WUOLAH

David97
www.wuolah.com/student/David97

12393

T1SOResumen.pdf

Resúmenes Teoría SO

2º Sistemas Operativos

Grado en Ingeniería Informática

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

Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.

Available on the App Store
GET IT ON Google Play

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

KEEP CALM AND ESTUDIA UN POQUITO

WUOLAH + #QuédateEnCasa

TEMA 1. INTRODUCCIÓN

1 Tipos de Sistemas Operativos

1.1 Sistemas concurrentes

Los sistemas concurrentes son sistemas software donde existen diversas actividades separadas en progreso en el mismo instante. Un SO es concurrente, ya que hay más de una actividad simultanea en curso; al fin y al cabo es un programa diseñado como tal.

1.2 Sistemas multiprogramados

Los primeros SO's eran sistemas por lotes (sistemas *batch*), los cuales planteaban diversos problemas en la 2ª generación debido a la dificultad para gestionar concurrencia.

Hoy en día la inmensa mayoría de los sistemas son multiprogramados. Estos optimizan la productividad del sistema ya que tienen varios procesos en ejecución.

- Los dispositivos de E/S pueden operar asíncronamente.
- Mantienen varios trabajos ejecutables cargados en memoria.
- Solapan procesamiento de E/S de un trabajo con cómputo de otro.
- Necesitan de interrupciones, o DMA (*Direct Memory Access*).

DMA realiza comprobaciones de asignación de cada byte a cada archivo y se encarga de avisar y coordinar los datos entre CPU y memoria principal. Cada vez que se lee un sector por el disco los datos se llevan a memoria principal (RAM), a una zona de memoria mapeada para un proceso en concreto.

1.2.1 Sincronicidad

Las llamadas pueden ser síncronas o asíncronas:

- **Llamada síncrona.** La reanudación de la ejecución de un proceso en CPU se sincroniza con la finalización de la E/S.
- **Llamada asíncrona.** La CPU sigue ejecutando instrucciones mientras el dispositivo de E/S realiza transferencias.

En C, por ejemplo, al realizar un `open` y un `read`, mientras se espera a los datos, el programa se bloquea; por defecto el modelo de programación es síncrono. Es en este momento cuando el proceso pasa a estado bloqueado y entonces el planificador de CPU del sistema operativo asigna la hebra a otro proceso.

Para conseguir que un programa no se bloquee con un modelo síncrono se utilizan las hebras, que rompen con dicho modelo de programación al hacerlo concurrente.

WUOLAH

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

BURN Energy – Encender tu llama cuesta muy poco - #StudyOnFire

1.3 Sistemas de tiempo compartido

Soportan el uso "interactivo" del sistema, donde cada usuario tiene la ilusión de tener disponible la máquina completa gracias a la optimización del tiempo de respuesta, basado sobre todo en el reparto de tiempos de CPU de forma equitativa entre procesos. Un sistema operativo multiprogramado tiene varios usuarios y varios programas en ejecución; en los de tiempo compartido, además, pasa a ejecutar otro proceso cuando pasa un quantum de tiempo. Son adecuados cuando el usuario final necesita acciones interactivas (movimiento de ratón, lectura de teclado, etc.).

1.4 Sistemas de tiempo real

Tienen procesos que tienen que cumplir un plazo estricto (*hard*) o no estricto (*soft*). Cada procesos tiene el tiempo necesario, pues suelen usarse en aplicaciones especializadas que tienen que garantizar respuestas a sucesos físicos en intervalos de tiempo fijos. Esto plantea el problema de poder ejecutar las actividades del sistema con el fin de satisfaces todos los requisitos críticos.

1.5 Sistemas distribuidos

Son sistemas formados por varios computadores, cada uno con su memoria y reloj, cuyo objetivo es compartir recursos distribuidos. El usuario percibe un único sistema operativo centralizado, haciendo más fácil el uso de la máquina. Son mucho más fiables, ya que si una parte falla el resto puede seguir su ejecución. Ejemplos: la nube, clusters.

Un *middleware* es una capa software que se ejecuta en un sistema operativo ya existente y que se encarga de gestionar un sistema distribuido o un multicomputador.

1.6 Sistemas paralelos

Son sistemas donde hay una única memoria con varios procesadores y cuyo objetivo es compartir recursos (comparten memoria y reloj). Sustentan aplicaciones paralelas cuyo objetivo es obtener un aumento de velocidad en las tareas computacionalmente complejas. Hoy en día todos los sistemas operativos son paralelos.

