

INVESTIGACIÓN

Lenguaje ensamblador

El lenguaje ensamblador surge con la idea de evitar las dificultades que presenta el trabajar en lenguaje máquina, siendo la misión de este la de simplificar la programación en un computador y teniendo a la vez un control directo sobre el hardware del mismo.

El lenguaje ensamblador es un lenguaje cuyas estructuras de datos se corresponden con las estructuras físicas de los registros de la memoria principal del computador para el que está pensado, y cuyas instrucciones tienen una relación directa y biunívoca con las instrucciones del lenguaje máquina.

Estructura

En el lenguaje ensamblador las líneas de código constan de dos partes, la primera es el nombre de la instrucción que se va a ejecutar y la segunda son los parámetros del comando u operandos. Por ejemplo:

add ah bh

Aquí "add" es el comando a ejecutar (en éste caso una adición) y tanto "ah" como "bh" son los parámetros.

El nombre de las instrucciones en éste lenguaje esta formado por dos, tres o cuatro letras. A estas instrucciones también se les llama nombres mnemónicos o códigos de operación, ya que representan alguna función que habrá de realizar el procesador.

Existen algunos comandos que no requieren parámetros para su operación, así como otros que requieren solo un par metro.

Algunas veces se utilizaran las instrucciones como sigue:

add al,[170]

Los corchetes en el segundo parámetro nos indican que vamos a trabajar con el contenido de la casilla de memoria número 170 y no con el valor 170, a esto se le conoce como direccionamiento directo.

Sintaxis

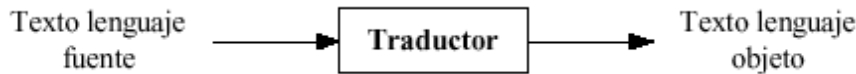
La Sintaxis de un lenguaje ensamblador es el conjunto de reglas que debe guardar el programa fuente y que estará compuesto por una serie de instrucciones, distinguiéndose cuatro campos: etiqueta, mnemotécnico, operando y comentario. El programa fuente deberá estar en un fichero ASCII que se genera con la ayuda de un programa editor, y no un procesador de textos, ya que estos generan códigos de control que los ensambladores son incapaces de interpretar. Un sencillo editor es el de MS-DOS. Las instrucciones en código máquina se codifican por campos, por lo que las instrucciones escritas en ensamblador se codificaran también por campos.

Línea de instrucción del ensamblador

ETIQUETA	NEMOTECNICO	OPERANDO	COMENTARIO
SUBR	MOVE.L	D0, - (SP) ;	Guarda D0

Compilador

Es un traductor que puede ser cualquier programa que toma como entrada un texto escrito en un lenguaje, llamado fuente y da como salida otro texto en un lenguaje, denominado objeto.



Un compilador no es un programa que funciona de manera aislada, sino que necesita de otros programas para conseguir su objetivo: obtener un programa ejecutable a partir de un programa fuente en un lenguaje de alto nivel. Algunos de esos programas son el preprocesador, el linker, el depurador y el ensamblador. El preprocesador se ocupa (dependiendo del lenguaje) de incluir ficheros, expandir macros, eliminar comentarios, y otras tareas similares. El linker se encarga de construir el fichero ejecutable añadiendo al fichero objeto generado por el compilador las cabeceras necesarias y las funciones de librería utilizadas por el programa fuente. El depurador permite, si el compilador ha generado adecuadamente el programa objeto, seguir paso a paso la ejecución de un programa. Finalmente, muchos compiladores, en vez de generar código objeto, generan un programa en lenguaje ensamblador que debe después convertirse en un ejecutable mediante un programa ensamblador.

Clasificación

El programa compilador traduce las instrucciones en un lenguaje de alto nivel a instrucciones que la computadora puede interpretar y ejecutar. Para cada lenguaje de programación se requiere un compilador separado. El compilador traduce todo el programa antes de ejecutarlo. Los compiladores son, pues, programas de traducción insertados en la memoria por el sistema operativo para convertir programas de cómputo en pulsaciones electrónicas ejecutables (lenguaje de máquina).

