



Benvenuti alla prima lezione per la provincia di Massa Carrara del **progetto TAG** (Toscana Area Giovani).

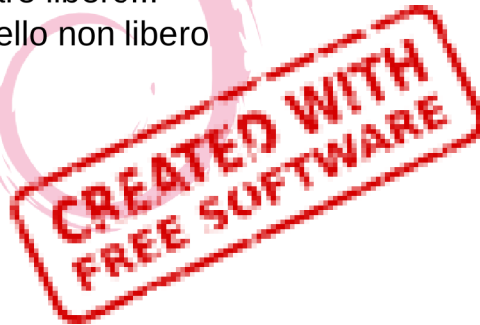
Questo progetto, promosso principalmente dall' **UPI Toscana** (Unione delle Province Italiane) in collaborazione con la **Fondazione Sistema** Toscana ed il **Corecom**, promuove la diffusione fra pari di conoscenze inerenti il mondo della tecnologia dell'informazione.

In questo incontro parleremo di software libero e faremo un'introduzione a Debian, il sistema operativo basato su Linux libero per eccellenza.

Prima di iniziare vorrei ringraziare l'UPI Toscana, l'amministrazione dell'assessorato alle politiche giovanili della Provincia di Massa Carrara, il Corecom e la Fondazione Sistema Toscana, grazie ai quali è stato possibile far nascere ed organizzare questo progetto.

Di cosa parleremo

- Software libero e non
- Un po' di storia sul software libero...
- ... e un po' di storia di quello non libero
- Impariamo a scegliere
- Da dove partire



www.filippomicheletti.it
filippomicheletti@liberainformatica.it
<http://www.facebook.com/filippomicheletti>

Perché siamo qui?



*La conoscenza fa la differenza: chi sa, può scegliere.
E fare.*

Filippo Micheletti - Progetto TAG

3

Innanzitutto è bene fare una precisazione del motivo per cui siamo qui a parlare di software libero.

Nell'ora che segue non faremo pubblicità né anti-pubblicità per convincere qualcuno ad utilizzare un tipo di software piuttosto che un altro.

Non esiste infatti un software in assoluto migliore o peggiore, ma esiste sicuramente un software più adatto alle proprie esigenze, in termini di prestazioni, in termini economici, o, perché no, etici.

L'importante è conoscere ciò fra cui si può scegliere per essere in grado di farlo liberamente.

Lo scopo di questa chiacchierata è proprio questo: mettervi a conoscenza dell'esistenza di un'alternativa, pregi e difetti inclusi, in modo che possiate fare la scelta per voi migliore.

Tipologie di software

Software libero

Software semi-libero

Software proprietario

“La libertà non è fare ciò che si vuole, ma non fare ciò che non si vuole.”

Filippo Micheletti - Progetto TAG

4

Cos'è un software?

Formalmente un **software** è un'insieme di istruzioni per una determinata macchina.

Le istruzioni, scritte in "linguaggio macchina" non sono comprensibili dall'uomo per cui, a monte di esse, c'è un codice scritto in un preciso linguaggio di programmazione: è l'operazione di **compilazione** che "traduce" le istruzioni in linguaggio macchina.

Un software ha quindi una duplice natura: quella di **codice sorgente** e quella di **codice eseguibile**.

Software libero

Il **software libero** è solo quello che fornisce il permesso per chiunque di **utilizzarlo**, **copiarlo**, **modificarlo** e **ridistribuirlo**, in forma originale o dopo averlo modificato, sia gratuitamente che per trarne profitto.

Comunemente ci si riferisce a queste condizioni necessarie a far sì che un software possa essere considerato libero come alle 4 libertà del software libero, che evidentemente può essere considerato tale solo se viene messo a disposizione assieme al codice sorgente (*"if it's not source, it's not software"*).

Software semilibero e software proprietario

Esistono poi altri gradi di semi-libertà, ossia casi in cui il software viene distribuito con limitazioni riguardo ad uno o più dei punti suddetti.

Ci si riferisce a software **semi-libero** quando questo è distribuito con i sorgenti, permette uso, copia, modifica e redistribuzione per qualsiasi scopo purché senza fine di lucro.

Il **software proprietario** è invece tutto quel software di cui è distribuito solo il codice che per essere utilizzato richiede il possesso di particolari privilegi a pagamento (licenze, chiavi hw ecc).

Esistono alcune varianti semi-libere del software proprietario alle quali si fa riferimento comunemente come **software shareware** quando è concesso un utilizzo gratuito temporaneo (periodo di prova) e/o limitato del software e **software freeware** quando ne è concesso un utilizzo gratuito illimitato.

E' nato prima l'uovo o la gallina?



Ossia: perché esiste il software proprietario?

... tutta "colpa" dei sistemi operativi

Filippo Micheletti - Progetto TAG

5

Nonostante le nostre abitudini ci portino a pensare "ovvio, per trarne profitto" la risposta a questa domanda non è affatto scontata.

Fino alla fine degli anni settanta infatti, anche se in misura via via decrescente, la componente principale e più costosa di un computer era l'hardware.

Dato che non esisteva ancora nemmeno il concetto di sistema operativo che rendesse possibile astrarre la progettazione del software dall'architettura hardware per cui veniva creato, il software veniva (faticosamente) scritto "ad hoc" per la macchina su cui doveva girare e poteva essere eseguito solo su di essa.

