

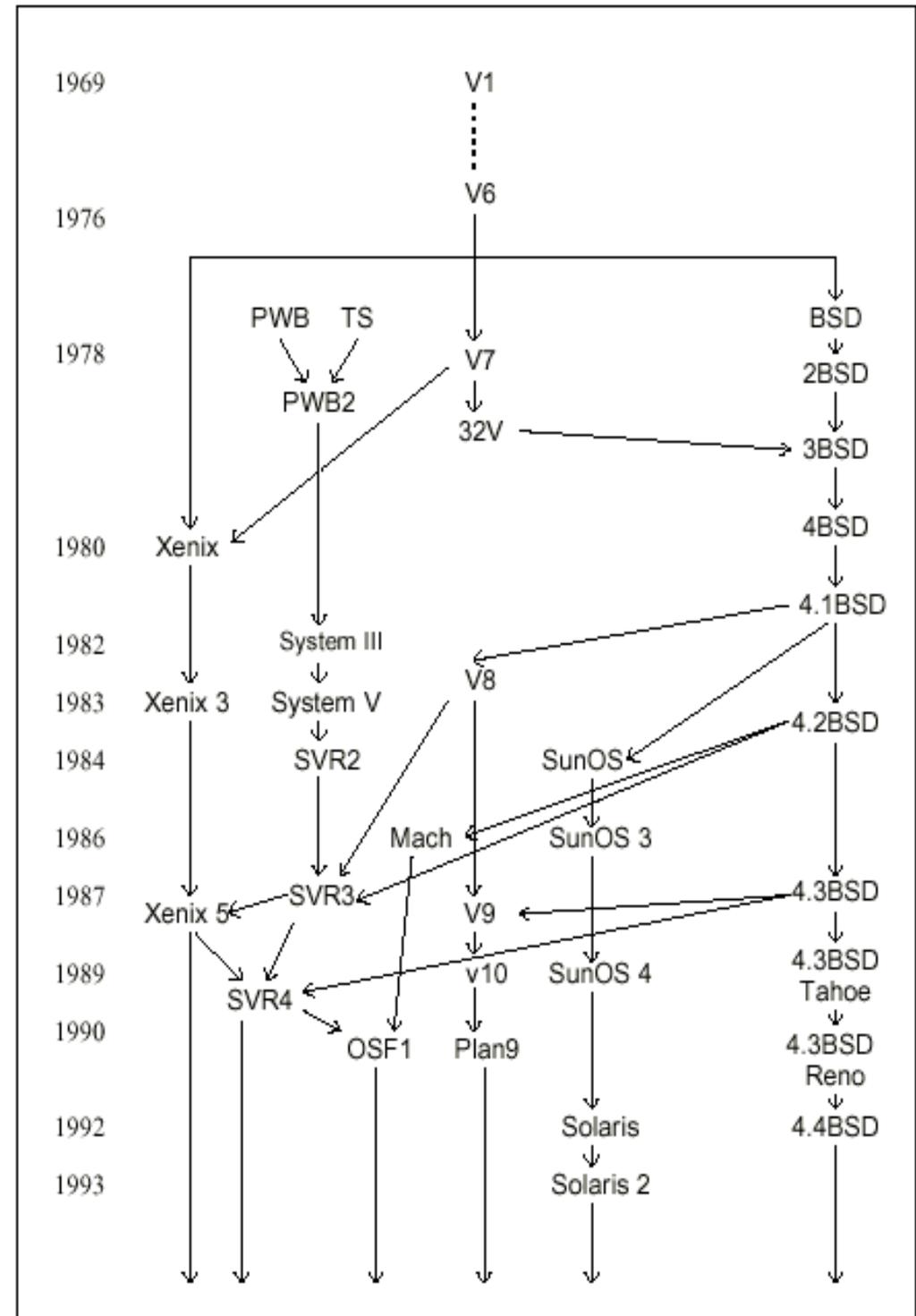
# Historia de UNIX

- 1945-55
  - Computadores de tubos de vacío
  - No hay sistema operativo ni programación en ensamblador.
  - No hay programadores y operadores, una sola persona hace todo el trabajo.
  - Programación mediante tableros de conmutación (interconexiones “a mano”)
- 1955-65
  - Computadores basados en transistores
  - Sistemas por lotes
  - Programador -> programa en ensamblador o fortran en tarjetas perforadas
  - Operador -> introduce las tarjetas y recoge los resultados.
  - Sistema operativo -> carga los trabajos, los ejecuta e imprime los resultados

