



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de
Ingeniería y Ciencias Sociales y Administrativas

Sistemas Operativos

Unidad I:

Estructura de los sistemas operativos

Introducción

M. en C. Hermes Francisco Montes Casiano
hermes.upiicsa@gmail.com



Objetivos

Objetivos

- 1 Proporcionar una visión general de los principales componentes de los sistemas operativos.
- 2 Proporcionar una panorámica sobre la organización básica de un sistema de cómputo.



Índice

- 1 **Introducción**
Introducción
- 2 Organización de una computadora
- 3 Llamadas al sistema
- 4 Estructura del sistema operativo
- 5 Evolución de los sistemas operativos





Introducción

Definición

Un **sistema operativo** es un programa que administra el hardware de una computadora. Ofrece una infraestructura de gestión, aislamiento y protección de los recursos de un sistema de cómputo.

Es el programa menos importante

No realiza un trabajo útil para el usuario, lo que hace es permitir que otros procesos hagan su trabajo.

- Proporciona las bases para los programas de aplicación y actúa como un intermediario entre el usuario y el hardware de una computadora.





Introducción

- Proporciona las bases para los programas de aplicación y actúa como un intermediario entre el usuario y el hardware de una computadora.
- Un aspecto sorprendente de los sistemas operativos es la gran variedad de formas para la realización de una tarea.
- Los sistemas operativos para *mainframe* están diseñados principalmente para optimizar el uso del hardware.
- Los sistemas operativos para PC soportan desde complejos juegos hasta aplicaciones de negocio.
- Algunos sistemas se diseñan para ser prácticos, otros para ser eficientes y otros para ambas cosas.



¿Qué es un sistema operativo?

Un sistema operativo es una parte importante de casi todo sistema de cómputo. De manera general un sistema de cómputo puede dividirse en cuatro componentes:

- Hardware
- Sistema Operativo
- Programas de aplicación
- Usuarios



¿Qué es un sistema operativo?

- 1 El **hardware** proporciona los recursos básicos de cómputo
 - La Unidad Central de Procesamiento
 - La memoria
 - Los dispositivos de entrada y salida
- 2 Los **programas de aplicación** definen las formas en que el hardware se utiliza para resolver los problemas de cómputo de los usuarios.
- 3 Los **usuarios** una persona, una máquina, otras computadoras, mismas que intentan resolver distintos problemas.
- 4 El **sistema operativo** controla y coordina el uso de *hardware* entre los diversos programas de aplicación por parte de los diversos usuarios.



¿Qué es un sistema operativo?

Se puede abstraer un sistema de cómputo como hardware, software y datos.

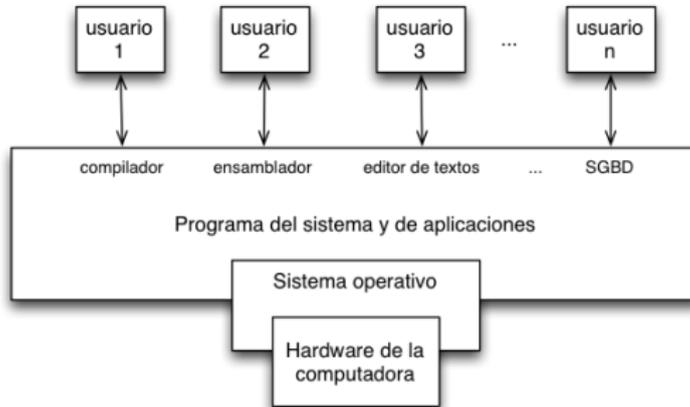


Figura: Esquema de los componentes de un sistema informático.



¿Qué es un sistema operativo?

¿Que proporciona?

El sistema operativo proporciona los medios para el *uso apropiado* de los recursos en la operación de un sistema de cómputo.

Analogía

Al igual que un gobierno, el sistema operativo por sí mismo no realiza alguna función útil. Simplemente proporciona un ambiente dentro del cual otros programas puedan realizar un trabajo útil.



¿Qué es un sistema operativo?

Asignador de recursos

Desde el punto de vista de la computadora, es el programa más íntimamente relacionado con el hardware. Administra los recursos de ésta: tiempo de CPU, espacio de memoria, espacio de almacenamiento de archivos, E/S, etc.

