Fundamentos de los Sistemas Operativos (FSO)

Departamento de Informática de Sistemas y Computadoras (DISCA)
Universitat Politècnica de València

Bloque Temático 2: Gestión de Procesos
Unidad Temática 3

Proceso:

Concepto e implementación

fSO

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DISCA

Proceso: Concepto e implementación

- **Objetivos:**
  - Definir el concepto de **proceso**
  - Analizar las diferencias entre ejecución **secuencial** y **concurrente**
  - Definir los diferentes **estados de un proceso**, así como las causas de las transiciones entre ellos
  - Estudiar las **estructuras básicas** que dan soporte a los procesos en el SO
  - Analizar el mecanismo del **cambio de contexto**
- **Bibliografía:**
  - "Fundamentos de sistemas operativos" Silberschatz 7ª Ed
  - "Sistemas operativos: una visión aplicada" Carretero 2º Ed

Contenido

- **Conceptos previos**
- Ficheros ejecutables
- Concepto de proceso
- Estados de un proceso
- Implementación de procesos: el PCB
- Tabla de llamadas para procesos y señales
- Ejercicios propuestos

Conceptos previos

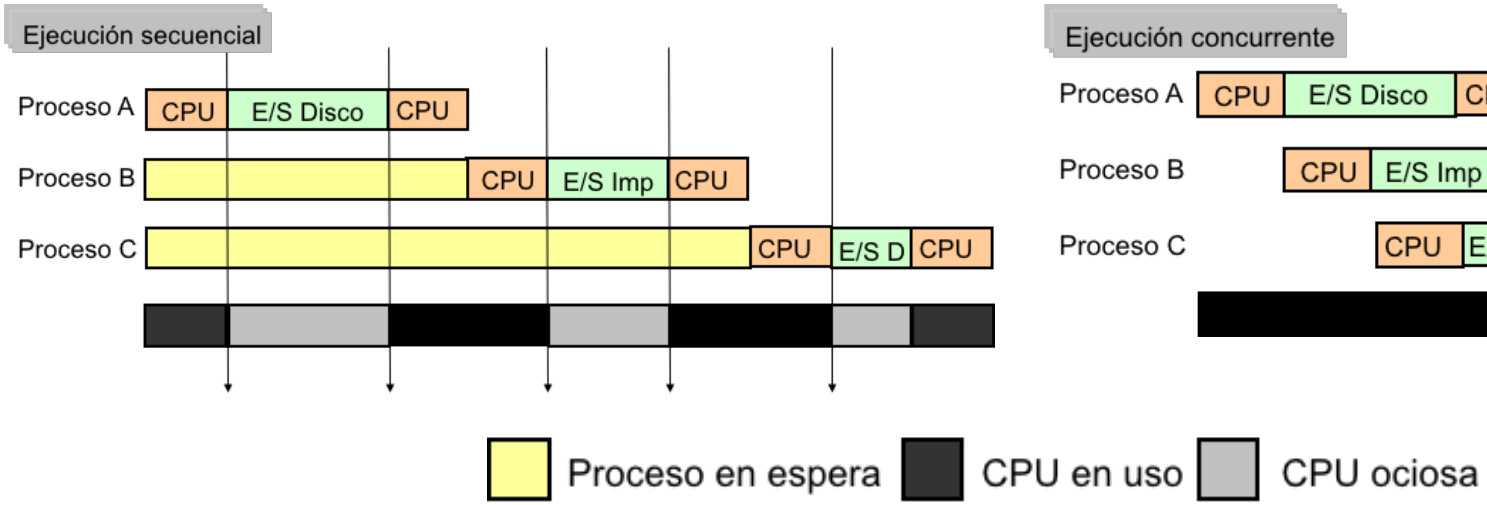
- Para poder entender el concepto de proceso, es importante tener claros algunos términos. En concreto las diferencias entre:
  - Programa/Proceso
  - Ejecución secuencial / Ejecución concurrente

Programa vs. Proceso

- **Programa**
  - **Fichero** ejecutable que es el resultado de un proceso de compilación y enlazado correcto
  - **Entidad pasiva**, no cambia con el tiempo
- **Proceso:** programa en ejecución:
  - Unidad de trabajo del sistema operativo
  - Consumidor de recursos
  - Cualquier tarea que ejecute un computador debe ser un proceso
  - Abstracción del sistema operativo que modela las actividades que aparecen en cada momento en el computador

Un proceso es una **entidad activa** que sufre cambios mientras existe

- **Ejecución secuencial vs. Ejecución Concurrente**
  - **Ejecución secuencial**: cuando la CPU es utilizada por un solo proceso desde que se inicia su ejecución hasta que termina
  - **Ejecución concurrente:** cuando la CPU es utilizada por varios procesos alternando el uso de la misma durante su vida
    - El **beneficio** más directo de la ejecución concurrente es el **incremento de uso de la CPU**, dado que un proceso no se encuentra constantemente demandando CPU sino que alterna sus demandas con ráfagas de E/S.

