Introduzione al movimento del Software Libero

Breve introduzione alla storia del movimento del Software Libero, da Unix al progetto GNU a Linux.

a cura di Stefano Sabatini, plapia@tele2.it

Cos'è il codice sorgente?

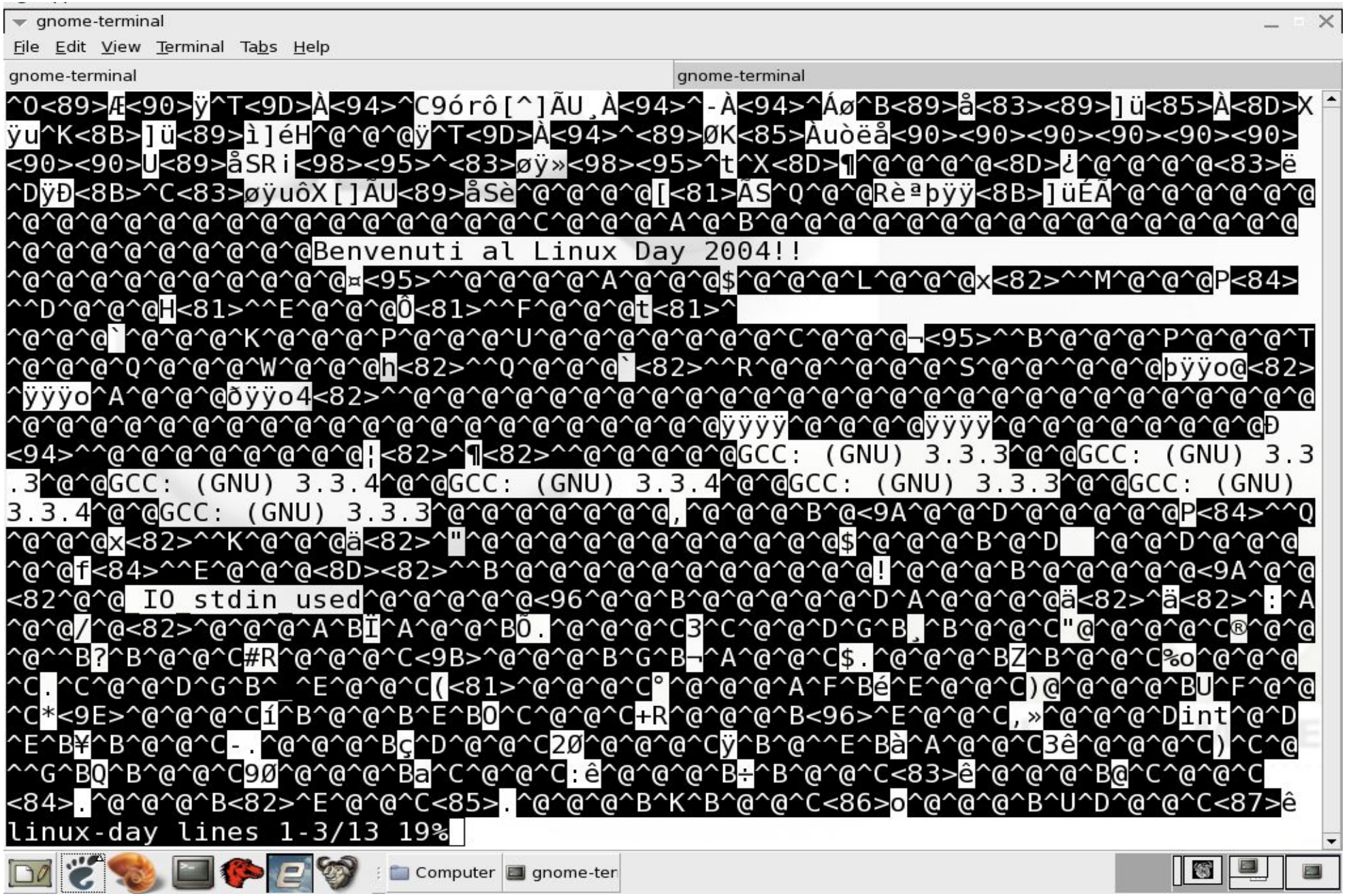
- Esempio di codice sorgente

```
#include <stdio.h>
#include <stdlib.h>

int main (void)
{
   /* Questo programma stampa un messaggio di benvenuto
    * per tutti i gentili ospiti del Linux day. */

   printf ("Benvenuti al Linux Day 2004!!\n");
   return 0;
}
```

Esempio di codice binario



Processo di compilazione del codice sorgente

esempio:
bash-shell $ gcc linux-day.c -o linux-day

- significato comprensibile ad un umano competente (human-readable);
- facilmente modificabile;
- compilabile per più macchine diverse (portabile).

