

SISTEMAS OPERATIVOS

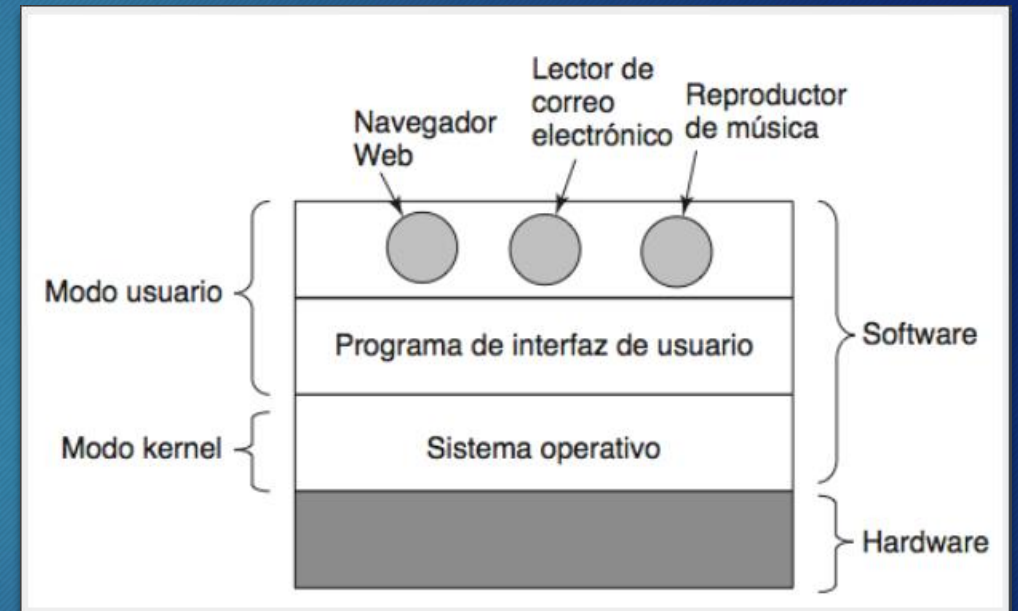
Ingeniería civil informática

Gonzalo Carreño

gonzalocarrenob@Gmail.com

INTRODUCCIÓN

- Un computador actual consta de varios elementos de hardware que lo convierten en un sistema complejo de administrar. Por esto existe el S.O. que es quien proporciona a los programas de usuario un "modelo" de computador más simple, además de encargarse de la administración de los recursos.



- Finalmente un computador se compone del hardware y software. El software se divide en dos modos: Kernel y Usuario.
- El S.O. es la pieza más importante de software, se ejecuta en el modo Kernel lo que le permite tener acceso total al hardware y a todas las instrucciones que la máquina pueda ejecutar.
- Todo el software que utilizamos y creamos (tanto shell como GUI) se ejecutan en modo Usuario, en donde se tiene un acceso restringido a las instrucciones de máquina.

- La interfaz de usuario GUI o shell es el nivel más bajo de software en modo usuario, sobre el se ejecutan otras piezas de software como: Juegos, Navegador, Reproductor, etc.



¿QUÉ ES UN S.O?

- S.O. como máquina virtual extendida: el desarrollador "promedio" no desea involucrarse en el proceso y funcionamiento a bajo nivel de algún elemento del hardware, si no, desea una "abstracción" de alto nivel que permita operar. (ej: el concepto "archivos" para referirse al contenido de un hdd).
- El trabajo de un S.O es crear buenas abstracciones, ocultar el hardware y presentar a los "programas" abstracciones agradables, elegantes, simples y consistentes

- S.O. como administrador de recursos: La anterior perspectiva es conocida como Top-Down. En el caso de considerar el S.O. como la herramienta para administrar las "piezas" de un sistema complejo es conocida como Bottom-up. En esta perspectiva el S.O. es encargado de entregar asignaciones ordenadas y controladas de cada elemento de hardware, entre los diversos programas que solicitan estos recursos.
- Manejar concurrencia en el consumo de recursos: Competencia. La tarea principal es llevar un registro de qué programa está utilizando qué recurso, contabilizar su uso y entregarlo a quien lo requiere.

