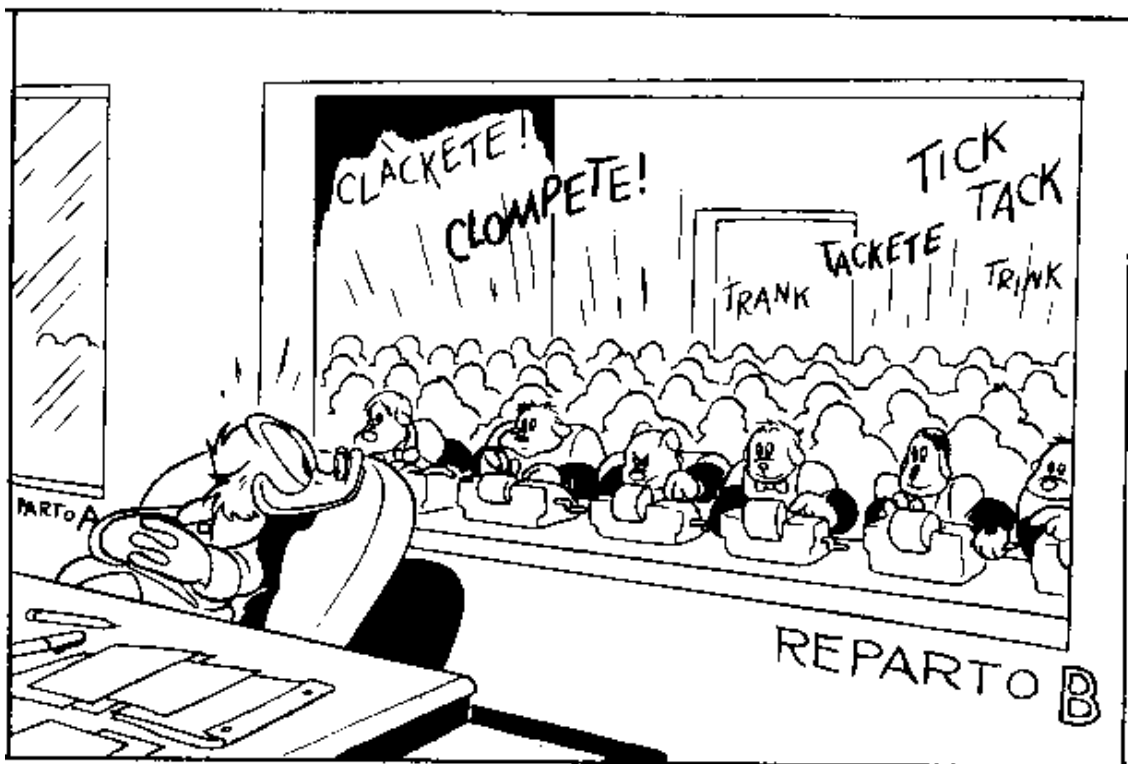


Unix: Una cultura di condivisione

alessandro.rubini@linux.it
rubini@gnu.org

Illustrazione di Romano Scarpa



Il postscript di questi lucidi e' su
<ftp://ftp.systemy.it/pub/develop/trento.ps.gz>

--
__ o Tu vuo' fa' l'americano, sient'a me chi t'o fa fa'
_ '\<, (Renato Carosone)
()/() alessandro.rubini@linux.it +39 382 529554

Cos'è Unix

**Unix è un progetto del 1969,
nasce dall'esperienza di MULTICS,
nasce per gioco**

Il sistema Unics (uniprogrammed computer system) è stato sviluppato nei laboratori AT&T ed è stato distribuito gratuitamente alle università (dal 1973)

Unix è scritto in C, un linguaggio scritto appositamente per la programmazione di sistema, "C" è il successore di "B", che è una semplificazione di BCPL.

*Naturalmente il compilatore C è scritto in C
(anche il compilatore Java è scritto in Java
e il compilatore elisp è scritto in elisp)*

Unix è un sistema estremamente leggero, sviluppato inizialmente su un sistema con 16kB di memoria

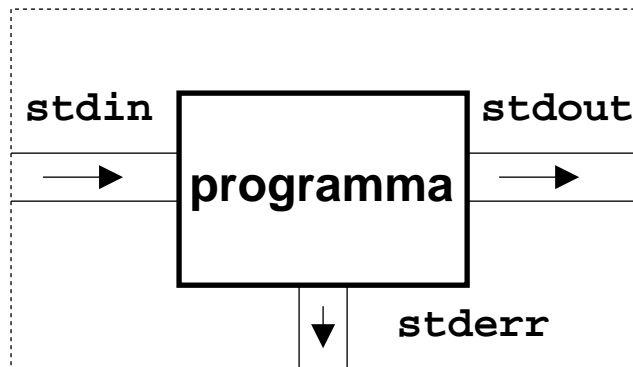
L'editor di testi "vi" è stato scritto nel 1977

