

# Il software libero

Alberto Cammozzo

[mmzz@pluto.it](mailto:mmzz@pluto.it)

Admin, progetto ReFun, Pluto

System manager, Università di Padova

presentazione per gli studenti del

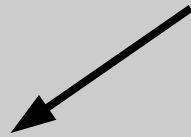
Corso di Laurea in Informatica,  
Facoltà di Scienze Matematiche, Fisiche e Naturali  
Università di Padova

8 Marzo 2004

# Sommario

- Introduzione: libero/ proprietario
- Storia e storie del software
- Libertà e prezzo
- Modelli di diffusione del software
- Modelli di produzione del software
- Modelli di licenze: GPL e le altre
- Aspetti socio-politici

# Codice sorgente (*source*)



- Chiuso o proprietario:
  - Il programma è:
    - × eseguibile a certe condizioni
    - × sorgente non leggibile
    - × non modificabile
- Libero o *free*\* o *open*:
  - Il programma è:
    - ✓ eseguibile liberamente
    - ✓ sorgente leggibile
    - ✓ modificabile

\* *free* = libero, non per forza gratis

# C'era una volta un programma...

- Sui primi computer:
  - software distribuito in sorgente
  - condivisione del nuovo software scritto
  - comunità di utenti/ sviluppatori
- Con la diffusione dei minicomputer e dei PC
  - software venduto a parte (UNIX, DOS, CP/ M,...)
  - solo eseguibile per evitare concorrenza
  - utenti isolati

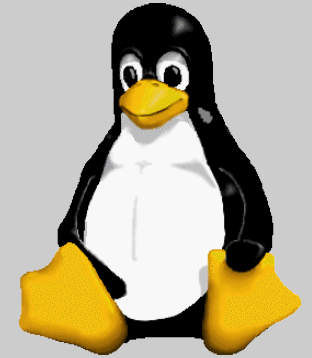
...adesso si chiama **applicativo.**

# Prima storia: GNU (*Gnu is Not Unix*)



- Richard Marshall Stallman (rms), MIT, 1982
- Nella chiusura del codice vede una aggressione alla libertà: fonda la Free Software Foundation.
- Sceglie di scrivere da zero un S.O. Unix-like, ma completamente libero: GNU project.
- *Tools*: Emacs('84), gcc('85), tar, ld, make, gawk, ...
- *Kernel*: GNU Hurd: è più difficile del previsto e blocca il progetto.
- Per mantenere libero il software: licenza GPL.
- *“Make the world a better place”*.

## Seconda storia: Linux



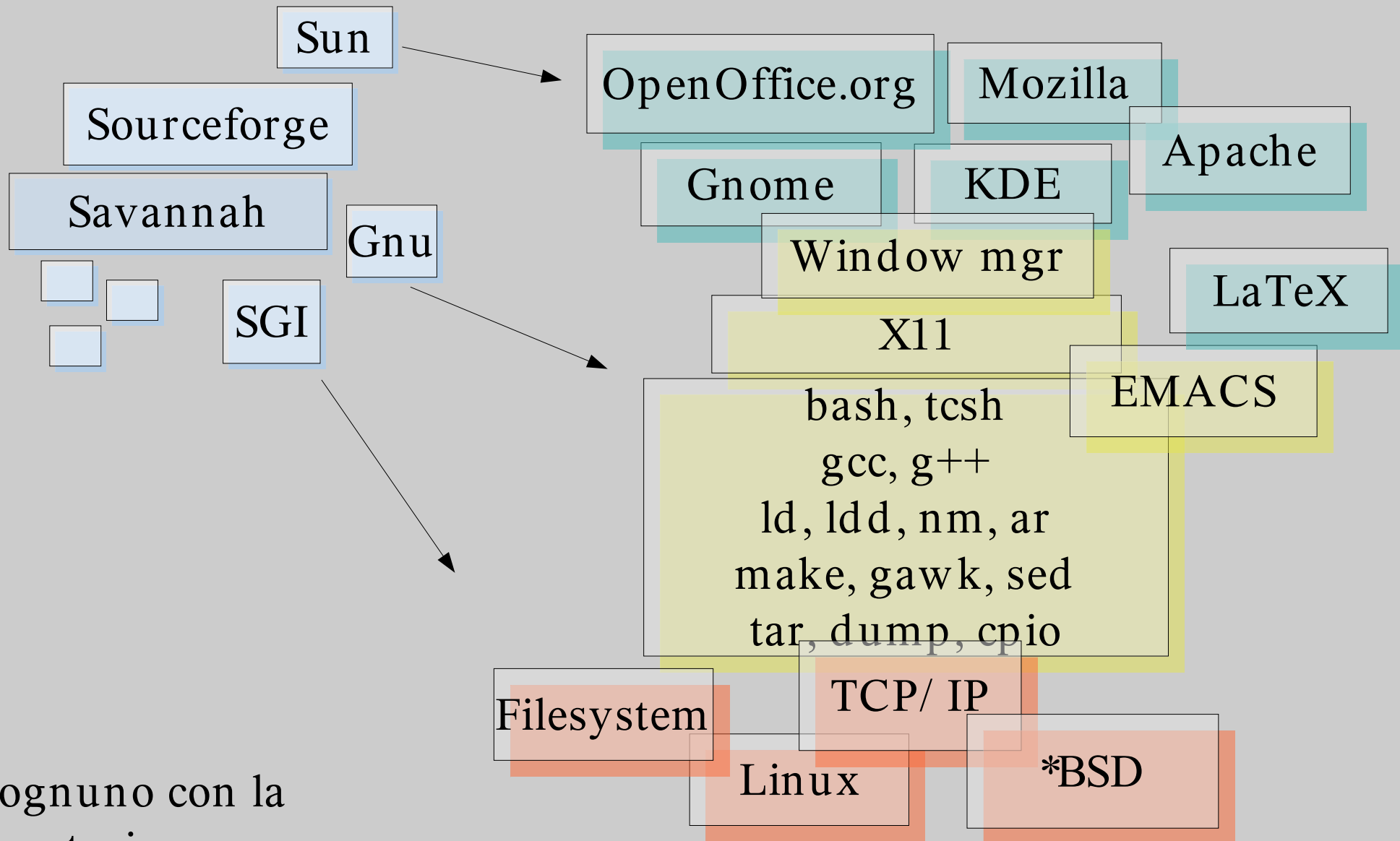
- Linus Torvalds, Helsinki University, 1991
- Pochi soldi, non può comprarsi uno Unix: ispirato da Minix decide di farselo da solo.
- Trova tutti i *tool* GNU già pronti e li usa per sviluppare l'embrione del *kernel*.
- Trova in Internet una comunità di programmatori pronti a ampliare, estendere e correggere il *kernel*.
- GNU/ Linux: completamento del progetto GNU.
- 1994-... Altri programmi ne ampliano il successo.