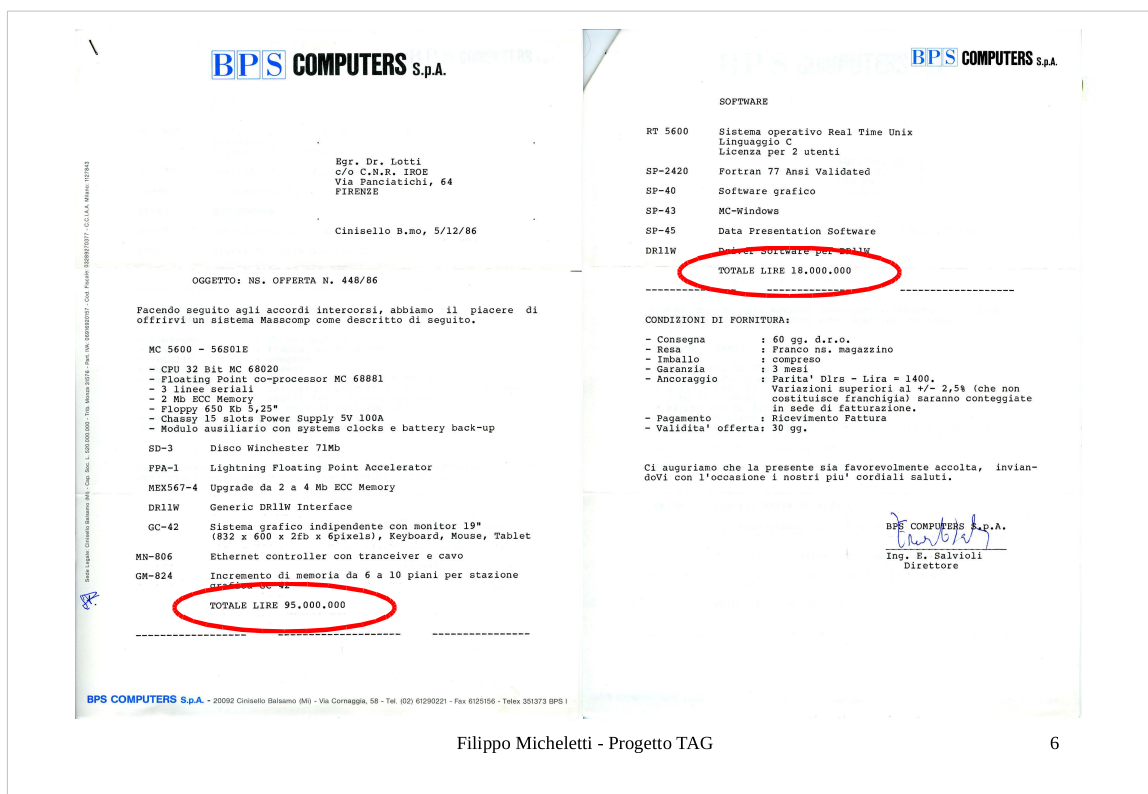
La scrittura del software inoltre non era affatto un'operazione tutto sommato alla portata di tutti come la conosciamo oggi: si trattava di una pratica riservata a pochi per le conoscenze richieste (la scrittura avveniva a basso livello per cui era necessario conoscere a menadito la macchina per la quale si stava creando il software), la tecnica utilizzata (basti pensare ai supporti utilizzati per memorizzare il programma, come le schede perforate), ed anche per i mezzi necessari (per provare un programma era ovviamente necessario poter accedere ad un calcolatore).

Infine in quel periodo storico la produzione di hardware era in mano a pochissime grandi aziende (DEC Digital Equipment Corporation, IBM, Xerox), l'evoluzione tecnologica lenta (un determinato modello aveva una vita di diversi anni) e i costi delle macchine altissimi (centinaia di migliaia di dollari).

Da ciò si capisce come l'acquisto di un computer fosse un investimento molto importante e una delle variabili determinanti nella scelta fra i modelli disponibili era quella della disponibilità di software già scritto.

E' chiaro allora che fosse nell'interesse di qualsiasi produttore di computer incentivare la creazione di software libero per facilitarne la diffusione, col fine di rendere più concorrenziali i propri prodotti.

In realtà in quel momento non era nemmeno necessario discriminare fra software libero e proprietario: il software era essenzialmente libero e spesso veniva fatta pagare la dotazione software di una macchina in termini di costo per preinstallare quel determinato software.



La possibilità di scrivere codice indipendente dall'hardware era già arrivata con i **linguaggi di programmazione**.

Questi infatti permettono di astrarre a livello di procedure la progettazione del software dall'hardware sottostante: il codice, scritto in un linguaggio "universale" è di per sé indipendente ed è l'operazione di compilazione a "tradurlo" in codice eseguibile specifico per quella determinata macchina.

La vera svolta però fu l'introduzione del **sistema operativo**.

Il sistema operativo infatti permette (fra le altre cose) di avere a disposizione un ambiente "virtuale" standardizzato all'interno del quale poter eseguire un programma senza doversi curare di tutti gli aspetti legati alla gestione dell'hardware che sono appunto delegati al SO.

L'introduzione di questo concetto rivoluzionario permise di avere programmi sempre più portabili, in quanto con lo stesso sistema operativo venivano equipaggiati diversi modelli di hardware.

La presenza di sistemi operativi funzionanti per macchine di differenti produttori hardware ampliava ulteriormente le possibilità di usare lo stesso codice in modo relativamente indipendente dall'hardware usato.

Inoltre l'avanzamento tecnologico permetteva un graduale abbattimento dei costi dell'hardware, rendendo l'acquisto di un computer sempre più accessibile fino ad arrivare all'introduzione del primo PC (ufficialmente l'Apple II, nel 1977, con il quale venne coniato il termine Personal Computer, in realtà l'Olivetti MD 800, del 1975, realizzava già il concetto di computer per uso personale) ed ampliando sempre più la differenziazione dell'offerta disponibile sul mercato.

In questo processo l'attenzione si spostò molto rapidamente sul software e dato che importanza=guadagno nacque il software proprietario, che ha fatto sbocciare migliaia e migliaia di software-house, alcune delle quali sono ad oggi fra le aziende col fatturato più alto del mondo.

Un po' di storia...



I primi sistemi operativi



UNIX

- si fa in modo che ogni programma svolga uno o pochi compiti ma bene;
- ci si aspetta che l'output di un programma diventi l'input di un altro;
- si sviluppa software con l'idea che esso verrà provato subito: non si esiti a condividere il programma;
- si usano degli strumenti appositi nella programmazione senza perdersi ogni volta nel "reinventare la ruota".

sistemi UNIX-like e UNIX standards

[The Open Software Foundation on Wikipedia \(EN\)](#)
[The Open Group on Wikipedia \(EN\)](#)
[The Open Group \(EN\)](#)
[UNIX \(EN\)](#)



Filippo Micheletti - Progetto TAG

UNIX[®]
Celebrating 40 years uptime

7

Se non si considerano i primi sistemi per il controllo di sistemi batch, come il Fortran Monitor System o l'IBSYS, il primo sistema operativo può considerarsi l' OS/360 della IBM, che introdusse l'implementazione dell'astrazione hardware, con un primo tentativo di multiprogrammazione (1964).