- incomprensibile anche ad un umano esperto;
- impossbile da modificare;
- eseguibile solo sulla macchina per cui è stato compilato.

Stato dell'informatica fine degli anni '60

- macchine grandi, costose, inaffidabili
- assenza di standard per la architettura hardware delle macchine; selva di macchine diverse
- sistemi operativi embrionali scritti nel linguaggio assembly delle varie macchine

- 1969: D. Ritchie, K. Thompson, B. Kernighan e altri iniziano a lavorare al sistema operativo UNIX presso la AT&T
- 1973: il kernel UNIX viene riscritto in C: vengono rilasciate varie versioni a Università, compagnie, enti governativi

UNIX

Caratteristiche di UNIX:

- scritto in un linguaggio di alto livello (C): rapidi tempi di sviluppo, portablità;
- adozione principi di ingegneria del software orientati alla modularità e alla riusabilità del codice; filosofia degli strumenti;
- tra i primi SO a supportare i protocolli di rete (primi anni 80: BSD UNIX);
- potenti strumenti di manipolazione di testo; tipo di approccio “tutto è testo”;
- determina la diffusione di una serie di norme culturali e di programmazione tra gli sviluppatori (UNIX philosophy);
- alla base dello standard POSIX che definisce l'interfaccia di un sistema operativo

'80: Diffusione del software proprietario e del PC

- versioni commerciali di UNIX distribuite senza il codice sorgente (1982: UNIX system III, AT&T, vecchie versioni (1-7) rivendute dalla Western Electric);
- 1981: PC IBM viene venduto con incluso MS-DOS (inizialmente QDOS) (40 $) oppure CP/M (250$)
- diffusione architettura PC IBM e cloni; MS-DOS+PC diventa lo standard di fatto nel mercato dei PC;
- politica commerciale aggressiva di MS anche nel campo degli applicativi: predominio su Lotus1-2-3 e WordPerfect;
- Diffusione della pratica dei patti di non-disclosure (non-disclosure agreement);
- tramonto della cultura hacker (basata sul concetto di condivisione del codice).

Il tramonto della cultura hacker

“Quando iniziai a lavorare al laboratorio di Intelligenza Artificiale del MIT nel 1971, divenni parte di una comunità dove il software veniva condiviso che esisteva da molti anni. La condivisione del software non era limitata alla nostra particolare comunità; era una pratica vecchia quanto i computer, così come lo scambio di ricette è vecchio come la cucina. Solo, noi lo facevamo più di altri. [...]

Noi non chiamavamo il nostro software "software libero" (free software), perchè quel termine ancora non esisteva; ma di fatto lo era. Tutte le volte che qualcuno da qualche università o compagnia voleva portare o usare un programma, eravamo ben lieti di lasciarglielo fare. [...]

La situazione cambiò drasticamente nei primi anni 80 [...].

Nel 1981, la compagnia Symbolics aveva assunto quasi tutti gli hacker del laboratorio di IA, e la comunità spopolata non era più in grado di autosostenersi. [...]

I moderni computer dell'epoca, come il VAX o il 68020, avevano i loro sistemi operativi, ma nessuno di essi era free software: era necessario firmare un patto di non-disclosure persino per avere la copia di un eseguibile.

Questo significava che il primo passo da fare per usare un computer era promettere di non aiutare il tuo vicino. Una comunità cooperativa veniva di fatto proibità. La regola stabilita dai rivendiori di software proprietario era, "Se scambi software con il tuo vicino, sei un pirata. Se vuoi una modifica del software, pregaci per averla."

Richard Stallman, da “THE-GNU-PROJECT”

Una dura scelta morale

“Avrei potuto entrare nel mondo del software proprietario, firmando patti di non-disclosure e promettendo di non aiutare i miei compagni hacker. Molto probabilmente avrei sviluppato programmi, diffusi con accordi di non-disclosure, aumentando così la pressione su altre persone a tradire a loro volta i loro compagni.