El multiprocesamiento simétrico (SMP) consiste en que todos los procesadores pueden ejecutar código del sistema operativo y de las aplicaciones.

Se puede especificar que ciertas hebras tengan siempre asignadas ciertos procesos. Hay un planificador más general que va asignando cada procesador en función de un histórico o de cada momento, según la necesidad.

WUOLAH

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

BURN Energy – Encender tu llama cuesta muy poco - #StudyOnFire

2 Estructuras de Sistemas Operativos

Los principales componentes de un sistema operativo son los módulos (servicios). Los servicios permiten ejecutar actividades, acceder a información y dispositivos, suministrar memoria, etc. Se integran normalmente en componentes (módulos ≡ subsistemas).

2.1 Componentes UNIX

- **Gestor de procesos.** El procesador no sabe que hay varios programas en ejecución; de esto se encarga el sistema operativo, abstrayéndolo a través de los procesos, que son instancias de programas en ejecución. El sistema operativo se encarga de la creación, destrucción, suspensión, reanudación, sincronización, comunicación, etc.
- **Gestor de memoria.** La memoria principal es el almacenamiento de acceso directo para la CPU y los dispositivos de E/S; es volátil. Tiene que implementar la compartición de memoria entre varios procesos. Otra abstracción del sistema operativo, además de la protección del espacio de direcciones, es la paginación y segmentación (memoria virtual).
- **Gestor de archivos.** Debe montar toda la información acerca de los archivos del sistema operativo. Un archivo es una colección de información con nombre (entidad básica de almacenamiento persistente). El sistema de archivos suministra las primitivas para manipular archivos y directorios y, además, realiza la correspondencia entre archivos y su almacenamiento secundario. También suministra servicios generales: backup, contabilidad y cuotas, etc.
- **Gestor de dispositivos de E/S.** La abstracción de los periféricos se realiza a través de archivos. Un manejador de dispositivo es un módulo, igual que un driver, encargado de gestionar un dispositivo. El driver es un software intermedio que permite que el hardware se entienda con el software (sistema operativo).
- **Sistema de protección.** Es el mecanismo para controlar los accesos de los programas a los recursos del sistema. Se encuentra presente en todo el sistema operativo. Protege tanto el subsistema de archivos como el de E/S. El primer mecanismo de protección del sistema operativo es el `login`. La protección se entiende desde el punto de vista del sistema operativo, mientras que la seguridad es desde fuera.
- **Intérprete de órdenes.** Es un último componente que proporciona el sistema operativo. Es un programa o proceso que maneja la interpretación de órdenes del usuario desde un terminal o archivo de órdenes para acceder a los servicios del sistema operativo. Se pasó de tener el intérprete incluido en el sistema operativo a ser un módulo externo que usa los servicios del mismo para dotarlo de flexibilidad.

BURN.COM
#StudyOnFire
BURN ENERGY DRINK

WUOLAH

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

2.2 Características de un "buen" Sistema Operativo

Dados los módulos, procesos, archivos, etc., todo se relaciona en base a una arquitectura que se diseña pensando qué se quiere conseguir con un sistema. Una arquitectura es un conjunto de componentes relacionados. Un sistema operativo con un buen diseño ha de ser:

- **Eficiente.** Para consumir el mínimo de ciclos posibles de CPU.
- **Fiable.** Se refiere a dos aspectos:
  - Robustez: el sistema operativo debe responder de forma predecible a errores.
  - Seguridad: debe protegerse activamente a sí mismo y a los usuarios de todo tipo de acciones.
- **Extensible.** Debe poder añadir fácilmente nuevas funcionalidades. Esto es algo interesante en sistemas operativos de propósito general.

2.3 Arquitecturas principales

Realmente existen tantas arquitecturas como SO's. Muchos adoptan estilos similares.

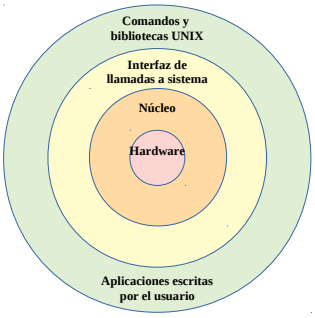
2.3.1 Estructura monolítica

Programado en C. Es la original y más extendida. Se caracteriza porque el SO es un único bloque/componente, aunque a nivel lógico hay varios módulos, pero realmente en el módulo hay "un único programa" que se ejecuta en un único espacio de direcciones. Es como un `main` con llamada a funciones donde la estructura no es clara ni está bien definida.

