

LINUX:
la forza di un pinguino
(e di uno GNU)

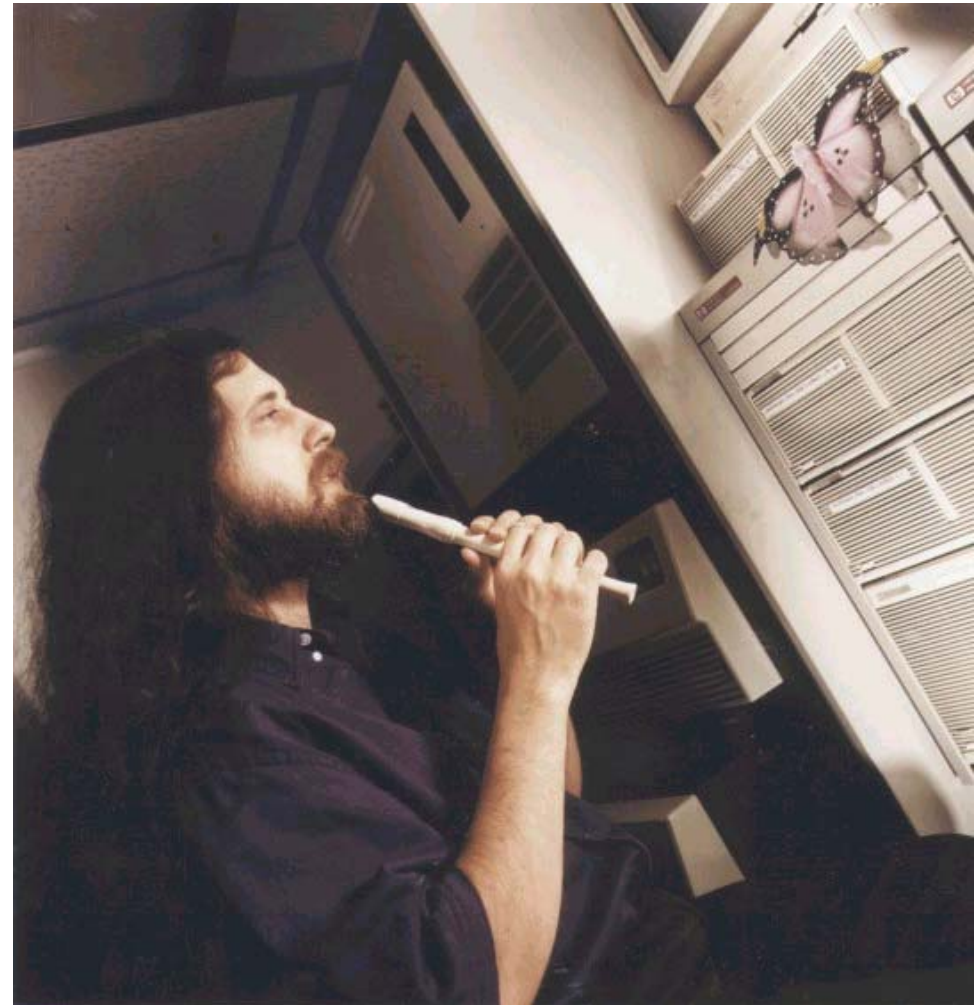


Prima è nato lo GNU

Nel 1984 Richard Stallman avvia lo GNU project basato sul principio del free software (reazione a S.O. proprietari)

Nel 1985 nasce la Free Software Foundation:
“free” as in “free speech” not as in “free beer”

freedom to run, copy, distribute, study, change and improve the software (gratis o a pagamento)



Free software vs. open source

- L'accesso al codice sorgente è condizione necessaria ma non sufficiente per le libertà di studiare e modificare il software
- Il software disponibile gratuitamente non necessariamente è free, nel senso di Stallman
- Se il software viene pubblicato (reso public domain) senza condizioni chi lo usa potrebbe trasformarlo da free a non free (p.e. venderlo senza rendere disponibili i sorgenti delle modifiche)

