

Programmazione e Algoritmi 1

Linux e software libero

prof. Gianluca Amato

Corso di Laurea in Economia e Informatica per l'Impresa
a.a. 2024/25

25 settembre 2024

Obiettivo di queste slide:

- Introdurre il concetto di software libero
- Introdurre brevemente il sistema operativo Linux

Queste slide sono più “piene” del solito perché gli argomenti trattati non sono presenti nel libro di testo.

Interazione

Gianluca
Amato

Software libero e proprietario

Introduzione a
Linux

1 Software libero e proprietario

2 Introduzione a Linux

- Le regole internazionali sui diritti d'autore proteggono il software come **opera dell'ingegno** (come la musica, i libri, ecc.).
 - 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**.

- Le regole internazionali sui diritti d'autore proteggono il software come **opera dell'ingegno** (come la musica, i libri, ecc.).
 - 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 redistribuire il software, sia a pagamento che gratis.
 - Vietano di modificare il programma per adattarlo alle proprie esigenze, o per correggerne i difetti.

- Le regole internazionali sui diritti d'autore proteggono il software come **opera dell'ingegno** (come la musica, i libri, ecc.).
 - 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 redistribuire 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.

Interazione

Gianluca
Amato

Software libero
e proprietario

Introduzione a
Linux

- 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.

- 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**, programmatore e hacker al MIT. Nel 1984 lascia il MIT per fondare:

- il **progetto GNU**, il cui obiettivo è produrre un sistema operativo completamente libero;
- la **Free Software Foundation**, una società senza fini di lucro il cui scopo è promuovere la diffusione del software libero.

- 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**, programmatore e hacker al MIT. Nel 1984 lascia il MIT per fondare:

- il **progetto GNU**, il cui obiettivo è produrre un sistema operativo completamente libero;
 - 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.

- 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.

- 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 redistribuire le copie.

- 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 redistribuire le copie.
- 4 Modificare il programma e distribuirne pubblicamente i miglioramenti.

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

Alcuni esempi:

- 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.

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 obbligatoriamente con la licenza originale.

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

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 obbligatoriamente 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 (che è un testo giuridico) è una cosa complicata.

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

Esempio

La licenza *BSD originale* è una licenza senza copyleft, ma richiede che qualunque materiale pubblicitario menzioni il software debba contenere questa clausola:

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

Questa restrizione:

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

Interazione

Gianluca
Amato

Software libero
e proprietario

Introduzione a
Linux

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

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.

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 il codice sorgente.
Se è disponibile solo il codice binario, per quanto detto prima, non si può considerare libero.

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 il 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 copyleft non è di pubblico dominio.

Interazione

Gianluca
Amato

Software libero
e proprietario

Introduzione a
Linux

- **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.

Esempi: Microsoft Edge, Acrobat Reader

- **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.

Esempi: 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.

Esempi: WinZip, Doom

- **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.
Esempi: 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.
Esempi: WinZip, Doom
- **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: Android
Esempio di software commerciale proprietario: Microsoft Word

Interazione

Gianluca
Amato

Software libero
e proprietario

Introduzione a
Linux

Un termine alternativo a quello di software libero 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.

Interazione

Gianluca
Amato

Software libero
e proprietario

Introduzione a
Linux

Un termine alternativo a quello di software libero 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.

Un termine alternativo a quello di software libero 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 programmi, ecc.) crei dei prodotti migliori.

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

Un termine alternativo a quello di software libero 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 programmi, ecc.) 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 termine alternativo a quello di software libero è quello di software **open source**.

