

Sistemas Operativos
2º Curso – Grado en Ingeniería Informática

Tema 0:

Repaso del Tema 2 de
Fundamentos del Software

Créditos

José Antonio Gómez Hernández, 2016.



Contenidos

- ▷ Definición y funciones.
- ▷ Componentes de un SO Multiprogramado:
 - SO multiprogramado y de tiempo compartido
 - Concepto de proceso
 - Modelo de estados
- ▷ Descripción y control de procesos e hilos:
 - Bloques de Control de Proceso e hilo
 - Control de procesos
- ▷ Gestión básica de memoria (necesario para Tema 3): Paginación y segmentación

Bibliografía

- ▷ W. Stallings, *Operating Systems: Internals and Design Principles, 8th Edition*, Pearson, 2014.
 - Capítulos 1, 2 (2.1-2.3), 3 (3.1-3.4) y 4 (4.1).
- ▷ Tema 2 de Fundamentos del Software (salvo los apartados de memoria).
- ▷ Cualquier texto básico de sistemas operativos.

1.

¿Qué es un SO?

Definición de SO



Definición de sistema operativo

- ▷ “El SO es un programa que controla la ejecución de los programas de aplicación y actúa como interfaz entre las aplicaciones y el hardware.”
[Stallings7th2011, §2.1]

2.

¿Qué hace el SO?

Cuales son sus principales funciones

Las dos funciones principales

Maquina abstracta:

Es responsabilidad del SO construir una máquina abstracta, que es:

- más fácil (esconde detalles del hardware)
- y uniforme (independencia del dispositivo)

de programar que el hardware subyacente.

Construye abstracciones:

- procesos e hilos
- archivos y directorios
- memoria lineal
- ...

Arbitraje:

Permite que los usuarios/procesos utilicen y compartan (espacial/temporal) de forma transparente los recursos como si estuviesen solos en la máquina.

Multiplexación:

- Espacial: memoria.
- Temporal: CPU, memoria.

3.

SOs multiprogramados

La base de los SOs actuales

SO multiprogramado

- > El SO mantiene en memoria varios programas y alterna su ejecución solapando la ejecución de un programa con la realización de entradas/salidas de otro, es decir, cuando un proceso libera la CPU para esperar por una E/S, el SO le asigna la CPU a otro programa.
- > Objetivo: aumentar la productividad de la CPU.

SO de tiempo compartido

- > ¿Que ocurriría si un proceso no realiza E/S?
Monopolizaría la CPU.
- > El SO mantiene en memoria varios programas y alterna su ejecución retirándole el control de la CPU a un proceso para dárselo a otro programa.
- > Objetivo: construir sistemas interactivos.
- > Se denominan así pues el SO fuerza a los procesos a compartir el tiempo de CPU.

2.