All'inizio degli anni '80 TCP/IP è stato sviluppato sotto Unix

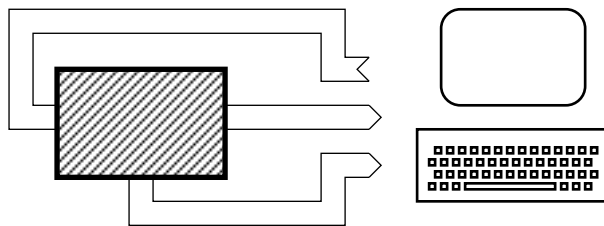
Lecture consigliate:

Twenty Years of Berkeley Unix (in "Open Sources")
Hackers, di Steven Levy

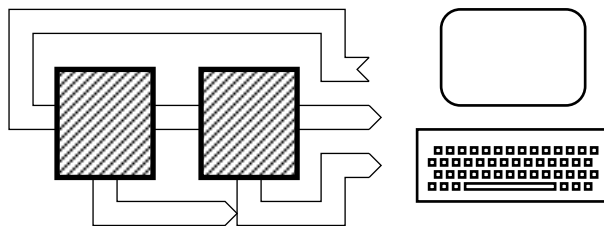
L'astrazione "file"



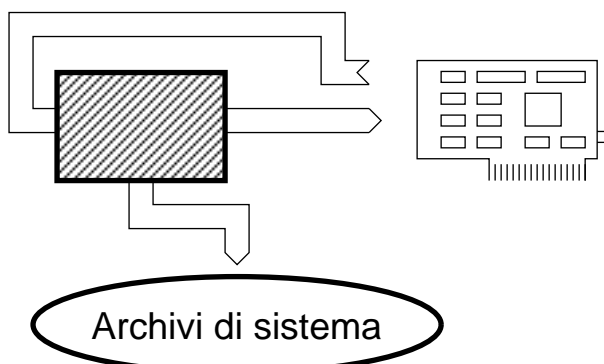
**Un programma e' sempre
connesso con almeno tre
"file"**



Operazione da terminale



"pipe" (tubo)

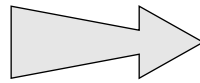


**Operazione tramite
connessione di rete**

Il filesystem

Tutto in Unix si appoggia sul filesystem

- File regolari
- Periferiche
- Socket di rete
- Nodi "FIFO"



Tutte le operazioni si basano su `open()`, `read()`, ...

La directory altro non e' che un tipo di file

```
prw-rw-r--  1 root    root           0 Nov 15 19:15 /dev/gpmdata
crw-rw-rw-  1 root    sys            1,  3 Nov 30 1993 /dev/null
srw-----  1 root    root           0 May  6 07:56 /dev/printer
crw-rw-rw-  1 root    tty            5,  0 Nov  2 1994 /dev/tty
crw-rw-rw-  1 root    sys            1,  5 Nov 30 1993 /dev/zero
-rw-r--r--  1 root    root          1294 Oct 23 1996 /etc/passwd
```

Le metainformazioni relative ad un file sono raccolte nel suo "inode"

Per comodita', i file che iniziano per punto sono "nascosti"

Per accedere ad un file con nomi diversi, esistono i link simbolici e gli "hard link"

```
morgana% ls -ld `which gzip` kmouse
-rwxr-xr-x  3 root    root           45056 Oct 21 1996 /bin/gzip*
lrwxrwxrwx  1 rubini  staff           11 Jan 30 11:59 kmouse -> kmouse-0.49/
```

I processi

Un processo e' una struttura dati rappresentante un flusso di calcolo

Il processo "init" viene fatto partire automaticamente tutto il resto dipende dalla configurazione del sistema

/etc/inittab

/etc/rc.d

Il comando "ps" riporta la situazione attuale del sistema

```
morgana% ps -aux
USER      PID  %CPU  %MEM  SIZE  RSS  TTY  STAT  START  TIME  COMMAND
root       1   0.0   0.7   868   240  ?    S     07:56  0:03  init auto
root       2   0.0   0.0     0     0  ?    SW    07:56  0:00  (kflushd)
root       3   0.0   0.0     0     0  ?    SW<   07:56  0:00  (kswapd)
root      30   0.0   0.7   868   220  ?    S     07:56  0:00  /usr/sbin/crond -l10
root      73   0.0   1.4  1072   460  ?    S     07:56  0:00  sendmail: accepting c
root      95   0.0   1.0   868   312  3    S     07:56  0:00  /sbin/agetty 38400 tt
root     291  11.6  11.6  5840  3568  ?    S     08:35  2:49  X :0
root     312   0.0   3.5  2168  1080  2    S     08:35  0:00  xterm -bw 2 -geometry
root     313   0.0   3.1  2088   976  2    S     08:35  0:00  xterm -bw 4 -geometry
root     366   0.0   3.4  2160  1072  2    S     08:52  0:00  xterm -n sandra -T sa
rubini    94   0.0   2.4  1036   752  2    S     07:56  0:00  /bin/login -- rubini
rubini   148   0.0   2.8  1368   876  2    S     07:58  0:01  -tcsh
rubini   290   0.0   1.0  1372   308  2    S     08:35  0:00  xinit /home/staff/rub
rubini   293   0.0   1.7  1192   544  2    S     08:35  0:00  sh /home/staff/rubini
rubini   300   0.3   3.4  1988  1052  2    S     08:35  0:05  vtwm
rubini   302   0.0   1.6  1556   520  2  S  N    08:35  0:00  xdaliclock -24 -fn -a
rubini   303   0.2   4.1  2312  1264  2  S  N    08:35  0:04  xman -geometry -5+5
rubini   368   0.0   1.2   884   384  p3   S     08:52  0:00  rlogin sandra
rubini   371   0.0   1.3   928   416  p1   R     08:59  0:00  ps -aux
```

