FUNCIONES PRINCIPALES DE UN SISTEMA OPERATIVO

- PROCESOS:
  - Necesidad y Definición
  - Creación/Terminación de procesos
  - Estados de procesos / Cambios
  - Jerarquía de procesos
  - Implementación de procesos en Linux
- SEÑALES
  - Distintos tipos de Señales
  - Manejador
  - Imprementación de señales en Linux

Procesos

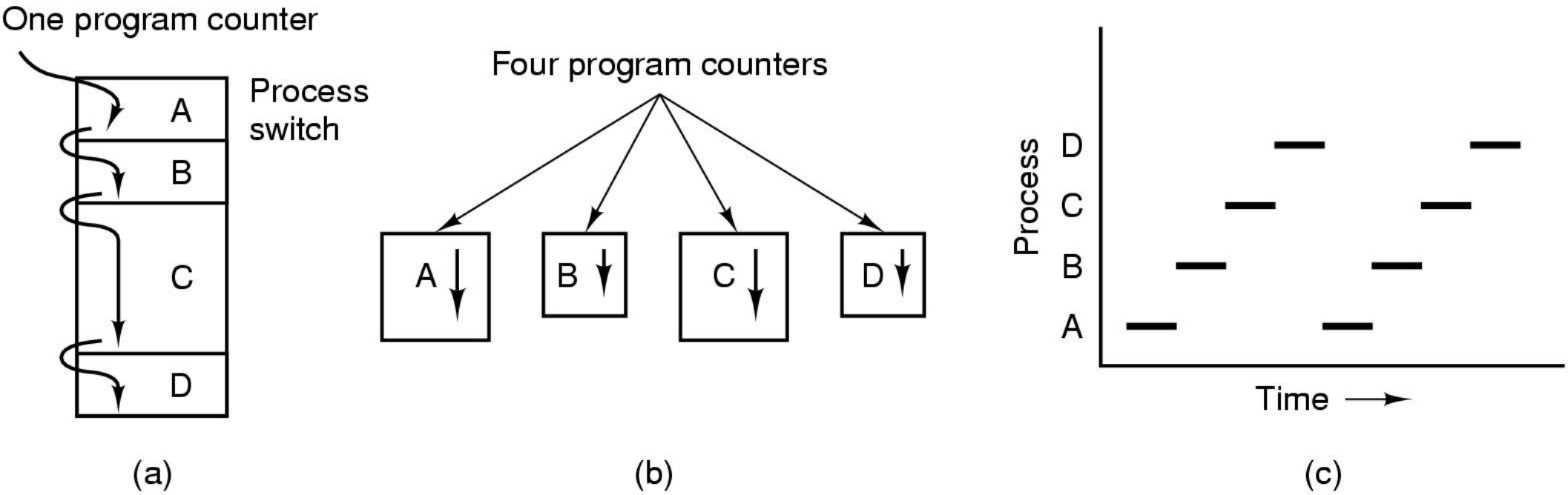
¿Necesidad de multiprocesamiento?

- Velocidad Procesador
- Velocidad Entrada Salida

- Pseudoparalelismo
  - Un solo procesador
  - Una instrucción por vez
  - ¿Cómo lo logro?

Procesos

Pseudoparalelismo



Programas DEBEN estar en memoria

Asignación de tiempos (Problemas con Real Time)

No programar basandose en supuestos de tiempo

Si se ejecuta nuevamente probablemente terminarán en distinto orden.

Procesos cont.

¿Qué es un proceso?

