

Concepto de Sistema Operativo

Definiciones

Evolución de los SSOO

Clasificación de los SSOO

Funciones de los SSOO

Tipos de implementaciones de los SSOO

Soporte HW

Modos de ejecución. Modo Kernel vs usuario

Soporte de interrupciones

Acceso Directo a Memoria (DMA)

Reloj del computador

Jerarquía de memoria: Memoria virtual (MMU)

Preguntas de teoría

Concepto de Sistema Operativo

Un sistema operativo es un programa o conjunto de programas que actúa como intermediario entre las aplicaciones del usuario y el hardware del computador del mismo. Normalmente, definimos el **sistema informático** como el conjunto de hardware y software para el tratamiento automático de la información. El objetivo principal del SO es proporcionar al usuario de una máquina amigable mientras gestiona de forma eficaz el hardware del computador. Existen dos visiones del SO

- Desde la máquina virtual: Se ve como una máquina fácil de usar. El SO proporciona las herramientas para el uso cómodo del HW para evitarnos programar sobre una máquina en *baremetal*
- Gestor del HW: Miramos la máquina desde el punto de vista del HW. Es un programa que administra los recursos sin consumirlos

Definiciones

- **Sistemas por lotes *batch***: Se agrupan trabajos similares que ejecuten el mismo programa para evitar perder el tiempo sustituyendo el programa usando tarjetas perforadas. Estos trabajos se ejecutan de forma **secuencial**
- **Multiprogramación**: Mantener varios programas en ejecución solapando operaciones de E/S y computación en la CPU. Podemos aumentar el rendimiento añadiendo una DMA
- **Microprogramación**:
- **Multiprocesador/Multitarea**: Puede ejecutar más de un proceso a la vez
- **Time Sharing**: Nos permite compartir un recurso de computación entre varios usuarios/aplicaciones proporcionando un tiempo máximo de uso a cada aplicación. Requiere de interrupciones por reloj, Si un SO es de tiempo compartido, entonces también es multiprogramado
- **Librerías**: Una librería es un conjunto de implementaciones funcionales que proporcionan una interfaz bien definida al programador. Estas pueden realizar llamadas al sistema (*syscalls*) para proporcionar funcionalidades del SSOO
- **CLI (Consola de comandos)**: Nos permite ejecutar órdenes en nuestro SSOO

Evolución de los SSOO

Los sistemas operativos han tenido 4 grandes fases:

1. Válvulas de vacío (45). La carga de programas se realizaba a mano usando interruptores y luces por los operarios de computadores. Los ordenadores ocupan habitaciones enteras. Aparecen los primeros compiladores.
2. Transistores (55). Aparición de los primeros SSOO que usan el sistema por lotes (*batch*) para aumentar el rendimiento. Las tareas similares se ejecutaban juntas mientras que el resto se encolaban para evitar desperdiciar tiempo de cómputo. Se añaden las primeras protecciones al HW (Protección de memoria, temporizado, interrupciones, instrucciones privilegiadas)
3. Circuitos integrados(65), discos magnéticos, terminales con monitor y teclado, multiprogramación, time sharing. Se añaden mecanismos para evitar que los programas lean/escriban en las direcciones de memoria de otros programas. Planificación de las tareas de E/S y CPU
4. Microprocesadores (70) con SO para computadoras personales (Macintosh, MS-DOS)

Clasificación de los SSOO

- **SSOO distribuidos:** Red de computadores sin una memoria ni reloj comunes. Visión de un único computador más potente. Muy fiable, funciona aunque uno de sus componentes falle
- **SSOO para multiprocesadores:** Velocidad y tolerancia a los fallos (comparten memoria y reloj)
- **SSOO de tiempo real:** Para gestión de HW de control. Procesan eventos en un plazo de tiempo acotado. Tienen restricciones temporales, planificador con prioridades
- **SSOO de internet:** SO basado en navegador de internet
- **SSOO dispositivos móviles:** Gestionan los recursos limitados sin consumirlos

Funciones de los SSOO

Los SSOO deben de ser programas **eficientes, fiables, fáciles de usar, ligeros y portables a varias arquitecturas**. Además, cumplen las siguientes características

- **Gestionan los recursos del ordenador:** Procesador, memoria, E/S. Establece protección entre los usuarios y el SO se encarga de la contabilidad del sistema y del *timesharing*. Además, no deben de consumir el HW disponible
- **Ejecución de servicios para programas:** Llamadas al sistemas (syscalls) y APIs. Gestión de procesos, ordenes de E/S, detección de errores
- **Interfaz con el usuario:** Textual o GUI
- **Funcionalidad del shell:** Manipular archivos y directorios, ejecución de programas, herramientas de desarrollo de programas, comunicación con la interfaz, comunicación con otros sistemas, control de acceso,...

Tipos de implementaciones de los SSOO

- **Monolíticos:** Sin estructura clara y definida. Todos los componentes están mezclados y sin organización. Son muy complicados de modificar y de ampliar. Implementa todas las funciones en un único programa
- **Estructurados**
 - **Sistemas por capas:** Cada capa ofrece servicios a la capa superior y usa servicios de la capa

inferior

- **Modelo cliente servidor (microkernel):** El núcleo del SO implementa las funciones básicas. El resto de funciones las proporcionan los servidores (procesos del sistema operativo/daemons)

Soporte HW

La mayoría de las tareas del SO se implementan por SW, mientras que otras se implementan con HW. Algunas de las herramientas de apoyo al SW son

Modos de ejecución. Modo Kernel vs usuario

Para evitar que los programas accedan a secciones críticas de otros programas o del SO, se limitan las posibilidades de la máquina ofrecida a cada programa. La mayoría de aplicaciones se ejecutarán en el modo usuario, a excepción del SO que se ejecutará en modo Kernel. El cambio de modo se produce mediante una *syscall*, donde el SO realizará la tarea en modo Kernel. El modo usuario tiene restricciones en determinadas **zonas de memoria (incluyendo E/S), Instrucciones y Registros de la CPU**

Soporte de interrupciones

Sirven de apoyo al mecanismo de transferencia de control. Si llega una interrupción y está habilitada se completa la instrucción máquina en curso y se procede a tratar la interrupción. Pase lo que pase, se almacena **PC**. Existen varios tipos de interrupciones:

- SW: *syscalls*
- HW: Periféricos, de reloj, reset,...
- Excepciones generadas dentro del procesador
 - SW: División por 0, overflow,...
 - HW: Sobre calentamiento, pico de tensión,...

Acceso Directo a Memoria (DMA)

El procesador delega las operaciones de E/S al controlador DMA, de tal forma que la CPU se encarga de la ejecución de varios programas mientras que el DMA carga los datos en memoria. Para realizar su tarea, necesita conocer **la dirección de memoria, el número de palabras que tiene que leer y la dirección del dispositivo**

Reloj del computador

Fecha y hora del sistema

Jerarquía de memoria: Memoria virtual (MMU)

Permite simular una memoria mucho más grande. Para ello el SO utiliza unas páginas donde almacena la relación entre la dirección física y la dirección virtual del programa.

Preguntas de teoría

- **No se considera un requisito hardware para un SO moderno:** El bus PCI
- **La técnica de timesharing permite:** Mejorar la interactividad con el usuario. Es una técnica proporcionar concurrencia de procesos
- **Las interrupciones que cambian el modo de ejecución de la CPU son:** Hardware y software
- **Siempre se almacena independientemente de la interrupción:** PC
- Un SO de tiempo compartido es multiprogramado, pero un sistema multiprogramado no tiene por qué ser de tiempo compartido