

Introducción

Departamento de Automática
Universidad de Alcalá

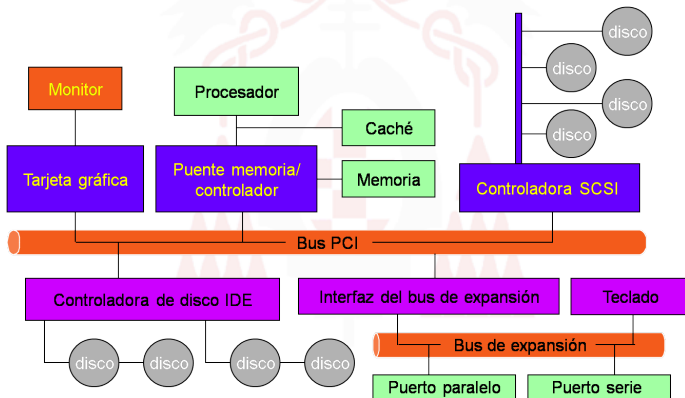


/gso>

Índice

- 1 Introducción a la asignatura
 - Una nueva máquina
 - ¿Cómo usar la máquina?
 - El Sistema Operativo
- 2 Los Sistemas Operativos dentro de la Ingeniería Informática
 - Esquema general
 - Arquitectura de Computadores: Conceptos básicos
- 3 Evolución histórica de los SSOO
 - Modelo de máquina desnuda
 - Monitor simple residente
 - Sistemas de procesamiento por lotes
 - Multiprogramación
 - Sistemas de tiempo compartido
 - Sistemas de tiempo real
 - Sistemas distribuidos


¿Qué aporta esta máquina?



Problemas para utilizar la máquina

- Cada elemento hardware es diferente y su uso es complicado.
 - ¿Tengo que aprender a usar cada uno de los dispositivos?
 - ¿Y si el fabricante cambia el modelo? ¿tendré que volver a aprender cómo se usa?
- Varios usuarios pueden compartir el mismo hardware
 - Recursos limitados, ¿quién los repartirá?
 - ¿Y si dos o más participantes quieren el mismo recurso?
 - ¿Cómo se asegura la privacidad?
 - ¿Los demás pueden ver y alterar mis datos?
- ¿Podrá utilizar la máquina alguien que no sepa lo que es el hardware ni el software?

¿Qué quiere el usuario?



```
Microsoft(R) Windows 98
(C)Copyright Microsoft Corp 1981-1998.

C:\WINDOWS>dir /?
Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/P] [/W] [/A[:attributes]]
[/O[:lsortorder]] [/S] [/B] [/L] [/U] [/4]

[drive:][path][filename]
    Specifies drive, directory, and/or files to list.
    <Could be enhanced file specification or multiple filespecs.>
/P
    Pauses after each screenful of information.
/W
    Uses wide list format.
/A
    Displays files with specified attributes.
attributes
    D Directories          R Read-only files
    H Hidden files        A Files ready for archiving
    S System files        - Prefix meaning not
/O
    List by files in sorted order.
sortorder
    N By name <alphabetic>      S By size <smallest first>
    E By extension <alphabetic> D By date & time <earliest first>
    G Group directories first   - Prefix to reverse order
    A By Last Access Date <earliest first>
/S
    Displays files in specified directory and all subdirectories.
/B
    Uses bare format <no heading information or summary>.
/L
    Uses lowercase.
/U
    Verbose mode.
/4
    Displays year with 4 digits <ignored if /U also given>.

Switches may be preset in the DIRCMD environment variable. Override
preset switches by prefixing any switch with - <hyphen>--for example, /-W.

C:\WINDOWS>
```

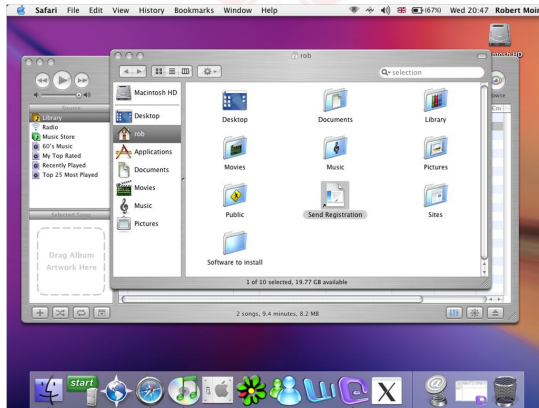
Introducción a la asignatura

Los Sistemas Operativos dentro de la Ingeniería Informática
Evolución histórica de los SSOO

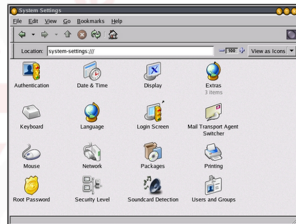
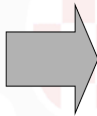
Una nueva máquina

¿Cómo utilizar la máquina de la mejor forma posible?
El Sistema Operativo

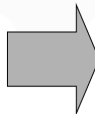
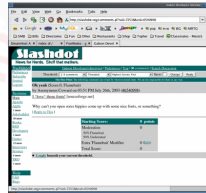
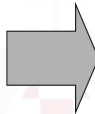
¿O mejor esto?