Procesos

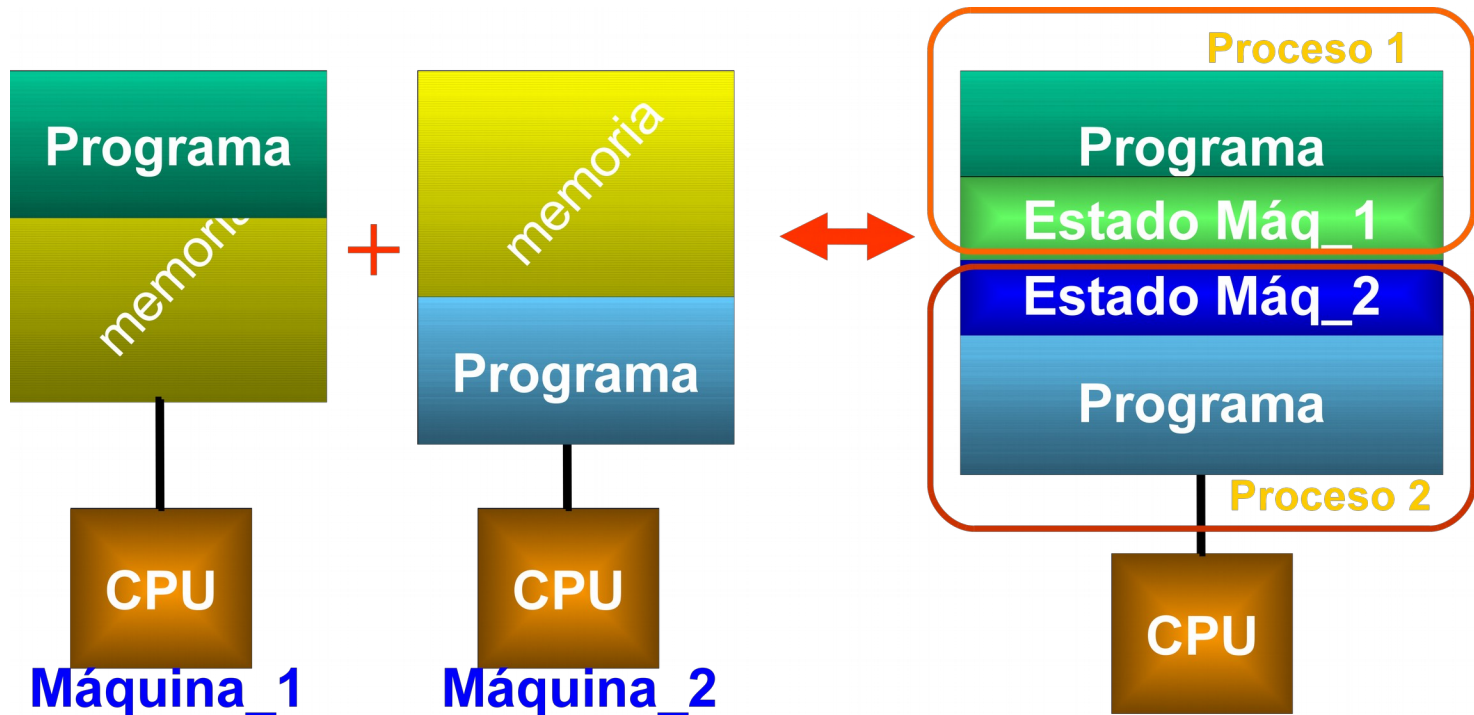
¿Qué es y cómo se materializa?

Procesos

- > Razones para introducir procesos:
 - ◆ Simplicidad – aislar las actividades
 - ◆ Velocidad – solapar actividades
 - ◆ Seguridad – limitar efectos de un error
- ◆> Un proceso es un recurso (CPU) virtual que nos hace creer que tenemos más de una CPU.
- ◆> Se materializa mediante la estructura de datos PCB (Bloque de Control de Proceso).

Procesos: implementación

> Multiplexamos la CPU en el tiempo: hacemos creer a cada proceso que es único.

