

# Sistemas Operativos

## Procesos e hilos

Eloy Anguiano

Rosa M. Carro

Ana González

Escuela Politécnica Superior  
Universidad Autónoma de Madrid



Escuela  
Politécnica  
Superior

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

# Parte I

## procesos

Un sistema informático debe poder realizar las siguientes acciones:

- Intercalar la ejecución de múltiples procesos para maximizar la utilización del procesador ofreciendo a la vez un tiempo de respuesta razonable.
- Asignar los recursos a los procesos.
- Dar soporte a la comunicación entre procesos y la creación de procesos por parte del usuario.

## Generalidades

Sistema informático

### Proceso

Ejemplo de  
planificación de  
procesos

Procedimiento de  
multiprogramación

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

- También se llama tarea
- Ejecución de un programa individual
- Traza del proceso
  - Listado de la secuencia de instrucciones que se ejecutan para dicho proceso
  - Distinguir entre Proceso (concepto dinámico) y Programa (concepto estático)

## Componentes

Proceso = Código en forma de objeto

- + BCP (PID, estado, prioridad, CP, registros, ficheros, E/S, ...)
- + Pila (datos temporales, parámetros, direcciones de retorno...)
- + Datos (globales)

# Generalidades

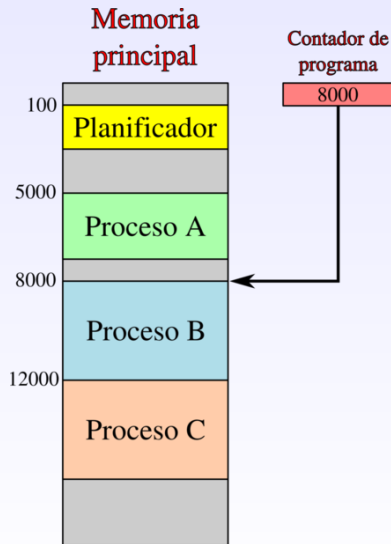
## Ejemplo de planificación de procesos

Traza de los procesos

A	B	C
5000	8000	12000
5001	8001	12001
5002	8002	12002
5003	8003	12003
5004		12004
5005		12005
5006		12006
5007		12007
5008		12008
5009		12009
5010		12010
5011		12011

Tabla de localización de procesos

A	B	C
5000	8000	12000



# Generalidades

## Ejemplo de planificación de procesos

Procesos e hilos

Suponemos que los procesos se ejecutan en orden: A, B y C.  $CP = 5000$ .

**100 = Dirección del inicio del planificador, Sombreado: ejecución del planificador**

Ciclo	Dirección	Ciclo	Dirección	Ciclo	Dirección	Ciclo	Dirección
1	5000	14	8001	27	12004	40	5011
2	5001	15	8002	28	12005	Fin de plazo	
3	5002	16	8003	Fin de plazo		41	100
4	5003	Terminación del Proceso		29	100	42	101
5	5004	17	100	30	101	43	102
6	5005	18	101	31	102	44	103
Fin de plazo		19	102	32	103	45	104
7	100	20	103	33	104	46	105
8	101	21	104	34	105	47	12006
9	102	22	105	35	5006	48	12007
10	103	23	12000	36	5007	49	12008
11	104	24	12001	37	5008	50	12009
12	105	25	12002	38	5009	51	12010
13	8000	26	12003	39	5010	52	12011

Generalidades

Sistema informático

Proceso

Ejemplo de  
planificación de  
procesos

Procedimiento de  
multiprogramación

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

# Generalidades

## Procedimiento de multiprogramación

Procesos e hilos

### Generalidades

Sistema informático

Proceso

Ejemplo de  
planificación de  
procesos

### Procedimiento de multiprogramación

Modelos de estados  
del proceso

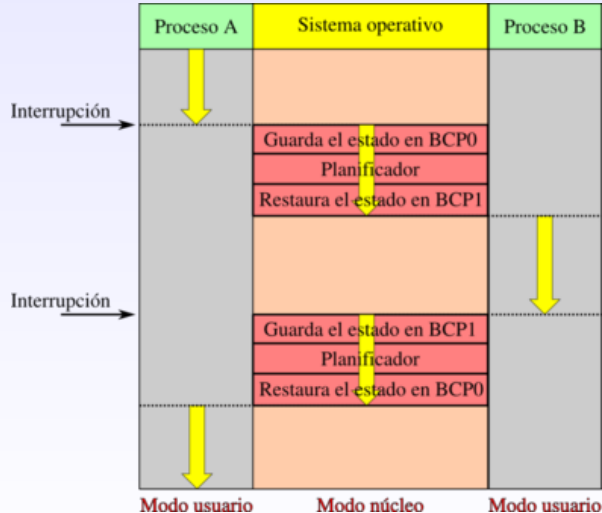
Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos



# Modelos de estados del proceso

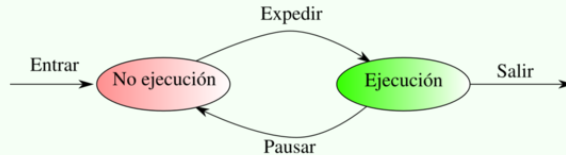
## 2 estados

### Estados

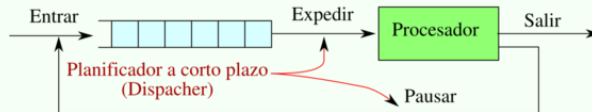
- 1 Ejecución
- 2 No ejecución

### Diagrama

#### Diagrama de transición de estados



#### Diagrama de colas





# Modelos de estados del proceso

## 2 estados

### Generalidades

#### Modelos de estados del proceso

2 estados

5 estados

6 estados

7 estados

#### Estructuras de control

#### Imagen de un proceso

#### Bloque de control de proceso

#### Modos de ejecución

#### Manipulación de procesos

### Procesos en el estado de no ejecución

- Listos para ejecutarse
- Esperando a que termine una operación de E/S

### Problemas

Sólo una cola de “no ejecución”  $\Rightarrow$  el distribuidor podría no seleccionar el proceso que lleva más tiempo en la cola porque éste podría estar bloqueado.

# Modelos de estados del proceso

## 5 estados

Un proceso puede estar en uno de estos cinco estados: **Nuevo, Listo, Ejecución, Bloqueado y Finalizado**

Diagrama de transición de estados

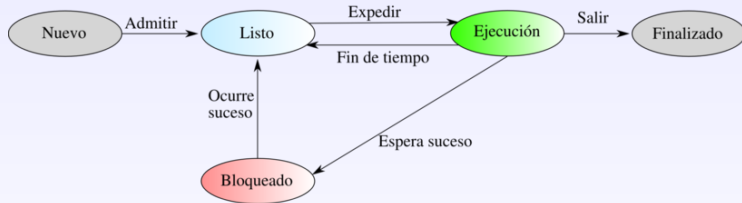
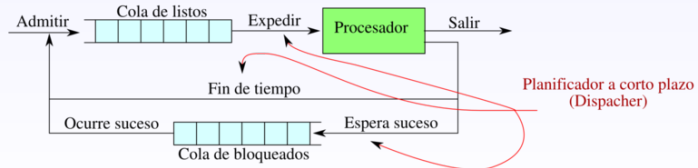


Diagrama de colas



# Modelos de estados del proceso

## 5 estados

### Generalidades

#### Modelos de estados del proceso

- 2 estados
- 5 estados**
- 6 estados
- 7 estados

#### Estructuras de control

#### Imagen de un proceso

#### Bloque de control de proceso

#### Modos de ejecución

#### Manipulación de procesos

### Estados

**Ejecución:** el proceso que está actualmente en ejecución.

**Listo:** procesos que están preparados para ejecutarse en cuanto se le dé la oportunidad.

**Bloqueado:** procesos que no se pueden ejecutar porque esperan que termine algún suceso (operación de E/S).

**Nuevo:** procesos que se acaban de crear, pero que aún no se han cargado en memoria principal.

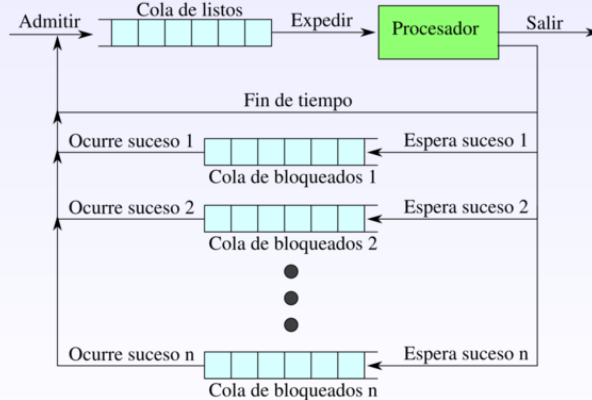
**Finalizado:** procesos que han sido excluidos por el S.O. del grupo de procesos ejecutables.

# Modelos de estados del proceso

## 5 estados

Típicamente existen múltiples colas para procesos en estado bloqueado, una para cada tipo de suceso. El diagrama de colas equivalente sería por tanto el siguiente:

### Diagrama de múltiples colas de bloqueados

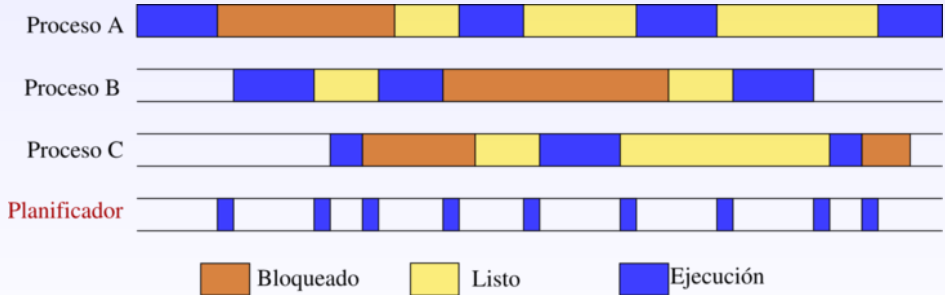


# Modelos de estados del proceso

## 5 estados

### Diagrama de Gantt

En los diagramas de Gantt no se suele marcar el planificador dado que su tiempo de ejecución suele ser despreciable frente a los tiempos de ejecución de los procesos.



