UNIX:
l’evoluzione di
un progetto fallito

Il progetto MULTICS

- Multiplexed Information and Computing Service
- Progetto di un sistema operativo moderno (~1965): centinaia di utenti, CPU di potenza paragonabile a 386 (ma ben carrozzato per I/O)
- M.I.T, Bell Labs, General Electrics, ARPA
- Ottime idee, difficoltà di sviluppo (rilascio 1969), ritiro di alcuni partner
- Scarso successo commerciale (80 esemplari) ma alcuni utenti fedelissimi (ultimo shutdown nel 2000) Vedi www.multicians.org

Dolce far niente

- Il ritiro dei Bell Labs dal progetto Multics (1969) lascia alcuni ricercatori senza “nulla da fare”
- Proposta di acquistare hardware per un nuovo progetto di S.O. respinta dal management
- Uno dei ricercatori (Thompson) nel “tempo libero” si scrive un videogioco (Space Travel) che gira su un mainframe (costo partita = 75$ CPU time)
- Per risparmiare, lo porta su un minicomputer (PDP-7) e sulla stessa piattaforma sviluppa un S.O. embrionale (in assembler)

Un insulto

- Il sistema progredisce e viene battezzato EUNUCHS (MULTICS senza gli attributi) e poi ribattezzato UNIX (Uniplexed Information and Computing Service)
- Nel 1970 viene approvato l’acquisto di una nuova macchina (PDP-11) per far girare il promettente sistema
- Alcuni utenti interni apprezzano il sistema come supporto ad attività di editing

Linguaggi di alto livello

- Gli autori sentono il bisogno di un salto di qualità
- Fortran: abbandonato dopo un giorno
- Se ne inventano uno: il B (interpretato, lento)
- Se ne inventano un altro: NB (aggiunta tipi, scrittura compilatore)
- Se ne inventano un altro: C (Ritchie, 1972)
- Riscrivono UNIX in C (Thompson&Ritchie, 1973)
- McIllroy inventa le pipe, Thompson le implementa (malvolentieri in una notte)

Diffusione e portabilità

- UNIX si diffonde nelle università (basso costo PDP e S.O. open source)
- Contributi e aggiunte da tutti (in particolare Berkeley)
- Per portare il codice C su altre architetture serviva scrivere il relativo compilatore C
- Lo sviluppo di un compilatore C a sua volta portabile favorisce la diffusione del sistema
- Nascono dialetti e linee di sviluppo distinte

Le versioni di UNIX: una storia a puntate (1)

- **1971 First Edition** It had a assembler for a PDP-11/20, file system, fork(), roff and ed. It was used for text processing.
- **1973 Fourth Edition** It was rewritten in C. This made it portable and changed the history of OS's.
- **1975 Sixth Edition** UNIX leaves home. Also widely known as Version 6, this is the first to be widely available out side of Bell Labs. The BSD version (1.x) was derived from V6.
- **1979 Seventh Edition** It was a "improvement over all preceding and following Unices" [Bourne]. It had C, UUCP and the Bourne shell. It was ported to the VAX and the kernel was more than 40 Kilobytes (K).

Le versioni di UNIX: una storia a puntate (2)

- **1980 Xenix** Microsoft introduces Xenix. 32V(AT&T per VAX) and 4BSD introduced.
- **1982 System III** AT&T's UNIX System Group (USG) release System III, the first public release outside Bell Laboratories. SunOS 1.0, HP-UX & Ultrix-11 introduced.
- **1983 System V** AT&T announces UNIX System V, the first supported release. Installed base 45,000.
- **1984 4.2BSD** University of California at Berkeley releases 4.2BSD, includes TCP/IP, new signals and much more. X/Open formed.
- **1984 SVR2** System V Release 2 introduced. At this time there are 100,000 UNIX installations around the world.