Avrei potuto fare soldi in questo modo, e forse mi sarei anche divertito a scrivere codice. Ma sapevo che alla fine della mia carriera, mi sarei voltato dietro e avrei visto anni passati a costruire muri per dividere le persone, e avrei sentito di aver speso la mia vita per fare del mondo un posto peggiore.

[...]

Allora mi guardai intorno alla ricerca di qualcosa di buono che potessi fare come programmatore.
Mi chiesi: c'è un programma o dei programmi che posso scrivere, in modo da rendere possibile ancora una volta una comunità?

La risposta era chiara: c'era bisogno come prima cosa di un sistema operativo. [...] Con un sistema operativo libero, potevamo ancora avere una comunità collaborativa di hacker--e invitare chiunque ad entrarvi. E chiunque sarebbe stato in grado di usare un computer senza aver intenzione di cospirare contro i suoi amici.”

Richard Stallman, da THE-GNU-PROJECT-1998

GNU is not UNIX

“Unix non è il mio sistema ideale, ma non è nemmeno tanto male. Le caratteristiche essenziali di UNIX sembrano essere buone, e penso di poter riuscire a migliorarlo ove è carente senza rovinare i suoi pregi. Inoltre molte persone avrebbero convenienza ad adottare un sistema compatibile con Unix.

[...]
A questo punto abbiamo un Emacs text editor con Lisp per scrivere commandi di editing, un debugger a livello sorgente, un parser generator yacc-compatibile, un linker, e circa 35 utilities. Una shell (interprete dei comandi) è quasi concluso. Un nuovo compilatore C, ottimizzante e portabile ha compilato se stesso e dovrebbe essere rilasciato quest'anno. Un kernel iniziale esiste ma molte nuove caratteristiche sono necessarie per emulare Unix. Quando il kernel e il compilatore saranno completati, sarà possibile distribuire un sistema GNU utilizzabile per lo sviluppo di programmi. Useremo TeX come il nostro sistema di formattazione di documenti, ma stiamo iniziando a lavorare su un nroff. Inoltre useremo il sistema libero, portabile X window. Dopo di questo aggiungeremo un Common Lisp portabile, un clone del gioco Empire, un foglio di calcolo, e centinaia di altre cose, più la documentazione on-line. Alla fine speriamo di fornire qualsiasi cosa si trova normalmente in un sistema Unix, e anche di più.”

Richard Stallman, GNU-manifesto, 1984

Free as in Freedom

“Il termine "software libero" ["free software"] spesso non viene ben compreso--non ha niente a che fare con il prezzo. Esso riguardo piuttosto la libertà. Ecco pertanto la definizione di software libero: un programma è software libero, per te, come utente, se:

* Hai la libertà di eseguire il programma, per qualsiasi scopo.
* Hai la libertà di modificare il programma per adattarlo alle tue esigenze. (Perchè questa libertà sia di fatto praticabile, devi avere accesso al codice sorgente, dal momento che fare modifiche in un programma senza avere il codice sorgente è eccezionalmente difficile.)
* Hai la libertà di ridistribuire copie, gratis oppure in cambio di una ricompensa.
* Hai la libertà di distribuire versioni modificate del software, in modo che la comunità posso beneficiare dei miglioramenti da te apportati.

[...]

La filosofia del free software rinnega uno specifico tipo di pratica diffusa degli affari [business], ma non è contro gli affari. Se la pratica affaristica rispetta la libertà degli utenti, noi gli auguriamo il pieno successo.”

Richard Stallman, da "The GNU manifesto",1984

Copyleft

"Copyleft--all right reversed."

Don Hopkins in una lettera a Stallman

“L'obiettivo principale di GNU è quello di dare libertà agli utenti, non solo di essere popolare. Così abbiamo bisogno di usare dei termini nella distribuzione che prevengano che il software GNU venga cambiato in software proprietario. Il metodo che usiamo è chiamato "copyleft".

Copyleft usa la legge del copyright, ma la ribalta in modo da servire all'opposto dei suoi soliti scopi: invece di diventare un mezzo per privatizzare il software, diventa un mezzo per mantenere il software libero.

L'idea centrale del copyleft è che noi diamo a chiunque i permessi per eseguire un programma, copiarlo, modificarlo, ridistribuire le versioni modificate--ma non il permesso di aggiungere restrizioni per proprio conto. Così, le libertà cruciali che definiscono il "free software" sono garantite a tutti quelli che hanno una copia; esse diventano di fatto dei diritti inalienabili.