# Terza storia: Open Source



- Il successo di Linux e degli altri programmi liberi attira (e spaventa) il mercato in pieno boom *' .com '*.
- **Free** (che suona *gratis*) è un termine tabù per il business: **Open Source** è il suo *marketing term*.
- Pragmatismo, meno idealismo: fare software migliore qualitativamente.
- Imprese affermate liberano i loro prodotti: OpenOffice.org (Sun), Mozilla (Netscape).
- Occasione di concorrenza a Microsoft.
- Ammette la coesistenza proprietario/ libero

# Mappa di un sistema libero



...ognuno con la sua storia.



Per essere *free* deve essere libero,  
inoltre  
può essere gratis (e spesso lo è)

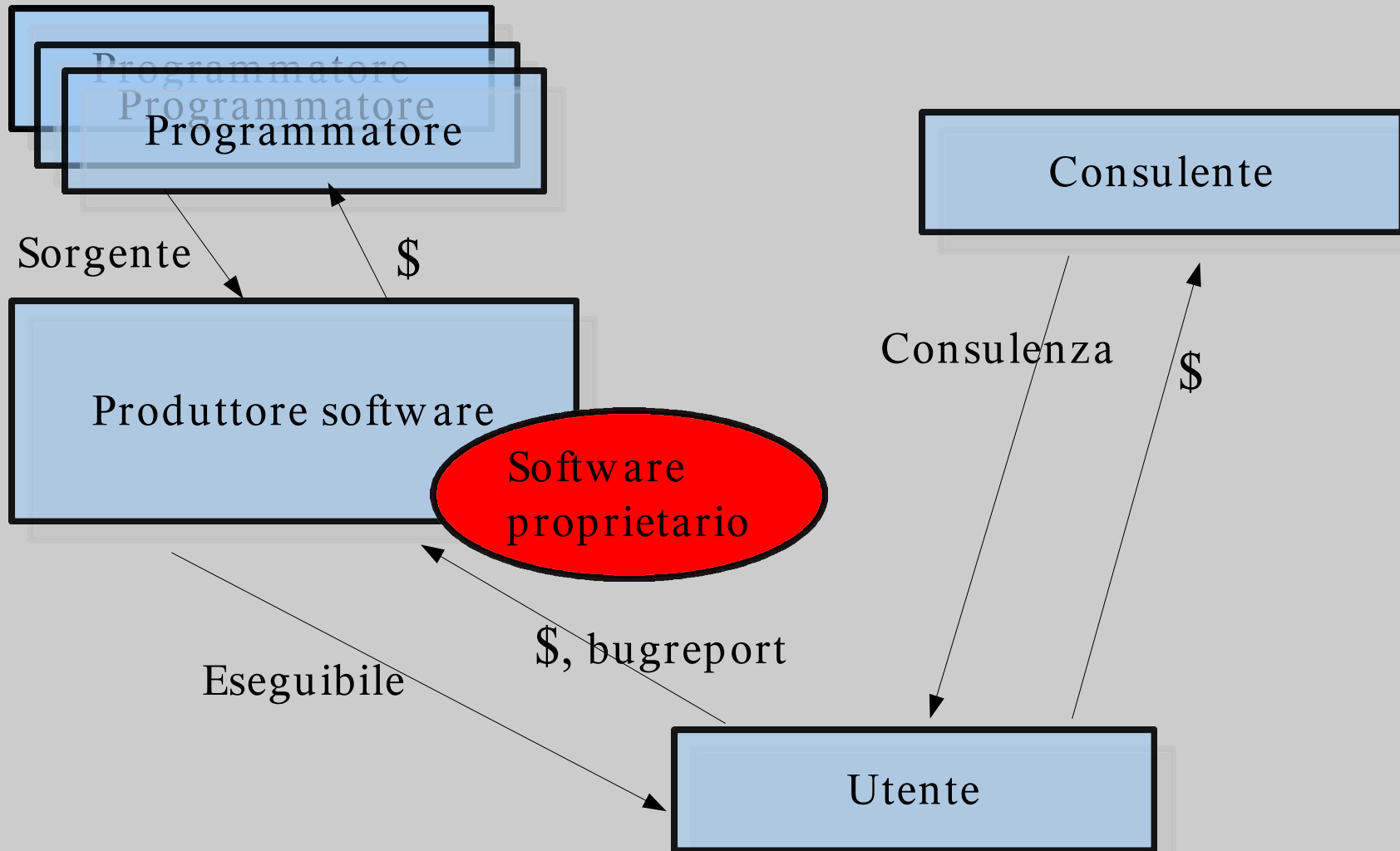
Richard Stallman (FSF) insiste sul termine *free*:

- “*proprietary software divides and subjugates the users*”
- “*a free OS will give users to have freedom while they use their computers*”
- “*The freedom to cooperate with other people, to have a community, is important for quality of life, [...] for having a good society that we can live in, and that is [...] even more important than having powerful and reliable software.*” [dal film *Revolution OS*]

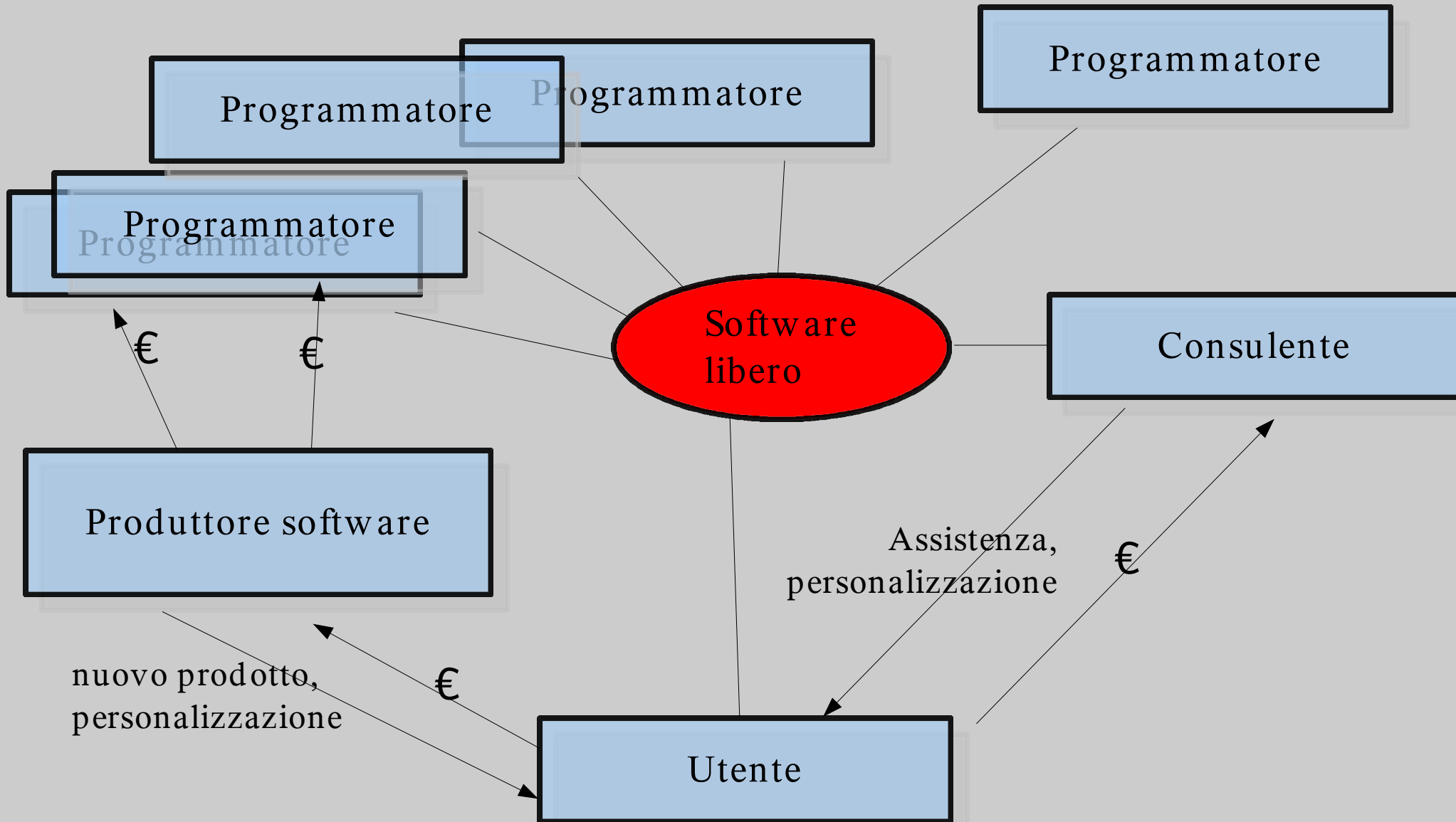
# Ancora due parole sul prezzo

- Il Software può essere:
  - A pagamento per tutti, o **commerciale**.
  - **Shareware**: distribuito gratis, ma l'uso è (di solito) a pagamento.
  - Gratis:
    - **Freeware**: senza i sorgenti, solo eseguibile
    - gratis solo per clienti *no profit*
    - gratis ma non rivendibile
    - gratis senza limitazioni

# Modello di diffusione tradizionale



# Il modello Free/ Open software



# Modelli di sviluppo

## (“la cattedrale e il bazaar” Raymond 1998 )

- **Tradizionale/Cattedrale**
- Sviluppo centralizzato
  - gruppo ristretto di sviluppatori, selezionati dal management
  - *feedback* utente con *bugreport*
- Project management forte
  - standard di sviluppo rigorosi
  - gerarchia più o meno pronunciata
  - disaccordo: abbandono progetto
- Diffusione solo di prodotti “finiti”
