



LINUX:

La gestione del software su Debian

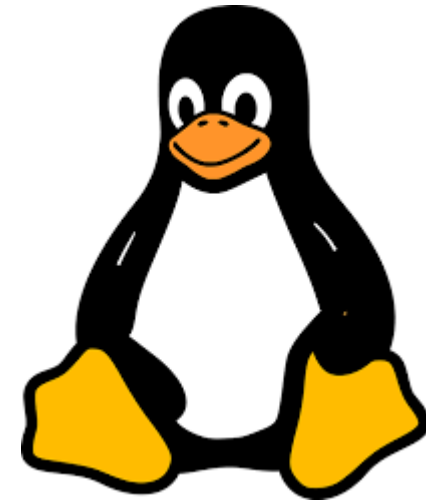
Relazione a cura di:
Nicolas Centofanti – Cristina Tinti

Corso realizzato da Zucchetti ©2025 - Tutti i diritti riservati.

La riproduzione, la registrazione, la comunicazione, la messa a disposizione al pubblico, il noleggio, il prestito, la diffusione senza l'autorizzazione di Zucchetti è vietata. Tutti i contenuti possono essere scaricati o utilizzati solo secondo le modalità previste dai diritti stessi e comunque non per uso commerciale. Ogni utilizzo dei contenuti in violazione delle norme di legge, è illecito e sarà pertanto perseguibile da Zucchetti.

Argomenti

- Formati archivio in Linux
- I pacchetti Linux e i loro gestori
- Il compilatore GCC



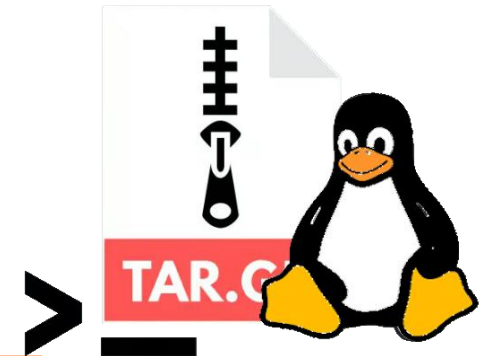
Formati di archivi di Linux

- In Linux, i formati di archivio sono utilizzati per consolidare più file in uno solo e spesso per comprimerli, riducendo lo spazio occupato.
- Sono fondamentali per backup, distribuzione software, e trasferimento dati.
- Di solito sono chiamati **tarball**.

Formati di archivi di Linux

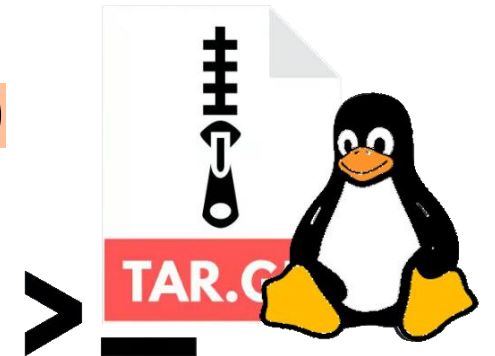
- I principali formati sono:

- **.tar** (Archivio) Unisce più file in uno solo, senza compressione
- **.tar.gz** (Archivio + compressione) Archivio .tar compresso con gzip
- **.tar.bz2** (Archivio + compressione) Archivio .tar compresso con bzip2
- **.tar.xz** (Archivio + compressione) Archivio .tar compresso con xz
- **.gz** (Compressione) File singolo compresso con gzip
- **.bz2** (Compressione) File singolo compresso con bzip2
- **.xz** (Compressione) File singolo compresso con xz
- **.zip** (Archivio + compressione) Archivio compresso compatibile con Windows
- **.7z** (Archivio + compressione) Alta compressione, supporta crittografia
- **.rar** (Archivio + compressione) Formato proprietario, supporto limitato su Linux
- **.zst** (Compressione) Basato su Zstandard, molto veloce e moderno



Formati di archivi di Linux: comando tar

- Il comando **tar**, che sta per «**tape archive**», è uno **strumento** potentissimo in Linux per archiviare, estrarre, e gestire file compressi.
- Le sue **funzioni principali** sono:
 - Archiviare** più file in un unico archivio .tar
 - Comprimere** l'archivio con gzip (.tar.gz), bzip2 (.tar.bz2), xz (.tar.xz)
 - Estrarre** file da archivi esistenti
 - Visualizzare** il contenuto di un archivio
 - Aggiornare** o **modificare** archivi esistenti