Dai principi alle leggi

- Gli autori dei programmi detengono il diritto d'autore di ciò che scrivono, automaticamente o previ adempimenti legali
- Gli autori possono mantenere o cedere (p.e. al datore di lavoro o altri) il copyright sui programmi
- Chi detiene il copyright decide se e quali diritti concedere ad altri e a quali condizioni: questi termini vengono stabiliti in una licenza
- Esistono numerosi tipi di licenze

Una classificazione di massima: software free o quasi

- Free software: viene reso disponibile con una licenza compatibile con la definizione di FSF
- Open source: viene reso disponibile il codice sorgente, ma può avere licenza restrittiva(non free)
- Public domain: gli autori rinunciano al copyright, chiunque può farne ciò che vuole
- Semi-free software: reso disponibile in termini free solo ad alcuni (p.e. singoli ma non enti)
- Private software: noto e usato dal solo sviluppatore

Una classificazione di massima: software non free

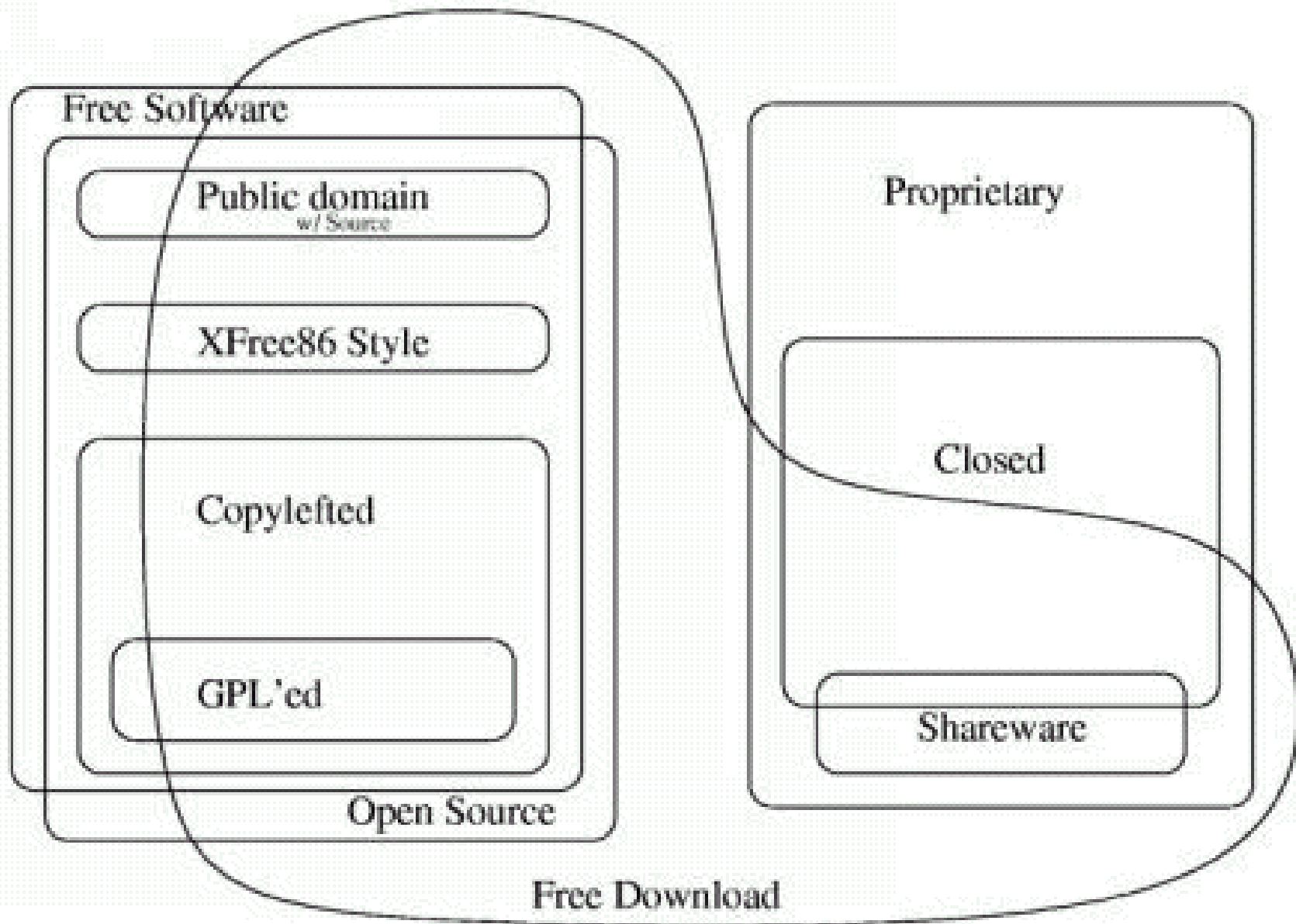
- Software proprietario: licenza vieta uso, redistribuzione, modifica o li rende impraticabili
- Freeware: permessa la distribuzione, non la modifica, di solito non sono open source
- Shareware: permessa la distribuzione, vietato l'uso prolungato, di solito non sono open-source
- Commercial software: qualcuno si fa pagare per distribuirlo o fornire assistenza. Distinzione ortogonale rispetto a free/non-free !

