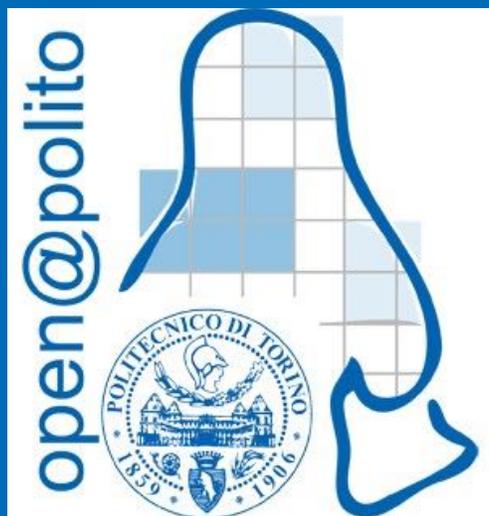




CORSO GNU/LINUX

Con il supporto di:



L'importanza delle
licenze software.

WHOAMI

Ciao sono Daniele “**DLx**” Lussana.

Sono membro dell’associazione del Politecnico di Torino “**Netstudent**”.

Credo nel ”sapere libero e comune” e nel “software libero” e “opensource”.

Sono un utente **GNU/LINUX**.



WHAT

GNU/Linux è un **sistema operativo**.

In virtù della licenza software che lo accompagna, è un software definito **libero**.

Il suo **codice sorgente è disponibile per tutti**, ognuno può modificarlo a proprio piacimento per soddisfare le proprie esigenze, ma deve rendere pubbliche le modifiche che ha apportato.

Non è necessario pagare alcunché per poterlo utilizzare sul proprio computer.



software commerciale

VS

software libero

WHAT (segue)



Talvolta, si distingue tra...

“**free**” (as in “free beer”):

non è previsto un compenso per il suo utilizzo (gratuito);

“free” (as in “freedom”) oppure “**open-source**”:

è garantito l'accesso al codice sorgente.

Quando coesistono le due caratteristiche, si usa l'acronimo
F.O.S.S. o **F.L.O.S.S.**: (Free/Libre open source software)

WHO & WHEN



“**Linux**” è in realtà il nome del cuore del sistema operativo GNU/Linux (**kernel**), e deriva dal nome del suo creatore ed attuale manutentore **Linus Torvalds**.

Torvalds, un programmatore finlandese, frequentando l'università, nel 1991 iniziò a creare un **clone di UNIX** detto “Freax” (questo il nome iniziale del progetto).

WHO & WHEN (SEGUE)

GNU è invece il progetto voluto da **Richard “RMS” Stallman** (a partire dal 1983) per la creazione di un sistema operativo completamente libero, chiamato appunto Sistema GNU (acronimo per “GNU is Not Unix”).

Stallman è stato il fondatore della **Free Software Foundation**.



WHAT (di nuovo...)



GNU

(comandi di base)

+

Linux

(kernel)

I due progetti sono quindi realtà complementari

WHY

«I started the free software movement to replace user-controlling non-free software with freedom-respecting free software.

With free software, we can at least control what software does in our own computers.»

WHY

L'importanza dell'aspetto “licenza”:

- 1) Richard Stallman iniziò il “progetto GNU” per questioni relativi alle licenze software;
- 2) Linus Torvalds iniziò a scrivere il Kernel Linux perché voleva un S.O. simile a UNIX ma “Free”

WHY

La licenza d'uso è il documento che accompagna il software e specifica i **diritti** e i **doveri** di chi lo riceve e di chi lo diffonde.

Come visto, esistono licenze libere e licenze proprietarie.

Le licenze di utilizzo e distribuzione del software libere (Open Source) sono numerose, ma quelle effettivamente diffuse sono poche: GPL (la licenza di Gnu-Linux), LGPL e BSD

Ha senso dire di aver “acquistato” un software?

Software realizzato su misura (“chiavi in mano”)

(appalto di servizi, diventa “mio”)

Software ceduto (dal programmatore alla software house)

(diventa... della software house)

Software concesso in licenza

(io acquisto solo la facoltà di usarlo, con esclusioni)

Contenuto di una tipica licenza commerciale

Limitazione di responsabilità del concedente per danni

Divieto di incorporare o riutilizzare il programma

Divieto di modificarlo

Divieto di creare opere derivate

Divieto di operazioni di decompilazione e disassemblaggio

Divieto di “sublicenziare” il prodotto o limite a tale facoltà

The Four freedoms (FSF: “What is Free Software”)

«A program is “free software” if the program’s users have the four essential freedoms:

The freedom to **run** the program as you wish [...].

The freedom to **study** how it works, and change it [...]

The freedom to **redistribute copies** [...]

The **freedom** to distribute copies of your modified versions **to others**. [...]»

Quale licenza usare



Per impedire ad altri di accedere al codice?

Una licenza proprietaria.

Ispiratevi al testo delle licenze MS e Apple.

Quale licenza usare



Per consentire di creare opere derivate del mio software?

GPL: è una licenza “virale” (specie nella **v.3**): obbliga a rilasciare le opere derivate con la medesima licenza.

Quale licenza usare



Per consentire di creare opere derivate del mio software,
senza obbligare a rilasciarle con la medesima licenza?

BSD(-like): consente a terze parti di riutilizzare il codice sotto
diversa licenza (“per qualsivoglia fine”)

Quale licenza usare



Per consentire di linkare il mio software libero (es: una libreria) ad altro software che abbia licenze diverse (magari proprietarie)

LGPL: permette ai terzi di “linkare” il software (generalmente librerie) ad altro software con licenza diversa

Quale licenza usare



Per pubblicare manuali del mio software libero

Esiste la **Gnu Free Documentation Licence (GFDL)**,

perché se i manuali fossero pubblicati sotto GPL costretto bisognerebbe includere il "codice sorgente" o, in assenza di questo offrire di inviare al lettore una copia di questo.

Quale licenza usare



Per essere sicuri di aver accesso al codice sorgente anche per un software eseguito lato server (e non distribuito sulle macchine degli utenti)

Affero GPL: se in un server viene eseguito un programma modificato e si permette che gli utenti comunichino con esso, il server deve anche consentire agli utenti di scaricare il codice sorgente della versione modificata.

Quale licenza usare



Se voglio una licenza modulare, che mi permetta di scegliere quali diritti attribuire e quali escludere

(es.: voglio escludere che vengano fatte opere derivate e l'utilizzo commerciale)?

Licenze Creative Commons

<https://creativecommons.it/chapterIT/>

Quale licenza usare



Per altre opere (grafica, musica, opere letterarie) che vorrei fossero “libere” o “copyleft”, con la certezza che sia compatibili con le regole del mio paese?

Licenze Creative Commons

<https://creativecommons.it/chapterIT/>

Le slide contengono esclusivamente opinioni personali del loro autore (che potrebbero non essere condivise dal Politecnico di Torino)

Rilasciate con licenza
Creative Commons
CC-BY-SA (3.0 IT)

