

Sistemas Operativos

Gestión de Procesos

Gestión de Procesos

- Estados de un proceso
- SO Como gestor de eventos
- SO Como gestor del recurso *Procesador*
- Contexto de un Proceso y PCB

[SGG]: capítulo 3.1
(y algo 3.2.3)

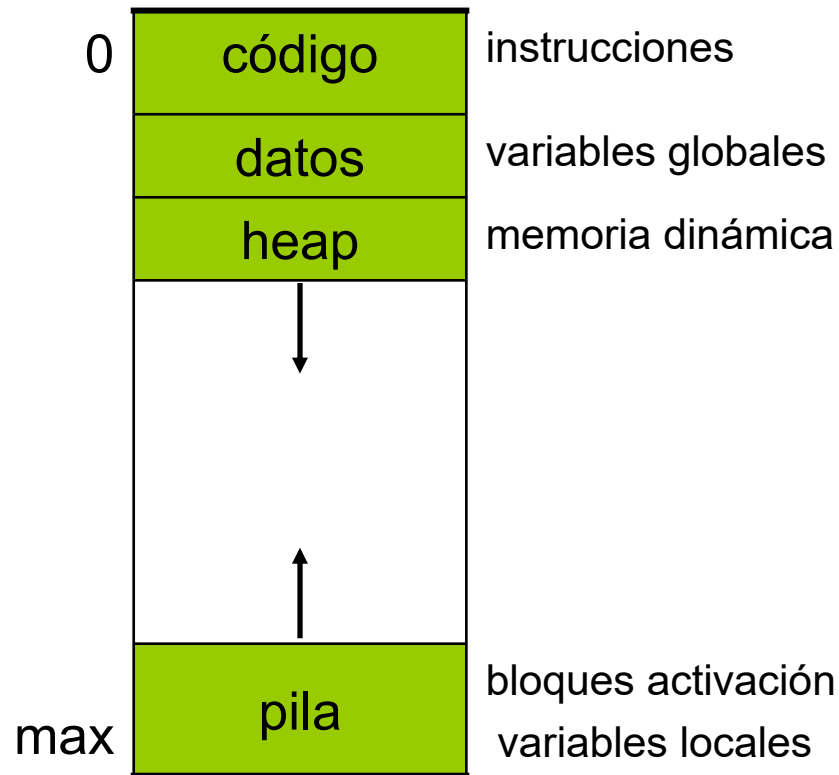
Proceso

- Definición de proceso:
 - Programa en ejecución
 - Unidad de trabajo en un SO moderno
- Nombres alternativos: Trabajo, tarea
- En un SO moderno se están ejecutando de forma concurrente
 - Procesos de usuario (modo usuario)
 - Procesos del sistema (modo kernel)
- En relación con los procesos, el SO se encarga de:
 - Crearlos y destruirlos (tanto de sistema como de usuario)
 - Planificación de procesos (scheduling)
 - Proveer mecanismos para sincronizar, comunicar y evitar bloqueos entre procesos

Proceso \neq programa

- Programa es una entidad pasiva
 - Fichero ejecutable con instrucciones y descripción de variables globales que está en disco
- Proceso es una entidad activa
 - pc que indica la siguiente instrucción que hay que ejecutar
 - Un conjunto de recursos asignados: CPU, MEM, E/S
- El programa se convierte en proceso cuando se carga en memoria y se le asignan recursos para su ejecución.
- Dado un programa, puede haber varios procesos distintos (p. ej. navegadores, editores, etc. del mismo o distintos usuarios) Aunque estos procesos tienen la misma sección de código, difieren en la de datos, pila, registros, heap.
- Formas de ejecutar:
 - Doble clic en icono (GUI = Graphical User Interface)
 - Nombre fichero ejecutable (CLI = Command Line Interpreter = Intérprete Comandos)

