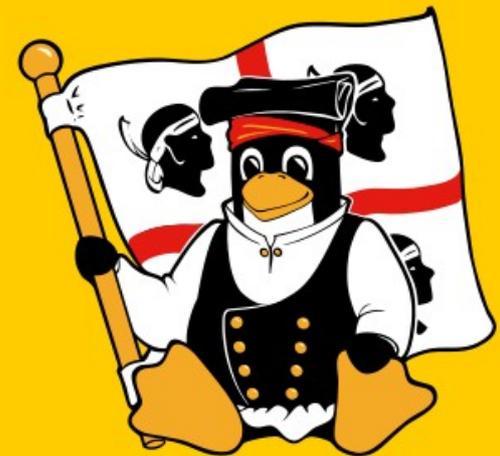


Software Libero - What to know

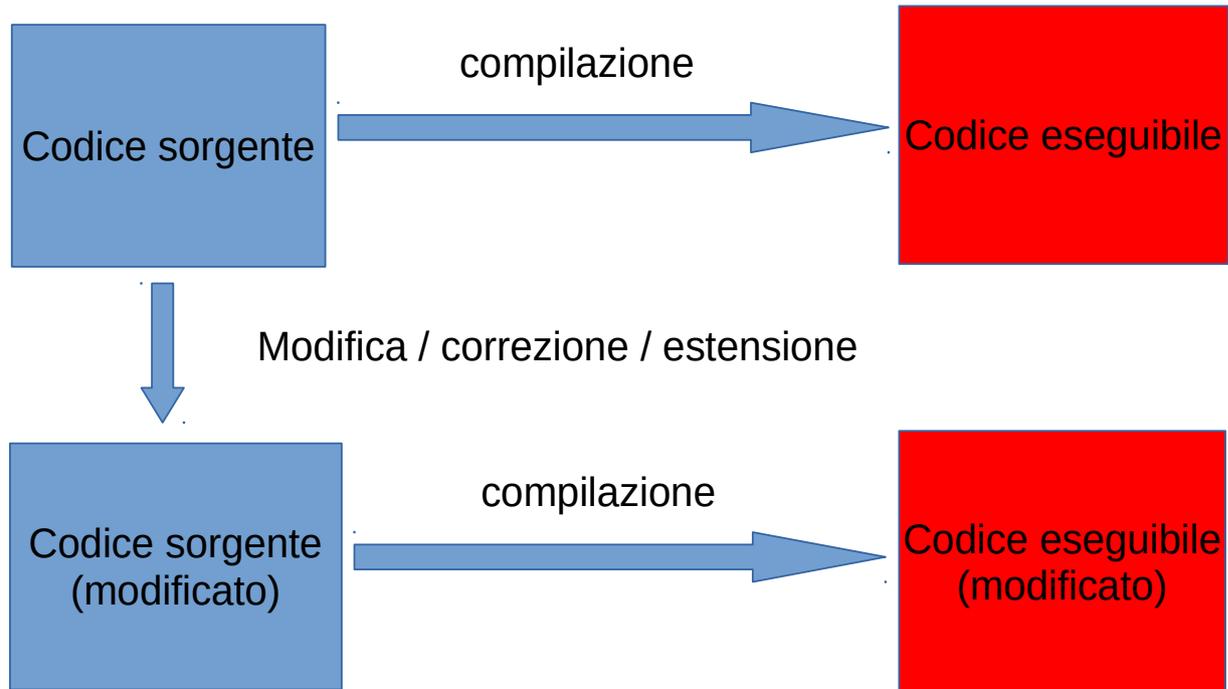
- Di Nicola Diego Torraco



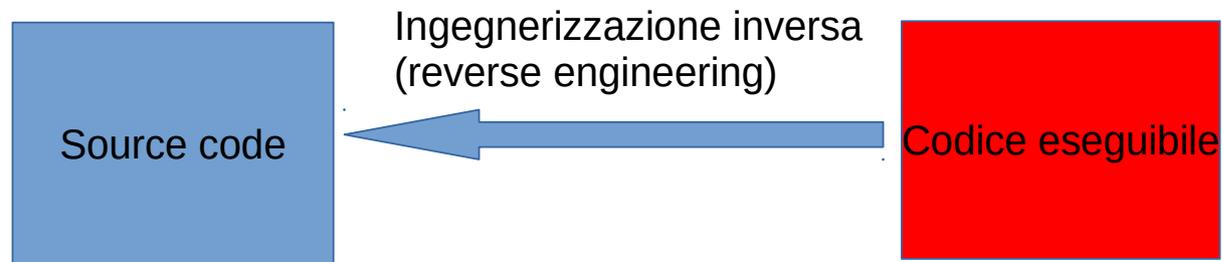
GULCh

Gruppo Utenti Linux Cagliari h...?

Codice sorgente vs codice binario



Il processo inverso non sempre è possibile!



Codice sorgente vs codice binario

- Il programma eseguito da un computer è un cosiddetto **binario (eseguibile)**, ovvero una sequenza di istruzioni elementari eseguiti dai processori del computer
- Il codice binario è specifico di una particolare architettura o sistema operativo, cioè non è eseguibile su una architettura hardware o sistema operativo diverso
- Nei moderni sistemi di sviluppo (a partire dagli anni '70 circa) il codice binario è prodotto mediante appositi strumenti (editor di testo, ambienti di sviluppo, compilatori, etc.) a partire da codice in linguaggi di programmazione “di alto livello”, detto **codice sorgente (source code)**
- La procedura inversa di conversione di un binario al codice sorgente di partenza è impossibile



Codice sorgente vs codice binario

- Senza il codice sorgente (e appositi strumenti di sviluppo), non è possibile facilmente studiare e modificare il programma, né è possibile rigenerarlo per renderlo eseguibile su altre piattaforme/sistemi operativi
- Nota 1: in alcuni casi il codice prodotto dalla compilazione è portabile nel senso di essere eseguibile da una VM (esempio il caso di bytecode), ma anche in questo caso non è possibile modificarlo