Sistemas operativos más comunes

Windows

- Es el sistema operativo mas utilizado del mundo, lo que significa que existen miles de aplicaciones.
- Gran compatibilidad con los dispositivos hardware del mercado.
- Facilidad de uso



Linux

- Seguridad y estabilidad
- Gratuito
- Muchas aplicaciones de softwares libre



Macintosh

- Seguridad y estabilidad
- Simplicidad
- Ideal para el diseño grafico



HISTORIA DE LOS S.O

- Los S.O. han evolucionado con los años, desde estar directamente relacionados con la arquitectura donde se ejecutarían hasta los S.O. modernos capaces de ejecutarse en cualquier máquina.
- ¿Cuál fue la primera computador digital?
- La máquina analítica de Babbage (1792 a 1871). Babbage notó que su máquina requería "software" por lo que contrató a una joven y prometedora matemática Ada Lovelace



LA PRIMERA GENERACIÓN: TUBOS DE VACÍO (1945 A 1955)

- Llegada la Segunda guerra mundial el progreso en construcción de máquinas de cálculo explotó, en esta época se construyeron las primeras máquinas consideradas computadoras digitales. Esta explosión ocurrió en varias universidades de forma simultánea:
 - Iowa State University: Tubos de vacío.
 - Berlín: Z3 usando "relé".
 - Inglaterra: Colossus.
 - Harvard: Mark I.
 - Pennsylvania: ENIAC.
- En estos tiempos un grupo de Ingenieros diseñaban, construían, programaban y mantenían las máquinas

- Todo se "programaba" en lenguaje de máquina, o creando circuitos eléctricos conectando miles de cables a distintos tableros para controlar las funciones de la máquina.
- NO EXISTIAN LENGUAJES DE PROGRAMACIÓN



LA SEGUNDA GENERACIÓN: TRANSISTORES (1955 A 1965)

- La creación del transistor significó un nuevo paso de gigantes en la construcción de máquinas digitales, siendo más confiables en el proceso de fábrica y funcionamiento. MAINFRAMES
- Junto con este gran paso nacen los primeros lenguajes de programación: Ensamblador y FORTRAN. Pero aún se usaban tarjetas perforadas para entregar el programa al computador.
- Nace el sistema de procesamiento por lotes: Mantener una bandeja llena de trabajos en el cuarto de entrada para ser pasados a una cinta magnética y ejecutar de manera continua los distintos procesos.

- Estas computadoras eran utilizadas principalmente para cálculos científicos y de ingeniería: Ecuaciones diferenciales parciales, Integrales de alta complejidad y análisis estadísticos.



TERCERA GENERACIÓN: CIRCUITOS INTEGRADOS Y MULTIPROGRAMACIÓN (1965 A 1980)

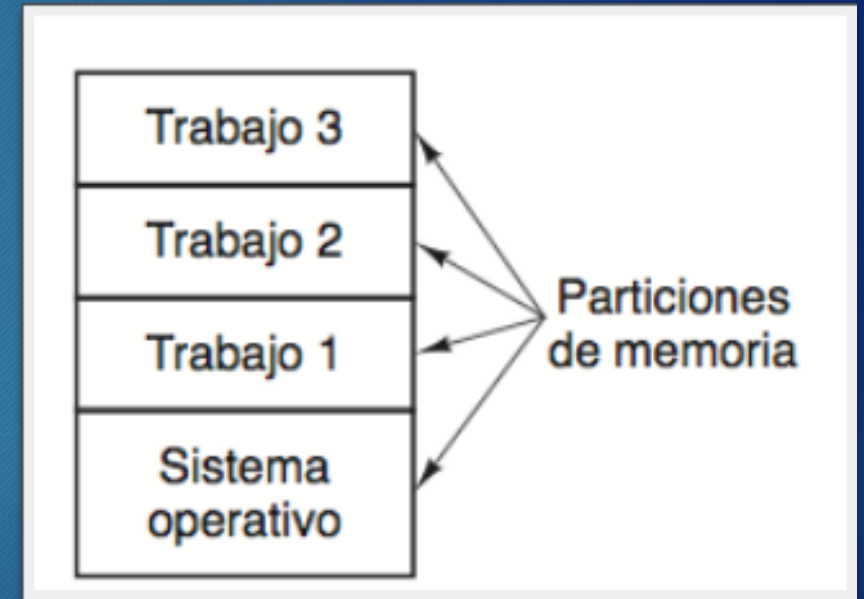
- Al principio existían dos líneas de producción: Computadoras científicas y computadoras comerciales (orientadas a ordenar cintas, imprimir datos, etc).
- IBM crea la solución: System/360 que consistía en una familia de máquinas de distinto poder pero compatible en el software siendo capaces de efectuar ambos trabajos (científico y comercial).
- Su mayor fortaleza se convirtió en su mas grande debilidad: No podían dar soporte de software a tal gama de hardware y requerimientos.