Più o meno contemporaneamente, in ambito accademico, nasceva CTSS del MIT, il primo sistema multiprogrammato, dal quale sarebbe partito il progetto Multics, in collaborazione con GE e Bell Laboratories.

In quegli anni uscivano i primi elaboratori "economici", come il PDP-1, per i quali furono sviluppati numerosi sistemi operativi.

Ma la vera rivoluzione fu **UNIX**. A partire dal 1969 un gruppo di ricercatori della **AT&T** presso i **Bell Laboratories**, tra cui erano presenti **Ken Thompson** (che lavorò anche al progetto Multics), **Dennis Ritchie** e **Douglas McIlroy**, diedero vita ad un sistema operativo con lo scopo di eseguire un programma chiamato "Space Travel", per la simulazione dei movimenti del sole e dei pianeti e la simulazione dell'atterraggio di una navicella spaziale: nasceva UNIX.

UNIX, scritto interamente in **C**, il linguaggio di programmazione sviluppato dallo stesso Dennis Ritchie, nasce come **sistema multiutente, multiprogrammato, portabile ed aperto**, dalle macerie del progetto Multics.

Le idee su cui è nato UNIX, che vedete citate nella slide, riassumono tutto ciò che sta alla base della buona programmazione e fanno capire meglio di ogni altra cosa quanto chiare fossero fin dall'inizio le idee di Thompson, Ritchie e McIlroy.

Con UNIX nasce infatti molto di più del primo vero sistema operativo...

AT&T non volle avere alcuna royalty sull'utilizzo e la modifica di UNIX, permettendo che il codice sorgente di UNIX venisse distribuito gratuitamente per fini di studio presso le università di tutto il mondo.

Per risolvere i problemi di condivisione del codice, si introdusse per la prima volta la copia di file tra sistemi dislocati in parti diverse del mondo attraverso la linea telefonica. Il sistema venne chiamato UUCP (Unix to Unix CoPy) e nonostante venne ben presto surclassato dalla nascente ARPANET (in seguito Internet) costituì di fatto uno degli esperimenti da cui è nata l'attuale Internet.

L'interesse commerciale da parte degli sviluppatori dei sistemi operativi derivati da UNIX (ricevuto **GRATUITAMENTE!!!**) ha portato a guerre di potere legate ai diritti di autore sul codice, battaglie fratricide spesso senza fine. Da UNIX nacquero così vari sistemi derivati, detti "UNIX-like", fra i quali **BSD** (Berkley Software Distribution) (1978), successivamente scissa per motivi legali in FreeBSD, NetBSD ed OpenBSD, e dalla quale hanno avuto origine SunOS e NextStep, "padre" di MacOS, **Minix**, **Xenix** (Microsoft), ed altri progetti di minor fama.

Attorno al modello UNIX si sono definiti due standard fondamentali: il **linguaggio C** (ISO/IEC 9899) e la definizione del sistema «UNIX» standard, noto come **POSIX** negli anni 1990 (Portable Operating System Interface uniX), ad opera della **OSF** (Open Software Foundation), poi divenuta **The Open Group**, che le ha evolute nelle **SUS** negli anni 2000 (Single Unix Specification).

Un po' di storia...

GNU & FSF

THE
Open
GROUP



*GNU project
Free Software Foundation
Richard Stallman on Wikipedia (EN)*

TAG
TOSCANA
AREA
GIOVANI



Filippo Micheletti - Progetto TAG

8

Verso la metà degli anni 80 in un rinnovamento del parco stampanti del MIT venne adottato un nuovo driver proprietario per una stampante XEROX che non implementava più la funzionalità di segnalazione automatica della carta inceppata, molto utile quando una stampante è condivisa da molte persone.

Fra i programmatori del MIT vi era un certo **Richard Stallman**, al quale non andava giù la tendenza che si stava diffondendo da parte delle aziende di assumere i migliori programmatori impiegati dai centri di ricerca delle università per impiegargli nella scrittura di software proprietario.

Il driver della XEROX fu la goccia che fece traoccare il vaso: Stallman si fece portavoce del problema e si attivò per combattere contro la privatizzazione del software, fondando nel 1985 la **FSF** (Free Software Foundation), con il preciso scopo di promuovere e sostenere la creazione e diffusione di software libero.

L'idea di Stallman si concretizzò presto con la creazione della licenza **GPL** (General Public License), una licenza a protezione del cosiddetto **copyleft**, una sorta di duale del copyright che punta a riconoscere diritti di paternità più che d'autore, secondo i principi del software libero.

In realtà la FSF non è la prima organizzazione a nascere con questo scopo, un anno prima era nato infatti il progetto **GNU** (GNU is Not Unix), con lo scopo di creare un sistema operativo e un insieme di software che fosse basato su (e compatibile con) Unix, ma completamente indipendente e libero.

Non fatevi ingannare dall'aspetto di Stallman: negli ultimi 25 anni ha ricevuto centinaia di prestigiosissimi riconoscimenti internazionali, fra cui 8 tra lauree e PhD *honoris causa* (uno anche dall'università di Pavia)!

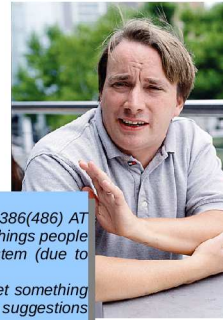
Un po' di storia...

Linux

*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 (torvalds [at] kruuna.helsinki.fi)
PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).*

—Linus Torvalds

[Linux History \(EN\)](#)
[Linus Torvalds on Wikipedia \(IT\)](#)
[Tux on Wikipedia \(IT\)](#)
[The Linux Foundation \(EN\)](#)



Filippo Micheletti - Progetto TAG

9

Linux nasce all'inizio degli anni 90 come un progetto personale di studio delle funzionalità di multiprogrammazione dei microprocessori x86-32 da parte di **Linus Torvalds**, all'epoca uno studente all'università di Helsinki, in Finlandia.

Inizialmente il lavoro di Linus Torvalds si basava su un sistema Minix, riscrivendo il kernel e adattando solo successivamente il compilatore e i programmi sviluppati dal progetto GNU.