Partes de un Proceso

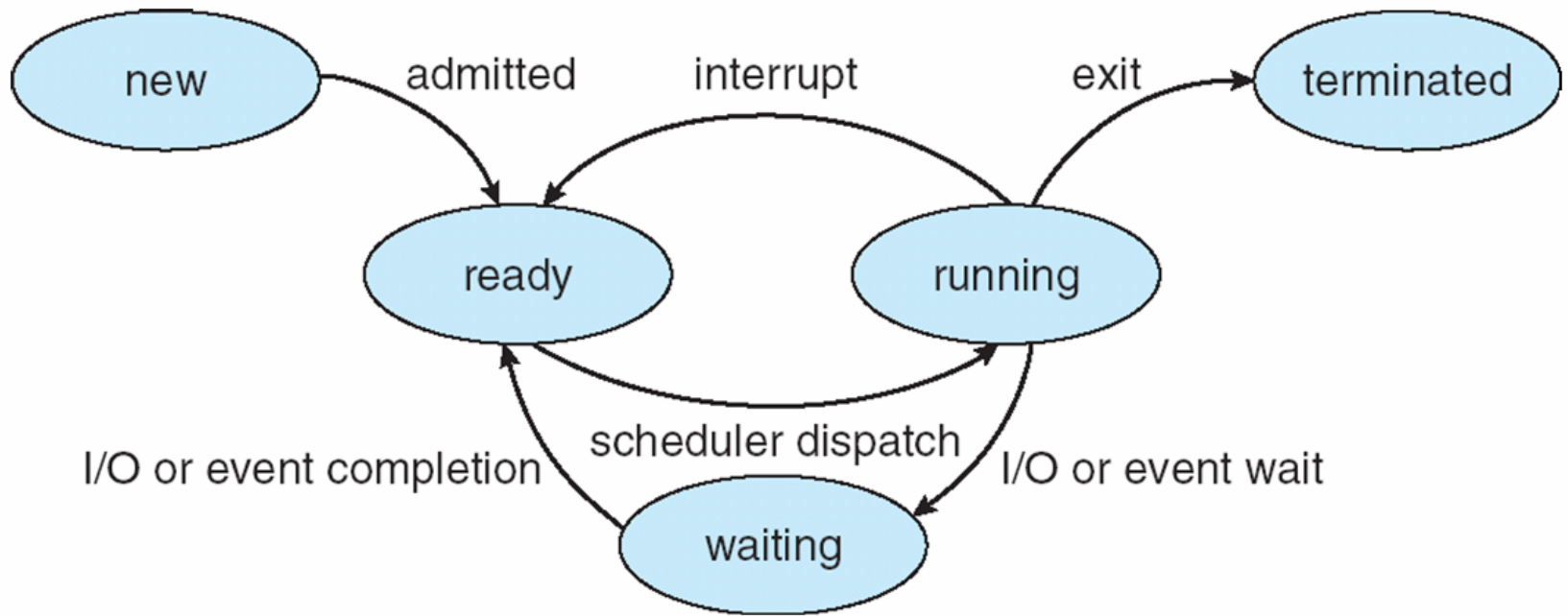


+ registros CPU
(pc, sp, cpsr(psw), ...)

Estados de un proceso

- **Nuevo:** El proceso se está creando (asignando recursos)
 - **Ejecución:** La CPU está ejecutando instrucciones del proceso
 - **Bloqueado:** El proceso está esperando la ocurrencia de algún evento (E/S, señal, mensaje)
 - **Preparado:** El proceso está esperando la asignación de CPU (scheduler)
 - **Terminado:** El proceso ha acabado (liberando recursos).
-
- En un instante dado, la CPU sólo está ejecutando 1 proceso
 - Si CPU de n núcleos, n procesos en ejecución en un instante determinado)
 - Pero puede haber muchos más preparados para ejecutar en el momento en el que el proceso actual se bloquee, agote su quantum o llegue otro más prioritario.

Estados de un proceso



Evolución de un proceso

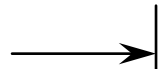
Ráfagas de ejecución separadas por E/S:

Repetir

Trabajo en CPU

Entrada / Salida

Hasta FIN



Programar controlador

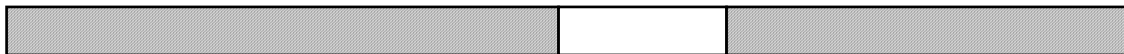
Esperar respuesta

Tiempos de E/S usualmente largos

Procesos limitados por E/S: E/S largas y frecuentes



Procesos limitados por CPU: muy poca E/S

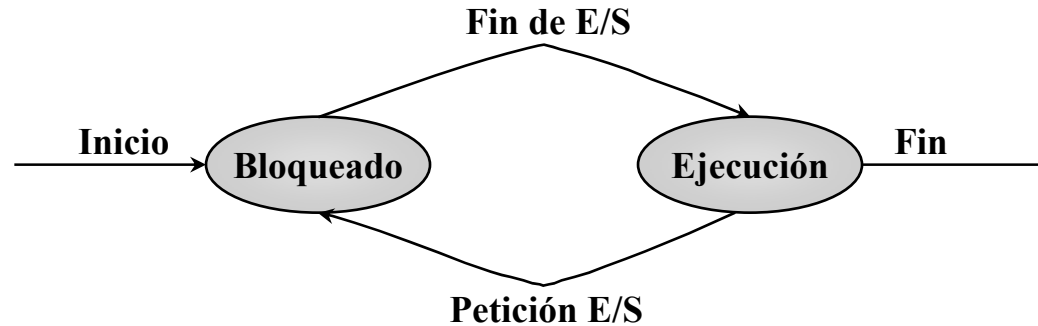


Cálculo



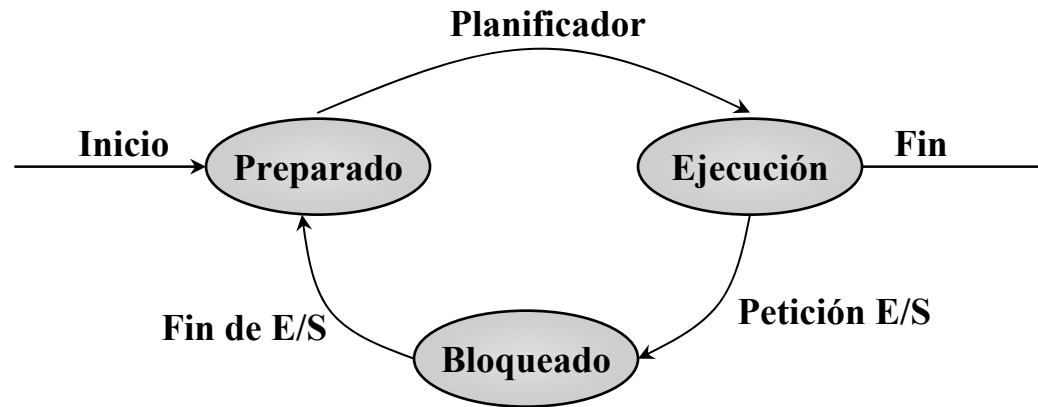
Entrada/Salida

Estados de un proceso (1)



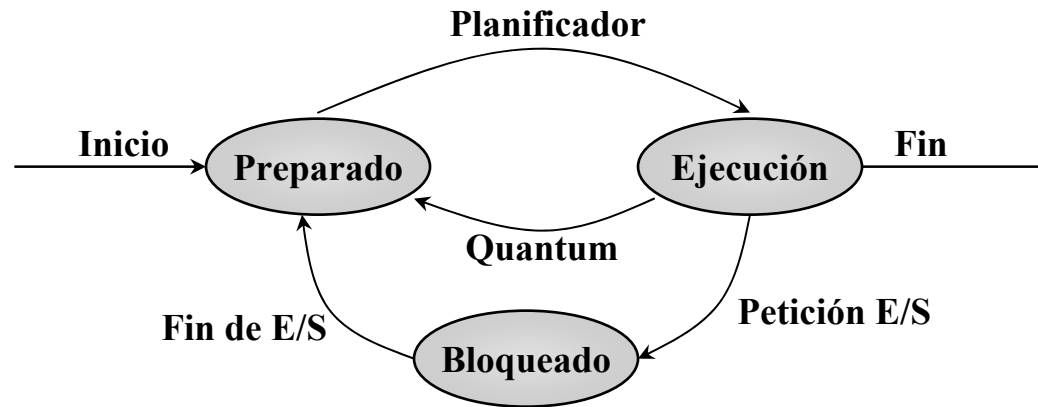
- Sistema Operativo Monoprogramado
 - E/S: pérdida de tiempo de CPU
- Sistema Operativo Multiprogramado
 - Varios procesos en el sistema
 - E/S bloqueante: cesión de la CPU a otro proceso

Estados de un proceso (2)