Mediante abstracciones



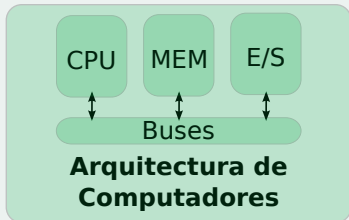
Más abstracciones



Algunas definiciones

- S. Sánchez: Un SO es un conjunto de programas que por medio de abstracciones ponen el hardware del ordenador, de modo seguro, a disposición del usuario
- H. Deitel: un SO es un programa que actúa como interfaz entre el usuario de un ordenador y el HW del mismo, ofreciendo el entorno necesario para que el usuario pueda ejecutar programas
- Katzan: conjunto de programas y datos que ayudan a crear otros programas y a controlar su ejecución
- Madnik y Donovan: conjunto de programas que gestionan los recursos del sistema, optimizan su uso y resuelven conflictos

Esquema general



Compiladores



Sistemas Operativos



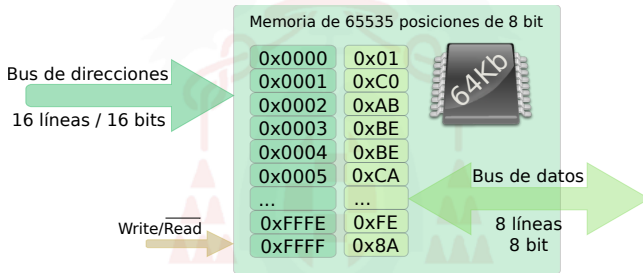
Windows

Memoria (1/2)

Características

- Permite el almacenamiento de datos y programas en un computador.
- Es una amplia tabla en la que cada dato se almacena en una posición.
- El número de posiciones de la tabla determina el tamaño de la memoria.
- El tamaño del dato almacenado y el número de posiciones de la tabla dependen del tipo de diseño.
- Los programas deben estar cargados en memoria principal para ser ejecutados.

Memoria (2/2)



Ejemplo

CARGAR R1, 0x0003

INCREMENTAR R1

GUARDAR R1, 0x0005

Unidad Central de Procesos (CPU)

Concepto

CPU

- CPU=procesador=microprocesador (en un chip)
- Consta de:
 - **ALU (Unidad Aritmético-Lógica)**: realiza funciones aritméticas y lógicas y
 - **Unidad de Control**: interpreta las instrucciones
- Controla al resto de componentes del sistema y se encarga de ejecutar las instrucciones de los programas.

Unidad Central de Procesos (CPU)

Modelo de programación

El modelo de programación incluye todos los elementos que la CPU pone a disposición de los programadores.

- Elementos de almacenamiento: registros generales, contador de programa, registro de estado, puntero de pila, mapa de memoria y mapa de E/S.
- Juego de instrucciones y modos de direccionamiento.

Modelo de programación

Registros de la CPU (1/2)

Los registros son los elementos de almacenamiento que existen dentro de la CPU.

- Tipos de registros:
 - Registros de propósito general y de coma flotante.
 - Registros de datos.
 - Registros de direcciones.
 - **Puntero de pila**: registro que apunta a la cabecera de la pila (operaciones *push* y *pop*.)
 - Registros de control y de estado.

Modelo de programación

Registros de la CPU (2/2)

- Los **registros de control** se emplean para determinar el funcionamiento de la CPU:
 - **Contador de programa**: contiene la dirección de la siguiente instrucción a ejecutar.
 - **Palabra de estado del programa (PSW)**: registro(s) con *flags* (acarreo, desbordamiento, etc.) fijados por el hardware de la CPU.

Modelo de programación

Juego de instrucciones

- Una **instrucción máquina** es una secuencia de bits que representa una operación a realizar y dónde se encuentran los operandos necesarios. Cada instrucción tiene asignado un código máquina.
- A los códigos máquina se le asigna un mnémónico, más cercano al lenguaje natural.
- Las **instrucciones** pueden necesitar un número variable de operandos.
- No todo el mundo puede ejecutar todas las instrucciones.
- El juego de instrucciones define qué operaciones puede realizar la CPU (¿puede la CPU multiplicar matrices de números?).