Il progetto di Torvalds concentrò ben presto una grossa comunità di sviluppatori in tutto il mondo.

Il testo il slide riporta il messaggio originale postato il 25 Agosto 1991 da Torvalds sul newsgroup "comp.os.minix" della Usenet dell'università di Helsinki: il resto è storia...

Il nome Linux, a dispetto dell'assonanza con il nome di Torvalds, è da attribuire a Ari Lemke, l'amministratore che rese per primo disponibile Linux su Internet via FTP. In particolare Linux era il nome della directory in cui risiedevano i file del nuovo sistema operativo. Il nome scelto da Torvalds era Freax, una combinazione tra "free", "freak" e "x", per indicare la caratteristica di un sistema Unix-like.

Nel 1996, ad opera di Larry Ewing, nasce **Tux** (Torvalds UniX), il pinguino mascotte ufficiale di Linux.

Nel 2007 viene fondata la **The Linux Foundation**, un'organizzazione non-profit che sostiene lo sviluppo del kernel Linux

Un po' di storia...



Debian

DFSG

Open Source

Debian.org
[Debian project history \(EN\)](#)
[DFSG \(EN\)](#)
[Open source on Wikipedia \(IT\)](#)
[Open Source Initiative \(EN\)](#)



Filippo Micheletti - Progetto TAG

10

Nel 1993 **Ian Murdock** fonda il progetto Debian, con lo scopo di realizzare una distribuzione GNU/Linux libera e accurata.

Debian è la fusione delle parole **Deb** e **Ian**, con cui Murdock volle dedicare il progetto alla allora fidanzata Debra Lynn.

Il progetto Debian prende il via ufficialmente con la pubblicazione del "Manifesto Debian", con il quale Murdock pubblica le proprie intenzioni presentando il progetto che ha in mente.

Successivamente, **Bruce Perens** stila la bozza di ciò che attualmente costituisce le «linee guida», per stabilire cosa sia da intendere «software libero» e cosa invece non rientri in questa categoria, ai fini della distribuzione stessa. Tali linee guida sono note con la sigla **DFSG** (Debian Free Software Guidelines).

Open Source

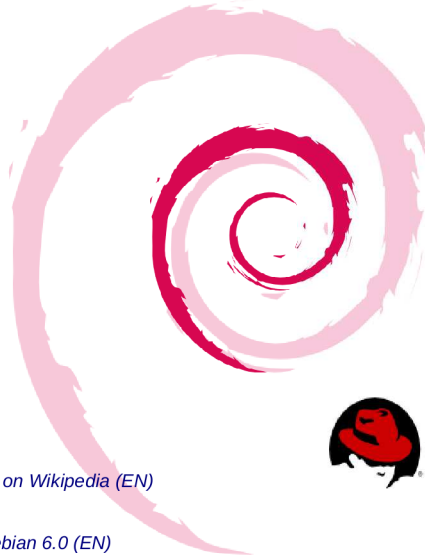
Sulla scia del progetto Debian, nel 1998 viene ufficializzata la definizione di Open Source, in seguito alla quale nasce anche un'organizzazione non-profit a cui aderiscono i principali progetti basati su software libero, con lo scopo di diffondere l'educazione all'open source, la **Open source Initiative**.

... fino ai “giorni nostri”...



Distribuzioni

Software



[Linux distributions timeline on Wikipedia \(EN\)](#)
[Distrowatch \(EN\)](#)
[Debian childs \(EN\)](#)
[Supported hardware by Debian 6.0 \(EN\)](#)

Filippo Micheletti - Progetto TAG

11

Il lavoro e la fatica delle persone di cui abbiamo parlato sono stati il seme per lo sviluppo di un'infinità di progetti che spaziano veramente in ogni ambito e direzione.

Qui abbiamo parlato di Linux in generale infatti enel particolare di Debian, il primo vero sistema operativo basato su Linux nato con un progetto preciso ed organizzato, ma certamente non l'unico.

Quasi contemporaneamente a Debian infatti sono nati altri 2 progetti basati sul kernel Linux: **Slackware** e **Red Hat**.

Da Debian, Slackware e Red Hat si sono poi sviluppati molti progetti che hanno dato vita a distribuzioni basate su di esse ma con variazioni orientate sia ad ambienti e caratteristiche particolari (ad es. per macchine server o per sistemi embedded), sia più semplicemente a creare qualcosa di diverso, magari perché non si dividevano alcuni punti del progetto base.

A tutte le distribuzioni derivate da Debian, alcune delle quali sono divenute decisamente più popolari di Debian stesso, come **Ubuntu**, sono dette “**Debian child**” in quanto, per un verso o per un altro, sono figlie, nipoti, bisnipoti di Debian.

Contemporaneamente si è sviluppata un'enormità di software applicativo per sistemi basati su Linux in ogni campo: dagli applicativi per ufficio a quelli di calcolo scientifico, dalla musica ai software specifici per la riabilitazione del linguaggio o l'apprendimento... ma di questo parleremo più avanti.

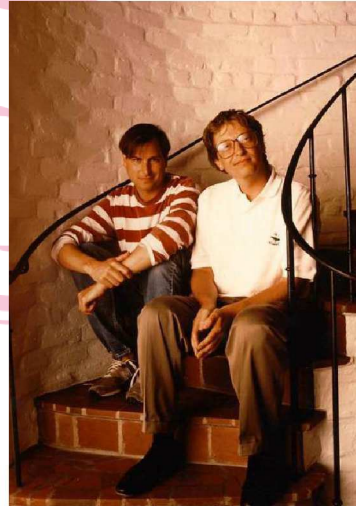
Giusto per rendere un'idea ad oggi esistono oltre 600 distribuzioni basate su Linux e rilasciate ufficialmente e nell'attuale progetto stable di Debian sono disponibili oltre 23000 pacchetti software ufficiali.

E nel frattempo?

Apple Computers



Microsoft Inc.



Filippo Micheletti - Progetto TAG

12

Come abbiamo detto a partire dalla metà degli anni 70 il software proprietario divenne un business che diede vita a moltissime software-house, ma l'emblema di questo mondo è rappresentato sicuramente da due grandi aziende che con i propri prodotti hanno saputo monopolizzare questo mercato: stiamo parlando di **Microsoft** ed **Apple Computers**.