Los compiladores pueden ser:

- De una sola pasada: examina el código fuente una vez, generando el código o programa objeto.
- De pasadas múltiples: requieren pasos intermedios para producir un código en otro lenguaje, y una pasada final para producir y optimizar el código producido durante los pasos anteriores.
- Optimización: lee un código fuente, lo analiza y descubre errores potenciales sin ejecutar el programa.
- Compiladores incrementales: generan un código objeto instrucción por instrucción (en vez de hacerlo para todo el programa) cuando el usuario teclea cada orden individual. El otro tipo de compiladores requiere que todos los enunciados o instrucciones se compilen conjuntamente.
- Ensamblador: el lenguaje fuente es lenguaje ensamblador y posee una estructura sencilla.
- Compilador cruzado: se genera código en lenguaje objeto para una máquina diferente de la que se está utilizando para compilar. Es perfectamente normal construir un compilador de Pascal que genere código para MS-DOS y que el compilador funcione en Linux y se haya escrito en C++.
- Compilador con montador: compilador que compila distintos módulos de forma independiente y después es capaz de enlazarlos.
- Autocompilador: compilador que está escrito en el mismo lenguaje que va a compilar. Evidentemente, no se puede ejecutar la primera vez. Sirve para hacer ampliaciones al lenguaje, mejorar el código generado, etc.

- **Metacompilador:** es sinónimo de compilador de compiladores y se refiere a un programa que recibe como entrada las especificaciones del lenguaje para el que se desea obtener un compilador y genera como salida el compilador para ese lenguaje.
- **Descompilador:** es un programa que acepta como entrada código máquina y lo traduce a un lenguaje de alto nivel, realizando el proceso inverso a la compilación.

Procesamiento por lotes

Es un modo de procesamiento que funciona en el sistema operativo de una computadora, donde los trabajos son procesados de manera secuencial o por lotes; es decir, el primer programa que entra es el primero que se ejecuta, luego el segundo y así sucesivamente hasta que se procesan todos los programas.

El inconveniente de este modo de procesamiento es que si se desea ejecutar un programa con alta prioridad, tendrá que esperar a que se procesen todos los programas que se encuentren antes que éste.

La tabla a seguir muestra, de forma resumida, el sistema de filas y comandos para manipulación de tareas en lote.

Sistema de Filas	NQS
Sumisión	QSUB
Remoción	QDEL
Monitoración	QSTAT
	JSTAT

El procesamiento por lotes (batch processing) se inicio alrededor de los años 60 con las primeras y enormes computadoras. Su manejo era tan complicado que solo expertos programadores tenían la capacidad de operarlas y dado su elevado costo, eran solo posible de ser adquiridas por bancos y entidades de gobierno.

Dentro de este tipo de procesamiento, la forma de comunicarse con las computadoras eran las tarjetas perforadas. Aquellas tarjetas de esquina recortada y agujeros rectangulares eran la programación, es decir las instrucciones que le indicaban, una vez dentro, lo que la máquina debía hacer.

Todos los trabajos asignados a la computadora no podían ser efectuados al mismo tiempo, por lo que era necesario determinar horarios y cantidades de tarjetas para ser procesadas secuencialmente y por lotes; es así como nace el nombre "procesamiento por lotes".

Su característica principal es : La no existencia de interacción entre el usuario y la maquina y el agrupamiento de equipos de computo.

Sistema operativo

Programa que actúa como intermediario entre el usuario y el hardware de un computador y su propósito es proporcionar un entorno en el cual el usuario pueda ejecutar programas.

El objetivo principal de un sistema operativo es lograr que el sistema de computación se use de manera cómoda y el objetivo secundario es que el hardware del computador se emplee de manera eficiente. Una definición más común es que el sistema operativo es el programa que se ejecuta todo el tiempo en el computador, siendo programas de aplicación todos los demás.