  - debugging a carico di chi sviluppa
- **Internet-centrico/Bazaar**
- Sviluppo distribuito, delocalizzato
  - gruppo di sviluppo ampio, autoselezionato
  - *feedback* diretto con gli utenti
- Project management lasco
  - *peer review*
  - coordinatore progetto = filtro
  - in caso di disaccordo: `fork()`
- Diffusione precoce di prodotti grezzi
  - “*more users find more bugs*” (Brooke)
  - il debugging è parallelizzabile

# Perché funziona

- Management capace di coordinare:
  - Modello *single guru*: Don Knuth (TeX)
  - Modello *master-disciples*, o *benevolent dictator*: Linux
  - Modello *project team*: Apache, X11
- Programmatori bravi
- *Timing* giusto
  - Sviluppo, *versioning* (beta, stabile, ...)
- Ampia base di utenti esperti e collaborativi
  - Debugging efficace e veloce
- Fortuna :-)

# Perché non funziona

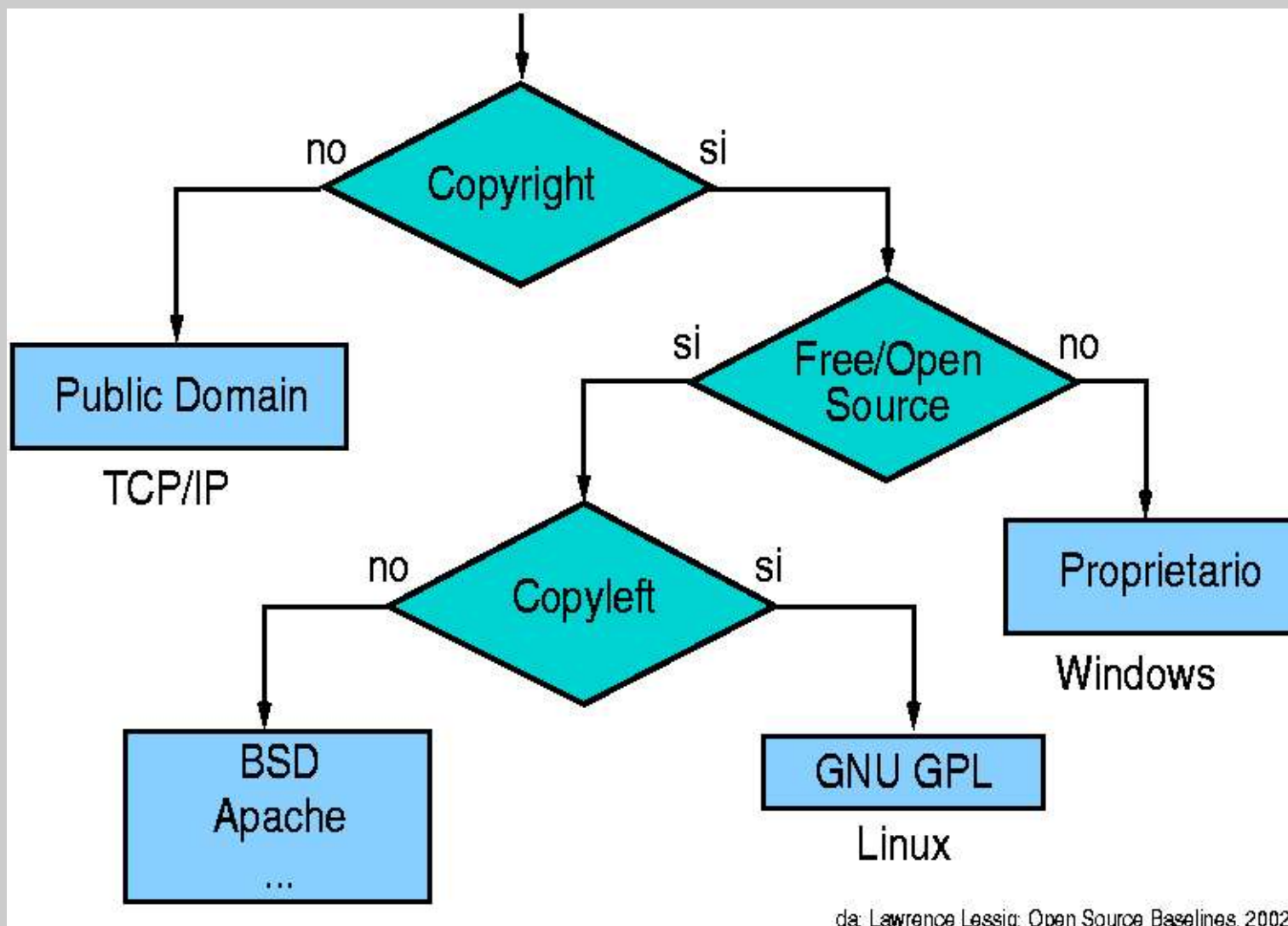
- Leader con le idee “troppo chiare”
- Gruppo di sviluppo chiuso o sparuto
- *Timing*:
  - e.g. GNU/ Hurd: ritardi fatali.
- Fork: sottraggono risorse al progetto principale
  - Emacs/ xemacs, gcc/ egcs,...
  - un esempio per tutti: UNIX
- Errori con le licenze:
  - software libero che diventa proprietario.
  - **cause**: e.g. USL-ATT vs. UCB-BSDI (tre anni per tre file)

