Un salto nel FOSS

Free & Open Source Software

Fabio Ginestri

Dottore in Ingegneria dell'Automazione

Lavori di Gruppo

Insieme di persone che si uniscono per raggiungere un obiettivo

- **Collaborazione:** unione degli sforzi intellettuali/manuali
- **Condivisione:** unione dei risultati ottenuti
- **Comunità:** unione di persone

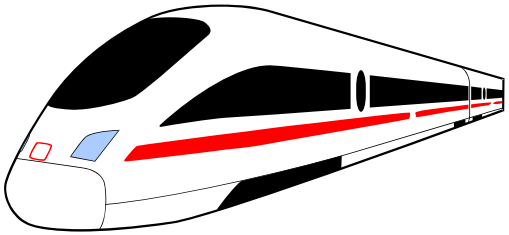
Pro	Contro
Suddivisione del lavoro in lavori più semplici	Difficoltà nell'unire risultati parziali
Più idee nella soluzione di un problema	Difficile mediazione fra idee differenti
Risoluzione di un problema più veloce se si massimizza il parallelismo fra i semilavorati	Tempi persi per intermediazione e disorganizzazione

Vantaggi >> Svantaggi

Lavori di Gruppo

1958 – Massachusetts Institute of Technology (MIT)
Boston (USA)

Nasce il
Tech Model Railroad Club



Gruppo Signals & Power (S&P)

- Le cose acquisiscono significato solo se si scopre il loro funzionamento, cioè se è possibile “metterci le mani”.

Fra di loro si chiamavano **hacker**.

Hacker – Chi è?

hacker s. ingl. [der. (con uso fig.) del verbo (to) hack «tagliare, fare a pezzi» e sim.] (pl. hackers), usato in ital. al masch. – Nel gergo dell'informatica, chi, servendosi delle proprie conoscenze nella tecnica di programmazione degli elaboratori elettronici, penetri abusivamente in una rete di calcolatori per utilizzare dati e informazioni in essa contenuti, per lo più allo scopo di copiarli per trarne vantaggi personali o di renderli pubblici, ma talora anche per manometterli con intenti vandalistici o sabotatori (è detto anche, con espressione connotata negativamente, pirata informatico).

da Treccani.it – *http://www.treccani.it/vocabolario/hacker/*

Un **hacker** (termine coniato negli Stati Uniti che si può rendere in italiano con smanettone) è una persona che si impegna nell'affrontare sfide intellettuali per aggirare o superare creativamente le limitazioni che gli vengono imposte, non limitatamente ai suoi ambiti d'interesse (che di solito comprendono l'informatica o l'ingegneria elettronica), ma in tutti gli aspetti della sua vita.
Esiste un luogo comune, usato soprattutto dai mass media (a partire dagli anni ottanta), per cui il termine hacker viene associato ai criminali informatici, la cui definizione corretta è, invece, "cracker".

da Wikipedia – *http://it.wikipedia.org/wiki/Hacker*

Hacker – Chi è?

English Lesson

- hack*er* = Colui che si occupa di *hack*
- *hack* – nello slang del MIT di quegli anni – era qualsiasi progetto o prodotto che mostrasse innovazione, stile e virtuosismo tecnico, costruito non soltanto per adempiere a uno scopo preciso, ma anche per il piacere di farlo.
  - Veniva anche usato con un significato analogo ad uno scherzo goliardico innocuo, una burla. Oggi è reso dal termine *goof*

Hack ≠ to Hack !!!

Un **hacker** è una persona che si impegna nell'affrontare sfide intellettuali per aggirare o superare creativamente le limitazioni che gli vengono imposte, non limitatamente ai suoi ambiti d'interesse (che di solito comprendono l'informatica o l'ingegneria elettronica), ma in tutti gli aspetti della sua vita.

2011 LINUX DAY ITALIA

Hacker – Evoluzione

Primi anni '60

- Al MIT arriva il primo Computer, un IBM 704
- Accedervi è proibitivo, anche dal punto di vista pratico: i risultati erano disponibili solo dopo un paio di giorni
- Gli hacker – furtivamente di notte – studiavano il computer "a tentativi", arrivando alla fine a scrivere un programma che faceva giocare a scacchi l'IBM 704

Hacker – Evoluzione

... in seguito ...

- Al MIT l'IBM 704 viene sostituito dal supercomputer TX-0
- L'elaboratore è quasi sprovvisto di software
- Gli hacker riuscirono a scrivere tutto il software necessario per il suo utilizzo, perfino un programma per ascoltare musica di Bach

2011 LINUX DAY ITALIA

Hacker – L'etica hacker

Il gruppo comincia a prendere la forma di una comunità, con i suoi concetti e convinzioni: nasce **l'etica hacker**

- *Hands on* (si impara smontando, studiando il funzionamento e riutilizzando ciò che si ha imparato per creare cose nuove)
- Ogni informazione deve essere accessibile a tutti
- Il miglior modo per promuovere lo scambio di informazioni è avere sistemi aperti
- Verrai giudicato per quello che sai fare, indipendentemente dalle credenziali
- Non c'è hacker senza passione

Altri Hacker – Gli Hobbisti

1974 – Ad Albuquerque (New Mexico, USA) il MITS presenta l'**Altair 8800**

- Più che un Microcomputer sembra un giocattolo: agendo sugli interruttori, si facevano accendere/spegnere i LED
- Alimenta nei giovani (in contestazione) appassionati di tecnologia il sogno di costruire un **computer per tutti**
- Il computer non doveva più essere uno strumento elitario alla mercé delle accademie!