Formati di archivi di Linux: comando tar

- I comandi **tar** che ogni sistemista Linux deve sapere, sono:

- Creare un archivio .tar

```
tar -cvf archivio.tar cartella/
```

- Estrarre un archivio .tar

```
tar -xvf archivio.tar
```

- Creare un archivio compresso .tar.gz

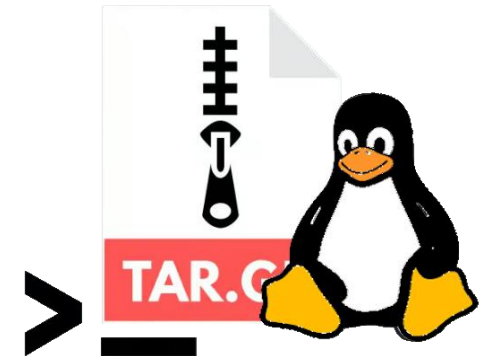
```
tar -czvf archivio.tar.gz cartella/
```

- Estrarre un archivio compresso .tar.gz

```
tar -xzvf archivio.tar.gz
```

- Visualizzare contenuto

```
tar -tvf archivio.tar
```



Formati di archivi di Linux: comando tar

- Aggiungere file a un archivio

```
tar -rvf archivio.tar nuovo_file
```

- Estrarre un file specifico

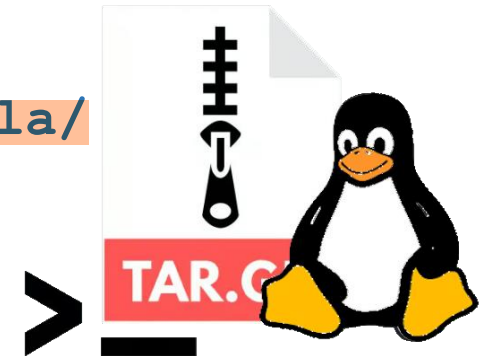
```
tar -xvf archivio.tar file.txt
```

- Escludere una cartella

```
tar -cvf archivio.tar --exclude='dir_esclusa' cartella/
```

- Estrarre in una directory diversa

```
tar -xvf archivio.tar -C /destinazione/
```



Formati dei pacchetti in Linux

- In Linux, i **formati di pacchetti di installazione** sono **file che contengono tutto il necessario per installare un'applicazione: binari, configurazioni, metadati e informazioni sulle dipendenze.**
- Esistono:
 - **formati tradizionali** legati alle distribuzioni;
 - **formati moderni** pensati per la portabilità e la sicurezza.

Formati dei pacchetti in Linux

- I formati tradizionali sono:
- **DEB**: ottimizzato per sistemi Debian-based (Debian, Ubuntu, Linux Mint), gestito con i comandi **apt** o **dpkg**.
- **RPM**: usato in ambienti Red Hat (Red Hat, Fedora, CentOS, openSUSE), gestito con i comandi **dnf**, **yum**, o **zypper**.

Formati dei pacchetti in Linux

- I **formati moderni** sono:
- **Snap**: sicuro e centralizzato, ma può essere più lento e pesante. Creato da Canonical (Ubuntu), sandbox, aggiornamenti automatici.
- **Flatpak**: più flessibile, ottimo per ambienti desktop. Sviluppato da Red Hat, sandbox, compatibile con Flathub.
- **ApplImage**: ideale per software portabile, basta scaricare e avviare. Portabile, non richiede installazione, nessun sandbox

Formati dei pacchetti in Linux



● I formati moderni sono:

● **Snap**: sicuro
Creato da Canonical

sante.
automatic.

● **Flatpak**: portabile
Hat, sandbox

pato da Red

● **AppImage**: portabile
Portabile, non

avviare.

sandbox:

«recinto virtuale» dove un'applicazione può funzionare senza accedere liberamente a file, rete, dispositivi o risorse critiche. Se il programma è malintenzionato o difettoso, i danni restano confinati all'interno. Permette di utilizzare il software in sicurezza.

