

Linux Cafe:

**Für Einsteiger: System aktuell halten / eMail
einrichten / etc.**

Rainer Peipp, 03.05.2023

Linux-Distributionen: Wo kommt die Software her?

- **Verschiedene Ansätze**

- „Normale“ Pakete (deb, rpm, ...), die zur Distribution gehören
- zusätzliche Paketquellen (z. B. Opera, Anydesk, aktuelles LibreOffice)
- manuell installierte Pakete (z. B. Master PDF Editor)
- App-Container (Flatpak, Snap)
- App-Images
- In Containern gekapselte Anwendungen (Docker, ...)
- Spezielle Anwendungen mit eigener „Modulverwaltung“ (Python, TeX, ...)
- Individuelles Installationsprogramm (z. B. VMWare Player)
- Installation aus dem Quellcode (./configure; make; make install)

- **gemischte Nutzung ist möglich (und üblich)**

Linux-Distributionen: Support-Zeiträume

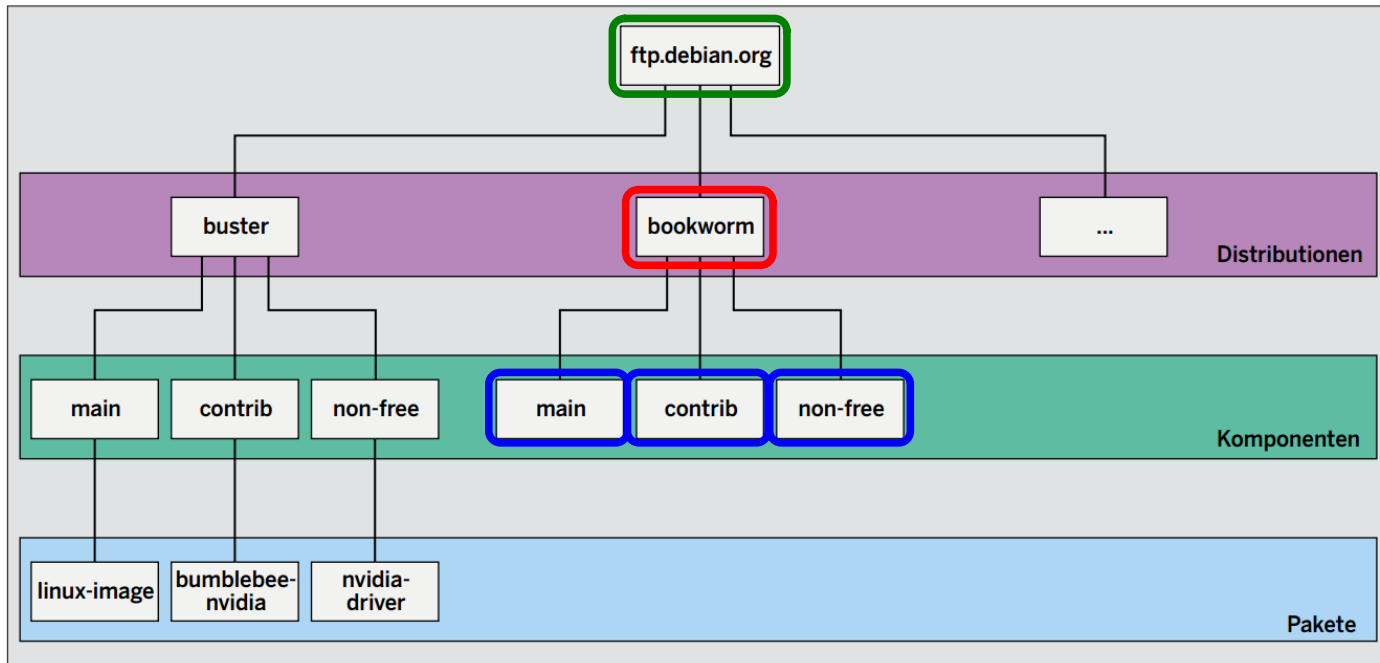
- **Unterschiedliche Support-Zeiträume**