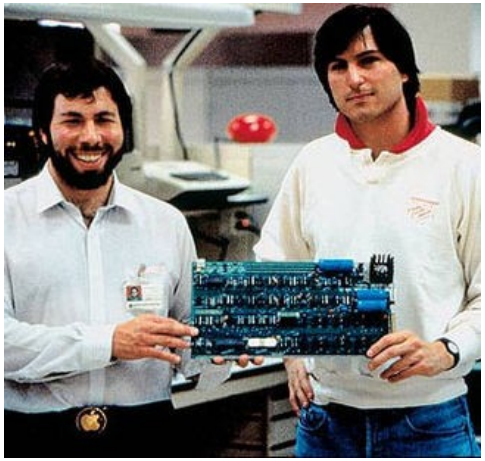
Nascono gli **hobbisti**

2011 LINUX DAY ITALIA

Software proprietario

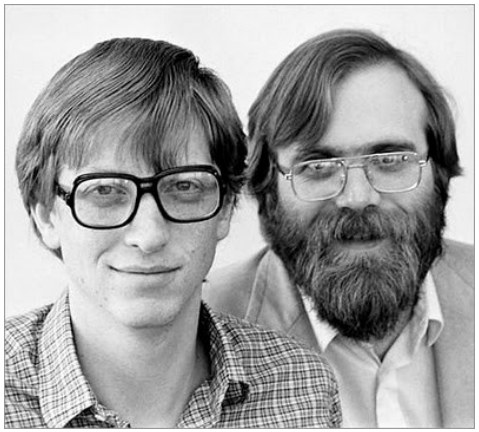
Tra gli hobbisti spiccano alcuni personaggi:

- Steve Wozniak, noto smanettone, incontra Steve Jobs e insieme fondano la **Apple** e costruiscono l'**Apple I**
- Bill Gates fonda la **Micro-soft** (con Paul Allen) e il suo primo prodotto commerciale è l'**Altair BASIC**, una versione di BASIC per l'Altair.



Bill Gates era un personaggio molto singolare:

- Gli hobbisti hanno sempre creato software **per passione e divertimento**
- Bill Gates è convinto che **con il software si debbano fare soldi**
  - Nasce il concetto di **software proprietario**



Micro-soft vs Hobbisti

La Micro-soft contatta la MITS, proponendogli di acquistare il loro Altair BASIC. In realtà loro hanno in mano solamente un BASIC sviluppato su un'architettura Intel 8008 (precedente all'Intel 8080 dell'Altair) emulata su un PDP-10 dell'Università di Harvard.
Il problema fu risolto da Allen, che costruì un codice macchina che permetteva all'Altair di avviare il programma solo nel viaggio in aereo che lo conduceva alla MITS. Il programma fu provato direttamente alla presentazione presso la MITS, con successo (fortunoso).
La MITS firmò un contratto di distribuzione in esclusiva con la Micro-Soft dell'Altair BASIC: essi percepivano 500$ a copia, 150$ se la copia viene venduta insieme con l'Altair.

Membri dell'*Homebrew Computer Club* (il più importante club di hobbisti) trafugano il codice dell'Altair BASIC e ne ridistribuiscono copie gratuite ai suoi membri.
Bill Gates scrisse la famosa **Lettera Aperta agli Hobbisti**, pubblicata sulla newsletter *Computer Notes*.

Micro-soft vs Hobbisti

Mentre Roberts (CEO di MITS) attacca gli hobbisti, definendoli **ladri**, Gates nella sua *Lettera* – a difesa del suo metodo di lavoro – pone queste domande...

> *Chi può permettersi di lavorare a livello professionale senza essere retribuito?*
>
> *Quale hobbista è in grado di dedicare tre anni di lavoro per un programma, sistemare ogni problema, documentarlo adeguatamente per poi distribuirlo gratuitamente?*

-2-

February 3, 1976

An Open Letter to Hobbyists

Bill Gates
General Partner, Micro-Soft

Per gli hobbisti, invece, la questione è un'altra.

- Il BASIC di Gates era in realtà nato in forma gratuita, da informatici che non percepivano alcun diritto d'autore, e veniva migliorato sempre da sviluppatori anonimi, in forma "sociale" e cooperativa e non "professionale".

Nacquero inoltre problemi con l'Università di Harvard, perché – ottenendo l'accesso ufficialmente per progetti didattici – Gates ed Allen avevano utilizzato strutture accademiche per fini personali e commerciali.

Il testo della "Lettera aperta agli hobbisti" è presente all'indirizzo:
http://www.blinkenlights.com/classiccmp/gateswhine.html

2011 LINUX DAY ITALIA

L'ultimo hacker – RMS

1970 – Richard M. Stallman è una matricola dell'Università di Harvard. Cerca e ottiene un lavoro al MIT, dove in quel periodo c'era una fervente e goliardica comunità di informatici di imprese e università di tutti gli Stati Uniti, la cui pratica comune era la condivisione dei programmi. Tutti potevano vedere il codice sorgente di un programma e utilizzare alcune parti di questo per creare un nuovo programma.

"Una cosa vecchia come il computer, proprio come condividere le ricette è antico come il cucinare."
(Richard M. Stallman)

L'ultimo hacker – RMS

1977 – Nel laboratorio di Informatica viene installato un sistema di password per limitare l'accesso ai computer.

- Stallman riesce a convincere un quinto degli utenti ad utilizzare la pressione del tasto “Invio” come password, facendo vedere che un sistema di password era assolutamente inutile (visto che era disponibile il codice sorgente del gestore di password)

“Vedo che hai scelto 'lilo' come password. Ti consiglio di cambiarla con 'invio'. È semplice da usare e sostiene il principio che non ci dovrebbero essere password.”
(Richard M. Stallman)

L'ultimo hacker – RMS

2011
LINUX DAY
ITALIA

La stampante capricciosa

- La Xerox dona al MIT la nuova stampante Xerox 9700. Con parecchi problemi di inceppamento.
- La stampante si inceppa spesso, ma i programmatori della stampante non avevano pensato di fare in modo che arrivasse un avviso ai computer degli utenti.
  - Gli hacker si devono fare tutte le volte due piani per andare a vedere il risultato della stampante, salvo poi vedere che si è bloccata.

