

Programmazione e Algoritmi 1

Linux e software libero

prof. Gianluca Amato

Corso di Laurea in Economia e Informatica per l'Impresa
a.a. 2023/24

26 settembre 2023

Interazione

Gianluca
Amato

Storia

Software libero
e proprietario

Linux

Obiettivi di queste slide:

- Presentare brevemente la storia del sistema operativo Linux
- Introdurre il concetto di software libero

Queste slide sono più “piene” del solito perché l’argomento trattato non è presente nel libro di testo.

Interazione

Gianluca
Amato

Storia

Software libero
e proprietario

Linux

1 Un po' di storia

2 Software libero e proprietario

3 Linux

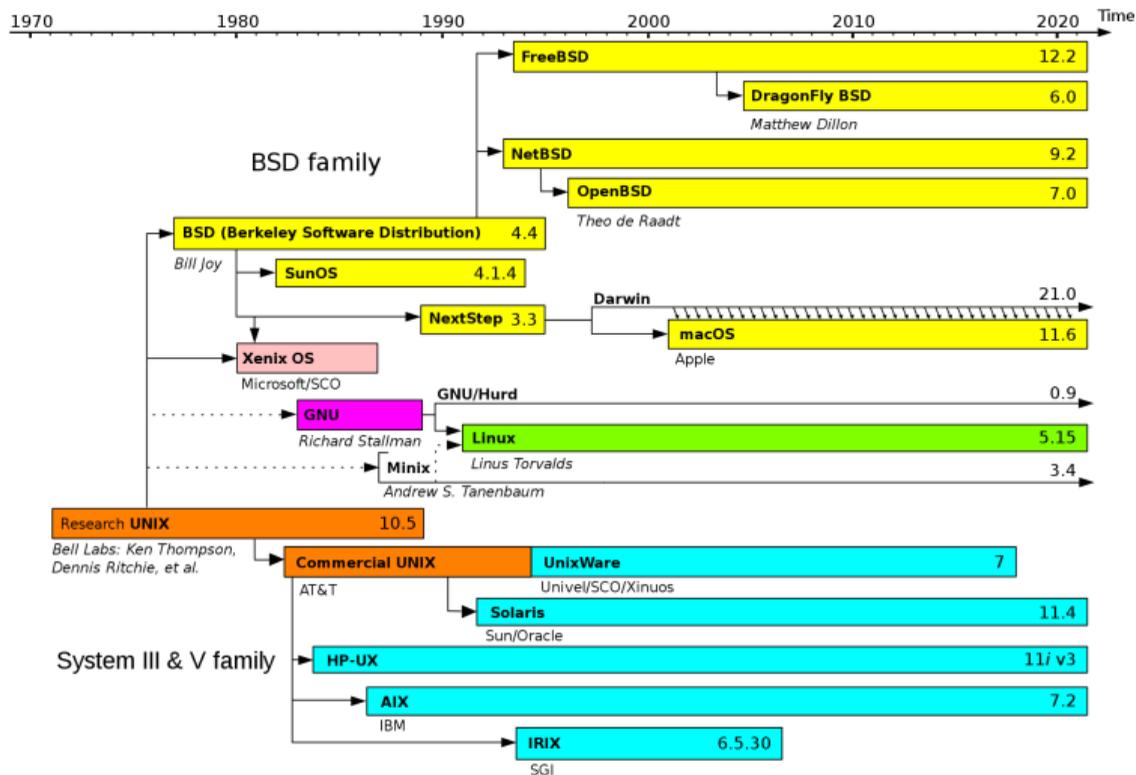
- Linux è un sistema operativo **Unix-like**, ovvero ispirato a Unix
- **Unix** è un sistema operativo:
 - nato nei primi anni '70
 - sviluppato ai Bell Labs della AT&T (la compagnia telefonica americana, allora monopolista)
- il nome Unix è un gioco di parole con Multics, un sistema operativo di poco precedente che non ebbe mai molto successo
 - Multics è un acronimo di **Multiplexed Information and Computing Service**, ed era un sistem operativo **multi-tasking** (come tutti quelli moderni), cioè poteva eseguire più programmi contemporaneamente;
 - Unix è un acronimo di **Uniplexed Information and Computing Service**, e fu scelto come accronimo perché, a differenza di Multics, era inizialmente un sistema operativo **mono-tasking**.
- Unix cominciò a diffondersi nel 1973, quando fu riscritto in C (prima era in Assembly)