# Diritto d'autore

- Il software è opera d'intelletto: come un libro o un progetto.
- È protetto dal “diritto d'autore” (copyright ©)
- L'autore può cedere il diritto sullo **sfruttamento** dei diritti (ma resta l'autore)
- La **licenza** regola la cessione di alcune *facoltà*:
  - copia
  - modifica
  - ulteriore cessione a terzi



# Tipi di licenze



da Lawrence Lessig: Open Source Baselines, 2002

# GNU GPL: General Public License

- Carica ideale FSF: libertà

- Richard M. Stallman, Cambridge MA, 1984

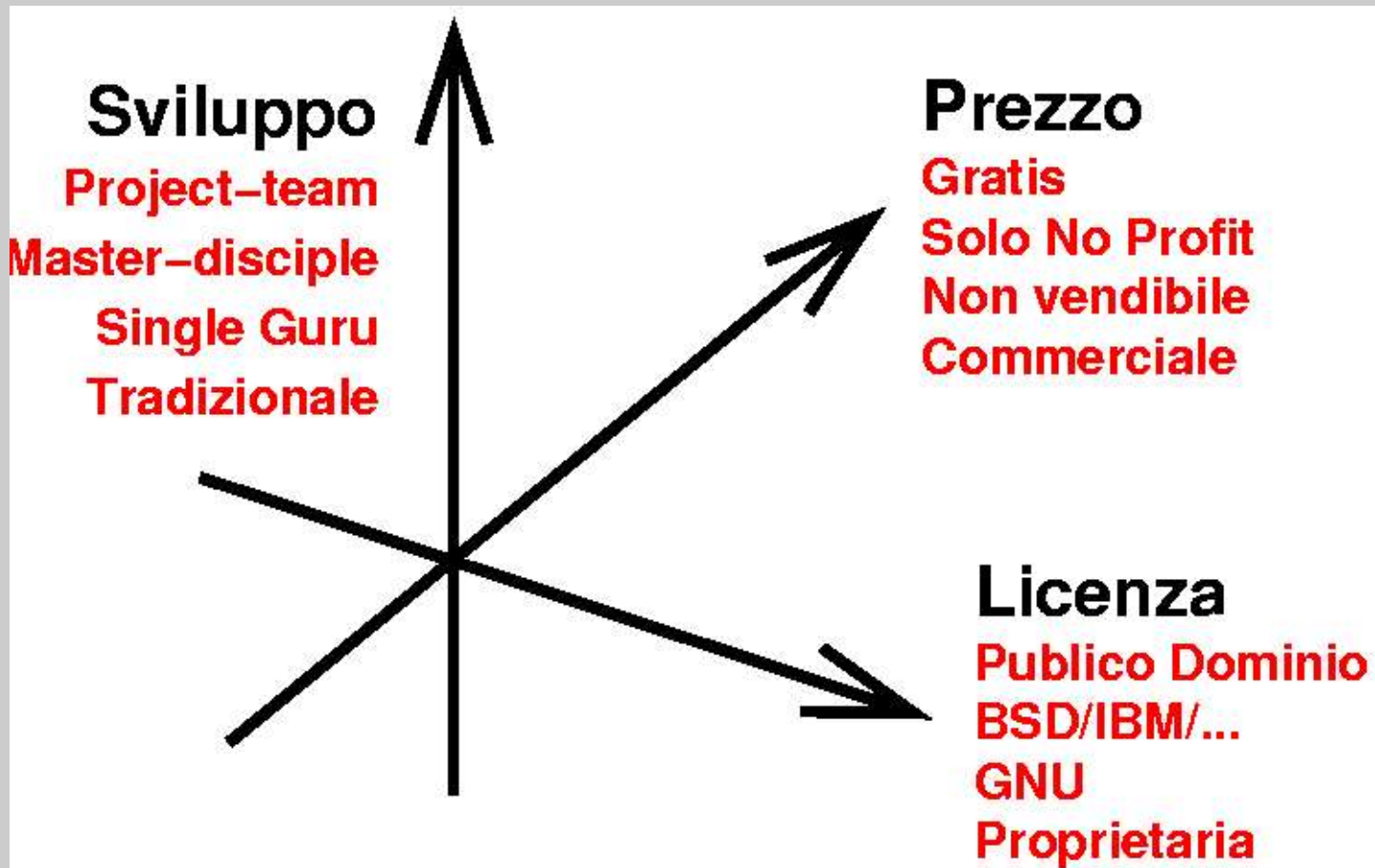


- 0: Libertà di **eseguire** il programma, per qualsiasi scopo.
    - 1: Libertà di **studiare come funziona** il programma, e **adattarlo** alle proprie necessità.
    - 2: Libertà di **ridistribuire** le copie in modo da aiutare il prossimo.
    - 3: Libertà di migliorare il programma, e **distribuirne pubblicamente i miglioramenti**, in modo tale che tutta la comunità ne tragga beneficio.
  - E' **irreversibile**: nessuno può negare i diritti concessi, ovvero chiudere il sorgente.
  - Anche le **opere derivate** devono essere GPL .

# In sintesi: posso classificare il software

- Per **prezzo**:
  - Gratis/ Freeware/ Shareware/ Commerciale
- Per **licenza**:
  - Con sorgenti : GPL, BSD, Public Domain, ...
  - Senza sorgenti :EULA
- Per **modelli di sviluppo**:
  - Cattedrale/ Bazaar
  - Single Guru/ Master-disciple/ Project team/ Tradizionale
  - ...

# Prezzo $\perp$ Modello sviluppo $\perp$ Licenza



# Aspetti Sociali

- Colmare il Digital Divide:
  - basso costo hw e sw
  - riutilizzabilità: sw patrimonio umanità.
- Motore di innovazione:
  - investimento locale
  - sviluppo competenze e risorse locali.
- Rompere la catena della dipendenza da altri:
  - oltre a darti il programma ti dico come farlo
  - adattabilità al contesto locale, anche culturale.