2011 LINUX DAY ITALIA

La stampante capricciosa

- Stallman tenta di modificare il driver della stampante, ma si accorge che – diversamente dalle altre volte – non è presente il codice sorgente.
- Incontrando un ricercatore della Carnegie Mellon University che stava lavorando su quella stampante, scopre che egli non poteva fornirgli una copia dei codici sorgenti perché aveva firmato un'accordo di non diffusione con la Xerox.

2011 LINUX DAY ITALIA

Si chiude la comunità

- Con l'interruzione dei finanziamenti della Difesa alla ricerca informatica, il MIT deve ricorrere a cercare fondi presso i privati.
- I nuovi sistemi operativi forniti non sono più liberi: nel 1982 la Digital offre al MIT il mainframe KL-10, con il suo sistema operativo non libero Tenex. Il MIT fu costretto a firmare un accordo di *non-disclosure agreement* anche solo per ottenere una copia eseguibile.
- I meccanismi di sicurezza di Tenex sono talmente fragili che ci si trova costretti a proibire gli accessi in laboratorio agli estranei.

X1296.97

- Contemporaneamente molti hacker lasciano il MIT per andare a lavorare in due società concorrenti: la Symbolics e la Lisp Machines.

Si apre la svolta – GNU

- Richard Stallman si rifiuta di lavorare sul KL-10 a quelle condizioni.
- Soffre per la "distruzione della propria casa". Di quel periodo dirà "Credevo che la mia vita fosse finita".
- **27 settembre 1983 alle 12:35:59 EST (18:35:59 CET)** – sul newsgroup *net.unix-wizards* compare un messaggio di Richard M. Stallman con oggetto *"new UNIX implementation"* le cui prime parole sono **Free UNIX!**
  (messaggio originale su http://www.gnu.org/gnu/initial-announcement.html)
  - Stallman propone un progetto per creare un sistema operativo libero che sia simile a UNIX ma che non sia UNIX

Nasce il GNU Project

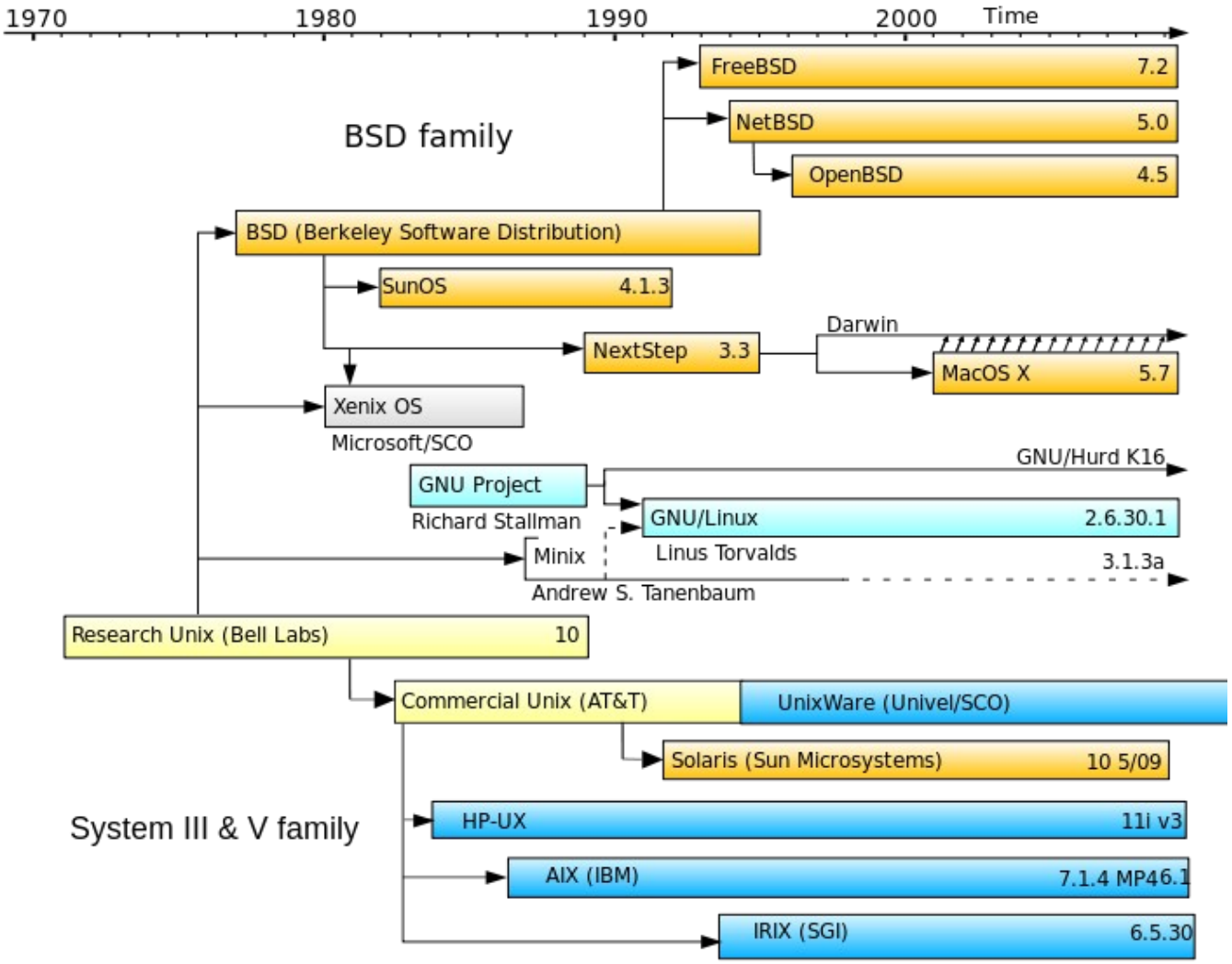
GNU è un acronimo ricorsivo (un vezzo goliardico degli hacker) che sta per