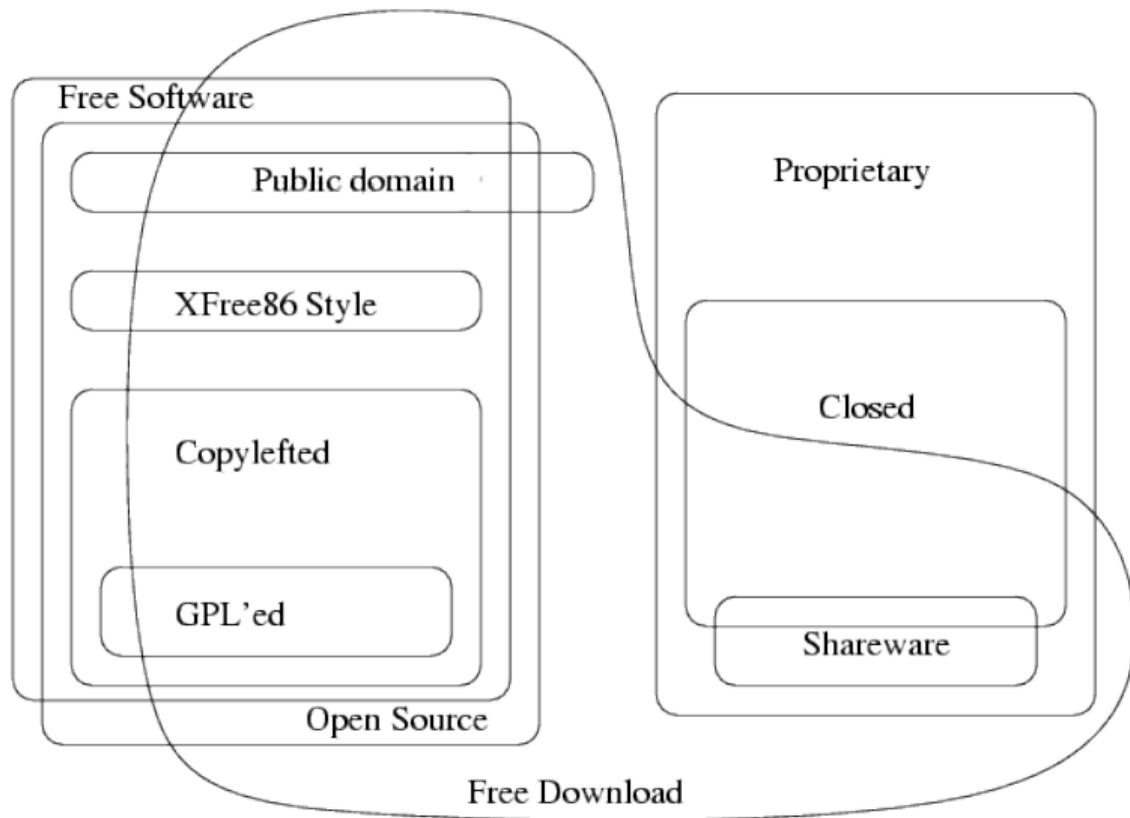
Dal punto di vista pratico, software libero e software open source sono quasi la stessa cosa, ma i principi che stanno alla base delle due terminologie sono molto diversi.

- 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 programmi, ecc.) 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 confronta software proprietario e open source dal punto di visto del modello di sviluppo è "**La cattedrale e il bazaar**" di **Eric S. Raymond**.