## Aspetti politici

- Software nella Pubblica Amministrazione:
  - richiede sicurezza (sia informatica, sia stabilità)
  - possibilmente a basso costo
  - strategico: è il riferimento del mercato
- I governi devono *sostenere* il S.Libero?
  - devono usare solo quello?
- Risposta del mondo commerciale proprietario
  - rilascio sorgenti con *Non Disclosure Agreement*
  - sconti, lobbying ('Initiative for Software Choice')

# Free/ Open future?

- Documentazione, manuali, lezioni universitarie
- Hardware
  - [www.open-hardware.org](http://www.open-hardware.org)
  - [www.opencores.org](http://www.opencores.org)
- Brevetti sul software
- Digital Rights (*Restriction*) Management:
  - Palladium e TCPA.
- Brevetti sul DNA.
- ...

# Domande (Libere)?

Questa presentazione si trova qui:

<http://www.stat.unipd.it/~mmzz/Papers/foss-math.pdf>



# Approfondimenti

- Vantaggi del F/ OSS per i vari attori
- Approfondimenti sulle licenze
- Alcuni esempi
- Miti sul F/ OSS
- Chi se ne occupa: puntatori
- Sigle
- Bibliografia

# Quali vantaggi: programmatore

- Dipende (da cosa vuole)
  - più soddisfazione (pubblicità, proprietà del codice) e interazione con gente interessante
  - *peer review*, può concentrarsi sulla qualità
  - meno tempo speso in debugging
  - meno €€ (ma se è bravo forse anche di più)
  - può attingere alla 'libreria' di tutto il software libero

# Quali vantaggi: utente/ cliente

- Molti o moltissimi
  - affidabilità: garanzia di supporto continuo
  - sicurezza informatica
  - minore o spesso nessun costo
  - maggior scelta di soluzioni o personalizzazioni
  - *“fly before buy”*

# Quali vantaggi: consulente

- Molti
  - può adattare il prodotto al cliente
  - ampia scelta di soluzioni da proporre
  - fa il consulente e non il venditore
  - interazione diretta col programmatore
  - può correggere direttamente un baco.

