

T2-Organizacion-de-ssoo.pdf



gab_martin_br



Sistemas Operativos



2º Grado en Ingeniería Informática



**Escuela Superior de Ingeniería y Tecnología
Universidad de La Laguna**

**70 años formando talento
que transforma el futuro.**

La primera escuela de negocios de España,
hoy líder en sostenibilidad y digitalización.



EOI Escuela de
organización
Industrial



Descubre EOI

Ya has abierto los apuntes,
te mereces ese descanso.

También te mereces que no te cobren
por tener una cuenta. **Cositas.**

Ven a la
Cuenta NoCuenta

Saber más



T2 Organización de los sistemas operativos

Componentes del Sistema

Gestión de Procesos

→Programa:

1. cargar el código en memoria
2. memoria de datos/pila/heap
3. información de gestión del ssoo → PID, E/S
4. ejecución

→Responsabilidad de la gestión de los procesos:

- crear procesos
- suspender y reanudar procesos
- eliminar procesos
- comunicar y sincronizar procesos

Gestión de Memoria Principal

Es el recurso fundamental para las operaciones

Es el único almacenamiento que es accesible para la CPU

Para ejecutar un programa debe ser copiado a la mem principal

→Responsabilidad de la memoria principal:

- controlar la memoria libre
- añadir o extraer memoria
- liberar y asignar espacio
- protección de la memoria virtual

Gestión del Sistema de E/S

- Componente de gestión de memoria E/S
- Interfaz genérica de acceso a los controladores
- Controladores de dispositivo

→ Buffering:

- uso de memoria inmediata de la memoria
- se almacenan datos de forma temporal en una zona de memoria(buffer)
- usado para compensar las esperas de los dispositivos lentos

→ Caching:

- se mantiene una copia de los datos leídos o escritos recientemente en los dispositivos de E/S



WUOLAH

- se almacenan en una fuente de memoria rápida y de fácil acceso(caché)
- Spooling:
- se utiliza en dispositivos que no admiten acceso simultáneo de varias aplicaciones a la vez como impresoras
- usa una cola para asignar trabajos
- los trabajos se envían a un área de almacenamiento(spool) y se imprimen en orden, permitiendo que se trabaje en segundo plano

Gestión del Almacenamiento Secundario

Los programas que se quieren ejecutar, deben estar en la memoria principal, pero no tiene tamaño para almacenar todos los datos y programas del sistema, por lo que se dispone del almacenamiento secundario

→Responsabilidad de la gestión del almacenamiento secundario:

- gestionar el espacio libre en los discos y dispositivos de almacenamiento
- asignar espacio de almacenamiento
- planificar el acceso a los dispositivos, ordenando las operaciones de forma eficiente

Gestión del Sistema de Archivos

Para simplificar el acceso al almacenamiento, el sistema operativo proporciona una vista lógica uniforme para todos los sistemas de almacenamiento, ofrece servicios para leer, escribir, identificar y modificar una colección de bytes, y se usan estructuras como directorios para su mejor organización

→Responsabilidad de la gestión del sistema de archivos:

- crear y borrar archivos
- crear y borrar directorios
- soportar operaciones básicas para manipular archivos y directorios como lectura y escritura de datos, cambios de nombre o permisos
- mapear en memoria los archivos de la memoria secundaria
- realizar copias de seguridad de los archivos en sistemas de almacenamiento seguros

Gestión de Red

El componente de red se ocupa de la comunicación con otros dispositivos interconectados mediante una red de ordenadores como internet o red locales

Interfaz de Programación de Aplicaciones (API)

- Windows API antiguamente Win32, Win16 o Win64 para cada arquitectura
- POSIX es una familia de estándares de API, para poder ejecutar un programa en varios sistemas operativos si son compatibles con POSIX

Llamadas al Sistema

Paso de Argumentos

Registros de CPU

Cargar los parámetros de la llamada en los registros

- muy eficiente
- limitado al número de registros disponibles

Tabla en Memoria

Copiar los parámetros de la llamada en una tabla de la memoria principal y se guarda la dirección de la tabla en un registro

- parámetros en la zona de memoria
- sin límite de parámetros
- menos eficientes

Pila del Proceso

Se insertan los parámetros en la pila del proceso junto las variables locales y el SO los recupera durante la llamada

- no limita el número de parámetros

Librería del Sistema

Las llamadas al sistema proporcionan una interfaz para invocar los procesos, pero al ser en ensamblador no son cómodas, por lo que se llaman a las funciones de la librería del sistema