Programa (entidad pasiva) en ejecución

\+ valor del contador de programa

\+ valor de los registros

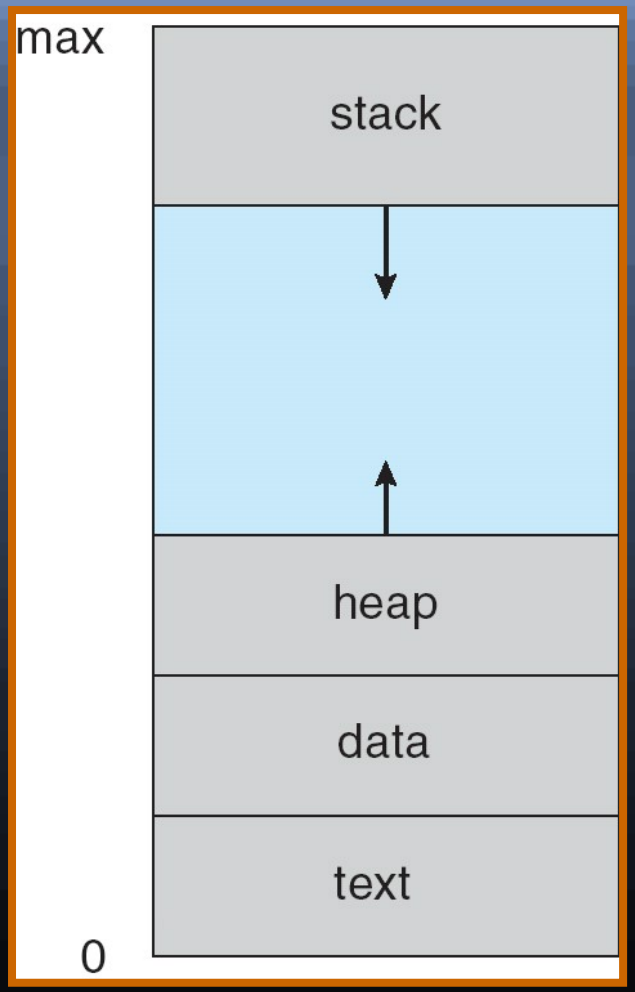
\+ pila del proceso (parámetros de funciones, direcciones de retorno, etc.)

PROCESO (entidad activa)

- Analogía

Procesos cont.

Ubicación del proceso en memoria:



Stack: argumentos para la función, direcciones de retorno

Heap: espacio para asignación dinámica de memoria (malloc())

Data: Variables globales, variables inicializadas