Unix fu venduto da AT&T a vari produttori di computer, che lo adattarono ai loro sistemi:

- ogni versione di Unix era diversa dalle altre, e non c'era compatibilità tra di esse
- anche il nome era diverso
 - IBM AIX
 - Sun Solaris
 - ...
- questo portò a una frammentazione del mercato, con molti produttori che vendevano sistemi incompatibili tra di loro

Un tentativo di standardizzazione fu lo standard **POSIX** (Portable Operating System Interface), ma ebbe solo un successo parziale.

La frammentazione impedì la diffusione di Unix tra i privati e consentì a Microsoft di diventare il leader del mercato dei sistemi operativi per personal computer prima, e dei server per aziende poi.



Interazione

Gianluca
Amato

Storia

Software libero
e proprietario

Linux

1 Un po' di storia

2 Software libero e proprietario

3 Linux

- Le regole internazionali sui diritti d'autore proteggono il software come **opera dell'ingegno** (come la musica, i libri, etc..)
 - Nessuno, tranne l'autore, può utilizzare il programma senza permesso.
 - All'utente finale, il permesso di utilizzo di un software viene concesso tramite una **licenza d'uso**.
- Molte licenze sono restrittive:
 - Vietano la copia del software.
 - Vietano di ridistribuire il software, sia a pagamento che gratis.
 - Vietano di modificare il programma per adattarlo alle proprie esigenze, o per correggerne i difetti.
- Spesso i programmi sono distribuiti solo sotto forma di codice binario.
 - Il codice sorgente è segreto.
 - Anche quando la modifica del programma non è espressamente vietata, senza il codice sorgente è quasi impossibile farlo.

- L'idea di commercializzare il software si sviluppa negli anni '70. Prima di allora, il software era quasi sempre distribuito gratuitamente assieme al codice sorgente.
- Negli anni '80, come reazione a questa tendenza, nasce il movimento del software libero.
- La figura più prominente del movimento è sicuramente Richard M. Stallman, ricercatore e hacker al MIT, che nel 1984 lascia il posto di lavoro:
 - Nasce il progetto GNU: l'obiettivo è produrre un sistema operativo completamente libero
 - Poco dopo nasce anche la Free Software Foundation, una società senza fini di lucro il cui scopo è promuovere la diffusione del software libero.
- Ma cosa vuol dire **software libero** ?
 - Stallman identifica quattro libertà principali che a un utente devono essere garantite.
 - Un programma la cui licenza garantisce all'utente queste libertà è un software libero.
 - L'opposto del software libero è il **software proprietario**.

- 1** Eseguire il programma, per qualsiasi scopo
 - Quindi senza nessuna restrizione che ne impedisce, ad esempio l'utilizzo per scopi commerciali.
- 2** Studiare come funziona il programma e adattarlo alle proprie necessità.
 - Ciò implica che l'accesso al codice sorgente è una caratteristica necessaria del software libero.
- 3** Copiare il programma e ridistribuire copie.
 - Quindi un software libero può essere copiato e regalato a un amico.
- 4** Modificare il programma e distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio.

Interazione

Gianluca
Amato

Storia

Software libero
e proprietario

Linux

Se il software libero può essere liberamente copiato e ridistribuito, come si può guadagnare con esso?

- Fornitura di supporto tecnico a pagamento.
- Modifiche su richiesta di software per adattarlo a specifiche esigenze.
- Produzione e vendita di manuali e libri.

Se un programma è libero quando esce dalle mani di un autore, non è detto che altri programmi da esso derivati lo siano.