# Quali vantaggi: produttore

- Dipende (è ancora presto per sapere)
  - minor spesa per sviluppo e debugging
  - può sfruttare il nuovo modello di sviluppo
  - spese legali?
  - spostare i ricavi dal prodotto ai servizi, marchi...
- C'è molto movimento...
  - Vedere “*pandora's box for open source*” su news.com, 12 feb 2004

# Licenze

- Esempi di licenze per software con sorgente aperto:
  - **Public Domain**: viene ceduto tutto, anche il ©.
  - **BSD** (Berkeley Standard Distribution): dono liberale del codice:
    - può essere reso proprietario, pur restando il © dell'autore.
  - **GPL** (General Public License): collaborativo;
    - insieme al sorgente devono essere ceduti tutti i diritti: “*copyleft*” o *permesso d'autore*.

# Modelli di licenza

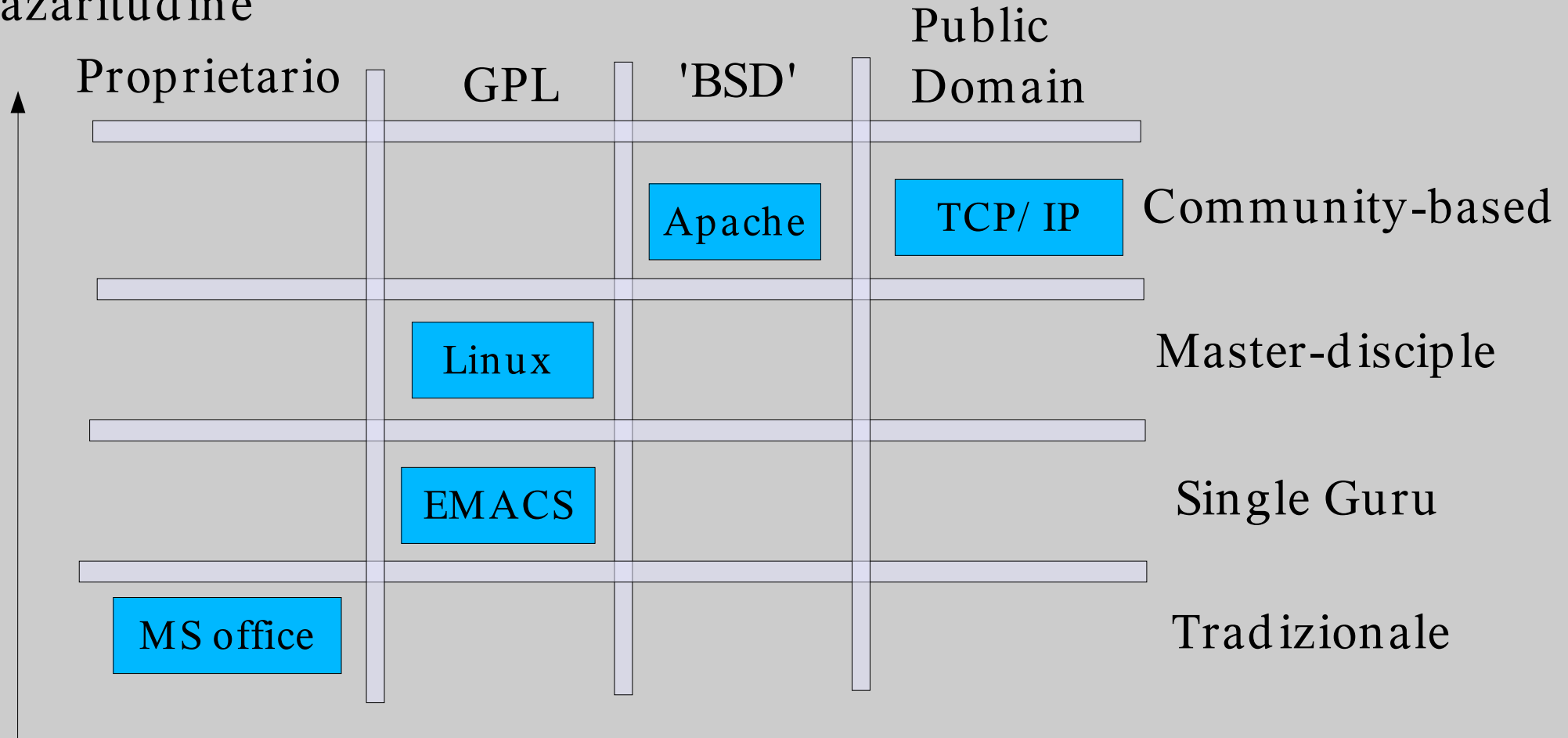
	Esecuz.	©	Lettura/Copia	Modif.	Distribuz.
Public Domain	Si	Si	Si	Si	Si/\$i
BSD	Si	No	Si	Si	Si/\$i
IBM Public license	Si	No	Si	Si	Si/\$i
GNU GPL	Si	No	Si	Si/No	Si/\$i
Shared Source	\$i	No	\$i/No	\$i/No	No
Freeware	Si	No	No	No	Si
Shareware	\$i	No	No	No	Si
Commerciale	\$i	No	No	No	No

Tabella delle *facoltà* o *diritti* concessi all'utente.