- z. B. Ubuntu
 - STS (Short Term Support): Pakete (und Kernel) alle 6 Monate neu
 - LTS (Long Term Support): Pakete für z. B. 5 Jahre stabil; Sicherheitsupdates laufend; Kernel wird zwischendurch erneuert
- anderer Ansatz: Rolling Release
 - Manjaro
 - Open Suse Tumbleweed
- Linux Mint
 - Kernel bleibt innerhalb einer Version der Distribution stabil
 - ggf. kann ein aktualisierter Kernel auf Wunsch installiert werden
- Kernel-Updates für Ubuntu und Linux Mint: <https://kernel.ubuntu.com/~kernel-ppa/>
- Anleitung: <https://m6u.de/BMKE>
- Alte Kernel entfernen: "sudo apt autoremove"

Paketmanagement mit deb-Paketen

- **Programmpakete können komfortabel verwaltet werden**
 - Installieren, Deinstallieren, Aktualisieren, Konfigurieren
 - Abhängigkeiten werden mit betrachtet
 - Pakete werden in „Repositories“ verwaltet und dort nach
 - Distribution, Komponenten und Paketen
 - Architektur
 - Programmen und Quellcodegegliedert
 - Mehrere Repositories möglich

Debian Repository



`/etc/apt/sources.d:`

```
deb http://ftp.de.debian.org/debian bookworm main contrib non-free
```

App-Container

- Flatpak (Gnome Foundation)
- Snap (Canonical, Ubuntu)
- App-Container setzen auf der jeweiligen Basis-Infrastruktur auf
- App-Container bringen eigene Bibliotheken mit und sind damit relativ unabhängig von der jeweiligen Distribution
- Beide App-Container sind in gängigen Distributionen wie Debian, Fedora, Arch Linux etc. installierbar
- Beide App-Container können parallel installiert werden
- Nachteile: deutlich erhöhter Platzbedarf; langsames Starten der Programme

AppImage

- <https://appimage.github.io/apps/>
- **Anwendungen ohne Installation laufen lassen**
- **App-Container bringen eigene Bibliotheken mit und sind damit relativ unabhängig von der jeweiligen Distribution**
- **Nachteile:**
 - wesentlich erhöhter Platzbedarf
 - langsames Starten der Programme
 - keine automatische Aktualisierung

snap

- Wird durch Canonical (Ubuntu) vorangetrieben
- <https://snapcraft.io> (zentraler Snapstore)
- Installation der Basis-Software auf nicht-Ubuntu-Systemen: `sudo apt install snap`
- Linux Mint blockiert snap standardmäßig:
`sudo mv /etc/apt/preferences.d/nosnap.pref
~/nosnap.backup`
- `snap search [Suchbegriff]` zur Suche nach App-Containern
- `snap list [--all]` zur Anzeige der installierten App-Container

Ubuntu: snap-Pakete

- Werden nicht über `apt` aktualisiert
- snap-Pakete werden standardmäßig 4 Mal pro Tag aktualisiert (`snapd`)
- `snap refresh --time` zeigt an, wann die nächste Aktualisierung erfolgen wird
- `sudo snap refresh` startet manuelle Aktualisierung
- `sudo snap refresh --hold=24h` stoppt die Aktualisierung für 24 Stunden (z. B. bei schlechter Netzverbindung)

Ubuntu: snap-Pakete

- **Firefox snap durch klassisches deb-Paket ersetzen**
 - `sudo snap remove firefox`
 - `sudo add-apt-repository ppa:mozillateam/ppa`
 - `sudo apt install -t 'o=LP-PPA-mozillateam' firefox`

 - `sudo sh -c 'cat > /etc/apt/preferences.d/mozilla-ppa' << EOF`
 - `Package: firefox*`
 - `Pin: release o=LP-PPA-mozillateam`
 - `Pin-Priority: 501`
 - `EOF`
- **Alternativ: ESR-Release installieren:**
 - `sudo apt install firefox-esr firefox-esr-locale-de`

Flatpak

- Zu Hause bei der GNOME Foundation (durch Red Hat entwickelt)
- kein zentraler Store
- Suche auf <https://flathub.org>
- `flatpak remotes` zeigt die im System eingebundenen Repositories
- `flatpak search [Suchbegriff]` sucht nach App-Containern
- `flatpak install [Application ID]` installiert App-Container
- `flatpak list` zeigt die installierten App-Container
- `flatpak uninstall [Name]` entfernt App-Container
- Flatpaks laufen (über cgroup und namespaces) in einer Sandbox
- Installation der Basis-Software auf Ubuntu-Systemen: `sudo apt install flatpak`
- Flatseal zum Anpassen der Rechte eines App-Containers (`flatpak install Flatseal`)

Beispiel: Installation eines Pakets über die Softwareverwaltung (Anwendungsverwaltung)