Modelo de programación

Tipos de instrucciones

Tipos de instrucciones

- Aritméticas: SUMAR, RESTAR, ...
- Lógicas: Y, O, NEGAR, ...
- Transferencia de datos: MOVER, CARGAR, ALMACENAR , APILAR, EXTRAER, ...
- Control/Salto: LLAMADA A RUTINA, RETORNO DE RUTINA, SALTO INCONDICIONAL, SALTO SI NO SON IGUALES, ...
- E/S: ENTRADA, SALIDA

⇒ Ejemplo:

MOVER \$2030, R3

SUMAR #4, \$2030

SALTO_SI_NO_IGUAL \$1000

Modelo de programación

Tipos de instrucciones

Tipos de instrucciones

- Aritméticas: SUMAR, RESTAR, ...
- Lógicas: Y, O, NEGAR, ...
- Transferencia de datos: MOVER, CARGAR, ALMACENAR , APILAR, EXTRAER, ...
- Control/Salto: LLAMADA A RUTINA, RETORNO DE RUTINA, SALTO INCONDICIONAL, SALTO SI NO SON IGUALES, ...
- E/S: ENTRADA, SALIDA

⇒ Ejemplo:

MOVER \$2030, R3

SUMAR #4, \$2030

SALTO_SI_NO_IGUAL \$1000

Modelo de programación

Modos de direccionamiento

Modos de direccionamiento

Formas de especificar e interpretar un operando de una instrucción.

Tipos de direccionamiento

- *Implícito*: el propio código de operación indica sobre qué operando actúa la operación.
- *Explícito*
 - *Inmediato*: el operando se incluye en la propia instrucción.
 - *Directo*: el operando se referencia con su dirección en memoria.
 - Otros: *Indirecto*, *Por Registro*, etc.

Unidad Central de Procesos (CPU)

Ciclos de ejecución de una instrucción

Ciclo de ejecución

- Búsqueda de la instrucción máquina.
- Interpretar la instrucción leída.
- Leer los datos de memoria referenciados en la instrucción.
- Ejecutar la instrucción.
- Almacenar los resultados de la ejecución.

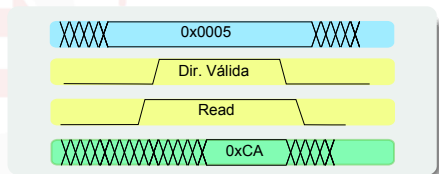
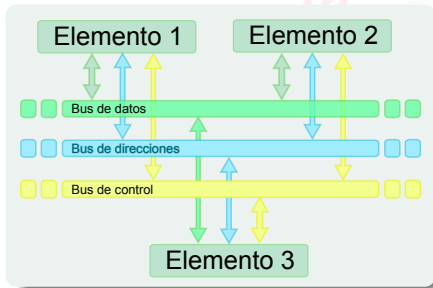
Buses

Descripción general

- Son los encargados de conectar entre si los distintos componentes del sistema.
- Inicialmente eran pasivos (cables), pero actualmente son elementos muy complejos.
- Un bus se define por varios parámetros:
 - Sus conectores y características físicas.
 - Su topología (serie, paralelo, estrella, ...).
 - Sus señales y protocolos.

Buses

Buses paralelo



Entrada y salida

Objetivo y características generales

Objetivo: Comunicación con el mundo exterior para ...

- Obtener los programas que hay que ejecutar (y colocarlos en memoria).
- Obtener los datos que hay que procesar.
- Comunicar con usuarios humanos y con otras máquinas.

Características

- La naturaleza de cada dispositivo es totalmente diferente.
- Todos deben comunicar con la CPU (buses).
- Es necesario establecer una arquitectura común: parte específica y parte común (controlador).

Entrada y salida

Arquitectura de un dispositivo de E/S: Controladores

- Son elementos hardware.
- Se conectan a los buses del computador y ofrecen varios registros:
 - Registro de datos.
 - Registro de control.
 - Registro de estado.
- Algunos dispositivos complejos utilizan otros elementos.

Entrada y salida

Técnicas de E/S

- Los dispositivos necesitan atención (¿urgente?)
- Dos aproximaciones:
 - Que la CPU le pregunte al dispositivo
 - ¿Con que frecuencia?
 - Carga de trabajo adicional para la CPU.
 - Ocupa los buses y los dispositivos de forma innecesaria.
 - Sencillo, no necesita hardware adicional.
 - Que el dispositivo **interrumpa** a la CPU cuando lo necesite.
 - Necesita hardware adicional.
 - Necesita que la CPU disponga del mecanismo de interrupciones.