Vedere [www.opensource.org](http://www.opensource.org) per le licenze considerate Open dalla *Open Source Initiative* secondo la *Open Source Definition* (OSD).

# Libero/ Proprietario $\perp$ Cattedrale/ Bazaar

Bazaritudine



Cattedralità

Chiusura

Apertura



# Caso MySQL AB



[www.mysql.com](http://www.mysql.com)

- Più diffuso database libero: 4.000.000 installazioni
- Doppia licenza: GPL/ Commerciale
- Clienti: Motorola, NASA, Texas Instruments,...
- 100 dipendenti nel mondo. Migliaia di collaboratori
- Guadagni:
  - 1: supporto/ assistenza online
  - 2: Licenze commerciali
  - 3: Sfruttamento marchio (certificazioni, corsi, ...)


# Caso Ernie Ball



[www.ernieball.com](http://www.ernieball.com)

- Visita BSA (Business Software Alliance) nel 2000:
  - licenze non in regola.
  - multa: \$65,000
  - spese legali: \$35,000
- “voglio Microsoft fuori dalla ditta in 6 mesi”.
- Risparmio: \$80,000 in 3 anni.
- *“The myth has been built so big that you can't survive without Microsoft”* Sterling Ball, 2003
- da [www.news.com](http://www.news.com) 20 agosto 2003

# Miti

- Qualità: “se non pago, è sicuramente scadente.”
- Costo: “il TCO (*Total Cost of Ownership*) del software libero è più alto.”
- Assistenza: “non posso avere un contratto di assistenza.”
- Sicurezza: “il fatto che il sorgente sia leggibile lo rende vulnerabile; meglio la vecchia *security through obscurity* (e le chiavi NSA )”. 

# Chi se ne occupa

- Internazionale:
  - FSF Free Software Foundation : [www.fsf.org](http://www.fsf.org)
  - Open Source Initiative: [www.opensource.org](http://www.opensource.org)
  - GNU : [www.gnu.org](http://www.gnu.org)
  - UNESCO [www.unesco.org/webworld/portal\\_freesoft](http://www.unesco.org/webworld/portal_freesoft)
  - League for Programming Freedom: [lpf.ai.mit.edu](http://lpf.ai.mit.edu)
  - Electronic Frontier Foundation: [www.eff.org](http://www.eff.org)
  - Foundation for a Free Information Infrastructure: [www.ffii.org](http://www.ffii.org)
- Italia
  - AsSoLi (Associazione Software Libero): [www.softwarelibero.it](http://www.softwarelibero.it)
  - ILS (Italian Linux Society) [www.linux.it](http://www.linux.it)
  - PLUTO: [www.pluto.it](http://www.pluto.it)
- Padova
  - PLUTO Padova: [www.plutopadova.org](http://www.plutopadova.org)

# SIGLE

- BSD : Berkeley Standard Distribution
- DRM : Digital Rights Management
- EULA: End User License Agreement
- FSF: Free Software Foundation
- GNU : GNU's Not Unix
- GPL : General Public License
- IPR : Intellectual Property Rights
- NSA : National Security Agency
- OS: *operating system*, sistema operativo
- TCO: Total Cost of Ownership
- TCPA : Trusted Platform Computer Architecture
- GNU Hurd: Hird of Unix-Replacing Daemons. Hird: Hurd of Interfaces Representing Depth.

# Bibliografia

- Raymond, Eric S. ; *A Brief History of Hakerdom*, 2000, [http:// www.catb.org/ ~esr](http://www.catb.org/~esr)
- Stallman, Richard M.; *Free Software, Free Society*; Boston 2002, GNU Press
- Raymond, Eric S.; *The Cathedral & the Bazaar*, (2 ed.) O'Reilly, Sebastapol, CA, 2001.
- Bezroukov, Nikolai; *A Second Look at the Cathedral and Bazaar*, First Monday, volume 4, number 12 (December 1999); [http:// firstmonday.org](http://firstmonday.org), [http:// www.softpanorama.org](http://www.softpanorama.org)
- DiBona, Chris; Ockman, Sam; Stone, Mark; editors; *Open Sources: Voices from the Open Source Revolution*. O'Reilly and Associates, Cambridge, Massachusetts, 1999
- van Reijswoud, Victor; Topi, Corrado; *Alternative Routes in the Digital World: Open Source Software in Africa*, 2003
- Torvalds, Linus; Diamond, David; *Just for Fun*, Texere, London, 2001. (*Rivoluzionario per caso*, Garzanti)
- Hahn, Robert W., editor; Bessen, James; Evans, David S.; Lessig, Lawrence; Smith, Bradford L.; *Government Policy toward Open Source Software*; AEI-Brookings, 2002
- Moore, J.T.S.; *Revolution OS*, Wonderview Productions, LLC, 2002 (film)

# Fine

(happy hacking)