Características

- es parte del SO
- es una colección de clases o funciones que ofrecen los servicios del SO a los programas
- traducciones de llamadas write()/close()
- es la verdadera API
- es común que estén implementada en C para usar directamente en c++

Operación del Sistema Operativo

Modo Dual

→ Modo usuario → procesos

- instrucciones básicas
- acceso a la memoria propia

→ Modo Privilegiado → procesos de administración

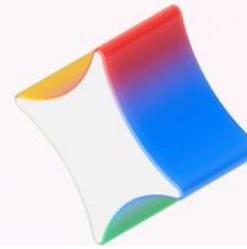
Google Gemini: Plan Pro a 0€ durante 1 año.

Tu ventaja por ser estudiante.

Oferta válida hasta el 9 de diciembre de 2025

Consigue la oferta

Después 21,99€/mes



- instrucciones restringidas
- acceso a toda la memoria
- acceso a todos los dispositivos

→ Memoria

- 0 → superusuario
- 1 → usuario

Ejecución de Instrucciones

1. se arranca en modo privilegiado para cargar el núcleo de instrucciones
2. el núcleo se cambia a modo usuario para que los procesos tengan menos privilegios
3. la CPU conmuta a modo privilegiado si hay excepciones antes de comenzar el código del sistema

Protección de la Memoria

Permite el aislamiento de los procesos y marcar los modos de acceso autorizados en las distintas zonas de memoria como lectura, escritura y ejecución, evitando el modo usuario acceda a zonas donde no debería

1. Espacio de núcleo (núcleo del sistema)
2. espacio de usuario (procesos de usuario)

Temporizador

Se configura por el sistema durante el arranque para interrumpir a la CPU en intervalos para transferir el control al núcleo y así conceder más tiempo al proceso en ejecución, detener procesos o dar mas tiempo más adelante, o tratar errores y acabar programas

→ El temporizador se utiliza para asegurar que ningún proceso acapara la CPU indefinidamente.

Máquinas Virtuales, Paravirtualización

Instalar en el sistema de las máquinas virtuales controladores de dispositivo desarrollados para usarse en mv, dichos controladores saben usar una interfaz que provee el sw de las mv para trasladar las peticiones sin usar el hw simulado

Arranque del Sistema

1. Llega a la CPU una señal de RESET causada por el inicio o reinicio
2. La CPU inicia el contador de programa al bootstrap inicial

Bootstrap: programa que se encarga en primera instancia del arranque del sistema que debe estar almacenado en una memoria no volátil, ya que la RAM está en un estado de arranque indeterminado

WUOLAH

Tareas del Bootstrap

1. Diagnóstico de la máquina (power-on self-test)
 - se detiene si no supera el diagnóstico
2. Iniciar el sistema
 - configurar los registros de la CPU y el contenido de la memoria
3. Iniciar el sistema

Sistemas Operativos por su estructura

Monolíticos

No tienen estructura bien definida ya que son un gran fichero fuente y gran parte de la funcionalidad se implementa en el núcleo

En Capas

Funcionalidad dividida en capas, donde una capa solo usa funciones de la capa inmediatamente inferior a través de una interfaz bien definida
Cada capa oculta su implementación a la capa superior
Son más escalables y corregibles que los de estructura sencilla
Menos eficientes ya que cada capa debe procesar los argumentos, se sobrecarga el funcionamiento del sistema
También son monolíticos ya que gran parte se implementa en el núcleo

Microkernell

Se utiliza en sistemas empotrados, sistemas de automoción o equipos médicos aunque necesita hw más potente que admite modo dual, sistemas que necesitan ser seguros y fiables o sistemas críticos

- elimina los componentes no esenciales del núcleo y se implementan como procesos de usuario
- proporcionan funcionalidades mínimas de gestión de procesos y memoria
- el mecanismo de comunicación permite que los procesos de usuario soliciten servicios de componente del sistema

→ su problema es que se sobrecarga el mecanismo de comunicación ya que los componentes son procesos individuales

Modular

Son eficientes ya que no necesitan un mecanismo de comunicación, aunque son menos fiables y seguros ya que un fallo en un módulo puede inutilizar el sistema, aunque se usan en sistemas de propósito general y en servidores

- se divide el núcleo en módulos y se comunican entre sí
- cada módulo oculta su implementación al resto de módulos