# Modelos de estados del proceso

## 5 estados

### Situaciones en las que se **crea un proceso**

- 1 Emisión de un trabajo por lotes.
- 2 Un nuevo usuario intenta conectarse.
- 3 Se crea para ofrecer un servicio, como por ejemplo la impresión.
- 4 Un proceso origina la creación de otro.

### Situaciones para el **cierre de un proceso**

- 1 Un trabajo por lotes debe incluir una instrucción de detención (End).
- 2 El usuario se desconecta.
- 3 El usuario puede abandonar una aplicación.
- 4 Una serie de errores y condiciones de fallo pueden llevarnos a la terminación de un proceso.

Generalidades

Modelos de estados  
del proceso

2 estados

**5 estados**

6 estados

7 estados

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

# Modelos de estados del proceso

## 5 estados

### Motivos de cierre de un proceso

- ❶ Terminación normal.
- ❷ Tiempo límite excedido.
- ❸ No hay memoria disponible.
- ❹ Violación de límites.
- ❺ Error de protección
  - Por ejemplo: escribir en un archivo que es sólo de lectura.
- ❻ Error aritmético.
- ❼ Tiempo máximo de espera rebasado.
  - El proceso ha esperado más allá del tiempo máximo especificado para que se produzca cierto suceso
- ❽ Fallo de E/S.

Generalidades

Modelos de estados  
del proceso

2 estados

**5 estados**

6 estados

7 estados

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

# Modelos de estados del proceso

## 5 estados

### Motivos de cierre de un proceso (II)

- 9 Instrucción ilegal
  - A menudo cuando intenta ejecutar los datos.
- 10 Instrucción privilegiada.
- 11 Mal uso de los datos.
- 12 Intervención del operador o del S.O.
  - Por ejemplo, si se produce un interbloqueo.
- 13 Terminación del padre, por lo que terminan los procesos de todos sus descendientes.
- 14 Solicitud del padre.

### Carencias del modelo de cinco estados

El distribuidor podría no seleccionar exactamente el proceso que está en el extremo más antiguo de la cola porque podría estar bloqueado a la espera de la finalización de una orden de E/S.



# Modelos de estados del proceso

## 5 estados

### Generalidades

#### Modelos de estados del proceso

2 estados

**5 estados**

6 estados

7 estados

### Estructuras de control

#### Imagen de un proceso

#### Bloque de control de proceso

#### Modos de ejecución

#### Manipulación de procesos

### Problema

- El procesador es más rápido que la E/S, por lo que suele ser habitual que todos los procesos de memoria estén esperando por E/S y por lo tanto se tiene toda la memoria ocupada con procesos pero ninguno se está ejecutando.

### Solución

- Intercambiar una parte del proceso o todo el proceso al disco para liberar la memoria principal → **Intercambio**.
- Creación de un nuevo estado, el de **Suspendido**, en el que el proceso se almacena en el disco.
- Cuando los procesos de la memoria principal están en el estado Bloqueado, el sistema operativo puede suspender un proceso poniéndolo en estado Suspendido.

# Modelos de estados del proceso

## 6 estados

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

2 estados

5 estados

**6 estados**

7 estados

Estructuras de control

Imagen de un proceso

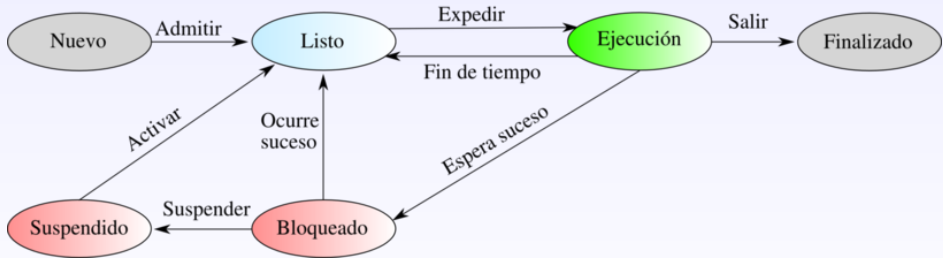
Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

Un proceso puede estar en uno de estos cinco estados: **Nuevo, Listo, Ejecución, Bloqueado, Suspendido y Finalizado**

### Diagrama de transición de estados



# Modelos de estados del proceso

## 6 estados

### Generalidades

#### Modelos de estados del proceso

2 estados  
5 estados  
**6 estados**  
7 estados

#### Estructuras de control

#### Imagen de un proceso

#### Bloque de control de proceso

#### Modos de ejecución

#### Manipulación de procesos

### Nuevo estado

**Suspendido:** el proceso está en la **memoria secundaria** esperando un suceso

### ¿Qué es suspender un proceso?

Pasar una parte de (o todo) el proceso al disco para liberar la memoria principal.

### ¿Por qué?

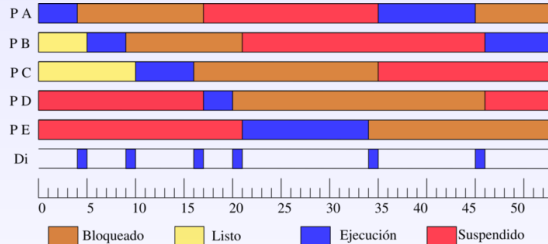
- El procesador es más rápido que la E/S  $\Rightarrow$  es habitual que haya procesos en memoria esperando a E/S.
- Cuando los procesos en memoria principal están bloqueados, el S.O. puede eliminarlos de la memoria poniéndolo en estado Suspendido, y así queda más memoria disponible para los procesos listos y para admitir nuevos procesos.

# Modelos de estados del proceso

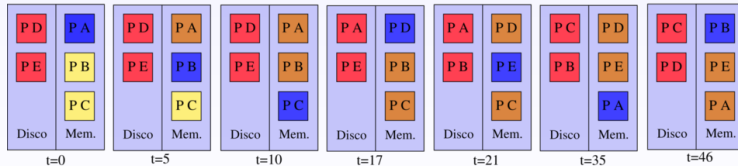
## 6 estados

Procesos e hilos

Diagrama de Gantt con el estado suspendido y estado de la memoria y el disco en cada instante.



### Memoria



# Modelos de estados del proceso

## 6 estados

### Generalidades

#### Modelos de estados del proceso

- 2 estados
- 5 estados
- 6 estados**
- 7 estados

### Estructuras de control

### Imagen de un proceso

### Bloque de control de proceso

### Modos de ejecución

### Manipulación de procesos

### Problema

- En este modelo sólo pueden ser suspendidos aquellos procesos que están en estado de bloqueado.  
Podría necesitarse memoria para admitir procesos nuevos y no existir procesos bloqueados candidatos a ser suspendidos.

### Solución

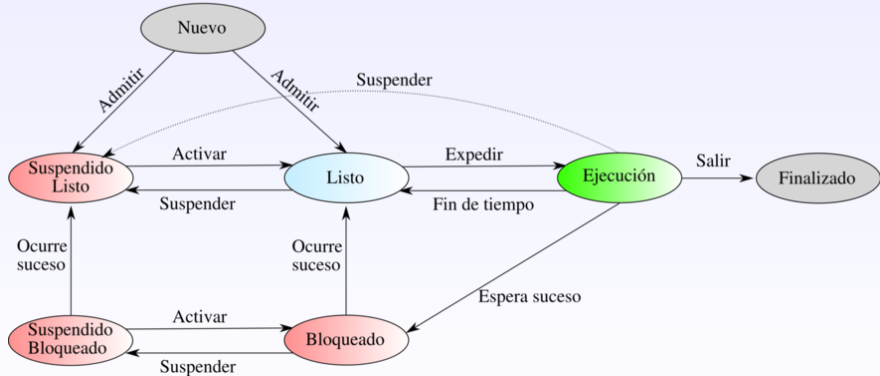
- Suspende procesos que estén en el estado de listos.
- El estado de suspendido se divide en dos estados: Suspendido-bloqueado y Suspendido-listo.

# Modelos de estados del proceso

## 7 estados

Un proceso puede estar en uno de estos cinco estados: **Nuevo**, **Listo**, **Ejecución**, **Bloqueado**, **Bloqueado-Suspendido**, **Listo-Suspendido** y **Finalizado**

Diagrama de transición de estados



# Modelos de estados del proceso

## 7 estados

### Generalidades

#### Modelos de estados del proceso

- 2 estados
- 5 estados
- 6 estados
- 7 estados**

#### Estructuras de control

#### Imagen de un proceso

#### Bloque de control de proceso

#### Modos de ejecución

#### Manipulación de procesos

### Nuevos estados

**Bloqueado y suspendido:** el proceso está en la **memoria secundaria** esperando un suceso

**Listo y suspendido:** el proceso está en la **memoria secundaria** disponible para su ejecución tan pronto como se cargue en la memoria principal.

# Modelos de estados del proceso

## 7 estados

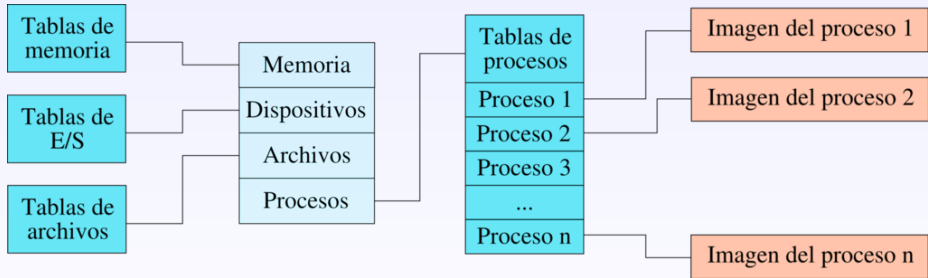
### Motivos para suspender un proceso

- **Intercambio:** El sistema operativo necesita liberar suficiente memoria principal para cargar un proceso que está listo para ejecutarse.
- **Otra razón del S.O.:** El sistema operativo puede suspender a un proceso subordinado o de utilidad, o a un proceso que se sospecha que sea el causante de un problema.
- **Solicitud de un usuario interactivo:** Un usuario puede querer suspender la ejecución de un programa con fines de depuración o en conexión con el uso de un recurso.
- **Temporización:** Un proceso puede ejecutarse periódicamente (por ejemplo, un proceso de contabilidad o de supervisión del sistema) y puede ser suspendido mientras espera el siguiente intervalo de tiempo.
- **Solicitud del proceso padre:** Un proceso padre puede querer suspender la ejecución de un descendiente para examinar o modificar el proceso suspendido o para coordinar la actividad de varios descendientes.