IBM S/360



Consola del operador de un System/360 Model 65, con válvula de registro, lámparas y switches (mitad de la fotografía), y arriba a la derecha, un interruptor de emergencia rojo.

- Aún así dejaron huella creando conceptos como la multiprogramación. Antes de esto cuando un trabajo se detenía, a la espera de alguna operación externa, la CPU quedaba inactiva, lo que generaba pérdidas económicas. Nace la partición de memoria.
- Bell y General Electric deciden crear un modelo de computadora basada en el concepto de la electricidad: "Si requieres energía, solo requieres conectarte a la pared". Nace **MULTICS**



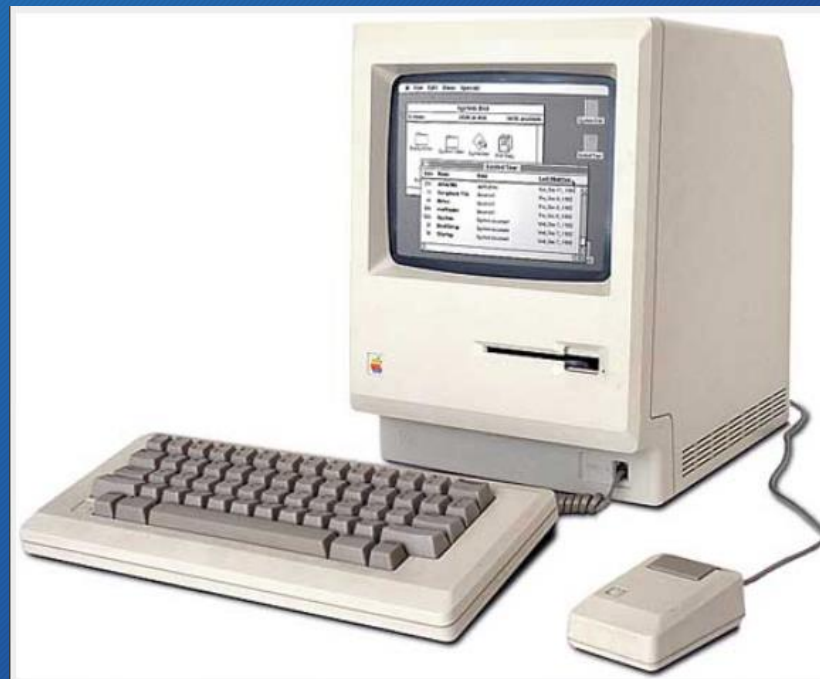
- MULTICS intentaba ser una "enorme computadora que podía proporcionar poder de cómputo a todos los usuarios de Boston" Fue un éxito parcial, la máquina era comparable aun Intel 386.
- Fue demasiado ambicioso, pero MIT persistió y consiguió algo funcional usado por algunos fieles clientes.
- Aquí irrumpe Ken Thompson quien quiso crear un MULTICS más simple para una pequeña máquina del Bell Labs (PDP-7): Nace UNIX.

- Thompson libera el código fuente permitiendo la creación de varias versiones como System V (AT&T) y BSD (Berkeley). Para estandarizar IEEE crea el protocolo POSIX, logrando compatibilidad entre las versiones UNIX.
- En 1987 nace MINIX basado en UNIX para el mundo académico
- La búsqueda de un UNIX gratuito y no educacional (minix) llevó a la creación de Linux en 1991 (escrito en Minix).

CUARTA GENERACIÓN: COMPUTADORAS PERSONALES (1980)

- Al crearse los circuitos LSI o "chip" se permitió la miniaturización de los componentes de una computadora, permitiendo la manufacturación de computadores de alto poder a bajo costo.
- En 1980 IBM crea la IBM PC, el software que lo controlaba era el interprete de BASCI de Gates, luego de malas decisiones de negocio de Digital Research IBM vuelve con Gates y compran la licencia de uso de MS-DOS.
- Ya en 1960 existía el concepto de GUI (Stanford - Engelbart).
- Xerox PARC construyeron la primera interfaz gráfica. Pero Xeros no vio su potencial quedando dormida en los laboratorios.

