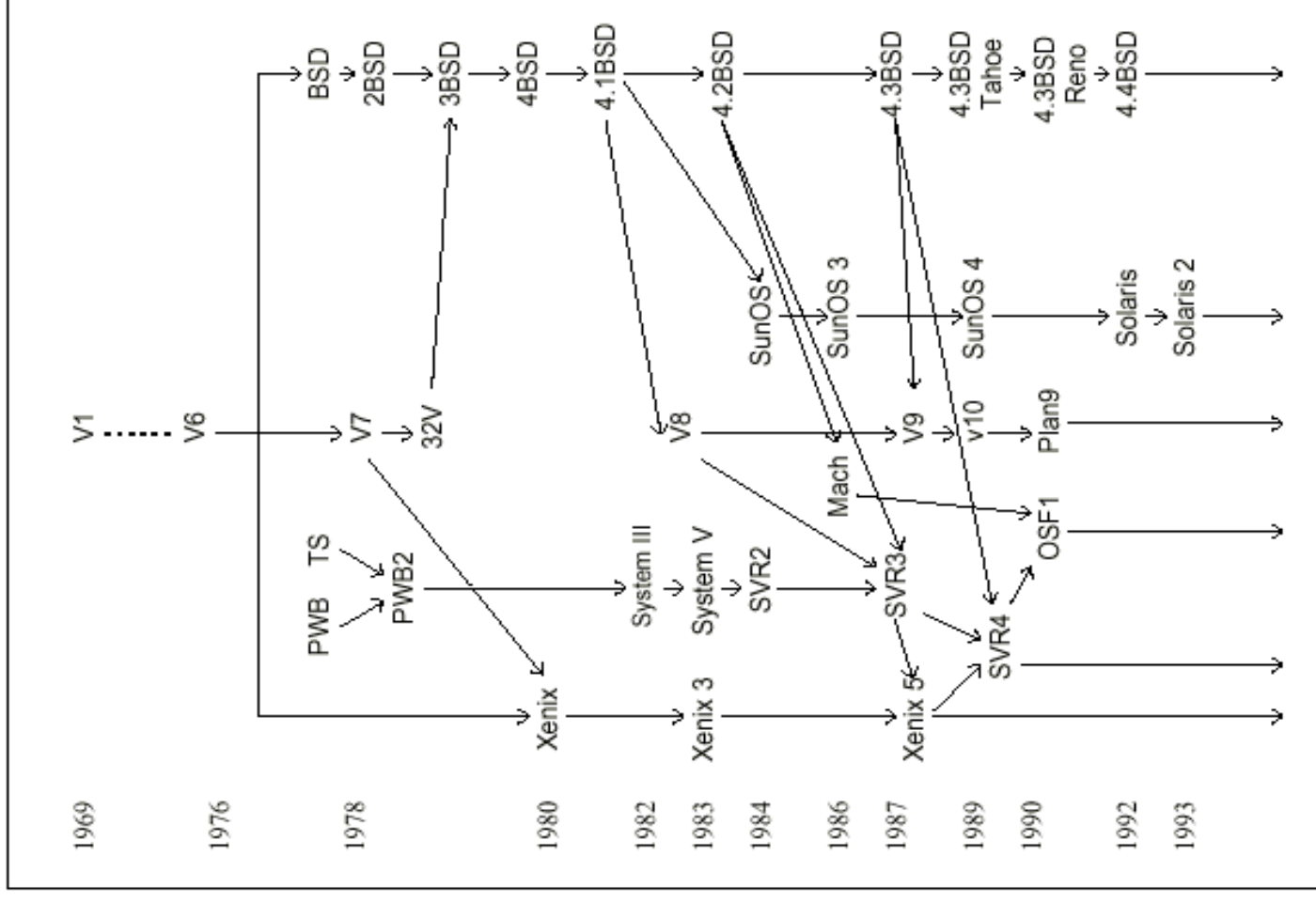


Historia de UNIX

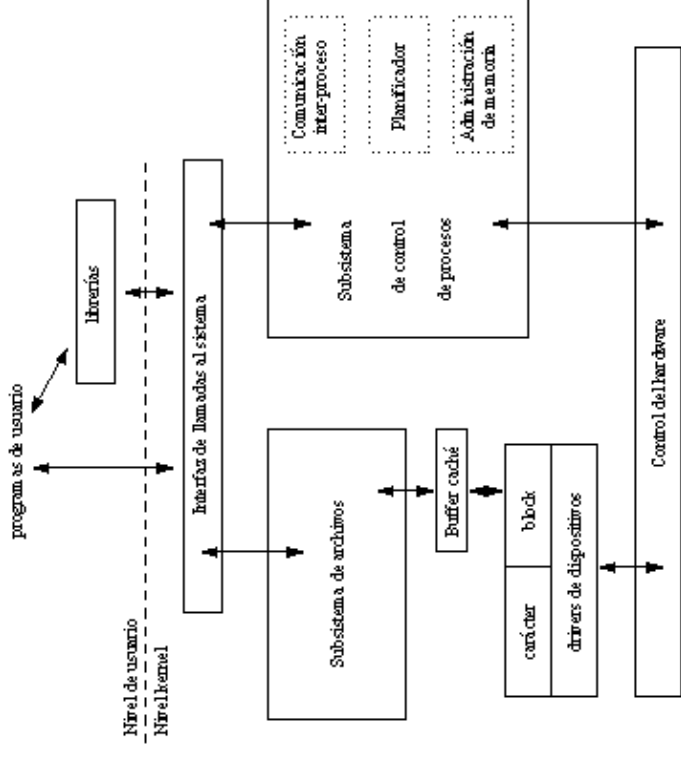
- 1969 -> desarrollo de Multics (Ken Thompson y Dennis Ritchie en los laboratorios Bell de AT&T)
- Versión reducida de Multics sobre un PDP-7
- Ritchie crea B y posteriormente C
- 1973 Unix se escribe en C para un PDP-11
- 1974 Versión V6->Se lleva a distintas universidades junto con el código fuente
- División:
 - BSD -> universidad de Berkeley
 - System V -> AT&T
 - Xenix -> Microsoft

- AT&T desarrolla una versión comercial llamada System III
- BSD → versión desarrollada por la Universidad de Berkeley.
- AT&T sigue con el desarrollo de su versión hasta la V10
- SCO desarrolla SVR a partir de System V. En 1993 Novel compra los derechos de Unix a AT&T y en 1995 esta los vende a SCO
- Sun desarrolla SunOS (luego Solaris) a partir de BSD4.1
- Finales 80
 - POSIX
 - X11
 - MOTIF



Características

- Multiusuario y multitarea
- Protección:
- Modos de ejecución:
 - Núcleo se ejecuta en modo supervisor
 - Aplicaciones modo usuario
 - Llamadas al sistema