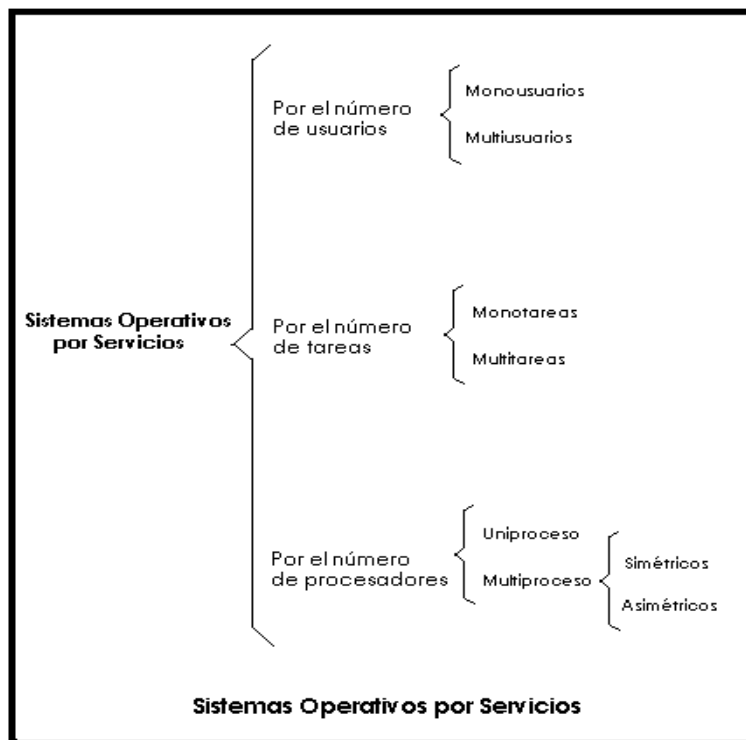
- **Objetivo principal:** los sistemas operativos existen porque se supone que es más fácil trabajar con uno de ellos que sin él. Esta situación es particularmente clara cuando se observan los sistemas operativos para los pequeños computadores personales.
- **Objetivo secundario:** es la utilización eficiente del sistema de computación. Este propósito tiene una importancia especial en los grandes sistemas multiusuario compartidos. En el pasado, las consideraciones de eficiencia a menudo eran más importantes que la comodidad, por lo que gran parte de la teoría de sistemas operativos se concentra en el uso óptimo de los recursos de computación.

Clasificación de los sistemas operativos

Actualmente los sistemas operativos se clasifican en tres clasificaciones: sistemas operativos por su estructura (visión interna), sistemas operativos por los servicios que ofrecen y sistemas operativos por la forma en que ofrecen sus servicios (visión externa).

Sistemas Operativos por Servicios(Visión Externa).

Se clasifican de la siguiente manera:



Por Número de Usuarios:

Sistema Operativo Monousuario.

Los sistemas operativos monousuarios son aquellos que soportan a un usuario a la vez, sin importar el número de procesadores que tenga la computadora o el número de procesos o tareas que el usuario pueda ejecutar en un mismo instante de tiempo. Las computadoras personales típicamente se han clasificado en este renglón.

En otras palabras los sistemas monousuarios son aquellos que nada más puede atender a un solo usuario, gracias a las limitaciones creadas por el hardware, los programas o el tipo de aplicación que se este ejecutando.

Sistema Operativo Multiusuario.

Los sistemas operativos multiusuarios son capaces de dar servicio a más de un usuario a la vez, ya sea por medio de varias terminales conectadas a la computadora o por medio de sesiones remotas en una red de comunicaciones. No importa el número de procesadores en la máquina ni el número de procesos que cada usuario puede ejecutar simultáneamente.

En esta categoría se encuentran todos los sistemas que cumplen simultáneamente las necesidades de dos o más usuarios, que comparten mismos recursos.

Este tipo de sistemas se emplean especialmente en redes. En otras palabras consiste en el fraccionamiento del tiempo (timesharing).

Por el Número de Tareas:

Sistema Operativo Monotarea.

Los sistemas monotarea son aquellos que sólo permiten una tarea a la vez por usuario. Puede darse el caso de un sistema multiusuario y monotarea, en el cual se admiten varios usuarios al mismo tiempo pero cada uno de ellos puede estar haciendo solo una tarea a la vez.

