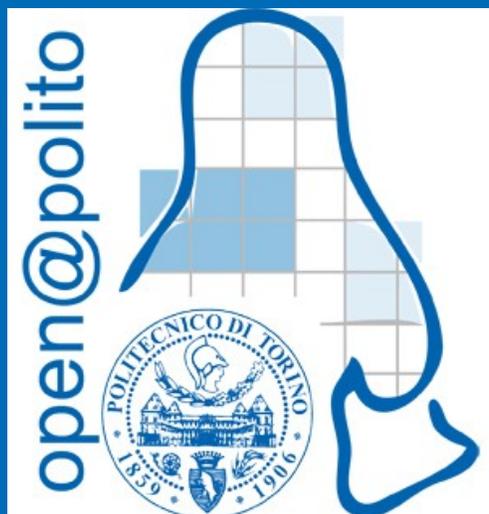




Con il supporto di:

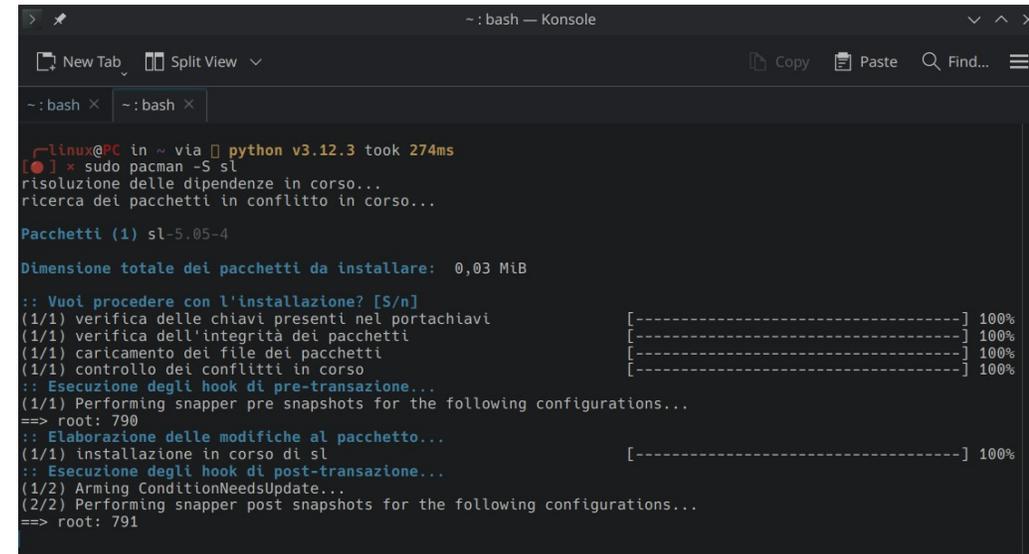


Gestione dei pacchetti

Di Francesco Bergesio,
Con aggiunte di Filippo Giacomini

Metodi di installazione su Linux

- Gestore dei pacchetti (apt, snap, pacman, ...)
- File di installazione (deb, rpm, ...)
- Compilazione:
 - Automatizzata (AUR, DUR)
 - Manuale (attraverso git clone e make)



```
~ : bash — Konsole
New Tab Split View
Copy Paste Find...
~ : bash x ~ : bash x
linux@PC in ~ via python v3.12.3 took 274ms
x sudo pacman -S sl
risoluzione delle dipendenze in corso...
ricerca dei pacchetti in conflitto in corso...

Pacchetti (1) sl-5.05-4

Dimensione totale dei pacchetti da installare: 0,03 MiB

:: Vuoi procedere con l'installazione? [S/n]
(1/1) verifica delle chiavi presenti nel portachiavi [-----] 100%
(1/1) verifica dell'integrità dei pacchetti [-----] 100%
(1/1) caricamento dei file dei pacchetti [-----] 100%
(1/1) controllo dei conflitti in corso [-----] 100%
:: Esecuzione degli hook di pre-transazione...
(1/1) Performing snapper pre snapshots for the following configurations...
==> root: 790
:: Elaborazione delle modifiche al pacchetto...
(1/1) installazione in corso di sl [-----] 100%
:: Esecuzione degli hook di post-transazione...
(1/2) Arming ConditionNeedsUpdate...
(2/2) Performing snapper post snapshots for the following configurations...
==> root: 791
```

Glossario

Pacchetto: Un pacchetto è un'unità di distribuzione del software. Permette modularità del software.