- Nota 2: molti linguaggi cosiddetti di **scripting** sono interpretati (es. Perl, Python, JavaScript), in questo caso il software non deve essere convertito a una forma binaria



Cos'è il Software Libero (secondo Stallman)

Un programma si dice libero se fornisce all'utente le seguenti quattro libertà:

0. Libertà di **eseguire** il programma, per qualsiasi scopo.
1. Libertà di **studiare** come funziona il programma e adattarlo alle proprie necessità. L'accesso al codice sorgente è un prerequisito.
2. Libertà di **ridistribuire** copie del programma originario
3. Libertà di **modificare** il programma e distribuirne **pubblicamente le modifiche**, in modo tale che tutta la comunità ne tragga beneficio.

Libero non significa necessariamente gratuito.



Software libero/Open Source/FLOSS

- **Freeware (Software)**: solitamente viene rilasciato il software gratuitamente, ma non il codice sorgente
- **Open Source Software (a sorgente aperto)**: letteralmente “a sorgente aperto”, ovvero viene distribuito anche il software sorgente (vincolato da apposite licenze). È il termine utilizzato dall'OSI (<https://opensource.org/>)
- **Free Software (libero)**: termine prevalente usato nel movimento GNU (enfasi sulle implicazioni etiche del software libero, rispetto dei 4 principi fondamentali)
- **FLOSS**: Free/Libre Open Source Software, termine ibrido con cui si accomuna sia il software Open-Source che quello “libero”



Software proprietario

- Software rilasciato **in forma solo eseguibile**, senza accesso al codice sorgente che lo ha prodotto (può essere rilasciato in forma gratuita o a pagamento)