Los sistemas operativos monotareas son más primitivos y, solo pueden manejar un proceso en cada momento o que solo puede ejecutar las tareas de una en una.

Sistema Operativo Multitarea.

Un sistema operativo multitarea es aquél que le permite al usuario estar realizando varias labores al mismo tiempo.

Es el modo de funcionamiento disponible en algunos sistemas operativos, mediante el cual una computadora procesa varias tareas al mismo tiempo. Existen varios tipos de multitareas. La conmutación de contextos (context Switching) es un tipo muy simple de multitarea en el que dos o más aplicaciones se cargan al mismo tiempo, pero en el que solo se esta procesando la aplicación que se encuentra en primer plano la que ve el usuario. En la multitarea cooperativa, la que se utiliza en el sistema operativo Macintosh, las tareas en segundo plano reciben tiempo de procesamiento durante los tiempos muertos de la tarea que se encuentra en primer plano (por ejemplo, cuando esta aplicación esta esperando información del usuario), y siempre que esta aplicación lo permita. En los sistemas multitarea de tiempo compartido, como OS/2, cada tarea recibe la atención del microprocesador durante una fracción de segundo.

Un sistema operativo multitarea puede estar editando el código fuente de un programa durante su depuración mientras compila otro programa, a la vez que está recibiendo correo electrónico en un proceso en background. Es común encontrar en ellos interfaces gráficas orientadas al uso de menús y el ratón, lo cual permite un rápido intercambio entre las tareas para el usuario, mejorando su productividad.

Un sistema operativo multitarea se distingue por su capacidad para soportar la ejecución concurrente de dos o más procesos activos. La multitarea se implementa generalmente manteniendo el código y los datos de varios procesos simultáneamente en memoria y multiplexando el procesador y los dispositivos de E/S entre ellos.

La multitarea suele asociarse con soporte hardware y software para protección de memoria con el fin de evitar que procesos corrompan el espacio de direcciones y el comportamiento de otros procesos residentes.

Por el Número de Procesadores:

Sistema Operativo de Uniproceto.

Un sistema operativo uniproceto es aquél que es capaz de manejar solamente un procesador de la computadora, de manera que si la computadora tuviese más de uno le sería inútil. El ejemplo más típico de este tipo de sistemas es el DOS y MacOS.

Sistema Operativo de Multiproceto.

Un sistema operativo multiproceto se refiere al número de procesadores del sistema, que es más de uno y éste es capaz de usarlos todos para distribuir su carga de trabajo. Generalmente estos sistemas trabajan de dos formas: simétrica o asimétricamente.

- Asimétrica: Cuando se trabaja de manera asimétrica, el sistema operativo selecciona a uno de los procesadores el cual jugará el papel de procesador maestro y servirá como pivote para distribuir la carga a los demás procesadores, que reciben el nombre de esclavos.
- Simétrica: Cuando se trabaja de manera simétrica, los procesos o partes de ellos (threads) son enviados indistintamente a cual quiera de los procesadores disponibles, teniendo, teóricamente, una mejor distribución y equilibrio en la carga de trabajo bajo este esquema.

Se dice que un thread es la parte activa en memoria y corriendo de un proceso, lo cual puede consistir de un área de memoria, un conjunto de registros con valores específicos, la pila y otros valores de contexto.

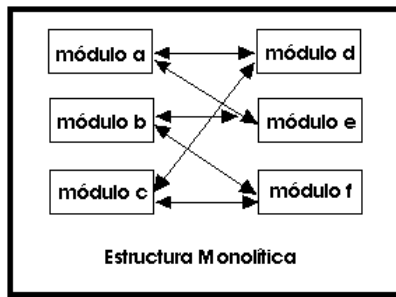
Un aspecto importante a considerar en estos sistemas es la forma de crear aplicaciones para aprovechar los varios procesadores. Existen aplicaciones que fueron hechas para correr en sistemas monoproceso que no toman ninguna ventaja a menos que el sistema operativo o el compilador detecte secciones de código paralelizable, los cuales son ejecutados al mismo tiempo en procesadores diferentes. Por otro lado, el programador puede modificar sus algoritmos y aprovechar por sí mismo esta facilidad, pero esta última opción las más de las veces es costosa en horas hombre y muy tediosa, obligando al programador a ocupar tanto o más tiempo a la paralelización que a elaborar el algoritmo inicial.

