UAEM | Universidad Autónoma del Estado de México

CENTRO UNIVERSITARIO UAEM
ZUMPANGO

Ingeniería en computación

Tipos de sistemas operativos

Unidad de competencia I: Introducción a los Sistemas Operativos Distribuidos

Ing. Diego Armando Ramírez Avelino

Unidad de competencia I

- Objetivo

  Conocer los diferentes tipos de sistemas operativos, presentando los conceptos básicos y estructura de los mismos.

Contenido

1.1. Tipos de sistemas operativos

Tipos de sistemas operativos

Para diferenciar a los sistemas operativos debemos conocer las características que clasifican a los sistemas operativos, básicamente se cubrirán tres clasificaciones:

- Sistemas operativos por su estructura (visión interna),
- Sistemas operativos por los servicios que ofrecen y finalmente,
- Sistemas operativos por la forma en que ofrecen sus servicios (visión externa).

1.2. Estructura de los sistemas operativos

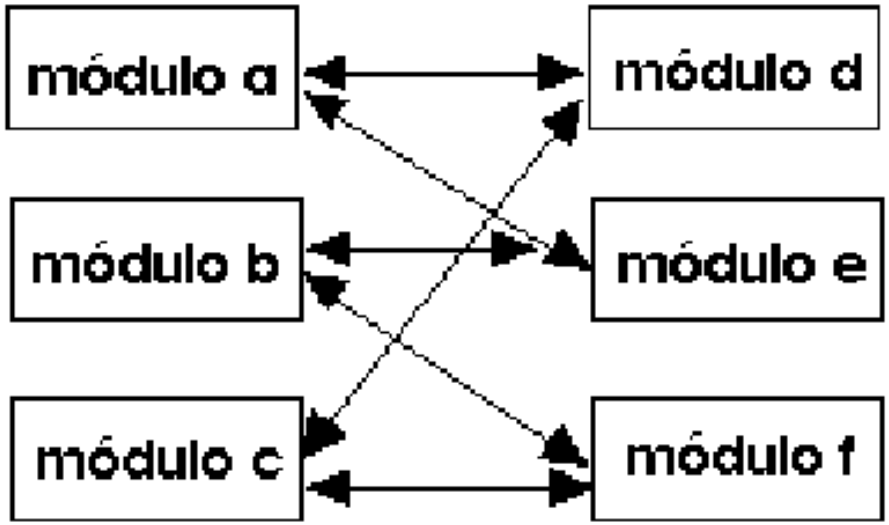
Sistemas operativos por su estructura

Se deben observar dos tipos de requisitos cuando se construye un sistema operativo, los cuales son:

- Requisitos por el usuario: Sistema fácil de usar y de aprender, seguro, rápido y adecuado al uso al que se le requiere destinar.
- Requisitos de software: Donde se engloban aspectos como el mantenimiento, forma de operación, restricciones de uso, eficiencia, tolerancia frente a los errores y flexibilidad.

Estructura monolítica

Es la estructura de los primeros sistemas operativos construidos fundamentalmente por un solo compuesto de un conjunto de rutinas entrelazadas de tal forma que cada una puede llamar a cualquier otra.



Estructura monolítica

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 (enlazador)
- 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 bien hechos a medida, por lo que son eficientes y rápidos en su ejecución y gestion, pero por lo mismo carecen de flexibilidad para soportar diferentes ambientes de trabajo o 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 sub partes 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 interfaz con con el resto de los elementos.