La storia di queste due aziende è piuttosto articolata e, anche se molti non ne sono a conoscenza, intimamente legata fin dalla nascita.

Apple

Apple I

Machintosh

Ipod, Iphone, Ipad, Macbook

Apple
Storia Apple Computers su Wikipedia (IT)
Steve Jobs su Wikipedia (IT)
Steve Wozniak su Wikipedia (IT)



Filippo Micheletti - Progetto TAG

13

La Apple Computers nasce il 1° aprile del 1976 da Steve Jobs e Paul Wozniak, due amici e due persone profondamente diverse: il primo caratterizzato da un'eccezionale mente imprenditoriale, il secondo da delle grandi doti tecniche.

I due riescono nell'impresa di realizzare un computer semplice e tutto sommato potente per l'epoca, l'Apple I (1977) del quale, grazie anche a sacrifici personali come la vendita della calcolatrice scientifica di Wozniak e del pullmino Volkswagen di Jobs, e grazie ai fondi ottenuti dall'associazione con Ronald Wayne, realizzeranno 200 esemplari (nel garage di casa Jobs).

E' la nascita di un'azienda che ha segnato fortemente il mondo dell'informatica: nel giro di pochi anni la Apple Computers si espande a dismisura.

L'impeto della Apple Computers è però croce e virtù: l'azienda entra ben presto in un processo autolesionista alimentato dal carattere visionario di Jobs che pone in forte competizione i due gruppi di sviluppo dei progetti Lisa e Macintosh.

Nel 1981 Wozniak lascia l'azienda e nel 1985 anche Steve Jobs viene allontanato, in corrispondenza del lancio del Macintosh.

La Apple adotta una strategia di mercato rischiosa fornendo computer completi dall'hardware al software e negli anni successivi sopravvive sostanzialmente con la linea Macintosh fino ad entrare in crisi a causa della forte concorrenza sia sull'hardware, a causa dell'espansione enorme del mercato, che sul software, da parte principalmente di Microsoft.

Nel 1996 Steve Jobs (che nel frattempo ha fondato la Pixar) viene riassunto come consigliere dell'attuale amministratore delegato, per seguire lo sviluppo di un nuovo sistema operativo, il MAC OS X.

MAC OS X viene rilasciato nel 2000, ma la vera rivoluzione dell'azienda arriva l'anno successivo, con l'uscita del lettore musicale iPod.

Da allora la Apple ha estremizzato la propria strategia di mercato rendendola sempre più legata al design dei propri prodotti, e alla fidelizzazione assoluta della clientela, ottenendo un grande successo non solo nel settore dei personal computers, con iPhone, Macbook ed iPad.

Microsoft



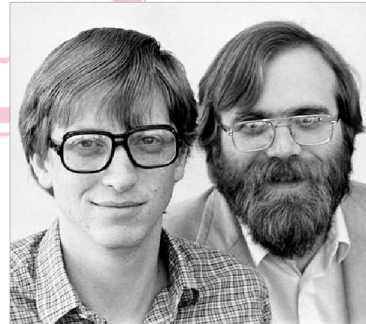
MS-DOS



Windows

Windows

*Microsoft Corporation
Storia della Microsoft Corporation su Wikipedia (IT)
MS DOS su Wikipedia (IT)
Windows su Wikipedia (IT)
Bill Gates su Wikipedia (IT)
Paul Allen su Wikipedia (IT)*



Filippo Micheletti - Progetto TAG

14

La storia della Microsoft Corporation inizia nel 1975 con Bill Gates e Paul Allen I quali propongono all'azienda MITS di utilizzare il linguaggio di programmazione BASIC.

Anche in questo caso si tratta di due figure fortemente contrapposte: da un lato il forte senso degli affari di Bill Gates e dall'altro le competenze tecniche di Paul Allen.

Fino ai primissimi anni 80 il gruppo basa il suo mercato sullo sviluppo di software ed hardware destinato prevalentemente all'Apple II, così come sullo sviluppo di linguaggi come il Microsoft BASIC.

L'evento che dà vita alla Microsoft come la conosciamo oggi è la commessa da parte di IBM per la creazione di un sistema operativo per il proprio personal computer di prossimo lancio, il PC IBM: nasce MC DOS, ottenuto apportando delle migliorie al QDOS, un sistema operativo esistente della Seattle Computer Products del quale la Microsoft, appena divenuta Microsoft Inc., acquisisce i diritti nel 1981.

Nel 1985, in concomitanza con la prima versione del programma di scrittura Microsoft Word, l'azienda introduce l'uso del mouse per i personal computers.

Subito dopo viene rilasciata la prima versione di Windows (3.1) che sorpasserà MS DOS soltanto alla sua seconda versione, Windows 95.

La Microsoft sceglie una strategia commerciale fondamentalmente diversa da quella della Apple: dato l'aumento esponenziale di produttori hardware disponibili sul mercato decide di concentrarsi sullo sviluppo software con cui equipaggiare, tramite opportuni accordi commerciali, l'hardware altrui.

Questa strategia si rivela vincente e porta la Microsoft ad instaurare un vero e proprio monopolio del software per personal computers, non di rado con problemi legali, specialmente in Europa, legati proprio alla monopolizzazione del settore.

Ad oggi Windows XP detiene ancora il primato di SO più utilizzato nel mondo.

ma... perché?

Il segreto è...

... l'interfaccia grafica!

*XEROX PARC su Wikipedia (IT)
Storia dell'interfaccia grafica
I pirati di Silicon Valley su MyMovies
(spezzone del film)*

XEROX

PARC

Filippo Micheletti - Progetto TAG

15

Il grandissimo successo di Apple e Microsoft deriva da un'idea estremamente innovativa che ha rivoluzionato l'utilizzo del computer e sulla quale le rispettive strategie commerciali delle due aziende hanno creato questi due giganteschi imperi.

Stiamo parlando dell'interfaccia grafica.

Fino agli anni 80 infatti l'unico modo di interfacciarsi con il computer era tramite un'interfaccia testuale: un semplice prompt dove l'utente impartisce comandi testuali al computer ottenendo indietro risultati sempre in forma testuale.

L'idea dell'interfaccia grafica però non è stata partorita né dalla Microsoft né dalla Apple, ma dalla Xerox.