[...]
La specifica implementazione del copyleft che usiamo per la gran parte del software GNU è la GNU General Public License [Licenza Pubblica Generale GNU], in breve GNU GPL. [...]”

Richard Stallman, The GNU project, 1984

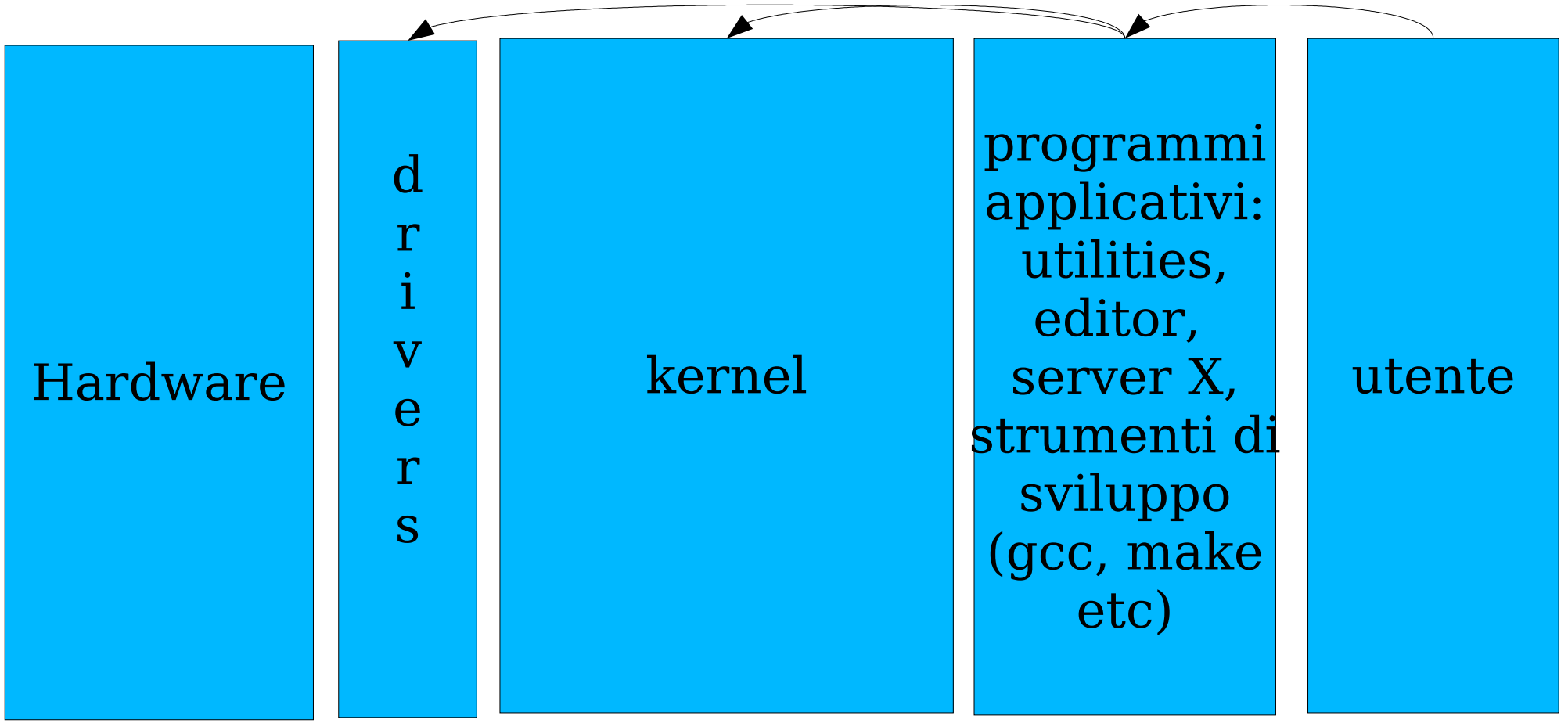
Implicazioni politiche e sociali della filosofia del Free Software

“Il mio scopo è quello di rendere possibile alle persone di rifiutare le catene che vengono imposte dal software proprietario. So che ci sono persono che vogliono farlo. [...]
Ora, le persone che sono consapevoli della sgradevolezza dei termini del software proprietario sentono di essere bloccate e di non avere un alternativa se non quella di non usare un computer. Ebbene, io sto cercando di dare a queste persone un'alternativa accettabile.
Altre persone potrebbero scegliere di usare un sistema GNU semplicemente perchè è tecnicamente superiore. [...]
Ma io continuerei a utilizzare un sistema GNU anche se non sapessi come renderlo tecnicamente migliore perchè io voglio che sia socialmente migliore. Il progetto GNU è per davvero un progetto sociale. Esso utilizza mezzi tecnici per conseguire un cambiamento nella società.