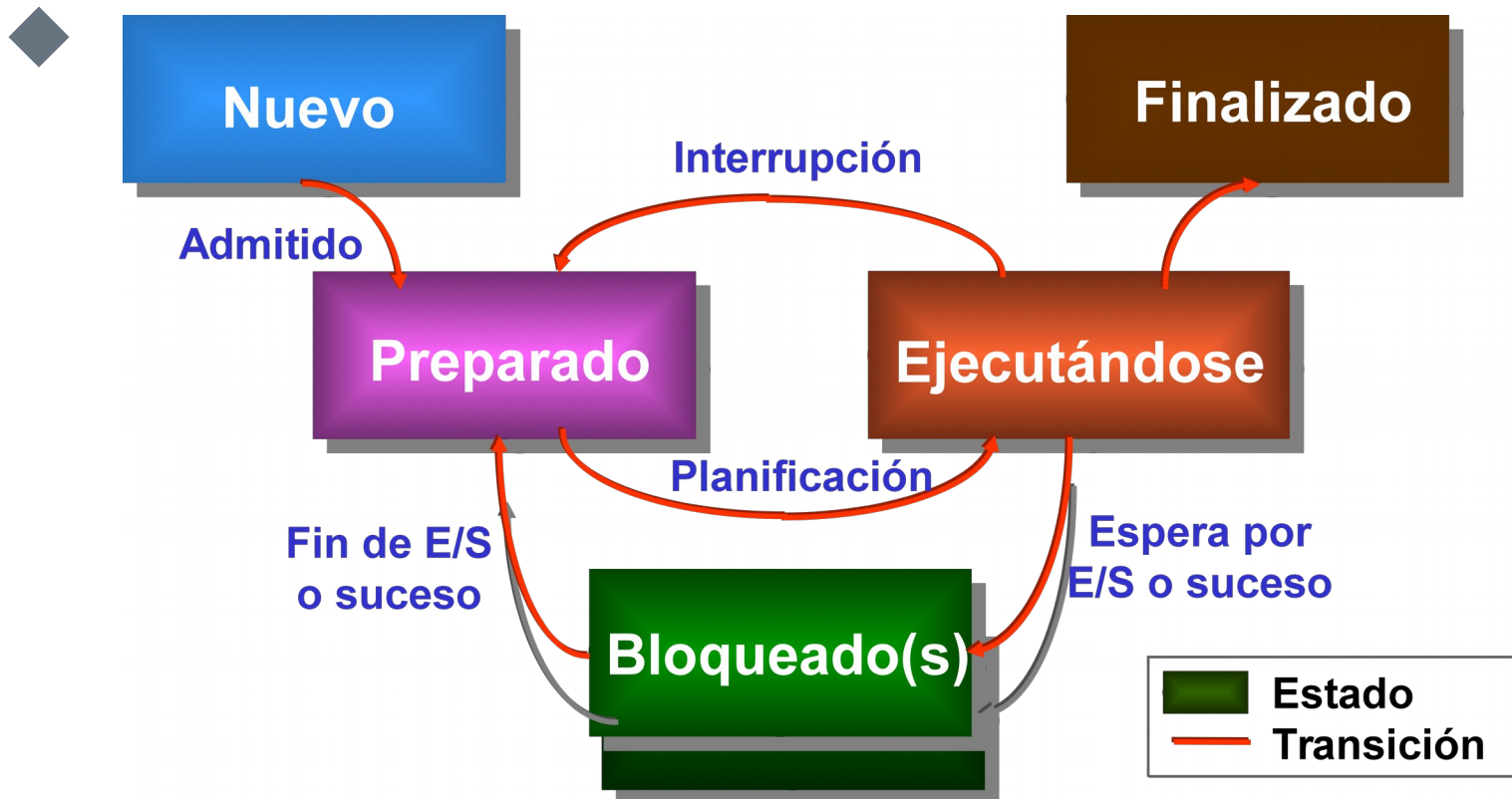


Qué contiene el PCB

- > Toda la información que necesitamos para retomar el estado de ejecución de un programa:
 - ✓ Dónde está el código y los datos
 - ✓ Una pila de ejecución
 - ✓ Valores de los registros del procesador
 - ✓ Recursos del sistema en uso: memoria, archivos en uso, etc.)
 - ✓ Estado de la ejecución

Estados y transiciones

> El estado define a que operaciones podemos someter a un proceso.