Alla fine degli anni 70 infatti alla Xerox, precisamente al PARC, il centro di ricerca della Xerox a Palo Alto, era in corso il progetto del sistema Alto, con lo scopo di migliorare l'interazione uomo-computer. Il team di sviluppo di Alto ebbe la brillante idea di associare l'utilizzo del computer alla gestione di una scrivania, ma la Xerox non credette che questa potesse essere un'intuizione così geniale e la relegò ad un semplice risultato di ricerca per scopi dimostrativi.

Quando nel 1979 Steve Jobs riuscì ad ottenere il permesso di visitare per 3 giorni i laboratori Xerox capì immediatamente il valore di quell'idea e dopo averne studiato l'implementazione la portò con sé.

Oggi è inutile discutere se l'idea di interfaccia grafica sia stata effettivamente rubata alla Xerox o se sia stata semplicemente "ripresa dal cestino" di qualcuno a cui interessava poco, fattostà che il successo delle due più grandi aziende di informatica del mondo è dovuto a farina del sacco di qualcun altro.

A questo proposito vi consiglio di guardare il film "I pirati della Silicon Valley", che racconta proprio la storia della nascita di Apple e Microsoft, storia che dopo l'uscita del film è stata riconosciuta vera sia da Steve Jobs che Bill Gates, e della quale vi faccio vedere uno spezzone significativo.



Ma quindi...

... cosa ha di tanto diverso il software libero?

Vantaggi...



Comunità, comunità, COMUNITA' !

LugMap Italia (IT)
ACROS, LUG Versilia (IT)
Linux Day Italia (IT)
Libera Informatica



Filippo Micheletti - Progetto TAG

18

Il mondo del software libero può avvantaggiarsi di una particolarità davvero unica: una **comunità**.

Il software libero nasce infatti da gruppi di persone che **collaborano** mettendo a disposizione idee e competenze unendo le forze per lo sviluppo di un progetto comune in modo aperto ed accessibile.

Allo sviluppo non contribuisce però solo chi **scrive** il software, ma anche chi lo **usa**, che partecipa a comunità virtuali (forum, liste, ecc) dove si condividono esperienze, si propongono novità che rispecchino le necessità di chi usa quel software, si segnalano bug e deficit.

Insomma la comunità rappresenta il più grande "customer service" spontaneo del mondo, che non potrà mai essere superato da nessun servizio "costruito" ad hoc.

Le comunità non sono però solo virtuali, tutt'altro.

Gli utenti di software libero sono coscenti e fieri di essere accomunati da qualcosa che va oltre l'utilizzo di un computer e si organizzano nelle forme più varie, da associazioni come i **LUG** (Linux User Group) a vere e proprie società (avete mai sentito parlare della Canonical?).

In questa zona il punto di riferimento è l'**associazione culturale ACROS**, LUG di Versilia, Lucca e Massa Carrara, mentre per quanto riguarda la mia esperienza personale faccio parte di un'associazione di promozione sociale che opera nella zona di Firenze e che si chiama **Libera Informatica**.

In questo ambito esistono moltissime iniziative un po' in tutte le direzioni, che vanno dalla promozione del software libero (**Linux Day**) a eventi di tipo ludico fino a semplici ritrovi: un aspetto unico di questo mondo (avete mai sentito parlare di un Windows day?) da non confondere con il rapporto pseudo religioso con cui invece certe multinazionali hanno assoggettato i propri clienti.

Pregi...

Personalizzato

Aggiornato

Documentato

Reperibile



[Linux vs Windows, FAQ \(EN\)](#)

Filippo Micheletti - Progetto TAG

19

Le conseguenze di questa collaborazione è che il software libero è il prodotto della collaborazione di molte persone che si impegnano (principalmente) per **passione** per uno scopo in cui **credono**.

Inoltre l'apertura dei progetti fa sì che nelle comunità di sviluppo possano confluire personalità con competenze di tutti i tipi e a tutti i livelli, che trovano velocemente la propria posizione scegliendo di fare quello in cui si sentono più preparate e più appassionate.

Le comunità di sviluppo sono quindi gruppi in cui collaborano moltissime persone, in maniera continuativa e dinamica, ma soprattutto in continuo contatto con chi utilizza il software prodotto, primi fra tutti gli sviluppatori stessi.

Di conseguenza il software libero è generalmente molto più **aggiornato**, **sicuro** (libero da bug) e **performante** degli equivalenti proprietari, inoltre è generalmente possibile interagire direttamente con chi sviluppa il software per richiedere informazioni, segnalare problemi o mancanze del software.

Un altro aspetto che apprezzerete molto se deciderete di avventurarvi in questo mondo è quello della **documentazione**: il software è documentato a tutti i livelli, dalle procedure di sviluppo ai manuali di utilizzo, in modo semplice e chiaro, e la documentazione è spesso presente in molte lingue

Come vedrete spessissimo si ricorre all'utilizzo delle documentazioni offline (man) per operazioni "rutinarie" come l'uso di un comando, o a quelle online (specialmente le guide ufficiali e gli how-to) per operazioni più complesse.

Un altro aspetto che distingue fortemente il software libero da quello proprietario è la **personalizzabilità**: tutto è concepito per essere customizzato in ogni suo aspetto, e anche quando qualche possibilità di customizzazione mancasse c'è sempre qualcuno che si cimenta per implementarla... perché non voi stessi per esempio? Ricordate che il codice è disponibile, basta rimboccarsi le maniche!

Infine il software libero è sempre **reperibile**, in tutte le versioni in cui è stato creato, insieme a tutta la documentazione originale. Potete comprarvi un computer del 91 e scaricare la prima distribuzione di Debian per divertirvi un po', come potete trovare il driver della vostra vecchia stampante e compilarlo per il vostro sistema a 64 bit.

Non sarete mai costretti a cambiare il computer perché l'ultima versione del software con cui lavorate non gira sul sistema che avete e il nuovo sistema richiede un hardware più potente del vostro, né dovrete mai cambiare la stampante perché i driver non sono più mantenuti per il sistema operativo che avete.

Ah, quasi dimenticavo... tutto questo E' **GRATUITO!**



... difetti...