[...]
Io sto cercando di cambiare il modo a cui la gente vede l'informazione e la conoscenza in generale. Penso che provare a possedere la conoscenza, provare a controllare se alle persone è consentito usarla, o provare a fermare altre persone dal condividerla, è sabotaggio. E' un'attività che favorisce la persona che la svolge al prezzo di impoverire tutta la società. [...]
Vorrei vedere le persone venire ricompensate per scrivere software libero e per incoraggiare altre persone ad utilizzarlo. Non voglio vedere persone venire ricompensate per scrivere software proprietario perchè quello non è un reale contributo alla società. “

Richard Stallman, intervista alla rivista BYTE, 1984

Schema di funzionamento di un sistema UNIX



Linux

> “Intorno al 1990, il sistema GNU era quasi completo: l'unico componente principale mancante era il kernel.“
>
> Richard Stallman, THE GNU-PROJECT

Message-ID: 1991Aug25.205708.9541@klaava.helsinki.fi
From: torvalds@klaava.helsinki.fi (Linus Benedict Torvalds)
To: Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system

Hello everybody out there using minix-I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386 (486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat

Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus

Linus Torvalds (foto recente)

The Cathedral and Bazaar

“[Lo sviluppo del kernel] richiederà una stretta comunicazione, e sarà sviluppato da un gruppo di sviluppatori piccolo e coeso.”

Richard Stallman, GNU-manifesto, 1984

“Linux è sovversivo. Chi avrebbe mai immaginato anche solo cinque anni fa che un sistema operativo in uso in tutto il mondo potesse nascere come per magia dal lavoro part-time di parecchie migliaia di programmatori sparsi per tutto il mondo, connessi solamente dai tenui legami di Internet?

[...]

Penso che la cosa più intelligente e ricca di conseguenze fatta da Linus non sia stata tanto la costruzione del kernel di Linux, bensì l'invenzione del modello di sviluppo di Linux.”

E.S.Raymond, The Cathedral and Bazaar, 1997

Sviluppo a cattedrale

> La complessità e i costi della comunicazione di un progetto software aumenta con il quadrato del numero degli sviluppatori, mentre il lavoro fatto cresce solo linearmente.
>
> Legge di Brooks

Un manager andò dal maestro programmatore e gli mostrò i documenti dei requisiti per una nuova applicazione. Il manager chiese al maestro: “Quanto tempo richiederà la progettazione di questo sistema se gli assegno cinque programmatori?”

“Ci vorrà un anno,” disse subito il maestro.

“Ma noi abbiamo bisogno di quel sistema immediatamente o al più presto possibile! Quanto tempo ci vorrà se gli assegno dieci programmatori?

Il maestro programmatore si accigliò: “In quel caso, ci vorranno due anni.”