Licenze

- Le licenze regolano la proprietà intellettuale del codice sorgente.
- Un progetto software solitamente adotta una o più licenze per la condivisione del software.
- Alcune delle licenze:
 - MIT
 - MPL (Mozilla Public License)
 - BSD
 - Apache
 - GNU GPL v.2, LGPL v2, GPL v3



Licenza GPL e viralità

- La licenza GNU GPL (GNU General Public License) è disponibile in varie versioni. Le più diffuse sono la GPL/LGPL v2 (Giugno 1991), v3 (Giugno 2007). Vedi <https://www.gnu.org/licenses/>.
- **No warranties** (assenza di garanzia da parte del produttore, cioè le conseguenze negative dell'utilizzo non sono imputabili all'autore del codice)
- Garantiscono le 4 libertà fondamentali (diritto di esecuzione, di studio dei sorgenti, diritto di copia, diritto di redistribuzione delle modifiche) – copy-left
- Impongono l'adozione della stessa licenza al codice derivato (**viralità**).
- Il codice con licenza GPL trasmette la proprio licenza anche al codice restante del programma (vincolo meno forte con uso di LGPL)



Sviluppo del software libero - alcune implicazioni

- I progetti di software tendono ad avere una **struttura di sviluppo centralizzata**, con un gruppo di sviluppatori chiave che gestiscono l'infrastruttura di amministrazione e i contributi inviati dalla comunità (ad es. sotto forma di patch)
- Processo iterativo di **discussione e revisione** dei contributi, le modifiche possono essere accettate o scartate
- In alcuni casi si creano diversi cosiddetti “**fork**” (letteralmente “biforcazione”) di un progetto, dove un gruppo di sviluppatori il codice di un progetto in modo indipendente dal progetto di partenza (es.: OpenOffice => LibreOffice)
- Esistono diversi **servizi per la pubblicazione e gestione** del codice sorgente (es.: sourceforge, github)