Dipendenze: Una dipendenza è la necessità di un software di appoggiarsi ad un altro per funzionare (librerie, driver, ...).

Repository: I repository sono archivi web nei quali vengono raggruppati i pacchetti software installabili da un certo gestore.

Compilazione: Processo per il quale il codice sorgente viene trasformato in un programma da poter poi installare.

Gestore dei pacchetti APT

Vantaggi:

- I repository ufficiali sono tutto sommato sicuri.
- Comodità di installazione.
- Versioni dei pacchetti più stabili.
- Gestione automatica degli update

Svantaggi:

- Non è presente tutto (soprattutto versioni molto vecchie).
- I pacchetti potrebbero non avere le ultime funzioni.

Advanced **P**ackage **T**ool

Installazione APT

install: Installa uno o più pacchetti.

remove: Rimuove uno o più pacchetti.

purge: Rimuove uno o più pacchetti e i file di configurazione.

autoremove: rimuove le dipendenze che non sono più utili.

--reinstall: Reinstalla il pacchetto da capo (install).

list: elenca i pacchetti in base a dei criteri (--installed, --upgradable).

Aggiornamenti APT

update: Aggiorna i repository e controlla la presenza di nuovo software.

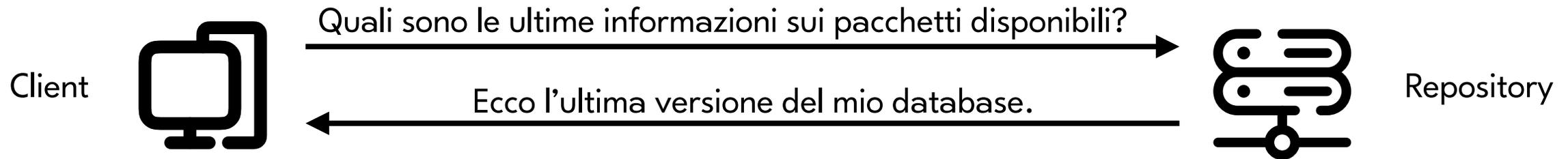
upgrade o **dist-upgrade:** Aggiorna i pacchetti che hanno una versione più recente nei repository.

clean: ripulisce la cache locale dei pacchetti.

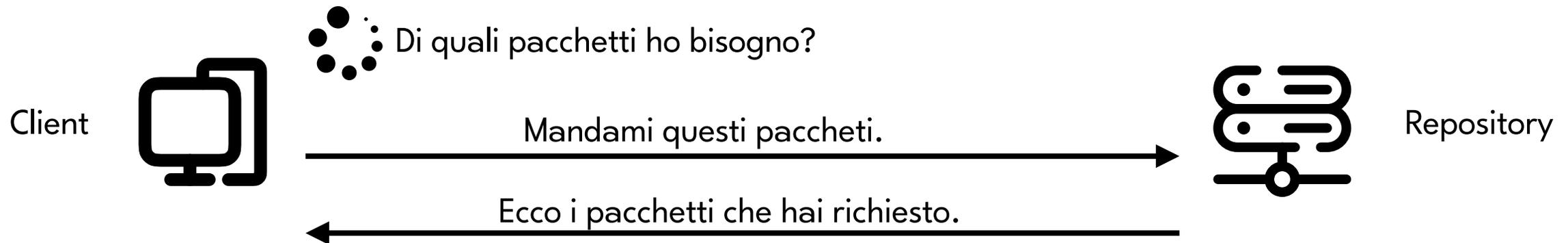
Di solito viene fatto prima **update** per scaricare le informazioni riguardanti i pacchetti dal repository e poi viene fatto **upgrade** per aggiornare.

Flusso di aggiornamento con gestore pacchetti

- 1) `sudo apt update`: Il client richiede le ultime informazioni sui pacchetti disponibili del repository.



- 2) `sudo apt upgrade`: Il client confronta le versioni dei pacchetti installati con quelle contenute nel database, poi scarica le nuove versioni dei pacchetti da aggiornare. Se ci sono nuove dipendenze richiede anche quelle.



File e cartelle interessanti

APT

`/etc/apt/sources.list` o `/etc/apt/sources.list.d/`: Contiene i repository a cui apt fa riferimento.

`/var/cache/apt/archives/`: Cartella in cui sono presenti i file della cache.

`/var/lib/apt/lists/lock`: File che serve ad avvertire che un'istanza di apt sta operando. Se il pc viene spento durante un'operazione rimane.