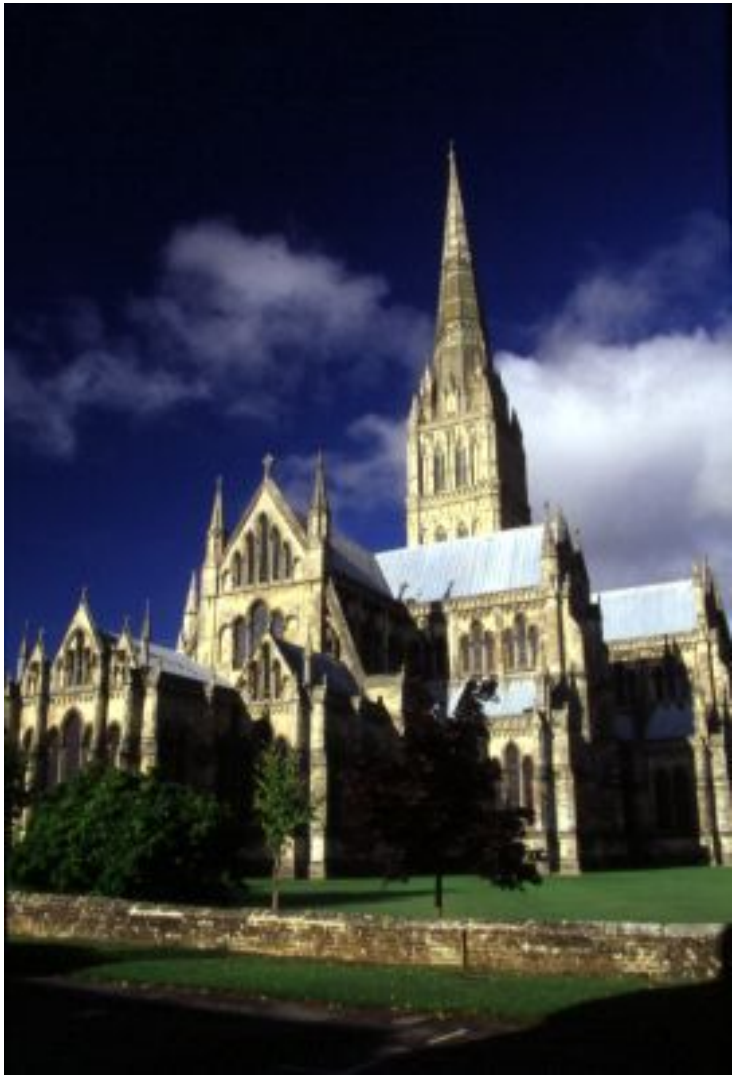
“E se invece assegno cento programmatori al progetto?

Il maestro programmatore si strinse nelle spalle: “Allora il progetto non verrà mai terminato”, disse.

Geoffrey James, The Tao Of Programming

Caratteristiche di un progetto stile Cattedrale

- Sviluppo verticale: struttura piramidale del gruppo di sviluppo;
- Gli sviluppatori lavorano a stretto contatto;
- Necessità di una figura carismatica forte all'interno della comunità di sviluppo e/o di una forte coesione ideologica e tecnica (gli sviluppatori sono d'accordo sul come e sul perchè dello sviluppo);
- Viene sviluppato ciò che è necessario in quel momento per il progetto;
- Il software è rilasciato solo a lunghi intervalli di tempo con delle sostanziali modifiche tra una release e quella successiva
- Esempi: GNU-Hurd, GNU-Emacs (C-core), in genere tutto progetti software “closed source”



Cattedrale di Salisbury (1220-60)

Bazaar-mode

> “La comunità di Linux sembra come un grande e rumoroso bazaar, ognuno con diversi interessi e incontri da fare [...] e da cui un sistema stabile e coerente potrebbe apparentemente emergere solo da una successione di miracoli.”
>
> E.S.Raymond, “The Cathedral and Bazaar”, 1997

• rilasci frequenti di software, anche se pieno di errori (bug) (“release early and often”);
• gli utenti e i co-sviluppatori sono coinvolti nel processo di sviluppo, vedendo un immediata ricompensa ai loro sforzi data dalle frequenti release (rilasci) del software;
• il software sviluppato risolve un problema per la comunità di utenti e sviluppatori, che sono incentivati a migliorarlo in tempi rapidi (“Every good work of software starts by scratching a developer's personal itch.”-- Ogni buon software parte dal risolvere una particolare esigenza di uno sviluppatore);
• il software è sottoposto **contemporaneamente** all'analisi di molti co-sviluppatori e utenti, da molti punti di visti e con diversi utilizzi: i bug vengono scoperti rapidamente e segnalati;
• i bug vengono sottoposti all'attenzione di molti sviluppatori: la soluzione al problema viene trovata rapidamente e integrata nel software (Linus' law: “Given enough eyeballs, all bugs are shallow.")
• i co-sviluppatori, utenti e beta-tester contribuiscono al progetto anche in altri modi, dando idee, suggerimenti e **gratificando** lo sviluppatore con il loro interesse.

bazaar mode--casi emblematici

- 1998: i dirigenti della Netscape Communication, ispirati dal saggio “The Cathedral and Bazaar” di E.S.R. decidono di aprire i sorgenti di Netscape dando origine al progetto **Mozilla**;
- 1999: Sun Microsystem acquista la tedesca Star Division per trasforamre la suite per ufficio StarOffice nell'equivalente Open Source **OpenOffice.org**;
- 2001: la NaN di Tom Roosendal, produttrice del software freeware **Blender** (modellazione e animazione 3d) fallisce; nel 2002 la Blender foundation fondata da Roosendal raccoglie 100.000 euro dalla comunità OS rendendo il codice disponibile sotto licenza GPL.