Tiene modo usuario y modo kernel; los componentes del sistema operativo se ejecutan en modo kernel (modo supervisor) y la relación entre ellos es compleja.

✓Ventajas: eficientes → no hay bibliotecas complicadas ni instancias (menos instrucciones)
✗ Inconvenientes: difíciles de comprender y no fiables → si hay un error se propaga. Además resulta complicado modificarlos para añadir nuevas funcionalidades y servicios.

Sistemas UNIX



En UNIX el hardware subyacente es gestionado por el software del sistema operativo, el cual se denomina frecuentemente como **núcleo del sistema** para destacar su aislamiento frente a los usuarios y a las aplicaciones.

Los programas de usuario pueden invocar los servicios del sistema operativo directamente o a través de bibliotecas. El modo usuario sólo puede ejecutar instrucciones no privilegiadas, mientras que el modo supervisor puede ejecutar cualquiera. La interfaz de llamada a sistemas es la frontera con el usuario y permite que el software de alto nivel obtenga acceso a funciones específicas de núcleo. En el otro extremo, el sistema operativo contiene rutinas primitivas que interaccionan directamente con el hardware. Entre estas dos interfaces, el sistema se divide en dos

WUOLAH

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

BURN Energy – Encender tu llama cuesta muy poco - #StudyOnFire

partes principales, una encargada del control de procesos (gestión de memoria, planificación, ejecución de procesos, sincronización, ...) y la otra de la gestión de ficheros de E/S (intercambio de datos entre la memoria y los dispositivos externos).

Actualmente el núcleo de UNIX está escrito de forma modular, proporcionando funciones y servicios necesarios para procesos del sistema operativo.

Los sistemas UNIX tienen unas interfaces comunes, creadas por el OSF (*Open Software Foundation*) para todos sus SO's: POSIX (***P**ortable **O**perating **S**ystem **I**nterface Uni**X***).

Linux viene de Minix; MacOS viene de FreeBSD; Android es un Linux; iOS está basado en MacOS (arquitectura híbrida: monolítica+microkernel).

Linux

Linux comenzó como una variante de UNIX en 1991. La mayoría de los **núcleos de Linux** son **monolíticos** (casi toda la funcionalidad del sistema operativo englobada en un gran bloque de código que ejecuta como un único proceso con un único espacio de direccionamiento). Todos los componentes funcionales del núcleo tienen acceso a todas las estructuras internas de datos y rutinas.

La estructura modular de Linux permite cargar bloques relativamente independientes denominados módulos cargables, que son ficheros cuyo código puede enlazarse y desenlazarse con el núcleo en tiempo real. La carga dinámica de módulos permite cargar módulos al sistema operativo para poder ser utilizados por los procesos mediante una configuración dinámica sin necesidad de reiniciar el sistema. Un módulo no se ejecuta como su propio proceso o hilo, aunque puede crear los hilos del núcleo que necesite; en su lugar, un módulo se ejecuta en modo kernel. Por tanto, a pesar de las dificultades propias de un sistema monolítico, la estructura modular evita dichas dificultades para desarrollar y evolucionar el núcleo. La capa HAL (*Hardware Abstraction Layer*) es la capa del sistema operativo más baja, que interacciona con el hardware, y la cual hace tan portables los sistemas Linux.

Componentes del núcleo

Los principales componentes del núcleo son:

- **Señales.** El núcleo las utiliza para llamar a un proceso, como por ejemplo la notificación de fallos.
- **Llamadas al sistema.** Son la forma en la cual un proceso requiere un servicio de núcleo específico. Se pueden agrupar básicamente en seis categorías: sistema de ficheros, proceso, planificación, comunicación entre procesos, socket (red) y misceláneos.
- **Procesos y planificador.** Crea, gestiona y planifica procesos.
- **Memoria virtual.** Asigna y gestiona la memoria virtual para los procesos.
- **Sistemas de ficheros.** Proporciona un espacio de nombres global y jerárquico para los ficheros, directorios y otros objetos relacionados con los ficheros. Además, proporciona las funciones del sistema de ficheros.