- La sua licenza potrebbe consentire a un altro soggetto di prendere il programma, modificarlo e distribuire la modifica con una licenza diversa, proprietaria e non più libera.
- Chi acquista il software modificato non ha più un software libero.

E se si vuole evitare che il proprio lavoro possa essere utilizzato in un software in proprietario?

- Si usa una licenza basata sul “permesso d’autore” (**copyleft**)
- Il permesso d’autore (copyleft) usa le leggi del diritto d’autore (copyright) ma per ottenere lo scopo opposto: invece che un metodo per privatizzare il software, diventa un metodo per mantenerlo libero.

Il succo dell'idea è questa:

- All'utente di un programma con licenza basata su copyleft sono garantite le 4 libertà fondamentali.
- Tuttavia, tutte le modifiche al programma vanno distribuite con la licenza originale.

La prima licenza copyleft e la più famosa è la GNU GPL (General Public License) con cui è distribuito Linux.

Tradizionalmente, si è voluto affermare che

- Le licenze senza copyleft puntano a favorire la massima diffusione del software, a scapito della libertà degli utenti.
- Le licenze con copyleft hanno l'obiettivo primario di difendere la libertà degli utenti, anche a scapito della sua diffusione, che potrebbe essere rallentata dall'impossibilità di utilizzarlo in ambienti proprietari.

La creazione di una nuova licenza libera è una cosa complicata...

- Problemi di incompatibilità (quasi sempre non volute) tra licenze
 - Rendono impossibile unificare due programmi distribuiti con licenze diverse.
- Difficoltà pratiche inaspettate nell'uso di una licenza

Ad esempio, la licenza BSD originale è una licenza senza copyleft, ma con una clausola particolare: qualunque materiale pubblicitario di un'azienda che menzioni il software, deve aggiungere questa clausola:

This product includes software developed by the Regents of the University of California.

Questa restrizione:

- la rende incompatibile con la GPL;
- cosa succede se 100 persone utilizzano questa licenza, modificando la clausola per inserire il proprio nome?

Prima ancora della diffusione del concetto di software libero esisteva il **software di pubblico dominio**.

- Nel software di pubblico dominio l'autore rinuncia ai diritti d'autore.
Il software è disponibile pubblicamente, e chiunque ne viene in possesso può fare con questo ciò che si desidera.
- Il software di pubblico dominio è software libero qualora sia disponibile in codice sorgente.
Se è disponibile solo il codice binario, per quanto detto prima, non si può considerare libero.
- Non tutto il software libero è di pubblico dominio.
Il software distribuito con licenze stile MIT o con copyleft non è di pubblico dominio.

- **Software Freeware**: non esiste una definizione univoca. Di solito si intende del software gratuito non modificabile. Non si ha accesso ai sorgenti e talvolta neanche possibilità di redistribuzione.

Esempio: Microsoft Edge, Acrobat Reader

- **Software Shareware**: software di cui è permessa la distribuzione ma che bisogna pagare per l'uso. Spesso non si ha accesso ai sorgenti.

Esempio: WinZip

- **Software commerciale**: è un software sviluppato da una azienda per trarre profitto dal suo uso. Può essere libero oppure proprietario.

Esempio di software commerciale ma libero: Oracle MySQL

Esempio di software commerciale proprietario: Microsoft Word

Tra i produttori di software commerciale libero è molto diffusa la pratica della doppia licenza

- Lo stesso software viene distribuito in due modi diversi:
 - gratuitamente con una licenza libera (spesso con clausola copyleft)
 - a pagamento con una licenza proprietaria.
- Vantaggio:
 - La maggior parte degli utenti usa la versione libera che (essendo protetta da copyleft) non potrà essere utilizzata da dei concorrenti per produrre varianti proprietarie del software.
 - Chi ha bisogno di utilizzare il software all'interno di un prodotto proprietario (per cui non può utilizzare la versione gratuita con copyleft) può richiedere a pagamento la versione proprietaria.
- Ad esempio: MySQL è distribuito con licenza GPL o proprietaria.