- Dateimanager Midnight Commander
- Es werden automatisch notwendige Pakete mit installiert
- Bei der Deinstallation werden die nur für Midnight Commander benötigten Pakete wieder mit deinstalliert
- Installation über Kommandozeile:
`apt install mc`
- Weitere Hinweise auf ggf. sinnvolle Pakete (zur Erweiterung des Funktionsumfangs des Midnight Commanders)
- Automatische Aktualisierung

Beispiel: Neue Paketquelle ergänzen (anydesk)

- <https://www.anydesk.de>
- <http://deb.anydesk.com/howto.html>
- `(als root (sudo -i))`
- `wget -q0 - https://keys.anydesk.com/repos/DEB-GPG-KEY | apt-key add -`
- `echo "deb http://deb.anydesk.com/ all main" > /etc/apt/sources.list.d/anydesk-stable.list`
- `apt update`
- `apt install anydesk`
- **automatische Aktualisierung**

Beispiel: Manuelle Installation ein deb-Pakets

- Master PDF Editor for Linux
- Download von <https://code-industry.net/free-pdf-editor/>
- a) Installation über grafische Oberfläche
- b) Installation über Kommandozeile:

```
sudo dpkg -i master-pdf-editor-5.9.40-qt5.x86_64.deb
```


(alternativ:

```
sudo apt install ./master-pdf-editor-5.9.40-qt5.x86_64.deb
```

)
- **Achtung:** keine automatische Aktualisierung (aber das Programm meldet selbst, wenn Updates bereit stehen)

Beispiel: Anzeigen der Abhängigkeiten von Paketen

- **Hilfsmittel: debtree**
- **Liefert grafische Darstellung (z. B. in einer png-Datei) der Abhängigkeiten eines Pakets**
- `debtree Paket | dot -Tpng > paket.png`
- **Beispiel:**
`debtree mc | dot -Tpng > mc.png`

Beispiel: Aktualisierung des gesamten Systems

- `sudo apt update` (aktualisiert die lokale Liste der Paketdatenbank)
- `sudo apt upgrade` (aktualisiert alle bereits installierten Pakete)
- (evtl. unvollständige Abhängigkeiten beheben
`sudo apt -f install`)

apt vs. apt-get

- **apt-get**: älteres und mit mehr Funktionen ausgestattetes Programm zum Paketmanagement
apt-get upgrade installiert nur aktualisierte Pakete, fügt aber keine neuen hinzu (ggf. notwendig für Linux-Kernel)
- **apt**: vereinfachtes Programm
apt upgrade installiert auch neue Pakete

Spezielle Meldungen von apt

- Die folgenden Pakete wurden automatisch installiert und werden nicht mehr benötigt
Pakete löschen mit:
`sudo apt autoremove`
- Die folgenden Pakete sind zurückgehalten worden
(Die Pakete werden erst nach und nach für alle freigegeben)
Anzeigen der betroffenen Pakete
`apt list -upgradable`
Anzeigen des Status eines Pakets
`apt show [Paketname]=[neue Versionsnummer]`

Upgrade auf eine neue Version einer Distribution

- Wichtigstes Prinzip: RTFM (Read The Fucking Manual)!
- Leicht unterschiedliche Befehle bei verschiedenen Distributionen, z. B.:
 - Linux Mint 21

```
sudo apt update
sudo apt install mintupgrade
sudo mintupgrade

...

sudo apt remove mintupgrade
sudo reboot
```
 - Ubuntu

```
sudo apt update
sudo apt upgrade
sudo apt full-upgrade
sudo do-release-upgrade
```

Upgrade auf eine neue Version einer Distribution (2)

- **I. d. R. sollten vor dem Upgrade alle zusätzlichen Repositories (PPA) deaktiviert werden.**
- **Ggf. sollten auch die darüber installierten Pakete vor dem Upgrade entfernt werden, insbesondere, wenn sie mit Paketen aus der Distribution in Konflikt stehen könnten (LibreOffice, ...).**
- **Ggf. nach dem erfolgreichen Upgrade die zusätzlichen Repositories wieder aktivieren.**

Grafische Oberflächen zum Paketmanagement

- **Ubuntu: Ubuntu Software**
- **Linux Mint: Anwendungsverwaltung**
- **Beide liefern nur eine Auswahl (meist grafischer Desktop-)Anwendungen**
- **alternativ: Synaptic liefert alle verfügbaren Pakete**

apt: Häufige Fehlermeldungen

- `E: Could not get lock /var/lib/dpkg/lock-frontend - open (11: Resource temporarily unavailable)` → **Es läuft schon ein andere apt-Prozess (automatische Updates?) → Warten und später noch einmal versuchen.**

dpkg-reconfigure: Paket neu konfigurieren

- `dpkg-reconfigure [Paketname]` fragt die Konfigurationsinformationen, die sonst nur bei der ersten Installation des Pakets abgefragt werden, erneut abhen.