WUOLAH

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

BURN Energy – Encender tu llama cuesta muy poco - #StudyOnFire

- **Protocolos de red.** Dan soporte a la interfaz Socket para los usuarios, utilizando la pila de protocolos TCP/IP.
- **Controladores de dispositivo de tipo carácter o bloque.** Gestionan los dispositivos que requieren el núcleo para enviar o recibir datos, y los dispositivos que leen y escriben datos en bloques, respectivamente.
- **Controladores de dispositivo de red.** Gestionan las tarjetas de interfaz de red y los puertos de comunicación que permiten las conexiones a la red.
- **Traps y fallos.** Gestionan las trampas (*traps*) y fallos generados por la CPU.
- **Memoria física.** Gestiona el conjunto de marcos de páginas de memoria real y asigna las páginas de memoria virtual.
- **Interrupciones.** Gestiona las interrupciones de los dispositivos periféricos.

2.3.2 Estructura de capas

Construye distintos niveles de abstracción donde cada capa debería utilizar servicios de la capa inferior y proporcionarlos a los de la capa superior , aunque esto no se suele respetar por eficiencia.

✓Ventajas: de tener que cambiar algo solamente afectaría a la capa a la que pertenece. Presenta ocultación de información. Simplicidad de construcción y depuración.

✗ Inconvenientes: los sistemas de capas son jerárquicos, pero los reales son mucho más complejos, de modo que algunas capas se sobrecargan y otras sólo están de paso.

2.3.3 Máquina virtual

Sigue el enfoque de capas para su conclusión lógica, donde la máquina virtual (VM) crea múltiples réplicas idénticas del hardware con la idea de abstraerlo de la computadora y compartirlo en entornos de ejecución diferentes de forma concurrente. Existe un monitor de máquina virtual denominado **hipervisor**. El sistema operativo organiza cada VM como un proceso, aunque luego cada una tenga los suyos propios. Algunos diseños añaden una capa llamada *exokernel*, que se ejecuta en modo privilegiado y que se encarga de asignar recursos a las VM's y de garantizar que ninguna máquina utilice recursos de otra.

✓Ventajas:

- El aislamiento de cada VM asegura la protección de los recursos.
- Sirve para investigar/desarrollar sistemas operativos ya que no interrumpe el funcionamiento del sistema.
- Permite la ejecución de aplicaciones realizadas para otro sistema operativo.
- Añade un nivel de multiprogramación.

✗ Tiene varias características no deseables:

- Aislar cada VM impide después la comunicación entre procesos y conlleva mucho esfuerzo llevarlo a cabo.

BURN.COM
#StudyOnFire
BURN ENERGY DRINK

WUOLAH

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

- Son difíciles de implementar perfectamente por la complejidad de duplicar hardware
- Añade sobrecarga computacional, lo que hace más lenta la ejecución.

2.3.4 Arquitectura microkernel

Consisten en el mínimo kernel posible, con lo más básico, donde algunas de las funciones del sistema operativo (partes no esenciales del kernel) se implementan como procesos de usuario. Los servicios más complejos, en principio, van aparte (en el sistema operativo), simulando un modelo cliente-servidor. Esto se hace para simplificar el núcleo (para depuración, por eficiencia en la implementación, etc.).

El microkernel sólo necesitaría contener la funcionalidad básica para crear y comunicar procesos. El modelo de comunicación es el modelo de paso de mensajes. El kernel se encarga de llevar a cabo la comunicación al ser el intermediario entre el proceso y el sistema operativo, al ser este parte de ambos.

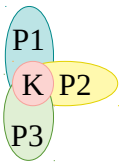
✓Ventajas: Más fiable (un error queda confinado en el espacio de direcciones del proceso que lo implementa), seguro, fiable, extensible y personalizable.
✗ Inconvenientes: menos eficiente al realizar el doble de cambios de modo durante el paso de mensajes: el sistema operativo se ve obligado a realizar un cambio de proceso/contexto que supone en realidad 4 cambios (×2 cambios de modo ×2 cambios de proceso). El procesador deja de ejecutar el proceso demasiadas veces.

2.3.5 Arquitectura Windows

No es cierto que Windows tenga una arquitectura microkernel. Tiene la filosofía de dicha arquitectura, pero los servicios se siguen ejecutando en el núcleo.

2.4 Espacios de direcciones

El espacio de direcciones de un proceso son aquellas direcciones de memoria que el proceso puede direccionar. El propio núcleo del sistema operativo se encuentra en el espacio de direcciones. El código del kernel además está en el propio espacio de direcciones de un proceso, de modo que el núcleo forma parte de todos los procesos (pero está protegido).



WUOLAH

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

BURN Energy – Encender tu llama cuesta muy poco - #StudyOnFire