

Einführung in die Linux-Paketverwaltung

ITS-Net-Lin

Sebastian Meisel

7. Dezember 2024

1 Einführung

Linux-Distributionen verwenden zur Installation und Verwaltung von Software-Paketen unterschiedliche Paketverwaltungen. Diese unterscheiden sich in der Art und Weise, wie Pakete installiert, aktualisiert und entfernt werden. Jede Distributionsfamilie hat ihre eigenen bevorzugten Werkzeuge und Formate.

1.1 Paketverwaltungen im Überblick

Es gibt eine Vielzahl von Paketverwaltungs-Tools, die je nach Distribution eingesetzt werden:

APT (Advanced Packaging Tool) Wird hauptsächlich in Debian-basierten Distributionen wie Ubuntu eingesetzt. Mit apt können Pakete schnell und einfach installiert und verwaltet werden.

DNF (Dandified YUM) Die moderne Paketverwaltung für Red Hat-basierte Distributionen wie Fedora und CentOS.

YUM (Yellowdog Updater, Modified) Älteres Paketverwaltungstool von Red Hat, das durch DNF ersetzt wurde, aber in älteren Versionen von CentOS und RHEL immer noch verwendet wird.

Zypper Wird in openSUSE und anderen SUSE-basierten Distributionen genutzt. Es bietet eine ähnliche Funktionalität wie apt und dnf, aber mit anderen Befehlen und Optionen.

Pacman Wird in Arch Linux und Arch-basierten Distributionen wie Manjaro verwendet. Pacman ist dafür bekannt, besonders einfach und effizient zu sein.

In den letzten Jahren haben auch die Paketverwaltungen für macOS und Windows an Bedeutung gewonnen:

Homebrew (brew) Eine beliebte Paketverwaltung für macOS und Linux, die speziell für die Installation von Open-Source-Software entwickelt wurde.

Cocoa Eine Sammlung von Frameworks für die Entwicklung von Software auf macOS. Wird nicht direkt zur Paketverwaltung verwendet, spielt jedoch eine wichtige Rolle in der macOS-Umgebung.

Winget Ein Paketmanager für Windows 10 und 11, der von Microsoft entwickelt wurde. Winget ermöglicht das Installieren und Verwalten von Software aus einem zentralen Repository.

1.2 APT unter Debian

In Debian (und auch in Ubuntu) wird die Paketverwaltung mit dem Tool apt durchgeführt. APT erleichtert die Installation, Aktualisierung und Entfernung von Softwarepaketen, die im .deb-Format vorliegen.

1.2.1 Wichtige Befehle

Hier sind einige der grundlegenden Befehle zur Verwendung von APT:

System aktualisieren Um alle installierten Pakete (Programme und Systemkomponenten) auf den neuesten Stand zu bringen, verwendet man:

```
1 sudo apt update    # Paketquellen aktualisieren
2 sudo apt upgrade   # Pakete aktualisieren
```

Zumindest der erste Befehl sollte einmal am Tag ausgeführt werden, bevor neue Software installiert wird. Generell ist es wichtig, das System regelmäßig zu aktualisieren.

Paket installieren Um ein Paket zu installieren, verwendet man den Befehl `apt install`:

```
1 sudo apt install <paketname>
```

Paket entfernen Ein Paket kann mit dem Befehl `apt purge` entfernt werden:

```
1 sudo apt purge <paketname>
```

Wenn man die Konfiguration erhalten möchte, kann man alternativ `apt remove` verwenden.

```
1 sudo apt remove <paketname>
```

Paket suchen Um nach einem Paket zu suchen, das bestimmte Kriterien erfüllt:

```
1 apt search <suchbegriff>
```

Paketdetails anzeigen Informationen zu einem installierten Paket bekommt man mit:

```
1 apt show <paketname>
```

1.2.2 Beispiel

Wenn man dem Editor `vim` installieren möchte, würde der Befehl folgendermaßen aussehen:

```
1 sudo apt install vim
```

APT kümmert sich dann darum, alle Abhängigkeiten zu installieren und das Paket auf dem System verfügbar zu machen.

Tip: `vim` ist ein sehr mächtiger Editor, den man auf der Kommandozeile verwenden kann. Allerdings hat er ein ungewöhnliches Bedienkonzept. Es lohnt sich aber, zumindest die Grundlagen zu lernen. Dafür steht das Programm `vimtutor` zur Verfügung, das Sie in wenigen Minuten mit der Bedienung vertraut macht.

1.3 DNF, YUM, Zypper und Pacman

Die Paketverwaltungen von Red Hat, openSUSE und Arch Linux haben ähnliche Funktionen wie `apt`, jedoch mit unterschiedlichen Befehlen und Mechanismen:

DNF (Fedora, CentOS, RHEL)

- System aktualisieren: `sudo dnf update`
- Installieren: `sudo dnf install <paketname>`

- Entfernen: `sudo dnf remove <paketname>`

YUM (ältere Red Hat-Distributionen) • System aktualisieren: `sudo yum update`

- Installieren: `sudo yum install <paketname>`
- Entfernen: `sudo yum remove <paketname>`

Zypper (openSUSE) • System aktualisieren: `sudo zypper update`

- Installieren: `sudo zypper install <paketname>`
- Entfernen: `sudo zypper remove <paketname>`

Pacman (Arch Linux und Manjaro) • System aktualisieren: `sudo pacman -Syu`

- Installieren: `sudo pacman -S <paketname>`
- Entfernen: `sudo pacman -R <paketname>`

1.4 Homebrew (brew) und Winget

Auch für macOS und Windows stehen Paketverwaltungen zur Verfügung:

Homebrew (brew) Ein äußerst populärer Paketmanager für macOS und Linux, der speziell für Open-Source-Software entwickelt wurde. Installationsbefehl:

```
1 brew install <paketname>
```

Winget Der Windows-Paketmanager von Microsoft. Um ein Paket zu installieren, verwendet man:

```
1 winget install <paketname>
```

1.5 Fazit

Obwohl jede Distribution ihre eigene Paketverwaltung hat, verfolgen sie alle das gleiche Ziel: Die einfache Verwaltung von Softwarepaketen. Mit Tools wie APT, DNF, YUM, Zypper und Pacman wird die Installation und Wartung von Software unter Linux erheblich vereinfacht. Auf macOS und Windows bieten Homebrew und Winget ähnliche Funktionen, um auch auf diesen Plattformen Software bequem zu verwalten.