Modifica dei repository APT

`add-apt-repository repo`: Aggiunge un repository alla lista.

`add-apt-repository -r repo`: Rimuove il repository repo dalla lista.

Aggiungendo manualmente la riga del repository nel file o nella cartella in un file apposito.

```
deb http://repository.spotify.com stable non-free
```

(esempio di repository)

Per rendere effettive le modifiche bisogna usare `update`. **Problemi di sicurezza.**

	Free software	Non-free software
Supported	Main	Restricted
Unsupported	Universe	Multiverse

Esercizio

APT

- Installare il pacchetto `lolcat` e le sue dipendenze.

Provare ad installare il pacchetto `php5.6`.

- Aggiungere ai repository di `apt` deb creando un file nella cartella `/etc/apt/sources.list.d/` `ppa:ondrej/php`.
- Installare il pacchetto che non era stato possibile installare prima dal repository esterno.
- Disinstallare i programmi installati e le loro dipendenze.
- Rimuovere il repository aggiunto

Installazione tramite file (deb)

Vantaggi:

- Posso installare versioni più recenti o obsolete.
- Ancora più o meno sicuro se scaricato dai siti ufficiali.

Svantaggi:

- Bisogna cercare il file deb sul web.
- Potrebbero essere molto meno stabili se più nuovi.
- Potrebbero avere falle di sicurezza se molto meno recenti.
- Gli update non sono gestiti in maniera automatica da `apt`

DPKG

`-i` o `--install`: Installa un pacchetto deb.

`--configure`: Configura i pacchetti già estratti.

`--configure -a` (al posto di un pacchetto): Configura i pacchetti che sono rimasti appesi, utile se si è spento il pc durante un'installazione.

`-r` o `--remove`: Rimuove un pacchetto.

`-p` o `--purge`: Rimuove un pacchetto e le configurazioni connesse.

`dpkg-reconfigure`: riconfigura un pacchetto già installato.

Ancora meno sicuro.

Debian PaKaGe

Esercizio DPKG

- Scarica il file deb dal link <https://launcher.mojang.com/download/Minecraft.deb>
- Usare `dpkg` per installare il file deb.
- Riconfigurare il pacchetto appena installato.
- Rimuovi il programma appena installato e le sue dipendenze.

Compilazione manuale di un programma

Vantaggi:

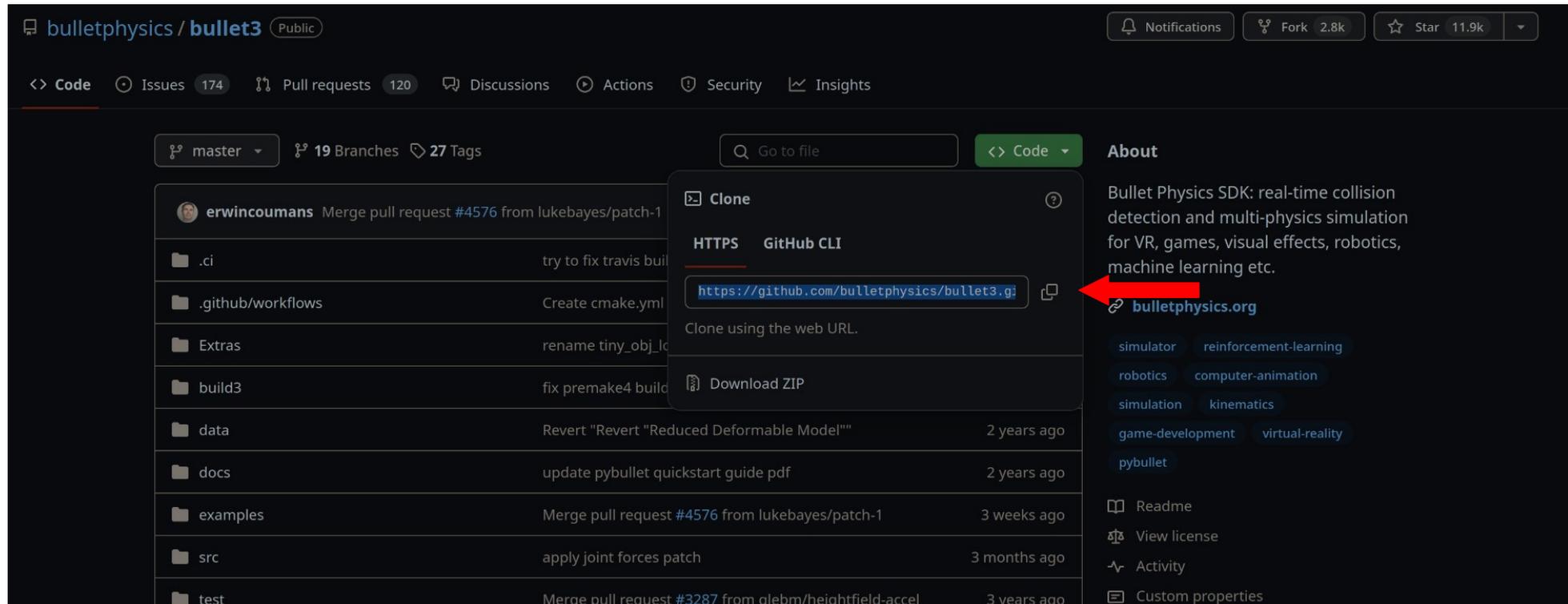
- Ultimissima versione del programma.