Programa de control

Otro punto de vista hace hincapié en la necesidad de controlar los distintos dispositivos de E/S y programas de usuario. Como programa de control, gestiona la ejecución de los programas de usuario para evitar errores y optimizar el uso de la computadora.



¿Qué no es es un sistema operativo?

- Un conjunto de programas para administrar archivos.
- El mecanismo para que los usuarios lancen programas a ejecución.
- El entorno gráfico o de consola con el que interactúa el usuario.



¿Qué ofrece un sistema operativo?

- Los sistemas operativos existen porque ofrecen una forma razonable de resolver el problema de crear un sistema de cómputo utilizable.
- Dado que el hardware no es fácil de utilizar, se desarrollaron programas de aplicación.
- Dichos programas necesitan operaciones comunes.
- Las operaciones habituales de control y asignación de recursos se incorporan en una misma pieza de software: el **sistema operativo**.

Definición

Es el programa que se ejecuta continuamente en la computadora (generalmente denominado **kernel**).



Índice

1 Introducción

2 Organización de una computadora

Funcionamiento de una computadora

Estructura de almacenamiento

Estructura de E/S

3 Llamadas al sistema

4 Estructura del sistema operativo

5 Evolución de los sistemas operativos



Funcionamiento de una computadora

Acceso a la memoria

- Una computadora moderna consta de una o más CPU's y de una serie de controladores de dispositivos conectados a través de un bus común que proporciona acceso a la memoria compartida.
- Cada controlador de dispositivo se encarga de un tipo de dispositivo específico.
- La CPU y los controladores de dispositivos pueden funcionar de forma concurrente, compitiendo por los ciclos de memoria.
- Para asegurar el acceso de forma ordenada a la memoria compartida, se proporciona un controlador de memoria.



Funcionamiento de una computadora

Arranque

- Para que una computadora comience a funcionar es necesario que tenga un programa de inicio que ejecutar.
- El **programa de arranque**, suele ser simple. Normalmente se almacena en memoria ROM o en una memoria EEPROM y se le conoce como **firmware**.
- El **firmware** inicializa todos los aspectos del sistema, desde los registros de la CPU hasta los controladores de dispositivo y el contenido de la memoria.
- El programa de arranque debe saber cómo cargar el sistema operativo e iniciar la ejecución de dicho sistema.
- Para conseguir este objetivo, el programa de arranque debe localizar el kernel del sistema operativo y cargarlo en memoria.



Funcionamiento de una computadora

Mecanismo de interrupciones

- Después, el sistema operativo comienza ejecutando el primer proceso, por ejemplo *init*, y espera a que se produzca algún suceso.
- La ocurrencia de un suceso normalmente se indica mediante una **interrupción** bien de *hardware* o de *software*.
 - ① El hardware puede activar una interrupción en cualquier instante enviando una señal a la CPU.
 - ② El software puede activar una interrupción ejecutando una operación especial denominada **llamada al sistema**.
- Cuando se interrumpe a la CPU, deja lo que esta haciendo e inmediatamente transfiere la ejecución a la rutina que atiende a la interrupción.



Funcionamiento de una computadora

Mecanismo de interrupciones

- La rutina de servicio de la interrupción se ejecuta y, cuando ha terminado, la CPU reanuda la operación que estuviera haciendo.
- Cada diseño de computadora tiene su propio mecanismo de interrupciones, aunque hay algunas funciones comunes.
- La interrupción debe transferir el control a la rutina de servicio apropiada a la interrupción.
- Debido a que las interrupciones deben tratarse rápidamente, para agilizar su atención, sólo es posible un número predefinido de interrupciones.
- Para el tratamiento de las interrupciones se dispone de una tabla de punteros a las rutinas de interrupción.



Funcionamiento de una computadora

Mecanismo de interrupciones

- La tabla de punteros se almacena en la zona inferior de la memoria, denominada **vector de interrupciones**. Estas posiciones almacenan las direcciones de las rutinas de servicio a la interrupción para los distintos dispositivos.
- El **vector de interrupciones** se indexa mediante un número de dispositivo unívoco.
- La arquitectura de servicio de las interrupciones también debe almacenar la dirección de la instrucción interrumpida.



Funcionamiento de una computadora

Mecanismo de interrupciones

- Las arquitecturas recientes almacenan la dirección de retorno en la pila del sistema.
- Después de atender a la interrupción, la dirección de retorno guardada se carga nuevamente en el contador de programa y el cálculo se reanuda como si la interrupción no se hubiera producido.