I dispositivi convenzionali di I/O sono predisposti per l'uso tramite stdin e stdout

```
morgana% sort data | a2ps | bold | \
rsh prozio "cat > /dev/lp0"
[Total: 3 pages on 2 sheets]
```

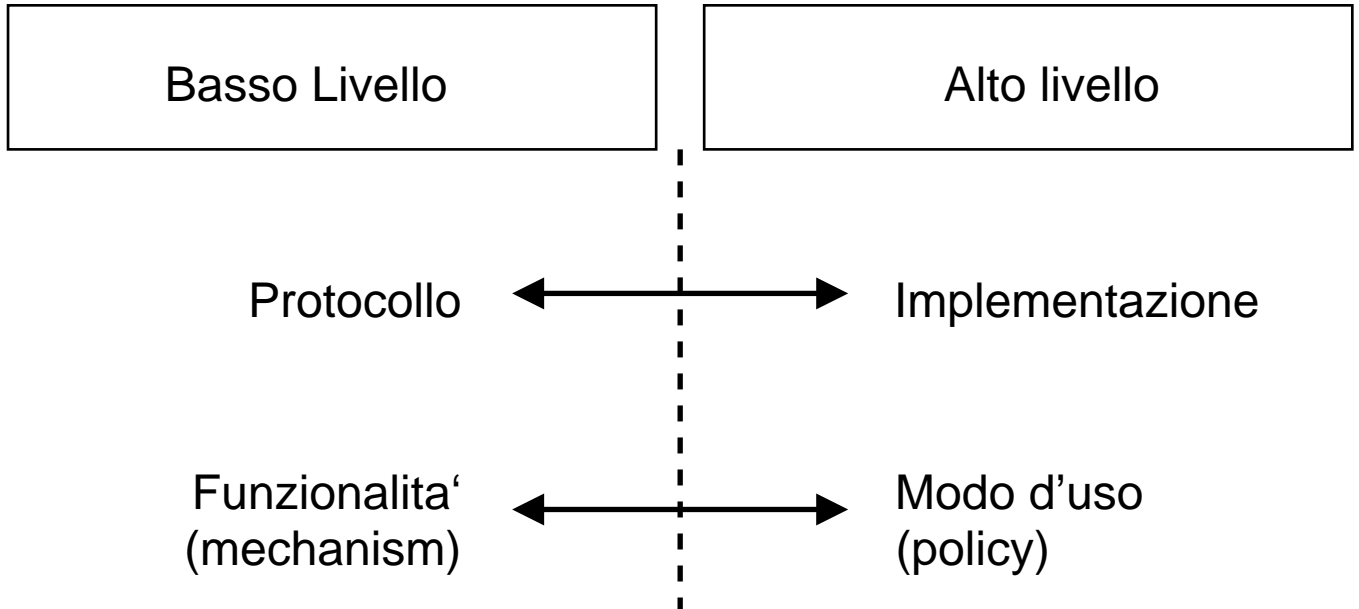
Struttura di rete

Uso massiccio del modello client/server

Interconnessione a livello di applicazione

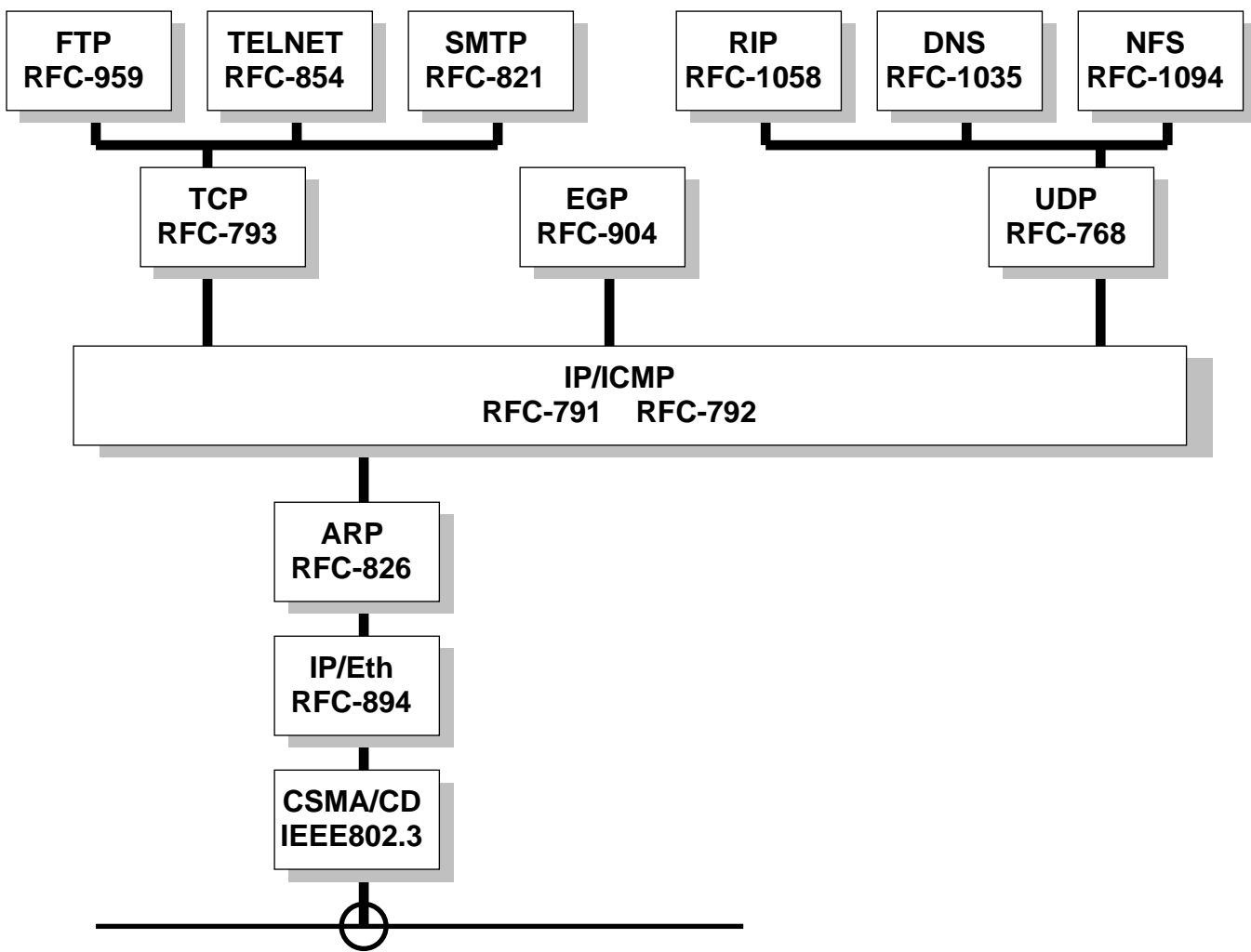


Interconnessione a livello circuitale

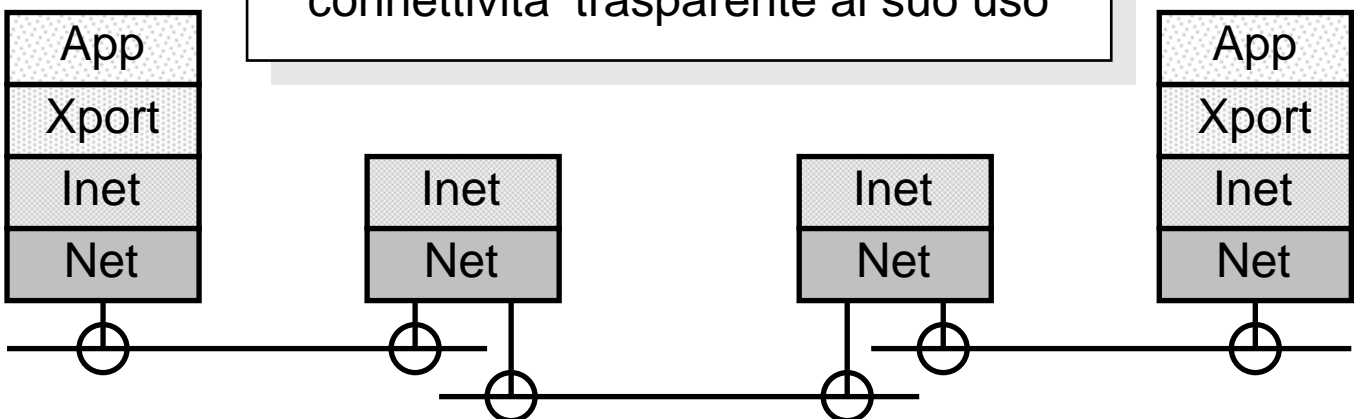


Gli strumenti devono essere modulari e potersi adattare alle esigenze dell'utente

Una rete di protocolli



Una buona divisione tra i livelli e' necessaria per fornire una connettivita' trasparente al suo uso

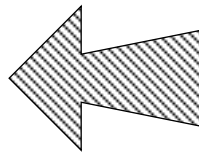


Esempi di server

Un server internet puo' girare sotto "inetd"
ed ignorare le problematiche di rete

```
#!/bin/sh

while true; do
  read a
  su nobody -c $a
done
```



Questo programma e'
un server completo.