Le versioni di UNIX: una storia a puntate (3)

- **1986 4.3BSD** 4.3BSD released, including internet name server. SVID introduced (System V Interface Definition: standard di AT&T). AIX announced. Installed base 250,000.
- **1987 SVR3** System V Release 3 including STREAMS, TLI, RFS. At this time there are 750,000 UNIX installations around the world. IRIX introduced.
- **1988** POSIX.1 published. Open Software Foundation (OSF) and UNIX International (UI) formed. Ultrix 4.2 ships.
- **1989** AT&T UNIX Software Operation formed in preparation for spinoff of USL. Motif 1.0 ships.
- **1989 SVR4** UNIX System V Release 4 ships, unifying System V, BSD and Xenix. Installed base 1.2 million.

Le versioni di UNIX: una storia a puntate (4)

- **1990 XPG3** X/Open launches XPG3 (X Portability Guide) Brand. OSF/1 debuts. Plan 9 from Bell Labs ships.
- **1991** UNIX System Laboratories (USL) becomes a company - majority-owned by AT&T. Linus Torvalds commences Linux development. Solaris 1.0 debuts.
- **1992 SVR4.2** USL releases UNIX System V Release 4.2 (Destiny). October - XPG4 Brand launched by X/Open. Solaris 2.0 ships.
- **1993 4.4BSD** 4.4BSD the final release from Berkeley. June 16 Novell acquires USL

Le versioni di UNIX: una storia a puntate (5)

- **Late 1993 SVR4.2MP** Novell transfers rights to the "UNIX" trademark and the Single UNIX Specification to X/Open. COSE initiative delivers "Spec 1170" to X/Open for fasttrack. In December Novell ships SVR4.2MP, the final USL OEM release of System V
- **1994 Single UNIX Specification** BSD 4.4-Lite eliminated all code claimed to infringe on USL/Novell. As the new owner of the UNIX trademark, X/Open introduces the Single UNIX Specification (formerly Spec 1170), separating the UNIX trademark from any actual code stream.

Le versioni di UNIX: una storia a puntate (6)

- **1995 UNIX 95** X/Open introduces the UNIX 95 branding programme for implementations of the Single UNIX Specification. Novell sells UnixWare business line to SCO. Digital UNIX introduced. UnixWare 2.0 ships. OpenServer 5.0 debuts.
- **1996** The Open Group forms as a merger of OSF and X/Open.
- **1997 Single UNIX Specification, Version 2** The Open Group introduces Version 2 of the Single UNIX Specification, including support for realtime, threads and 64-bit and larger processors. The specification is made freely available on the web. IRIX 6.4, AIX 4.3 and HP-UX 11 ship.