Estructura de almacenamiento

- Los programas de la computadora deben hallarse en la memoria principal (RAM) para ser ejecutados.
- La memoria principal es la única área de almacenamiento de gran tamaño a la que el procesador puede acceder directamente.
- El acceso a la RAM se implementa con una tecnología de semiconductores denominada DRAM que forma una matriz de palabras de memoria.
- La interacción se consigue a través de una secuencia de cargas o almacenamiento de instrucciones en direcciones específicas de memoria.



Estructura de almacenamiento

- Un ciclo típico de instrucción-ejecución, cuando se ejecuta en un sistema con una arquitectura **Von Neuman** consta de lo siguiente:
 - ① Extrae la instrucción de memoria y almacena la instrucción en el registro de instrucciones.
 - ② Se decodifica la instrucción y puede dar lugar a que se extraigan operandos de la memoria y se almacenen en algún registro interno.
 - ③ Se ejecuta la instrucción con los operandos necesarios.
 - ④ Se almacena el resultado de nuevo en memoria.



Estructura de almacenamiento

- Es deseable que los procesos y los datos residan en memoria de forma permanente, sin embargo no es posible debido a:
 - ① Normalmente, la memoria principal es demasiado pequeña como para almacenar todos los procesos y datos necesarios de forma permanente.
 - ② La memoria principal es un dispositivo de almacenamiento volátil.
- Por tal motivo, la mayor parte de los sistemas de cómputo proporcionan **almacenamiento secundario**.
- El requerimiento fundamental del almacenamiento secundario es que debe poder almacenar grandes cantidades de datos de forma permanente.
- Sólo debe utilizarse memoria cara cuando sea necesario y emplear memorias baratas y no volátiles cuando sea posible.



Estructura de almacenamiento

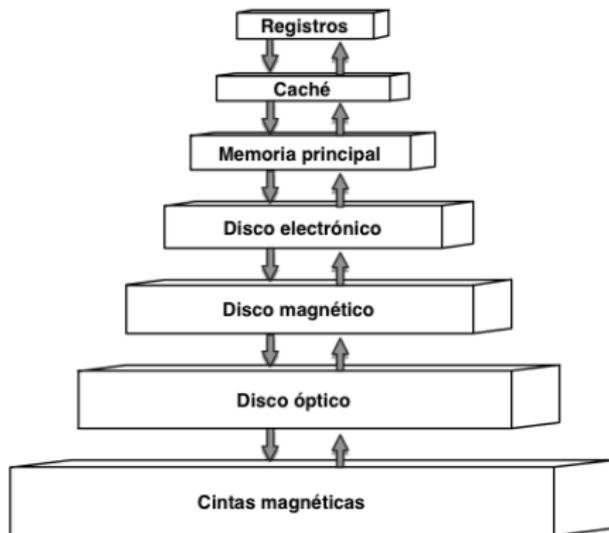


Figura: Jerarquía de dispositivos de almacenamiento.



Estructura de E/S

- Gran parte del código del sistema operativo se dedica a gestionar la entrada y salida.
- Cada controladora de dispositivo se encarga de un tipo específico de dispositivo.
- Dependiendo de la controladora, puede haber más de un dispositivo conectado.

Ejemplo

Siete o más dispositivos pueden estar conectados a la controladora SCSI.

Una controladora de dispositivo mantiene algunos búferes locales y un conjunto de registros de propósito especial.



Estructura de E/S

- Los sistemas operativos tienen un controlador de dispositivo (*driver*) para cada controladora de dispositivo.
- El hardware controlador de dispositivo es capaz de entenderse con la controladora hardware y presenta al resto del sistema operativo una interfáz uniforme para comunicarse con el dispositivo.
- Al iniciar una operación de E/S:
 - 1 El controlador de dispositivo carga los registros apropiados de la controladora hardware.
 - 2 La controladora examina el contenido de estos registros para determinar que acción realizar.
 - 3 La controladora inicia la transferencia de datos desde el dispositivo a su búfer local.
 - 4 A través de una interrupción la controladora hardware informa al *driver* que ha terminado la operación.