GNU is Not UNIX

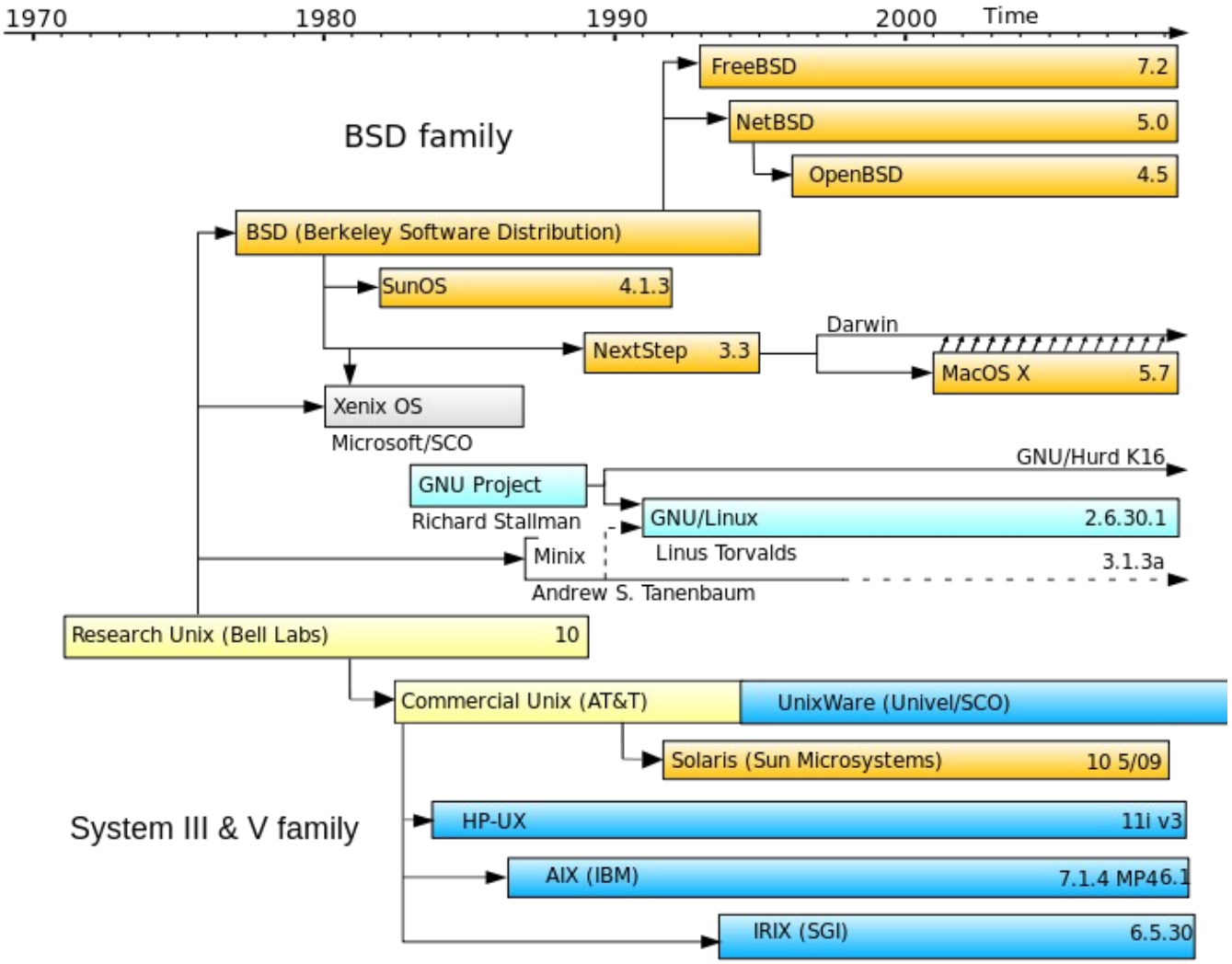
UNIX, questo sconosciuto

- UNIX è un sistema operativo sviluppato da AT&T e Bell Laboratories
- Fra i primi sviluppatori figurano anche Ken Thompson e Dennis Ritchie, gli inventori del linguaggio di programmazione C (servito per una riscrittura completa proprio di UNIX).
  - L'invenzione di C ha consentito a UNIX di essere il primo sistema operativo della storia ad essere in grado di funzionare su più ambienti diversi, ovvero il primo sistema operativo **portabile**
- Caratteristiche di UNIX:
  - Multiutenza
  - Multitasking
  - Gestione della memoria virtuale
  - Portabilità

UNIX e Famiglia



UNIX e Famiglia



Free Software Foundation

- Stallman aveva capito che molti programmatori erano scontenti della commercializzazione del software di sistema: si poteva anche guadagnare bene, ma ciò li costringeva a porsi in conflitto con gli altri programmatori
  - "L'atto di amicizia fondamentale è condividere i programmi"

  (Richard Stallman, "Free Software, Free Society: The Selected Essays Of Richard M. Stallman,GNU Press, Los Angeles 2002)

- Poiché il lavoro sarebbe stato titanico, Stallman decise di attingere e/o riadattare software libero già esistente (riuso del software) e aggiungere parti nuove se necessarie.
- Il primo programma creato da Stallman per GNU è l'editor avanzato GNU Emacs
  - Emacs fece aumentare il consenso attorno al progetto, e consequenzialmente le donazioni di denaro e hardware

- 1985 – Per gestire e sfruttare meglio questo materiale, Stallman ed altri hacker fondano la Free Software Foundation (FSF), un'associazione *no-profit* per lo sviluppo di software libero.