Un termine che si è molto diffuso ultimamente è quello di software **open source**.

Dal punto di vista pratico, software libero e software open source sono quasi la stessa cosa, ma i principi che stanno alla base delle due terminologie sono molto diversi:

- Software libero: pone l'accento sulla questione della libertà degli utenti.
- Software open source: difende l'idea che il modello di sviluppo del software libero (sorgenti disponibili a tutti, chiunque può modificare e migliorare i prodotti. . .) crei dei prodotti migliori.

Il principio di libertà degli utenti è assente o in secondo piano.

L'idea di software open source è più appetibile alle aziende, in quanto pone l'accento sui vantaggi competitivi e quindi sui profitti.

Un saggio che esamina questi due modelli di sviluppo è “**La cattedrale e il bazaar**” di Eric S. Raymond

Sono due metodi di sviluppo di software

■ La cattedrale

- Il programma viene realizzato da un numero limitato di esperti che scrivono il codice in quasi totale isolamento.
- Il progetto ha una suddivisione gerarchica molto stretta e ogni sviluppatore si preoccupa della sua piccola parte di codice.
- Tipico del software proprietario, ma utilizzato anche per qualche software libero.

■ Il bazaar

- il codice sorgente della revisione in sviluppo è disponibile liberamente, gli utenti possono interagire con gli sviluppatori e se ne hanno le capacità possono modificare e integrare il codice.
- Lo sviluppo è decentralizzato e non esiste una rigida suddivisione dei compiti.
- Tipico del software libero.