Estructura de E/S

- Al iniciar una operación de E/S:
 - 1 El controlador de dispositivo carga los registros apropiados de la controladora hardware.
 - 2 La controladora examina el contenido de estos registros para determinar que acción realizar.
 - 3 La controladora inicia la transferencia de datos desde el dispositivo a su búfer local.
 - 4 A través de una interrupción la controladora hardware informa al *driver* que ha terminado la operación.
- Una vez terminada la operación de E/S, el controlador devuelve el control al sistema operativo, de volviendo posiblemente datos, o un puntero a los datos, etc.



Estructura de E/S

- El manejo de E/S por medio de interrupciones es adecuado para transferir cantidades pequeñas de datos.
- Cuando se trata de grandes cantidades de información se utiliza acceso directo a memoria (DMA), para evitar que interfiera la CPU.

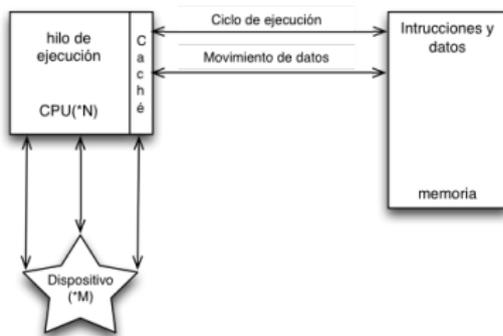


Figura: Funcionamiento de un sistema informático moderno.



Índice

- 1 Introducción
- 2 Organización de una computadora
- 3 Llamadas al sistema**
Servicios del sistema operativo
- 4 Estructura del sistema operativo
- 5 Evolución de los sistemas operativos





Llamadas al sistema

Llamadas al sistema

La interfáz entre el sistema operativo y los programas de usuario está definida por el conjunto de *operaciones extendidas* que el sistema operativo ofrece.

Las llamadas al sistema generalmente pertenecen a dos categorías:

- Las que se enfocan a los procesos
- Las que se ocupan del sistema de archivos



Procesos

- Cada proceso es básicamente un programa en ejecución. Un programa por sí sólo no es un proceso (entidad pasiva).
- Cada proceso tiene asociado un espacio de direcciones, es decir, una lista de posiciones de memoria desde algún mínimo (0) hasta un máximo, en las que el proceso puede leer y escribir.
- En el espacio de memoria se encuentra: el código, área de datos y la pila.
- Además se le asocia un conjunto de registros, que incluyen el contador del programa (**program counter**), el apuntador de la pila (**stack pointer**) y otros registros de hardware.



Procesos

- El *program counter* especifica la siguiente instrucción a ejecutar.
- La ejecución de un proceso es secuencial, la CPU ejecuta una instrucción de un proceso después de otra y así hasta completarlo.
- Un proceso multihilo tiene múltiples contadores de programa, cada uno de ellos a la siguiente instrucción que el hilo tiene que ejecutar.
- Un proceso es una unidad de trabajo en un sistema: procesos del sistema operativo y procesos del usuario.
- Todos los procesos pueden ejecutarse en un procesador de forma concurrente.



Procesos

- Los procesos que se encuentran en ejecución son almacenados en una tabla del sistema operativo llamada tabla de procesos.
- La tabla de procesos es un arreglo o una lista de estructuras de datos, cada una contiene toda la información de los procesos.
- Un proceso puede crear otros procesos, a estos procesos se les denomina hijos.
- Un mecanismo común de comunicación entre procesos son las *señales*. Las señales son el análogo en software a las interrupciones en hardware.



Procesos

En lo que respecta a la administración de procesos el Sistema Operativo es responsable de:

- 1 Crear y borrar procesos de usuario y del sistema.
- 2 Suspender y reanudar procesos.
- 3 Proporcionar mecanismos para la sincronización entre procesos.
- 4 Proporcionar mecanismos para la comunicación entre procesos.
- 5 Proporcionar mecanismos para el tratamiento de bloqueos.



Archivos

- Una función importante del SO es ocultar las peculiaridades de los discos y otros sistemas de E/S y presentar un modelo agradable y legible de archivos independientes de los dispositivos.
- Creación, eliminación, lectura y escritura de archivos.
- Un **directorio** es la forma utilizada para agrupar los archivos.
- Los archivos son organizados en forma de árbol.
- Las jeraquías de los procesos no son de gran profundidad (3 niveles), caso contrario a la jerarquía de los archivos.
- Los mecanismos de acceso a los archivos permiten que los archivos y directorios sean accedidos por un grupo más amplio que los propietarios.