Organizzazione e gestione della comunità

Lato economico

Canonical on Wikipedia (EN)

Filippo Micheletti - Progetto TAG

21

E' chiaro che non c'è solo una faccia della medaglia e non sarebbe realistico non ammettere anche qualche piccolo difetto del software libero.

Innanzitutto la **gestione della comunità** a tutti i livelli non è semplice. A volte vi entrano a far parte persone poco competenti o che agiscono addirittura contro il progetto per cui c'è sempre bisogno di meccanismi di autocontrollo e di una regolamentazione precisa.

Proprio la mancanza di una buona organizzazione ha decretato il declino di molti progetti e dall'altro lato l'ottima organizzazione dei progetti più diffusi è stata sicuramente una delle armi più vincenti.

Come è possibile immaginare poi un aspetto difficile di questo mondo è quello **economico**: salvo eccezioni i progetti sopravvivono sostanzialmente di donazioni, sia in termini finanziari che in termini di tempo di chi vi si dedica gratuitamente. Il segreto sta tutto nel minimizzare le spese ed annullare gli sprechi.

Certo anche in questo caso non mancano le eccezioni, come per il progetto Ubuntu che è supportato dalla Canonical Ltd, un'azienda sudafricana fondata dal magnate Mark Shuttleworth.

... e luoghi comuni!

Difficoltà

Protezione

Garanzie

Uso professionale

Malware

Costi

[LGPL license \(EN\)](#)
[Creative Commons Italia \(IT\)](#)
[Licenza BSD on Wikipedia \(IT\)](#)
[Linux & malware article \(EN\)](#)

Filippo Micheletti - Progetto TAG

22

Infine credo che sia necessario sfatare alcuni **luoghi comuni**.

Il software libero è più difficile da usare: questa è la più grossa assurdità che si sente dire di solito a proposito del software libero. Spesso purtroppo il cambiamento spaventa e molte persone abituate ad usare certi software proprietari preferiscono rassegnarsi al pagamento di licenze o a compiere azioni illegali procurandosi software crackato piuttosto che provare qualcosa di semplicemente diverso. Nella maggior parte dei casi il software libero offre molte alternative che possono semplificare le cose o complicarle per offrire più servizi, in ogni caso troverete sempre qualcosa facile da usare almeno quanto l'equivalente proprietario a cui siete abituati.

Il software libero non è regolamentato e protetto: non è assolutamente vero. Il principio fondamentale su cui si basa il software libero è quello del copyleft (o permesso d'autore): un modo che ci si è inventati per rappresentare (e proteggere) chi, mentre difende il proprio diritto di autore, vuole difendere la libertà della sua opera, imponendo che questa e le sue derivazioni restino libere. Sulla base del principio di copyleft sono nate alcune licenze a cui abbiamo già accennato, GPL in primis, estese poi a qualsiasi forma di opera creativa (Creative Commons, Art Libre, ecc). Un'altra licenza spesso utilizzata per software libero è la BSD.

Il software libero offre minori garanzie di quello proprietario: sbagliato! In generale sul software proprietario non vi sono particolari garanzie se non quelle legate all'integrità dei dati sul supporto con cui il software viene venduto... il software libero è reperibile online in qualsiasi momento.

Non esiste software libero professionale e i software proprietario di questo tipo sono incompatibili con i sistemi liberi: ancora sbagliato. Esiste una grossa varietà di software libero professionale in grado di rimpiazzare egregiamente il sostituto proprietario, spesso in maniera anche migliore (a questo proposito vi segnalo il Linux Day 2012 il cui filone sarà "Software libero per le piccole e medie imprese"). Inoltre ormai praticamente qualsiasi software professionale viene distribuito compilato per Linux per cui anche volendo (o dovendo) utilizzare un particolare software proprietario è comunque possibile farlo in un ambiente libero. E in casi estremi c'è sempre la virtualizzazione... ma di questo parleremo più avanti!

Linux è indipendente da malware: (purtroppo) non è vero, anche se è sicuramente molto più robusto di qualsiasi SO commerciale ed esistono molti meno virus scritti per Linux.

Passare al software libero permette di non spendere più nulla. Bisogna fare attenzione quando si parla di costi relativi all'utilizzo del software libero, soprattutto in ambito professionale e istituzionale, per non generare false aspettative. L'utilizzo del software libero permette, fra le altre cose, di abbattere i costi legati al software eliminando completamente le spese di licenza, ma questo non vuol dire che non si spenda più nulla. Restano infatti i costi "umani" legati alla manutenzione necessaria su sistemi informatici complessi come quelli che possiamo trovare in un'azienda o un ente pubblico, come una scuola. Nonostante questo il risparmio è generalmente enorme.

Per dimostrare quello di cui si è appena parlato vediamo un video tratto da una puntata di Report, dove si parla di una delle purtroppo pochissime esperienze di migrazione al software libero da parte di un'istituzione in Italia, la provincia autonoma di Bolzano.



Da dove iniziare?



E' necessario Linux?

Scegliere una distribuzione

SUSE
Ubuntu
Ubuntu archivio release (IT)



ubuntu

Filippo Micheletti - Progetto TAG

24

Iniziare ad utilizzare il software libero è più facile di quanto si possa pensare.

Innanzitutto facciamo una precisazione: buona parte del software libero è disponibile anche per sistemi operativi proprietari (Windows, MacOS).

Nonostante la maggior parte dei progetti siano infatti nati e inquadrati in ambiente Linux, è prerogativa di chi sviluppa questi software non negare la possibilità di utilizzarli a nessuno, compreso chi non utilizza un sistema operativo libero. D'altra parte sarebbe controproducente per il progetto stesso limitare la diffusione del software per cui molto spesso potrete trovare software libero compilato anche per Windows o MacOS.

Il problema è semmai inverso: molto spesso i sistemi operativi proprietari pongono vincoli all'installazione e all'utilizzo di software libero, impediscono o rendono molto difficoltosa e complicata l'integrazione nell'ambiente di lavoro, bloccano alcune funzionalità di sistema e così via.

Un altro problema frequente è quello dei formati.

Il software libero infatti predilige l'utilizzo di formati liberi, che generalmente sono anche i più performanti ed i più standardizzati, quindi quelli che riservano meno (spiacevoli) sorprese, ma non vieta l'utilizzo e l'interazione con quelli proprietari, anzi, generalmente non solo lo permette ma lo facilita proprio perché anche in questo caso questo approccio favorisce l'utilizzo di quel dato formato permettendo così di conoscerlo e scoprirne le potenzialità.