Entrada y salida

Mecanismo de interrupciones

- 1 La CPU ejecuta un programa normalmente.
- 2 Un dispositivo requiere atención y **solicita una interrupción** a la CPU.
- 3 La CPU deja de ejecutar el programa y pasa a ejecutar **otro programa** denominado ISR.
- 4 La ISR atiende al dispositivo y da por finalizada la interrupción.
- 5 La CPU reanuda la ejecución del programa **en el punto en el fue interrumpida**.

Modelo de máquina desnuda

Objetivo

Ejecutar programas almacenados en memoria.

- Se programaba directamente sobre el hardware.
- No existía nada similar al SO.
- Los usuarios introducían código máquina mediante interruptores.
- Problemas:
 - El usuario debía conocer todo el hardware.
 - Un cambio mínimo en el hardware invalidaba todo el programa.
 - Se repite entre programas.

Monitor simple residente

Objetivo

Reutilizar código.

- El código común se agrupa dentro de un *monitor simple residente*
 - El monitor es el primer germen de SO.
 - Agrupa fundamentalmente rutinas de E/S.
 - Un monitor es, esencialmente, un manejador de dispositivo o *driver*.
- El programador no tiene que programar directamente la E/S.
- Se puede, por ejemplo, cambiar el lector de tarjetas sin cambiar los programas, basta con cambiar el monitor.
- Problema: Se pierde mucho tiempo entre la finalización de un proceso y el lanzamiento del siguiente.

Sistemas de procesamiento por lotes (sistemas *batch*)

Objetivo

Reducir los tiempos de espera de E/S en la carga de programas.

- Para la E/S se utilizan ordenadores de bajo coste dedicados.
- La ejecución de los programas se realiza por medio de un ordenador de altas prestaciones y alto coste.

Sistemas de procesamiento por lotes (sistemas *batch*)

Modo de operación

- Un operario introduce las tarjetas perforadas en un ordenador de bajo coste.
- Una vez se termina, se obtiene una cinta que es cargada en el ordenador principal.
- El ordenador principal va ejecutando los programas de la cinta mientras el operario introduce nuevos programas.
- Cuando el ordenador principal termina, el operario retira la cinta y la introduce en un tercer ordenador que imprime los resultados.

⇒ Problema: hace falta la intervención de un operario humano.

Multiprogramación

Objetivo y requerimientos hardware

Objetivo

Solapar las operaciones de E/S con la utilización de la CPU en la misma máquina.

- Supone un salto de gran importancia y dificultad: tener más de un programa cargado en memoria a la vez.
- Mientras se realiza una operación de E/S el microprocesador ejecuta otro programa de los que están en memoria.
- Surgen nuevas necesidades hardware:
 - Interrupciones.
 - Acceso directo a memoria (DMA).

Multiprogramación

Requerimientos del Sistema Operativo

- Surgen la mayoría de los problemas clásicos de los SSOO:
 - Gestión de la memoria.
 - Planificación del procesador.
 - Planificación de los dispositivos.
 - Protección de la ejecución de diferentes programas.
 - Control de la concurrencia.
- Problemas:
 - ⇒ Un programa en ejecución puede monopolizar CPU.
 - ⇒ Poco adecuado para sistemas interactivos.

Sistemas de tiempo compartido

Objetivo

Garantizar a los procesos el acceso equitativo a la CPU.

- Protección contra monopolización de la CPU: interrupciones.
- El tiempo de ejecución asignado a cada programa se divide en quantum.
- Si un programa en ejecución agota su quantum, la CPU pasa a ejecutar otro programa (round-robin).
- Los SSOO de tiempo compartido son los habituales actualmente en la informática de consumo.

Sistemas de tiempo real

Concepto y características

Objetivo

Garantizar que los programas se ejecutan dentro de un plazo de tiempo acotado.

- Para satisfacer restricciones de tiempo real no basta con tener un hardware más potente.
- Tiempo real no es sinónimo de rápido.
- Suelen encontrarse en sistemas empotrados.
- Dos tipos de aplicaciones de tiempo real:
 - Duras.
 - Blandas.

Sistemas distribuidos

Concepto y características

Objetivo

Repartir sus funciones entre varias máquinas, dando la impresión ser una única máquina.

- Es distribuido si está formado por varios elementos que cooperan para dar un servicio único.
- Un SOD no es un sistema simplemente replicado. Hay mentalidad de grupo.
- Si puedes decir qué máquinas estás usando, dónde corren los programas, dónde están los archivos, entonces NO es SOD.
- Introducen el concepto de *middleware*.

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.