

Conceptos de Sistemas Operativos

Electrónica IV

Mg.Ing. Esteban Volentini (evolentini@herrera.unt.edu.ar)

<https://facetvirtual.facet.unt.edu.ar/course/view.php?id=165>

¿Que es un sistema operativo?

- ▶ Un programa que actúa como un intermediario entre la electrónica y el usuario de una computadora
- ▶ **Un gestor de recursos:** Maneja todos los recursos y define pedidos conflictivos para una asignación eficiente y justa
- ▶ **Un programa de control:** Controla la ejecución de los programas para prevenir errores y el uso incorrecto de los recursos

Objetivos de un sistema operativo

- ▶ Simplificar el uso del sistema tanto para los usuarios como para los desarrolladores
- ▶ Utilizar los recursos de forma más eficiente simplificando el uso compartido de un mismo recurso por varios programas
- ▶ Controlar la ejecución de los programas del usuario fijando límites en las acciones que estos pueden realizar

Enlazamiento dinámico

- ▶ El uso de llamadas SVC permite que el programa sea compilado por separado del sistema operativo
 - ▶ El programa no necesita conocer las direcciones de las rutinas de servicio porque se resuelve en tiempo de ejecución
- ▶ Es necesario un conjunto de funciones en el lenguaje de programación que preparen las llamadas SVC
 - ▶ Estas funciones si son compiladas con el programa
- ▶ Esto permite usar el sistema operativo sin brindar acceso al código fuente
 - ▶ Solo se distribuye el código objeto listo para la ejecución

¿Porque en un sistema embebido?

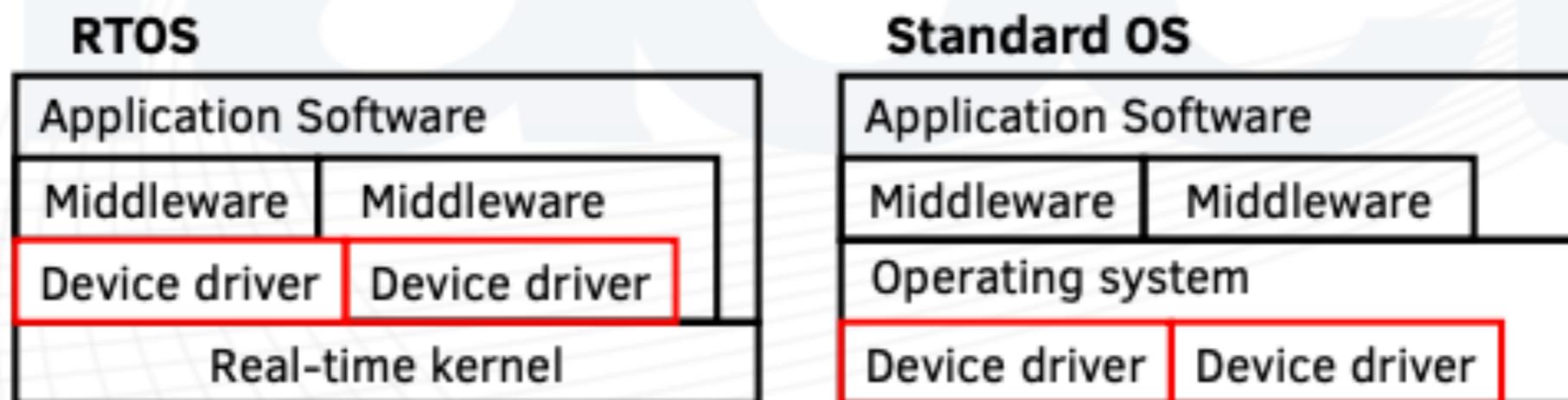
- ▶ Por las mismas razones que en una computadora de escritorio
- ▶ En lugar de interactuar con el usuario lo hacen con el programador de la aplicación
- ▶ No todos los servicios se requieren en todos los dispositivos
- ▶ Características particulares respecto a los sistemas operativos de propósito general

Capacidad de configuración

- ▶ Cubrir todas las necesidades no resulta eficiente
- ▶ Se eliminan las funciones que no utilizan:
 - ▶ Eliminar las funciones no utilizadas con el enlazador
 - ▶ Compilación condicional con `#if` y `#ifdef`
 - ▶ Herramientas de configuración que validan las opciones seleccionadas
 - ▶ Generación de código a partir de la configuración definidas
- ▶ Es necesario tener acceso al código fuente del sistema operativo!