# Estructuras de control

- El objetivo es mantener la información sobre el estado actual de cada proceso y de cada recurso.
- El sistema operativo construye tablas de información sobre cada entidad que esté administrando.





# Estructuras de control

## Tablas de memoria

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

Estructuras de control

**Tablas de memoria**

Tablas de E/S

Tablas de archivos

Tablas de procesos

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

### Las tablas de memoria almacenan la siguiente información:

- La asignación de memoria principal a los procesos.
- La asignación de memoria secundaria a los procesos.
- Atributos de protección de bloques de memoria principal o virtual. P.ej: qué procesos pueden acceder a ciertas regiones compartidas de memoria.
- Cualquier información necesaria para gestionar la memoria virtual.

# Estructuras de control

## Tablas de E/S

### Las tablas de E/S almacenan la siguiente información:

- Estado del dispositivo de E/S, que puede estar disponible o asignado a un proceso en particular.
- Estado de la operación de E/S.
- Posición de memoria principal que se está utilizando como origen o destino de la transferencia de E/S.

# Estructuras de control

## Tablas de archivos

### Generalidades

### Modelos de estados del proceso

### Estructuras de control

Tablas de memoria

Tablas de E/S

**Tablas de archivos**

Tablas de procesos

### Imagen de un proceso

### Bloque de control de proceso

### Modos de ejecución

### Manipulación de procesos

## Las tablas de archivos almacenan la siguiente información:

- La existencia de los archivos.
- La posición de los archivos en la memoria secundaria.
- El estado actual de los archivos.
- Otros atributos de archivos.
- A veces esta información es mantenida por un sistema de gestión de archivos independiente.

# Estructuras de control

## Tablas de procesos

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Tablas de memoria

Tablas de E/S

Tablas de archivos

Tablas de procesos

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

### Las tablas de procesos almacenan la siguiente información:

- Ubicación del proceso, que incluye:
  - Instrucciones a ejecutar.
  - Datos para las variables locales y globales.
  - Constantes definidas.
  - Pila para almacenamiento de parámetros y direcciones de retorno de procedimientos.
- Bloque de control del proceso, que incluye una colección de atributos. Conjunto de **metadatos** necesarios para la administración del proceso:
- Atributos del proceso necesarios para su administración:
  - ID del proceso.
  - Estado del proceso.
  - Ubicación en la memoria.

# Imagen de un proceso

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

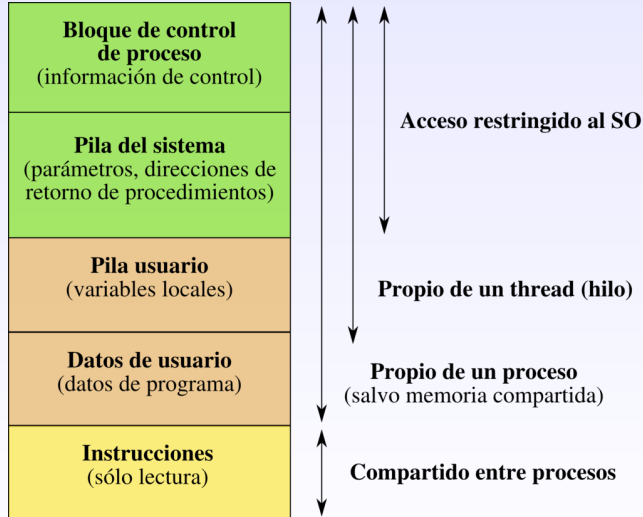
Estructuras de control

**Imagen de un proceso**

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos



# Bloque de control de proceso

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

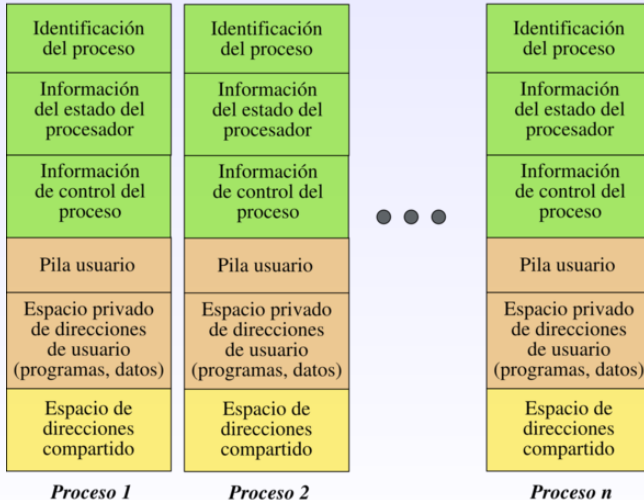
**Bloque de control de  
proceso**

Identificadores de  
proceso

Información del  
estado del procesador  
Información de  
control del proceso

Modos de ejecución

Manipulación de  
procesos



***Bloque de  
control de  
proceso***

# Bloque de control de proceso

## Identificadores de proceso

Son una serie de identificadores numéricos entre los que deben estar:

- Identificador de este proceso.
- Identificador del proceso que creó a este proceso (el proceso padre).
- Identificador del usuario.

### Ejemplo UNIX (ps -aux)

```
PPID PID PGID SID TTY TPGID STAT UID TIME COMMAND
1 3759 3759 3759 tty1 4066 Ss 0 0:00 /bin/login --
1 3760 3760 3760 tty2 3760 Ss+ 0 0:00 /sbin/getty 38400 tty2
1 3761 3761 3761 tty3 3761 Ss+ 0 0:00 /sbin/getty 38400 tty3
1 3762 3762 3762 tty4 3762 Ss+ 0 0:00 /sbin/getty 38400 tty4
1 3763 3763 3763 tty5 3763 Ss+ 0 0:00 /sbin/getty 38400 tty5
1 3764 3764 3764 tty6 3764 Ss+ 0 0:00 /sbin/getty 38400 tty6
3759 4066 4066 3759 tty1 4066 S+ 0 0:00 -bash
7979 7986 7986 7986 pts/0 7986 Ss+ 1000 0:00 /bin/bash SSH_AGENT_PID
15977 15985 15985 15985 pts/1 16725 Ss 1000 0:00 bash LC_ALL=es_ESeuro
15985 16725 16725 15985 pts/1 16725 R+ 1000 0:00 ps eaj SSH_AGENT_PID=3
```



# Bloque de control de proceso

## Información del estado del procesador

### Generalidades

### Modelos de estados del proceso

### Estructuras de control

### Imagen de un proceso

### Bloque de control de proceso

### Identificadores de proceso

### Información del estado del procesador

### Información de control del proceso

### Modos de ejecución

### Manipulación de procesos

- 1 Formada por el contenido de los registros del procesador.

- Registros visibles para el usuario.
- Registros de control y de estado.
- Punteros de pila.

- 1 **Registros visibles para el usuario:**

- Son aquellos a los que puede hacerse referencia por medio del lenguaje de máquina que ejecuta el procesador. Normalmente, existen de 8 a 32, aunque algunas implementaciones RISC tienen más de 100. Por ejemplo, en C se puede hacer referencia a ellos de la siguiente forma: `register int i;`

- 2 **Punteros de pila:**

- Cada proceso tiene una o más pilas LIFO del sistema asociadas. Las pilas se utilizan para almacenar los parámetros y las direcciones de retorno de los procedimientos y de las llamadas al sistema. El puntero de pila siempre apunta a la cima de la pila.

# Bloque de control de proceso

## Información del estado del procesador

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Identificadores de  
proceso

**Información del  
estado del procesador**

Información de  
control del proceso

Modos de ejecución

Manipulación de  
procesos

- ③ **Registros de control y de estado.** Hay varios registros del procesador que se emplean para controlar su funcionamiento. Entre éstos se incluyen:
- Contador de programa: contiene la dirección de la próxima instrucción a leer.
  - Códigos de condición: muestran el resultado de la operación aritmética o lógica más reciente (signo, cero, acarreo, igualdad, desbordamiento).
  - Información de estado: incluye los indicadores de habilitación o inhabilitación de interrupciones y de modo de ejecución. PSW (Palabra de estado del programa).

CF = Marca de acarreo

# Bloque de control de proceso

## Información de control del proceso

### Generalidades

### Modelos de estados del proceso

### Estructuras de control

### Imagen de un proceso

### Bloque de control de proceso

### Identificadores de proceso

### Información del estado del procesador

### Información de control del proceso

### Modos de ejecución

### Manipulación de procesos

- ④ Información de planificación y de estado. La necesita el sistema operativo para llevar a cabo sus funciones de planificación. Elementos típicos de esta información:
  - **Estado del proceso:** disposición del proceso para ser elegido para su ejecución (en ejecución, listo, bloqueado).
  - **Prioridad:** se puede usar uno o más campos para describir la prioridad de planificación de los procesos. En algunos sistemas se necesitan varios valores (por omisión, actual, la más alta permitida).
  - **Información de planificación:** dependerá del algoritmo de planificación utilizado. Ejemplos: cantidad de tiempo que el proceso ha estado esperando, cantidad de tiempo que el proceso ejecutó la última vez, etc.
  - **Suceso:** identidad del suceso que el proceso (bloqueado) está esperando antes de poder reanudarse.

# Bloque de control de proceso

## Información de control del proceso

### Generalidades

### Modelos de estados del proceso

### Estructuras de control

### Imagen de un proceso

### Bloque de control de proceso

### Identificadores de proceso

### Información del estado del procesador

### Información de control del proceso

### Modos de ejecución

### Manipulación de procesos

## 5 Estructuración de datos

- Un proceso puede estar enlazado con otros procesos en una cola, un anillo o alguna otra estructura. Por ejemplo, todos los procesos que están en estado de espera de un nivel determinado de prioridad pueden estar enlazados en una cola. Un proceso puede mostrar una relación padre-hijo (creador-creado) con otro proceso. El bloque de control de proceso puede contener punteros a otros procesos para dar soporte a estas estructuras.

## 6 Comunicación entre procesos:

- Puede haber varios indicadores, señales y mensajes asociados con la comunicación entre dos procesos independientes. Una parte de esta información o toda ella se puede guardar en el bloque de control de proceso.