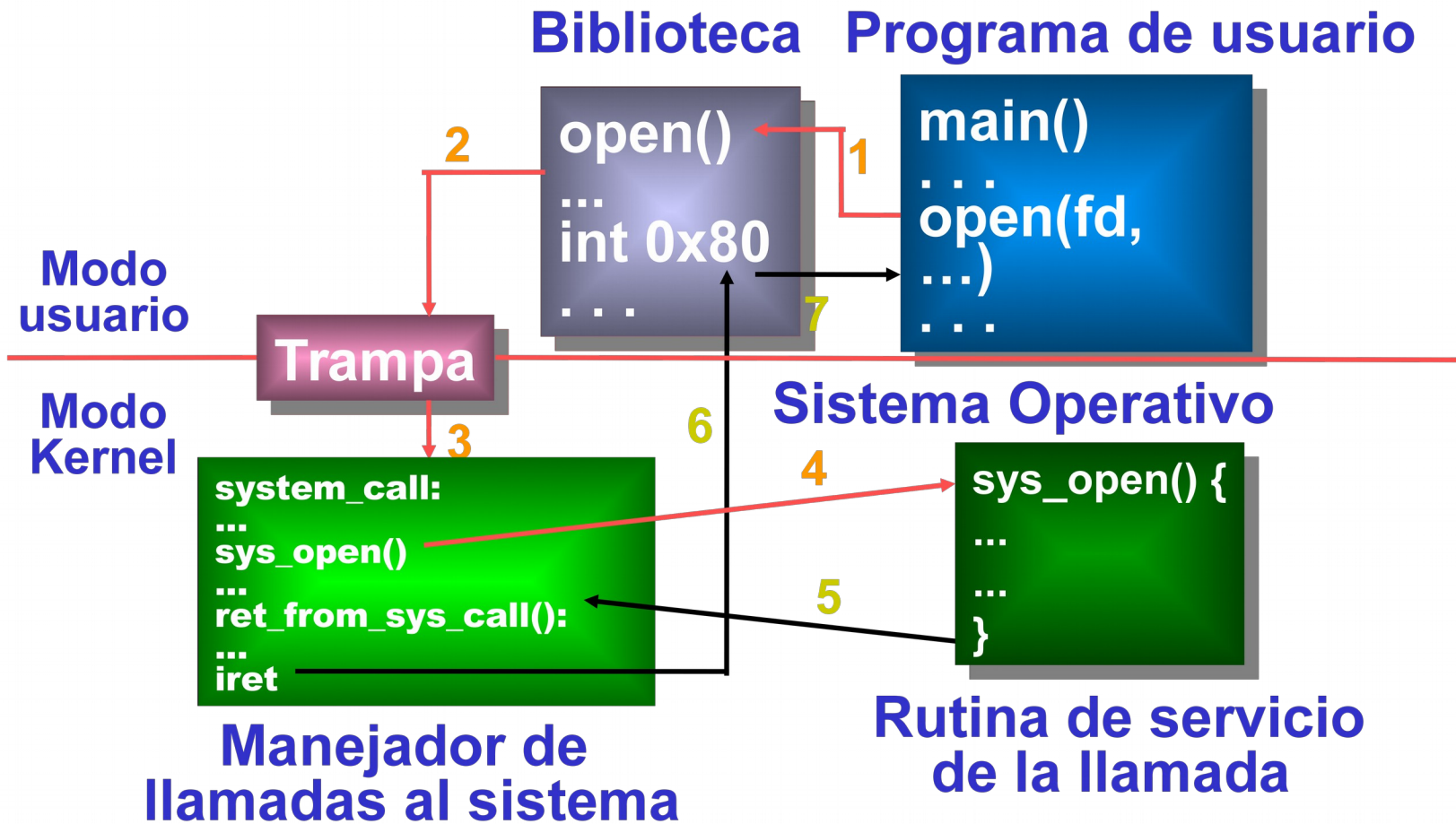
PCB y estado hardware

- > Cuando un proceso esta ejecutándose, los valores de los registros (su **contexto**) son los del procesador.
- > Cuando el SO detiene la ejecución de un proceso para bloquearlo, salva los registros del procesador en el PCB.
- > La acción de conmutar entre procesos se denomina **cambio de contexto**.

