

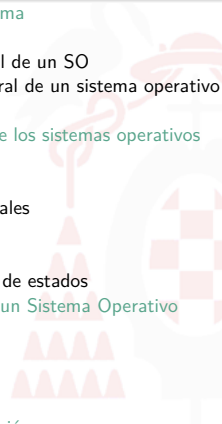
Estructura del Sistema Operativo

Departamento de Automática
Universidad de Alcalá



/gso>

Índice

- 
- ➊ Descripción de un sistema
 - Planteamiento
 - Descripción funcional de un SO
 - Descripción estructural de un sistema operativo
 - El SO y el hardware
 - ➋ Estructura jerárquica de los sistemas operativos
 - Diseño por capas
 - ➌ El núcleo del SO
 - Características generales
 - Funciones básicas
 - Gestión de eventos
 - Soporte al diagrama de estados
 - ➍ Enfoques de diseño de un Sistema Operativo
 - Tipos de enfoques
 - Monitor monolítico
 - Micronúcleo
 - Comparativa
 - ➎ Ejemplos de estructuración
 - Sistema operativo Linux
 - Sistema operativo Mach
 - Sistema operativo Windows
 - Sistema operativo Android

Planteamiento

Necesidad

Disminuir la complejidad en el estudio y diseño de un sistema.

- Solución: diferentes tipos de descripciones
 - Descripción funcional.
 - Descripción estructural.
 - Descripción de la implementación.

Descripción funcional

Objetivo

Funciones que proporciona un sistema operativo.

- Esta descripción sólo considera qué obtiene el usuario del sistema operativo.
- El sistema operativo es básicamente una caja negra, cuyo contenido no interesa.
- ¿Todos los usuarios perciben las mismas funciones?
 - Visión amplia:
 - Propia de los usuarios.
 - Visión restringida:
 - Propia de los desarrolladores del sistema operativo.

Descripción funcional

Visión restringida

- Un SO es un administrador de recursos y una interfaz entre los programas, los usuarios y el hardware.
- Sus funciones son:
 - Compartir el hardware entre usuarios.
 - Facilitar la E/S.
 - Planificar recursos entre usuarios.
 - Definir la “interfaz de usuario”.
 - Permitir a los usuarios compartir los datos.
 - Recuperarse de los errores.
- Los recursos claves que un SO administra son:
 - El microprocesador.
 - La memoria principal.
 - Los dispositivos de E/S.
 - Almacenamiento y los datos.

Descripción estructural

Objetivo

Buscar una organización interna que incremente la portabilidad, extensión y facilite la comprensión y el mantenimiento del sistema

- ¿Cómo abordar el desarrollo de un programa de millones de líneas de código?
- ¿Qué características generales deberá tener un sistema operativo? ¿Y el hardware?

El SO y el hardware

- El SO debe ser seguro, robusto, estable y eficiente.
- El diseñador de SSOO solicita al arquitecto de computadores los siguientes mecanismos:
 - **Modo dual de ejecución.**
 - Modo usuario y modo privilegiado.
 - Mecanismos para pasar de modo usuario a modo supervisor: **Eventos.**
Petición de servicio del SO: Llamada al sistema (SVC)
Otros eventos: interrupciones de E/S, ...
 - **Mecanismos de protección del hardware:**
 - De memoria: Hardware de protección.
 - De dispositivos: Hardware de protección o instrucciones privilegiadas.
 - Del procesador: Interrupciones.
 - **Mecanismos para maximizar el rendimiento del hardware.**
 - Solapar operaciones de la E/S con el uso de la CPU (DMA).

Ventajas de la estructuración por capas

- Las estructuras internas y algoritmos de una capa no son visibles a las demás.
- El sistema puede evolucionar fácilmente manteniendo las interfaces.
- Pueden existir realizaciones alternativas procedentes de distintos grupos de trabajo.
- Algunas capas pueden ser transparentes si sus servicios no son necesarios.
- Cada capa se codifica y prueba de modo independiente (importante en el desarrollo del software).