# Bloque de control de proceso

## Información de control del proceso

### Generalidades

### Modelos de estados del proceso

### Estructuras de control

### Imagen de un proceso

### Bloque de control de proceso

### Identificadores de proceso

### Información del estado del procesador

### Información de control del proceso

### Modos de ejecución

### Manipulación de procesos

## 7 Privilegios de los procesos.

- A los procesos se les otorgan privilegios en términos de la memoria a la que pueden acceder y el tipo de instrucciones que pueden ejecutar. Además, también se pueden aplicar privilegios al uso de los servicios y utilidades del sistema.

## 8 Gestión de memoria.

- Esta sección puede incluir punteros a las tablas de páginas o segmentos que describen la memoria virtual asignada al proceso.

## 9 Propiedad de los recursos y utilización.

- Se pueden indicar los recursos controlados por el proceso, como los archivos abiertos. También puede incluir un historial de la utilización del procesador o de otros recursos; esta información puede ser necesaria para el planificador.

# Modos de ejecución

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

**Modos de ejecución**

Manipulación de  
procesos

- **Modo de usuario:**
  - Es el modo menos privilegiado.
  - Los programas de usuarios ejecutan normalmente en ese modo.
- **Modo del sistema, modo de control o modo del núcleo:**
  - Es el modo más privilegiado.
  - Núcleo del sistema operativo.

# Manipulación de procesos

## Creación de procesos

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

**Creación de procesos**

Cambio de proceso

Llamadas al sistema

Cambio de contexto

¿Qué sucede cuando  
llega una  
interrupción?

Después de atender  
una interrupción

Cambio de modo

Secuencia en un  
cambio de proceso

Ejecución del SO

- 1 Asignar un único identificador al nuevo proceso.
- 2 Asignar espacio para el proceso.
- 3 Iniciar el bloque de control del proceso.
- 4 Establecer los enlaces apropiados.
  - Por ejemplo: añadir un proceso nuevo a una lista enlazada que se utiliza como cola de planificación.
- 5 Crear o ampliar otras estructuras de datos.
  - Por ejemplo: mantener un archivo de contabilidad, actualizar la tabla de gestión de memoria.



# Manipulación de procesos

## Cambio de proceso

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

Creación de procesos

**Cambio de proceso**

Llamadas al sistema

Cambio de contexto  
¿Qué sucede cuando  
llega una  
interrupción?

Después de atender  
una interrupción

Cambio de modo

Secuencia en un  
cambio de proceso

Ejecución del SO

Existen varias situaciones que pueden hacer que el SO cambie el estado de un proceso:

- ❶ Interrupción de reloj:
  - El proceso en ejecución ha consumido la fracción máxima de tiempo permitida.
- ❷ Interrupción de E/S.
- ❸ Fallo de memoria (con memoria virtual):
  - La dirección de memoria se encuentra en la memoria virtual, por lo tanto debe ser llevada a la memoria principal.
- ❹ Cepos:
  - Se ha producido un error.
  - Puede hacer que el proceso que se estaba ejecutando pase al estado de Terminado.
- ❺ Llamada del supervisor o llamadas al sistema:
  - Como la operación de abrir un archivo.

# Manipulación de procesos

## Llamadas al sistema

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

Creación de procesos

Cambio de proceso

**Llamadas al sistema**

Cambio de contexto

¿Qué sucede cuando  
llega una  
interrupción?

Después de atender  
una interrupción

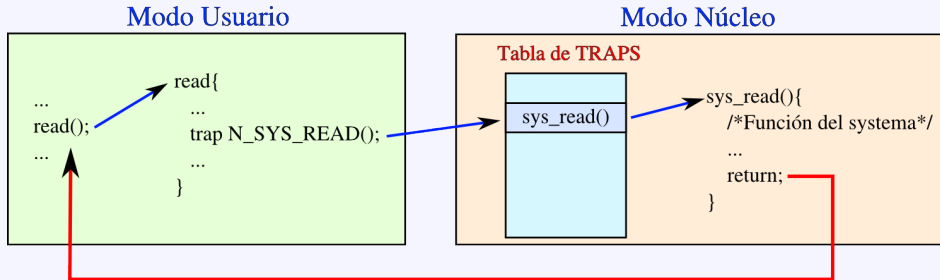
Cambio de modo

Secuencia en un  
cambio de proceso

Ejecución del SO

- Los procesos se comunican con el SO a través de llamadas al sistema
- Cada llamada se corresponde normalmente con un procedimiento que lee los parámetros de la llamada y los pasa al SO, junto con el control, mediante un TRAP. Esta llamada pasa de modo usuario a modo supervisor.

### Esquema simplificado





# Manipulación de procesos

## Cambio de contexto

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

Creación de procesos

Cambio de proceso

Llamadas al sistema

**Cambio de contexto**

¿Qué sucede cuando  
llega una  
interrupción?

Después de atender  
una interrupción

Cambio de modo

Secuencia en un  
cambio de proceso

Ejecución del SO

### Guardar contexto

- 1 Guarda CP en la pila de núcleo del proceso (o en el BCP)
- 2 Guarda el contenido de los registros del procesador en el BCP del proceso (o en la pila de núcleo del proceso)
- 3 Guarda el puntero de pila en el BCP



# Manipulación de procesos

## Cambio de contexto

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

Creación de procesos

Cambio de proceso

Llamadas al sistema

**Cambio de contexto**

¿Qué sucede cuando  
llega una  
interrupción?

Después de atender  
una interrupción

Cambio de modo

Secuencia en un  
cambio de proceso

Ejecución del SO

**Recuperar contexto: copiar en los registros del procesador los valores almacenados en el BCP del proceso a ejecutar**

1. Recupera puntero de pila del BCP
2. Recupera los demás registros del BCP y de la pila
3. Cambia algunos bits de la palabra de estado:
  - Cambia el bit para habilitar las interrupciones
  - Cambia el modo de ejecución (pasa a modo usuario)
4. Recupera el CP. Al recuperar el valor de CP deja de ejecutar el S.O. y se ejecuta el programa de usuario.

# Manipulación de procesos

## ¿Qué sucede cuando llega una interrupción?

Ocurre siempre que llega una interrupción, no sólo cuando hay que cambiar de un proceso a otro

- 1 Se **salva el contexto** del programa que se ejecuta (copia la información del estado del procesador: CP en la pila, registros en BCP, puntero de pila en el BCP)
- 2 Asigna al **CP** el valor de la dirección de **comienzo del programa de atención a la interrupción**
- 3 Cambia de modo usuario a **modo núcleo**, para que en el procesamiento de interrupción pueda haber **instrucciones privilegiadas**
- 4 **Ejecuta la rutina** de atención a la interrupción

Cambio de modo dentro del proceso

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

Creación de procesos

Cambio de proceso

Llamadas al sistema

Cambio de contexto

¿Qué sucede cuando  
llega una  
interrupción?

Después de atender  
una interrupción

Cambio de modo

Secuencia en un  
cambio de proceso

Ejecución del SO

# Manipulación de procesos

## Después de atender una interrupción

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

Creación de procesos

Cambio de proceso

Llamadas al sistema

Cambio de contexto

¿Qué sucede cuando  
llega una  
interrupción?

**Después de atender  
una interrupción**

Cambio de modo

Secuencia en un  
cambio de proceso

Ejecución del SO

### Dos posibilidades:

- El sistema operativo determina que el proceso actual debe continuar con su ejecución  $\Rightarrow$  Se ha interrumpido un proceso de usuario para tratar una rutina del S.O. y continua con el mismo proceso.

Solo cambio de modo: **usuario  $\Rightarrow$  núcleo  $\Rightarrow$  usuario**

- El S.O. determina que se debe realizar un cambio de proceso  $\Rightarrow$  El control pasa a la rutina de sistema de cambio de proceso.

Hay cambio de proceso



# Manipulación de procesos

## Cambio de modo

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

Creación de procesos

Cambio de proceso

Llamadas al sistema

Cambio de contexto

¿Qué sucede cuando  
llega una  
interrupción?

Después de atender  
una interrupción

**Cambio de modo**

Secuencia en un  
cambio de proceso

Ejecución del SO

- El cambio de modo **no implica** cambio del proceso que se esté ejecutando
- Se puede cambiar de modo, ejecutar instrucciones privilegiadas en modo núcleo y volver a ejecutar instrucciones del mismo proceso y por tanto:
  - No es necesario hacer cambio completo de proceso
  - Basta con guardar información del estado del procesador (contexto)

# Manipulación de procesos

## Secuencia en un cambio de proceso

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

Creación de procesos

Cambio de proceso

Llamadas al sistema

Cambio de contexto

¿Qué sucede cuando  
llega una  
interrupción?

Después de atender  
una interrupción

Cambio de modo

Secuencia en un  
cambio de proceso

Ejecución del SO

- ➊ **Salva el contexto** de P1 (BCP)
- ➋ Asigna al registro CP la dirección de **comienzo** de programa **atención a la interrupción de cambio de proceso**
- ➌ Cambia a **modo núcleo**
- ➍ **Ejecuta la rutina** de atención a la interrupción
  - ➊ **Actualiza el bloque de control del proceso** que está en estado de ejecución (nuevo estado: listo/bloqueado/...)
  - ➋ **Mueve** el bloque de control del proceso **a la cola apropiada** (listos/bloqueados/)
  - ➌ **Selecciona otro proceso** para su ejecución P2.
  - ➍ **Actualiza el BCP** seleccionado (nuevo estado: ejecución)
  - ➎ **Actualiza las estructuras de datos** de la gestión de memoria (para traducción de direcciones)
  - ➏ **Restaura el contexto del proceso** seleccionado (actualizar registros del procesador), **incluyendo modo (usuario)**



# Manipulación de procesos

## Ejecución del SO

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

Creación de procesos

Cambio de proceso

Llamadas al sistema

Cambio de contexto

¿Qué sucede cuando  
llega una  
interrupción?

Después de atender  
una interrupción

Cambio de modo

Secuencia en un  
cambio de proceso

Ejecución del SO

- ① Núcleo fuera de todo proceso:
  - Ejecuta el núcleo del sistema operativo fuera de cualquier proceso.
  - El código del sistema operativo se ejecuta como una entidad separada que opera en modo privilegiado.
- ② Ejecución dentro de los procesos de usuario:
  - Software del sistema operativo en el contexto de un proceso de usuario.
  - Un proceso se ejecuta en modo privilegiado cuando se ejecuta el código del sistema operativo.
- ③ Sistema operativo basado en procesos:
  - Las funciones más importantes del núcleo se organizan en procesos separados.
  - Útil en un entorno de multiprocesador o de varios computadores.