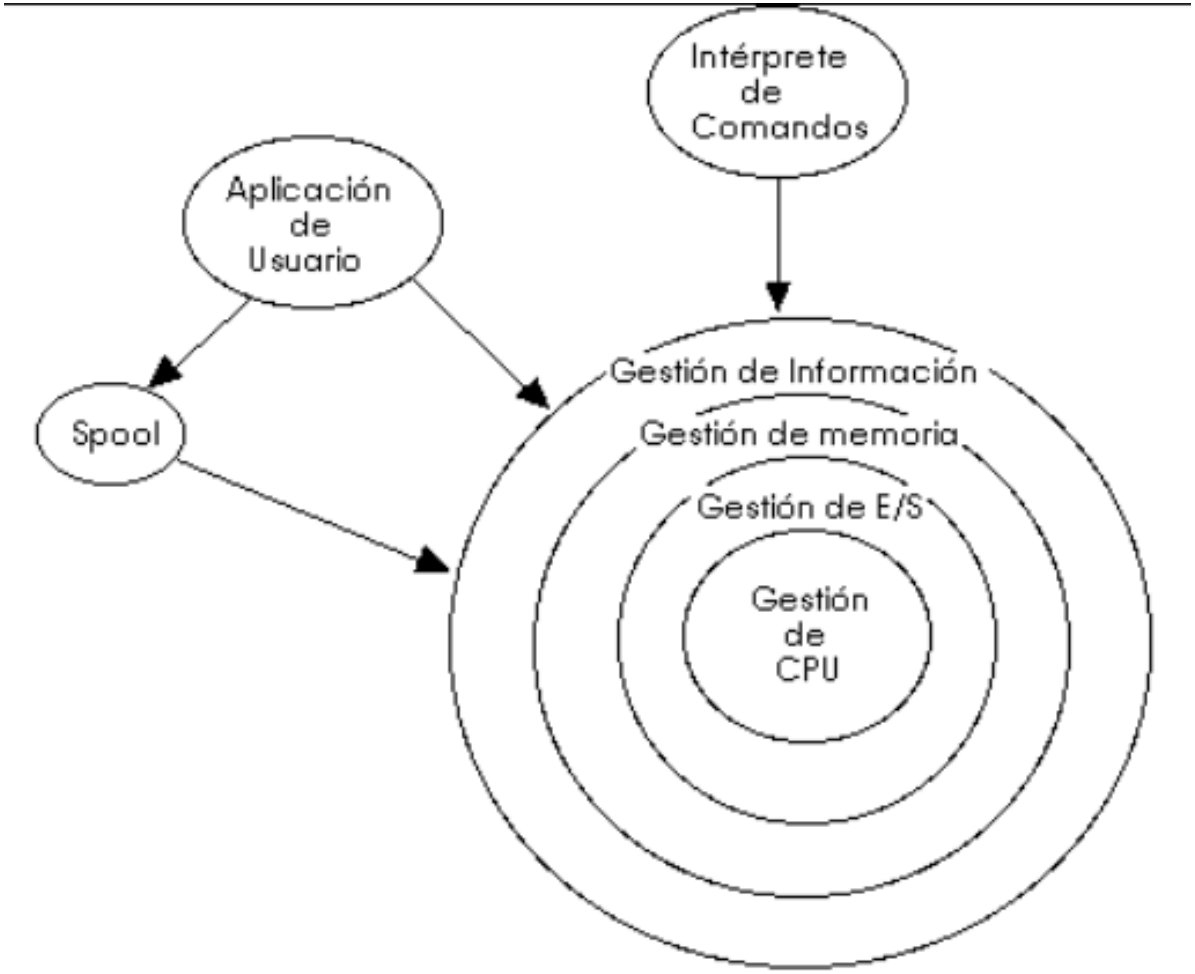
Se constituyo una estructura jerárquica o de niveles en los sistemas operativos, el primero de los cuales fue denominado THE (Technische Hogeschool, Eindhoven), de Dijstra, que se utilizo con fines didácticos

Estructura Jerárquica

Se puede pensar también en estos sistemas como si fueran multicapa. Multics y UNIX caen en esa categoría

En la estructura anterior se basa prácticamente la mayoría de los sistemas operativos actuales. Otra forma de ver este tipo de sistema operativo es la denominada de anillos concéntricos o RINGs.

Estructura Jerárquica



1.3. Sistemas operativos de red

Sistemas operativos 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 canal de transmisión con el objetivo de intercambiar información, transferir archivos, ejecutar comandos remotos y un sin fin de otras actividades.

El punto crucial de estos sistemas operativos es que el usuario debe saber la sintaxis de un conjunto de comandos o llamadas al sistema para ejecutar operaciones, ademas de la ubicación de los recursos que se desean acceder.

1.4. Sistemas Distribuidos

Definición

- “Un sistema Distribuido es una colección de computadoras independientes o autónomas que aparecen ante los usuarios del sistema como una única computadora”.

  Andrew Tanenbaum

- “Es aquel en el que los componentes de hardware y software se localizan en computadoras unidos mediante red, comunican y coordinan sus acciones sólo mediante paso de mensajes”.