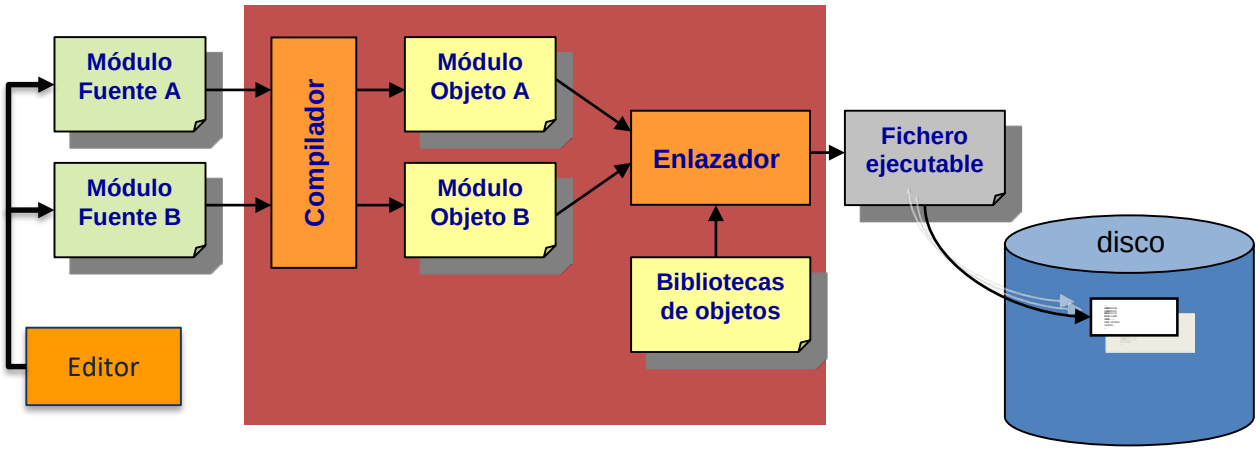


Contenido

- Conceptos previos
- **Ficheros ejecutables**
- Concepto de proceso
- Estados de un proceso
- Implementación de procesos: el PCB
- Tabla de llamadas para procesos y señales
- Ejercicios propuestos

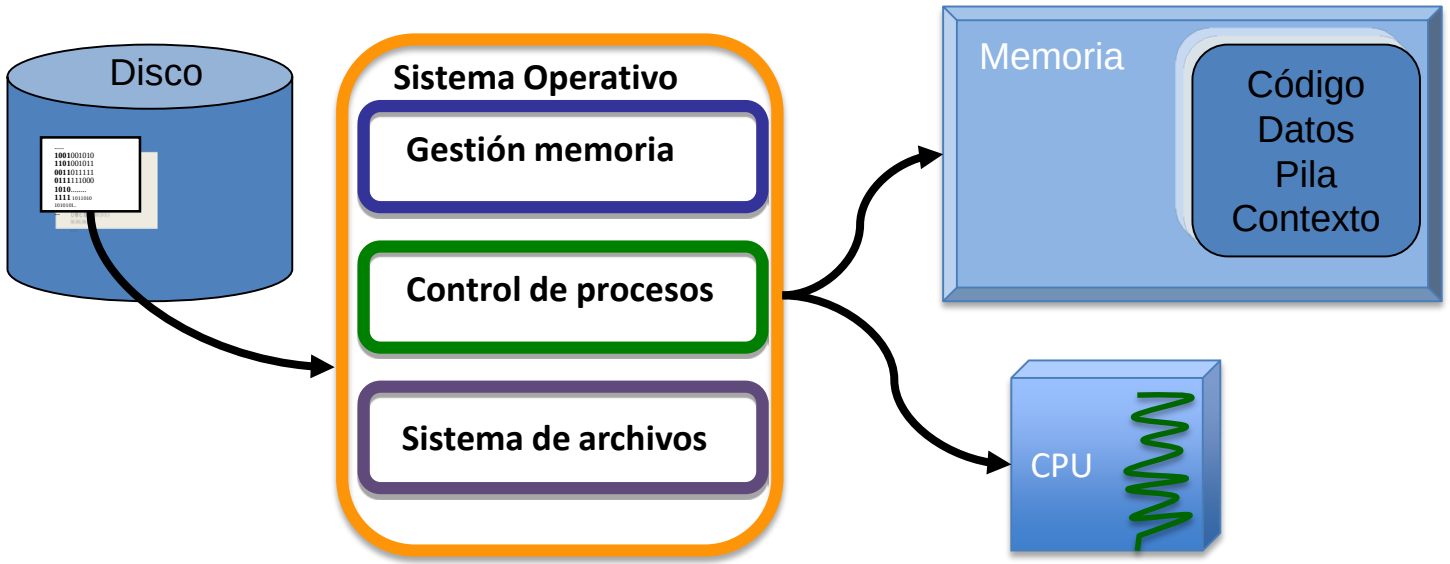
Ficheros ejecutables

- Para lanzar a ejecución un proceso es necesario partir de un **fichero ejecutable**, es decir, un fichero que contenga **código ejecutable**.
- Para obtener dicho fichero, se deben seguir los siguientes pasos**:**
  1. Disponer de un fichero de texto con el programa escrito en un lenguaje de alto nivel, que denominaremos fichero de **código fuente**.
  2. **Compilar** el fichero con **código fuente** para obtener **un código objeto**
  3. El código objeto obtenido se **enlaza con** el código de **librerías** del sistema u otras de usuario. El resultado final es un **fichero ejecutable**
     - Las librerías nos dan posibilidad de incorporar código externo