Email: Grundlagen

- **Nutzung eines Mailproviders
(eigener Mailserver ist i. d. R. nicht sinnvoll)**
- **Zugang über:**
 - Web-Oberfläche
 - Email-Programm (Protokolle: ~~POP3~~, IMAP, SMTP, Exchange (Microsoft))
 - Smartphone-App

Email: Beispiel

- **Einrichtung eines Email-Programms**
- **Notwendig**
 - Zugangsdaten
 - Benutzername
 - Passwort
 - Serverdaten
 - IMAP-Server (Posteingang): Adresse, Port, Transportverschlüsselung
 - SMTP-Server (Postausgang): Adresse, Port, Transportverschlüsselung

Email: Beispiel Thunderbird und mail@web.de

- **Vollautomatische Einrichtung mit Benutzername und Passwort**
 - Benutzername: linuxcafenbg@web.de
 - Passwort: linux.cafe.230503
- **Serverdaten werden automatisch erkannt**
- **Verbindung zu den Servern (Postein- und Postausgang) erfolgen verschlüsselt**

Email: Beispiel Thunderbird und mail@peipp.eu

- **Vollautomatische Einrichtung mit Benutzername und Passwort scheitert!**
- **Postausgangsserver falsch**
- **Benutzername falsch**
- **Manuelle Eingabe nach Daten des Mailproviders**
- **Überprüfung: Werden verschlüsselte Verbindungen genutzt?**

Email: Beispiel Thunderbird und mail@peipp.eu

Ihre E-Mail-Konten

Dem Konto "wp1037730-lc" sind folgende Mailadressen zugeordnet:
lc@peipp.eu

Posteingang

Der Posteingangsserver lautet "wp1037730.mail.server-he.de".

Bitte nutzen Sie folgende Portangaben:

IMAP	143 (unverschlüsselt)
IMAP (SSL/TLS)	993 (verschlüsselt)
IMAP (STARTTLS)	143 (verschlüsselt)
POP	110 (unverschlüsselt)
POPs	995 (verschlüsselt)
Authentifizierung	Plain, SSL/TLS, STARTTLS

Postausgang

Der Postausgangsserver lautet "wp1037730.mailout.server-he.de".

Bitte nutzen Sie folgende Portangaben:

SMTP	25 (unverschlüsselt)
SMTP	25 (verschlüsselt TLS)
Authentifizierung	PLAIN, LOGIN, TLS, STARTTLS

Alternativ Port 465/587, falls Port 25 nicht funktionieren sollte.

Email: Beispiel Thunderbird und mail@peipp.eu

- **Vollautomatische Einrichtung mit Benutzername und Passwort scheitert!**
- **Postausgangsserver falsch**
- **Benutzername falsch**
- **Manuelle Eingabe nach Daten des Mailproviders**
- **Überprüfung: Werden verschlüsselte Verbindungen genutzt?**

Email: Beispiel Thunderbird und mail@peipp.eu

- **Vollautomatische Einrichtung mit Benutzername und Passwort scheitert!**
- **Postausgangsserver falsch**
- **Benutzername falsch**
- **Manuelle Eingabe nach Daten des Mailproviders**
- **Überprüfung: Werden verschlüsselte Verbindungen genutzt?**

Email: Sicherheit

- Die für Email genutzten Protokolle IMAP und SMTP erlauben nur eine Authentifizierung über Benutzername/Password → keine Zwei-Faktor-Authentifizierung
- Web-Portale sind ggf. mit Zwei-Faktor-Authentifizierung ausgestattet
- In jedem Fall müssen die Verbindungen über IMAP und SMTP verschlüsselt sein! SSL, STARTTLS
- Zusätzlicher Schutz über Ende-zu-Ende-Verschlüsselung
 - PGP
 - S/MIME
- **Alternative: Ende-zu-Ende verschlüsselte Messenger mit Desktop-Anwendung**
 - Signal Desktop