Il gestore dei pacchetti APT

- Il gestore di pacchetti **APT (Advanced Packaging Tool)** è uno strumento fondamentale **per le distribuzioni Linux basate su Debian**, come Ubuntu, Linux Mint e molte altre.
- Serve per **installare, aggiornare, rimuovere e gestire software** in modo semplice e automatizzato.
- APT è un **resolver di dipendenze**: quando installi un programma, APT si occupa di scaricare anche tutte le librerie e i pacchetti necessari per farlo funzionare correttamente.
- Funziona in sinergia con **dpkg**, che è il gestore di pacchetti a basso livello.

Il gestore dei pacchetti APT: dove prende i pacchetti?

- Il gestore di pacchetti **APT** scarica i pacchetti dai **repository ufficiali** o da quelli **personalizzati** definiti nel file **/etc/apt/sources.list**.
- Esempio per Debian 12:

```
#deb cdrom:[Debian GNU/Linux 12.6.0 _Bookworm_ - Official amd64 NETINST ...
```

```
deb http://deb.debian.org/debian/ bookworm main non-free-firmware  
deb-src http://deb.debian.org/debian/ bookworm main non-free-firmware
```

- Può anche gestire PPAs (Personal Package Archives) per software non incluso nei repository standard è il gestore di pacchetti a basso livello.
- I pacchetti sono firmati digitalmente per garantirne l'integrità

Il gestore dei pacchetti APT: dove prende i pacchetti?

- Un **repository Debian** contiene diversi **rilasci** (ovvero versioni della distribuzione). E' una raccolta di software, aggiornamenti e pacchetti ufficiali della distribuzione.
- I nomi in codice hanno alias, le cosiddette **suites** (**stable, oldstable, testing, unstable**).
- Un **rilascio** è diviso in varie componenti.
- In Debian queste si chiamano **main, contrib** e **non-free** e il nome indica i termini delle licenze del software che contengono.
- Un rilascio ha anche pacchetti per varie architetture (amd64, i386, mips, powerpc, s390x, ...), oltre e pacchetti sorgente ed indipendenti dall'architettura



I rilasci di Debian prendono il nome da personaggi dei film della serie "Toy Story" (wheezy, jessie, stretch, ...).

La distribuzione Debian



Alcuni Debian codename e versioni:

- Duke (Debian 15) Annunciato solo il codename
- Forky (Debian 14) Testing. Data di rilascio non ancora comunicata
- Trixie (Debian 13) Rilasciata il 9 Agosto 2025
- Bookworm (Debian 12)
- Bullseye (Debian 11)
- Buster (Debian 10)
- Jessie (Debian 8)
- Wheezy (Debian 7)
- Lenny (Debian 5.0)
- Woody (Debian 3.0)
- Potato (Debian 2.2)
- Buzz (Debian 1.1)



Per ogni distribuzione viene definita una durata della vita (lifecycle):

- *Data Rilascio*
- *Data fine supporto EOS (End Of Support)*
- *Data fine supporto LTS (Long Time Support)*
- *Data fine supporto ELTS (Extended LTS)*

Esempio per Trixie:

- *Rilascio: 09/08/2025*
- *EOS: 09/08/2028*
- *LTS: 30/06/2030*
- *ELTS: 30/06/2033*

Il gestore dei pacchetti APT

- APT permette di installare software in maniera semplice e potente.
- E' necessaria connessione Internet per raggiungere i repository.
- Di seguito i comandi usati frequentemente:
- Aggiornare elenco pacchetti (da eseguire prima di ogni installazione)

`sudo apt update`

- Installare il pacchetto desiderato

`sudo apt install nome_pacchetto`



È frequente usare apt-get al posto di apt.

Il gestore dei pacchetti APT

Altri comandi utili:

Comando	Descrizione
<code>apt search nome</code>	Cerca un pacchetto
<code>apt show nome</code>	Mostra dettagli del pacchetto
<code>sudo apt remove nome</code>	Rimuove il pacchetto
<code>sudo apt purge nome</code>	Rimuove il pacchetto e le configurazioni
<code>sudo apt autoremove</code>	Elimina dipendenze inutilizzate
<code>sudo apt upgrade</code>	Aggiorna i pacchetti installati
<code>sudo apt dist-upgrade</code>	Aggiorna la minor della distribuzione

Il gestore dei pacchetti APT

- Si possono utilizzare **repository aggiuntivi** per:
 - accedere a software non incluso nei repository ufficiali;
 - ottenere versioni più aggiornate di programmi;
 - installare strumenti di terze parti.
- È una pratica comune, ma va fatta con attenzione per evitare conflitti o problemi di sicurezza: si consiglia di usare solo repository affidabili e verificati.
- Meglio non aggiungere troppi repository esterni: possono verificarsi conflitti di dipendenze.