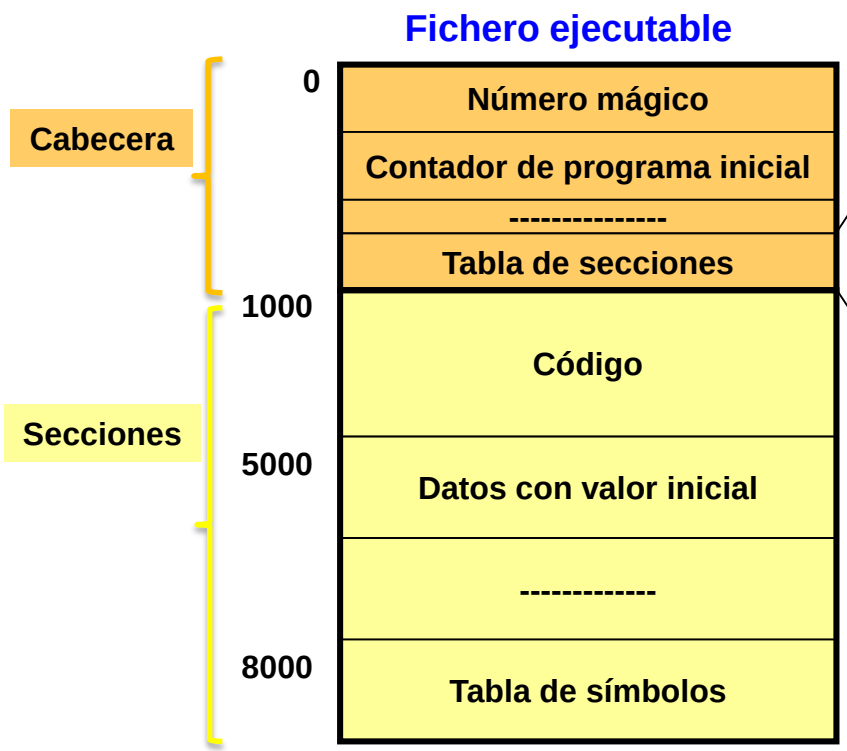
Ficheros ejecutables

- Los ficheros ejecutable son ficheros cuya estructura es bien conocida por el sistema operativo ya que contiene:
  - El código a ejecutar
  - Los datos inicializados
  - Las funciones de librería
- Con toda esta información el SO puede asignar los recursos necesarios para su ejecución. Cargar datos, código etc. e iniciar la ejecución de instrucciones, es decir, **lanzar el proceso a ejecución**



Ficheros Ejecutables

- Como **ejemplo** de formato de fichero ejecutable tenemos los **ficheros ELF (Executable and Linking Format)**
  - Es un formato de fichero para ejecutables, código objeto, librerías y volcados de memoria.
  - Muy extendido en Unix, Linux, Solaris y BSD.
  - Extensiones: .o, .so, .elf , .prx, .exe, .dll



	Despl.	Tam.
Código	1000	4000
Datos con v.i	5000	1000
Datos sin v.i	------	500
----------	------	-----
T. símbolos	8000	1000

Otros formatos:
- Windows NT: PE/COFF
- Mac OSX: Mach-O

Contenido

- Conceptos previos
- Ficheros ejecutables
- **Concepto de proceso**
- Estados de un proceso
- Implementación de procesos: el PCB
- Tabla de llamadas para procesos y señales
- Ejercicios propuestos

Concepto de proceso

- Para definir el concepto de proceso y llegar a su implementación, es necesario definir tres aspectos fundamentales:
  - **Atributos** o características que ayuden a definirlos y a gestionarlos dentro del sistema
  - **Comportamiento**, al ser un ente activo, puede pasar por diferentes estados y es necesario definir tanto los estados como las transiciones entre ellos
  - **Operaciones** que se pueden realizar sobre ellos

Atributos de un proceso

- Los atributos de un proceso son aquellas propiedades, recursos o características propias del proceso que mantiene el SO para poder gestionarlos
- A pesar de que cada sistema operativo guarda un conjunto de atributos que pueden variar, los más típicos suelen ser:
  - **Atributos de identidad:**
    - Identificador del proceso
  - **Atributos de entorno de ejecución**:
    - Directorio actual
    - Descriptores de ficheros abiertos
  - **Atributos de estado:**
    - Estado del proceso
    - Contexto máquina (contador de programa, puntero pila, registros uso general)
    - Etc.
  - **Atributos de memoria:**
    - Área de código, área datos, área pila
  - **Atributos de planificación:**
    - Tiempos de consumo de procesador
    - Prioridad
    - Monitorización

- **Operaciones sobre procesos**
  - Como pasaba con los atributos, el número y tipo de las **operaciones** que se pueden hacer sobre los procesos **varía en función del sistema** operativo.
  - No obstante, siempre podemos encontrar operaciones de:
    - Creación
    - Comunicación
    - Espera
    - Acceso a recursos
    - Finalización