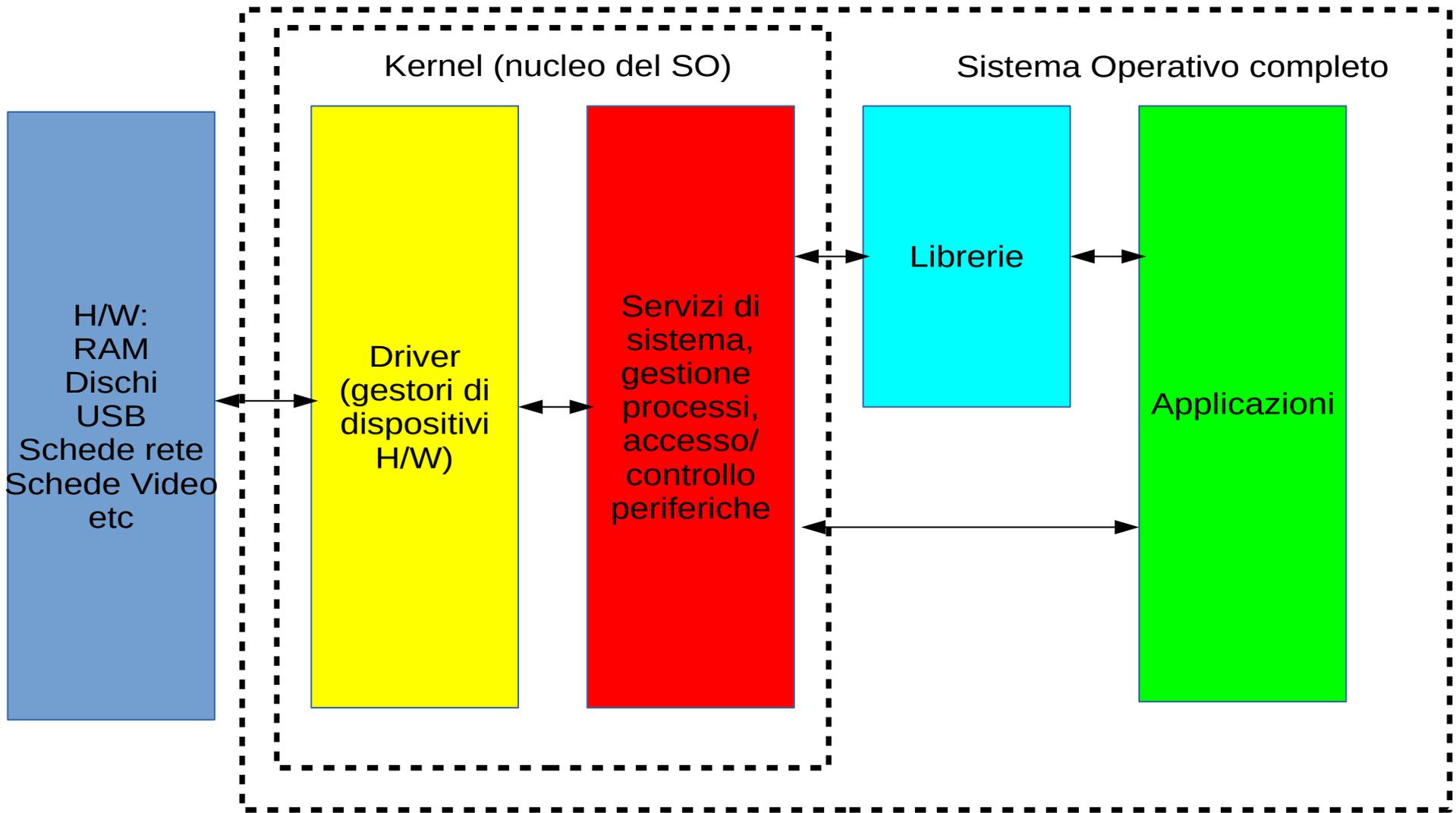
Il progetto GNU



- GNU = GNU is Not Unix
- Progetto fondato da Richard Stallman (a capo del movimento GNU)
- Progetto collaborativo nato nel 1983 per la creazione di un sistema operativo composto esclusivamente da software libero.
- Attualmente identifica un insieme di strumenti che sono stati adottati da vari sistemi operativi.
- Esempi: bash, GCC, Emacs, GIMP



Architettura di un Sistema Operativo



Il progetto Linux



- Nasce come progetto hobbistico, come implementazione di uno UNIX su un PC, sviluppato dallo studente finlandese **Linus Torvalds** a partire dal 1991
- Il nome “Linux” nasce da una fusione tra il nome di Linus e il sistema operativo “Unix/Minix”
- Evoluzione del sistema operativo didattico MINIX
- I sorgenti vengono rilasciati, lo sviluppo avviene mediante Internet tramite lo scambio di patch sulla mailing-list del progetto
- Attrae un notevole numero di sviluppatori, e viene adottato per completare il sistema sviluppato dal gruppo GNU. L'insieme viene chiamato solitamente sistema **GNU/Linux**.



Il progetto Linux

- Il kernel Linux, iniziato da Linus Torvalds, è stato rilasciato come codice sorgente liberamente modificabile nel 1991.
- La prima licenza è una licenza di software proprietario. Tuttavia, con la versione 0.12, nel febbraio 1992, pubblicarono il progetto sotto la GNU General Public License.
- Essendo molto simile a Unix, kernel di Torvalds ha attirato l'attenzione di programmatori volontari.
- Nel 1998 è stata inoltre creata un'organizzazione, l'Open Source Initiative, che fornisce licenze libere.