Il gestore dei pacchetti APT

- Per aggiungere repository aggiuntivi è necessario:
 - Trovare il repository e la chiave GPG dal sito ufficiale del software
 - Aggiungere la chiave GPG
 - Aggiungere il repository
 - Aggiornare l'elenco dei pacchetti
 - Installare

Il gestore dei pacchetti APT

- 🕒 Esempio aggiunta **repository** per installare il Database Server PostgreSQL:

```
# sudo apt update
```

```
# sudo apt install gnupg2 wget
```

```
# sudo sh -c 'echo "deb
```

```
http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs) -  
pgdg main" > /etc/apt/sources.list.d/pgdg.list'
```

```
# curl -fsSL
```

```
https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo gpg  
--dearmor -o /etc/apt/trusted.gpg.d/postgresql.gpg
```

```
# sudo apt update
```

```
# sudo apt install postgresql-16 postgresql-contrib-16
```

Gestire manualmente le installazioni

- Quando si installano manualmente i software occorre una politica rigida di gestione.
- Ad esempio si consiglia di installare in /opt seguendo la logica:

`/opt/<software>versione/<cartelle>`

- E poi creare link simbolici.
- Ad esempio:

`/opt/jdk-17.0.15+6/...`

`/opt/apache-tomcat-9.0.106_2/..`

`/opt/jdk -> /opt/jdk-17.0.15+6/`

`/opt/tomcat -> /opt/apache-tomcat-9.0.106_2/`

Gestire manualmente le installazioni

- Quando si aggiorna la versione del software, si procede nel modo seguente:
 - Si scompatta e si installa il nuovo software.
 - Si fanno le opportune verifiche di corretta installazione.
 - Se i test sono positivi: si cambiano i link simbolici alle cartelle
 - Se i test sono negativi: si mantiene la configurazione corrente e si analizzano le problematiche.

Gestire manualmente le installazioni: compilazione

- Ricompilare i sorgenti in Linux è una pratica comune per chi vuole **personalizzare, ottimizzare o semplicemente installare software** non disponibile come pacchetto precompilato.
- È anche un ottimo modo per imparare come funziona davvero il sistema.
- Passaggi fondamentali per ricompilare un software da sorgente
 1. Preparare l'ambiente di compilazione
 2. Scaricare i sorgenti
 3. Configurare il progetto
 4. Compilare
 5. Installare

Gestire manualmente le installazioni: compilazione

- Per ricompilare i sorgenti è necessario avere a disposizione un compilatore.
- Linux offre una vasta gamma di **compilatori**, ognuno pensato per linguaggi diversi e con caratteristiche specifiche.
- I compilatori più diffusi:
 - **GCC (GNU Compiler Collection)**
 - **Java**
 - **Go (golang)**
 - **Clang**

Il compilatore GCC

- Il **compilatore GCC** (GNU Compiler Collection) è uno dei pilastri dello sviluppo software su Linux.
- Nato come compilatore per il linguaggio **C**, oggi supporta anche **C++**, **Objective-C**, **Fortran**, **Ada**, **Go**, e altri linguaggi.
- È gratuito, open source, e disponibile su praticamente tutte le distribuzioni Linux.
- È un **compilatore a riga di comando** che traduce il codice sorgente in linguaggio macchina, generando file eseguibili.
- È noto per la sua **stabilità**, **portabilità**, e **ampio supporto** da parte della community.



*Su Debian il pacchetto da installare per avere il compilatore gcc si chiama **build-essential***





Il software che crea successo



© Copyright by Zucchetti – 2025

Diritti di traduzione, di memorizzazione elettronica, di riproduzione e di adattamento, totale o parziale, con qualsiasi mezzo, sono riservati per tutti i paesi.
L'elaborazione dei testi, anche se curata con scrupolosa attenzione, non può comportare specifiche responsabilità per eventuali involontari errori o inesattezze.