Contenido

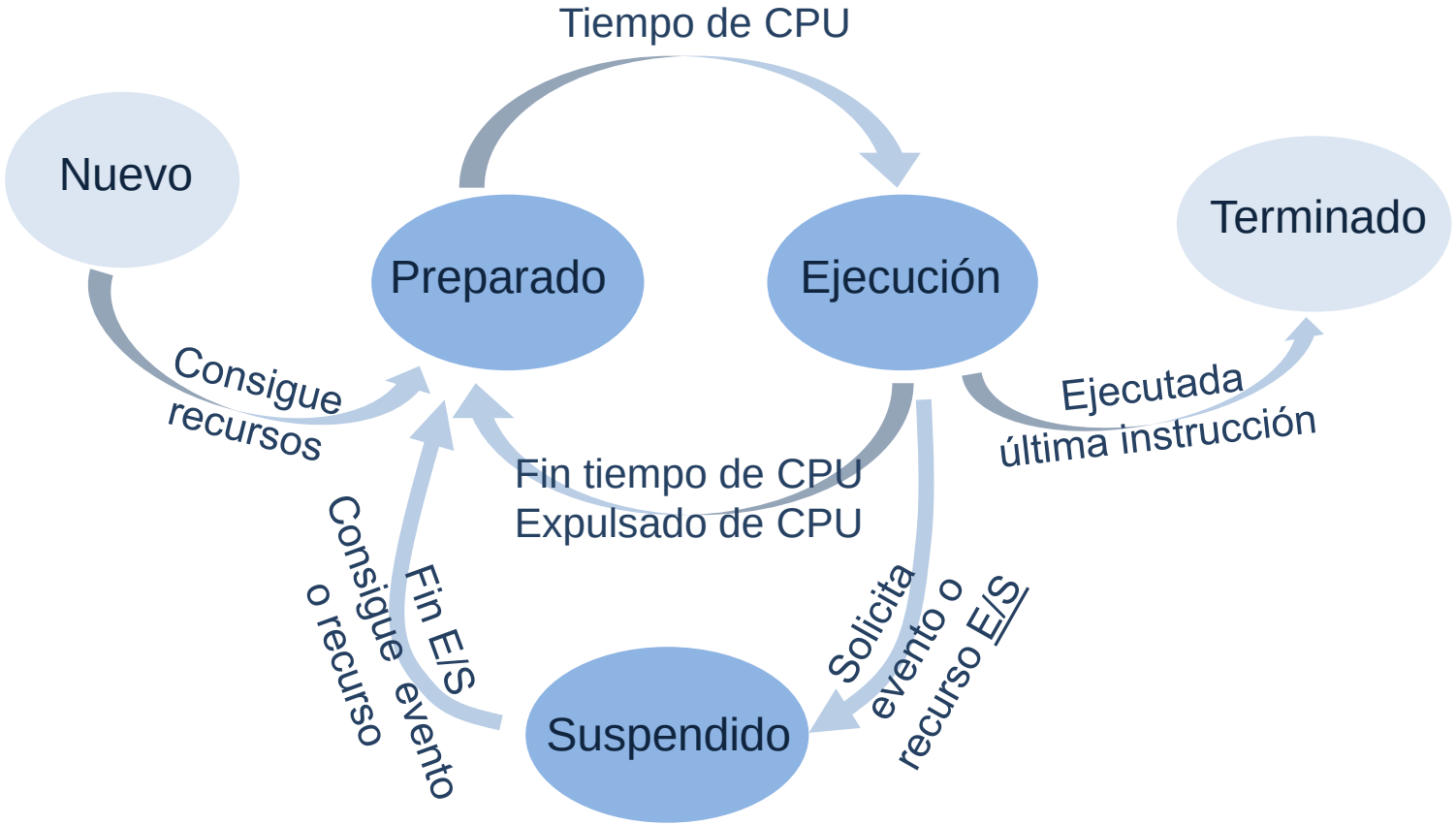
- Conceptos previos
- Ficheros ejecutables
- Concepto de proceso
- **Estados de un proceso**
- Implementación de procesos: el PCB
- Tabla de llamadas para procesos y señales
- Ejercicios propuestos

Estados de un proceso

- Los **procesos** al ser **entidades activas** durante su vida pasan por **diferentes estados**