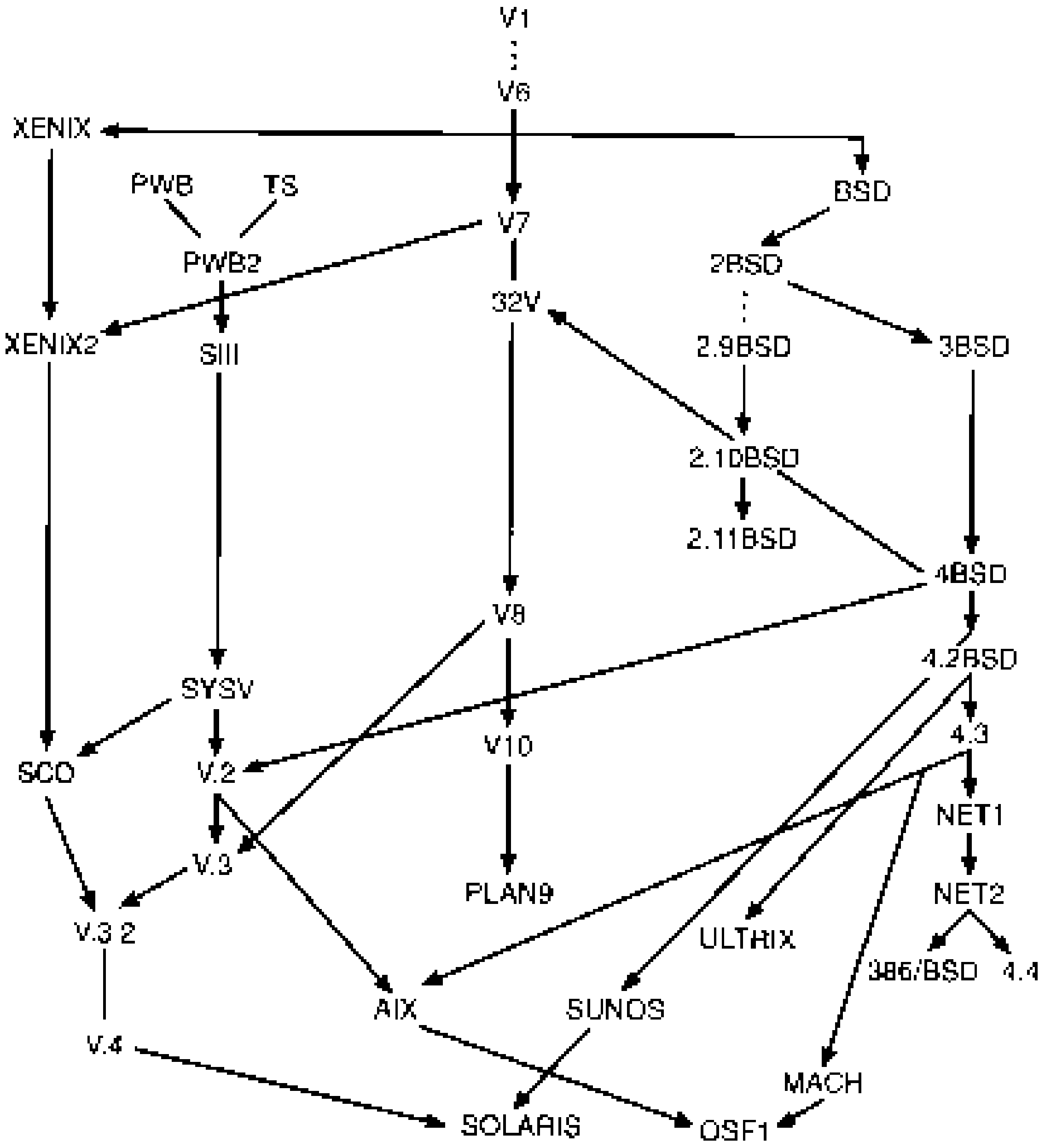
Le versioni di UNIX: una storia a puntate (7)

- **1998 UNIX 98** The Open Group introduces the UNIX 98 family of brands, including Base, Workstation and Server. First UNIX 98 registered products shipped by Sun, IBM and NCR. The Open Source movement starts to take off with announcements from Netscape and IBM. UnixWare 7 and IRIX 6.5 ship.
- **1999 UNIX at 30** The UNIX system reaches its 30th anniversary. Linux 2.2 kernel released. The Open Group and the IEEE commence joint development of a revision to POSIX and the Single UNIX Specification. First LinuxWorld conferences. Dot com fever on the stock markets. Tru64 UNIX ships.