Archivos

- Cada uno de los archivos que se encuentra dentro de la jerarquía del directorio puede determinarse mediante el **nombre de la ruta de acceso**.
- La parte superior de la ruta de acceso se le denomina *directorio raíz*.
- Cada proceso tiene activo un **directorio de trabajo**.
- Es importante tener un medio de protección que garantice la privacidad de los archivos de cada usuario.



Archivos

- En UNIX los archivos y directorios se protegen al asignarles un código de protección de *9 bits*.

$$drwxrwxr - x \quad (1)$$

- d = directorio
 - r = lectura
 - w = escritura
 - x = ejecución
- La estructura es la siguiente:

$$\underbrace{rwx}_{\text{usuario}} - \underbrace{rwx}_{\text{grupo}} - \underbrace{rwx}_{\text{el resto}} \quad (2)$$



Archivos

Estructura de árbol

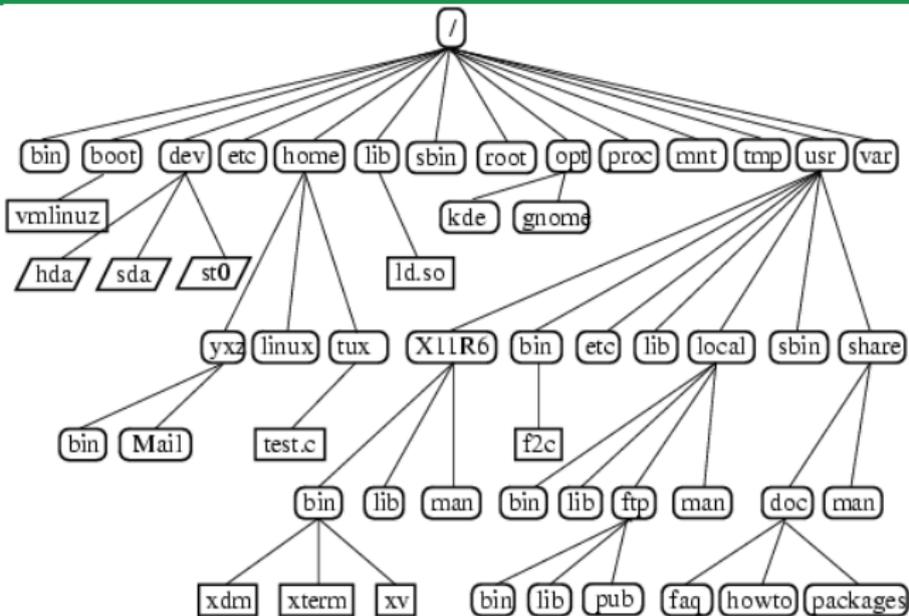


Figura: Estructura de un sistema de archivos



Archivos

Permisos

- En los sistemas operativos basados en Linux los permisos se modifican con el comando *chmod*.

Lectura	Escritura	Ejecución	Permiso
0	0	0	0 - Sin permisos
0	0	1	1 - Ejecución
0	1	0	2 - Escritura
0	1	1	3 - Escritura y ejecución
1	0	0	4 - Lectura
1	0	1	5 - Lectura y ejecución
1	1	0	6 - Lectura y escritura
1	1	1	7 - Lectura, escritura y ejecución



Acceso a archivos

- Antes de que un archivo pueda leerse o escribirse en él, debe abrirse.
- Si se permite el acceso a un archivo, el sistema regresa un entero llamado **descriptor de archivo**.
- Si se prohíbe el acceso regresa un código de error.
- En los SO basados en UNIX todos los dispositivos de E/S se abstraen en **archivos especiales**.
- Existen 2 tipos de archivos especiales:
 - Archivos especiales de bloque
 - Archivos especiales de carácter



Archivos especiales

1 Bloque:

- Se utilizan para modelar los dispositivos que constan de un conjunto de bloques de direccionamiento aleatorio (p.e. discos).
- Se utiliza el mismo mecanismo de protección para leer los demás archivos.

2 Carácter:

- Se utiliza para modelar los dispositivos que constan de un flujo de caracteres, en lugar de bloques de direccionamiento.