- 1965-1980
  - Computadoras basadas en circuitos integrados
  - Línea de computadores 360 de IBM -> sistema OS/360.
    - Compatible con distintos sistemas
    - Introduce la multiprogramación (pero no el tiempo compartido -> no interactivo)
  - CTSS (Compatible Time Sharing System)
    - Desarrollado en el MIT sobre un IBM7090 (32K RAM, 3000 m<sup>2</sup>)
    - Primer sistema operativo serio de tiempo compartido (hasta 32 usuarios)
  - MULTICS(Multiplexed Information and Computing Service)
    - Desarrollado sobre un GE-645 -> sistema de 36 bits, multiprocesador con hardware de protección y memoria virtual segmentada y paginada. Memoria de 256Kx36.
    - Proyecto de S.O. Para dar servicio a miles de usuarios.
    - Sistema operativo de tiempo compartido -> interactivo
    - Escrito casi entero en PL/1
    - Llega a ser operativo y usarse en el MIT, aunque no a tan gran escala como se pretendía (llega a 300 usuarios)

- 1968: UNICS (Uniplexed Information and Computing System)
  - Ken Thompson programa un juego de marcianos “Space Travel” sobre MULTICS -> funciona muy lento
  - Junto con Dennis Ritchie decide portarlo a una minicomputadora DEC PDP-7 => escribe un S.O. Para un sólo usuario -> UNICS
- UNIX
  - 1969: Se añade soporte para dos usuarios y el S.O. Se lleva a un PDP11/20 (minicomputadora de 16bits con 64KB RAM) Se convierte en UNIX, licenciado por AT&T
  - Thompson y Ritchie crean el lenguaje B.
  - 1972 -> aparece la segunda versión de UNIX, escrita en B (10 usuarios)
  - Mejoran el lenguaje => lenguaje C. Tercera versión de UNIX parte en C (16 usuarios). Incorpora el uso de “pipes”
  - 1973 -> cuarta versión, completamente en C. 33 usuarios => portabilidad.
  - 1975 -> versión 6. Sale de los laboratorios de AT&T. Por ley AT&T no puede ganar dinero con productos que no sean de Telefonía => licencia el S.O. Y el código fuente para uso gratuito => se lleva a universidades.
  - 1978 -> cambio de leyes. AT&T puede vender el S.O. => continúa con su desarrollo

- 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 SystemV. 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



# Software libre en torno a UNIX

- 1983 -> Richard Stallman crea el proyecto GNU y la FSF
  - FSF = Free Software Foundation -> propone
    - Usar el programa, con cualquier propósito
    - Estudiar cómo funciona el programa, y adaptarlo a tus necesidades
    - Distribuir copias
    - Mejorar el programa y hacer públicas las mejoras a los demás
  - Licencia Pública GNU (GPL)
    - Permite la redistribución binaria y la de las fuentes
    - Permite realizar modificaciones sin restricciones, aunque sólo se pueda integrar código licenciado bajo GPL con otro código que se encuentre bajo una licencia idéntica o compatible -> no puede dejar de ser libre.
  - Proyecto GNU
    - Proyecto de sistema operativo libre -> S.O. Compatible UNIX
    - Incluye el núcleo del S.O. Y las herramientas.
    - El sistema operativo no ha llegado a realizarse, pero las herramientas se ha utilizado junto con otros núcleos libres, como GNU/Linux.

- **LINUX**

- 1991 Linux Torvalds (estudiante de la Universidad de Helsinki) lanza el desarrollo de Linux
- Núcleo de S.O. Compatible UNIX.
- Basado en MINIX
- Se propone como sistema operativo libre. Se licencia bajo GPL.
- Su desarrollo va muy ligado a internet -> el desarrollo se realiza a partir de una lista de correo.
- Se utiliza herramientas y software GNU para realizar un S.O. completo.
- 1994 -> versión 1.0 (primera versión estable)

