

WUOLAH



mma66

www.wuolah.com/student/mma66



18066

Resumen TEMA 1.pdf

Resúmenes Tema1 y Tema2 -SO-



2º Sistemas Operativos



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
UGR - Universidad de Granada

Como aún estás en la portada, es momento de redes sociales. Cotilléanos y luego a estudiar.



Wuolah



Wuolah



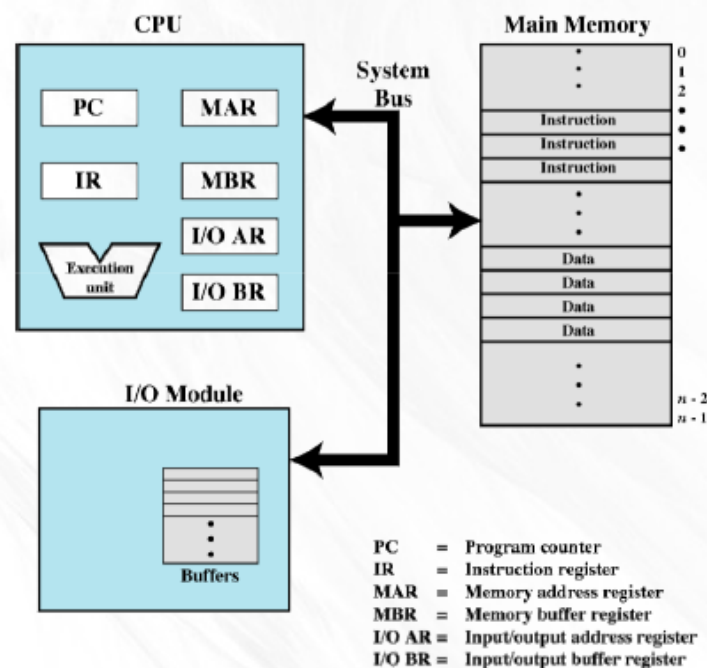
Wuolah_apuntes

WUOLAH

Resumen TEMA 1 –Sistemas Operativos

Soporte hardware (HW) al sistema operativo.

HARDWARE (HW)



CPU

MAR → Registro de dirección de memoria que contiene la dirección de la siguiente R/W.

MBR → Registro de buffer de memoria que contiene los datos que van a escribirse en memoria o recibe los datos que se leerán en memoria.

I/O AR → I/O address register, registro de dirección de E/S.

I/O BR → I/O buffer register, registro de buffer de E/S.

PC → Program Counter. Contiene la dirección de la siguiente ejecución a ir a buscar.

IR → Instruction Register. Contiene la última instrucción que se fue a buscar.

PSW → Program Status Word. Contiene códigos de condición, información para interrupciones y modo de ejecución del procesador.

RAM

Tabla o array lineal compuesto por un número de elementos (celdas de memoria) con un tamaño.

Cada número es la **dirección de memoria** (*memory address*) de la correspondiente celda de memoria.

Espacio de direcciones (*address space*). Conjunto de números que representa las direcciones de una memoria.

Espacio de direcciones en IA-32: En IA-32 los programas no acceden directamente a la memoria física sino indirectamente usando los **modelos de memoria**.

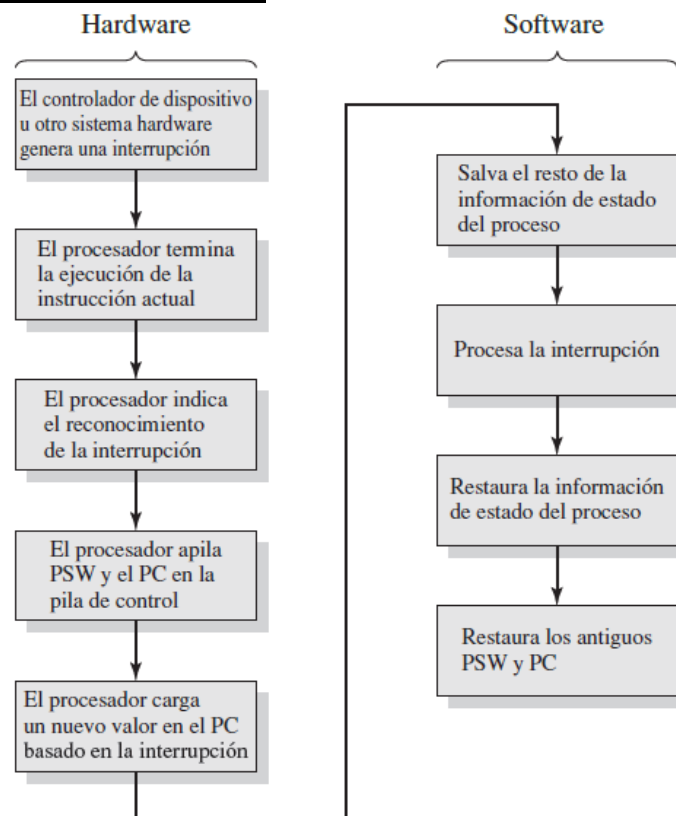
1. **Flat memory model.** Es un espacio lineal con direcciones consecutivas en el rango $[0, 2^{32}-1]$. Permite direccionar con granularidad de un byte.