Sistemas Operativos por su Estructura (Visión Interna).

Estructura Monolítica.

Es la estructura de los primeros sistemas operativos constituidos fundamentalmente por un solo programa compuesto de un conjunto de rutinas entrelazadas de tal forma que cada una puede llamar a cualquier otra. Las características fundamentales de este tipo de estructura son:

- Construcción del programa final a base de módulos compilados separadamente que se unen a través del ligador.
- Buena definición de parámetros de enlace entre las distintas rutinas existentes, que puede provocar mucho acoplamiento.
- Carecen de protecciones y privilegios al entrar a rutinas que manejan diferentes aspectos de los recursos de la computadora, como memoria, disco, etc.

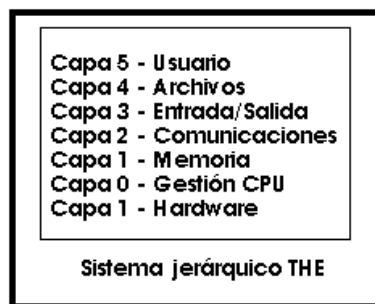


Generalmente están hechos a medida, por lo que son eficientes y rápidos en su ejecución y gestión, pero por lo mismo carecen de flexibilidad para soportar diferentes ambientes de trabajo o tipos de aplicaciones.

Estructura Jerárquica.

A medida que fueron creciendo las necesidades de los usuarios y se perfeccionaron los sistemas, se hizo necesaria una mayor organización del software, del sistema operativo, donde una parte del sistema contenía subpartes y esto organizado en forma de niveles.

Se dividió el sistema operativo en pequeñas partes, de tal forma que cada una de ellas estuviera perfectamente definida y con una clara interface con el resto de elementos.



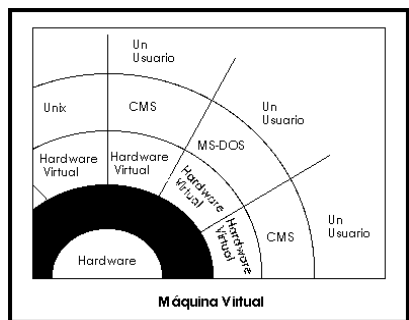
Se constituyó una estructura jerárquica o de niveles en los sistemas operativos, el primero de los cuales fue denominado THE (Technische Hogeschool, Eindhoven), de Dijkstra, que se utilizó con fines didácticos. Se puede pensar también en estos sistemas como si fueran 'multicapa'. Multics y Unix caen en esa categoría.

Máquina Virtual.

Se trata de un tipo de sistemas operativos que presentan una interface a cada proceso, mostrando una máquina que parece idéntica a la máquina real subyacente. Estos sistemas operativos separan dos conceptos que suelen estar unidos en el resto de sistemas: la multiprogramación y la máquina extendida.

El objetivo de los sistemas operativos de máquina virtual es el de integrar distintos sistemas operativos dando la sensación de ser varias máquinas diferentes.

El núcleo de estos sistemas operativos se denomina monitor virtual y tiene como misión llevar a cabo la multiprogramación, presentando a los niveles superiores tantas máquinas virtuales como se soliciten. Estas máquinas virtuales no son máquinas extendidas, sino una réplica de la máquina real, de manera que en cada una de ellas se pueda ejecutar un sistema operativo diferente, que será el que ofrezca la máquina extendida al usuario.



Cliente-Servidor(Microkernel).

El tipo más reciente de sistemas operativos es el denominado Cliente-servidor, que puede ser ejecutado en la mayoría de las computadoras, ya sean grandes o pequeñas.

Este sistema sirve para toda clase de aplicaciones por tanto, es de propósito general y cumple con las mismas actividades que los sistemas operativos convencionales.