- Control de memoria
 - Límites de zona de memoria
 - Llamadas de petición y liberación
- Control de CPU
 - Asignación rotativa por prioridades
 - Control de accesos
- Trata los dispositivos como ficheros
- Sistema de ficheros jerárquico
- SO. de red. Muy ligado a TCP/IP
- Memoria virtual: SWAP

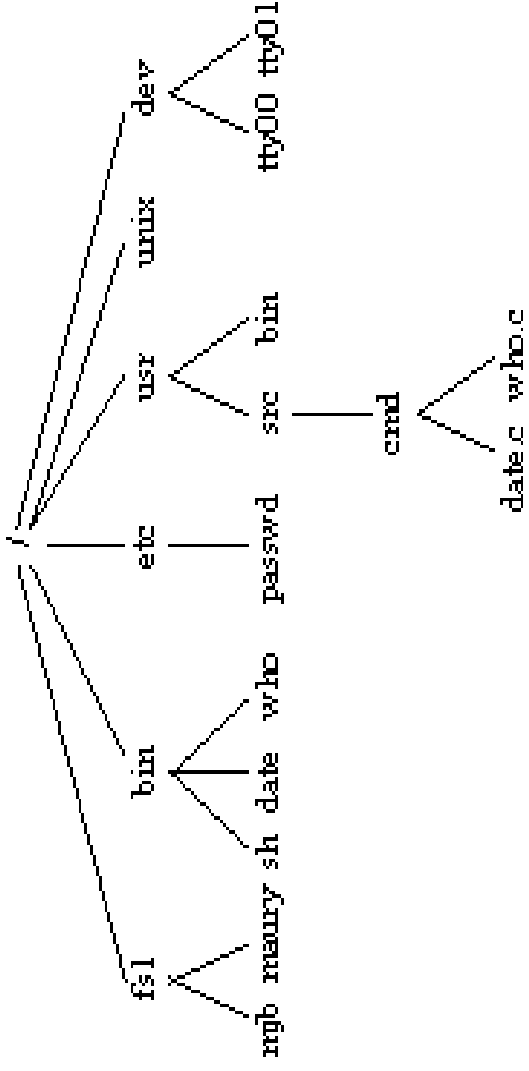


Gestión de procesos

- Proceso identificado por su PID
- Modos de ejecución kernel y usuario
 - Determinadas instrucciones no se pueden usar en modo usuario
- Contexto de un proceso:
 - Valores de variables y estructuras de datos
 - Registros
 - Pila de usuario, contador de programa, etc.
- Asignación de CPU
 - Rotación por prioridades
 - Cambio de contexto
- Comunicación entre procesos
 - Señales (sucesos asíncronos)
 - Pipes → canales unidireccionales
 - Sockets (comunicación remota)

Sistema de archivos

- Sistema jerárquico



- Los distintos sistemas de archivos se montan en el sistema de ficheros
- Los dispositivos “son” archivos
- Tipos de archivos: normales, directorios, dispositivos, enlaces simbólicos