Svantaggi:

- Molto suscettibile a bug e **problemi di sicurezza**.
- Per aggiornare bisogna riscaricare e ricompilare il codice.
- I metodi di compilazione possono variare da programma a programma.
- Tutte le dipendenze devono essere installate manualmente e potrebbero essere altri pacchetti da compilare.
- Non sempre è presente uno script per disinstallare.

Compilazione manuale di un programma GIT CLONE

`git clone`: Ci permette di clonare un repository di `git` utilizzando il link fornito dalla pagina web dedicata.



The screenshot shows the GitHub interface for the repository 'bulletphysics/bullet3'. The 'Clone' dropdown menu is open, displaying the following options:

- Clone (with a help icon)
- HTTPS (selected)
- GitHub CLI
- Clone using the web URL.
- Download ZIP

The HTTPS URL `https://github.com/bulletphysics/bullet3.git` is highlighted in blue, and a red arrow points to the copy icon next to it. The repository page also shows the 'About' section with the description: 'Bullet Physics SDK: real-time collision detection and multi-physics simulation for VR, games, visual effects, robotics, machine learning etc.'

Compilazione manuale di un programma

Nel file `README.md`, che è lo stesso visualizzato nella parte inferiore, di solito c'è la guida per compilare, installare ed usare il programma scaricato con `git clone`.
Assieme al file `LICENSE.txt` dove è presente la licenza con cui è distribuito il software.

```
README License

The files in this repository are licensed under the zlib license, except for the files under 'Extras' and examples/ThirdPartyLibs.

Bullet Continuous Collision Detection and Physics Library
http://bulletphysics.org

This software is provided 'as-is', without any express or implied warranty.
In no event will the authors be held liable for any damages arising from the use of this software.
Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it freely,
subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software.
If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is
not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original
software.
3. This notice may not be removed or altered from any source distribution.
```

```
README License

Linux and Mac OSX gnu make

Make sure gcc and cmake is installed ( sudo apt-get install build-essential and sudo apt-get install
cmake for Linux, brew install cmake for Mac, or https://cmake.org)

In a terminal type:

./build_cmake_pybullet_double.sh

This script will invoke cmake and build in the build_cmake directory. You can find pybullet in Bullet/examples/
pybullet. The BulletExampleBrowser binary will be in Bullet/examples/ExampleBrowser.

You can also build Bullet using premake. There are premake executables in the build3 folder. Depending on
your system (Linux 32bit, 64bit or Mac OSX) use one of the following lines Using premake:

cd build3
./premake4_linux --double gmake
./premake4_linux64 --double gmake
./premake4_osx --double --enable_pybullet gmake

Then

cd gmake
make

Note that on Linux, you need to use cmake to build pybullet, since the compiler has issues of mixing shared
and static libraries.
```

Esercizio

Compilazione manuale

- Clona il repository di htop con `git clone` al link <https://github.com/htop-dev/htop>
- Seguire il file `README.md` per installare le dipendenze, compilare e installare il programma.
- Usare `make uninstall` per rimuovere il programma appena installato.

Compilazione Automatica

AUR

AUR è un repository pubblico dove vengono caricati i PKGBUILD.

I PKGBUILD contengono le istruzioni per compilare programmi da sorgenti o da pacchetti già esistenti e installarli con `pacman` (solo per distribuzioni basate su Arch Linux).