Altri esempi

```
if [llength $argv]!=2 {
  puts stderr "$argv0: Use: \"$argv0 <lname>
<remote IP:port>\"
  exit 1
}
set host [lindex $argv 0]
set dest [lindex $argv 1]
if [scan $dest "%i.%i.%i.%i:%i%s" a b c d e s]!=5 {
  puts stderr "$argv0: bad destination \"$dest\"
  exit 1
}
set sock [udp open client $dest]

# just read /proc/kmsg forever
set F [open /proc/kmsg]
while 1 {
  gets $F string
  udp sendto $sock [list $host $string]
}
```

```
if [llength $argv]!=1 {
  puts stderr "$argv0: Use: \"$argv0 <udp-port>\"
  exit 1
}
set sock [udp open $argv]

while 1 {
  set string [udp recvfrom $sock from]
  puts "$from $string"
}
```


L'ambiente grafico

Il sorgente del server X dice:

This is the X Consortium implementation of the X Window System. X is a vendor-neutral, system-architecture neutral network-transparent window system and user interface standard. X runs on a wide range of computing and graphics machines. For an overview of X, see the X manual page.

Ancora una volta, si e' cercato di dividere il problema della gestione di un ambiente grafico in tanti pezzi semplici:

X server

Il server si occupa della gestione dell'hardware. Esiste un diverso server per ogni architettura supportata. Il server disegna le finestre su indicazione delle applicazioni

window manager

Gestisce le varie finestre nella loro interazione con l'utente. Ogni window manager "civile" e' completamente riconfigurabile, e non ha nessuna relazione con l'hardware grafico

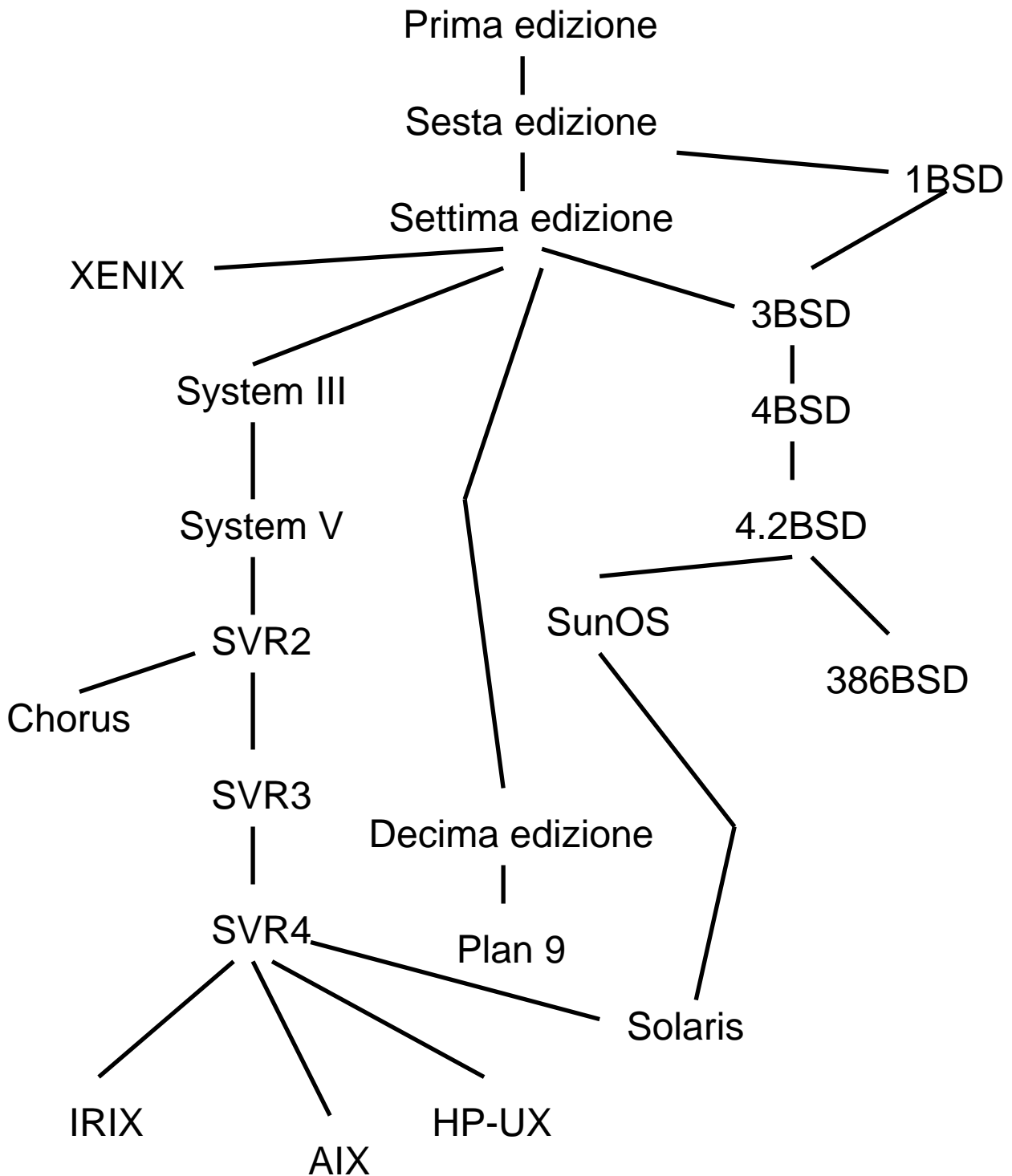
Xlib

La libreria X e' un insieme di funzioni C che permettono alle applicazioni di parlare in maniera "quasi comprensibile" con il server. Il protocollo di basso livello viene cosi' nascosto alle applicazioni.