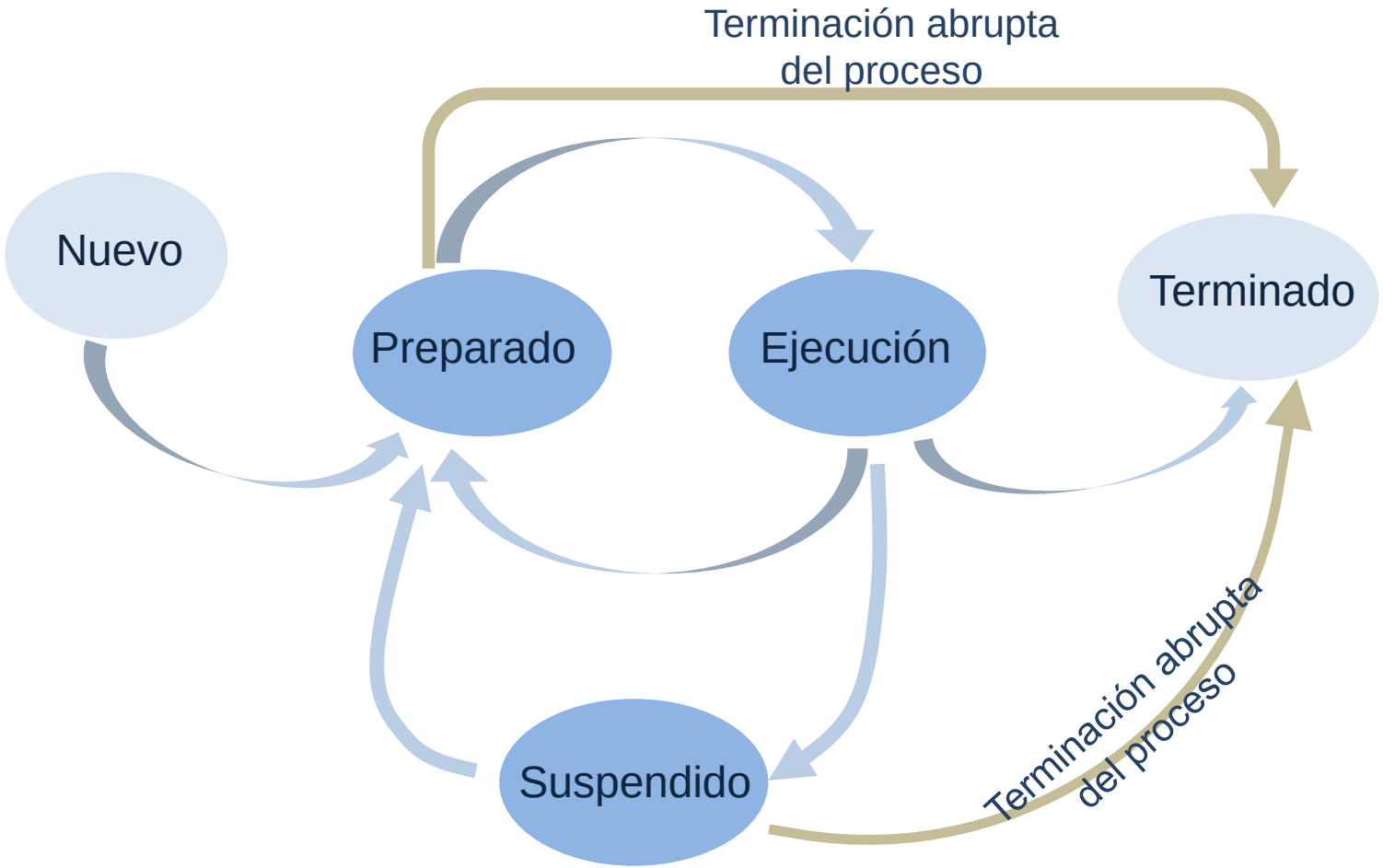
Estados de un proceso

- Para definir el comportamiento es necesario definir también las **transiciones entre estados** de un proceso en situaciones normales

