

Conceptos de Sistemas Operativos

Ing Esteban Volentini
evolentini@gmail.com

Dr. JuanPablo Gruer
jpgruer@gmail.com

EITI - Sistemas Tiempo Real Avanzados



¿Que es un sistema operativo?

Un programa que actúa como un intermediario entre la electrónica y el usuario de una computadora.

Objetivos de un sistema operativo:

- Controlar la ejecución de los programas del usuario.
- Simplificar el uso del sistema.
- Utilizar los recursos de forma más eficiente.

¿Que es un sistema operativo?

- **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.

[SGG12] [SGG12]

¿Porque un sistema operativo 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.
 - Aplicaciones críticas con mucha funcionalidad como aplicaciones medicas, aeroespaciales o automatización de procesos.
 - Aplicaciones críticas con poca funcionalidad como los sistemas de ABS, Airbag o marcapasos.
 - Sistemas no muy críticos con funcionalidad variada como tarjetas inteligentes, electrodomésticos o automatización hogareña.

[Thi17]

Características de los Sistemas Operativos Embebidos

Capacidad de Configuración

- Un único SO que puede cubrir todas las necesidades no sería eficiente, debemos eliminar las funciones que no necesitamos.
- Existen varias alternativas para configurar el sistema operativo:
 - Eliminar las funciones no utilizadas con el enlazador.
 - Utilizar compilación condicional con `#if` y `#ifdef`.
 - Utilizar herramientas de configuración que validan las opciones seleccionadas.
 - Utilizar generación de código a partir de la configuraciones definidas.

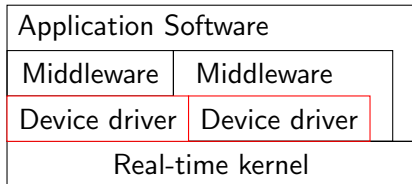
[Thi17]

Características de los Sistemas Operativos Embebidos

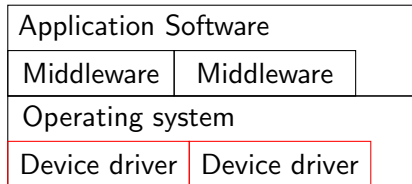
Gestion de los dispositivos

- En general los dispositivos son gestionados directamente por las tareas.
- Mejora la predictibilidad en el comportamiento del sistema.
- El sistema operativo no necesita soportar todos los dispositivos en todas las versiones.

RTOS



Standard OS



[Thi17]

Características de los Sistemas Operativos Embebidos

Las interrupciones pueden ser utilizadas por cualquier proceso

- En los sistemas operativos normales esto es una enorme fuente de problemas de confiabilidad.
- En los sistemas embebidos existen múltiples instancias de pruebas.
- 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.

[Thi17]

Características de los Sistemas Operativos Embebidos

Los mecanismos de protección no son necesarios

- Los sistemas embebidos en general son diseñados para un único propósito, normalmente no se utilizan programas que no hayan sido probados.
- No es necesario utilizar los modos privilegiados, es normal que una tarea realice sus propias operaciones de entrada/salida.
- Eventualmente puede ser necesario utilizar mecanismos de protección por razones de seguridad.

[Thi17]

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 ser simples y claramente definidas para poder conocer de antemano los resultados del mismo.
- El sistema debe ser determinístico: para todos los servicios del sistema operativo existe un límite definido en el tiempo de ejecución.

[Thi17]

Espera Activa y Pasiva

- Se llama espera activa a la técnica donde un proceso repetidamente verifica una condición mediante el uso de un lazo.
 - Es la forma mas 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.
- Se llama espera pasiva a la técnica donde un proceso permanece en un estado sin acceso al procesador hasta que se verifica la condición que la misma esperá.
 - 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 o tarea es un programa en ejecución, y constituye toda la base de la computación
- 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.

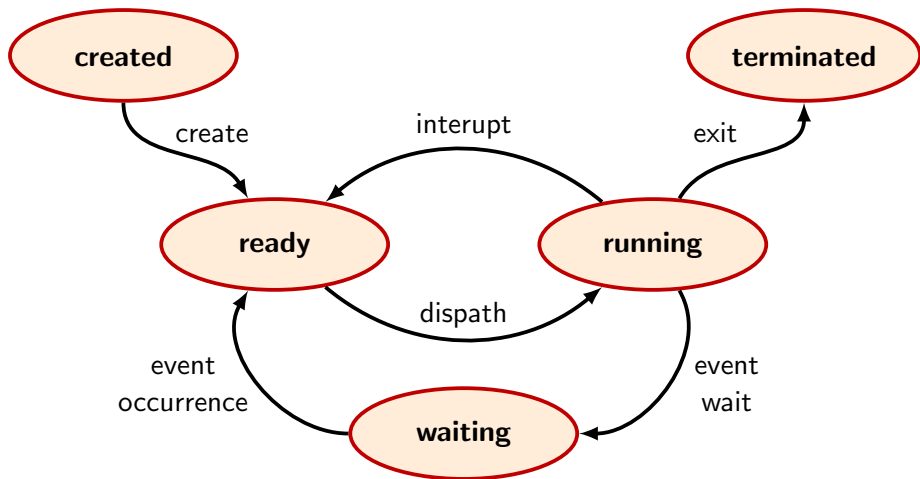
[SGG12]

Partes de un proceso

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

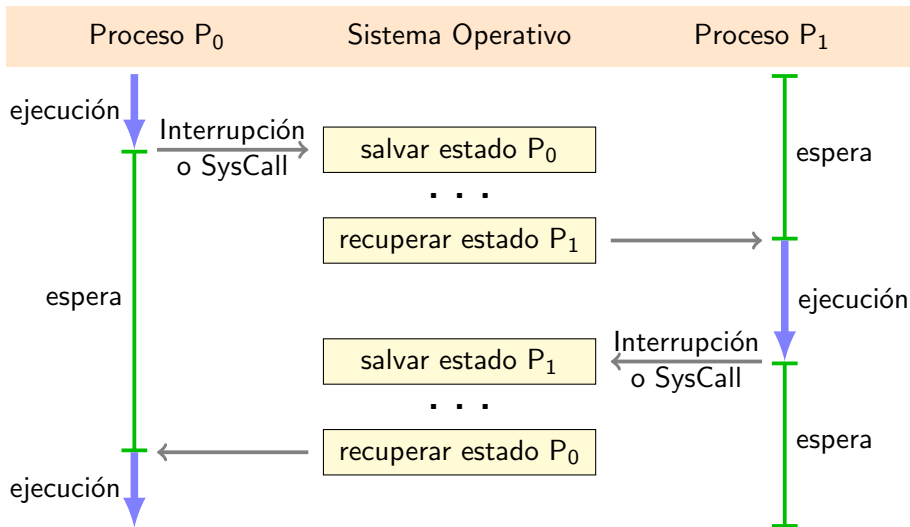
[SGG12]

Estados de un proceso



[SGG12]

Cambio de contexto



[SGG12]

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**.
- **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 Expropiativos y Cooperativos

- **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 mas simples de implementar porque no tienen problemas de sección critica.
- **Sistemas Expropiativos:** Este tipo de sistemas cuenta con la capacidad de quitar el procesador al proceso que esta ejecutando sin intervención del mismo.
 - No existe la posibilidad de que un proceso monopolice el uso del procesador.