2011 LINUX DAY ITALIA

Copyleft – La GNU GPL

- Fino ad allora, la maniera più semplice per rendere libero un programma era quello di distribuirlo come software di **pubblico dominio**
  - Questo però permetteva a chiunque anche di prendere quel software e imporgli una licenza, rendendolo ad esempio un software proprietario
    - Non era garantita, quindi, la libertà del software
- Per fare in modo che il software – nella sua vita ed evoluzione – rimanesse libero, Stallman aveva bisogno di un nuovo tipo di licenza.
  - "Una forma di jujitsu intellettuale, in cui il sistema legale impostato dai magnati del software proprietario si ritorcesse contro di loro"

Nasce la GNU GPL (GNU General Public License) che introduce il principio del Copyleft

Copyleft

- Il Copyleft in pratica si basa sul concetto di Copyright al contrario:
  - Con il **Copyright** *(Diritto d'autore)* l'autore – oltre ad avere la paternità sul software – ne detiene tutti i diritti su di esso
  - Con il **Copyleft** *(Permesso d'autore)* l'autore mantiene la paternità sul software, ma concede all'utente alcune libertà fondamentali

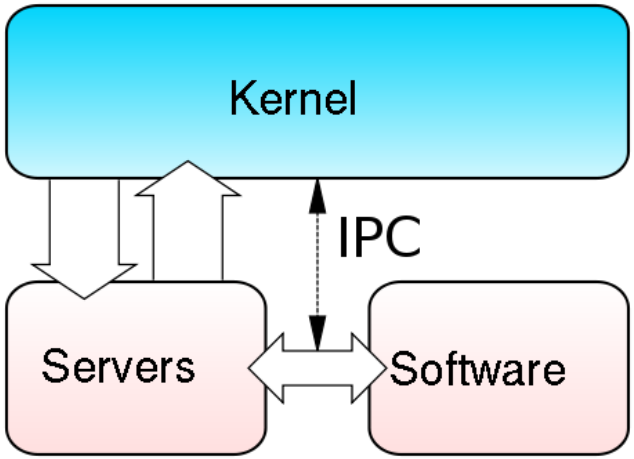
Libertà 0: Libertà di eseguire il programma per qualsiasi scopo.
Libertà 1: Libertà di studiare il programma e modificarlo
Libertà 2: Libertà di ridistribuire copie del programma in modo da aiutare il prossimo
Libertà 3: Libertà di migliorare il programma e di distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio

Il Copyleft si sta piano piano trasferendo dal software ad altre forme di prodotti intellettuali. Per esempio le licenze **Creative Commons** (su cui si basa attualmente Wikipedia) sono licenze di tipo Copyleft.

Manca qualcosa ...

1990 – Il sistema operativo GNU era quasi completo. L'unico componente che mancava era il *kernel*, ovvero lo strato più basso di un sistema operativo, quello che si occupa di distruibuire le risorse hardware ai vari tool del sistema.

La comunità ha in mente un *kernel* (di tipo *microkernel*) molto all'avanguardia – **GNU Hurd** – che però è molto difficile da rendere stabile (non esiste ancora oggi una versione stabile di Hurd). Servers e Kernel, infatti, comunicano fra loro attraverso messaggi asincroni, il che rende maledettamente difficile rendere prevedibile il funzionamento del sistema.



Linus Benedict Torvalds

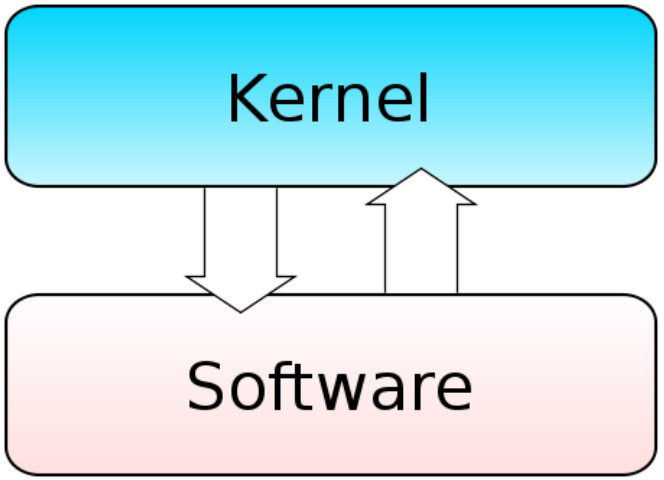
```
Path: gmdzi!unido!fauern!ira.uka.de!sol.ctr.columbia.edu!zaphod.mps.ohio-state.edu!wupost!uunet!mcsun!news.funet.fi!hydra!klaava!
torvalds
From: torva...@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Keywords: 386, preferences
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki
Lines: 20
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
(same physical layout of the file-system (due to practical reasons)
among other things).
I've currently ported bash(1.08) and gcc(1.40), and things seem to work.
This implies that I'll get something practical within a few months, and
I'd like to know what features most people would want.  Any suggestions
are welcome, but I won't promise I'll implement them :-)
                Linus (torva...@kruuna.helsinki.fi)
PS.  Yes - it's free of any minix code, and it has a multi-threaded fs.
It is NOT protable (uses 386 task switching etc), and it probably never
will support anything other than AT-harddisks, as that's all I have :-(.
```

« Sto programmando un sistema operativo (gratuito e solo per hobby, non vuole essere grande e professionale come GNU) per cloni di AT 386(486). È in preparazione da Aprile, e sta iniziando a funzionare. Mi piacerebbe sapere cosa vi piace e non vi piace in Minix, siccome il mio Sistema Operativo gli assomiglia in parte (fra le altre cose, lo stesso layout fisico del filesystem, per ragioni pratiche). Ho convertito la shell bash (v.1.08) e GCC (v.1.40), e sembrano funzionare. Ciò denota che otterrò qualcosa di funzionante in pochi mesi e mi piacerebbe sapere quali funzionalità vuole la maggior parte della gente. Ogni suggerimento è ben accetto, anche se non posso promettervi che lo implementerò. »

Linus Benedict Torvalds

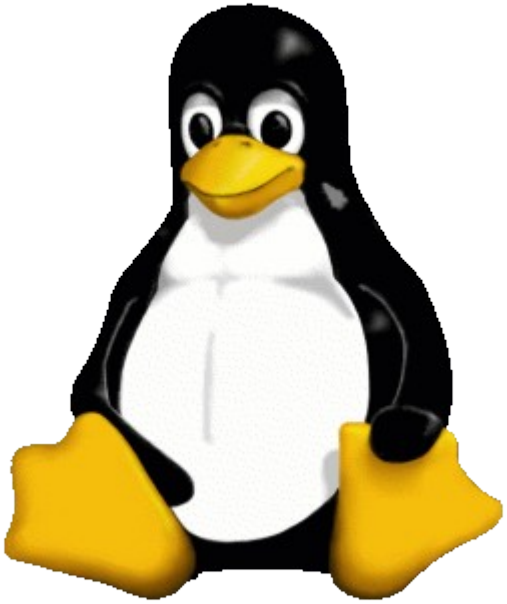
1991 – Linus Benedict Torvalds vuole creare un sistema operativo ("solo per hobby", per utilizzarlo nel suo computer) per sistemi 386 e 486, sulla base del sistema operativo didattico Minix (creato dal suo professore Andrew Tanenbaum).