Text : código (binario) del programa

Creación de Procesos

¿Para qué creo procesos?

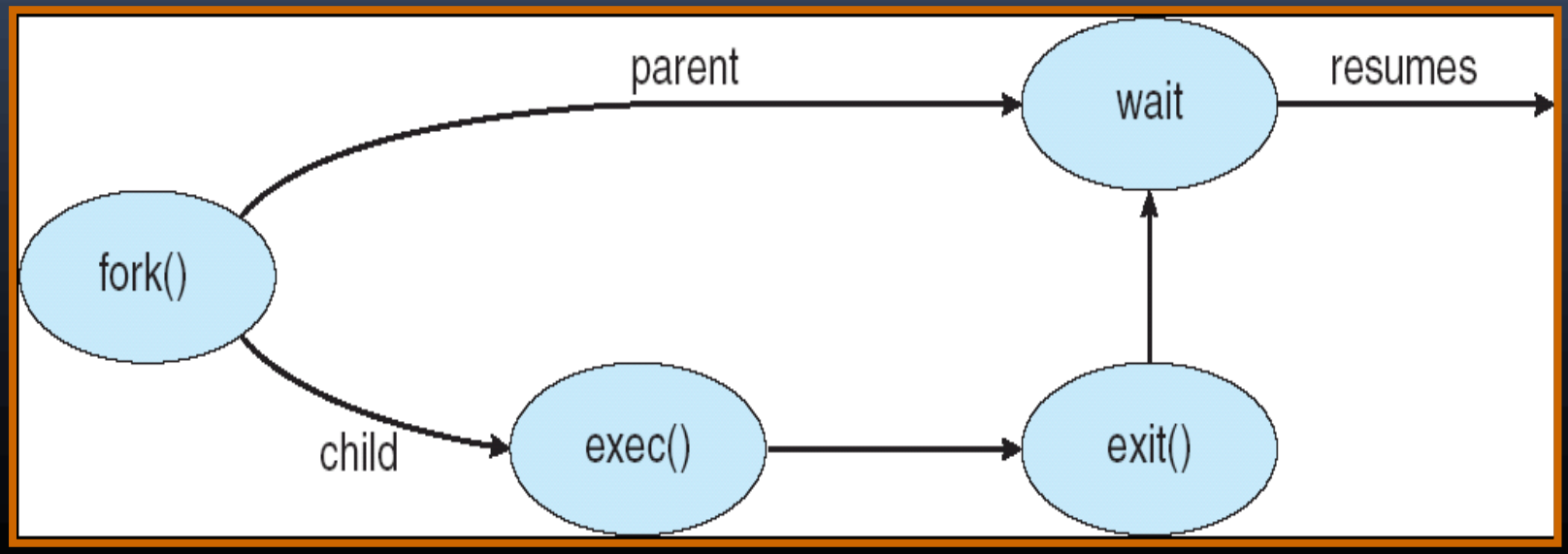
- Inicialización del Sistema
  - Init o systemd
  - Servicios
    - Background, foregraund (bg,fg, &, at)
- Proceso hace llamada a systema “fork()”
  - Crea un nuevo proceso … copia...
- Desde Interprete de comandos (o doble “click”)
  - Para sistemas interactivos
- Nuevos trabajos bach
  - Para sistemas de prosesamiento por lotes