Niveles clásicos en el diseño por capas de un SO

Objetivos del nivel

Gestión del espacio de nombres lógicos y protección de la información

Gestión de E/S en función de los dispositivos del sistema

Manejo de procesos a alto nivel

Repartir la memoria entre los procesos

Compartir la CPU entre procesos

NIVEL 5: GESTIÓN DE LA INFORMACIÓN

NIVEL 4: GESTIÓN DE DISPOSITIVOS

NIVEL 3: GESTIÓN DE PROCESOS

NIVEL 2: GESTIÓN DE MEMORIA

NIVEL 1: GESTIÓN DEL PROCESADOR

Funciones principales del nivel

- Creación y destrucción de archivos/directorios.
- Apertura y cierre de archivos.
- Lectura/Escritura de archivos.
- Protección de acceso.
- Creación de procesos de E/S.
- Asignación y liberación de dispositivos de E/S.
- Planificación de E/S.
- Creación y destrucción de proc.
- Intercambio de mensajes.
- Detención y arranque de proc.
- Asignación y liberación de memoria.
- Control de violación de acceso.
- Sincronización entre procesos.
- Conmutación de la CPU.
- Gestión de interrupciones.

Funciones básicas

- Proporcionar el entorno adecuado para la existencia de procesos
 - Gestión de eventos.
 - Conmutación del procesador entre procesos (soporte al diagrama de estados).
 - Mecanismos básicos de comunicación entre procesos.
 - Carga inicial y activación de la configuración del sistema.

Tipos de eventos



Gestión de eventos

TRAPS - llamadas al sistema

- ¿Cómo puede el usuario ejecutar una instrucción privilegiada?
 - Petición de servicio al SO: SVC (SuperVisor Call).
 - Modo usuario \Rightarrow **TRAP** \Rightarrow Modo supervisor.
 - Método: a través de una instrucción máquina específica según la arquitectura del ordenador.
 - Al ejecutarse esta instrucción toma el control el S.O.

Llamada al sistema

Son el mecanismo controlado de acceso por parte de los procesos a los recursos del S.O.

Gestión de eventos

Tipos de llamadas al sistema

- Comunicación y sincronización de procesos.
- Ejecución de programas.
- Manejo de operaciones de E/S.
- Manipulación del sistema de archivos.
- Detección y control de errores en tiempo de ejecución.
- Comunicaciones en red.
- Funciones de tiempo.
- Personalización de la interfaz.

Gestión de eventos

Programación con llamadas al sistema

- Las llamadas al sistema se definen a nivel de ensamblador.
- Interfaz POSIX: estándar de interfaz de sistemas operativos portables de IEEE basado en UNIX.
 - API (*Application Programming Interface*) estándar (*routine wrapper* en Linux).
 - Objetivo: portabilidad de las aplicaciones.
 - Disponible en todas las versiones de Unix/Linux.
 - Incluye servicios de sistema para muchos entornos de aplicación.

Gestión de eventos

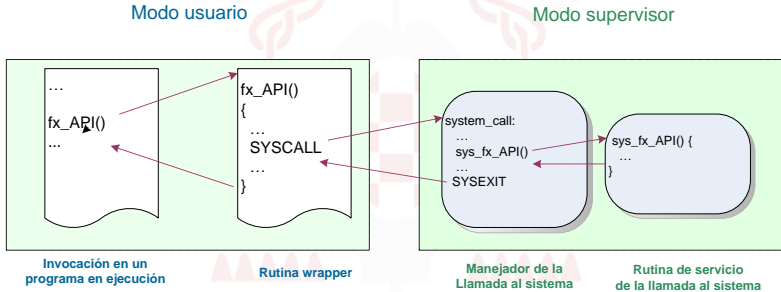
Programación con llamadas al sistema - un ejemplo

write:

```
    pushl %ebx          ; salvar en pila ebx
    movl 8(%esp), %ebx   ; paso de parámetros
    movl 12(%esp), %ecx  ; en registros a rut.
    movl 16(%esp), %edx  ; de tratamiento de la sc
    movl $4, %eax        ; número de sc en eax
    int $0x80           ; invocar a la sc
    cmpl $-126, %eax     ; chequeo cód. de ret
    jbe .L1             ; saltar si no error
    negl %eax            ; complem. valor eax
    movl %eax, errno     ; result. en errno
    movl $-1, %eax       ; poner eax a -1
.L1: popl %ebx          ; restaurar ebx
    ret                 ; retornar al proceso
```