El núcleo tiene como misión establecer la comunicación entre los clientes y los servidores. Los procesos pueden ser tanto servidores como clientes. Por ejemplo, un programa de aplicación normal es un cliente que llama al servidor correspondiente para acceder a un archivo o realizar una operación de entrada/salida sobre un dispositivo concreto. A su vez, un proceso cliente puede actuar como servidor para otro.

Este paradigma ofrece gran flexibilidad en cuanto a los servicios posibles en el sistema final, ya que el núcleo provee solamente funciones muy básicas de memoria, entrada/salida, archivos y procesos, dejando a los servidores proveer la mayoría que el usuario final o programador puede usar. Estos servidores deben tener mecanismos de seguridad y protección que, a su vez, serán filtrados por el núcleo que controla el hardware.

Sistemas Operativos por la Forma de Ofrecer sus Servicios

Esta clasificación también se refiere a una visión externa, que en este caso se refiere a la del usuario, el cómo accesa a los servicios. Bajo esta clasificación se pueden detectar dos tipos principales: sistemas operativos de red y sistemas operativos distribuidos.

Sistema Operativo de Red.

Los sistemas operativos de red se definen como aquellos que tienen la capacidad de interactuar con sistemas operativos en otras computadoras por medio de un medio de transmisión con el objeto de intercambiar información, transferir archivos, ejecutar comandos remotos y un sin fin de otras actividades. El punto crucial de estos sistemas es que el usuario debe saber la sintaxis de un conjunto de comandos o llamadas al sistema para ejecutar estas operaciones, además de la ubicación de los recursos que desee acceder. Por ejemplo, si un usuario en la computadora hidalgo necesita el archivo matriz.pas que se localiza en el directorio /software/codigo en la computadora morelos bajo el sistema operativo UNIX, dicho usuario podría copiarlo a través de la red con los comandos siguientes: hidalgo% hidalgo% rcp morelos:/software/codigo/matriz.pas . hidalgo%. En este caso, el comando rcp que significa "remote copy" trae el archivo indicado de la computadora morelos y lo coloca en el directorio donde se ejecutó el mencionado comando. Lo importante es hacer ver que el usuario puede acceder y compartir muchos recursos.

El primer Sistema Operativo de red estaba enfocado a equipos con un procesador Motorola 68000, pasando posteriormente a procesadores Intel como Novell Netware.

Los Sistemas Operativos de red mas ampliamente usados son: Novell Netware, Personal Netware, LAN

Manager, Windows NT Server, UNIX, LANtastic.

Sistemas Operativos Distribuidos.

Los sistemas operativos distribuidos abarcan los servicios de los de red, logrando integrar recursos (impresoras, unidades de respaldo, memoria, procesos, unidades centrales de proceso), en una sola máquina virtual que el usuario acceda en forma transparente. Es decir, ahora el usuario ya no necesita saber la ubicación de los recursos, sino que los conoce por nombre y simplemente los usa como si todos ellos fuesen locales a su lugar de trabajo habitual.

Todo lo anterior es el marco teórico de lo que se desearía tener como sistema operativo distribuido, pero en la realidad no se ha conseguido crear uno del todo, por la complejidad que suponen: distribuir los procesos en las varias unidades de procesamiento, reintegrar sub-resultados, resolver problemas de concurrencia y paralelismo, recuperarse de fallas de algunos recursos distribuidos y consolidar la protección y seguridad entre los diferentes componentes del sistema y los usuarios. Los avances tecnológicos en las redes de área local y la creación de microprocesadores de 32 y 64 bits lograron que computadoras mas o menos baratas tuvieran el suficiente poder en forma autónoma para desafiar en cierto grado a los mainframes, y a la vez se dio la posibilidad de intercomunicarlas, sugiriendo la oportunidad de partir procesos muy pesados en cálculo en unidades más pequeñas y distribuirlos en los varios microprocesadores para luego reunir los sub-resultados, creando así una máquina virtual en la red que exceda en poder a un mainframe.