Le ramificazioni

La distribuzione di Unix in forma sorgente ha permesso la ramificazione

Le licenze d'uso adottate hanno impedito la completa riunificazione



GNU Non e' Unix



GNU's not Unix: we've higher standards

L'uso del termine "hacker" nel senso di "pirata" è una confusione di termini creata dai mezzi di informazione. Noi hacker ci rifiutiamo di riconoscere questo significato, e continuiamo ad utilizzare la parola nel senso di "uno che ami programmare, e a cui piaccia essere bravo a farlo"

La scelta di Unix come ambiente di sviluppo di GNU e' motivata principalmente dalla modularita' e portabilita' del sistema.

Non chiamavamo il nostro software "software libero", poiché questa espressione ancora non esisteva, ma si trattava proprio di questo.

La situazione cambiò drasticamente all'inizio degli anni '80 [...] I moderni elaboratori di quell'epoca, come il VAX o il 68020, avevano il proprio sistema operativo, ma nessuno di questi era libero: si doveva firmare un accordo di non-diffusione persino per ottenerne una copia eseguibile.

Questo significava che il primo passo per usare un computer era promettere di negare aiuto al proprio vicino. Una comunità cooperante era vietata. La regola creata dai proprietari di software proprietario era: «se condividi il software col tuo vicino sei un pirata. Se vuoi modifiche, pregaci di farle».

L'idea che la concezione sociale di software proprietario -- cioè il sistema che impone che il software non possa essere condiviso o modificato -- sia antisociale, contraria all'etica, semplicemente sbagliata, può apparire sorprendente a qualche lettore. Ma che altro possiamo dire di un sistema che si basa sul dividere utenti e lasciarli senza aiuto?

Allora cercai un modo in cui un programmatore potesse fare qualcosa di buono. Mi chiesi dunque: c'erano un programma o dei programmi che io potessi scrivere, per rendere nuovamente possibile l'esistenza di una comunità?

La risposta era semplice: innanzitutto serviva un sistema operativo. [...] Essendo un programmatore di sistemi, possedevo le competenze adeguate per questo lavoro. [...] Scelsi di rendere il sistema compatibile con Unix, in modo che fosse portabile, e che gli utenti Unix potessero passare facilmente ad esso.

Linux non e' Minix

**Linux e' un progetto del 1991,
nasce dall'esperienza di Minix,
nasce per gioco**



"Make Minix freely available and one of my biggest gripes with it will disappear"

"I got my 386 last January, and Linux was partly a project to teach me about it"

If you write programs for Linux today you shouldn't have too many surprises when you recompile them for Hurd in the 21st century. As has been noted (not only by me), the linux kernel is a minuscule part of a complete system: full sources for Linux currently runs to about 200kB compressed - full sources to a somewhat complete development system is at least 10MB compressed (and easily much, much more). And all of that source is portable, except for this tiny kernel that you can (probably: I did it) re-write totally from scratch in less than a year without having any prior knowledge.

Quando ho scritto Linux, conoscevo Unix da sei mesi e ho pensato di poterlo riscrivere da zero.