Contexto y modo dual

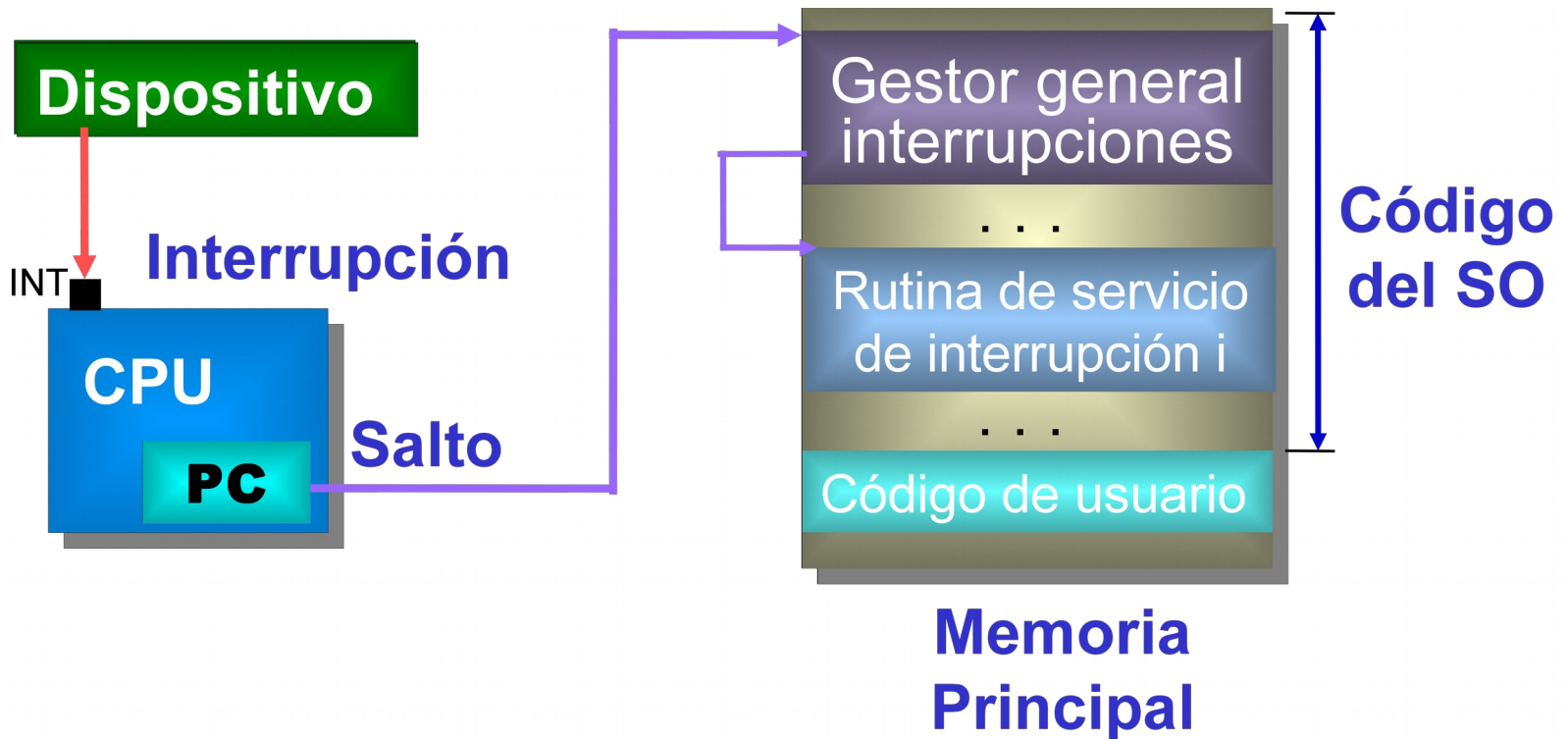
- > En procesadores con dos modos de funcionamiento, separamos el contexto en dos partes (una por modo):
 - ✓ Contexto de usuario
 - ✓ Contexto kernel – usado por el SO
- > Un proceso se puede ejecutar en dos modos.