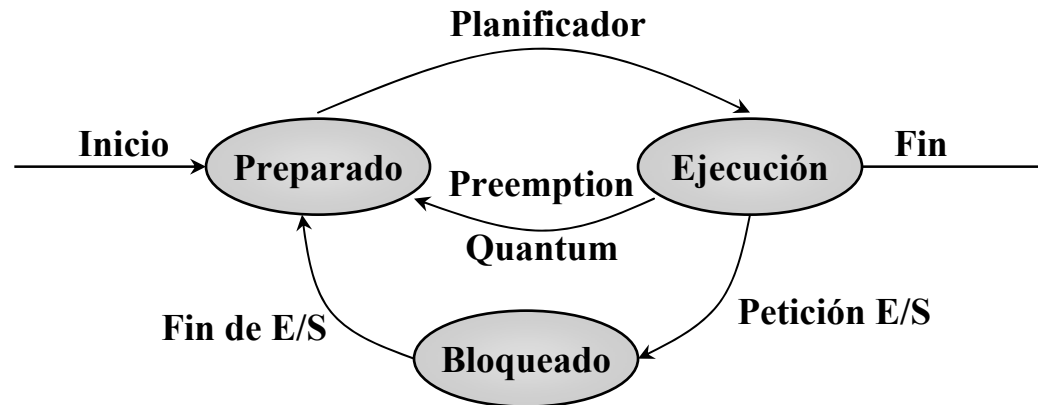
- **Planificador (Scheduler)**
 - Selecciona un proceso entre los *Preparados*
- El proceso en *Ejecución* deja CPU cuando pide E/S
 - Otros procesos pueden esperar indefinidamente
 - No apto para Sistemas *Interactivos* (solo para *Lotes*)

Estados de un proceso (3)



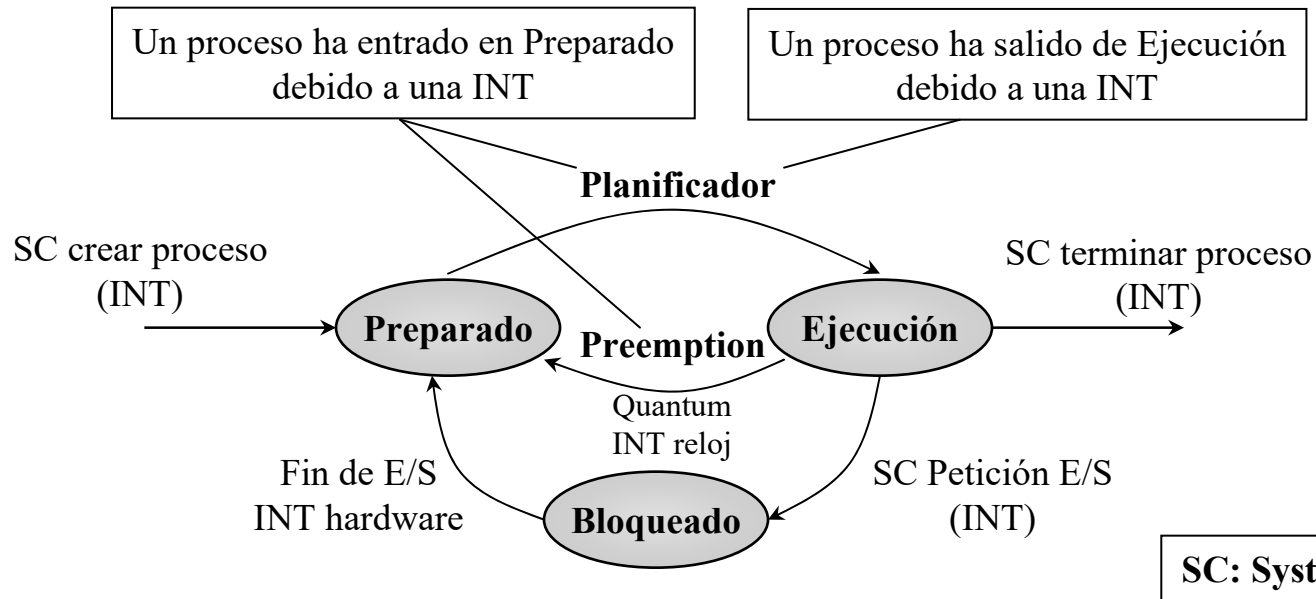
- Quantum
 - Tiempo máximo de permanencia en *Ejecución*
 - Controlado por la rutina de interrupción del reloj
 - El proceso en *Ejecución* deja CPU cuando pide E/S o cuando agota su tiempo de Quantum
 - Tiempo de espera de otros procesos limitado
 - Apto para Sistemas *Interactivos* (*Tiempo Compartido*)