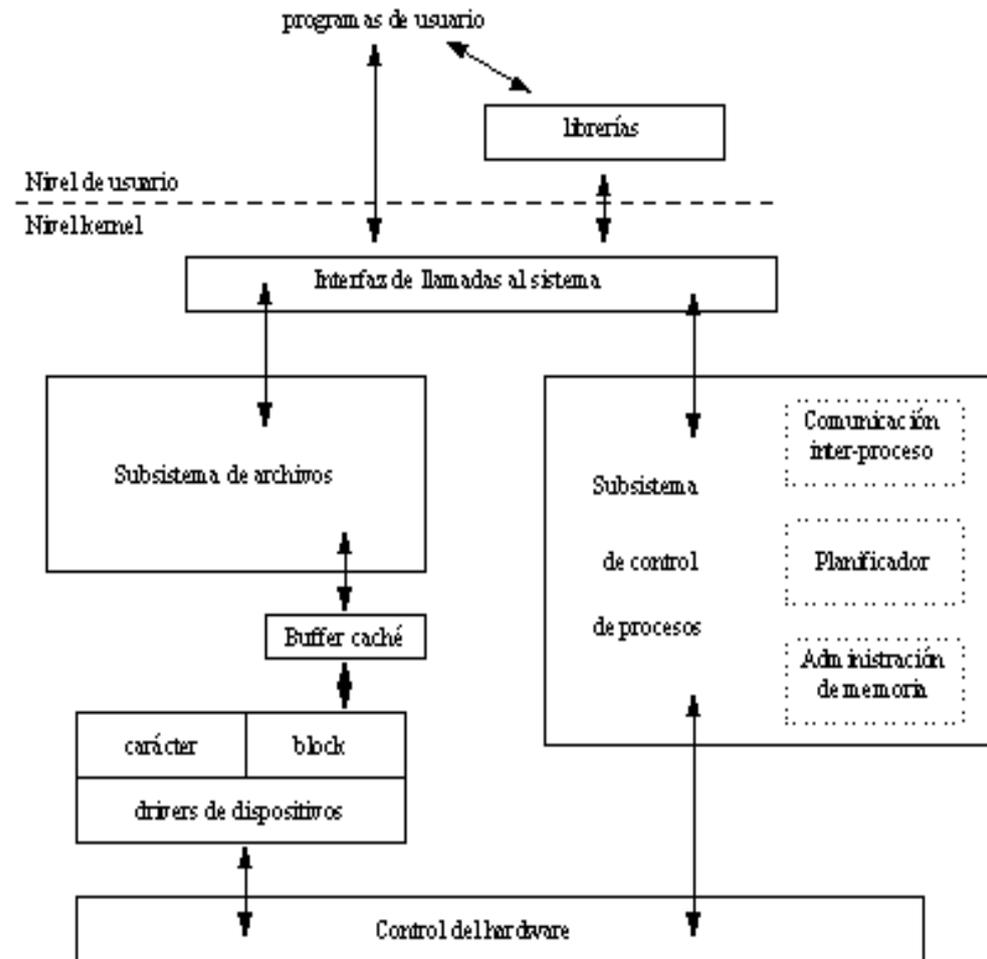
- **FreeBSD**

- Heredera de la distribución UNIX de la Universidad de Berkeley
- Versión 1.0 en 1993 -> con código procedente del UNIX de AT&T
- Versiones paralelas: NetBSD y OpenBSD

- **Otro software libre: KDE, GNOME, OpenOffice, Apache, Mozilla..... sourceforge.....**

# 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

- Llamadas al sistema:
  - Mecanismo para solicitar al núcleo el uso de recursos
  - Los programas realizan dos tipos de llamadas al sistema
    - Directas
    - Mediante librerías
- Subsistema de archivos -> controla los recursos del sistema de archivos
  - Manipulación de archivos
  - Reserva de espacio de memoria para los archivos
  - Control de acceso a los archivos
  - Administración del espacio libre
  - Intercambio de datos archivo-usuario

- Subsistema de control de procesos
  - Sincronización de procesos
  - Comunicación entre los procesos
  - Control de la memoria principal -> incluye el subsistema de manejo de memoria
    - Controla la asignación de memoria a los procesos
    - Colabora con el hardware en la traducción de direcciones virtuales
  - Planificación de procesos -> incluye el planificador
- Manejadores de dispositivos (drivers) -> comunicación núcleo-dispositivo
- Módulo de control de hardware -> responsable del manejo de las interrupciones y la comunicación con la máquina