Linux - il kernel del sistema operativo

GNU/Linux

- Linux non è un sistema operativo completo, ma ne è la parte centrale, il **kernel** (nucleo).
- Lo sviluppo è sostenuto dalla Linux Foundation, associazione senza scopo di lucro nata nel 1997.
- Viene attualmente utilizzato in dispositivi completamente differenti:
 - PC, Tablet e Console
 - Cellulari e smartphone (Android OS)
 - Server, Mainframe e supercomputer
 - Router, sistemi embedded



Distribuzioni GNU/Linux

- Una **distribuzione** linux (o distro) fornisce un sistema per l'installazione di un sistema operativo basato su GNU/Linux e per la gestione di varie componenti software supportate dal sistema.
- L'installazione avviene solitamente mediante una memoria di supporto (un CD oppure una memoria esterna USB). Spesso è possibile eseguire la distribuzione in prova in modalità “live”.
- L'installazione di una distro Linux può coesistere nello stesso computer con altri sistemi operativi, ad esempio Windows (**dual boot**).
- Le varie componenti software sono installate tramite un **sistema di gestione dei pacchetti software** (packet manager), e un repository (magazzino) remoto centralizzato che contiene tutti i “pacchetti”.



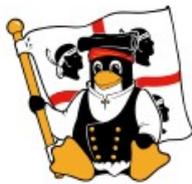
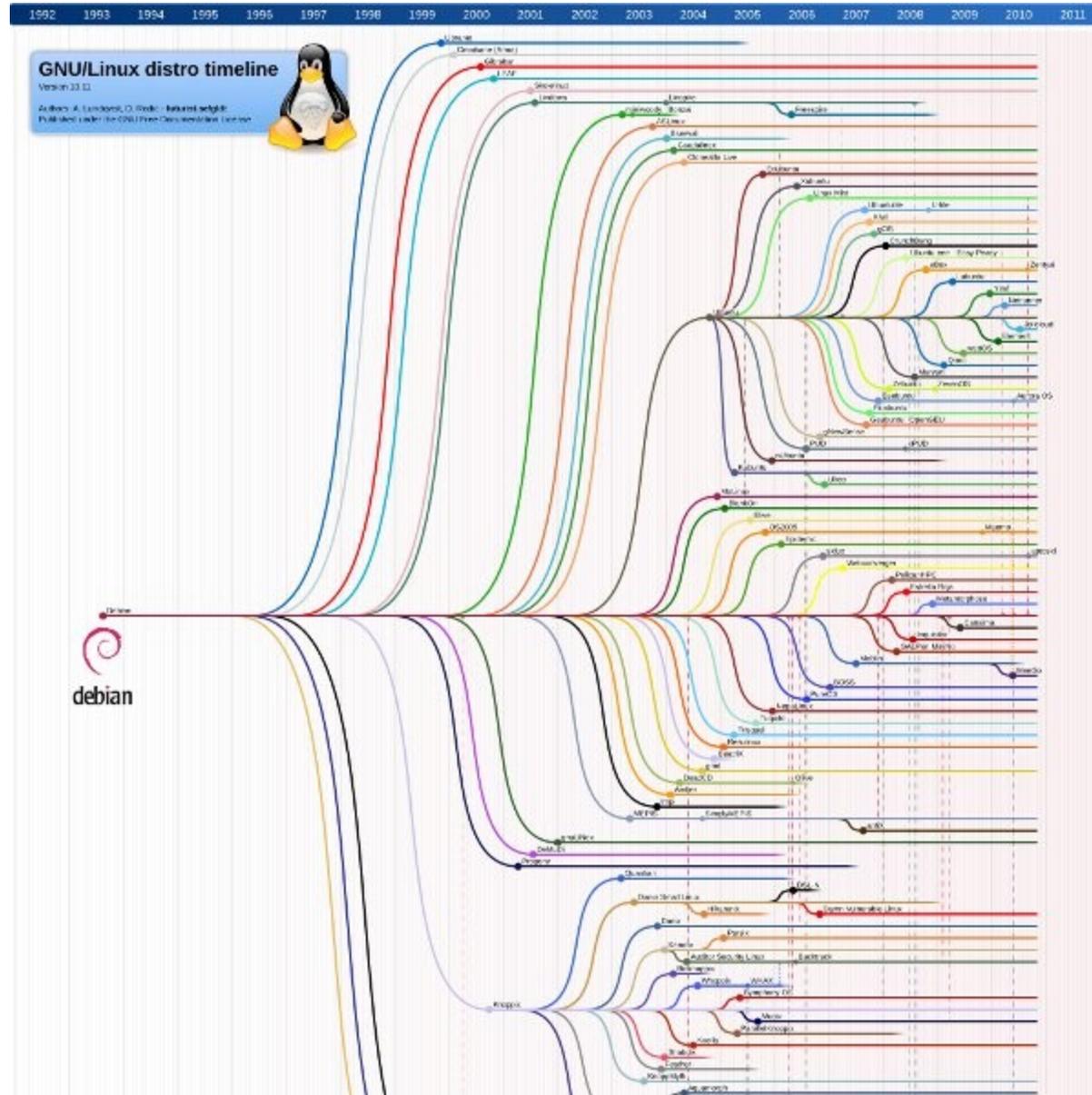
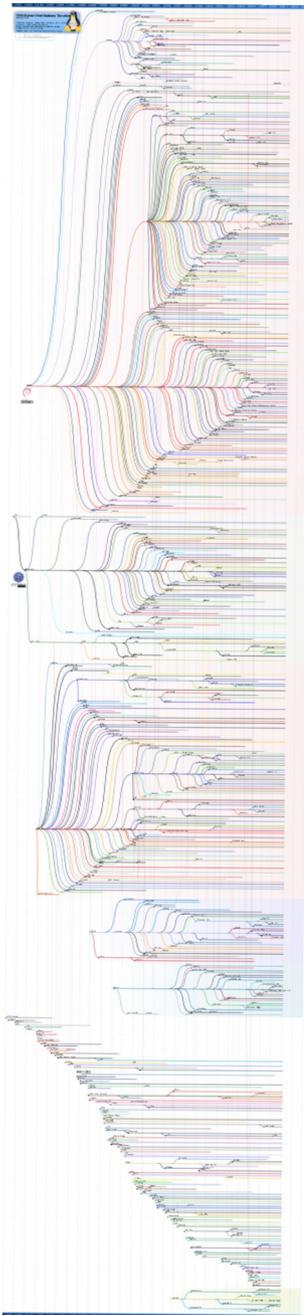
Distribuzioni GNU/Linux