La struttura scelta da Linus per il kernel del suo sistema era quella del kernel *monolitico*, dove il kernel è un'unica entità indivisibile. Una scelta opposta rispetto a Minix (e sarà causa di discussioni tecniche con Tanenbaum) che, come GNU Hurd, era di tipo *microkernel*.



Linux

- Il messaggio di Linus attira una valanga di persone e in un mese, Linus rilascia in rete la versione 0.01 del suo sistema.
- La cartella in cui era contenuto il sorgente all'interno del suo PC aveva nome Linux (vista l'abitudine, nel mondo Unix, di chiamare qualsiasi cosa con una *x* alla fine, come una legge non scritta). Ma visto che gli sembrava un nome troppo “egocentrico”, sostituì Linux con Freax.
- Tuttavia il gestore del sito FTP che ospitava il sistema per renderlo pubblico in rete cambiò poi autonomamente il nome in **Linux**, e tale rimase per sempre.
- In seguito nacque anche la mascotte **Tux**, un pinguino paffuto dall'aria contenta, creato nel 1996 da Larry Ewing con GIMP.
- Linus Torvalds decide di adottare per Linux la licenza **GNU GPL**.



- Linux si distinse subito per la sua stabilità, efficienza e scarsa concentrazione di bug.

GNU+Linux = GNU/Linux !

- Con Linux rilasciato con licenza GNU GPL vengono quindi ad esistere tutti i “mattoncini” per creare un sistema operativo libero.
- Il sistema operativo non viene alla luce né dal progetto GNU (impegnato a realizzare Hurd) né da Linus (il quale lavorava solo sul suo Linux).
- 1993 – Liberi sviluppatori presero liberamente gli elementi del progetto GNU e colmarono la lacuna costituita dalla mancanza di un kernel utilizzando il kernel di Linux: nascono le distribuzioni.

Volgarmente tali distribuzioni vengono denominate *“sistemi operativi Linux”* o *“distribuzioni Linux”*, ma in realtà sarebbe più giusto riconoscere anche l'importante parte del progetto GNU :-)

Sono distribuzioni GNU/Linux

Distribuzioni GNU/Linux

- Le distribuzioni sono tante e si possono trovare su Internet o in edicola in allegato alle riviste specializzate.
- Principalmente si differenziano per alcuni aspetti
  - Supporto alla distribuzione (comunità di utenti e/o azienda privata)
  - Sistema di distribuzione del software
  - Livello di personalizzazione
  - Disponibilità o meno di elementi di software proprietario (principalmente i codec audio/video per i più diffusi formati, purtroppo proprietari)

Non è un caso che – se tutte le distribuzioni siano costituite da FOSS praticamente al 100% – la *Free Software Foundation* considera distribuzioni completamente libere solamente 9 distribuzioni, fra l'altro poco note.
http://www.gnu.org/distros/free-distros.html

Fraintendimenti

- La confusione potrebbe nascere facilmente, in quanto in italiano la parola *free* è traducibile in entrambi i modi, e la generale gratuità delle distribuzioni potrebbe trarre in inganno
- Il *free software* è mal visto dal mondo dell'industria del software
  - La gratuità viene vista come qualcosa in netto contrasto con la logica industriale del profitto
  - La libertà del software viene vista come qualcosa di poco "controllabile" e di sovversivo (ricordate la *"Lettera aperta agli hobbisti"*?)

Nonostante alcuni esempi di software a licenza libera di grande successo (come l'Apache Web Server) o di aziende che sviluppano e/o danno supporto a software libero, le grandi aziende del settore nutrono ancora scetticismo al mondo del software libero.

Cattedrali e Bazaar

27 maggio 1997 – Eric S. Raymond presenta il saggio sullo sviluppo del software *"La Cattedrale e il Bazaar"*, nel quale vengono messi a confronto due diversi modelli:

- Nel modello a **cattedrale** il programma viene realizzato da un numero limitato di "esperti", con una gerarchia ben definita. Le revisioni si susseguono con relativa lentezza e gli sviluppatori cercano di rilasciare programmi il più possibile completi e senza bug.
- Nel modello a **bazaar** il codice sorgente della revisione in sviluppo è disponibile liberamente. Lo sviluppo è decentralizzato e non esiste una rigida suddivisione dei compiti, un programmatore di buona volontà può modificare e integrare qualsiasi parte del codice.

"La Cattedrale e il Bazaar" è generalmente considerato il manifesto del movimento *open source*.