Gestión de los dispositivos

- ▶ Los dispositivos son gestionados directamente por las tareas
- ▶ Mejora la predictibilidad en el comportamiento del sistema
- ▶ No es necesario soportar todos los dispositivos en todas las versiones



Gestión de las interrupciones

- ▶ Las interrupciones pueden ser utilizadas por cualquier proceso
- ▶ En los sistemas de tiempo real es muy importante conocer el tiempo de respuesta de una interrupción y en lo posible minimizar el mismo
- ▶ Las interrupciones pueden iniciar o detener tareas ya sea en forma directa o bien interactuando con el sistema operativo

Sin mecanismos de protección

- ▶ Los sistemas embebidos son diseñados para un único propósito y normalmente no se utilizan programas que no hayan sido probados
- ▶ Es normal que una tarea realice sus propias operaciones de entrada/salida utilizando los modos privilegiados
- ▶ Puede ser necesario utilizar los mecanismos de protección por razones de seguridad

Sistemas operativos de tiempo real

- ▶ Es un sistema operativo que soporta la construcción de sistemas de tiempo real
- ▶ El comportamiento temporal debe ser predecible: las reglas de ejecución deben permitir conocer de antemano los resultados del mismo
- ▶ Debe ser determinístico: para todos los servicios del sistema operativo existe un límite definido en el tiempo de ejecución

Espera activa

- ▶ Es la técnica donde un proceso verifica una condición repetidamente mediante el uso de un lazo
- ▶ Es la forma más natural de programación en sistemas no concurrentes
- ▶ No debe ser utilizada en sistemas concurrente de tiempo real dado que el proceso no libera uso del procesador

Espera pasiva

- ▶ Es la técnica donde un proceso permanece sin acceso al procesador hasta que se verifica la condición se espera
- ▶ El sistema operativo brinda servicios para gestionar las condiciones que el proceso espera