# Procesos en UNIX

- Llamada `fork()` -> creación de un nuevo proceso
- Estructuras asociadas (memoria)
  - Tabla de procesos -> una entrada por cada proceso
  - Tablas de segmentos y de segmentos por proceso -> uso de memoria
- Datos asociados a un proceso
  - UID y GID (y UID efectivo)
  - Estado del proceso (modo supervisor, modo usuario, dormido, preparado para ejecución, zombi)
  - Temporizadores de ejecución
  - Errores
  - Tabla de descriptores de ficheros abiertos por el proceso

- Contexto de un proceso

- Incluye mucha información sobre el proceso:

- Parte estática: código del proceso, datos del proceso, pila del usuario, memoria compartida
    - Parte dinámica: todos los registros del uP
    - Contexto del sistema: entradas en la tabla de procesos, segmentos, etc, pila del núcleo...

- Cambio de contexto

- Se produce en: interrupciones, llamadas al sistema o por petición del núcleo (p.e. Dormir)
    - Se guarda el contexto del proceso.
    - Para reanudar la ejecución, se recupera el contexto.

- Asignación de CPU

- Rotación por prioridades -> varias colas de prioridad. Prioridad aumenta con tiempo de espera

- Cambio de contexto

- Comunicación y sincronización entre procesos

- Señales (sucesos asíncronos, p.e. SIGTERM o SIGKILL)
  - Pipes -> canales unidireccionales -> todos los procesos pueden redirigir su entrada y su salida
  - Semáforos -> acceso a recursos compartidos
  - Mensajes -> comunicación y sincronización/acceso a recursos compartidos
  - Sockets (comunicación remota)

# Sistema de archivos

- Estructuras de datos

- En el disco

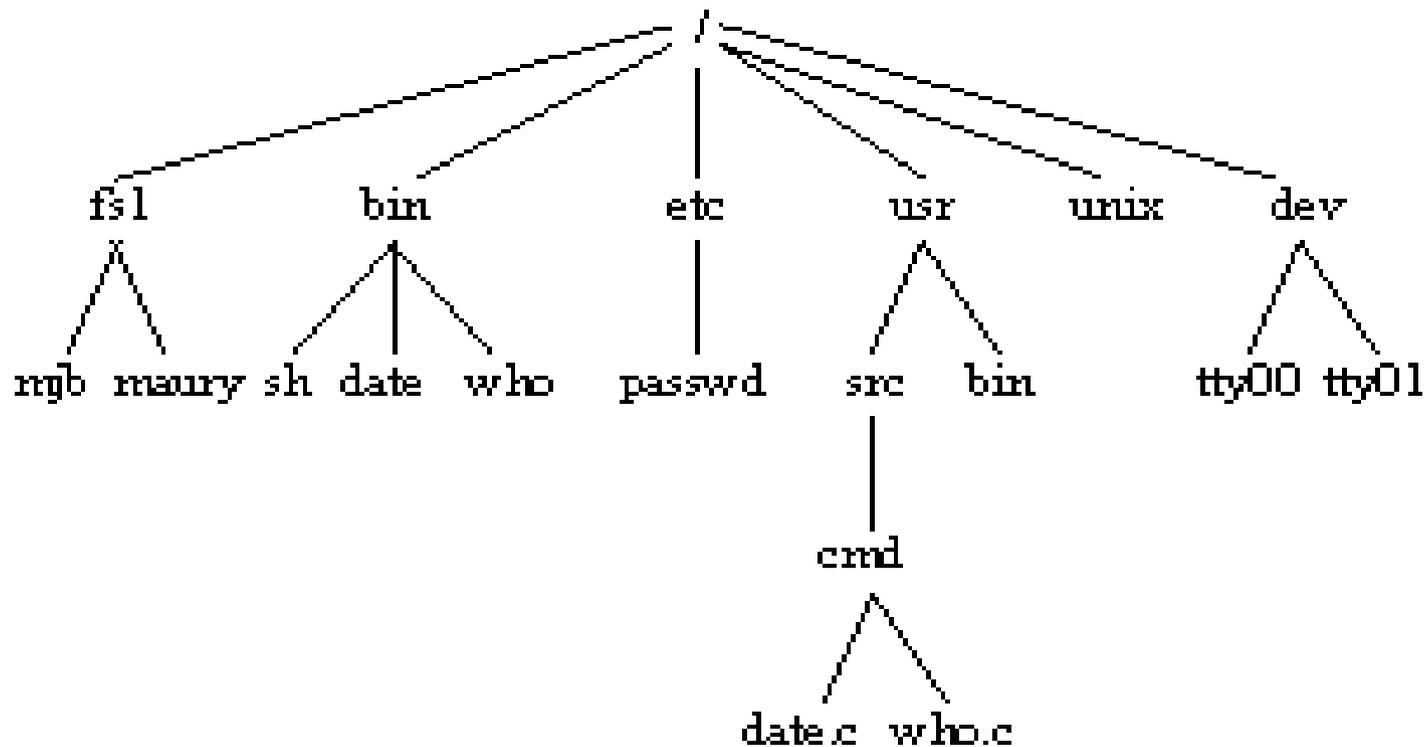
- Boot -> arranque
    - Superbloque -> tamaño, capacidad, espacio libre, fechas de montaje, etc.)
    - Tabla de inodos -> cada inodo: propietario y grupo, tipo de fichero, tiempos de acceso, punteros a los datos (directos, 1 indirecto simple, 1 indirecto doble, 1 indirecto triple)
    - Directorio -> inodo-nombre
    - Bloques de datos