# Manipulación de procesos

## Ejecución del SO

Procesos e hilos

Generalidades

Modelos de estados  
del proceso

Estructuras de control

Imagen de un proceso

Bloque de control de  
proceso

Modos de ejecución

Manipulación de  
procesos

Creación de procesos

Cambio de proceso

Llamadas al sistema

Cambio de contexto

¿Qué sucede cuando  
llega una  
interrupción?

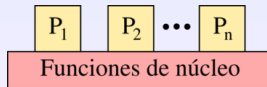
Después de atender  
una interrupción

Cambio de modo

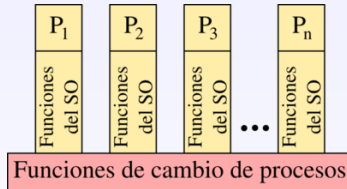
Secuencia en un  
cambio de proceso

Ejecución del SO

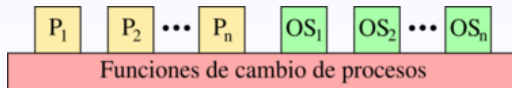
- 1 Núcleo fuera de todo proceso



- 2 Ejecución dentro de los procesos de usuario



- 3 Sistema operativo basado en procesos





Procesos e hilos

Estados

Ejecución del Sistema  
Operativo UNIX

Procesos en UNIX

Creación de procesos  
en UNIX

Jerarquía de procesos

Comunicación entre  
procesos

Llamadas al sistema

## Parte II

# Procesos en UNIX

## Estados

Ejecución del Sistema  
Operativo UNIX

Procesos en UNIX

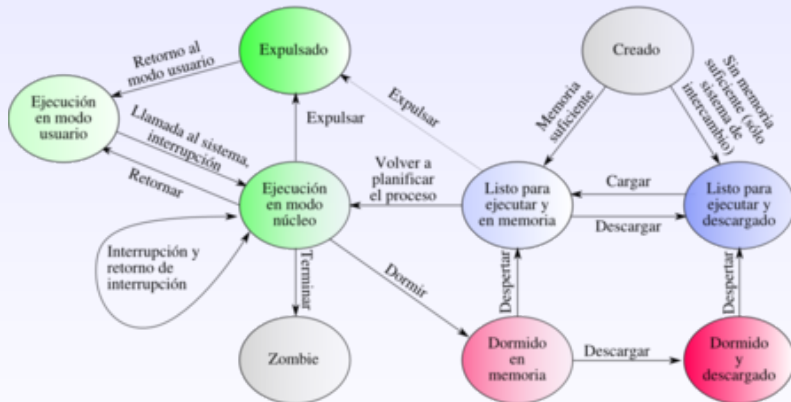
Creación de procesos  
en UNIX

Jerarquía de procesos

Comunicación entre  
procesos

Llamadas al sistema

Diagrama de transición de estados de UNIX



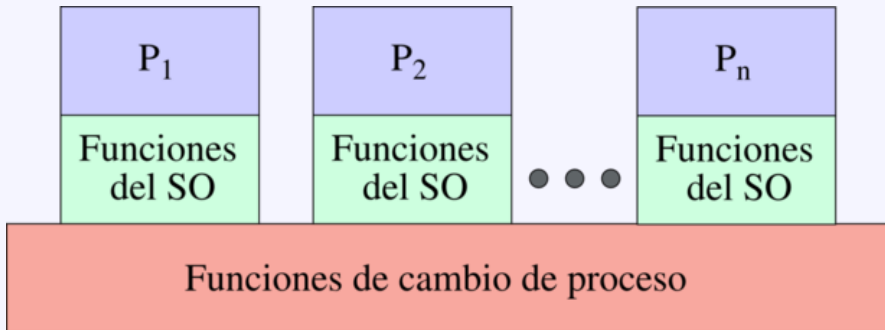
La expulsión de un proceso en ejecución en modo núcleo solo puede ocurrir justo cuando va a volver a ejecutar en modo usuario (p.ej. al elegir otro de mayor prioridad).

- **Ejecución en modo de usuario:** Ejecutando en modo de usuario.
- **Ejecución en modo del núcleo:** Ejecutando en modo de núcleo.
- **Listo para ejecutar y en memoria:** Listo para ejecutar tan pronto como el núcleo lo planifique.
- **Dormido y en memoria:** Incapaz de ejecutar hasta que se produzca un suceso; el proceso está en memoria principal.
- **Listo para ejecutar y descargado:** El proceso está listo para ejecutar, pero se debe cargar el proceso en memoria principal antes de que el núcleo pueda planificarlo para la ejecución.
- **Dormido y descargado:** El proceso está esperando un suceso y ha sido expulsado al almacenamiento secundario.
- **Expulsado:** El proceso retorna del modo del núcleo al modo de usuario, pero el núcleo lo expulsa y realiza un cambio de contexto para planificar otro proceso.
- **Creado:** El proceso está recién creado y aún no está listo para ejecutar.
- **Zombie:** El proceso ya no existe, pero deja un registro para que lo recoja el proceso padre.

# Ejecución del Sistema Operativo UNIX

## Características del SO en UNIX

- En UNIX la mayor parte del sistema operativo se ejecuta como procesos de usuario.
- Existen usuarios privilegiados.



# Procesos en UNIX

Estados

Ejecución del Sistema  
Operativo UNIX

**Procesos en UNIX**

Creación de procesos  
en UNIX

Jerarquía de procesos

Comunicación entre  
procesos

Llamadas al sistema

## Características

- Un proceso UNIX tiene dos áreas principales de memoria: ejecución y control.
- El área de ejecución es el programa (texto) más las zonas de memoria reservadas para la pila y los datos. Está en la zona de memoria virtual asignada al proceso.
- El área de control
  - contiene los bloques de control que conserva la información acerca del proceso. Esta zona la gestiona el núcleo del SO.
  - tiene información que debe residir en memoria continuamente e información que puede ser enviada a disco con el área de ejecución cuando el planificador de procesos decida.

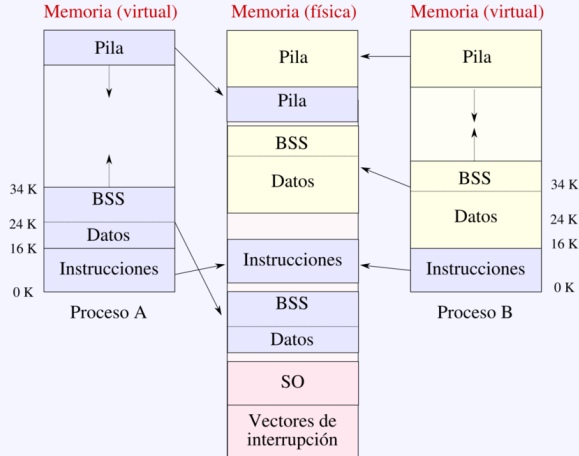
# Creación de procesos en UNIX

- Un proceso se crea mediante una llamada a sistema **fork()**.
- Efectos del fork:
  - 1 Crea una nueva entrada (estructura de proceso) en la tabla de procesos, en la que da un nuevo ID al proceso hijo.
  - 2 Asigna memoria y copia en ella los segmentos de datos y la pila del padre, junto a la cual copia la estructura de usuario.
  - 3 Pone al hijo en estado Listo para ejecutar.
  - 4 El código del padre y del hijo son idénticos salvo el valor de retorno del fork.
  - 5 El hijo recibe 0 como retorno del fork.
  - 6 El padre recibe el pid del hijo como retorno del fork.



# Creación de procesos en UNIX

## Pila de usuario, datos e instrucciones

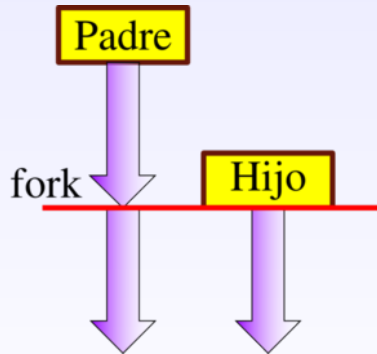


B hijo de A (instrucciones iguales). BSS: segmento de datos para variables estáticas

# Creación de procesos en UNIX

## Ejemplo de uso de fork

```
1  #include <unistd.h>
2
3  main()
4  {
5      int var=2;
6      int id;
7
8      id=fork();
9      switch(id)
10     {
11         case -1: /*error*/ break;
12         case 0: /*hijo*/ var++; break;
13         default: /*padre*/ var--; break;
14     }
15     printf("Var_=_ %d\n",var);
16 }
```



# Creación de procesos en UNIX

## Proceso de creación

- Asignar una nueva entrada a la tabla principal de procesos.
- Asignar espacio para la imagen del proceso.
- Inicializar el bloque de control de proceso.
- Establecer los enlaces apropiados
  - Ej: Añadir un proceso nuevo a una lista enlazada que se utiliza como cola de planificación.
- Crear o ampliar otras estructuras de datos
  - Ej: Mantener un archivo de contabilidad (tiempo ejecución del hijo igual a cero), actualizar la tabla de gestión de memoria...

# Creación de procesos en UNIX

## Zombie

Un hijo termina y el padre no recoge el código de finalización del hijo.

### Características

- Un proceso zombie no tiene ni área de código, ni área de datos, ni pila, ni estructura ..., pero afecta a la tabla de procesos del sistema, pues sigue ocupando una entrada de la tabla.
- Los zombies permanecen en el sistema hasta que alguien los espere, `waitpid()`.
- En los UNIX modernos el SO se encarga de “eliminar” a los zombies.