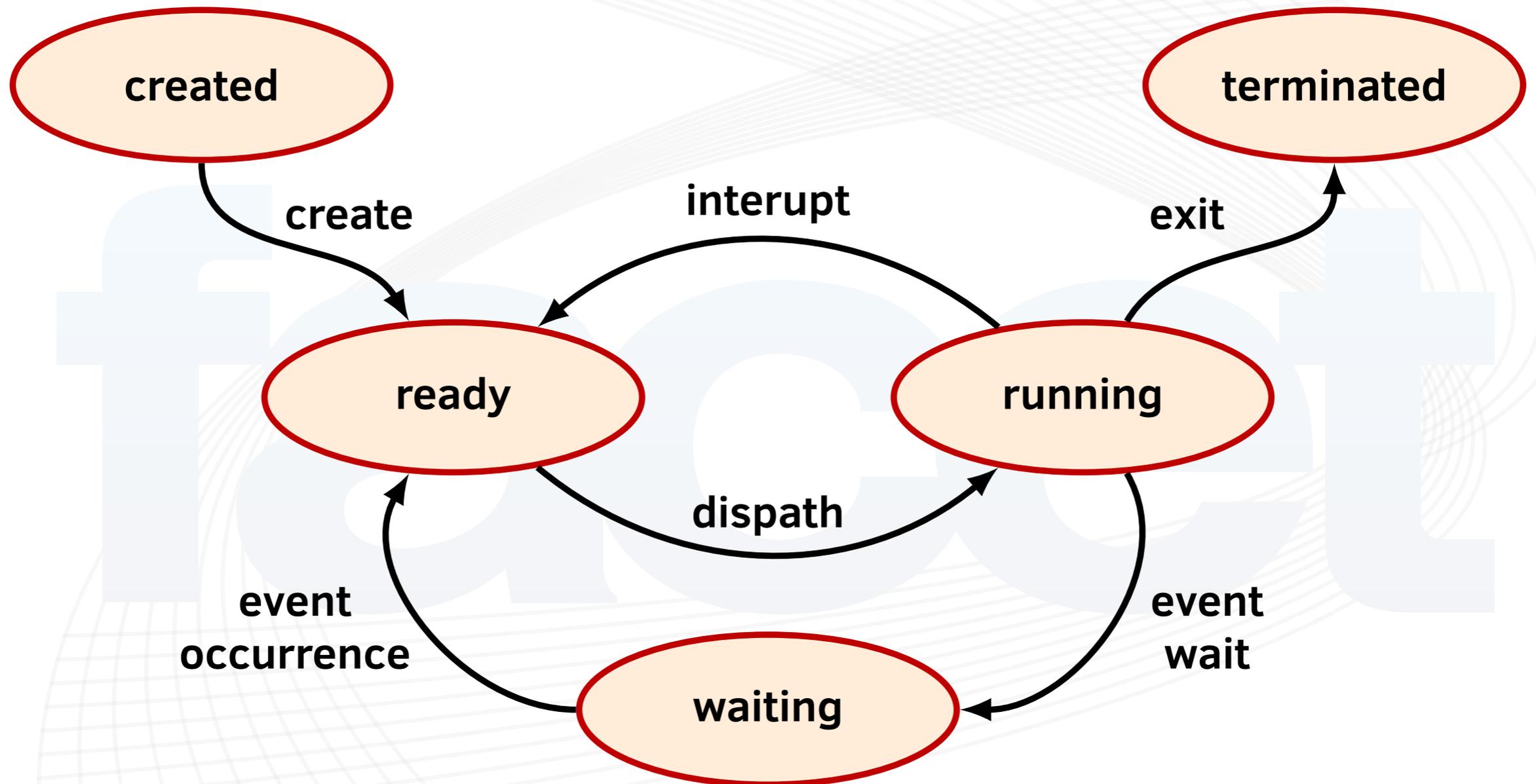
Programas, Tareas y Procesos

- ▶ Un programa es una lista de instrucciones
- ▶ Un proceso es un programa en ejecución
- ▶ Un programa se convierte en un proceso cuando el mismo es puesto en ejecución
- ▶ Un programa es una entidad pasiva que reside en el almacenamiento, mientras que un proceso es una entidad activa
- ▶ Un único programa puede convertirse en varios procesos, cada uno de ellos tendrá las mismas instrucciones pero distintos estados