Creación de Procesos con fork()

Alternativas Padre:

- Ejecuta concurrentemente con hijo
- Espera que finalize el hijo

- Alternativas Hijo:
  - Usa mismo programa y datos que padre
  - Carga un nuevo programa



Terminación de Procesos

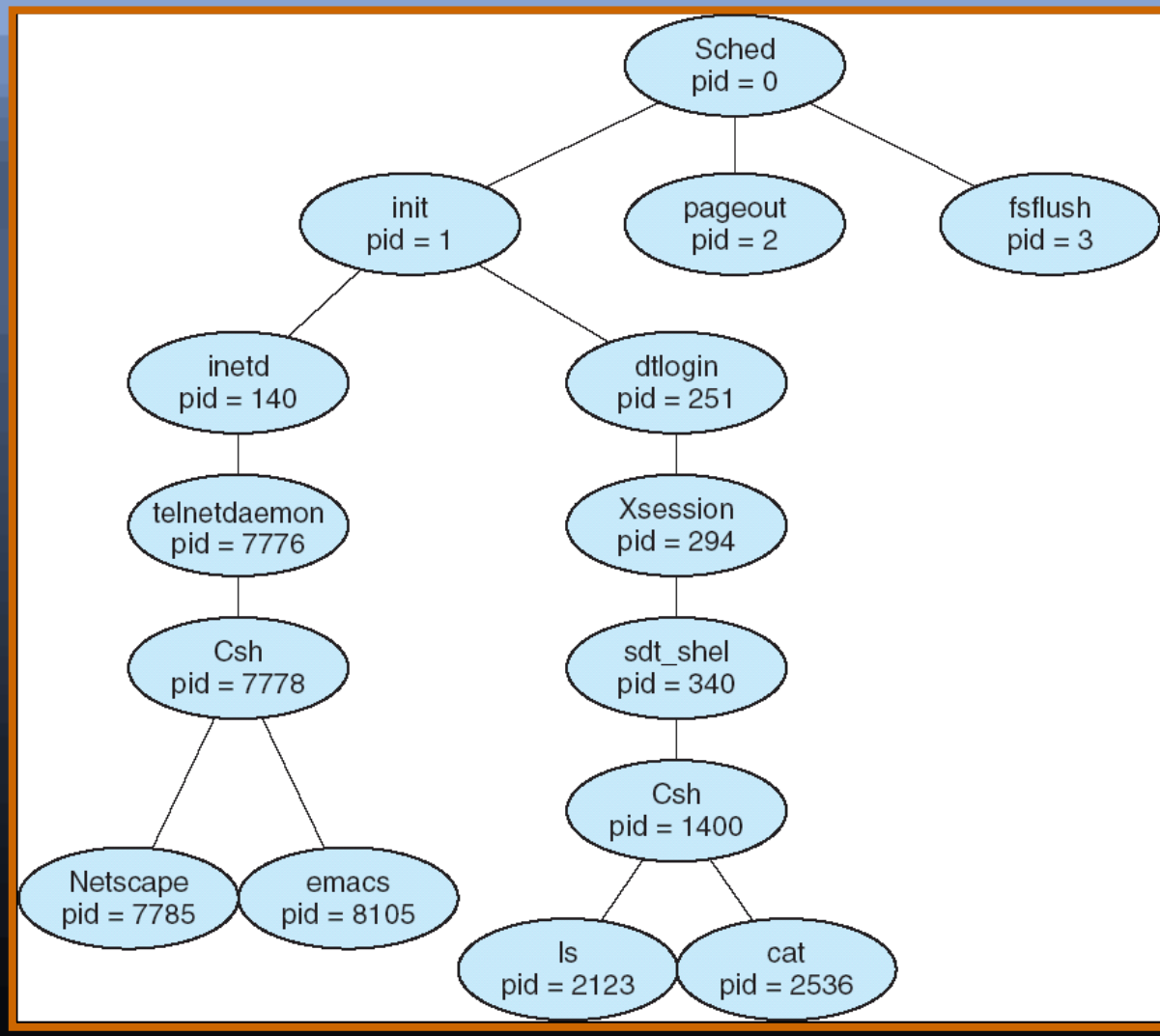
Tipos de terminación

- Terminación Normal
  - Tareas que realiza el SO
  - echo $?
- Terminación Anormal Voluntaria
  - echo $?
  - standard error
- Terminación Anormal Involuntaria
  - No hay mas memoria
  - Violación de Segmento
  - División por cero
- Recepción de una señal (Que es una señal?)
  - Comando kill, llamada a sistema “kill()”
  - Handler de señales, llamada a sistema “signal()”

