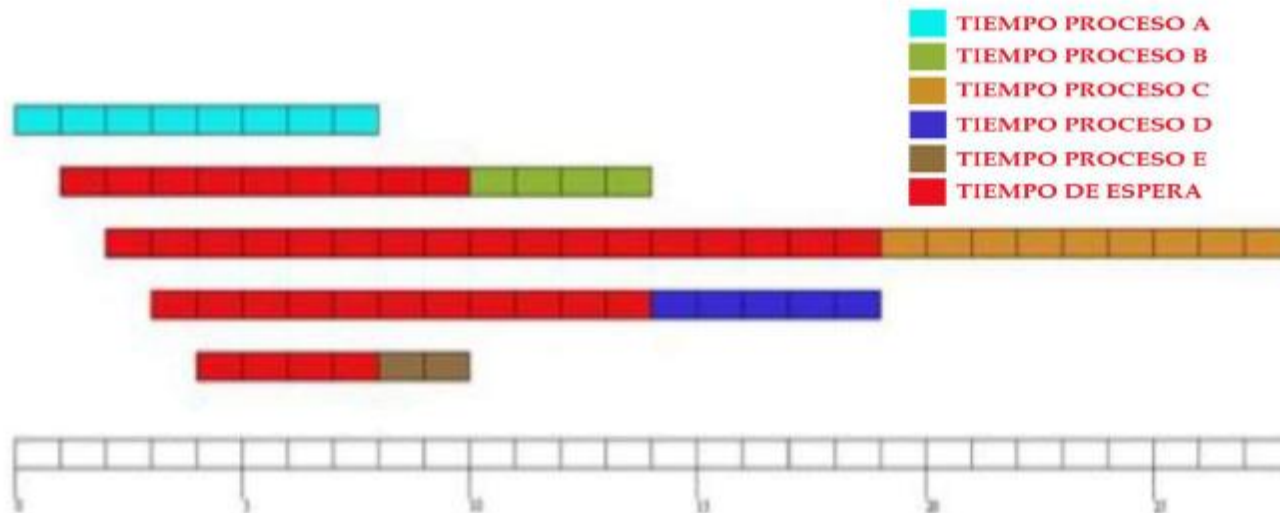
Algoritmo SJF

- Shortest Job First > primero el más corto.
- 2 procesos iguales > usa FIFO.
- **Ventaja:**
 - ❑ Procesos más cortos acaban primero.
- **Inconveniente:**
 - ❑ Procesos más largos podrían llegar a no ejecutarse (inanición).

Gestión de procesos

Algoritmo SJF

Proceso	Tiempo de ejecución	Tiempo de llegada	Tiempo de comienzo	Tiempo de finalización	Tiempo de retorno	Tiempo de espera
A	8	0	0	8	8	0
B	4	1	10	14	$14 - 1 = 13$	$13 - 4 = 9$
C	9	2	19	28	$28 - 2 = 26$	$26 - 9 = 17$
D	5	3	14	19	$19 - 3 = 16$	$16 - 5 = 11$
E	2	4	8	10	$10 - 4 = 6$	$6 - 2 = 4$



Gestión de procesos

Algoritmo por prioridades

- Criterios establecer prioridades > variados.
- Puede ser: expulsor / no expulsor.
- **Ventaja:**
 - ❑ Procesos más importantes (con prioridad alta) van antes.
- **Inconveniente:**
 - ❑ Procesos con prioridad baja podrían no llegar a ejecutarse.

Gestión de procesos

Algoritmo Round-Robin

- Se usa un temporizador > cada proceso tiene un tiempo determinado ($q = \text{quantum}$).
- Similar a FIFO pero expulsor / expropiativo.
- **Ventaja:**
 - ❑ Más “justo”.
- **Inconvenientes:**
 - ❑ q grande > “se tarda mucho en atender” (FIFO).
 - ❑ q pequeño > sobrecarga / degradación aprovechamiento CPU.