Di nuovo però non vale (di solito) il contrario: la maggior parte dei software proprietari tende ad impedire o almeno sfavorire l'utilizzo di formati liberi per mantenere il monopolio su quel determinato tipo di documento e tentare di costringere l'utente a non abbandonare quel determinato software.

Sarà abbastanza chiaro allora che per poter sfruttare a pieno le potenzialità del software libero conviene partire innanzitutto da un sistema operativo libero.

Ma come scegliere se qualche slide fa abbiamo detto che esistono centinaia di distribuzioni basate su Linux?

La cosa migliore da fare è partire con una distribuzione semplice che permetta un passaggio poco "traumatico" dalle vecchie alle nuove abitudini.

Tra le distribuzioni più semplici da installare ed utilizzare ci sono sicuramente Ubuntu e Suse.

Sui siti dei progetti si trovano guide chiarissime che spiegano in pochi semplici passaggi come installare il sistema e addirittura Ubuntu permette l'installazione da Windows come se si trattasse di un comune programma a cui sarete sicuramente abituati!

Distribuzioni live e dual boot

Live: recuperare, riparare... e provare!



[Debian Live Project \(EN\)](#)

Filippo Micheletti - Progetto TAG

25

Per alcune distribuzioni basate su Linux inoltre c'è la possibilità di provare il sistema senza modificare il proprio computer utilizzando una distribuzione live.

Una distribuzione live è semplicemente una versione di quel dato sistema operativo che può essere eseguita senza bisogno di essere installata. Semplicemente è sufficiente scaricarla, metterla su un supporto ed avviare il computer: la distribuzione live si caricherà sulla RAM permettendovi di utilizzarla fino a quando spegnerete il computer che al riavvio sarà esattamente come l'avevate lasciato (a meno che nel frattempo non provochiate qualche danno s'intende!).

Le distribuzioni live in realtà non nascono con l'intento di far testare un determinato sistema ma sono state pensate principalmente per scopi di recupero dati, manutenzione e ripristino di sistemi danneggiati.

Ubuntu offre la live, dalla quale potete anche lanciare l'installer del sistema.

Abbastanza di recente è nata anche la live di Debian.

Un'altra possibilità per avvicinarsi ad un sistema operativo diverso da quello a cui si è abituati infine è il dual boot.

In poche parole si tratta di installare la distribuzione che avete scelto (o perché no, più di una) mantenendo anche il sistema operativo già installato (previa disponibilità di spazio chiaramente).

Il dual boot è gestito da un software detto boot-loader che permette di far convivere i vari sistemi operativi installati e di scegliere quale avviare al momento dell'accensione del computer.

Personalmente ritengo il dual boot una cosa da evitare specialmente quando ci si avvicina per la prima volta al software libero perché la coscienza di avere ancora il vecchio sistema a cui siamo abituati spesso "invita a desistere" anche di fronte a quelle prime piccolissime difficoltà che si possono incontrare.

Se avete intenzione di mantenere il sistema che avete fatto: è decisamente molto più costruttivo installare Linux magari su un vecchio pc in disuso e provarlo da lì che fare il dual boot (che fra l'altro complica un po' le cose).

Trovare il software



Googolate ragazzi!

[osalt \(EN\)](#)



Filippo Micheletti - Progetto TAG

26

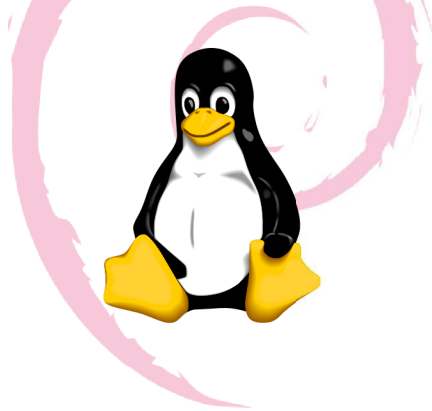
La paura di non trovare il software di cui sia ha bisogno spesso attanaglia chi si avvicina per la prima volta a questo mondo ma nessuna paura, esiste un'alternativa libera a tutto e molto di più!

Il metodo principe per trovare il software che si sta cercando è banalmente googolare, ma esistono anche dei siti specifici come osalt (Open Source ad ALternative), un motore di ricerca specializzato in cui inserendo nel campo di ricerca il nome del software proprietario di cui si cerca l'alternativa, si ottiene una lista di risultati di software equivalenti ma liberi.

Infine le più famose distribuzioni basate su Linux di cui abbiamo parlato, come Ubuntu, Suse, Debian, hanno un incorporati dei sistemi di distribuzione e gestione del software "automatici" che permettono di cercare, scaricare, installare ed aggiornare automaticamente il software che si desidera da degli archivi sicuri ed aggiornati, detti repository.

Un po' come avviene oggi con l'Apple store per IOS o il Play store di Android: un'idea molto intelligente inventata una ventina di anni fa nel progetto GNU.

Grazie a tutti per l'attenzione :)



Filippo Micheletti - Progetto TAG

27

La paura di non trovare il software di cui sia ha bisogno spesso attanaglia chi si avvicina per la prima volta a questo mondo ma nessuna paura, esiste un'alternativa libera a tutto e molto di più!

Il metodo principe per trovare il software che si sta cercando è banalmente googolare, ma esistono anche dei siti specifici come osalt (Open Source ad ALternative), un motore di ricerca specializzato in cui inserendo nel campo di ricerca il nome del software proprietario di cui si cerca l'alternativa, si ottiene una lista di risultati di software equivalenti ma liberi.

Infine le più famose distribuzioni basate su Linux di cui abbiamo parlato, come Ubuntu, Suse, Debian, hanno un incorporati dei sistemi di distribuzione e gestione del software "automatici" che permettono di cercare, scaricare, installare ed aggiornare automaticamente il software che si desidera da degli archivi sicuri ed aggiornati, detti repository.

Un po' come avviene oggi con l'Apple store per IOS o il Play store di Android: un'idea molto intelligente inventata una ventina di anni fa nel progetto GNU.