Essendo un repository gestito dagli utenti è molto facile imbattersi in **codice malevolo** quindi **bisogna fare attenzione**.

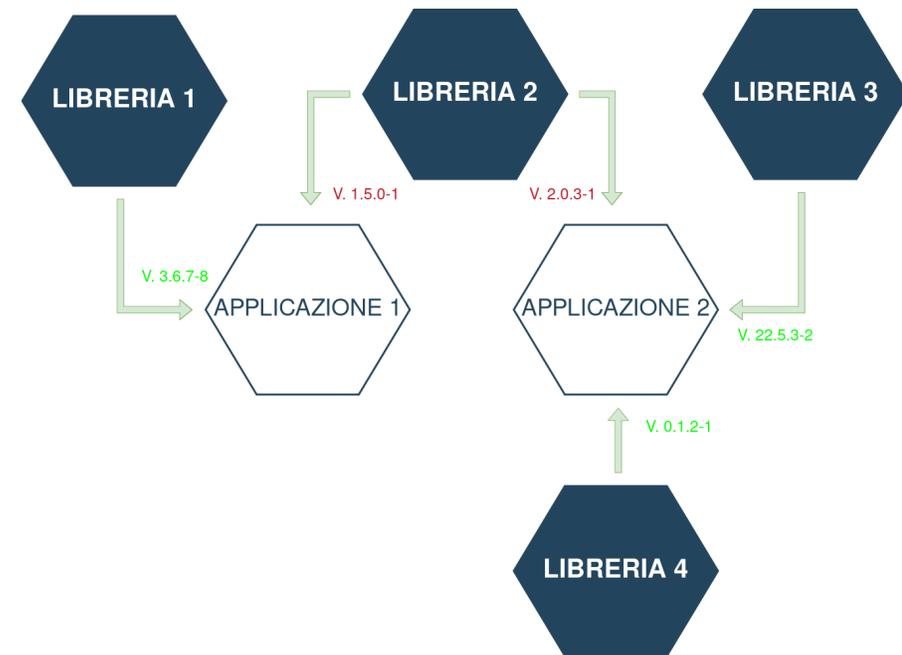
Un progetto simile è DUR (Debian User Repository), ma non ci sono più molte notizie.

Arch User Repository

Problema delle dipendenze

Può succedere che due software usino diverse versioni di librerie, anche molto distanti, ma su una distribuzione ne può venir installata una sola versione.

Se le due versioni non sono compatibili, non posso far funzionare entrambi i software contemporaneamente.



Dependency hell

🌐 10 languages ▾

Article [Talk](#)

Read [Edit](#) [View history](#) [Tools](#) ▾

From Wikipedia, the free encyclopedia

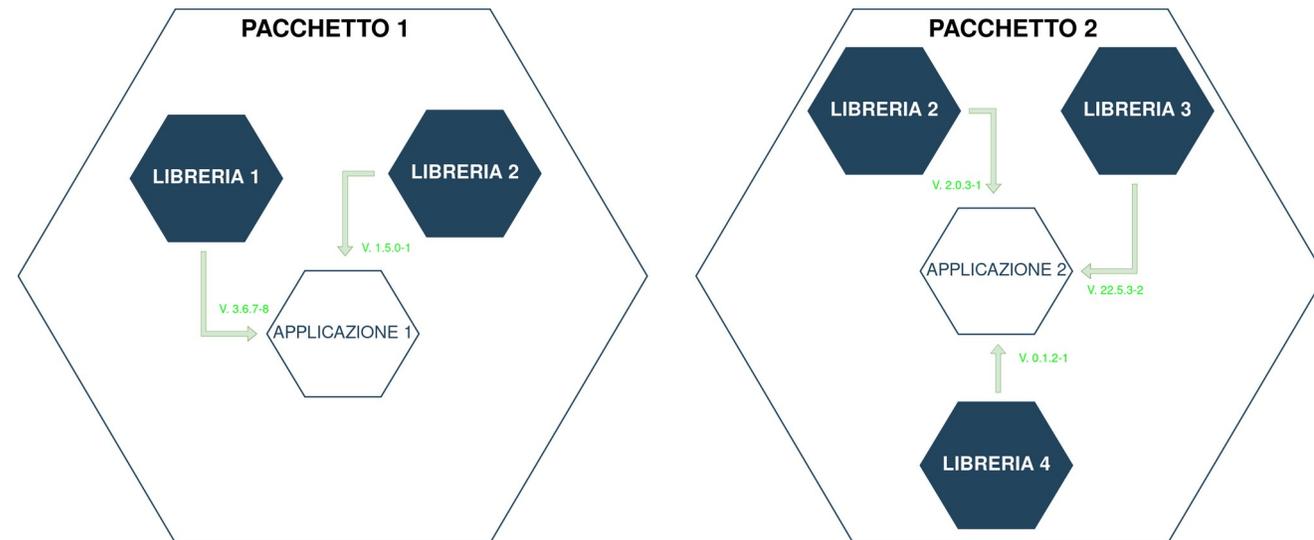
Dependency hell is a [colloquial term](#) for the frustration of some software users who have installed [software packages](#) which have [dependencies](#) on specific [versions](#) of other software packages.^[1]