Estados de un proceso (4)



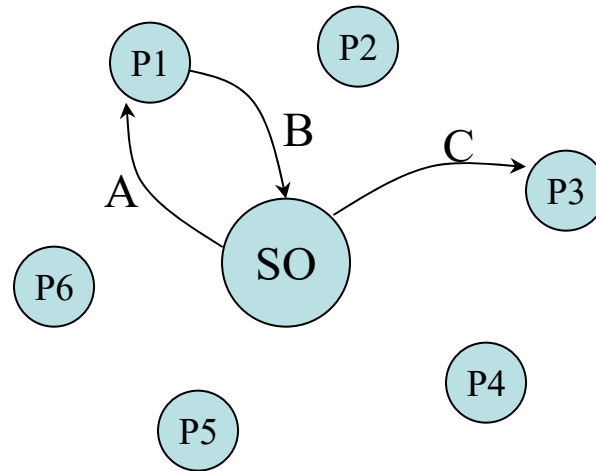
- Preemption
 - Si $\text{Prioridad}(\text{Preparado}) > \text{Prioridad}(\text{Ejecución})$
=> expulsar proceso de *Ejecución*
- Otros autores entienden por Preemption:
 - Un proceso lógicamente ejecutable cede la CPU a otro
 - Por agotamiento de Quantum
 - Por mayor Prioridad

SO Como gestor de eventos



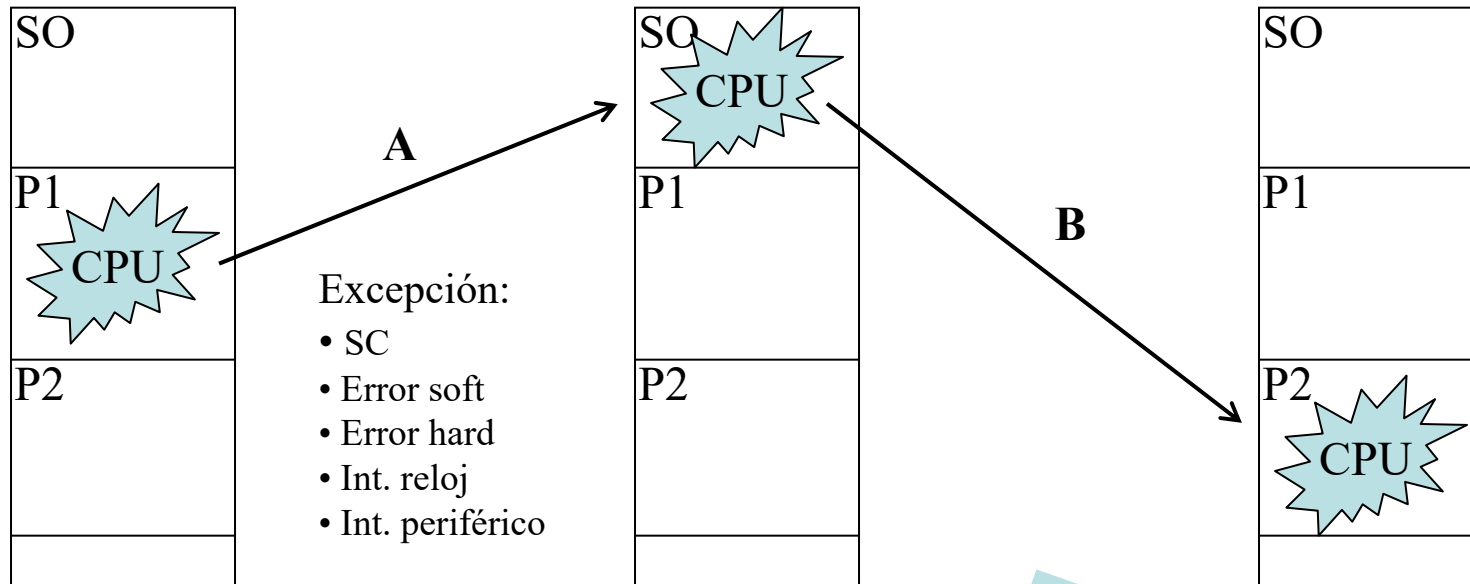
- Todo cambio de estado proviene de una Interrupción
 - El Sistema genera eventos (interrupciones)
 - En cada evento, el Sistema Operativo toma el control y gestiona los cambios de estado pertinentes