# Creación de procesos en UNIX

Procesos e hilos

Estados

Ejecución del Sistema  
Operativo UNIX

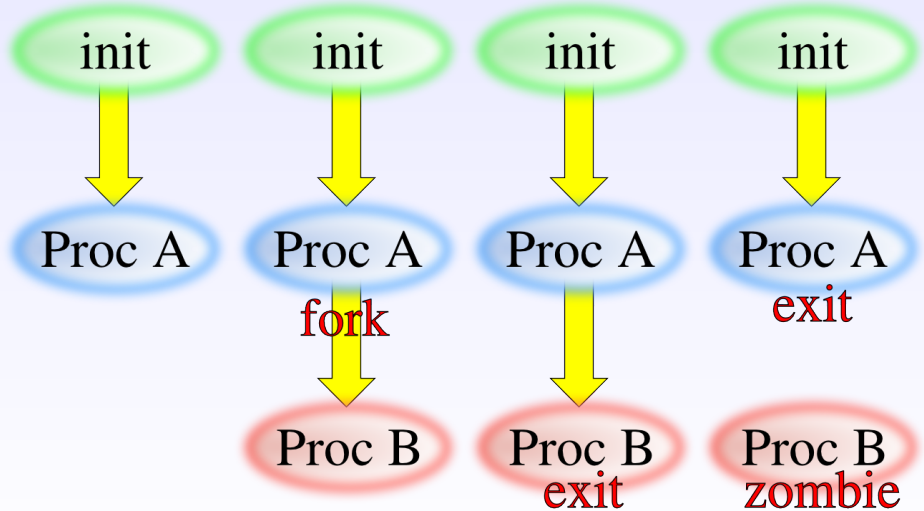
Procesos en UNIX

Creación de procesos  
en UNIX

Jerarquía de procesos

Comunicación entre  
procesos

Llamadas al sistema



# Creación de procesos en UNIX

## Huérfano

El proceso padre:

- muere antes de que el hijo haya terminado
- termina sin recoger/esperar al proceso hijo *wait()*

## Características

- Un hijo huérfano es adoptado por el proceso init del sistema.
- El proceso init (PID=1) es el ancestro de todos los procesos que se crean con posterioridad.
- El proceso con PID=0, creado por el núcleo cuando arranca el S.O., es el único que no se crea por una llamada a *fork()*.
  - Después de la llamada a *fork()* se convierte en el proceso intercambiador  
⇒ Gestión de la memoria virtual.

# Creación de procesos en UNIX

Procesos e hilos

Estados

Ejecución del Sistema  
Operativo UNIX

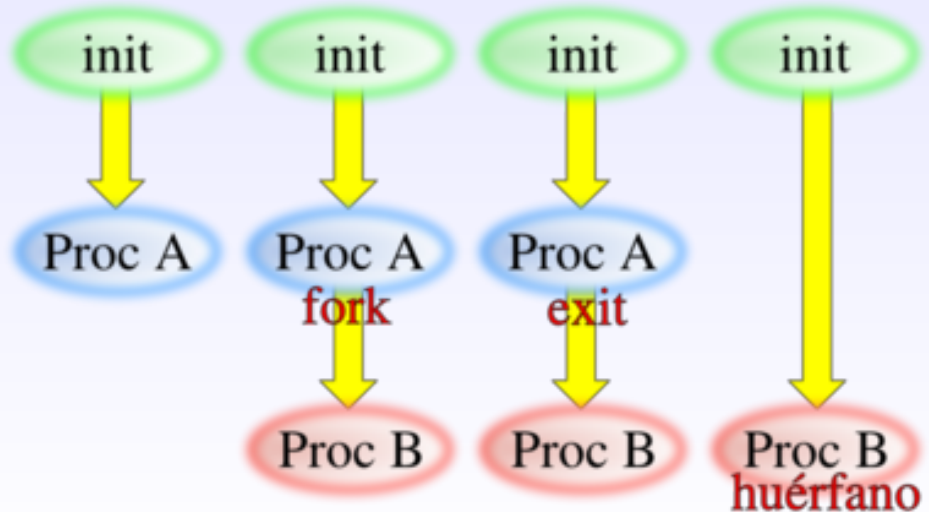
Procesos en UNIX

**Creación de procesos  
en UNIX**

Jerarquía de procesos

Comunicación entre  
procesos

Llamadas al sistema

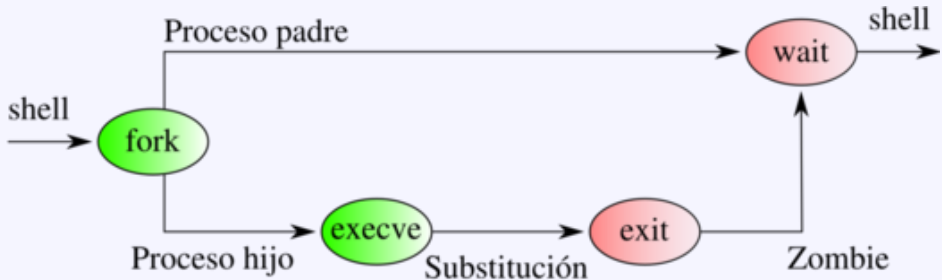


# Creación de procesos en UNIX

Otras dos funciones importantes en la creación de procesos son:

- **execve**: es realmente una familia completa de funciones. Son usadas por un proceso para cargar un nuevo ejecutable binario en el espacio virtual de memoria del proceso que hace la llamada.
- **vfork**: es igual que fork, pero no copia los datos ni la pila.

Secuencia normal en la creación de un proceso





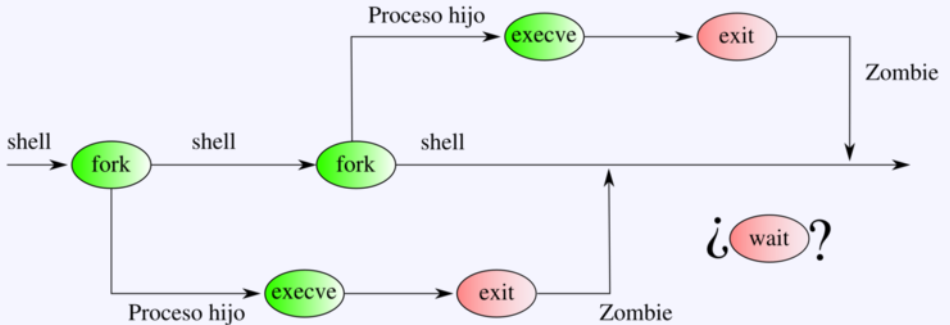
# Creación de procesos en UNIX

## Ejemplo de uso de execve

```
1  main() /* programa wait.exe */
2  {
3      int pid, status, hijoPid;
4      printf(" Soy_el_proceso_padre_y_mi_Pid_es: %d\n", getpid());
5      pid = fork();
6      if(pid == -1) exit(EXIT_FAILURE);
7      if(pid!=0) {
8          printf(" Soy_el_proceso_padre_con_PID_ %d_y_PPID_ %d\n",
9                 getpid(),getppid());
10         hijoPid= wait(&status);
11         printf(" Un_hijo_con_PID_ %d_ha_terminado_con_codigo_de_salida_ %d\n",
12                hijoPid,status>>8);
13     }
14     else {
15         printf(" Soy_el_hijo_con_PID_ %d_y_PPID_ %d\n",
16                getpid(),getppid());
17         exit(42);
18     }
19     printf(" El_proceso_de_PID_ %d_termina\n", getpid());
20 }
```

# Creación de procesos en UNIX

## Procesos en background



# Jerarquía de procesos

Procesos e hilos

Estados

Ejecución del Sistema  
Operativo UNIX

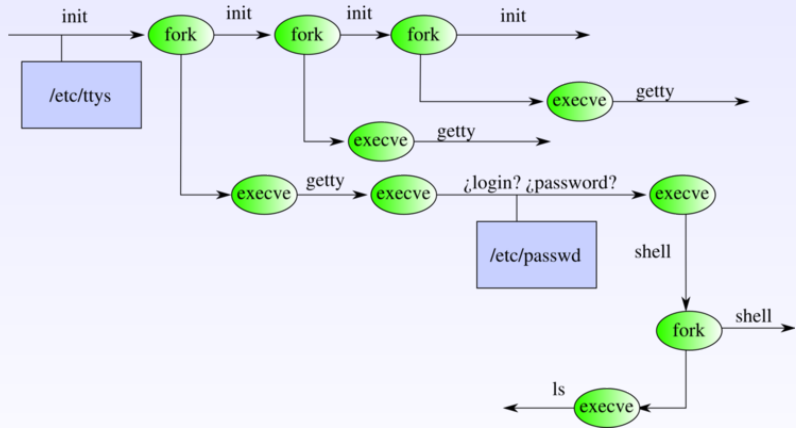
Procesos en UNIX

Creación de procesos  
en UNIX

**Jerarquía de procesos**

Comunicación entre  
procesos

Llamadas al sistema



`init` crea un hijo para cada terminal, que ejecuta `getty` (prompt de login). Cuando usuario entra, se ejecuta `shell`, que espera comandos y lanza un hijo (`fork+exec`) por comando.

# Jerarquía de procesos

## Identificadores de proceso

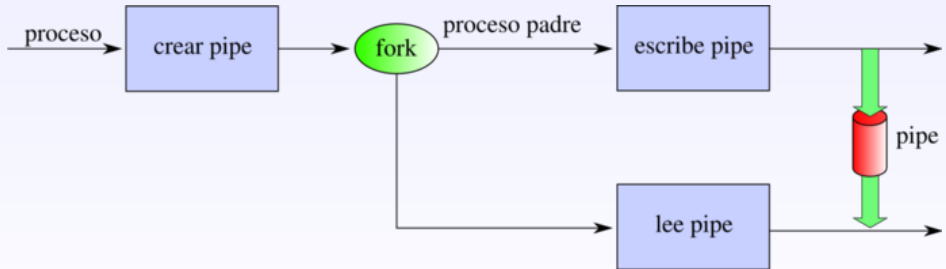
- **uid:** Identificador de usuario real es el del usuario que ha abierto la sesión (leído por login en /etc/passwd).
- **gid:** Identificador de grupo es el del grupo al que pertenece el usuario que inició la sesión.
- **euid:** Identificador de usuario efectivo es el del propietario del fichero ejecutado. Se aplica si el programa cargado usando execve tiene activado el bit setuid.
- **egid:** Identificador de grupo efectivo. Se aplica si el programa cargado usando execve tiene activado el bit setgid.

## Llamadas para obtener los identificadores de un sistema

- getuid (uid)
- geteuid (euid)
- getgid (gid)
- getegid (egid)

# Comunicación entre procesos

- Los procesos se pueden comunicar mediante tuberías (**pipes**) que deben ser creadas antes de la ejecución del fork.
- A los pipes se accede a través de un descriptor de fichero, como cualquier otro “fichero”. Tienen tamaño variable.