The dependency issue arises when several packages have dependencies on the same *shared* packages or libraries, but they depend on different and incompatible versions of the shared packages. If the shared package or library can only be installed in a single version, the user may need to address the problem by obtaining newer or older versions of the dependent packages. This, in turn, may break other dependencies and push the problem to another set of packages.

https://en.wikipedia.org/wiki/Dependency_hell

Problema delle dipendenze

Per questo motivo si è tentato di pacchettizzare le applicazioni direttamente con le proprie dipendenze in modo da poterle eseguire in maniera standalone e distribuire il tutto in un unico pacchetto. Applmage è un formato di packaging che fa proprio questo.



Cenni su altri metodi di gestione dei pacchetti (SNAP)

Vantaggi:

- Installa le applicazioni con le librerie necessarie.
- Permette l'isolamento dei programmi.
- Permette la presenza di molteplici versioni del software.
- Si può usare anche per IoT.

Svantaggi:

- Aumento dello spazio necessario per ogni programma.
- Lo startup del programma è più lento.
- Ha parte del sorgente non open source.
- Sviluppato da un'azienda (Canonical).

Utilizzare Snap

`snap find <termini di ricerca>`: cerca uno snap nei repository.

`snap install <nome snap>`: installa lo snap richiesto.

`snap install --dangerous <percorso/allo/snap>`: installa un pacchetto dal disco.

`snap remove <nome snap>`: rimuovi uno snap.

`snap list`: mostra gli snap installati.

`snap refresh`: aggiorna manualmente gli snap.

Snap non ha comandi per gestire i repository perché ne ha solo uno ed è preinstallato.

Cenni su altri metodi di gestione dei pacchetti (FLATPAK)

Vantaggi:

- Installa le applicazioni con le librerie necessarie.
- Permette l'isolamento dei programmi.
- Permette la presenza di molteplici versioni del software.
- Sviluppato dalla community.

Svantaggi:

- Aumento dello spazio necessario per ogni programma.
- Solo per applicazioni desktop e limitatamente server.

Utilizzare Flatpak

`flatpak search <termini di ricerca>`: cerca un pacchetto nei repository.

`flatpak install <nome pacchetto>`: installa un pacchetto.

`flatpak uninstall <nome pacchetto>`: rimuovi un pacchetto. Aggiungi `--delete-data` per rimuovere anche i dati.

`flatpak uninstall --unused`: rimuovi i pacchetti non utilizzati.

`flatpak list`: mostra i pacchetti e i runtime installati.

Gestire i repository di Flatpak

`flatpak remote-add <nome> <indirizzo>`: aggiungi il repository all'indirizzo fornito.

`flatpak remote-delete <nome>`: rimuovi un repository.

`flatpak remotes`: mostra tutti i repository.

`flatpak remote-ls <nome>`: mostra tutti i pacchetti forniti dal repository.