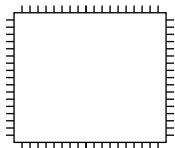
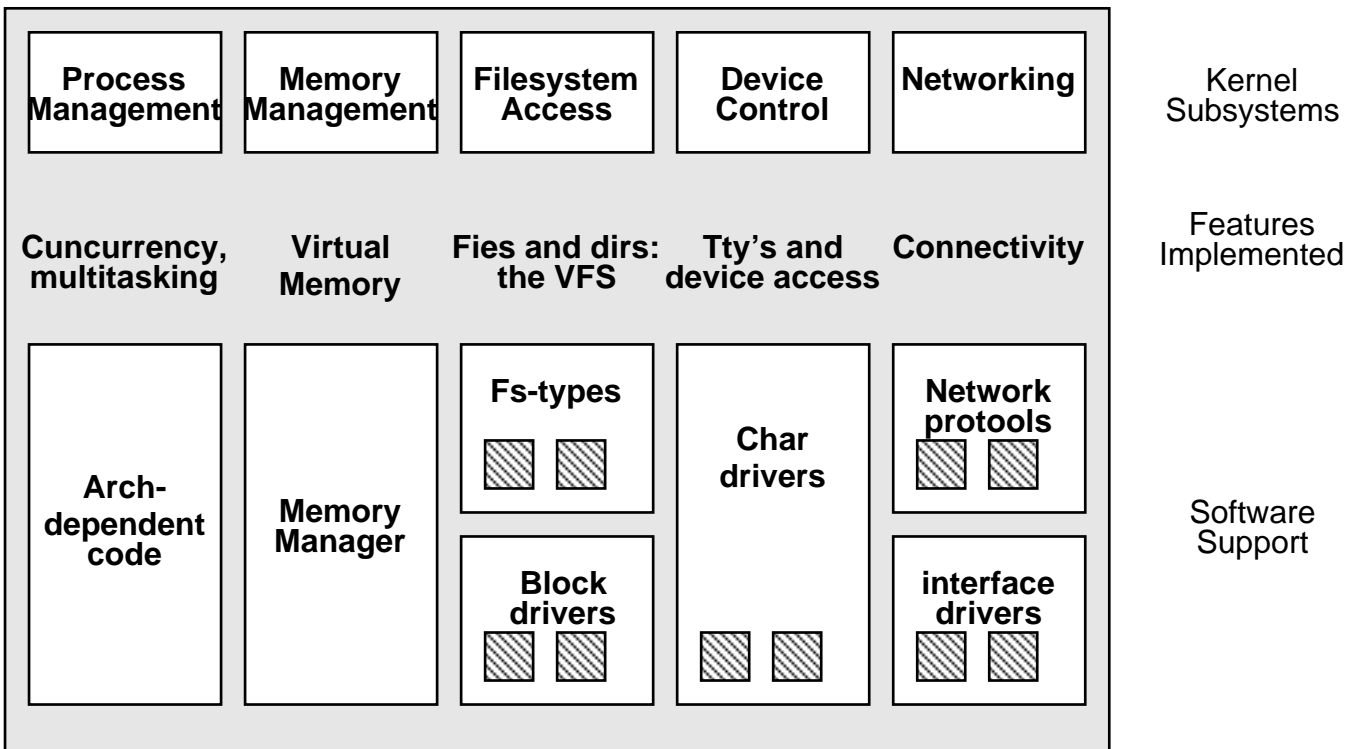
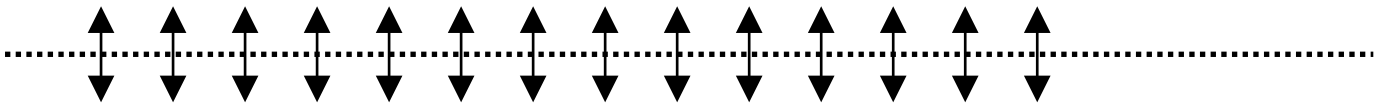
Se lo avessi conosciuto meglio, non avrei iniziato.

Cos'e' "Linux"?

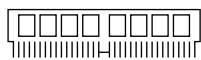
Processes:

- Numerical Analysys
- Database Administration
- Network Services
- Graphical Display
- User Interfaces
- Program Development

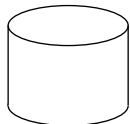
The System Call Interface



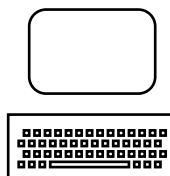
CPU



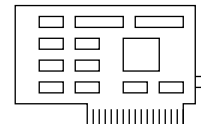
Memory



Disks and CD's



Console etc.



Network Interfaces

Hardware

E cos'è il Software?

Per parlare di “free software” occorre prima analizzare cosa sia il software

- **Un programma è informazione pura**
- **Un programma è un prodotto culturale**
- **Un programma è un prodotto scientifico**
- **Un programma viene copiato a costo zero**

La “licenza d'uso” del software viene imposta da chi ne detiene il copyright, cioè la proprietà intellettuale

Ci sono due diverse modalità di agire per chi scrive software:

- **Cedere la proprietà al datore di lavoro o al cliente**
- **Mantenere la proprietà e dare il programma in licenza**

Questa seconda ipotesi risulta decisamente più interessante.....

La questione è simile a quanto succede nell'editoria e nella fotografia

Anche in questi casi, infatti, occorre definire chiaramente cosa è lecito e non è lecito fare. Anche in questi casi ci sono molte questioni poco chiare.

Software libero

Esistono differenti licenze libere

Licenza BSD

Tutto il sw BSD
Tcl/Tk
Sendmail

I lavori derivati
devono
mantenere le
note originali nel
sorgente

Licenza MIT

L'ambiente
grafico X

I lavori deprivati
devono
mantenere la
nota di copyright
originale

Gnu GPL

Tutto il sw GNU
Il kernel di Linux
Perl

Artistic Licence

Perl

NOTE! This copyright does **not** cover user programs that use kernel services by normal system calls - this is merely considered normal use of the kernel, and does **not** fall under the heading of "derived work". Also note that the GPL below is copyrighted by the Free Software Foundation, but the instance of code that it refers to (the linux kernel) is copyrighted by me and others who actually wrote it.

Linus Torvalds

Free beer
Free speech

E' un peccato che una lingua cosi' importante manchi di una parola cosi' importante

Richard Stallman

La licenza pubblica GNU, a differenza delle altre, e' volta a garantire la liberta' di condividere e modificare il software. Vuole assicurarsi che il programmi rimangano liberi per tutti gli utenti.