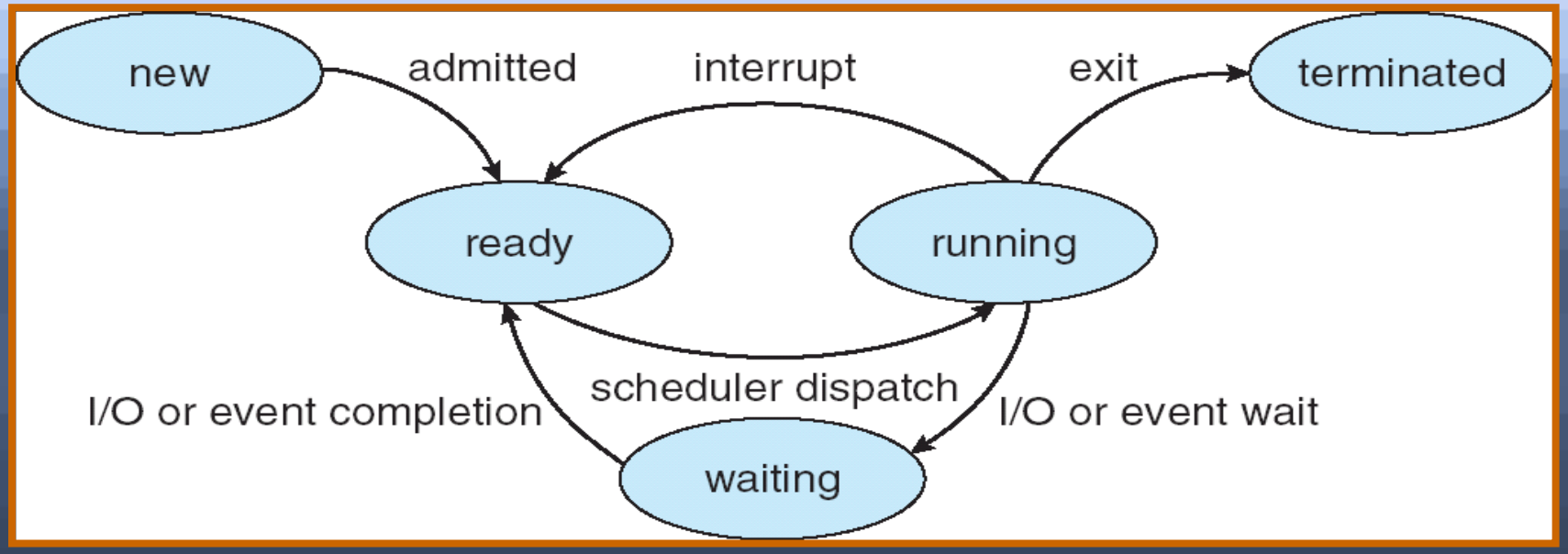
Jerarquía de Procesos

Relación entre procesos:

- Arbol de procesos
  - Comando pstree
  - Proceso Sched
  - Proceso init o systemd
- Llamadas a sistema:
  - “getpid()”
  - “getppid()”



Estado de los Procesos



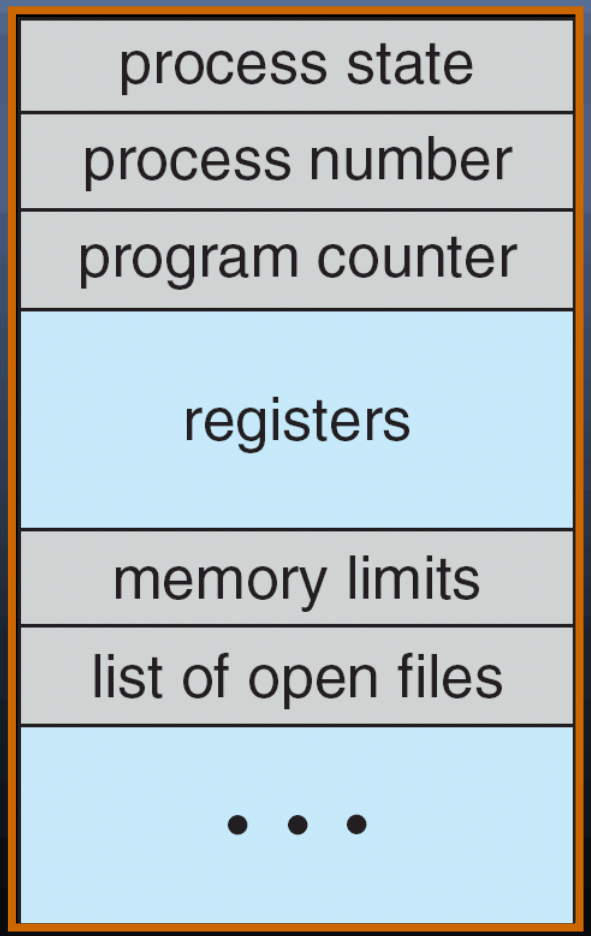
Diferencia entre los estados:

- ¿Cómo cambian de estado?
  - Scheduler .....que??
  - Interrupciones
  - Usuario puede forzar cambio de estado

Bloque de Control Procesos (PCB)