ApplImage

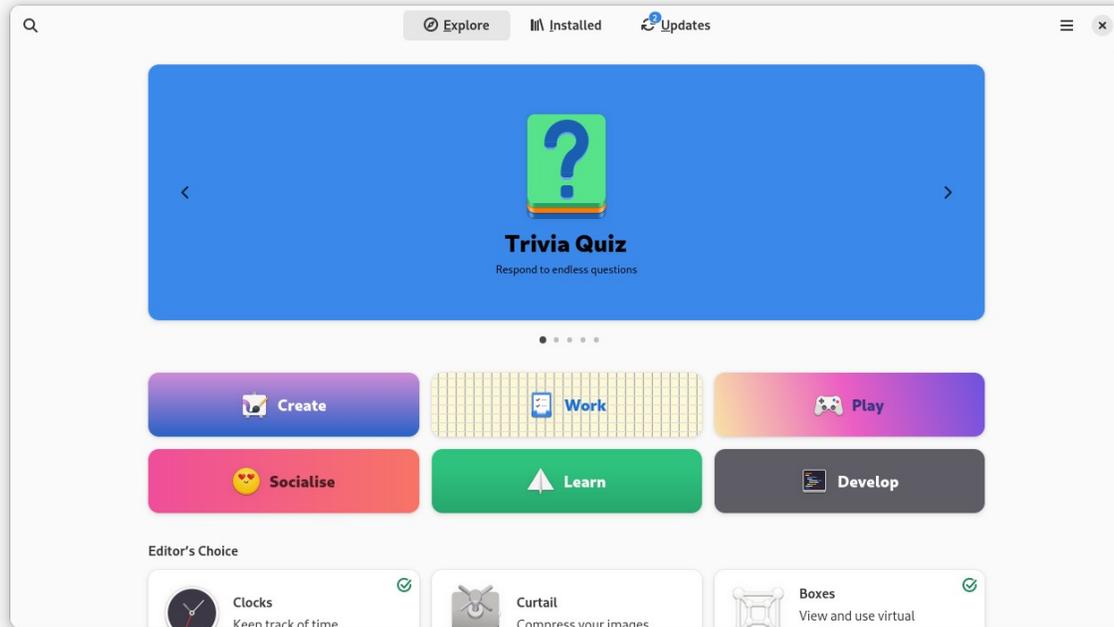
Vantaggi:

- Semplicissimo!
- Compatibile con tutte le distribuzioni.
- Le dipendenze sono già incluse.
- È portabile.
- Non richiede permessi di root.

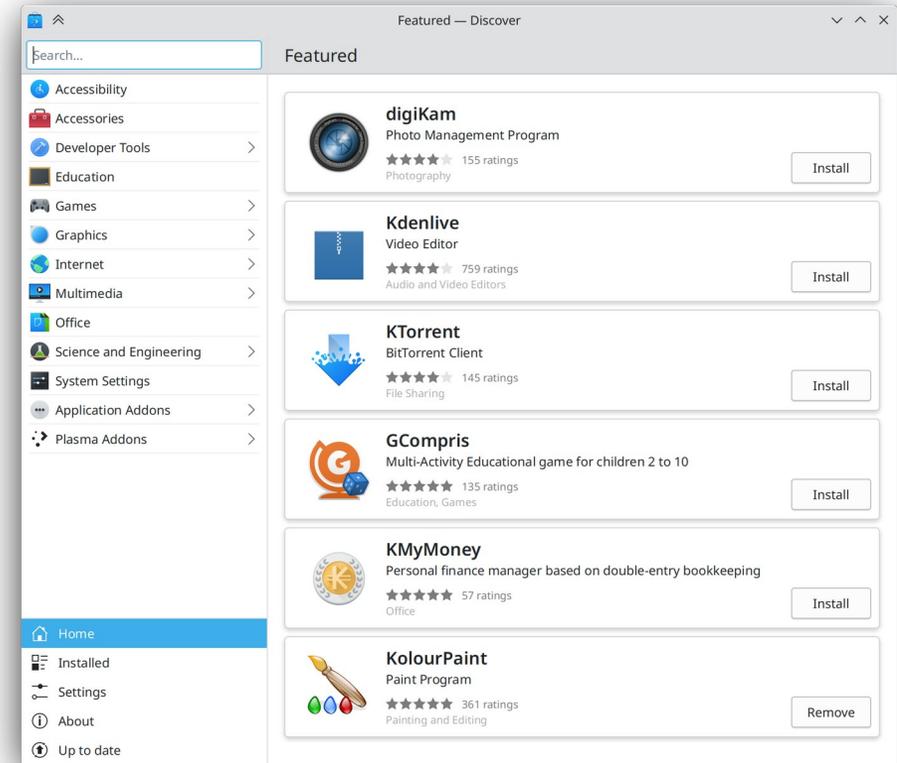
Svantaggi:

- Bisogna scaricare e aggiornare i programmi manualmente. (poco sicuro)
- I programmi occupano più spazio che con altri metodi.

Si può anche usare l'interfaccia grafica!



GNOME Software
(incluso in GNOME)



Discover
(incluso in KDE Plasma)