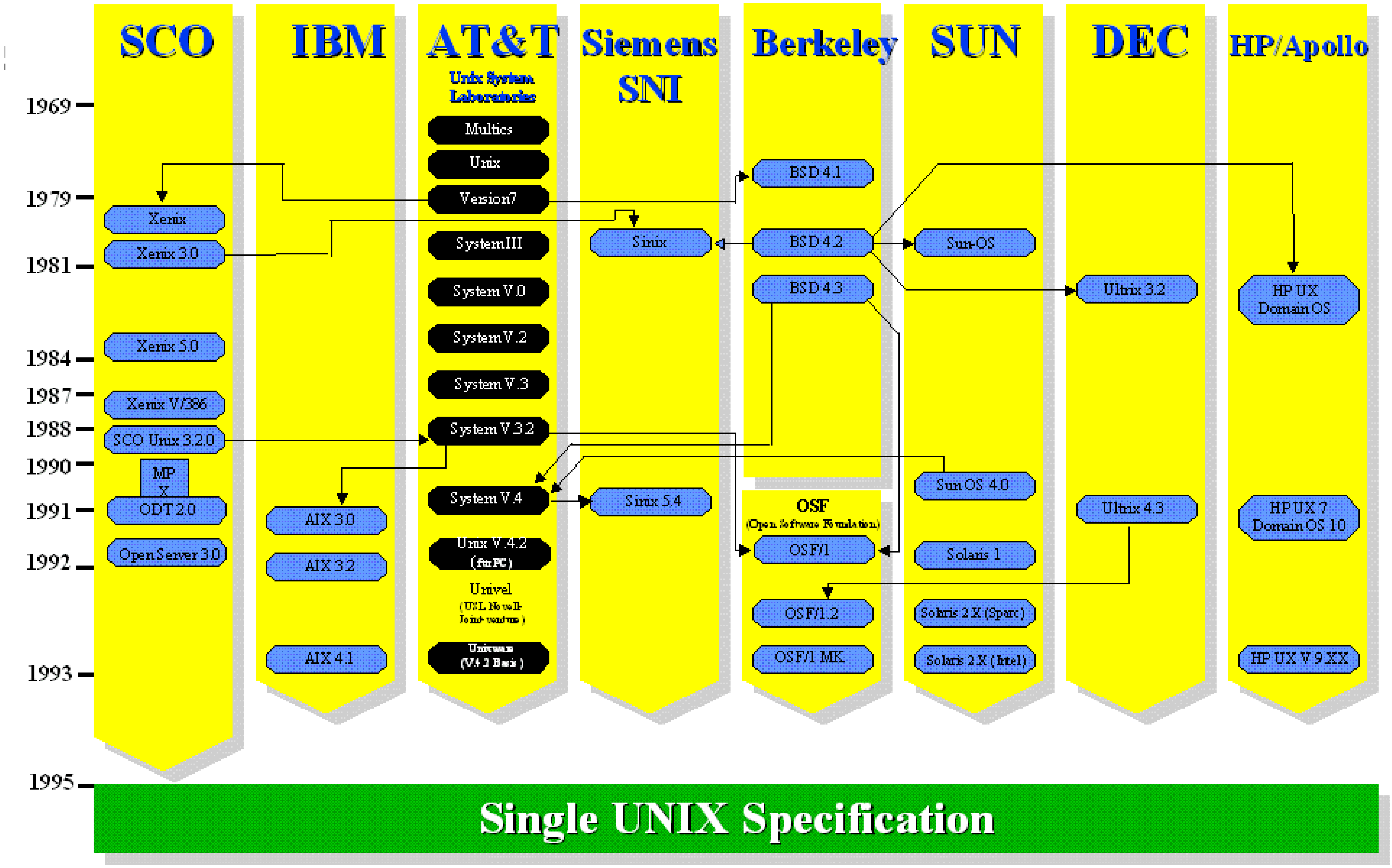
Le versioni di UNIX: una storia a puntate (8)

- **2001 Single UNIX Specification, Version 3** Version 3 of the Single UNIX Specification unites IEEE POSIX, The Open Group and the industry efforts. Linux 2.4 kernel released. IT stocks face a hard time at the markets. The value of procurements for the UNIX brand exceeds $25 billion. AIX 5L ships.
- **2003 ISO/IEC 9945:2003** The core volumes of Version 3 of the Single UNIX Specification are approved as an international standard. The "Westwood" test suite ship for the UNIX 03 brand. Solaris 9.0 E ships. Linux 2.6 kernel released.