  George Coulouris

Sistemas Distribuidos - Características

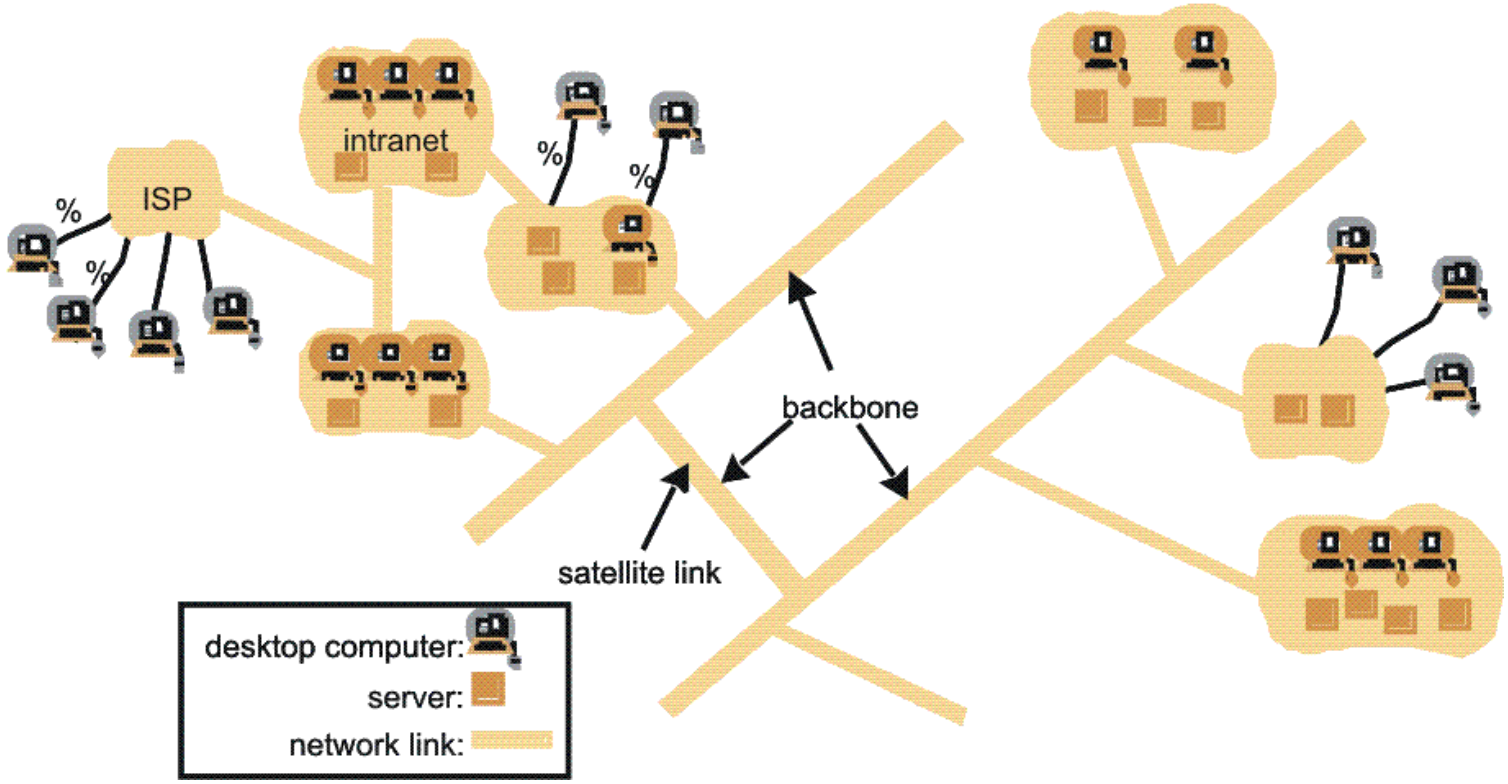
- Un conjunto de unidades con memoria propia.
- Sistemas globales (locales o remotos) para sincronizar y comunicar a todos los CPU´s.
- Algunos CPU´s pueden dejar de comunicarse con otros, pero el sistema distribuido no puede fallar en su totalidad.
- En caso de existir alguna falla en algunos CPU´s, deben existir formas de recuperar la información y el sistema debe de continuar funcionando.
- Deben existir sistemas de protección global del sistema.