- En memoria

- Tabla de inodos -> añade información extra: inodo bloqueado, identificador único (numero de dispositivo lógico+número inodo), inodo modificado, etc.
    - Tabla de descriptores de archivo (de cada proceso)
      - Archivos abiertos por el proceso -> apuntan a la tabla de archivos
    - Tabla de archivos -> estructura global (del núcleo)
      - Una entrada por cada archivo abierto (todos)
      - Información: modo de apertura, inodo, derechos de acceso, posición en el archivo (puntero R/W), etc.

- VFS (sistema de archivos virtual)

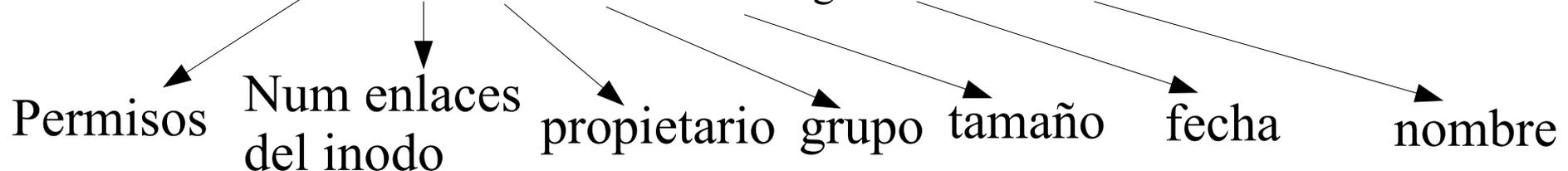
- Todo en UNIX es un fichero -> se trata como fichero cualquier dispositivo.
- Un sólo sistema de archivos -> se montan dispositivos, sistemas de ficheros remotos .....
- Sistema de ficheros jerárquico
- Tipos de archivos: normales, directorios, dispositivos, enlaces simbólicos



# 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 (llamar a un mismo archivo de varias formas)
  - Puede utilizarse en lugar del fichero o del directorio al que apunta
  - Similar a los "Accesos Directos" de Windows 95
  - Enlaces duros (hard link)
    - Dos entradas en el directorio apuntan al mismo inodo (inodo con dos nombres)
    - Sólo en el mismo sistema de archivos
  - Enlaces simbólicos
    - El archivo apunta al nombre de otro archivo
    - Pueden ser archivos en distintos sistemas de archivos
- Permisos y propiedades