Gestión de eventos

Ejecución de un servicio POSIX



Gestión de eventos

Manipulación de archivos

Crear/Borrar

```
int creat(const char *archivo, mode_t modo);  
int unlink(const char *archivo);
```

Abrir/Cerrar

```
int open(const char *archivo, int operacion,  
         [mode_t mode]);  
int close(int descriptor);
```

Leer/Escribir

```
int read(int descriptor, char *buf, int n_bytes);  
int write(int descriptor, char *buf, int n_bytes);
```

⇒ Otros servicios POSIX: rename, lseek, etc.

Gestión de eventos

Ejemplo de servicio POSIX

Prototipo

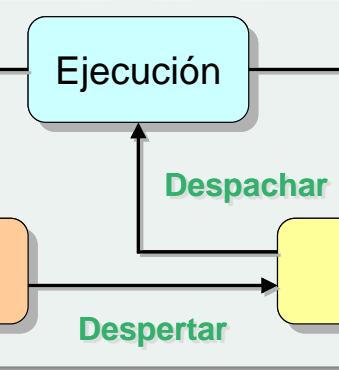
```
int open(const char *archivo, int operacion,  
         [mode_t mode]);
```

- **Descripción**

Busca el archivo de nombre `archivo` en el sistema de archivos y, si la operación es válida, le asocia un descriptor de archivo.

- **Devuelve**

Un entero, descriptor de archivo. Si la llamada fracasa, devuelve -1.

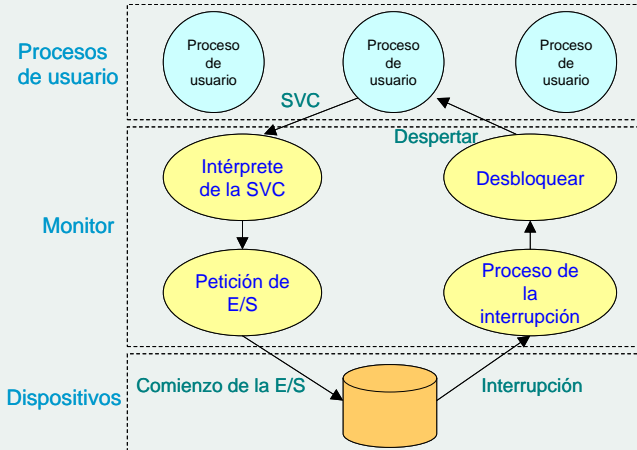


Tipos de enfoques

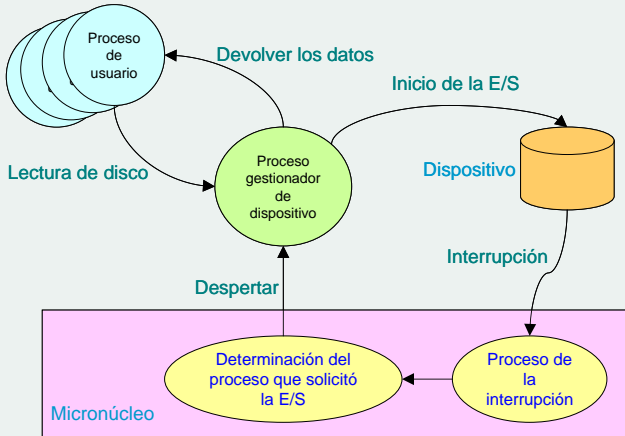
- A grandes rasgos, cabe considerar dos enfoques de diseño:
 - Enfoque **monitor monolítico**
 - Toda la funcionalidad del SO se incluye en el núcleo
 - *MS-DOS, Linux*
 - Enfoque **micronúcleo**
 - Parte de la funcionalidad del SO se realiza por procesos de sistema que se sirven de la funcionalidad mínima proporcionada por un micronúcleo
 - *Mach, L4, Chorus*
- ¿Dónde se clasificará Windows?