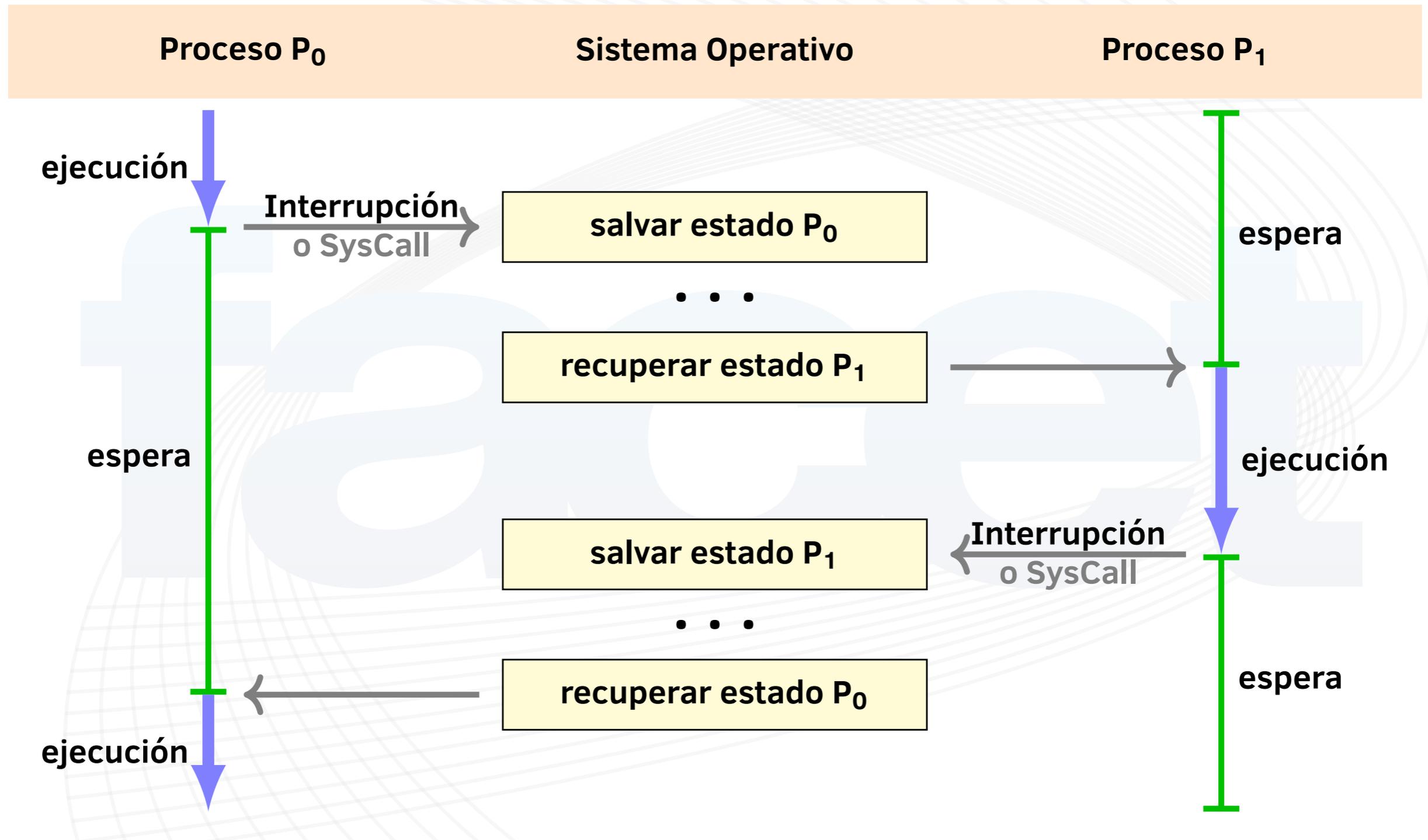
Partes de un proceso

- ▶ **Text área:** contiene el código del programa
- ▶ **Data área:** contiene las variables globales
- ▶ **Stack área:** contiene datos temporales como parámetros de funciones, direcciones de retorno y variables locales
- ▶ **Heap área:** contiene datos creados en forma dinámica durante la ejecución
- ▶ **Context:** contiene el estado actual del proceso, por ejemplo el contador de programa y los registros del procesador

Estados de un proceso



Cambio de contexto usando Traps



Planificador o Scheduler

- ▶ Es la parte del sistema operativo encargada de asignar el uso del procesador
- ▶ La asignación se realiza siguiendo un conjunto de reglas llamadas **Política de Scheduling**
- ▶ Las políticas más utilizadas son **Round Robin y FiFo**

Políticas más utilizadas

▶ **Round Robin:**

- ▶ El procesador se asigna a todos los procesos por igual siguiendo una ronda
- ▶ Es la política más utilizada en los sistemas operativos de escritorio

▶ **FiFo:**

- ▶ El procesador se asigna al proceso de mayor prioridad que lleva más tiempo esperando
- ▶ Es la política más utilizada en los sistemas operativos de tiempo real

Sistemas cooperativos

- ▶ En este tipo de sistemas el proceso que hace uso del procesador debe ceder voluntariamente el mismo para que otro proceso pueda ejecutarse
- ▶ Son más simples de implementar porque no tienen problemas de sección crítica
- ▶ Es más difícil cumplir límites de tiempo

Sistemas expropiativos

- ▶ Este tipo de sistemas cuenta con la capacidad de quitar el procesador al proceso que está ejecutando sin intervención del mismo
- ▶ No existe la posibilidad de que un proceso monopolice el uso del procesador
- ▶ Es más difícil predecir el comportamiento del sistema