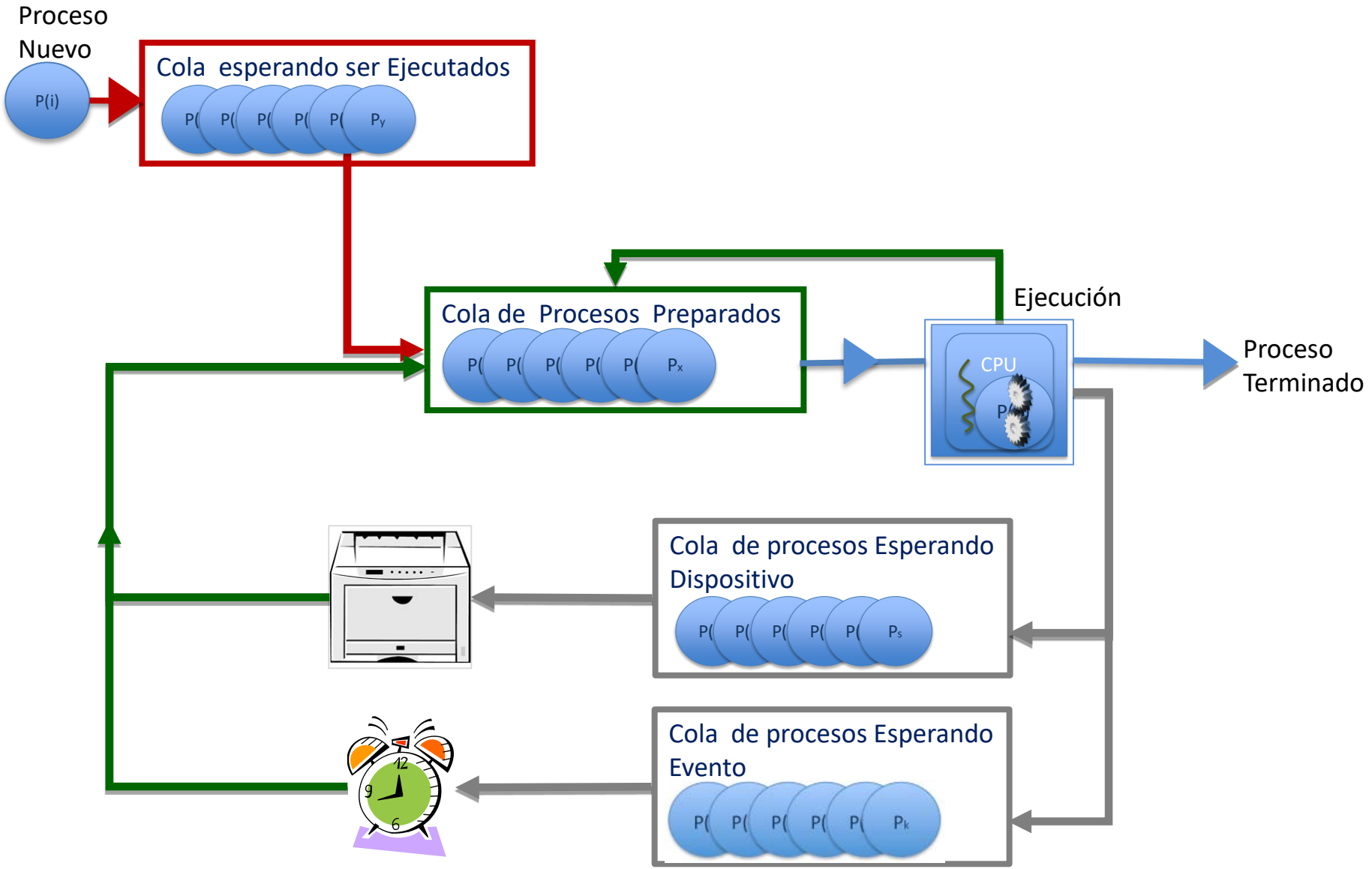


Estados de un proceso

- Y también las **transiciones** causadas **por situaciones anómalas**



Estados de un proceso

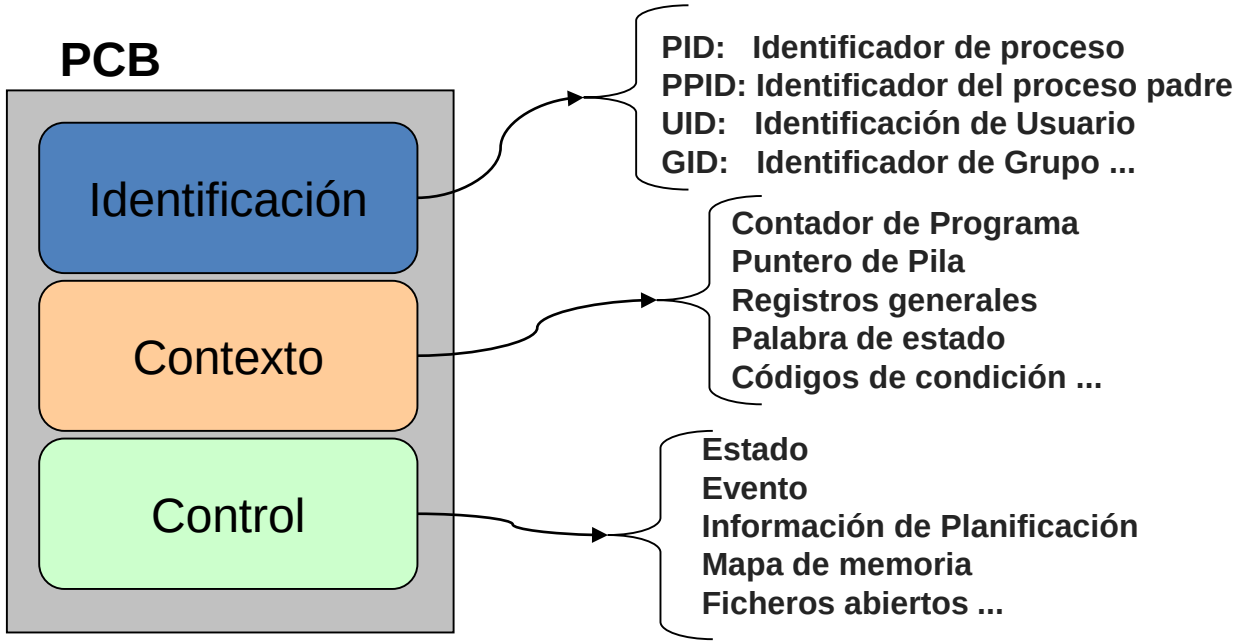


Contenido