Open Source o Free Software?

Nota storica
- Febbraio 1998: E. S. Raymond dopo l'annuncio della Netscape di aprire i sorgenti di Netscape Navigator propone il termine “Open source”;
- poco dopo E. S. Raymond, Bruce Perens e altri fondano la Open Source Initiative (supporto di Linus); opposizione di Stallman;
- Febbraio 1999: B. Perens: “It's time to talk about Free Software again”

“Free Software”:
- termine preferenziale per la FSF e il progetto GNU;
- ambiguità: “free” significa anche “gratuito”;
- enfasi sul concetto filosofico e politico di libertà (libertà del software);
- tende ad essere visto con sospetto dalle società di affari, in quanto viene mal inteso come denotante “anti-commercialismo”.

“**Open Source”**:
- termine preferenziale per l'OSI;
- ambiguità: significato naturale: “a sorgente aperto”, non fa riferimento alle libertà del software;
- enfasi sulle caratteristiche tecniche del modello di sviluppo del software;
- termine accettato con maggiore disinvoltura dalle imprese legate al business.

Terminology wars

“Il termine [Free Software] fa diventare nervosi un sacco di uomini d'affari. Se questo non mi dà minimamente fastidio, tuttavia ora abbiamo un interesse pragmatico a convertire questa gente piuttosto che a fargli le boccacce [thumbing our noses at them]. C'è una seria possibilità di avere dei notevoli successi nel mondo del mainstream business senza compromettere nè i nostri ideali nè il nostro impegno verso l'eccellenza tecnica –così è tempo di assumere una nuova posizione. Abbiamo bisogno di una etichetta nuova e migliore.
[...]
[Linus] sostiene che dobbiamo entrare nel mercato e guidarlo verso i nostri obiettivi, piuttosto che rimanere imbottigliati in una posizione marginale da avversari.”

E. S. Raymond, “Goodbye, "free software"; hello, "open source"”, 8 Febbraio 1998

“Oggi molte persone iniziano ad utilizzare software libero solo per motivi pratici. Ma questa non è la sola cosa che vogliamo! Attirare utenti verso il software libero non è tutto, è solo il primo passo.
Presto o tardi quest i utenti saranno invitati a tornare indietro al software proprietario per qualche vantaggio pratico. Un numero sconfinato di compagnie cerca di offrire tali tentazioni, e perchè gli utenti dovrebbero rifiutare? Rifiuteranno solo se avranno imparato il valore della libertà del software libero per la salvaguardia della stessa libertà. Sta a noi difendere questa idea—e per farlo, dobbiamo parlare di libertà.”

Richard Stallman, “It's Still Free Software”, 16 Febbraio 1998

“La maggior parte degli hacker sa che Free Software e Open Source sono due parole per lo stesso concetto. Sfortunatamente, però, il termine Open Source ha tolto l'enfasi dall'importanza delle libertà implicate nel Free Software. E' tempo di porre rimedio a questo fatto. Dobbiamo rendere chiaro al mondo che queste libertà sono ancora importanti, e che del software come Linux non potrebbe esistere senza.”

Bruce Perens, “It's Time to Talk About Free Software Again”, 17 Febbraio 1999

Riferimenti a risorse impiegate

"Use the source, Luke!"

Riferimenti web-bibliografici:

- "Il Software Open Source e gli standard aperti", M. Sciabarrà, Mc Graw Hill, 2003;
- Files GNU, INTERVIEW, THE-GNU-PROJECT, GNU-LINUX nella directory / usr/share/emacs/21.X/etc inclusi in ogni distribuzione di GNU-Emacs;
- wikipedia.org
- The Cathedral and Bazaar, E. S. Raymond
- The Tao Of Programming, Geoffrey James

Strumenti informatici utilizzati:

- Sistema GNU/Linux-Mozilla-X/GNOME;
- GNU-Emacs;
- OpenOffice.org;