Descriptores

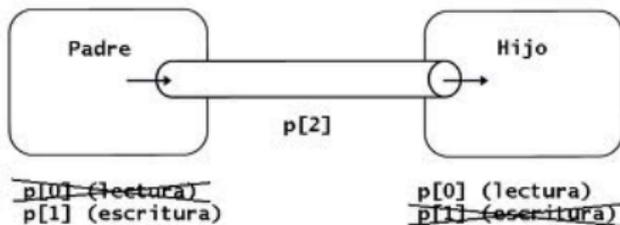
Al iniciar un proceso se establece descriptor de archivo 0 (standard input), el descriptor 1 (standard output) y el descriptor 2 (standard error).



Archivos especiales

Tuberías

- Una tubería es una especie de archivo utilizado por dos procesos para comunicarse entre sí.





Índice

- 1 Introducción
- 2 Organización de una computadora
- 3 Llamadas al sistema
- 4 Estructura del sistema operativo**
- 5 Evolución de los sistemas operativos



Estructura del sistema operativo

- Monolítico** Todas las operaciones del sistema operativo son ejecutadas por el mismo supero proceso.
- Microkernel** Este tipo de sistemas operativos cuentan con un núcleo con el menor número de funcionalidad posible, delegando a un pequeño conjunto de procesos tareas específicas.



Sistemas monolíticos

- No cuentan con una estructura definida.
- El SO se describe como un conjunto de procedimientos, en donde un procedimiento puede invocar a los demás procedimientos si así lo requiere.
- Todos los procedimientos tienen una interfáz de comunicación bien definida.
- En un sistema monolítico:
 - 1 El programa principal llama a un procedimiento de servicio
 - 2 Un conjunto de procedimientos de servicios que lleva a cabo las llamadas al sistema.
 - 3 Un conjunto de procedimientos de utilerijía que ayudan al procedimiento de servicio.



Sistemas monolíticos

La secuencia de ejecución de una llamada al sistema en un sistema operativo monolítico es:

- 1 El programa de usuario es atraído hacia el núcleo
- 2 El SO determina el número de servicio solicitado
- 3 El SO localiza y llama al procedimiento correspondiente al servicio.
- 4 El control regresa al programa de usuario.



Sistemas monolíticos

La secuencia de ejecución de una llamada al sistema en un sistema operativo monolítico es:

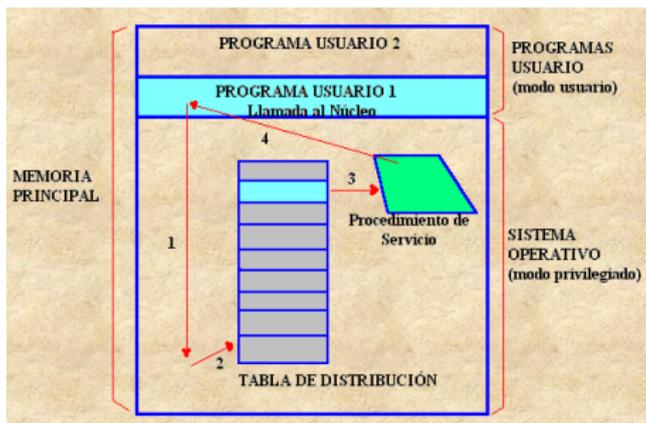


Figura: Ejecución de una llamada al sistema.



Sistemas microkernel

- 1 Se delegan al espacio de usuario todos los procesos que sea posible
- 2 Código resultante más limpio
- 3 Son más lentos que los monolíticos
- 4 Su implementación es más difícil de realizar correctamente



Índice

- 1 Introducción
- 2 Organización de una computadora
- 3 Llamadas al sistema
- 4 Estructura del sistema operativo
- 5 Evolución de los sistemas operativos**



Historia de los Sistemas Operativos

Los sistemas operativos y su evolución han estado muy ligados con la arquitectura de las computadoras en las que se ejecutan.

- Máquina analítica: La primera computadora digital fue construida por el matemático inglés *Charles Babbage*. Esta máquina no contaba con un Sistema Operativo.

Anécdota

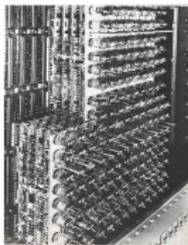
... *Babbage* se dio cuenta de que su máquina analítica necesitaría software, así que contrató a una joven mujer llamada *Ada Lovelace*, como la primera programadora de la historia. El lenguaje de programación *Ada* recibió su nombre en honor a ella.