La traduzione italiana de *"La cattedrale e il bazaar"* si può trovare all'indirizzo http://it.wikisource.org/wiki/La_cattedrale_e_il_bazaar

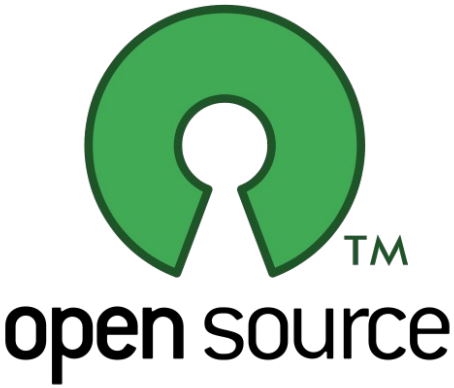
2011
LINUX DAY
ITALIA

Open Source

- Bruce Perens (già Project Leader della distribuzione Debian GNU/Linux) insieme ad Eric S. Raymond e ad altri hacker coniano per la prima volta il termine **Open Source**. A Perens e Raymond si deve invece la scrittura della *Open Source Definition*, una sorta di metalicenza. La prima bozza di questo documento deriva direttamente dalle *Debian Free Software Guidelines*.
- Lo scopo è vincere le resistenze dell'industria del software verso il termine *Free Software*
- Sono considerate come *Open Source* diverse licenze: le prime furono la GNU GPL (in realtà è stata la licenza a cui ci si è ispirati) e la BSD (altra licenza libera) che esisteva prima di Linux.

Open Source Definition

Per essere definito come Open Source, il software deve soddisfare questi criteri:

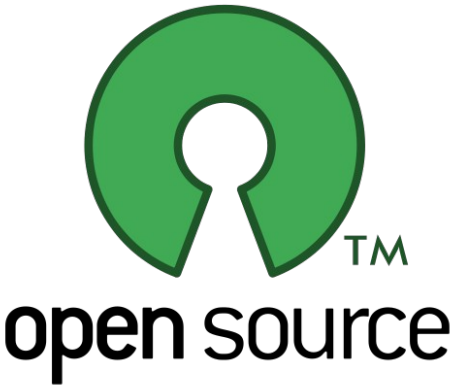


- **Libera redistribuzione:** la licenza non può limitare alcuno dal vendere o donare il software che ne è oggetto, come componente di una distribuzione aggregata, contenente programmi di varia origine.
- **Codice sorgente:** il programma deve includere il codice sorgente e ne deve essere permessa la distribuzione sia come codice sorgente che in forma compilata. Laddove alcune forme di un prodotto non siano distribuite con il relativo codice sorgente, deve essere chiaramente indicato il modo per ottenerlo, ad un costo non superiore ad una ragionevole spesa di distribuzione, preferibilmente scaricandolo gratuitamente da Internet.
- **Prodotti derivati:** la licenza deve permettere modifiche e prodotti derivati, e deve permetterne la distribuzione sotto le stesse condizioni della licenza del software originale.
- **Integrità del codice sorgente originale:** la licenza può impedire la distribuzione del codice sorgente in forma modificata, a patto che venga consentita la distribuzione dell'originale accompagnato da "patch", ovvero file che permettono di applicare modifiche automatiche al codice sorgente in fase di compilazione. La licenza deve esplicitamente permettere la distribuzione del software prodotto con un codice sorgente modificato. La licenza può richiedere che i prodotti derivati portino un nome o una versione diversa dal software originale.

La traduzione italiana della *"Open Source Definition"* si può trovare all'indirizzo http://it.wikisource.org/wiki/La_definizione_di_Open_Source

Open Source Definition

Per essere definito come Open Source, il software deve soddisfare questi criteri:



- **Discriminazione contro persone o gruppi:** la licenza non deve discriminare alcuna persona o gruppo di persone
- **Discriminazione per campo d'applicazione:** la licenza non deve impedire di far uso del programma in un ambito specifico
- **Distribuzione della licenza:** i diritti allegati a un programma devono essere applicabili a tutti coloro a cui programma è redistribuito
- **Specificità ad un prodotto:** i diritti allegati al programma non devono dipendere dall'essere il programma parte di una particolare distribuzione di software.
- **Vincoli su altro software:** la licenza non deve porre restrizioni su altro software distribuito insieme al software licenziato
- **Neutralità rispetto alle tecnologie:** la licenza non deve contenere clausole che dipendano o si basino su particolari tecnologie o tipi di interfacce.

La traduzione italiana della *"Open Source Definition"* si può trovare all'indirizzo http://it.wikisource.org/wiki/La_definizione_di_Open_Source

Da Netscape a Mozilla

1998 – Netscape, la società che sviluppa il browser Netscape Navigator, si trova in cattive acque in quanto non riesce a contrastare la spietata concorrenza di Microsoft che distribuisce il suo browser Microsoft Internet Explorer insieme al suo sistema operativo Microsoft Windows 98 (anzi, sistema operativo e browser web sono strettamente correlati l'un l'altro). Qualche dirigente di Netscape ha letto *"La cattedrale e il bazaar"* e pensa che sia la soluzione al problema dell'azienda, che deve contemporaneamente abbassare i prezzi dei suoi prodotti mantenendo però la qualità dei suoi servizi, cosa molto difficile da ottenere con il personale disponibile in azienda.

Il documento interno alla Netscape che illustra perché l'azienda dovrebbe rilasciare il codice sorgente ha come titolo *"Netscape Source Code As Netscape Product"* (ovvero *"Codice Sorgente Netscape Come Prodotto Netscape"*): il codice sorgente non deve quindi più essere qualcosa con cui si costruisce il prodotto, bensì **deve essere il codice stesso il prodotto dell'azienda**, che può essere utilizzato dalle altre persone.

Da Netscape a Mozilla

Gennaio 1998 – L'idea ha successo e Jim Barksdale (CEO di Netscape), decide di rilasciare il codice sorgente di Netscape Navigator. Il rilascio avviene due mesi più tardi, il 13 marzo 1998, e contemporaneamente Netscape Communicator viene rilasciato gratuitamente.

Per sviluppare il codice sorgente di Netscape nasce il Mozilla Project, e da qui la Mozilla Foundation. La Mozilla Foundation oggi si occupa dello sviluppo della Mozilla Suite (oggi nota come Seamonkey), e in seguito altri software come Mozilla Firefox e Mozilla Thunderbird.