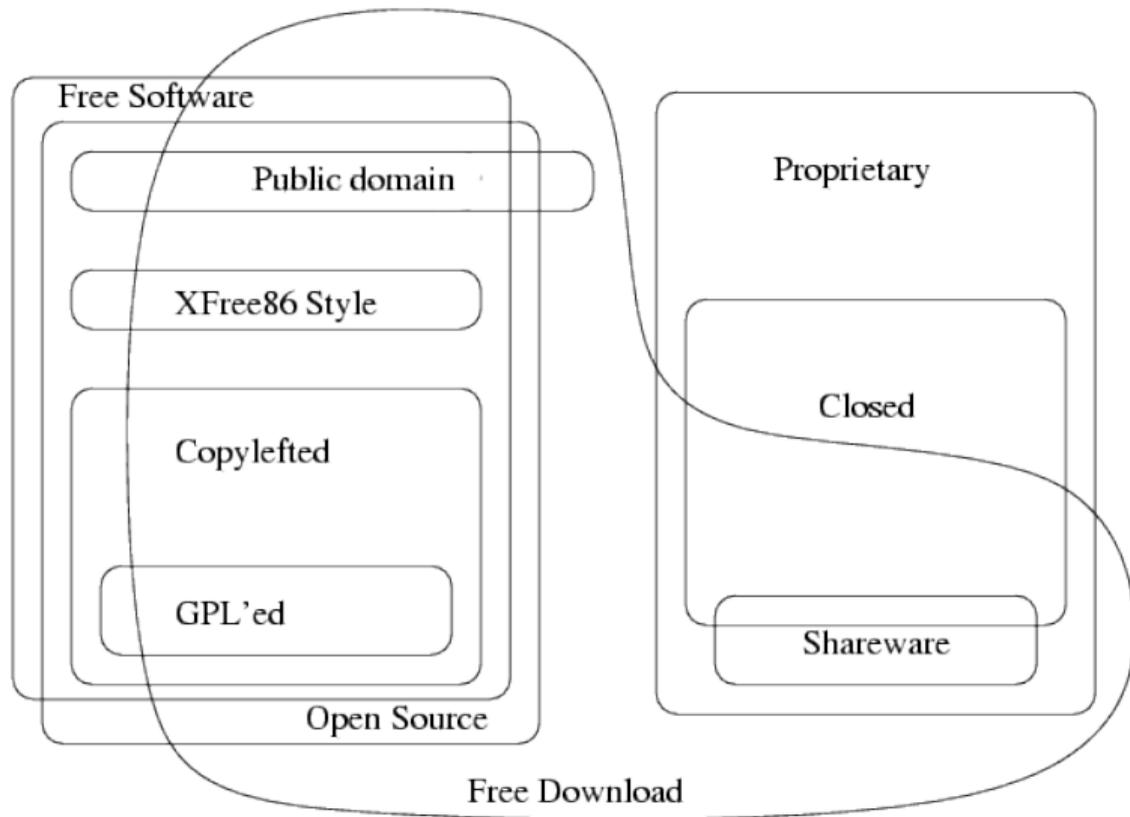
Interazione

Gianluca Amato

Storia

Software libero e proprietario

Linux



Interazione

Gianluca
Amato

Storia

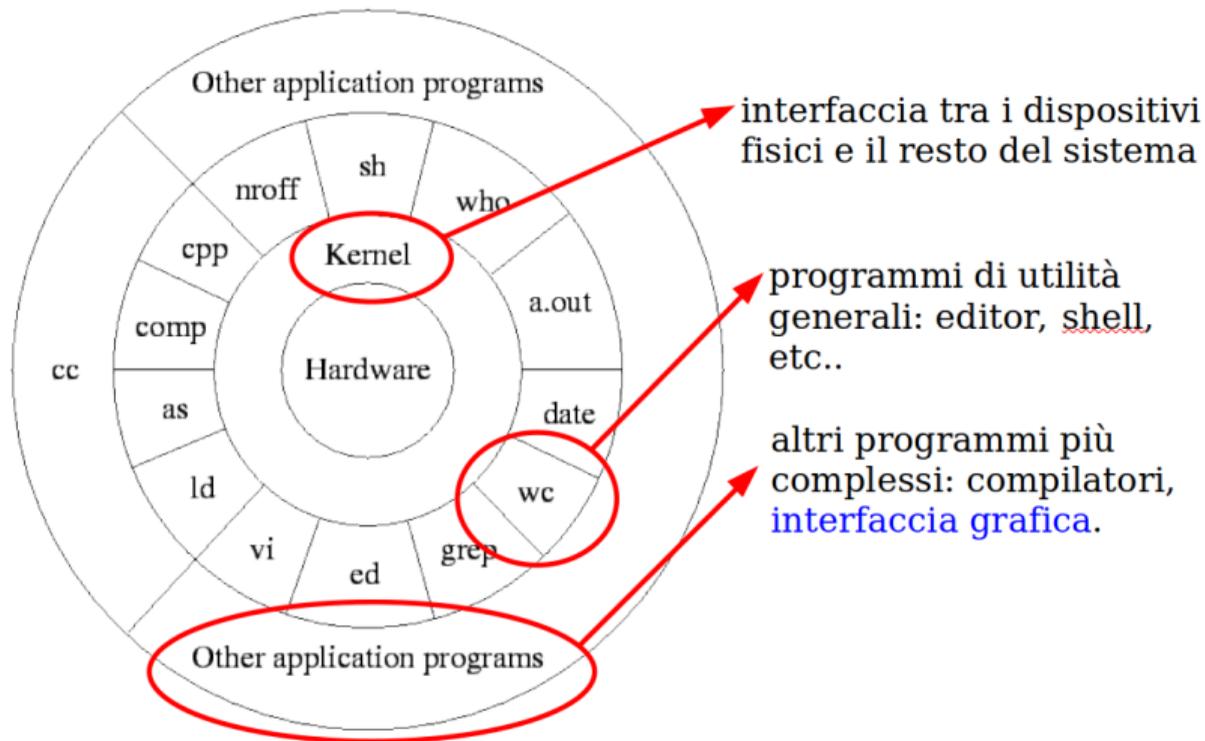
Software libero
e proprietario

Linux

1 Un po' di storia

2 Software libero e proprietario

3 Linux



- All'inizio degli anni '90, il progetto GNU sviluppa gran parte degli strumenti necessari per un sistema operativo Unix-like, tranne il **kernel**
 - un po' come avere un'automobile senza motore
 - il kernel del progetto GNU è Hurd, ma non è mai stato completato
- nel 1991 Linus Torvalds, studente del corso di Sistemi Operativi a Helsinki, annuncia la versione 0.02 di Linux: è il pezzo che mancava
 - il modello di sviluppo "a bazaar" portato avanti da Linus ha un grande successo e tantissime persone partecipano allo sviluppo.