# Comunicación entre procesos

## Ejemplo de uso de pipes

```
1  #define LEER 0
2  #define ESCRIBIR 1
3
4  char *frase = "Envia_esto_a_traves_de_un_tubo";
5
6  main() /* programa escribe.exe */
7  {
8      int fd[2], bytesLeidos;
9      char mensaje[100];
10
11     if(pipe(fd) == 0) {
12         if(fork() == 0) {
13             close(fd[LEER]);
14             write(fd[ESCRIBIR],frase,strlen(frase) +1);
15             close(fd[ESCRIBIR]);
16         } else {
17             close(fd[ESCRIBIR]);
18             bytesLeidos = read(fd[LEER],mensaje,100);
19             printf(" Leidos_ %d_bytes:_ %s\n",bytesLeidos,mensaje);
20             close(fd[LEER]);
21         }
22     } else exit(EXIT_FAILURE);
23 }
```

# Comunicación entre procesos

## Señales

- Son recursos (similares a las interrupciones) para atender algunas circunstancias excepcionales.
- Origen: teclado, error en el proceso, eventos, llamada a kill.
- Señales principales:
  - **SIGINT**: interrupción desde teclado (normalmente ^C). Detiene un comando antes de que termine su ejecución.
  - **SIGQUIT**: desde teclado (^bs). Detiene ejecución y vuelca el contenido de la memoria en un fichero core.
  - **SIGSEGV**: Acceso no válido a memoria.
  - **SIGKILL**: Termina con un proceso (no se puede ignorar o atrapar).
- Las señales pueden ser ignoradas o atendidas por una rutina en el proceso de usuario.
- Las señales pueden perderse y no se acumulan.

# Llamadas al sistema

- **pid = fork():** crea un proceso hijo, que es igual al padre. Devuelve pid=0 en el hijo y el pid del hijo al padre.
- **s = waitpid(pid,&status,opts):** ejecutada por el proceso padre, le pone en espera hasta la terminación del hijo. El proceso padre puede esperar a la terminación de un hijo o de cualquiera (pid=-1) y recibe el estado del proceso saliente (status). El padre puede quedar bloqueado o regresar si ningún proceso ha terminado (opt).
- **s = execve(name,argv,envp):** reemplaza el código del proceso llamante por un nuevo código. Ejecuta el proceso name, con un puntero de argumentos argv y un puntero a entorno envp (nombre=valor).
- **exit(status):** es enviada por un proceso que termina su ejecución. Devuelve el estado de conclusión y salida del proceso hijo al padre, que está esperando mediante un waitpid.



# Llamadas al sistema

## Estados

## Ejecución del Sistema Operativo UNIX

## Procesos en UNIX

## Creación de procesos en UNIX

## Jerarquía de procesos

## Comunicación entre procesos

## Llamadas al sistema

- **`s=sigaction(sig,&act,&oact)`**: manifiesta la disponibilidad del proceso frente a señales, y especifica la acción a llevar a cabo como respuesta. `sig` indica qué señal se atiende; `act` apunta a una estructura que contiene un puntero al procedimiento de manejo de la señal; `oact` apunta a una estructura en la que el sistema devuelve información acerca del manejo de señales, por si debe ser restaurado.
- **`s = kill(pid,sig)`**: envía una señal a otro proceso relacionado con él.
- **`s = alarm(segundos)`**: genera una señal después de un periodo de tiempo especificado.
- **`pause()`**: interrumpe la ejecución del proceso hasta la llegada de la siguiente señal.



Procesos e hilos

Procesos e Hilos

Hilos

Hilos a nivel de  
usuario

Hilos a nivel de núcleo

Aproximaciones  
combinadas

Relación Hilo-Proceso

# Parte III

## Hilos



# Procesos e Hilos

## Características

Procesos e hilos

Procesos e Hilos

Características

Sistema Multihilo

Diferencia

Proceso-Hilo

Beneficios de los hilos

Uso de los hilos

Hilos

Hilos a nivel de usuario

Hilos a nivel de núcleo

Aproximaciones combinadas

Relación Hilo-Proceso

- **Unidad de propiedad de los recursos:**
  - Proceso necesita espacio de direcciones virtuales para mantener su imagen
  - Proceso tiene asignados recursos en un momento determinado
- **Unidad de expedición:**
  - Procesos sigue un camino de ejecución que puede ser intercalada con la de otros procesos
- Estas dos características son tratadas de **manera independiente** por el sistema operativo:
  - La unidad de expedición se conoce como **hilo**
  - La unidad de propiedad de los recursos se conoce como **proceso o tarea**

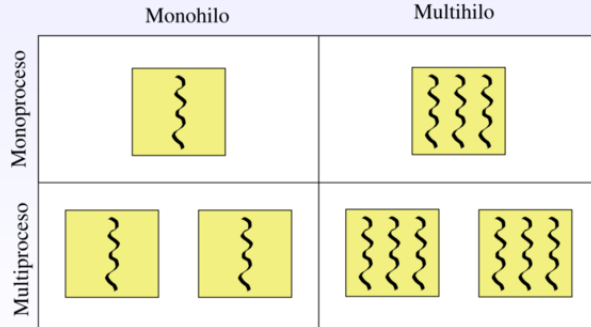
# Procesos e Hilos

## Sistema Multihilo

Procesos e hilos

**Sistema operativo que mantiene varios hilos de ejecución dentro de un mismo proceso**

- MS-DOS soporta un solo hilo.
- UNIX soporta múltiples procesos de usuarios, pero sólo un hilo por proceso.
- Windows 2000, Solaris, Linux, Mach, y OS/2 soportan múltiples hilos.



Procesos e Hilos

Características

**Sistema Multihilo**

Diferencia

Proceso-Hilo

Beneficios de los hilos

Uso de los hilos

Hilos

Hilos a nivel de usuario

Hilos a nivel de núcleo

Aproximaciones combinadas

Relación Hilo-Proceso

# Procesos e Hilos

## Diferencia Proceso-Hilo

Procesos e hilos

Procesos e Hilos

Características

Sistema Multihilo

Diferencia

Proceso-Hilo

Beneficios de los  
hilos

Uso de los hilos

Hilos

Hilos a nivel de  
usuario

Hilos a nivel de núcleo

Aproximaciones  
combinadas

Relación Hilo-Proceso

### Proceso

- Tiene **un espacio de direcciones virtuales**, que contiene la imagen del proceso.
- **Acceso protegido a los procesadores**, a otros procesos, archivos y a recursos de E/S.
- Tiene un hilo o más.

### Hilo

- Bloque de control de hilo: posee un **estado de ejecución** (Ejecución, Listo, etc.) y el **contexto del procesador** se salva cuando no está ejecutando
- Tiene una **pila de ejecución**: almacenamiento para las **variables locales**
- **Acceso a la memoria y a los recursos del proceso**, compartidos con todos los hilos del mismo



# Procesos e Hilos

## Beneficios de los hilos

Procesos e hilos

Procesos e Hilos

Características  
Sistema Multihilo  
Diferencia  
Proceso-Hilo  
**Beneficios de los hilos**  
Uso de los hilos

Hilos

Hilos a nivel de usuario

Hilos a nivel de núcleo

Aproximaciones combinadas

Relación Hilo-Proceso

- Se tarda menos tiempo en crear un nuevo hilo en un proceso existente.
- Se tarda menos tiempo en terminar un hilo que un proceso.
- Se tarda menos tiempo en cambiar entre dos hilos de un mismo proceso.
- Puesto que los hilos de un mismo proceso comparten recursos (memoria, archivos, etc.), pueden comunicarse entre sí sin invocar al núcleo → coordinación más rápida → mayor eficiencia.

# Procesos e Hilos

## Uso de los hilos

Procesos e hilos

### Procesos e Hilos

Características  
Sistema Multihilo  
Diferencia  
Proceso-Hilo  
Beneficios de los  
hilos

### Uso de los hilos

#### Hilos

Hilos a nivel de  
usuario

Hilos a nivel de núcleo

Aproximaciones  
combinadas

Relación Hilo-Proceso

En un sistema multihilo y monousuario los hilos permiten:

- Trabajo interactivo y en segundo plano.
  - Ej: calcular – recibir datos de teclado (bloques)
- Procesamiento asíncrono.
  - Ej: cada X tiempo copia de seguridad. No necesario comprobaciones externas de tiempo, ni coordinación entre E/S, porque el hilo se ocupa de ello
- Aceleración de la ejecución.
  - Ej: procesar lotes – leer siguiente
- Estructuración modular de los programas.
  - Ej: varias actividades a la vez y/o varios dispositivos

# Hilos

## Especificaciones

### Estados de un hilo

- **Creación:** Se crea un nuevo hilo
  - Crear un proceso = crear 1 primer hilo
  - Un hilo puede crear otros hilos
  - El nuevo hilo tiene su **propio contexto y espacio de pila**. Pasa a la cola de listos.
- **Bloqueo**
  - El hilo necesita **esperar suceso**
  - **Se guarda el contexto** del hilo (registros, contador de programa, puntero de pila)
  - No tienen por qué bloquearse el resto de los hilos del mismo proceso
- **Desbloqueo:** El suceso ocurre, el hilo pasa a cola de listos
- **Terminación:** Se liberan su contexto (del hilo) y sus pilas.
- La suspensión de un proceso implica la suspensión de todos los hilos de un proceso, puesto que todos comparten el mismo espacio de direcciones.
- La terminación de un proceso supone terminar con todos los hilos dentro de dicho proceso.