Esistono centinaia di distribuzioni di Linux, ognuna caratterizzata da diversi ambienti grafici, diverse collezioni di pacchetti software e diversi requisiti minimi.

- Debian
- Linux mint
- Ubuntu e varianti (Kubuntu, Lubuntu, Xubuntu)
- Kali Linux
- Slackware
- Fedora/CentOS
- RedHat (supporto commerciale fornito da azienda omonima)
- Centinaia dedicate a compiti specifici (multimedia, embedded, networking, hacking, etc.)



Distribuzioni GNU/Linux



Altri kernel liberi non-Linux

- BSD (Berkeley Software Distribution) e varianti:
 - FreeBSD
 - NetBSD
 - OpenBSD
 - DarwinOS (alla base di Mac OS)
- GNU Mach
- Plan 9 from System labs
- Open Solaris
- Etc. etc.



Esempi di Software libero - sviluppo del software

- **GCC** – GNU C compiler – compilatore C
- **LLVM**
- **Python / Perl / PHP / Ruby / Java** etc. - la maggior parte dei linguaggi di programmazione in uso fornisce strumenti di compilazione e di sviluppo basati sul software libero
- Ambienti di sviluppo: **emacs, vim, ECLIPSE, notepad++**
- **Migliaia** di librerie/moduli/estensioni nei vari linguaggi



Codice sorgente e codice binario - esempio in linguaggio C

```
nicola@asus ~> cat > ciao-linuxday.c
```

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

```
int main(void)
```

```
{
```

```
    printf("Ciao linuxday 2018!");
```

```
    return 0;
```

```
}
```

```
nicola@asus ~> file ciao-linuxday.c
```

```
ciao-linuxday.c: C source, ASCII text
```

```
nicola@asus ~> gcc ciao-linuxday.c -o ciao-linuxday
```

```
nicola@asus ~> ./ciao-linuxday
```

```
Ciao linuxday 2018!↵
```

```
nicola@asus ~> file ciao-linuxday
```

```
ciao-linuxday: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (uses  
shared libs), for GNU/Linux 2.6.24,
```

```
BuildID[sha1]=14be8008af3a0331d68b7b351634309470f3e6bb, not stripped
```

```
nicola@asus ~> ls -lh ciao-linuxday
```

```
-rwxr-xr-x 1 nicola torraco 8.4K Oct 15 17:18 ciao-linuxday*
```



Esempi di software libero - suite per Ufficio e grafica

- **OpenOffice/LibreOffice** – suite di strumenti per la scrittura di documenti (Writer - documenti di testo, Impress – presentazioni, Calc – fogli di calcolo, Draw – editor di diagrammi)
- **Dia**
- **Abiword**
- **Calligra** (ex Koffice)
- **Scribus** (creazione di contenuti per la pubblicazione di prodotti stampati – riviste, brochure, volantini etc.)
- **Tex/LaTex** (sistema di composizione tipografica, usato specialmente in ambito accademico)



Esempi di software libero - programmi di grafica

- **GIMP** (grafica raster, elaborazioni immagini digitali)
- **Krita** (grafica raster)
- **Blender** (grafica e animazione di modelli 3D)
- **Inkscape** (grafica vettoriale)
- **LibreCAD** (grafica CAD)



Esempi di software libero - Navigatori web

- **Firefox:** sviluppato dalla Mozilla Foundation
- **Chromium:** sviluppato da Google, alla base di **Chrome**, con delle componenti non-libere aggiuntive (codec audio/video, supporto per Flash)



Esempi di software libero: audio e video

- **VLC** (player multimediale)
- **FFmpeg** (librerie e strumenti di conversione A/V)
- **Kodi** (software Media Center)
- **Kdenlive** (editor video)
- **Amarok** (riproduttore audio)



Esempi di software libero - Database

- **MySQL**
- **MariaDB**
- **Cubrid**
- **FirebirdSQL**
- **SQLite**
- **PostgreSQL**
- **MongoDB** (noSQL database)
- **Neo4j** (graph database)



Esempi di software libero - Sviluppo web (JavaScript)

- **jQuery**
- **AngularJS**
- **Bootstrap**
- **ReactJS**
- **Bower**
- **RequireJS**
- **Node.js** (sviluppo di applicazione server in Javascript)
- **emscripten**



Esempi di software libero - virtualizzazione ed emulazione

- **Msys2/MINGW64** (fornisce un ambiente di sviluppo UNIX-like su Windows: in particolare rende possibile il porting di programmi sviluppati su Linux/Unix su Windows)
- **Virtualbox** (sistema di virtualizzazione macchine all'interno della macchina ospite)
- **WINE - Wine Is Not an Emulator** (rende possibile l'esecuzione di programmi binari Windows in ambienti Linux, con alcune limitazioni)
- **DOSBox** (emulatore DOS)



Motivazioni diffusione del software libero

- Reperibilità universale, e software disponibile principalmente in modo gratuito
- Forti comunità di utenti e di sviluppatori, strumenti di sviluppo
- Garanzie in termini di qualità (robustezza, sicurezza)
- Possibilità di adattare facilmente il software alle proprie esigenze