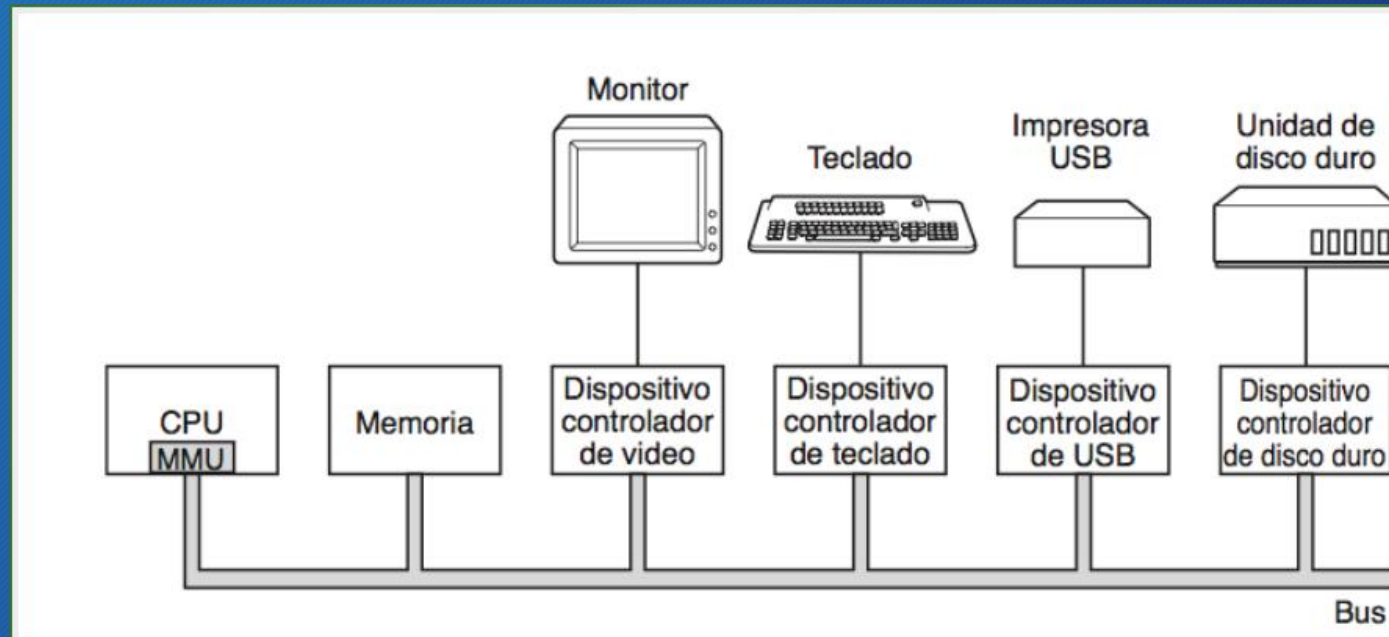
- Steve Jobs supo de la existencia de este experimento, naciendo Lisa (jamás vendido) y el Macintosh.



- Esto hace nacer a Windows que inicialmente (10 años) era más una shell que un verdadero S.O (se ejecutaba sobre DOS).
- Además comienzan a nacer los S.O distribuidos en red.

HARDWARE

- Un S.O. está relacionado con el hardware donde se ejecuta.
- Extiende las instrucciones y administra sus recursos.
- Conceptualmente una computadora es un modelo simple:



PROCESADOR

- Es el "cerebro" de toda computadora. Obtiene instrucciones desde la memoria y las ejecuta retornando resultados.
- Cada CPU tiene un conjunto específico de instrucciones posibles de ejecutar, generando incompatibilidad entre diversas arquitecturas, modelos e incluso "marcas".
- Toda CPU contiene "registros", pequeños espacios de memoria de alta velocidad ocupados para almacenar variables temporales, instrucciones, etc.
- El S.O. debe estar al tanto de todos los registros. Cuando la CPU se multiplexa en el tiempo el S.O. detiene un proceso/trabajo para (re)iniciar otro. En cada detención de un proceso el S.O. debe almacenar los registros de la CPU para volver a utilizarlos.