# Hilos

## Llamada a RPC

Procesos e hilos

Procesos e Hilos

Hilos

Especificaciones

**Llamada a RPC**

Tipos de hilos

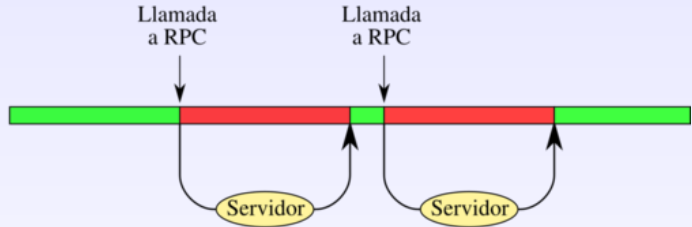
Hilos a nivel de  
usuario

Hilos a nivel de núcleo

Aproximaciones  
combinadas

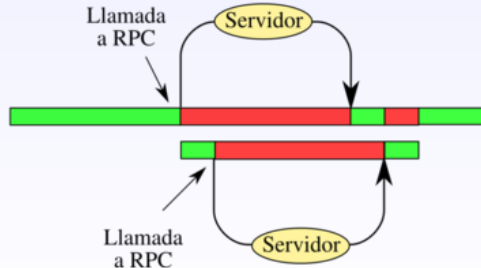
Relación Hilo-Proceso

Proceso A



Hilo A

Hilo B



# Hilos

## Tipos de hilos

### Procesos e Hilos

#### Hilos

Especificaciones  
Llamada a RPC  
**Tipos de hilos**

#### Hilos a nivel de usuario

#### Hilos a nivel de núcleo

#### Aproximaciones combinadas

#### Relación Hilo-Proceso

### Hilos a nivel de usuario

- La aplicación (biblioteca de hilos) realiza todo el trabajo de gestión de hilos.
- El núcleo no tiene conocimiento de la existencia de hilos.
- Biblioteca de hilos - código para ...
  - ... crear y destruir hilos
  - ... intercambiar mensajes y datos entre hilos
  - ... planificar ejecución de hilos
  - ... salvar y restaurar el contexto de los hilos



# Hilos

## Tipos de hilos

Procesos e hilos

Procesos e Hilos

Hilos

Especificaciones  
Llamada a RPC  
Tipos de hilos

Hilos a nivel de  
usuario

Hilos a nivel de núcleo

Aproximaciones  
combinadas

Relación Hilo-Proceso

### Hilos a nivel de núcleo

- W2K, Linux y OS/2 son ejemplos de este tipo de aplicación.
- El núcleo mantiene la información de contexto del proceso y de los hilos.
- La planificación se realiza directamente en función de los hilos.

### Aproximaciones combinadas

- Un ejemplo es Solaris.
- La creación de hilos se realiza en el espacio de usuario.
- La planificación y sincronización de los hilos se realiza en el espacio de usuario.

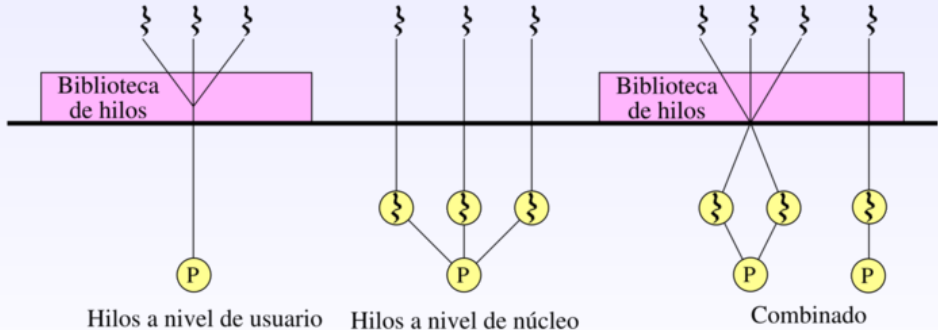
# Hilos

## Tipos de hilos

Procesos e hilos

 Hilo a nivel de núcleo    
  Hilo a nivel de usuario    
  Proceso

Espacio de usuario



Espacio de núcleo

Procesos e Hilos

Hilos

Especificaciones

Llamada a RPC

Tipos de hilos

Hilos a nivel de usuario

Hilos a nivel de núcleo

Aproximaciones combinadas

Relación Hilo-Proceso

# Hilos a nivel de usuario

## Cambios de estado

Procesos e hilos

Procesos e Hilos

Hilos

Hilos a nivel de  
usuario

Cambios de estado

Ventajas y  
desventajas

Hilos a nivel de núcleo

Aproximaciones  
combinadas

Relación Hilo-Proceso



Proceso ejecutándose (hilo 2)  
El hilo 2 hace llamada al sistema  
*¿Qué ocurre?*

# Hilos a nivel de usuario

## Cambios de estado

Procesos e hilos

Procesos e Hilos

Hilos

Hilos a nivel de  
usuario

Cambios de estado

Ventajas y  
desventajas

Hilos a nivel de núcleo

Aproximaciones  
combinadas

Relación Hilo-Proceso



Proceso ejecutándose (hilo 2)  
El hilo 2 hace llamada al sistema



Pasa a modo núcleo  
El proceso bloqueado  
Estado de los hilos se mantiene  
(nivel usuario)

# Hilos a nivel de usuario

## Cambios de estado

Procesos e hilos

Procesos e Hilos

Hilos

Hilos a nivel de  
usuario

Cambios de estado

Ventajas y  
desventajas

Hilos a nivel de núcleo

Aproximaciones  
combinadas

Relación Hilo-Proceso



Proceso ejecutándose (hilo 2)

Interrupción de reloj

El tiempo para proceso A termina

**¿Qué ocurre?**

# Hilos a nivel de usuario

## Cambios de estado

Procesos e hilos

Procesos e Hilos

Hilos

Hilos a nivel de usuario

Cambios de estado

Ventajas y desventajas

Hilos a nivel de núcleo

Aproximaciones combinadas

Relación Hilo-Proceso



Proceso ejecutándose (hilo 2)  
Interrupción de reloj  
El tiempo para proceso A termina



Pasa a modo núcleo  
El proceso A pasa a listo  
El estado de los hilos (nivel usuario)  
se mantiene



# Hilos a nivel de usuario

## Cambios de estado

Procesos e hilos

Procesos e Hilos

Hilos

Hilos a nivel de  
usuario

Cambios de estado

Ventajas y  
desventajas

Hilos a nivel de núcleo

Aproximaciones  
combinadas

Relación Hilo-Proceso



Proceso ejecutándose (hilo 2)

El hilo 2 necesita que ocurra  
un suceso que depende del  
hilo 1

*¿Qué ocurre?*

# Hilos a nivel de usuario

## Cambios de estado

Procesos e hilos

Procesos e Hilos

Hilos

Hilos a nivel de  
usuario

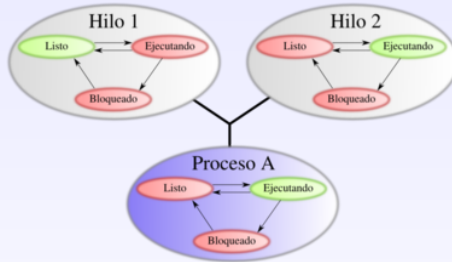
Cambios de estado

Ventajas y  
desventajas

Hilos a nivel de núcleo

Aproximaciones  
combinadas

Relación Hilo-Proceso



Proceso ejecutándose (hilo 2)  
El hilo 2 necesita que ocurra  
un suceso que depende del  
hilo 1



El hilo 2 se bloquea (nivel usuario)  
El hilo 1 pasa de estado listo a  
ejecutando.  
El proceso continua ejecutándose

# Hilos a nivel de usuario

## Ventajas y desventajas

### Ventajas

- El intercambio de hilos no necesita privilegios del modo núcleo
  - Gestión de hilos en el espacio de direcciones de usuario
  - Se evitan 2 cambios de modo: usuario → núcleo → usuario.
- Se puede realizar planificación específica a nivel de hilos.
- Los hilos se pueden ejecutar en cualquier S.O.
  - Biblioteca hilos = utilidades de aplicación

### Desventajas

- Las llamadas al sistema suelen ser bloqueantes en el S.O.
  - Un hilo ejecuta llamada al sistema → bloqueo del resto de hilos (todo el proceso)
- No se aprovechan las ventajas de multiprocesadores
  - Núcleo asigna el procesador a un proceso → sólo 1 hilo del proceso en ejecución
  - Los hilos a nivel de usuario son “invisibles” para el S.O.

# Hilos a nivel de núcleo

## Características

Procesos e hilos

Procesos e Hilos

Hilos

Hilos a nivel de  
usuario

Hilos a nivel de núcleo

Características

Ventajas y  
desventajas

Aproximaciones  
combinadas

Relación Hilo-Proceso

- El núcleo mantiene la información de contexto del proceso y de los hilos
  - Hilos visibles y gestionados por el núcleo
- La planificación se realiza en función de los hilos → hilos del mismo o distinto proceso compiten por el procesador
- Ejs: W2K, Linux y OS/2

# Hilos a nivel de núcleo

## Ventajas y desventajas

Procesos e Hilos

Hilos

Hilos a nivel de  
usuario

Hilos a nivel de núcleo

Características

**Ventajas y  
desventajas**

Aproximaciones  
combinadas

Relación Hilo-Proceso

### Ventajas

- Si hay varios procesadores, es posible ejecutar en paralelo hilos del mismo proceso
- Las propias funciones del núcleo pueden ser multihilo

### Desventajas

- Es necesario cambiar a modo núcleo para pasar de un hilo a otro (incluso con hilos de un mismo proceso)

# Aproximaciones combinadas

## Características

### Procesos e Hilos

### Hilos

### Hilos a nivel de usuario

### Hilos a nivel de núcleo

### Aproximaciones combinadas

### Características

### Relación Hilo-Proceso

- Los hilos se crean en el espacio de usuario
- La planificación y sincronización se realizan en el espacio de usuario
- Varios hilos de usuario se asocian con varios hilos a nivel de núcleo
  - De igual número o distinto, ajustable por el programador
  - Los puede planificar el núcleo
- Esquema combinado de planificación
- Ej.: Solaris

## Ventajas

- Hilos de un mismo proceso se pueden ejecutar en paralelo en varios procesadores
- Las llamadas al sistema bloqueantes no necesitan bloquear todo el proceso

# Relación Hilo-Proceso

Procesos e Hilos

Hilos

Hilos a nivel de  
usuario

Hilos a nivel de núcleo

Aproximaciones  
combinadas

Relación Hilo-Proceso

Hilos:Procesos	Descripción	Sistemas Ejemplo
1:1	Cada hilo de ejecución es un único proceso con sus propios recursos y espacio de direcciones	Implementaciones UNIX clásicas
M:1	Un proceso define un espacio de direcciones y unos recursos dinámicos propios. Pueden crearse varios hilos que ejecuten en dicho proceso.	Windows NT, Solaris, OS/2, OS/390, MACH
1:N	Un hilo puede emigrar del entorno de un proceso a otro. Esto permite que un hilo se pueda mover fácilmente entre sistemas distintos.	Ra (Clouds), Emerald
M:N	Combina los atributos de los casos M : 1 y 1 : M	TRIX