Los archivos

- Formato de archivos:
 - Nombres hasta 255 caracteres, y con más de un punto en ellos: por ejemplo, Este.es.un.nombre.de.fichero.MUY.largo.
 - sensible a las mayúsculas y las minúsculas en los nombres de ficheros o comandos
 - no hay extensiones obligadas como .COM y .EXE para los programas, o .BAT para los ficheros de procesamiento por lotes
- Enlaces simbólicos
 - Es un puntero o enlace a un fichero o a un directorio
 - Puede utilizarse en lugar del fichero o del directorio al que apunta
 - Similar a los "Accesos Directos" de Windows 95
- Permisos y propiedades
 - `-rwxr-xr-x 1 root bin 27281 Aug 15 1995 fich1`
 - `-rwxr-xr-x = permisos`
 - `root = propietario`
 - `bin = grupo`
 - `27281 = tamaño`
 - `Aug 15 1995 = fecha`
 - `fich1 = nombre del fichero`

Permisos

- **rwX rxw rwx**
- primer campo: tipo de archivo:
 - **-** = fichero normal
 - **d** = directorio
 - **l** = enlace simbólico
- segundo campo: permisos para el propietario:
 - **r** = lectura
 - **w** = modificación (incluye borrado)
 - **x** = ejecución (entrar si es un directorio)
- tercer campo: permisos para el grupo
- cuarto campo: permisos para el resto de usuarios

Control de accesos

- Entrada al sistema:
 - Login:
 - Password:
- Cambio de contraseña:
 - passwd
- Consolas virtuales → alt-Fx o cntrl-alt-Fx
- Salida del sistema
 - exit
 - cntrl-d
- Apagar el sistema: (solo root o usuarios autorizados)
 - halt
 - suthdown
 - cntrl-alt-supr

Algunos comandos de unix

- `cd` Cambia el directorio de trabajo actual.

Sintaxis: `cd <directorio>`

`<directorio>` es el directorio al que cambiamos. ("." se refiere al directorio actual, ".." al directorio padre.)

- `ls` Muestra información sobre los ficheros o directorios indicados.

Sintaxis: `ls <fichero1> <fichero2> ...<ficheroN>`

- Uso de comodines * y ?

- `cp` Copia fichero(s) en otro fichero o directorio.

Sintaxis: `cp <fichero1> .. <ficheroN> <destino>`

- `mv` Mueve fichero(s) a otro fichero o directorio. Sintaxis igual que `cp`

- `rm` Borra ficheros. Los ficheros borrados son irre recuperables

Sintaxis: `rm <fichero1> <fichero2> ...<ficheroN>`

- `mkdir` Crea directorios nuevos.
Sintaxis: `mkdir <dir1> <dir2> ...<dirN>`
- `rmdir` Esta orden borra directorios vacios. Al usar `rmdir`, el directorio de trabajo actual no debe de estar dentro del directorio a borrar.
Sintaxis: `rmdir <dir1> <dir2> ...<dirN>`
- `man` Muestra la pagina de manual del comando o recurso
- `more` Muestra el contenido de los ficheros indicados, una pantalla cada vez.
Sintaxis: `more <fichero1> <fichero2> ...<ficheroN>`
- `cat` Oficialmente usado para concatenar ficheros, `cat` tambien es usado para mostrar el contenido completo de un fichero de una vez.
Sintaxis: `cat <fichero1> <fichero2> ...<ficheroN>`

- **grep** Muestra todas las líneas de un fichero dado que coinciden con un cierto patrón.
Sintaxis: `grep <patron> <fichero1> ...<ficheroN>`
- **find**: busca archivos. Permite ejecutar programas condicionalmente
Sintaxis: `find <directorio> <expresión>`
- **Redirección:**
 - `pipes -> |`
 - Redirección de la salida `-> orden > salida`
 - Redirección de la entrada `-> orden < entrada`
- **Control de procesos**
 - `ps ->` muestra los procesos activos:
 - `ctrl-C ->` aborta la ejecución de un programa
 - `ctrl-z -Z` interrumpe la ejecución de un programa (lo deja "suspendido")
 - `bg ->` manda un proceso a segundo plano (background)
 - `fg ->` manda un proceso a primer plano (fg)
 - `kill ->` mata un proceso

Configuración

- `/etc --> configuración global`
 - `/etc/passwd` y `shadow --> usuarios y contraseñas`
 - `/etc/group --> grupos`
 - `/etc/profile` establece el `PATH` y otras variables, etc.
 - `/etc/bashrc` define alias y funciones, etc.
- `/home/mi_home/ --> configuración de usuario`
 - `/home/mi_home/.bashrc` define mis alias y mis funciones
 - `/home/mi_home/.bash_profile` establece el entorno y ejecuta mis programas
 - `/home/mi_home/.profile` idem