Gestión de procesos

Algoritmo Round-Robin

Proceso	Tiempo de ejecución	Tiempo de llegada	Tiempo de comienzo	Tiempo de finalización	Tiempo de retorno	Tiempo de espera
A	8	0	0-14 -23	3 -17 -25	25	$25 - 8 = 17$
B	4	1	3 -17	6 -18	$18 - 1 = 17$	$17 - 4 = 13$
C	9	2	6 - 18 - 25	9 -21 -28	$28 - 2 = 26$	$26 - 9 = 17$
D	5	3	9 - 21	12 -23	$23 - 3 = 20$	$20 - 5 = 15$
E	2	4	12	14	$14 - 4 = 4$	$10 - 2 = 2$

TIEMPO PROCESO A
TIEMPO PROCESO B
TIEMPO PROCESO C
TIEMPO PROCESO D
TIEMPO PROCESO E
TIEMPO DE ESPERA

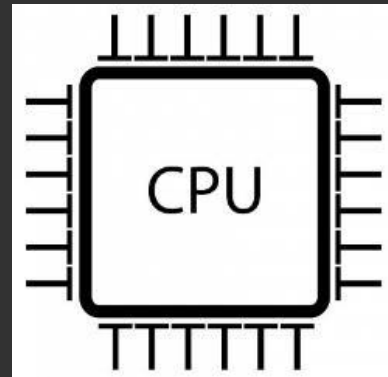


Gestión de procesos

Cambio contexto CPU (1/2)

Se realiza cada vez que la CPU cambia la ejecución de un proceso a otro distinto. Hay que:

- Guardar el estado actual de la CPU.
- Restaurar el estado de CPU del proceso que se va a ejecutar.

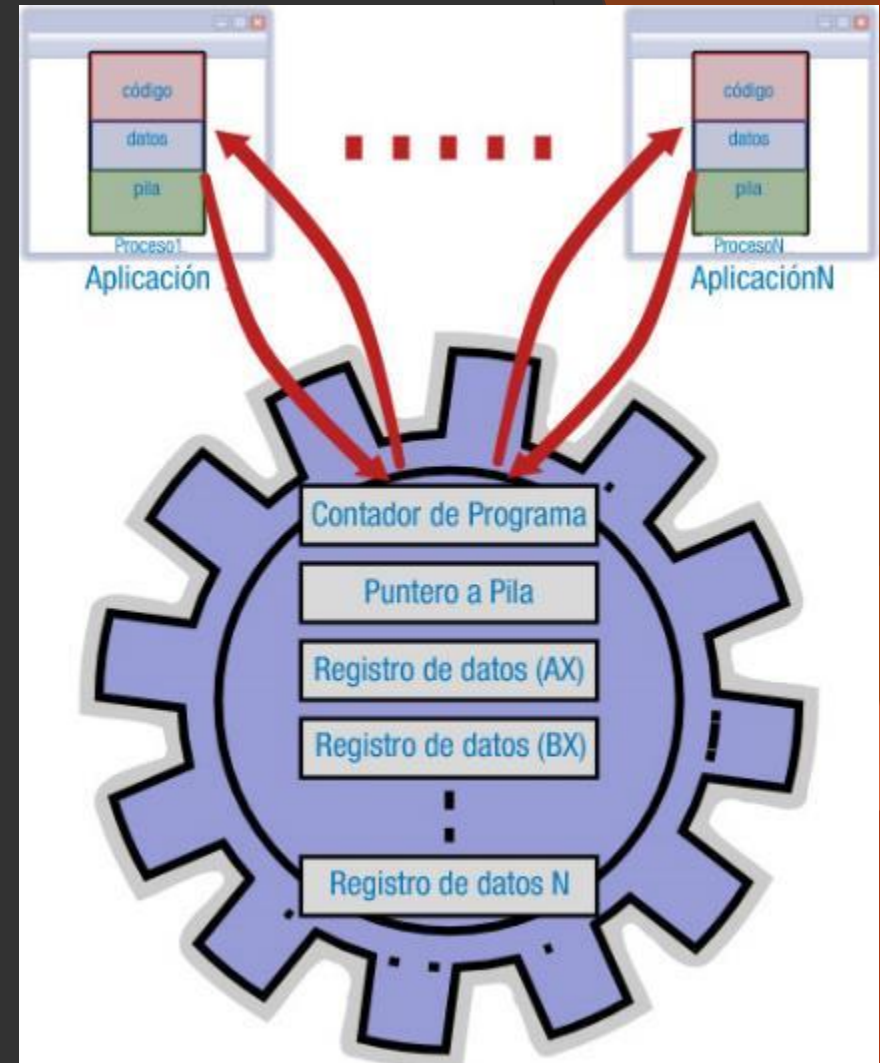


Gestión de procesos

Cambio contexto CPU (2/2)

Estado CPU > conjunto registros:

- Contador programa > almacena la dirección de la siguiente instrucción a ejecutar.
- Puntero a pila > almacena el contexto de la CPU.
- ...



DUDAS, PREGUNTAS, INQUIETUDES