La licenza GPL

Quando si parla di free software, ci si riferisce alla liberta', non al prezzo. Le nostre Licenze (la GPL e la LGPL) sono progettate per assicurare che ciascuno abbia la liberta' di distribuire copie del software libero (e farsi pagare per questo, se vuole), che ciascuno riceva il codice sorgente o che lo possa ottenere se lo desidera, che ciascuno possa modificare il programma o usarne delle parti in nuovi programmi liberi e che ciascuno sappia di potere fare queste cose.

E' lecito copiare e distribuire il Programma [...] a patto che [...] il Programma sia corredato dal codice sorgente completo, in una forma leggibile dal calcolatore [...]

Per "codice sorgente completo" di un lavoro si intende la forma preferenziale usata per modificare un lavoro.

Per un programma eseguibile, "codice sorgente completo" significa tutto il codice sorgente di tutti i moduli in esso contenuti, piu' ogni file associato che definisca le interfacce esterne del programma, piu' gli script usati per controllare la compilazione e l'installazione dell'eseguibile.

L'ambiguita' del termine "free" e l'importanza del libero accesso al codice sorgente hanno portato alla definizione del concetto di "Open Source", e alla elezione di tale termine a marchio registrato.

La dizione "Open Source" e' meglio accettata a livello commerciale.

Esiste un documento chiamato "Open Source Definition" che specifica chiaramente cosa sia open-source e cosa non lo sia

La licenza GPL e' conforme alla definizione di open source.

Si vedano a questo proposito:

www.fsf.org

www.opensource.org

www.debian.org

<http://www.linux.it/GNU>

Le distribuzioni

Molte case predispongono il kernel Linux e tutti gli applicativi perche' possano essere usati facilmente

Ne' la GPL ne' le altre licenze impediscono la vendita del software
Il prezzo commerciale risulta appropriato al valore del prodotto

Slackware

RedHat

Debian

Caldera

Ogni distribuzione e' leggermente diversa dalle altre, ma le differenze sono marginali e spesso si limitano agli strumenti per l'amministrazione di sistema

Suse

Poiché "free" si riferisce alla libertà e non al prezzo, vendere copie di un programma non contraddice il concetto di software libero. In effetti, la libertà di vendere copie di programmi è essenziale: raccolte di software libero vendute su CD-ROM sono importanti per la comunità, e la loro vendita è un modo di raccogliere fondi importante per lo sviluppo del software libero. Di conseguenza, un programma che non può essere liberamente incluso in tali raccolte non è software libero.

Commercio di sw

Copie vendute

Quante si riesce a piazzarne

Una sola, o poco piu'

Commercializzazione del prodotto

Publicita' e rete di distribuzione

Attraverso il contatto personale

Personalizzazione ulteriore

L'acquirente dipende dal
programmatore originale

Qualunque programmatore
puo' continuare il lavoro

Contratti di assistenza

Senza un contratto e'
impossibile vendere il prodotto

Il programmatore puo' fare il
contratto o non farlo

Il lavoro di programmazione

Deve essere tutto personale

Puo' attingere dal lavoro di altri

L'acquisizione di conoscenze

Occorre firmare NDA e
comprare manuali e sorgenti

E' tutto disponibile in rete, e
basta usare grep e less

Cosa dice l'utente

Assistenza Tecnica

Vincolata al fornitore iniziale

Puo' essere fatta da altri tecnici

Costo del prodotto

Calato dall'alto

Commisurato al lavoro fatto

Installazioni Multiple

Non permesse, a meno di non
moltiplicare l'acquisto

Permesse e incoraggiate.
E' permessa anche la rivendita

Possibilita' di esclusiva

Solo se si acquisisce la
proprietà del sorgente

Contrattabile con l'autore in
base alle esigenze

Piattaforme Hardware

Solo recenti e di un tipo
(non sempre)

Anche vecchie o diverse
(non sempre)

Formato dei dati

Formati proprietari, non
utilizzabili da altri programmi
(non sempre)

Adesione agli standard e uso
di formati documentati

Altre questioni

Il mercato attuale dell'informatica

- E' teatro di inutile consumismo
- Richiede grossi investimenti anche per svolgere piccoli compiti
- Vede una grossa differenza di ruoli tra le varie nazioni
- E' dominato dai colossi del settore, lasciando poco spazio ai piccoli programmatori

Alcuni possibili cambiamenti

- Limitare lo spreco di macchinari e il conseguente inquinamento permettendo l'uso di macchine obsolete per compiti poco gravosi (la maggior parte dei compiti, in effetti, sono poco gravosi se affrontati bene)
- Limitare la "fuga di cervelli" dai paesi meno sviluppati. grazie alla maggiore connettivita' e alla compatibilita' tra i vari ambienti di lavoro.
- Aprire il mercato anche all'iniziativa individuale, sfruttando la competizione "meritocratica" tra sistemi aperti.
- Diffondere la cultura informatica in maniera piu' capillare, per fornire un'assistenza agli utenti piu' diffusa sul territorio.
- Dirottare i capitali adesso investiti in licenze e macchinari verso le persone che offrono consulenza diretta ai clienti