Red VS Sistemas Distribuidos

- Diferencias entre Sistemas de Red y Sistemas Distribuidos
- En una red, los procesos se ejecutan en forma local autónoma. Estos procesos deben interactuar pero tomando decisiones locales sin tomar en cuenta procesos y recursos remotos. Se comparten recursos pero sólo en forma de comunicación.
- Un sistema distribuido es un sistema expandido en toda la red, pero visto como un solo sistema. Los procesos pueden suceder en forma local o remota sin que el usuario se de cuenta. La tolerancia a fallas es más alta. Las decisiones y los recursos son administrados en forma global.

Ejemplos de Sistemas Distribuidos

Internet

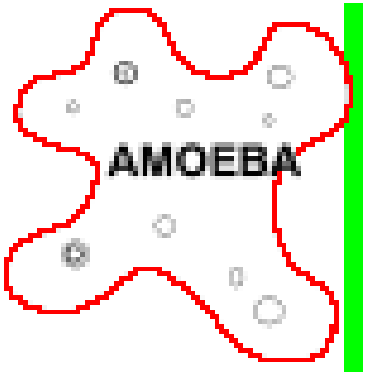


Distributed Systems: Concepts and Design, Coulouris, Dollimore and Kindberg, Third Edition, Addison Wesley 2001

1.5. Instalar un sistema Operativo Distribuido

1.6. Ejemplos de sistemas operativos distribuidos

Amoeba



Amoeba

Historia:

- El desarrollo de Amoeba inició en 1981 en Vrije Universiteit en Amsterdam Holanda como un proyecto de Cómputo Paralelo y Distribuido.
- Fue diseñado inicialmente por Andrew Tanenbaum y 3 estudiantes de doctorado, Frans Kaashoek, Sape J. Mullender y Robert Van Renesse.
- EN 1983 se logra Amoeba 1.0 como un prototipo, pero ya tenía un nivel operacional.

Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

Amoeba

Características:

- Inició desde cero sin preocuparse por la compatibilidad con otros sistemas operativos.
- El objetivo era crear un sistema operativo distribuido transparente.
- En Amoeba no hay máquina origen y destino, es decir cliente servidor, todas las máquinas hacen un todo.
- Las máquinas no tienen propietario.
- Cada nuevo proceso es ejecutado por la computadora con menor carga (balanceo de carga).
- Amoeba esta escrito en lenguaje C.

Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

Amoeba

Características:

- El sistema se diseñó pensando en implementarse en un ambiente con gran número de CPU´s cada uno con gran cantidad de memoria.
- El sistema no se basaba en memoria compartida.
- Puede utilizar CPU´s 680x0, 386 ó SPARC.
- Amoeba esta formado por un micronúcleo que es ejecutado en todas las computadoras.

Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

Amoeba

El micronúcleo se encarga de realizar las siguientes operaciones:

- 1) Controlar procesos e hilos.
- 2) Proporcionar el soporte de la administración de memoria de bajo nivel.
- 3) Soportar la comunicación.
- 4) Controlar la Entrada/Salida de bajo nivel.

Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

Amoeba

- Andrew Tanenbaum, creador de Amoeba.
- Actualmente es profesor de la División de Matemáticas y Ciencias de la Computación en Vrije Universiteit, en Amsterdam Holanda.

Andrew Tanenbaum http://www.cs.vu.nl/~ast/photos/wi/Tanenbaum_Andy.jpg mayo 2003

Amoeba

- Colección de 80 computadoras SPARC conectadas por medio de ethernet en Vrije Universiteit, Amsterdam Holanda, corriendo Amoeba.

The Amoeba Distributed Operating System http://www.cs.vu.nl/pub/amoeba/amoeba.html mayo 2003

MACH

MACH

Historia:

- El desarrollo de MACH inició en 1984 por Richard Rashid en Carnegie Mellon University, USA, y en 1986 apareció la primera versión para una computadora con 4 cpu´s, la VAX 11/784.

- Debido a DARPA se aisgnaron recursos para mejorar MACH y modificaron la versión 4.1 BSD (el UNIX de la Universidad de Berkeley) para incorporar el código de MACH.

- Posteriormente se combinó MACH con las versiones 4.2 y 4.3 de BSD lo cual permitió una completa compatibilidad de MACH para poder correr aplicaciones UNIX.

- Posteriormente la OSF (Open Software Foundation) elige a MACH como su sistema operativo y lo lanza como OSF/1.

Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

MACH

Historia:

- Para 1988 el núcleo de MACH 2.5 era grande y monolítico debido a la presencia de gran parte del código de UNIX BSD por lo que se decidió quitar del núcleo todo el código BSD y ponerlo en la parte del usuario, por lo que sólo quedó un micronúcleo de MACH.

- MACH sigue corriendo aplicaciones UNIX pero por medio de un emulador.

Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

MACH

Objetivos de desarrollo:

- 1) Proporcionar una base para la construcción de otros sistemas operativos (por ejemplo UNIX).
- 2) Soporte de espacio de direcciones de gran tamaño.
- 3) Permitir el acceso transparente a los recursos de la red.
- 4) Explotar el paralelismo tanto en el sistema como en las aplicaciones.
- 5) Hacer que MACH se pueda transportar a una colección más grande de máquinas.

Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

MACH

Características del Micronúcleo:

- El desarrollo del micronúcleo se realizó pensando en emular sistemas operativos como UNIX.
- La emulación se lleva a cabo mediante una capa de software que se ejecuta fuera del núcleo, en el espacio del usuario.
- Se pueden ejecutar varios emuladores al mismo tiempo, por lo que es posible ejecutar programas en 4.3BSD, UNIX System V y MS-DOS, en la misma máquina y al mismo tiempo.

Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

CHORUS

CHORUS

Historia:

- Surge en 1980 en INRIA Francia y se desarrollaron sólo 4 versiones (de la 0 a la 3).
- Es un sistema operativo distribuido que se basa en una colección de actores. Un actor es en realidad un autómata de estado finito.
- Cada máquina ejecuta el mismo núcleo del sistema operativo.
- La versión 0 de Chorus se desarrolló en Pascal.

Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

CHORUS

Objetivos de desarrollo:

- 1) Emulación de UNIX de alto rendimiento.
- 2) Uso en Sistemas Distribuidos.
- 3) Correr aplicaciones en tiempo real.
- 4) Integración de programas orientados a objetos.

Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

CHORUS

Partes de CHORUS:

1) Núcleo:

- Se encarga de la administración de nombres, procesos, hilos, memoria y comunicación.

2) Procesos del Núcleo:

- Se cargan y eliminan de manera dinámica durante la ejecución del sistema.

3) Procesos del Sistema:

- Se ejecutan en modo usuario y junto con los procesos del núcleo forman un subsistema.

4) Procesos del Usuario:

- Aquí se encuentran los procesos del usuario que se encargan de llamar a los procesos del sistema.

Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

CHORUS

Características:

- El uso de subsistemas permite construir nuevos sistemas operativos sobre el micronúcleo de manera modular.
- Un proceso posee ciertos recursos y cuando el proceso termina se liberan sus recursos.
- Dentro de un proceso pueden existir uno o más hilos.
- Cada hilo tiene su propia pila, código y registros o datos.
- Todos los hilos de un proceso comparten el mismo espacio de direcciones.
- Los hilos son independientes entre sí.
- Los hilos de los procesos se pueden comunicar entre sí por medio de transferencia de mensajes.
- Para comunicarse se utilizan puertos.
- Cada puerto pertenece a un proceso.
- Chorus tiene un subsistema llamado Mix el cual es compatible con Unix System V.
- La versión Mix 3.2 es compatible con BSD 4.2

Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

Plan 9

Plan 9

Historia:

- A mitades de los 80´s se utilizaban grandes computadoras centralizadas conectadas a pequeñas computadoras normalmente estaciones de trabajo UNIX.
- UNIX es un sistema de tiempo compartido que tiene problemas con nuevos módulos que se le han integrado como los de gráficos y comunicación en red.
- Plan 9 se empieza a finales de los 80´s y se buscaba un sistema formado por microcomputadoras que realizaran diferentes tareas y que estuvieran conectadas a grandes servidores compartidos.

Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Plan 9

Historia:

- Se creó un nuevo protocolo a nivel de red llamado P9 que permite a las computadoras acceder a los archivos en sitios remotos.
- Para 1989, Plan 9 ya era usado en diferentes partes como sistema principal, el cual ya contiene nuevos compiladores, lenguajes, librerías, sistemas de ventanas y nuevas aplicaciones.
- Para permitir compatibilidad con UNIX se creó un emulador que corre en una ventana, el cual permite ejecutar comandos POSIX, pero todo el sistema corre en Plan 9.

Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Plan 9

Características:

- Los recursos tienen nombres y son accesados como archivos en orden jerárquico.
- Para nombrar recursos se tienen espacios locales de nombres y espacios globales de nombres donde los procesos buscan los recursos que necesitan, ya sean locales o globales.
- Para acceder a los recursos se crea el protocolo P9.
- Se tiene un número de computadoras conectadas entre sí, cada una realizando un servicio en particular.
- Esta formado por multiprocesadores compartidos que proveen ciclos de cómputo al sistema distribuido.

Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Plan 9

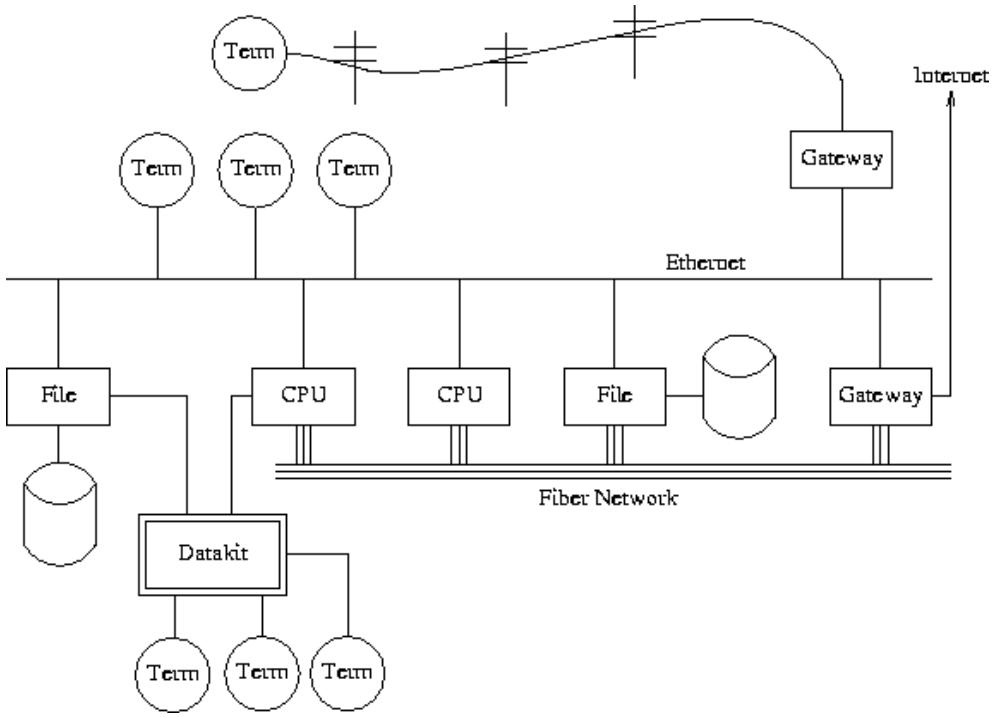
Características:

- Algunas computadoras se dedican a almacenamiento de archivos.
- Estas computadoras están conectadas por una red de alto rendimiento.
- Los clientes o terminales del sistema (por lo general PC´s), se conectan a los servidores mediante redes de bajo rendimiento (Ethernet ó ISDN).
- Cuando alguien utiliza una PC como terminal de Plan 9, se crea una terminal especial (en software) para éste usuario determinado con las características de sus variables locales de entorno (indicando tipo de video) esto es para evitar configurar en forma manual cada terminal en base al hardware que tienen.

Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Plan 9

Arquitectura de Plan 9:



Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Plan 9

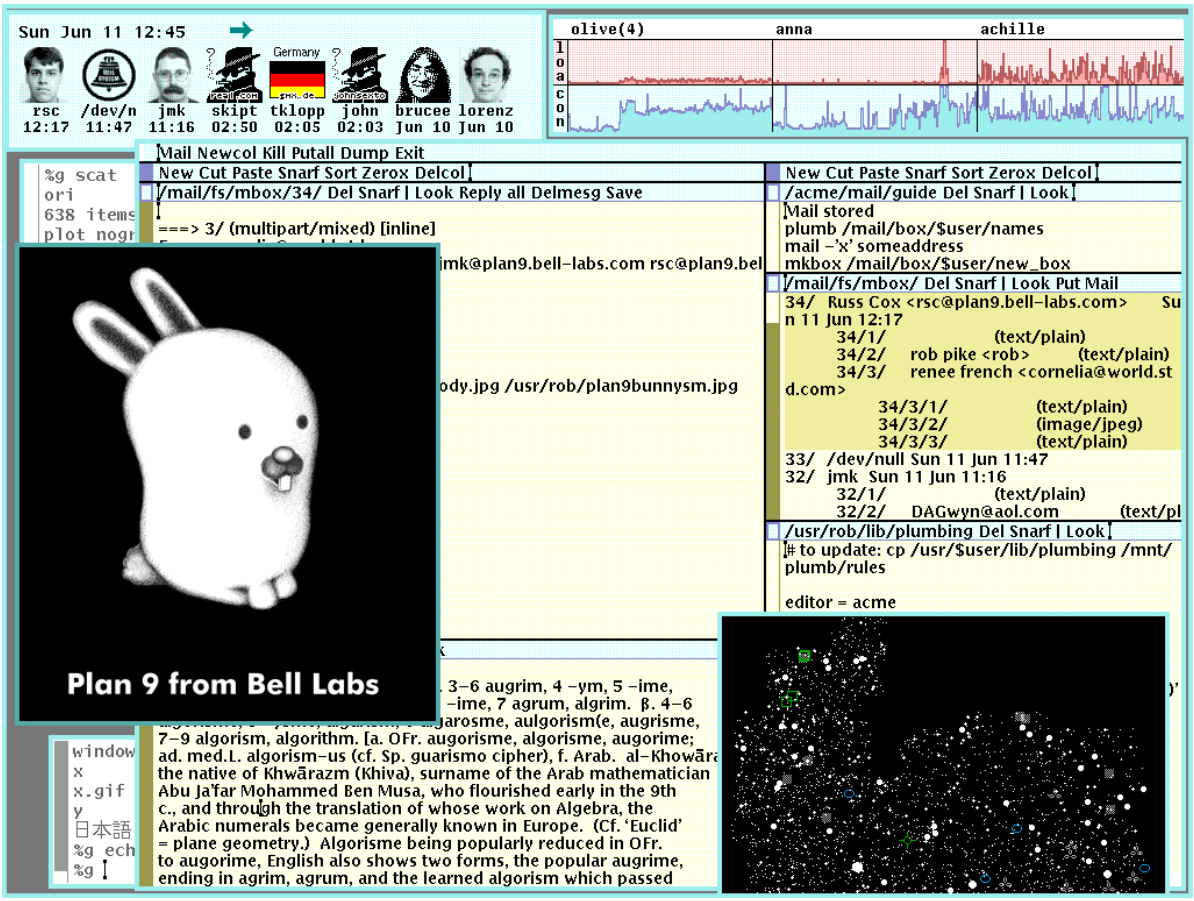
Características:

- Plan 9 es portable en varias plataformas y puede usar microprocesadores intel en la ventana de una PC terminal y comunicarse con un servidor SPARC con cpu´s MIPS.
- Para procesos paralelos, se creó un lenguaje propio para realizar programación paralela que se llama ALEF.

Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Plan 9

Pantalla de Plan 9:



Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Plan 9

Pruebas de comparación:

Prueba	Plan 9	IRIX
Context Switch	39 µs	150 µs
System Call	6 µs	36 µs
Light Fork	1300 µs	2200 µs
Pipe Latency	110 µs	200 µs
Pipe Bandwidth	11,678 Kb/s	14,545 Kb/s

Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Bibliografía

- Distributed Systems: Principles and Paradigms Andrew Tannenbaum and Maarten van Steen, Prentice Hall, 2001
- Distributed Operating Systems & Algorithms Randy Chow, Theodore Johnson, Yuan-Chieh Chow Addison Wesley Publishing Company (March, 1997)
- Distributed Operating Systems, Andrew Tannenbaum, Prentice Hall; 1st edition (August 25, 1994)
- Sistemas Operativos Modernos, Tenenbaum, Andrew S. Ed. Prentice Hall, 2001
- Sistemas Operativos una visión aplicada, Carretero, Pérez Jesús; García Caballeira Félix; Anasagasti Pedro de M.; Pérez C. Fernando. Mc Graw Hill, 2003


HUMANISMO QUE TRANSFORMA
2013-2017