El sistema integrador de los microprocesadores que hace ver a las varias memorias, procesadores, y todos los demás recursos como una sola entidad en forma transparente se le llama sistema operativo distribuido.

Las razones para crear o adoptar sistemas distribuidos se dan por dos razones principales: por necesidad (debido a que los problemas a resolver son inherentemente distribuidos) o porque se desea tener más confiabilidad y disponibilidad de recursos. En el primer caso tenemos, por ejemplo, el control de los cajeros automáticos en diferentes estados de la república. Ahí no es posible ni eficiente mantener un control centralizado, es más, no existe capacidad de cómputo y de entrada/salida para dar servicio a los millones de operaciones por minuto. En el segundo caso, supóngase que se tienen en una gran empresa varios grupos de trabajo, cada uno necesita almacenar grandes cantidades de información en disco duro con una alta confiabilidad y disponibilidad. La solución puede ser que para cada grupo de trabajo se asigne una partición de disco duro en servidores diferentes, de manera que si uno de los servidores falla, no se deje dar el servicio a todos, sino sólo a unos cuantos y, más aún, se podría tener un sistema con discos en espejo (mirror) a través de la red, de manera que si un servidor se cae, el servidor en espejo continúa trabajando y el usuario ni cuenta se da de estas fallas, es decir, obtiene acceso a recursos en forma transparente.

Los sistemas distribuidos deben de ser muy confiables, ya que si un componente del sistema se compone otro componente debe de ser capaz de reemplazarlo.

Entre los diferentes Sistemas Operativos distribuidos que existen tenemos los siguientes: Sprite, Solaris-MC, Mach, Chorus, Spring, Amoeba, Taos, etc.

Generaciones de los sistemas operativos

Los Sistemas Operativos, al igual que el Hardware de los computadores, han sufrido una serie de cambios revolucionarios llamados generaciones. En el caso del Hardware, las generaciones han sido marcadas por grandes avances en los componentes utilizados, pasando de válvulas (primera generación) a transistores (segunda generación), a circuitos integrados (tercera generación), a circuitos integrados de gran y muy gran escala (cuarta generación). Cada generación Sucesiva de hardware ha ido acompañada de reducciones substanciales en los costos, tamaño, emisión de calor y consumo de energía, y por incrementos notables en velocidad y capacidad.

Generación Cero (década de 1940)

Los primeros sistemas computacionales no poseían sistemas operativos. Los usuarios tenían completo acceso al lenguaje de la maquina. Todas las instrucciones eran codificadas a mano.

Primera Generación (década de 1950)

Los sistemas operativos de los años cincuenta fueron diseñados para hacer mas fluida la transición entre trabajos. Antes de que los sistemas fueran diseñados, se perdía un tiempo considerable entre la terminación de un trabajo y el inicio del siguiente. Este fue el comienzo de los sistemas de procesamiento por lotes, donde los trabajos se reunían por grupos o lotes. Cuando el trabajo estaba en ejecución, este tenia control total de la maquina. Al terminar cada trabajo, el control era devuelto al sistema operativo, el cual limpiaba y leía e iniciaba el trabajo siguiente.

Al inicio de los 50's esto había mejorado un poco con la introducción de tarjetas perforadas (las cuales servían para introducir los programas de lenguajes de máquina), puesto que ya no había necesidad de utilizar los tableros enchufables.

Además el laboratorio de investigación General Motors implementó el primer sistema operativo para la IBM 701. Los sistemas de los 50's generalmente ejecutaban una sola tarea, y la transición entre tareas se suavizaba para lograr la máxima utilización del sistema. Esto se conoce como sistemas de procesamiento por lotes de un sólo flujo, ya que los programas y los datos eran sometidos en grupos o lotes.

La introducción del transistor a mediados de los 50's cambió la imagen radicalmente.

Se crearon máquinas suficientemente confiables las cuales se instalaban en lugares especialmente

acondicionados, aunque sólo las grandes universidades y las grandes corporaciones o bien las oficinas del gobierno se podían dar el lujo de tenerlas.