- -rwxr-xr-x 1 root bin 27281 Aug 15 1995 fich1



# 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

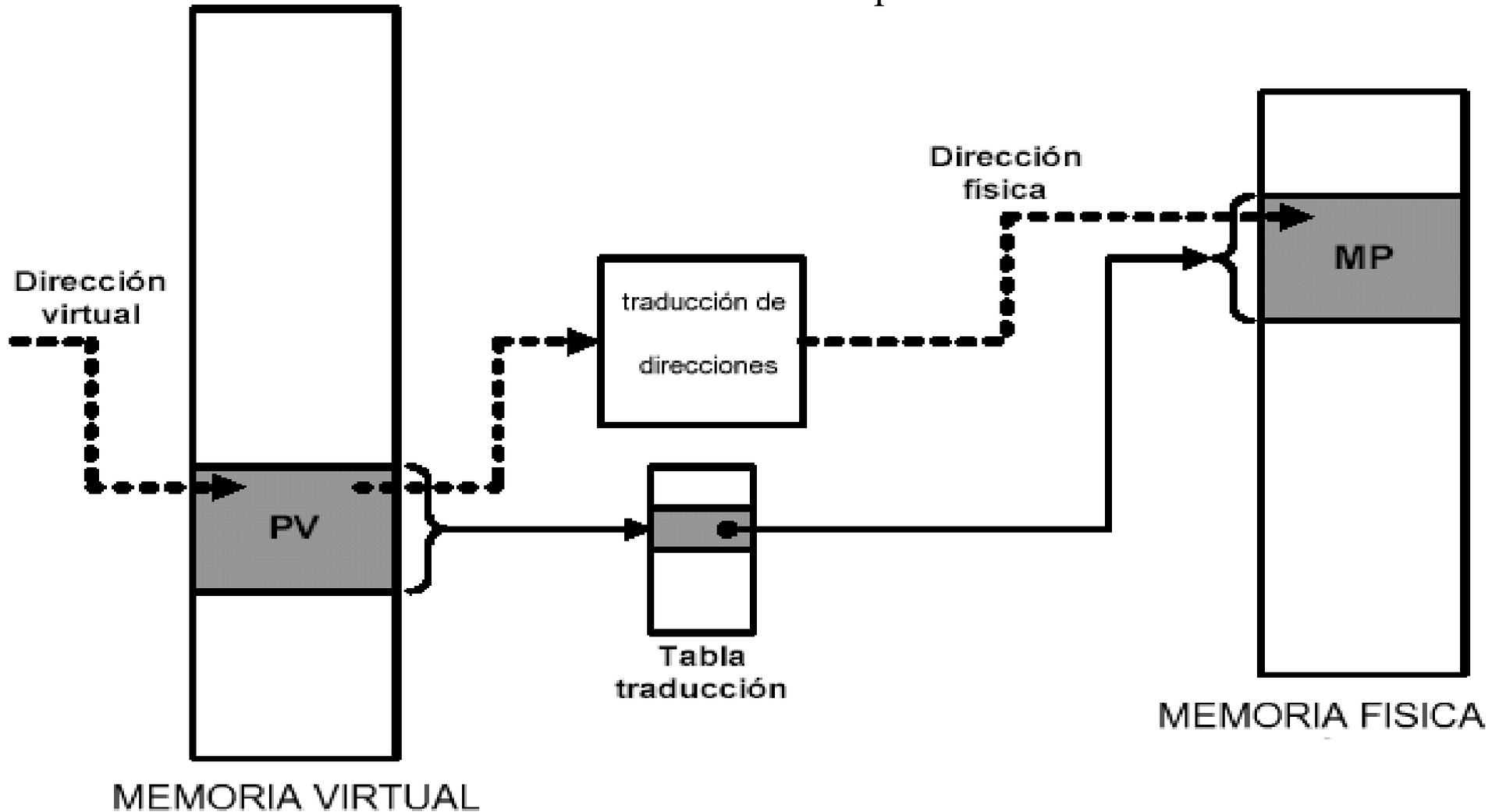
# Memoria virtual

- Características

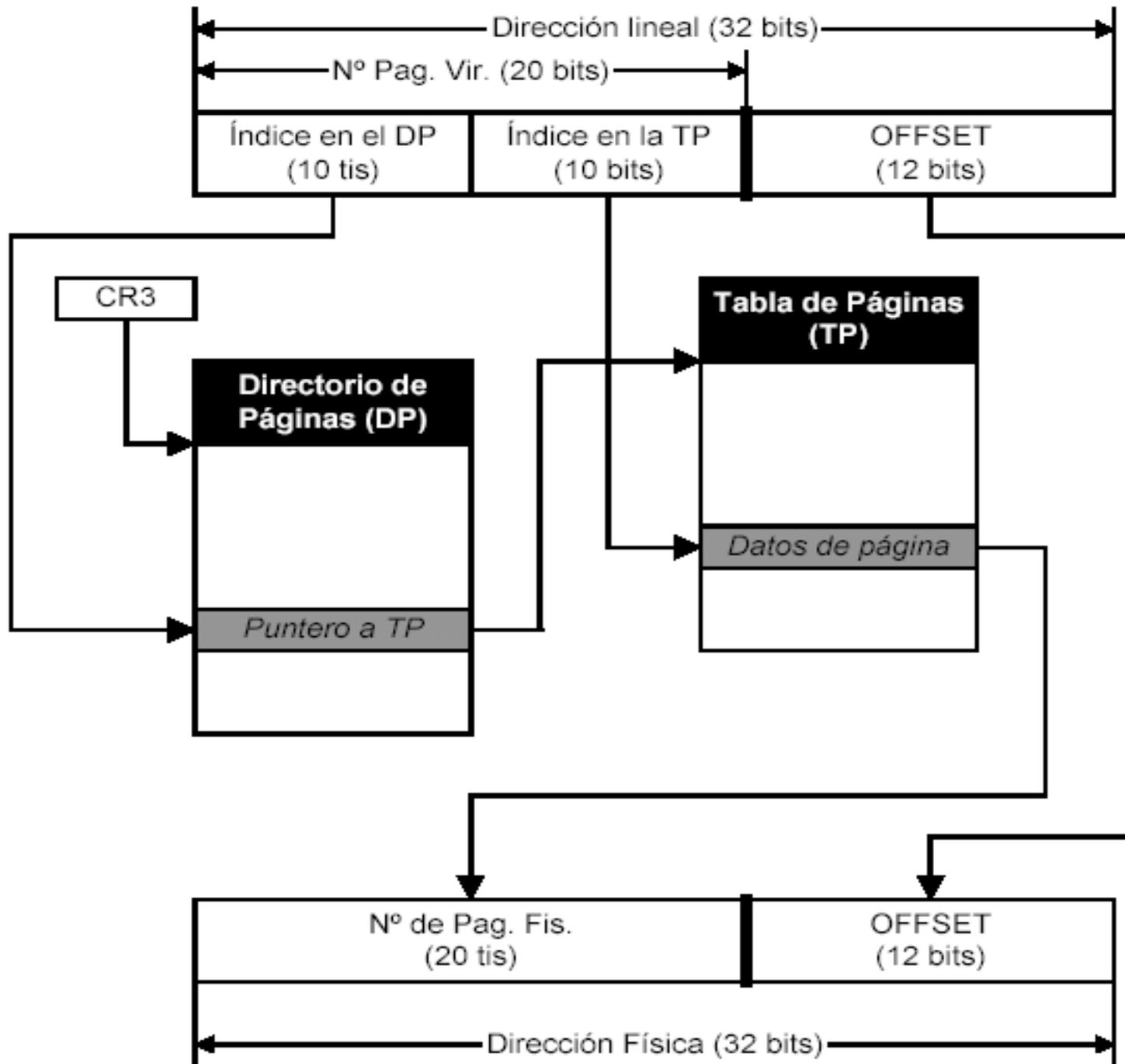
- Paginación por demanda
- Intercambio (swapping)

- Entradas en tabla de páginas

- Dirección física
- Bits de modificado y presencia
- Bits protección



# Direccionamiento virtual en arquitectura i386



# La E/S en UNIX

- Dispositivos
  - Todo “son” archivos -> bajo /dev
  - Tipos
    - Orientados a carácter -> ratón, teclado, impresora, t.sonido, t.video, modem....
    - Orientados a bloque -> dispositivos de almacenamiento
- Controladores de dispositivo
  - Son procesos
  - Llamada a través del núcleo -> llamada al sistema->núcleo->driver->hardware
  - Abstracción del hardware -> programas independientes del dispositivo
- Buffers
  - Asociados a cada dispositivo de E/S
  - Permiten acelerar de forma global las transferencias

# Control de accesos

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

# Ordenes

nombre [-opciones] [argumentos]

- Nombre

- Cualquier fichero ejecutable que sea accesible
  - nombre absoluto
  - nombre relativo (se busca en PATH)

- Opciones: son modificadores del programa

- casi siempre precedidas por -
- casi siempre acumulativas **-abcd --opcion**
  - **ls -alR .**
  - **ls -lR --all .**
- a veces con argumentos `man -S 1 ls`