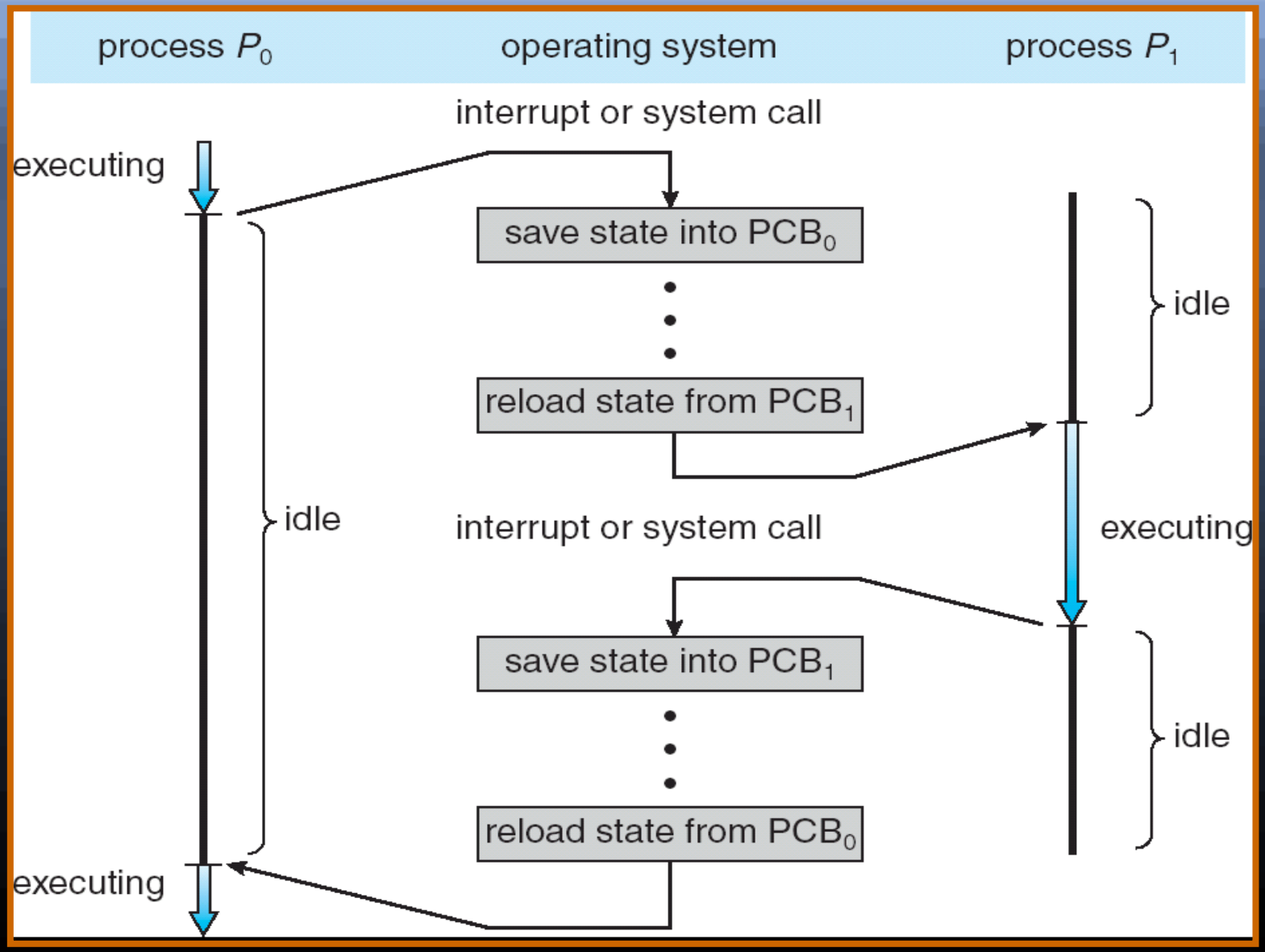
¿Cómo representa el Sistema Operativo los procesos???

- Número de Proceso (PID)
- Estado
- PC (contador de programa)
- Registros (generales, puntero de pila)
- Espacio de memoria que ocupa
- Descriptores de archivos abiertos
- Usuario / grupo
- etc.



Conmutación entre Procesos

(cambio de contexto)