Interazione

Gianluca
Amato

Software libero
e proprietario

Introduzione a
Linux

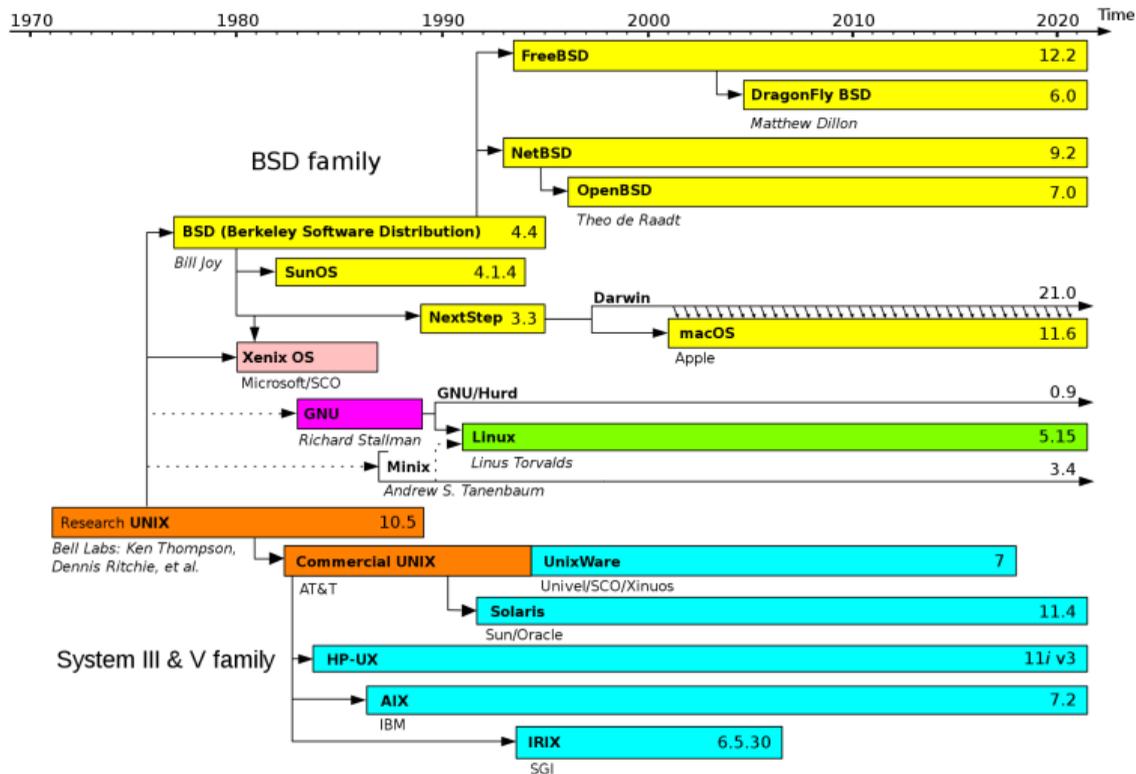
1 Software libero e proprietario

2 Introduzione a Linux

- Linux è un sistema operativo **Unix-like**, ovvero ispirato a Unix

- Linux è un sistema operativo **Unix-like**, ovvero ispirato a Unix
- **Unix** è un sistema operativo nato nei primi anni '70
 - fu sviluppato ai Bell Labs della AT&T (la compagnia telefonica americana)
 - fu venduto dalla AT&T a vari produttori di computer, dando origine a una pletera di versioni tra loro simili ma non perfettamente compatibili
 - IBM AIX
 - Sun Solaris
 - ...

- Linux è un sistema operativo **Unix-like**, ovvero ispirato a Unix
- **Unix** è un sistema operativo nato nei primi anni '70
 - fu sviluppato ai Bell Labs della AT&T (la compagnia telefonica americana)
 - fu venduto dalla AT&T a vari produttori di computer, dando origine a una pletera di versioni tra loro simili ma non perfettamente compatibili
 - IBM AIX
 - Sun Solaris
 - ...
- La frammentazione impedì la diffusione di Unix tra i privati e consentì a Microsoft di diventare, negli anno '90, il leader del mercato dei sistemi operativi per computer personali prima e aziendali dopo.



- All'inizio degli anni '90, il progetto GNU (quello fondato da Stallman) ha già sviluppato gran parte degli strumenti necessari per un sistema operativo Unix-like, tranne il **kernel**.
 - il kernel¹ è la parte centrale del sistema operativo;
 - un sistema operativo senza kernel è un po' come un'automobile senza motore;
 - il kernel sviluppato del progetto GNU si chiama Hurd, ma non è mai stato completato.

¹Maggiori informazioni su cosa è un kernel lo vedrete nel modulo di Architettura e Sistemi Operativi al 2° semestre.

- All'inizio degli anni '90, il progetto GNU (quello fondato da Stallman) ha già sviluppato gran parte degli strumenti necessari per un sistema operativo Unix-like, tranne il **kernel**.
 - il kernel¹ è la parte centrale del sistema operativo;
 - un sistema operativo senza kernel è un po' come un'automobile senza motore;
 - il kernel sviluppato del progetto GNU si chiama 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.

¹Maggiori informazioni su cosa è un kernel lo vedrete nel modulo di Architettura e Sistemi Operativi al 2° semestre.

- Linux è l'acronimo di **Linux Is Not UniX**:
 - ricorda il nome del creatore, Linus Torvalds;
 - mette l'accento sul fatto che, sebbene ispirato a Unix, il codice sorgente non è basato su quello di Unix.

- Linux è l'acronimo di **Linux Is Not UniX**:
 - ricorda il nome del creatore, Linus Torvalds;
 - mette l'accento sul fatto che, sebbene ispirato a Unix, il codice sorgente non è basato su quello di Unix.
- Sebbene Linux sia tecnicamente solo il kernel, con questo termine si intende impropriamente anche l'intero sistema operativo, composto da:
 - il kernel Linux;
 - i software del progetto GNU (compilatori, editor, l'interfaccia grafica GNOME, ecc.);
 - altri software come LibreOffice, Firefox, ecc.