Il copyleft

- Chi rilascia software libero con licenza può imporre delle condizioni a chi lo usa
- Copyleft:
 - » obbliga chi usa il software libero a rilasciare come liberi i prodotti da esso derivati
 - » non impedisce lo sfruttamento commerciale in sè ma l'uso di software libero per realizzare software non libero
- Permette (o forza) riuso e cooperazione
- Tutela da “appropriazioni indebite”

Le licenze

- Esistono numerosissimi tipi e testi di licenze
- Le licenze di riferimento del mondo free sono
GPL: General Public License (con copyleft)
LGPL: Lesser GPL (senza copyleft)
GFDL: GNU Free Documentation License (con copyleft)
- La FSF distingue le altre licenze in:
free o non free, con o senza copyleft,
compatibile o no con GPL (permette software derivati che includano software GPL o no)



Non tutti sono a favore ...

- Leggi varie in vari paesi in certi casi sono (volutamente o meno) un potenziale ostacolo alla diffusione del software libero
- Problema dei “brevetti software”: in alcuni paesi si possono brevettare algoritmi, così come oggetti software
- Il reverse engineering può essere punito
- Bollino SIAE (legge 248/2000)
- Argomenti contro il software libero

Non tutti sono sulla stessa linea ...

- Open Source Initiative (www.opensource.org): organizzazione non-profit dedita alla promozione e certificazione di software “open source”
- Decalogo dell’open source, simile a principi FSF
- Promozione di open source come caratteristica tecnica anziche’ come principio filosofico
- Concetto di free software considerato fuorviante e poco commerciabile
- Abbandonare l’immagine “underground” e confrontarsi col mercato (e col marketing)

Leggende metropolitane ?

- Pratiche FUD (Fear, Uncertainty, and Doubt)
- Il software libero è:
 - meno sicuro
 - meno affidabile
 - meno efficiente
 - meno assistito
 - meno aggiornato
 - roba da hacker
 - rende impossibile ai programmatori guadagnarsi il pane
 - sempre meglio del software commerciale
 - va bene per chiunque
 - la comunità risponde ad ogni domanda
 - aggiornato continuamente e tempestivamente

Vantaggi del software libero

- Il (non)prezzo
 - » non per tutti è il fattore cruciale
 - » investimento nel know-how non nel prodotto
 - » non tutto il free è necessariamente gratis
- Indipendenza da un fornitore unico
- Verificabilità e modificabilità del codice
- Patrimonio pubblico
 - » accesso libero alla tecnologia
 - » cooperazione diffusa, riutilizzo del codice, miglioramenti
 - » incentivo al volontariato

Dalle parole ai fatti

- I buoni propositi non servono a nulla se non sono accompagnati da software BEN funzionante !
- Sul sito FSF sono elencati più di 3500 package di software di ogni tipo
- Compilatori e librerie per C, C++, Fortran, Java, Ada (GCC Gnu Compiler Collection)
- Editor di testo: emacs
- Utility di ogni tipo: compressione file, manipolazione immagini, e-mail, giochi, ...