SO Como gestor del recurso *Procesador*



- A: El SO cede la CPU a un proceso
- B: El proceso cede la CPU al SO (SC, int. hardware, ...)
 - Los procesos no pueden pasarse el control de CPU entre si
 - Devuelven siempre el control al SO (voluntaria o forzosamente)

Dinámica del SO

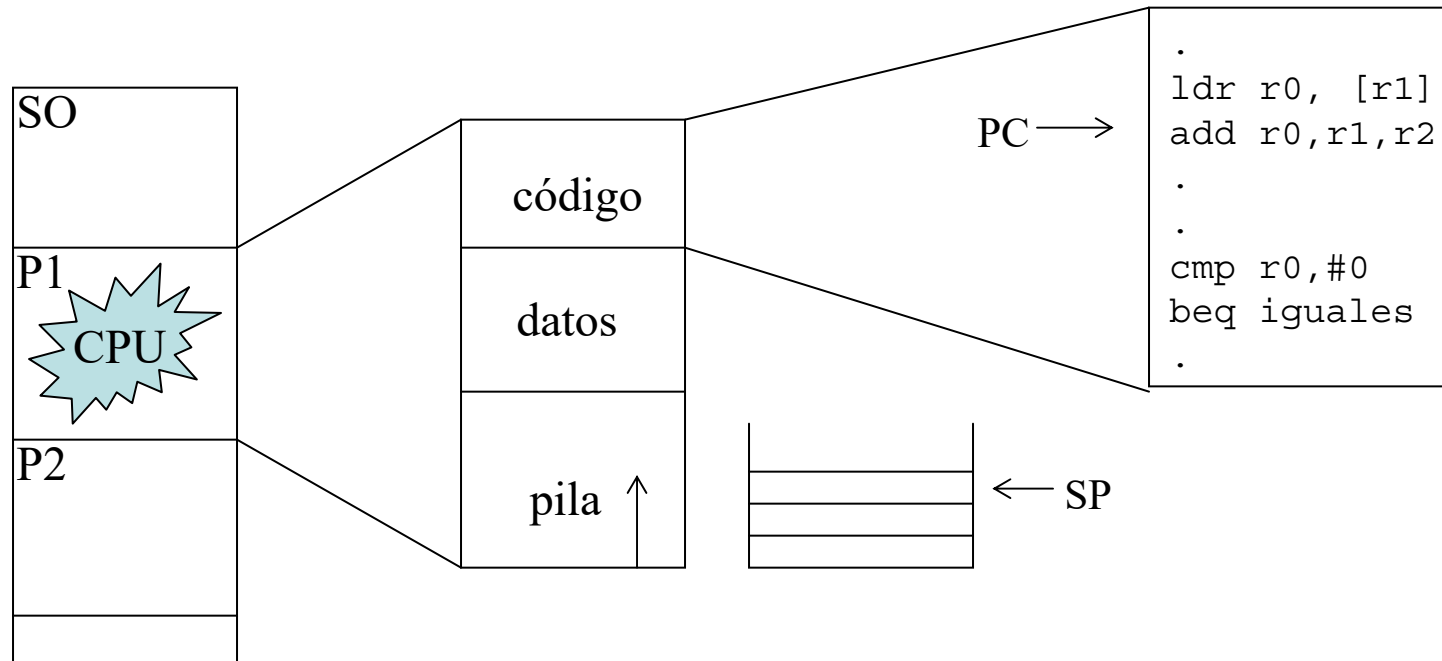


cambio de contexto: de A a B

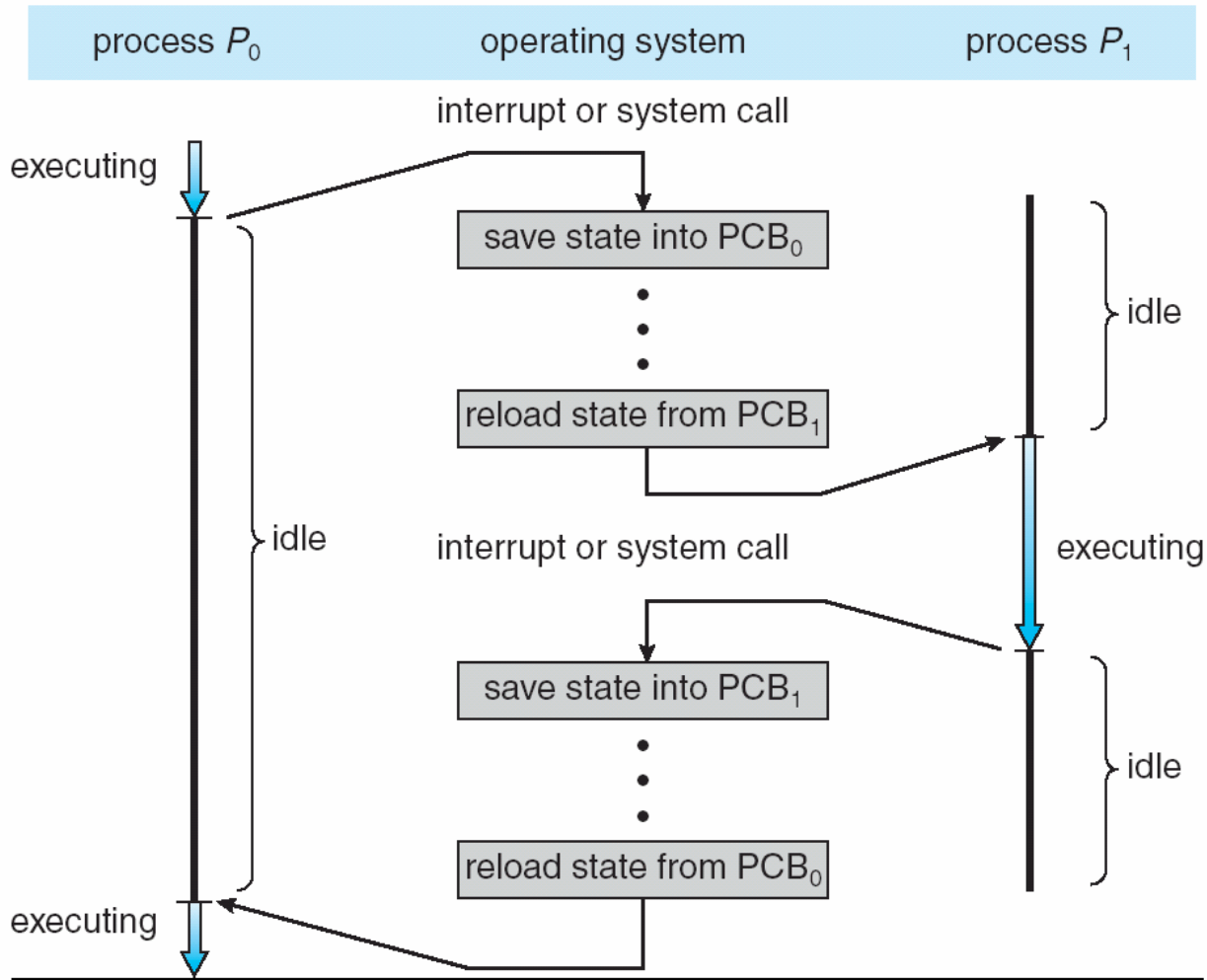
- **A:** Salvar contexto del proceso P1
- Realizar tarea específica para responder al evento
- Poner proceso P1 en el estado apropiado (*Preparado* o *Bloqueado*)
- Seleccionar proceso entre *Preparados* (P2): Planificador (Scheduler)
- **B:** Restaurar contexto del proceso P2 (Dispatcher)

Contexto de un Proceso

- Todo lo que debemos salvar cuando la CPU deja de ejecutar un proceso, para poder retomar su ejecución en otro momento



Cambio de contexto



Bloque de Control de Proceso (PCB)

- Process Control Block:
contiene toda la información relativa a un proceso
 - Identificador
 - Estado
 - Prioridad
 - Padre e hijos
 - contexto I
 - Punteros a zonas de memoria
 - contexto II
 - pc, cprs, registros
 - Info E/S
 - Peticiones pendientes
 - Disp. E/S asignados
 - Ficheros abiertos
 - Info Contabilidad
 - Tiempos acumulados
 - Fechas
 - Cuotas (memoria, tiempo, ...)
- Cada vez que un proceso cambia de estado, el SO anota los cambios en los campos apropiados de su PCB
- Tabla de Procesos: un struct PCB por proceso