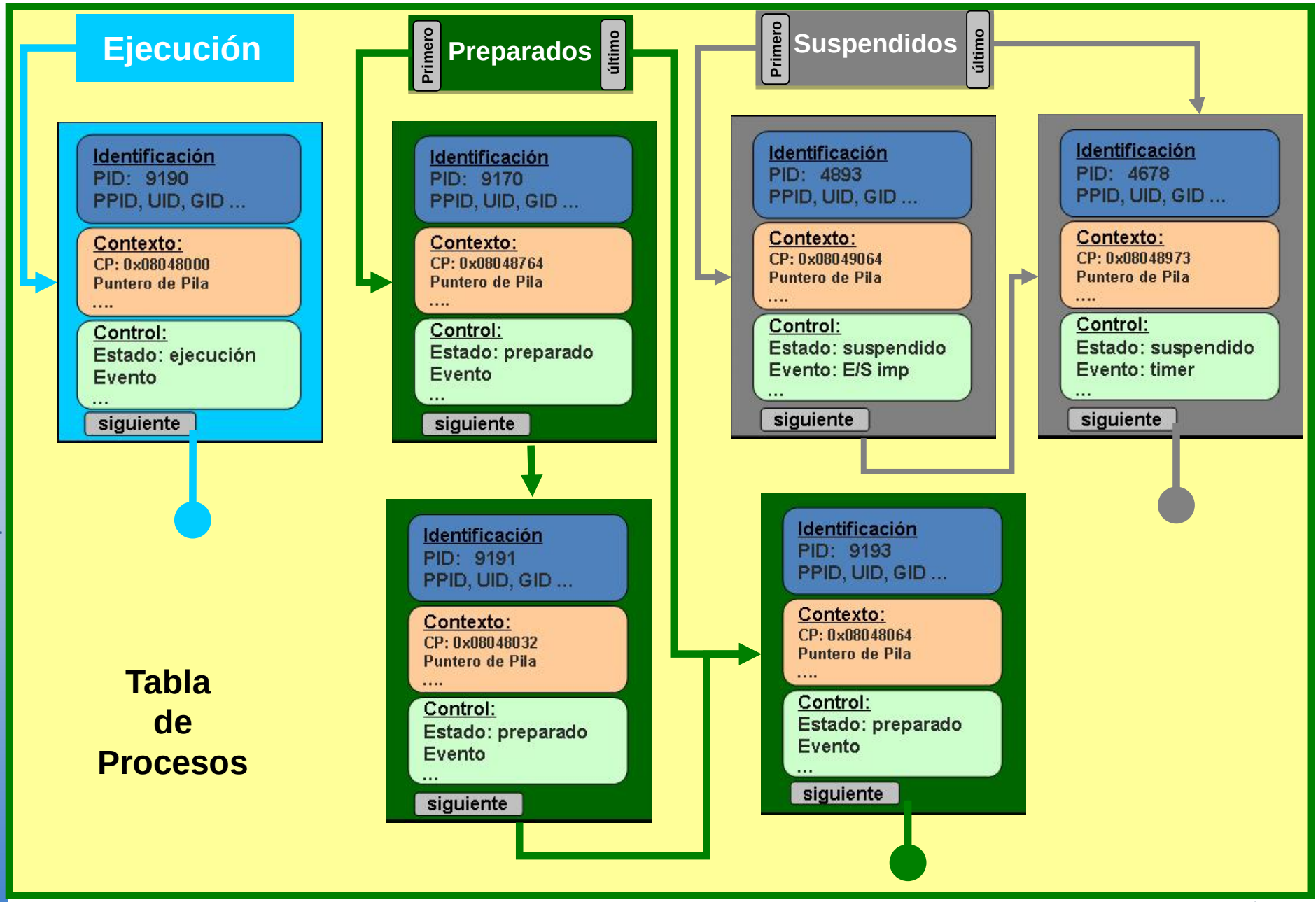
- Conceptos previos
- Ficheros ejecutables
- Concepto de proceso
- Estados de un proceso
- **Implementación de procesos: el PCB**
- Tabla de llamadas para procesos y señales
- Ejercicios propuestos