UNIX VERSIONS TREE



UNIX Chronology



Iniziative di standardizzazione

- Ce ne sono state varie (“UNIX wars” tra consorzi).
- I residui del conflitto si trovano un po’ sparsi ovunque (differenza di ogni tipo)
- Lo standard più noto è POSIX (varie versioni)
- Iniziativa di convergenza (Single UNIX) http://www.unix.org/single_unix_specification/
- Non tutti si preoccupano della completa rispondenza allo/agli standard: Digital UNIX, HP-UX,AIX,SCO UnixWare,SGI IRIX,Sun Solaris sono certificati, Linux NO ! (ma è uno UNIX di fatto)

Cosa è standardizzato

- System interface (procedure di libreria): 1123 elencate nel documento del 2003
  - » non tutte obbligatorie, non tutte system call
  - » includono pthread e socket
- Utilities interface (comandi utente): 160 elencati
- Headers interface (file .h): 84 elencati
- XCURSES interface (procedure gestione terminali a caratteri): 372 elencate
- Portable character set, regular expressions e altre cose implicate dalle definizioni

Cosa è poco standardizzato

- L'interprete dei comandi (shell) è standardizzato nella versione base sh (28 pagine di specifica !)
- In pratica ci sono e si usano shell più sofisticate (bash,csh,tcsh) ognuna con le sue particolarità
- Quasi nulla sulla gerarchia di directory: deve esserci / e si dice qualcosa su /dev e /tmp
- Iniziativa parallela sulla gerarchia di directory: FHS (File system Hierarchy Standard) www.pathname.com/fhs/

FHS: directory nella root

bin	Essential command binaries
boot	Static files of the boot loader
dev	Device files
etc	Host-specific system configuration
lib	Essential shared libraries and kernel modules
*media**	Mount point for removeable media
mnt	Mount point for mounting a filesystem temporarily
opt	Add-on application software packages
sbin	Essential system binaries
*srv**	Data for services provided by this system
tmp	Temporary files
usr	Secondary hierarchy
var	Variable data
home	User home directories
root	Home directory for the root user

FHS: /bin e /etc

- /bin contains commands that may be used by both the system administrator and by users, but which are required when no other filesystems are mounted (e.g. in single user mode). I comandi sono 30 + 9 opzionali
- /etc contiene file di configurazione in parte direttamente in parte in sottodirectories. 31 file specificamente elencati
- /usr is the second major section of the filesystem. /usr is shareable, read-only data. Any information that is host-specific or varies with time is stored elsewhere.

Alcuni file di configurazione in /etc

fstab	Static information about filesystems
group	User group file
hosts	Static information about host names
inittab	Configuration file for init
issue	Pre-login message and identification file
ld.so.conf	List of extra directories to search for shared libraries
motd	Post-login message of the day file
mtab	Dynamic information about filesystems
passwd	The password file
profile	Systemwide initialization file for sh shell logins
protocols	IP protocol listing
resolv.conf	Resolver configuration file
services	Port names for network services
shells	Pathnames of valid login shells
syslog.conf	Configuration file for syslogd

FHS: /usr

- /usr is the second major section of the filesystem. /usr is shareable, read-only data. Any information that is host-specific or varies with time is stored elsewhere.
- The /usr/local hierarchy is for use by the system administrator when installing software locally. It needs to be safe from being overwritten when the system software is updated. It may be used for programs and data that are shareable amongst a group of hosts, but not found in /usr.
- The /usr/share hierarchy is for all read-only architecture independent data files.

Sottodirectory di /usr

bin	Most user commands
include	Header files included by C programs
lib	Libraries
local	Local hierarchy (empty after main installation)
sbin	Non-vital system binaries

share	Architecture-independent data *X11R6* XWindow System (optional)
games	Games and educational binaries (optional)
lib<qual>	Alternate Format Libraries (optional)
src	Source code (optional)

The following symbolic links to directories may be present. This possibility is based on the need to preserve compatibility with older systems until all implementations can be assumed to use the /var hierarchy.

```
/usr/spool -> /var/spool
/usr/tmp -> /var/tmp
/usr/spool/locks -> /var/lock
```