Da Sun e LibreOffice

2000 – La Sun Microsystem – creatrice del linguaggio *object-oriented* Java) – acquisisce la software house tedesca StarDivision, entrando in possesso della suite d'ufficio proprietaria StarOffice. Decide di rilasciarne il codice sorgente con licenza GNU LGPL e SISSL: nasce **OpenOffice.org**, il cui codice sviluppato dalla comunità va a migliorare la versione proprietaria, commercializzata insieme a contenuti aggiuntivi e al servizio di assistenza.

2006 – Per contrastare l'ascesa di MS.Net, linguaggio *object-oriented* di Microsoft in grossa ascesa, Sun decide di rendere Open Source anche la piattaforma software **Java** con licenza GNU GPL.

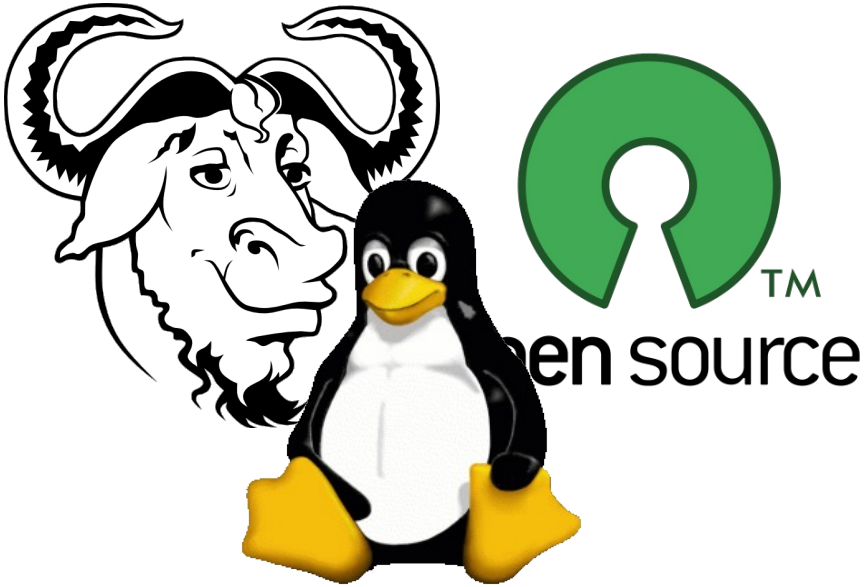
2010 – Sun Microsystem viene acquisita da Oracle, che tergiversa sul supporto al progetto OpenOffice.org. Il timore che il progetto venga in qualche modo ostacolato e/o chiuso spinge la comunità di sviluppatori a svincolarsi da Oracle, creando **The Document Foundation** e continuando a sviluppare il codice di OpenOffice.org nel nuovo software **LibreOffice** (sostenuto da Google, Red Hat, Canonical e Novell), distribuendolo totalmente con licenza GNU LGPL.

2011 – Rimasta sola, Oracle dona il codice di OpenOffice.org a Apache Software Foundation.

The Document Foundation permette di avere una suite d'ufficio libera (LibreOffice) e un **formato di documentazione libero** e standard ISO (Open Document Format, ODF): questo consente di rendere i nostri documenti sempre leggibili ed utilizzabili, a differenza dei formati di documento proprietari.

Tre modi di vedere il FOSS ?

- Perché spesso per rappresentare un'ideale non basta un solo punto di vista
  - Stallman ritiene il software proprietario il male assoluto “che va debellato”
  - Torvalds e il movimento Open Source ritengono che software libero e proprietario possano coesistere
- Perché spesso per concretizzare un'idea occorrono figure diverse che concorrano per il raggiungimento dello scopo
  - Stallman è il filosofo, pensa alla libertà
  - Torvalds è il tecnico, pensa all'efficienza
  - Perence e Raymond i fautori di un modello di sviluppo “industriale” aperto e libero
- Perché la libertà è anche pensare differente
- Perché la libertà è discutere su come ottenere le scelte migliori per il bene della comunità



… e la Comunità ?

- Ciò che è importante è creare soluzioni che siano utili alla collettività
  - Creare qualcosa che diventi un bene comune
- È importante che le istituzioni pubbliche si interessino e adottino il FOSS
  - È un bene collettivo, una sorta di bagaglio culturale
  - Molto del FOSS è gratuito, un risparmio per la collettività
  - Il FOSS richiede generalmente meno risorse tecniche rispetto ai sistemi operativi proprietari, il ché aumenta il tempo di vita medio di un PC e riduce i RAEE
  - Mettendo in luce le differenze fra le licenze, conduce ad un uso consapevole del software e riduce la pratica delle copie non autorizzate di software properietario
  - Riduce la piaga – e il costo – sociale del Digital Divide
- È importante che la collettività si interessi al FOSS
  - Fa crescere culturalmente la società stessa e la rende in grado di aiutarsi da sé
    - Libera la cultura informatica dal dominio di poche aziende oligopolistiche
    - Trasforma l'informatica da elemento alienante in elemento socializzante (I vari LUG e FSUG ne sono una testimonianza)

2011 LINUX DAY ITALIA

Fonti ...

- J.T.S. Moore, “Revolution OS – Rivelazioni di hacker”, 2001 – http://www.revolution-os.com/
- Roberto Bosio, “Guida al Software Libero” - Editrice Missionaria Italiana, Bologna – ISBN 8830714194
- Wikipedia, l'enciclopedia libera – http://it.wikipedia.org/

... e approfondimenti

- GNU Operating System – http://www.gnu.org/home.it.html
- Free Software Foundation – http://www.fsf.org/ | FSF Europe – http://fsfe.org/index.it.html
- The Linux Foundation – http://www.linuxfoundation.org/
- Open Source Initiative – http://www.opensource.org/
- Daniele Giacomini, “Appunti di Informatica Libera”, 2008 - http://a2.pluto.it/
- Wikipedia, “Storia del Software Libero” – http://it.wikipedia.org/wiki/Storia_del_software_libero
- Videoblog di Christian Biasco - http://biasco.ch/videoblog

Un salto nel FOSS

Free & Open Source Software

Fabio Ginestri
Dottore in Ingegneria dell'Automazione

Fine