Llamada al SO

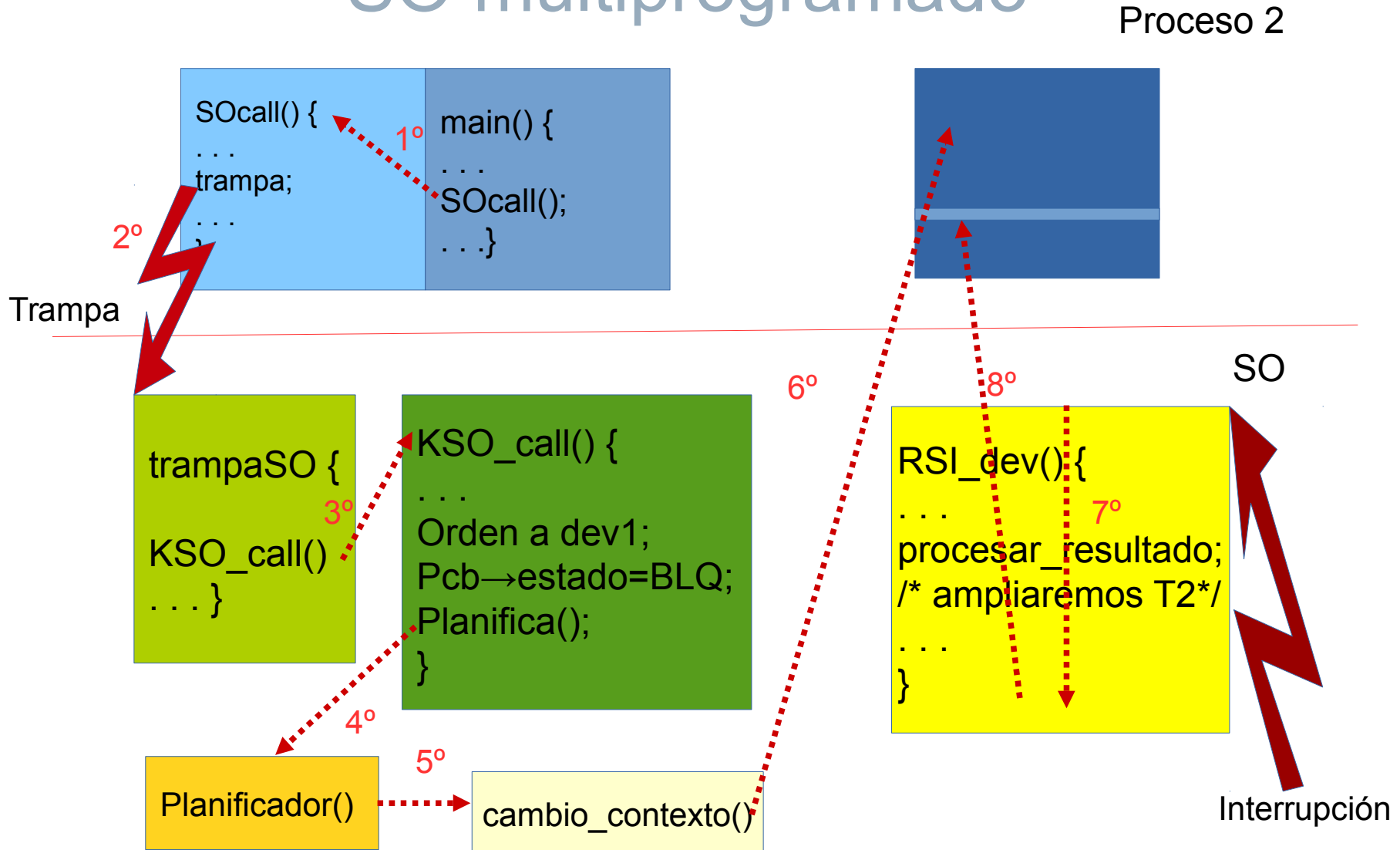


Interrupciones y RSIs

> Proceso de tratamiento de una interrupción:



Estructura general de un SO multiprogramado



Estructura general de un SO multiprogramado: detallado

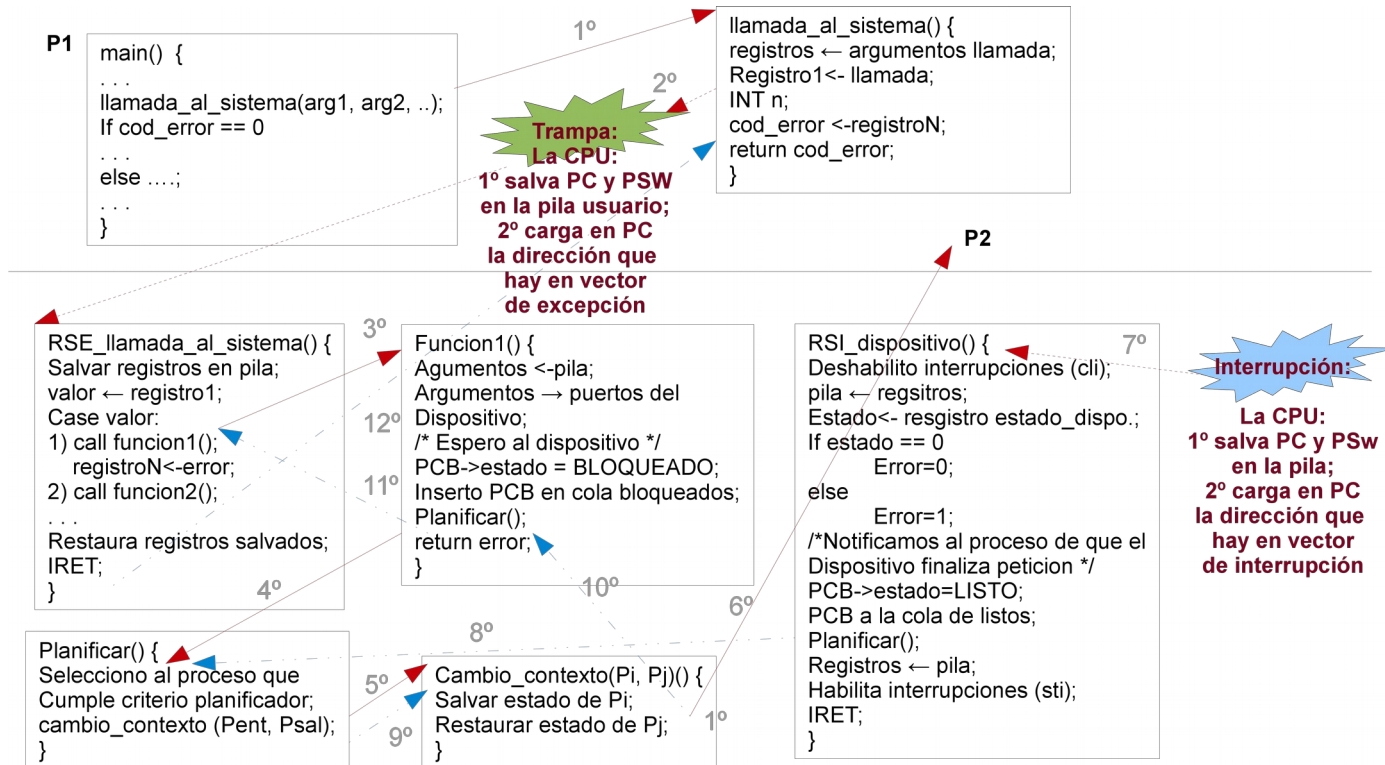


Tabla de vectores de interrupción-excepción

Vector excepción	...
	Dirección de la RSE_llamada_al_sistema()
Vector interrup.	...
	Dirección de la RSI_dispositivo()
	}

Llamada
 Retorno
 Interrupción /Excepción