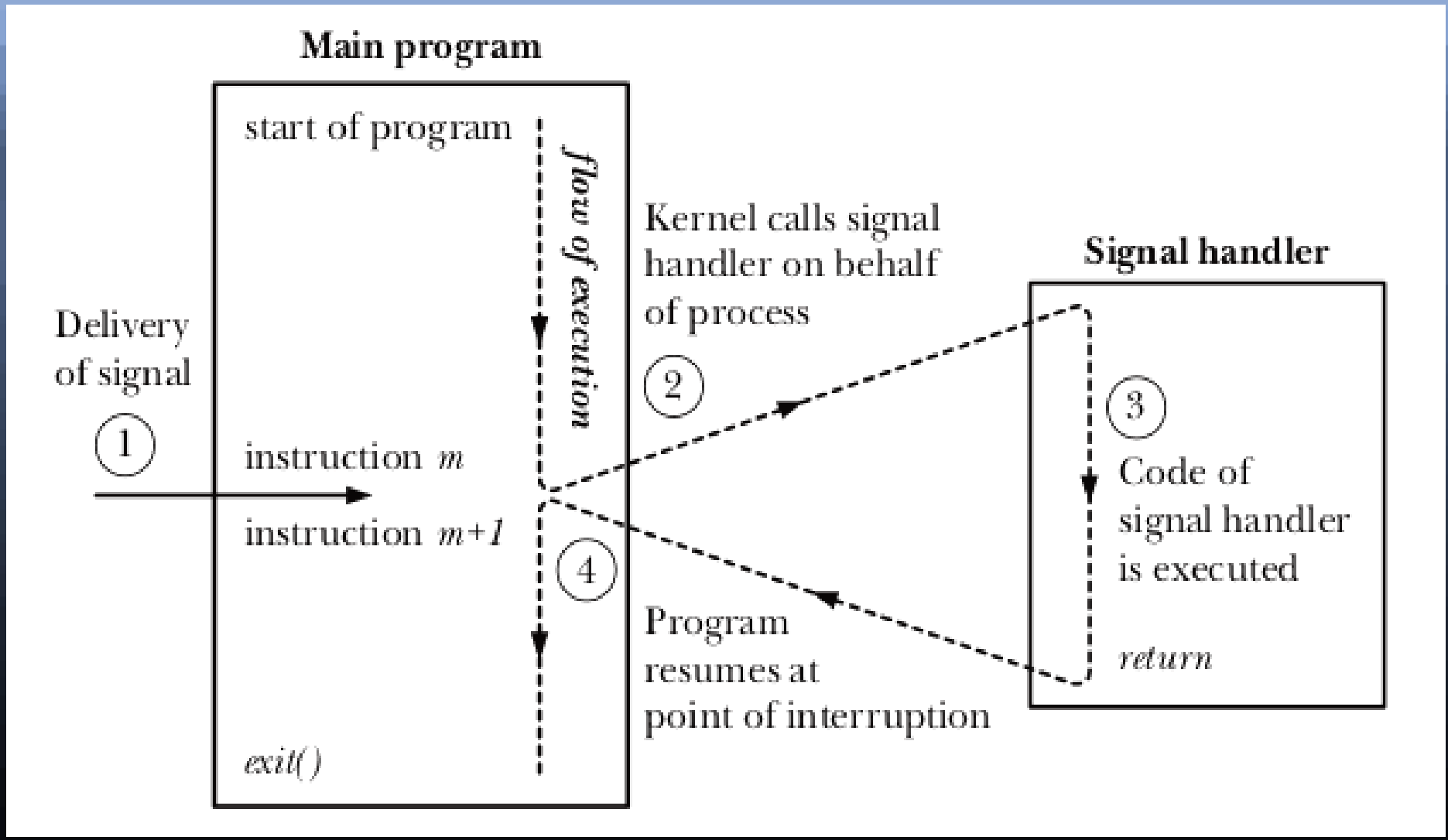
Señales

¿Qué son ?

- Una señal es una notificación asíncrona entregada a un proceso a partir de la ocurrencia de un evento

- Para que se usan ?
  - Aviso del kernel de una excepción de Harware
    - División por cero, Violación de Segmento, etc.
  - Notificar al proceso de un evento de software
    - Un hijo termina la ejecución, un contador llega a cero, etc.
  - Para controlarlo desde el teclado
    - Suspender el proceso, terminar el proceso.

Manejador de la Señal



Manejador de la Señal

- Cuando un proceso recibe una señal este puede:
  - Ignorar la señal.
  - Dejar que se ejecute una acción por defecto relacionada con la señal.
  - Ejecuta un manejador para la señal: programa encargado de “hacer algo” con la señal recibida.