2. **Segmented memory model.** Los programas ven el espacio de memoria como un grupo de espacios independientes llamados segmentos.
3. **Real-address mode memory model.** Implementación de segmentación con limitaciones en el tamaño de los segmentos, 64KB, y en el espacio de memoria final accesible, 2^{20} bytes.

MÓDULO E/S

- Control y temporización.
- Comunicación con el procesador y con los dispositivos.
- Almacenamiento temporal de datos (*buffers*).
- Detención de errores.

TRATAMIENTO DE INTERRUPCIONES



Estructura del SO.

Proceso. Un proceso debe tener asociados dos elementos: El programa a ejecutar y la información para supervisar y controlar la ejecución del programa.

Funcionalidades del gestor de procesos

- 1● **Creación** del PCB asociado a un programa que va a ejecutarse y **eliminación** una vez finalizada la ejecución del programa.
- 2● **Bloqueo** y **desbloqueo** de los procesos dependiendo de las condiciones por las que debe esperar un programa para poder continuar su ejecución.
- 3● Proporcionar mecanismos de **comunicación** y **sincronización** entre procesos.

Funcionalidades del gestor de memoria

- 1● **Protección** de la región/regiones de memoria principal ocupada/s por el kernel y por los programas.
- 2● **Compartición** de parte de las regiones ocupadas por los programas para permitir comunicación entre estos.
- 3● **Asignación/liberación** de la memoria disponible para un programa en cualquier nivel de la jerarquía de memoria y de forma transparente al programador.
- 4● Mantener información sobre la **memoria asignada** a los procesos, núcleo, etc. y la **memoria libre** en cualquier nivel de la jerarquía.

Funcionalidades del gestor de archivos y directorios

Un **archivo** es una colección de información identificada por nombre que reside en almacenamiento secundario.

El sistema de archivos permite:

- **Crear, eliminar, copiar, renombrar**,... archivos y directorios.
- **Mantiene los atributos** de archivo en una estructura de datos.
- **Abrir y cerrar sesiones** de trabajo con archivos, leer, escribir, variar el puntero de lectura/escritura.
- **Asignación y liberación** de espacio en disco para mantener la información de los archivos y directorios.
- Mantener la ubicación en disco de los datos asociados a archivos y directorios así como los **metadatos del Sistema de Archivos**.

Funcionalidades del gestor de dispositivos de E/S

- 1● **Controlar** el funcionamiento de los dispositivos de E/S.
- 2● **Proteger** el acceso mediante API de llamadas al sistema.
- 3● Proporcionar una **interfaz estándar** para todos los dispositivos.
- 4● **Manejador de dispositivos**. Gestiona un determinado dispositivo y se encarga de implementar las operaciones de la interfaz, el manejo de interrupciones, etc.

Funcionalidades del gestor de recursos del SO (gestor de comunicaciones)

- 1● **Equitatividad**. El acceso a los recursos debe estar garantizado para todos los procesos.
- 2● **Respuesta diferencial**. El SO debe planificar la asignación de los recursos teniendo en cuenta características diferenciales de los clientes, procesos, etc.
- 3● **Eficiencia**. El SO debe maximizar la productividad, minimizar el tiempo de respuesta y, en sistemas de tiempo compartido, permitir el trabajo simultáneo de tantos usuarios como sea posible.
- 4● **Protección**. Mecanismo para controlar el acceso de procesos o usuarios a recursos del SO.
- 5● **Seguridad**. Defensa del sistema frente a ataques internos o externos.
- 6● **Control de acceso**. Los sistemas distinguen usuarios para determinar quién puede hacer que.

LLAMADAS AL SISTEMA (system calls)

DEFINICIÓN: Interfaz de programación a los servicios proporcionados por el SO. Los programas lo utilizan mediante una API en lugar de acceso directo a la system call.

PASOS:

1. Se asocia un número con cada llamada al sistema proporcionada por el SO.
2. El interfaz de llamadas invoca la llamada requerida por el programador y devuelve el estado de finalización y los valores de retorno.
3. Paso de parámetros:
 - En registros de la CPU.
 - Parámetros en memoria, paso la dirección del bloque en un registro.
 - Parámetros en una pila.