Esquema de operación

Monitor monolítico



Esquema de operación Micronúcleo



Comparativa

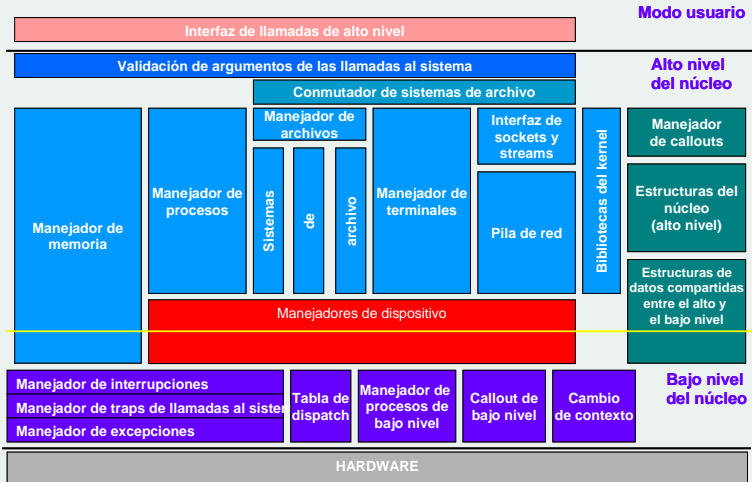
Monitor monolítico

- Todo el SO se ejecuta en modo supervisor
- Menos robusto
- El SO es **ininterrumpible***
- Mayor rendimiento
- Muy adecuado para sistemas pequeños
- Difícil de modificar en tiempo de ejecución
- Menos adaptable

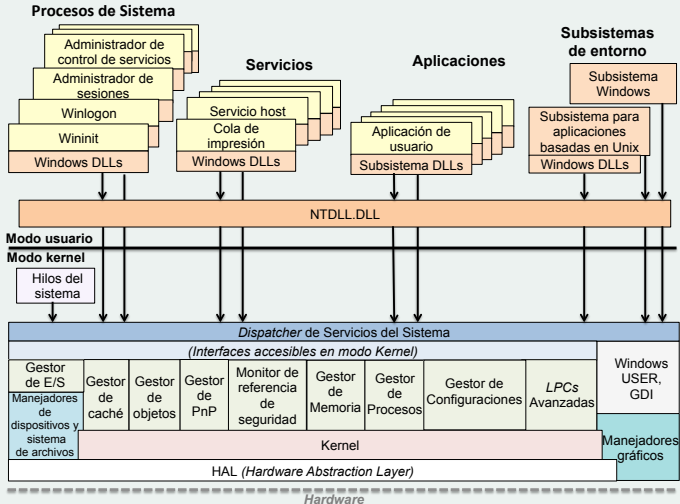
Micronúcleo

- Sólo el micronúcleo se ejecuta en modo supervisor
- Más robusto
- El SO es interrumpible
- Menor rendimiento (sobrecarga de comunicaciones)
- Muy adecuado para sistemas grandes
- Fácil de modificar en tiempo de ejecución
- Más adaptable

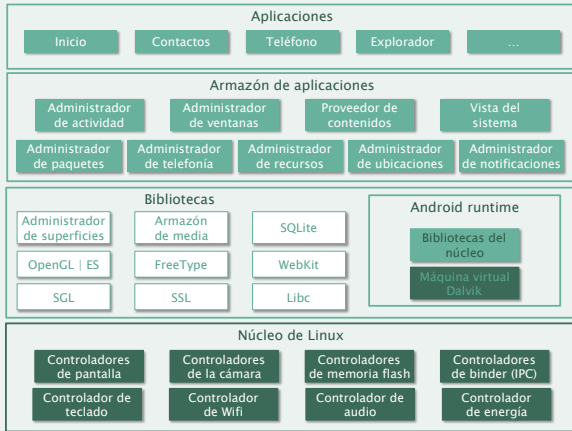
Sistema operativo Linux



Sistema operativo Windows



Sistema operativo Android



Referencias bibliográficas I



[Sánchez, 2005] S. Sánchez Prieto.

Sistemas Operativos.

Servicio de Publicaciones de la UA, 2005.



[Tanenbaum, 2009] A. Tanenbaum.

Sistemas Operativos Modernos.

Ed. Pearson Education, 2009.



[Stallings, 1999] W. Stallings.

Organización y arquitectura de Computadores.

Ed. Prentice Hall, 1999.