Sottosottodirectory di /usr

Sotto /usr/local

bin	Local binaries
etc	Host-specific system configuration
games	Local game binaries
include	Local C header files
lib	Local libraries
man	Local online manuals
sbin	Local system binaries
share	Local architecture-independent hierarchy
src	Local source code

Sotto /usr/share

man	Online manuals
misc	Miscellaneous architecture-independent data
dict	Word lists (optional)
doc	Miscellaneous documentation
info	GNU Info system’s primary directory (optional)
locale	Locale information
terminfo	Directories for terminfo database (optional)
zoneinfo	Timezone information and configuration (optional)

Eseguibili sparsi

- Utilities used for system administration (and other root-only commands) are stored in /sbin, /usr/sbin, and /usr/local/sbin. /sbin contains binaries essential for booting, restoring, recovering, and/or repairing the system in addition to the binaries in /bin. [18] Programs executed after /usr is known to be mounted (when there are no problems) are generally placed into /usr/sbin. Locally-installed system administration programs should be placed into /usr/local/sbin.

FHS: /var

- /var contains variable data files. This includes spool directories and files, administrative and logging data, and transient and temporary files.
- If /var cannot be made a separate partition, it is often preferable to move /var out of the root partition and into the /usr partition.

Sottodirectory di /var

cache	Application cache data
lib	Variable state information
local	Variable data for /usr/local
lock	Lock files
log	Log files and directories
opt	Variable data for /opt
run	Data relevant to running processes
spool	Application spool data
tmp	Temporary files preserved between system reboots

account	Process accounting logs (optional)
crash	System crash dumps (optional)
games	Variable game data (optional)
mail	User mailbox files (optional)
yp	Network Information Service (NIS) database files (optional)

/proc: una chicca di Linux

- The /proc filesystem is the de-facto standard Linux method for handling process and system information. We strongly encourage this for the storage and retrieval of process information as well as other kernel and memory information.
- Non è un FS vero, ma virtuale il cui contenuto viene creato e aggiornato in esecuzione (eventualmente solo a richiesta)
- /proc contiene direttamente dei file di testo (una trentina) p.e. *cpuinfo, meminfo, uptime, devices, interrupts, partitions*
- I file in proc sono di norma read-only ma alcuni (in proc/sys) si possono sovrascrivere per riconfigurare il sistema
- /proc contiene poi varie sottodirectory

Sottodirectory di /proc

- Una directory per processo con nome (un numero) uguale al PID con vari file (*cmdline, cpu, status*)
- Altre dir con nomi autoesplicativi

bus	Info sui bus (usb,pci,isa,...)	*sys*	Varie parti del sistema
driver	Drivers in uso dal kernel	*sys/dev*	Dispositivi vari
fs	File system esportati NFS	*sys/fs*	File system
ide	Dispositivi IDE	*sys/kernel*	Funzionamento kernel
irq	Interrupt	*sys/net*	Rete (configurazione)
net	Rete	*sys/vm*	Virtual memory
sysvipc	Interprocess comunication		
tty	Terminali a caratteri		

Come si riconosce uno UNIX (o un'imitazione/emulazione)

- Dal certificato di conformità allo standard
- Dal nome che finisce per X
- Per l'utente: da alcuni tipici comandi di shell (p.e. ls, cp, cat, echo, man ...)
  da alcune directory tipiche nel file system e relativi file contenuti (p.e. /dev, /usr, /etc)
- Per il programmatore: dalla disponibilità di una libreria C con tipiche funzioni (p.e. fork, syslog, socket)
- Dall'esistenza di un superutente denominato root

Filosofia UNIX

- Nato da programmatori per programmatori
- Nato per l’interazione a linea di comando
- Piccolo è bello: programmi che fanno bene una sola cosa, facili da capire, debuggare, mantenere
- Script per funzionalità evolute: i comandi utente nascono per essere usati anche e soprattutto in script di shell e in pipe con altri programmi
- In molti casi la portabilità è più importante dell’efficienza
- Usa codice altrui e lascia usare il tuo