Conseguenze della diffusione del software libero

- **Progetti di software libero diventano standard de facto** dovuti all'impiego di soluzioni universalmente accessibili e con una forte comunità di utenti e sviluppatori
- Come tendenza i prodotti commerciali spesso si basano su uno **stack di progetti FLOSS** (e.g. Linux, mysql, Perl/Python/PHP, GCC, FFmpeg etc etc etc) sulla cui base o integrazione offrono servizi aggiuntivi
- Tendenza alla fornitura di **servizi** piuttosto che alla vendita di prodotti software, visto che il software per se diventa una “commodity”



Vantaggi nell'adozione di software libero

- Il processo di produzione del codice è solitamente **trasparente e tracciabile**, questo implica che:
 - **Garanzia di qualità** legata al processo di revisione tra pari (peer review)
 - il SW non potrà contenere backdoor o altri meccanismi che possano compromettere la sicurezza o la privacy dei dati o funzionalità parassite (monitoring, advertising)
 - I dati sono gestiti in modo trasparente e riutilizzabili da altri programmi analoghi
- È possibile modificare il codice in modo da apportare modifiche migliorative o adattive senza dipendere da chi ha prodotto originariamente il software



Vantaggi nell'adozione di software libero

- Disponibilità di software per il riutilizzo di hardware ritenuto obsoleto (trashware)
- In ambito scientifico permette la **riproducibilità** del processo di generazione dei dati
- Fornisce degli strumenti e un ambiente **aperto** per studenti o appassionati interessati allo studio dei principi di sviluppo dell'informatica
- L'ecosistema del software e dei servizi associati beneficia della diffusione di strumenti e di saperi permessi dalla libertà d'uso del software



Open information (informazione aperta)

- Gli stessi concetti legati alla redistribuzione del codice sorgente possono essere applicati alle informazioni generiche (in linguaggio umano o mediante supporti multimediali) nell'ambito di **progetti collaborativi**.
- L'organizzazione **Creative Commons** nasce nel 2001 con l'obiettivo di gestire i diritti delle informazioni condivise in progetti aperti
- Wikipedia adotta la CC per tutti i contenuti pubblicati a partire dal 2008



Creative Commons Public Licenses - Some rights reserved

- Estremi: Tutti i diritti riservati - nessuna protezione verso l'autore
- Rende possibile la scelta da parte dell'autore di **quali** diritti riservare all'opera prodotta **“Some rights reserved”**
- Protocollo di attribuzione degli attributi CC associati ad un'opera digitale - integrato / integrabile nei motori di ricerca dei contenuti (ad esempio: Google Images)
- Quattro condizioni: attribution (BY), share-alike (SA), non-commercial (NC), no derivative works (ND)
- Le condizioni possono essere combinate: e.g. CC-BY-SA, CC-BY-NC-ND



BY



SA



NC



ND



Creative Commons Public License

L'ultima versione della CC (la 4.0) è stata rilasciata nel 2013

Licenza	Può essere usato commercialmente?	Si possono creare nuove versioni?
 Attribuzione	Sì	Sì
 Attribuzione - Condividi allo stesso modo	Sì	Sì, ma la versione creata deve avere una nuova licenza "Condividi allo stesso modo"
 Attribuzione - Non opere derivate	Sì	No
 Attribuzione - Non commerciale	No	Sì, ma la versione creata deve avere una nuova licenza "Non commerciale"
 Attribuzione - Non commerciale - Condividi allo stesso modo	No	Sì, ma la versione creata deve avere una nuova licenza "Non commerciale" e "Condividi allo stesso modo"
 Attribuzione - Non commerciale - Non opere derivate	No	No

Copyright Wikipedia

Viralità della licenza



Contatti relatore



nicola.diego@torraco.it



<https://www.linkedin.com/in/nicolatorraco/>



@NicolaTorraco



nicolatorraco

GRAZIE



Licenza

Rilasciato con licenza Creative Commons

Attribution-NonCommercial-ShareAlike, vedi:

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