- **Linux** è un kernel, ovvero la parte di un sistema operativo che gestisce l'hardware
- Linux è l'acronimo di **Linux Is Not Unix**:
 - ricorda il nome del creatore, Linux Torvalds;
 - mette l'accento sul fatto che il codice sorgente non è basato su Unix.
- Col termine Linux si intende però impropriamente anche l'intero sistema operativo, composto da:
 - il kernel Linux;
 - il progetto GNU;
 - altri progetti, come X.org, GNOME, KDE, Apache, Firefox, ecc.
- Stallman e i suoi collaboratori preferiscono chiamare il sistema operativo completo **GNU/Linux**.

- GNU/Linux è composto da vari programmi
- Se tutti fossero da scaricare separatamente sarebbe un incubo per gli utenti: nascono le **distribuzioni**.
 - una collezione di programmi, imperniati sul kernel di Linux e su vario software aggiuntivo che costituisce un S.O. completo.
- Varie distribuzioni
 - *Ubuntu*: la più diffusa al momento, semplice da utilizzare, installare ed aggiornare, è quella installata in aula informatica;
 - *Fedora*: quella che uso io;
 - *RedHat Enterprise Linux*: orientata al mondo aziendale, con supporto commerciale;
 - *Debian*: legata fortemente ai principi del software libero, e portata avanti da una organizzazione di volontari.
 - ...

- Ogni distribuzione ha un suo numero di versione, non in sincronia
 - Ubuntu 22.04
 - Fedora 36
- Ogni applicazione (e lo stesso kernel Linux) ha una propria versione specifica:
 - Linux 5.15.0 (kernel)
 - GNOME 42 (interfaccia grafica)
 - LibreOffice 7.3.5 (suite di programmi da ufficio)
- Possiamo quindi fare affermazioni del tipo:
 - *La Ubuntu 22.04 è basata su Linux 5.15 e GNOME 42*

- Computer pre-installato con Linux
 - Abbastanza comune nel mercato dei server
 - Poca scelta per i computer desktop o portatili
- Installare Linux a fianco di Windows o macOS
 - Scelta molto comune, ma non è semplicissimo per chi non l'ha mai fatto neanche una volta
 - Per macOS è ancora più difficile perché spesso l'hardware proprietario del Mac non funziona al 100% con Linux
- Utilizzare Linux in modalità live da DVD o memoria flash USB
 - Si fail boot direttamente da DVD o memoria USB e non richiede installazione
 - Funzionamento molto più lento del normale
- Utilizzare Linux all'interno di una macchina virtuale
 - caso particolare: il **Sottosistema Windows per Linux**

- È un software libero
 - adattabile a tutte le esigenze
 - esente da “codice spia” che controlla l’attività degli utenti
 - non è legato a nessun produttore in particolare, quindi non c’è rischio di vendor lock-in
- È meno vulnerabile al concetto di virus
- È disponibile su un gran numero di piattaforme:
 - Personal Computer
 - Macchine Server
 - Cellulari: Android è una versione di Linux sviluppata da Google
 - Accessori vari: la maggior parte dei sistemi IoT (Internet of Things) funziona su Linux
- Dominante nel mondo dei server e dell’IoT.

- Supporto hardware non ottimale
 - di alcune periferiche non esistono driver per Linux
- Mancano alcuni tipi di applicazioni
 - ad esempio, non c'è una grande disponibilità di videogiochi commerciali
- L'interfaccia utente è in certi punti carente
 - la varietà di soluzioni e di ambienti possibili crea infatti una certa disomogeneità nel funzionamento.