Arquitecturas monolíticas, microkernel y máquinas virtuales.**ARQUITECTURA MONOLÍTICA****ARQUITECTURA MICROKERNEL**

Ventajas	Ventajas
El SO es un único programa que se ejecuta en el modo más privilegiado del procesador	Una pequeña parte de la funcionalidad del SO se implementa como kernel y el resto como procesos de usuario.
Las dependencias entre los distintos módulos son complejas salvo para algunos elementos bien establecidos (¡NO oculta información!)	El microkernel soporta memoria virtual de bajo nivel, creación de procesos (hebras) y, comunicación y sincronización.
El modelo de obtención de servicios es la llamada a procedimiento.	Fiabilidad. Un error en un módulo que se implemente como proceso de usuario solo provoca una caída parcial del sistema que puede recuperarse
	Extensibilidad. Podemos incluir nuevos servicios como procesos de usuario.
Desventajas	Desventajas
La fuerte dependencia entre módulos provoca dificultades a la hora de comprender el código y de realizar modificaciones.	Peor rendimiento que la arquitectura monolítica porque para obtener un servicio se realizan más cambios de modo y de espacio de direcciones.
Al ser un solo programa cargado en memoria el fallo de un módulo puede provocar la caída del sistema.	

VIRTUALIZACIÓN

Se entiende como un duplicado aislado de un computador real. Una VM es capaz de usar los recursos HW (CPU, memoria, almacenamiento y E/S), virtualizados por el VMM. Un VMM o hipervisor es el software que proporciona la abstracción de la máquina real a las máquinas virtuales.

- **Equivalencia.** Un programa ejecutándose sobre un hipervisor debe comportarse de forma idéntica a si se ejecutase sobre la máquina física real equivalente.
- **Control de recursos.** El hipervisor es el único software que controla los recursos virtualizados.
- **Eficiencia.** La mayor proporción de instrucciones deben ejecutarse sin la intervención del hipervisor.

TIPOS DE VIRTUALIZACIÓN

1. **Hipervisor y Máquinas virtuales:** La máquina virtual dispone de una serie de recursos virtuales proporcionados por el hipervisor y sobre ella se ejecuta el SO (*guest OS*), las bibliotecas y las aplicaciones.
2. **OS-level virtualization.**

TIPOS DE HIPERVISOR

Tipo 1, *native o bare-metal*. Se ejecutan directamente en el host HW para proporcionar VM a los SO invitados. *Ej: Xen y VMware ESX/ESXi.*

Tipo 2, *hosted*. Se ejecutan sobre un SO convencional como el resto de programas y el SO invitado se ejecuta sobre la abstracción proporcionada por el hipervisor.

Ej: VMware y VirtualBox .

COMPARACIÓN TIPO1 VS TIPO2

Los hipervisores tipo 1 obtienen mejor rendimiento que los tipo 2 ya que:

- Disponen de todos los recursos para las VM.
- Los múltiples niveles de abstracción entre el SO invitado y el HW real no permiten alto rendimiento de la máquina virtual.

Los hipervisores tipo 2 permiten realizar virtualización sin tener que dedicar toda la máquina a dicho fin.

FUNCIONES DEL HIPERVISOR

- Gestión de la ejecución de MVs.
- Emulación de dispositivos y control de acceso.
- Ejecución de instrucciones privilegiadas.
- Gestión de Máquinas Virtuales.
- Administración del hipervisor.

Sistemas operativos de propósito específico.

SO de Tiempo Real (RTOS)

Un RTOS debe garantizar la corrección no sólo del resultado lógico de la computación sino también del tiempo empleado en producir los resultados.

El problema de estos SO es planificar bien los procesos para poder satisfacer todos los requisitos de tiempo. Los eventos ocurren en “tiempo real” por lo que los procesos RT tienen un tiempo límite que especifica cuando debe comenzar a ejecutarse un proceso o cuando debe finalizar su computación.

Características:

● **Determinismo y Reactividad.** Grado en el que el sistema puede resolver todos los procesos RT cumpliendo todos los plazos de tiempo. Ambas son los factores determinantes del **tiempo de respuesta**.

- El *determinismo* tiene que ver con el tiempo empleado en reconocer una interrupción.
- La *reactividad* tiene que ver con el tiempo que tarda el RTOS en la RSI.

● **Control de la prioridad de los procesos RT.** El usuario debe poder controlar la prioridad del proceso RT.

● **Fiabilidad (tolerancia a fallos).** La degradación en las prestaciones de un RTOS puede ser muy peligrosa. Por tanto, la **estabilidad** del sistema es crítica.

- Un RTOS es **estable** cuando, ante la imposibilidad de cumplir los plazos de tiempo de todos los procesos RT, el sistema cumple los plazos de los procesos más críticos.

SO para Sistemas Empotrados

Un sistema empotrado es un computador que forma parte de una máquina (coche, ABS, TV Digital, GPS, etc).

La principal característica que debe proporcionar es la **robustez**, en cuanto a la ejecución de procesos, ya que debe lidiar con restricciones de memoria y potencia de cómputo.