Software “libero” non GNU

- Servizi in rete:
 - Bind (DNS)
 - Apache (Web)
 - Sendmail (E-mail)
- Linguaggi di programmazione
 - Perl
 - Tcl/Tk
 - Python
- Tantissimo altro ...

Cosa manca(va) allo GNU ?

- Tra i software GNU c'è un kernel (Hurd) il cui obiettivo era rimpiazzare i kernel UNIX commerciali
- Hurd non ha mai preso piede
- I software liberi erano “costretti” a girare su S.O. commerciali
- Poi è arrivato Linux
- L'accoppiata GNU/Linux, programmi/S.O. entrambi free e di qualità è stata la chiave del successo

Storia di Linux

- Nel 1991 Linus Torvalds, studente di Tanenbaum, inizia a scrivere un nuovo sistema operativo per PC (all'inizio conosce Minix e sa poco di UNIX, POSIX, ecc.)

Sul newsgroup comp.os.minix dice:

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones.

- In pochi mesi realizza una prima versione funzionante (usando MINIX e strumenti GNU, p.e. gcc) e la mette a disposizione in rete: si crea una comunità di collaboratori
- La dimensione e gli obiettivi del progetto crescono con la comunità di collaboratori
- Nasce anche FreeBSD ma è ritardato da questioni legali

Versioni del kernel

- Il primo numero indica una revisione maggiore, il secondo una importante, il terzo bug fixing e simili
 - Il secondo numero è pari per versioni stabili, dispari per versioni sperimentali
 - 1994: 1.0.0
 - 1996: 2.0.0
 - 1999: 2.2.0
 - 2000: 2.4.0
 - 2003: 2.6.0
- 1998: KDE e GNOME
- oggi: 2.6.36

Diffusione di Linux

- Crescita fortissima specialmente in ambito professionale e universitario
- Non solo bassi costi, ma affidabilità e prestazioni
- Portabilità
- Notevole disponibilità di applicazioni
- C'è anche un po' di "moda" e aggregazione sociale
- Numerosi "case history" di successo presso enti "famosi" (NASA, Boeing, ...)
- Supportato da IBM, Oracle, ...

Problemi di compatibilità

- Linux non gira necessariamente ovunque e può essere (temporaneamente) non dotato di driver per certe periferiche
- Problemi più probabili a livello di PC personali che di server: PC portatili, schede audio e video, stampanti, scanner, dispositivi USB e Firewire, masterizzatori vari, modem.

Ottenere Linux

- Si può ottenere una macchina Linux funzionante scaricando tutti i sorgenti o eseguibili necessari e compilando/installando il tutto (Linux from scratch)
- E' piu' comodo scaricare o ottenere una copia di una distribuzione
- Una distribuzione è una raccolta di: kernel, software di base, programmi applicativi vari
- Le distribuzioni possono differire per: tipo di licenza, costi, supporto fornito, organizzazione del FS e dei servizi, disponibilità driver, ...

Distribuzioni varie

- Di tutto. Un elenco si puo' ottenere da www.linux.org/dist:
 - » Debian: un po' spartana, molto fedele al concetto free
 - » Gentoo: orientata all'ottimizzazione e configurabilità
 - » Knoppix: tutto su CD (demo o emergenza)
 - » Mandriva: general purpose e "easy-to-use"
 - » Red Hat: general purpose e "easy-to-use", ha cambiato politica e non esiste più free con questo nome
 - » Fedora: derivato free di Red Hat
 - » Ubuntu : sponsorizzata da un miliardario sudafricano, Linux per tutti
 - » Slackware: la più UNIX-like (?)
 - » Suse: azienda tedesca
 - » Trustix: orientata alla sicurezza
 - » Turbolinux: diffusa in Giappone e Cina

L'installazione: fase zero

- Prima di tutto si *dovrebbe*:
 - fare backup se non è una nuova installazione
 - creare floppy o avere CD per boot emergenza
 - verificare eventuali incompatibilità hardware
 - verificare requisiti minimi
- Sicuramente si deve:
 - sapere se la macchina può fare boot da CD e configurare il BIOS a tale scopo se necessario
 - avere una partizione libera per il sistema (ripartizionare se necessario)