- Argumentos

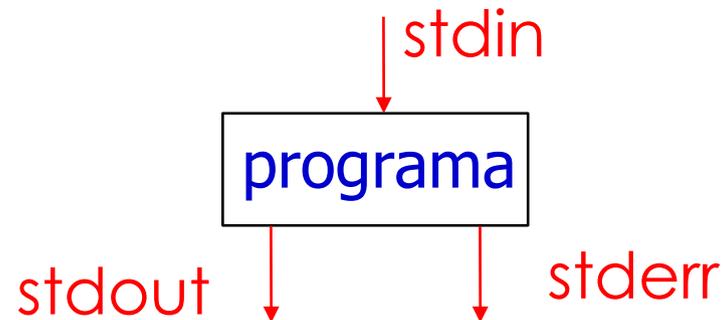
- son los operandos del programa

# Tipos de órdenes

- Informativas
  - man (por secciones), info, netscape, lynx, gv
  - date, cal, ps, pwd
- Ejecutivas
  - De usuario: lpr, cp, mv, rm
  - De sistema: sync, halt, reboot, shutdown
- Procesadoras
  - cmp
  - editores (kedit, kwrite, nedit, pico, vi, emacs)
  - od, strings, head, tail, grep, awk
  - diff
  - find

# Dispositivos estándar

- Disponible para todos los procesos
- Los utilizan los programas como entrada y salida por defecto.
- Dispositivos estándar de los procesos
  - entrada estándar
  - salida estándar
  - salida de error



# Redirección de dispositivos estándar

- **Entrada**

- `sort < fichero`

0 = stdin

1 = stdout

2 = stderr

- **Salida**

- `ls -l > listado`

- `ls -l >> listado`

- `ls kk 2> listado_errores`

- `ls kk . > todo 2>&1`      `-> stderr redir. a stdout`

- **Encadenamiento**

- `du | sort -n`

- **Filtros**

- `cat, more, less, sort, grep`

# Composición de órdenes

- Secuencial: `date ; who`
- Paralelo: `date & who`
- Agrupamiento:
  - `(sleep 5; date) & date`
  - `(date ; who) | tee espia | wc`
  - `(cd / ; ls) | wc -l`
- Continua si no falla:
  - `grep -i amor poema > temp && wc temp`
- Continua si falla
  - `grep -i amor poema > temp || rm temp`

# Control de procesos

- Mostrar los procesos y su estado ps

- Envío de señales:

- Interrupción ^c
- Detención ^z

- Estado del sistema y los procesos (top)

```
7:26pm up 5 days, 17:26, 0 users, load average: 0.16, 0.04, 0.01
38 processes: 37 sleeping, 1 running, 0 zombie, 0 stopped
CPU states: 0.5% user, 1.1% system, 0.0% nice, 98.4% idle
Mem: 30844K av, 29092K used, 1752K free, 17428K shrd, 6760K buff
Swap: 102812K av, 192K used, 102620K free 13584K cached
```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	LIB	%CPU	%MEM	TIME	COMMAND
18876	root	19	0	716	716	552	R	0	1.5	2.3	0:00	top
18856	root	1	0	828	828	504	S	0	0.1	2.6	0:00	sshd
1	root	0	0	340	340	280	S	0	0.0	1.1	0:04	init

# Control de procesos

- **sh : control mínimo**

```
du / | sort -n > uso
```

```
^C
```

```
du / | sort -n > uso &
```

```
ps
```

```
kill -9 12345
```

- **csh/bash : control por lista de trabajos**

- `du / | sort -n > uso`

- `^Z` (se detiene el proceso)

- `jobs` (para ver que procesos hay en la lista)

- `bg` ejecución en segundo término

- `fg` ejecución principal

- referencia a un trabajo `%3`, `bg %3`, `kill -9 %3`

# 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

# Entorno gráfico

- Sistema Xwindow
  - Parte visible (gráfica)
  - Permite ejecutar aplicaciones locales y remotas
  - Sistema cliente-servidor
    - Servidor (local) -> maneja el hardware
    - Cliente (local o remoto) -> aplicación de usuario
  - Permite ejecución multiplataforma
- Gestor de ventanas
  - Programa cliente X
  - Controla las ventanas: posición, tamaño, etc.
- Gestor de escritorio
  - Ambiente de usuario común
  - Proporciona integración de aplicaciones
    - Comunicación entre aplicaciones
    - Copiar y pegar
    - Configuración centralizada .....