- Linux è l'acronimo di **Linux Is Not UniX**:
 - ricorda il nome del creatore, Linus Torvalds;
 - mette l'accento sul fatto che, sebbene ispirato a Unix, il codice sorgente non è basato su quello di Unix.
- Sebbene Linux sia tecnicamente solo il kernel, con questo termine si intende impropriamente anche l'intero sistema operativo, composto da:
 - il kernel Linux;
 - i software del progetto GNU (compilatori, editor, l'interfaccia grafica GNOME, ecc.);
 - altri software come LibreOffice, Firefox, ecc.
- Stallman e i suoi collaboratori preferiscono chiamare il sistema operativo completo **GNU/Linux**.

- GNU/Linux è quindi 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 sistema operativo completo.

- GNU/Linux è quindi 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 sistema operativo completo.
- Varie distribuzioni
 - *Ubuntu*: la più diffusa al momento, semplice da utilizzare, installare ed aggiornare, è quella presente 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 24.04
 - Fedora 40

- Ogni distribuzione ha un suo numero di versione, non in sincronia
 - Ubuntu 24.04
 - Fedora 40
- 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)

- Ogni distribuzione ha un suo numero di versione, non in sincronia
 - Ubuntu 24.04
 - Fedora 40
- 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

- 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 i Mac è ancora più difficile perché spesso l'hardware proprietario del Mac non funziona bene con Linux

- 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 i Mac è ancora più difficile perché spesso l'hardware proprietario del Mac non funziona bene con Linux
- Utilizzare Linux in modalità live da memoria flash USB (chiavette USB)
 - Si fa il boot direttamente da chiavette e non richiede installazione
 - Funzionamento molto più lento del normale

- 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 i Mac è ancora più difficile perché spesso l'hardware proprietario del Mac non funziona bene con Linux
- Utilizzare Linux in modalità live da memoria flash USB (chiavette USB)
 - Si fa il boot direttamente da chiavette e non richiede installazione
 - Funzionamento molto più lento del normale
- Utilizzare Linux all'interno di una macchina virtuale
 - Una macchina virtuale è un computer virtuale simulato dal computer reale con software quali **VMWare** o **VirtualBox**.
 - È la strada che vi consiglio se volete iniziare a “smanettare” con Linux

Interazione

Gianluca
Amato

Software libero
e proprietario

Introduzione a
Linux

- È un software libero.
 - 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 un software libero.
 - 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 a virus e malware.
 - Windows è il sistema operativo più diffuso per utenti privati, quindi è il più attaccato;
 - Windows (soprattutto in passato) puntava sulla semplicità d’uso a scapito della sicurezza.

- È un software libero.
 - 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 a virus e malware.
 - Windows è il sistema operativo più diffuso per utenti privati, quindi è il più attaccato;
 - Windows (soprattutto in passato) puntava sulla semplicità d’uso a scapito della sicurezza.
- È disponibile su un gran numero di piattaforme.
 - Personal Computer
 - 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

- È un software libero.
 - 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 a virus e malware.
 - Windows è il sistema operativo più diffuso per utenti privati, quindi è il più attaccato;
 - Windows (soprattutto in passato) puntava sulla semplicità d’uso a scapito della sicurezza.
- È disponibile su un gran numero di piattaforme.
 - Personal Computer
 - 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 di Internet e dell’IoT.
 - Linux e altri sistemi Unix servono l’86% dei siti web del mondo ([fonte](#)).
 - Nel 2018, Linux girava nel 71% dei dispositivi IoT ([fonte](#)).

- Poco usato in ambito personale.
 - il market share di Linux sui computer desktop si attesta intorno al 6%, contando anche ChromeOS che è una versione di Linux che gira sui Chromebook ([fonte](#));
 - macOS: 15%, Windows: 70%.

- Poco usato in ambito personale.
 - il market share di Linux sui computer desktop si attesta intorno al 6%, contando anche ChromeOS che è una versione di Linux che gira sui Chromebook ([fonte](#));
 - macOS: 15%, Windows: 70%.
- Supporto hardware non ottimale.
 - per alcune periferiche non esistono driver per Linux o non sono ottimali;
 - la situazione è molto migliore che in passato, ma non è ancora perfetta.

- Poco usato in ambito personale.
 - il market share di Linux sui computer desktop si attesta intorno al 6%, contando anche ChromeOS che è una versione di Linux che gira sui Chromebook ([fonte](#));
 - macOS: 15%, Windows: 70%.
- Supporto hardware non ottimale.
 - per alcune periferiche non esistono driver per Linux o non sono ottimali;
 - la situazione è molto migliore che in passato, ma non è ancora perfetta.
- Mancano alcuni tipi di applicazioni.
 - non c'è una grande disponibilità di videogiochi;
 - molti produttori non sviluppano software per Linux (es. Adobe)