Para poder correr un trabajo (programa), tenían que escribirlo en papel (en Fortran o en lenguaje ensamblador) y después se perforaría en tarjetas. Enseguida se llevaría la pila de tarjetas al cuarto de introducción al sistema y la entregaría a uno de los operadores. Cuando la computadora terminara el trabajo, un operador se dirigiría a la impresora y desprendería la salida y la llevaría al cuarto de salida, para que la recogiera el programador.

Segunda Generación (a mitad de la década de 1960)

La característica de los sistemas operativos fue el desarrollo de los sistemas compartidos con

multiprogramación, y los principios del multiprocesamiento. En los sistemas de multiprogramación, varios programas de usuario se encuentran al mismo tiempo en el almacenamiento principal, y el procesador se cambia rápidamente de un trabajo a otro. En los sistemas de multiprocesamiento se utilizan varios procesadores en un solo sistema computacional, con la finalidad de incrementar el poder de procesamiento de la maquina.

La independencia de dispositivos aparece después. Un usuario que desea escribir datos en una cinta en sistemas de la primera generación tenia que hacer referencia especifica a una unidad de cinta particular. En la segunda generación, el programa del usuario especificaba tan solo que un archivo iba a ser escrito en una unidad de cinta con cierto numero de pistas y cierta densidad.

Se desarrollo sistemas compartidos, en la que los usuarios podían acoplarse directamente con el computador a través de terminales. Surgieron sistemas de tiempo real, en que los computadores fueron utilizados en el

control de procesos industriales. Los sistemas de tiempo real se caracterizan por proveer una respuesta inmediata.

Tercera Generación (mitad de década 1960 a mitad década de 1970)

Se inicia en 1964, con la introducción de la familia de computadores Sistema/360 de IBM. Los computadores de esta generación fueron diseñados como sistemas para usos generales . Casi siempre eran sistemas grandes, voluminosos, con el propósito de serlo todo para toda la gente. Eran sistemas de modos múltiples, algunos de ellos soportaban simultáneamente procesos por lotes, tiempo compartido, procesamiento de tiempo real y multiprocesamiento. Eran grandes y costosos, nunca antes se había construido algo similar, y muchos de los esfuerzos de desarrollo terminaron muy por arriba del presupuesto y mucho después de lo que el planificador marcaba como fecha de terminación.

Estos sistemas introdujeron mayor complejidad a los ambientes computacionales; una complejidad a la cual, en un principio, no estaban acostumbrados los usuarios.

Cuarta Generación (mitad de década de 1970 en adelante)

Los sistemas de la cuarta generación constituyen el estado actual de la tecnología. Muchos diseñadores y usuarios se sienten aun incómodos, después de sus experiencias con los sistemas operativos de la tercera generación.

Con la ampliación del uso de redes de computadores y del procesamiento en línea los usuarios obtienen acceso a computadores alejados geográficamente a través de varios tipos de terminales.

Los sistemas de seguridad se ha incrementado mucho ahora que la información pasa a través de varios tipos vulnerables de líneas de comunicación. La clave de cifrado esta recibiendo mucha atención; han sido necesario codificar los datos personales o de gran intimidad para que; aun si los datos son expuestos, no sean de utilidad a nadie mas que a los receptores adecuados.

El porcentaje de la población que tiene acceso a un computador en la década de los ochenta es mucho mayor que nunca y aumenta rápidamente.

El concepto de maquinas virtuales es utilizado. El usuario ya no se encuentra interesado en los detalles físicos de; sistema de computación que esta siendo accedida. En su lugar, el usuario ve un panorama llamado maquina virtual creado por el sistema operativo.

Los sistemas de bases de datos han adquirido gran importancia. Nuestro mundo es una sociedad orientada hacia la información, y el trabajo de las bases de datos es hacer que esta información sea conveniente accesible de una manera controlada para aquellos que tienen derechos de acceso.

BIBLIOGRAFÍA

Biblioteca de Consulta Microsoft Encarta 2002

www.usuarios.lycos.es

www.geocities.com

www.rincondelvago.com

www.spisa.act.uji.es