Primera Generación (1944 -1955)

Tubos de vacío

- Al principio, toda la programación se realizaba en lenguaje máquina, generalmente alambrando tableros de conmutación, no existían los lenguajes de programación (ni siquiera los de ensamblador).
- Nadie había oído hablar de los Sistemas Operativos.
- En 1950's, ya fue posible escribir programas en tarjetas perforadas e introducirlas para ser leídas.





Primera Generación (1944 -1955)

Tubos de vacío

- Al principio, toda la programación se realizaba en lenguaje máquina, generalmente alambrando tableros de conmutación, no existían los lenguajes de programación (ni siquiera los de ensamblador).
- Nadie había oído hablar de los Sistemas Operativos.
- En 1950's, ya fue posible escribir programas en tarjetas perforadas e introducirlas para ser leídas.

Construcción de calculadoras

En los 40's Howard Aiken, John von Neumann, J. Presper Eckert y William Mauchley, entre otros, lograron construir máquinas calculadoras usando tubos de vacío.



Segunda Generación (1955 - 1965)

Transistores y sistemas por lote

- Introducción de los transistores revolucionó la computación. Ello permitió que las computadoras comenzaran a fabricarse y vender a clientes comerciales.
- Los Sistemas Operativos típicos eran FMS (Fortran Monitor System) e IBSYS (sistema operativo de IBM para la 7004).
- Surgen los primeros sistemas de procesamiento por lotes. Computadoras de recursos y capacidad limitada se utilizaban para realizar tareas que no demandaban gran poder de cómputo (leer y grabar cintas magnéticas), mientras que el procesamiento se utilizaba para cálculos matemáticos.



Tercera Generación (1965-1980)

Circuitos integrados y multiprogramación

- Se fabrican las primeras familias de computadoras, mismas que tienen compatibilidad de software.
- Surge el SO/360, escrito en lenguaje ensamblador, de gran tamaño y muy complejo.
- Una característica de ésta generación de SO es la **multiprogramación**



Tercera Generación (1965-1980)

Circuitos integrados y multiprogramación

- Los SO implementan también una variante de la multiprogramación, *el tiempo compartido*, con la finalidad de utilizar en mayor medida el tiempo de procesador, permitiendo que más usuarios realizaran sus tareas.
- Surgen las minicomputadoras (PDP-x, todas incompatibles entre sí excepto la familia IBM).

Orígenes de UNIX

Ken Thompson, computólogo de los laboratorios Bell escribió una versión del SO MULTICS reducida al mínimo, mismo que evolucionó para convertirse en UNIX.



Cuarta Generación (1980 - Actual)

Computadoras personales

Computadoras personales

Con la invención de los circuitos integrados a gran escala (LSI), chips que contienen miles de transistores en un cm^2 de silicio, nació la era de la computadora personal.

- Comienza el desarrollo de software con fines comerciales.
- Dos sistemas operativos dominaron: MS-DOS de Microsoft (arquitecturas 8088 y posteriormente Pentium) y UNIX.
- A mediados de los 80's comenzaron a popularizarse las redes de computadoras personales y con ello los *Sistemas Distribuidos*.



Cuarta Generación (1980 - Actual)

Computadoras personales

- En un sistema operativo en red los usuarios están conscientes de la existencia de múltiples computadoras y pueden ingresar en máquinas remotas y copiar archivos de una máquina a otra.
- Los sistemas operativos distribuidos necesitan un controlador de la interfaz de red y software de bajo nivel para operarlo, así como aplicaciones para iniciar sesiones remotas.
- Un sistema distribuido, en cambio, ante el usuario aparece como un sistema tradicional de un solo procesador (transparencia).



Cuarta Generación (1980 - Actual)

Computadoras personales

- Los sistemas distribuidos permiten a las aplicaciones ejecutarse en varios procesadores al mismo tiempo, por lo que requieren algoritmos de planificación de procesador más complejos a fin de optimizar el paralelismo.



Historia de MINIX

- MINIX inicialmente se basó en la versión 7 de UNIX, misma que fue liberada con una licencia que prohibía su estudio con fines académicos.
- Fue desarrollado por *Andrew S. Tanenbaum* en lenguaje C.

Surgimiento de Linux

Debido a la negativa de *Tanenbaum* de modificar MINIX, un estudiante finlandés (*Linux Torvalds*) decidió escribir un clon de MINIX con la finalidad de ser un sistema de producción con abundantes capacidades, más que una herramienta educativa.