Implementación de procesos: el PCB

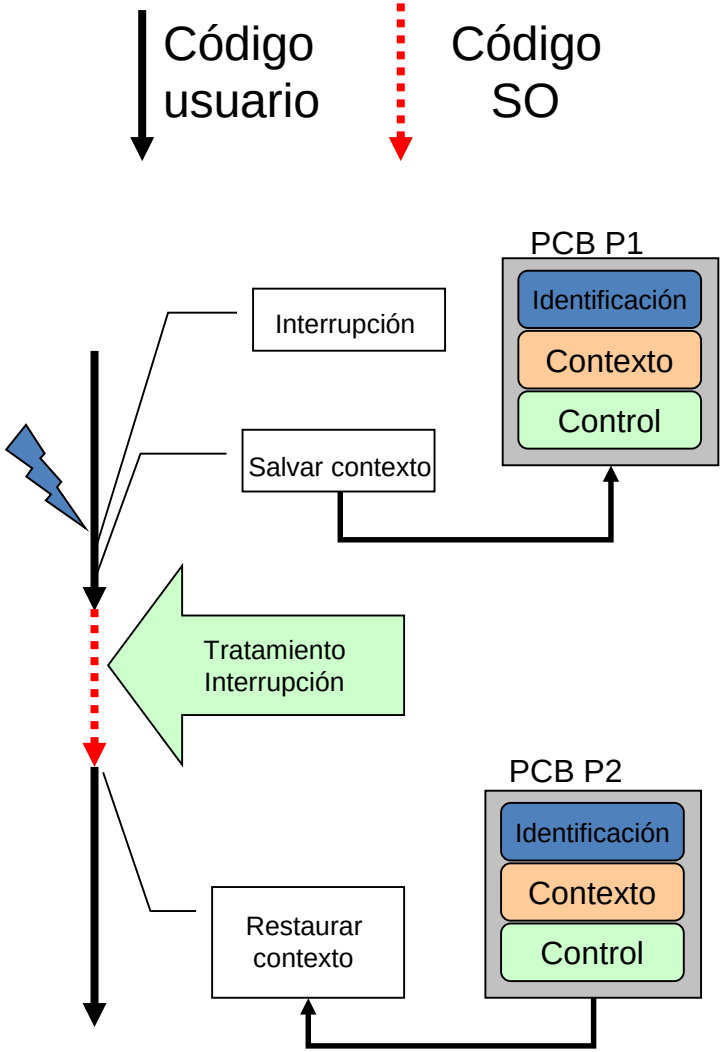
- **PCB (Process Context Block)**
  - Un PCB es la **estructura de datos** sobre la que el S.O. sustenta el concepto (abstracción) de proceso
    - Un SO también es un programa, y por tanto se basa en el uso de **algoritmos** y contiene **estructuras de datos**
  - Mantiene información relevante del proceso, la cual cambia durante la vida del mismo.
  - Cada sistema operativo tiene su propia estructura.



Implementación de procesos: el PCB



- **Cambio de contexto**
  - Mecanismo que permite a un SO detener la secuencia actual de ejecución de un proceso para iniciar o retomar la ejecución de otro proceso.
    - Este mecanismo se activa mediante una interrupción (p.ej. interrupción de reloj).
  - ¿Que se hace?
    - Se salva la información relevante de estado (contexto) del proceso en ejecución.
    - El gestor de procesos interviene para actualizar los PCBs y las colas.
    - Se transfiere el control de la CPU al nuevo proceso.



Contenido

- Conceptos previos
- Ficheros ejecutables
- Concepto de proceso
- Estados de un proceso
- Implementación de procesos: el PCB
- **Tabla de llamadas para procesos y señales**
- Ejercicios propuestos

Tabla de llamadas para procesos y señales

	Procesos
fork	Creación de un proceso hijo
exit	Terminación del proceso en ejecución
wait	Espera la terminación de un proceso
exec	Cambia imagen de memoria por la de un ejecutable (ejecuta programa)
getpid	Obtiene atributos de un proceso
setsid	Modifica atributos de un proceso

	Señales
kill	Enviar una señal
alarm	Generar una alarma (señal de reloj)
sigaction	Permite instalar un manejador de señales
sigsuspend	Suspende a un proceso mientras espera que ocurra una señal
sigemptyset	Iniciar una máscara para que no tenga señales seleccionadas
sigfillset	Iniciar una máscara para que contenga todas las señales
sigaddset	Poner una señal específica en un conjunto de señales
sigdelset	Quitar una señal específica en un conjunto de señales
sigismember	Consultar si una señal pertenece a un conjunto de señales
sigprocmask	Examinar/modificar máscara de señales

Contenido

- Conceptos previos
- Ficheros ejecutables
- Concepto de proceso
- Estados de un proceso
- Implementación de procesos: el PCB
- Tabla de llamadas para procesos y señales
- **Ejercicios propuestos**

Ejercicio UT03.1 Dada la siguiente lista de acciones, diga para cada una de ellas si es el código del sistema operativo (SO) o/y el código del interprete de órdenes (IO) el responsable de llevarla a cabo. Indíquelo marcando con una cruz el lugar que corresponda

SO	IO	
		Leer una línea de órdenes e interpretarla
		Programar un controlador de dispositivo
		Proporcionar una interfaz de llamadas al sistema
		Seleccionar un proceso para asignarle CPU
		Invocar una llamada al sistema
		Proporcionar una interfaz cómoda de usuario

Ejercicio UT03.2 ¿Cuál sería el estado (nuevo, preparado, ejecución, suspendido, terminado) en que se encuentran cada uno de los siguientes procesos:

Proceso		Estado
P1	La CPU está ejecutando instrucciones de P1	
P2	P2 ha solicitado un acceso a disco, pero el disco está siendo accedido por P3	
P3	P3 está accediendo a disco	
P4	P4 es un proceso que corresponde a un usuario que finaliza todos sus trabajos en un terminal y se desconecta	
P5	Al proceso P5 se le ha asignado un identificador de procesos y sólo se han construido las tablas necesarias para gestionarlo	
P6	Las tablas necesarias para la gestión de P6, así como su imagen de memoria , han sido cargadas en memoria	

Ejercicio UT03.3. Dada la siguiente línea de órdenes:

```
$ cat f1 f2 f3 | grep comienza | wc -l >traza
```

Indique:

a) ¿Cuántos procesos se crearían durante su ejecución por un sistema UNIX?

b) ¿Qué archivos de E/S lleva asociada cada orden?