

Zusammenfassung Tag 5

Paketverwaltung

- Durch die Paketverwaltung ist ein einfaches Installieren und Updaten von Programmen möglich.
- **Auflösen von Abhängigkeiten**
 - Firefox benötigt unter anderem „freetype“ und „GTK2“ um zu funktionieren
 - Durch die Abhängigkeiten werden diese beiden Pakete in der richtigen Version mit installiert.
 - Abhängigkeiten können über mehrere ebenen gehen.
- **Unterschiedliche Paketverwaltung je nach Distribution**
 - Ubuntu / Debian nutzt apt / dpkg (Programmpaket: .deb)
 - Red Hat / CentOS nutzt yum (Programmpaket: .rpm)
 - Umwandeln von Programmpaket möglich führt aber zu Problemen

Paketverwaltung unter Ubuntu mit GUI(graphical user interface)

- **Synaptic Package Manager**
 - wird genutzt um Pakete mittels einer Grafischen Oberfläche zu installieren.

Pakete über das Terminal installieren (Ubuntu)

- Wird genutzt wenn zum Beispiel keine Grafische Oberfläche zur Verfügung steht
- **Befehl für die Paketverwaltung:** *apt* (je nach system auch: *apt-get* oder *aptitude*)
- *sudo* wird beim installieren/updaten von Paketen benötigt
- **Ubuntu Paketlisten überprüfen/updaten** *sudo apt update*
- **Ubuntu Paketliste installieren** *sudo apt upgrade*
 - *sudo apt dist-upgrade* fügt zusätzlich neue Abhängigkeiten hinzu falls benötigt (Updatet dann auch Pakete die eine neue Abhängigkeit benötigen)
- **Ubuntu Paket installieren:** *sudo apt install paketname*
 - *sudo apt install htop*

Nach Paketen Suchen, Paketquellen (Ubuntu)

- ***packages.ubuntu.com* / *packages.debian.com***
 - Website mit grafischer Oberfläche zum suchen von Paketen
- ***sudo apt search paketname***
 - Terminal Befehl zum suchen von Paketen
- **Pfad zur Datei mit Paketquellen:** */etc/apt/sources.list*
- **Pfad zum Ordner für eigene Paketquellen:** */etc/apt/sources.list.d*

zusätzliche Paketquellen (Ubuntu)

- **Heruntergeladenes Paket installieren mit:**
 - `sudo dpkg -i /pfad/zumPaket/`
- Nach der Installation taucht das Paket dann auch unter `/etc/apt/sources.list.d` auf
- Man muss dem Autor einer Paketquelle vertrauen, da schädlicher Code darüber in das System gelangen kann.

Ubuntu und PPAs

- **PPA (Personal Package Archives)**
 - eigene Pakete zur Verfügung stellen
 - von anderen Usern Pakete runterladen
(man muss wieder den Autoren des Paketes vertrauen)

Paket manuell kompilieren (Ubuntu)

- **`sudo apt install make gcc g++ automake`**
 - wird benötigt um ein Paket zu kompilieren
- **Der Quellcode wird benötigt**
 - auf der Website des Programms ist meistens auch eine Anleitung vorhanden wie man dieses richtig kompiliert
- Nach dem Kompilieren kann das Programm aus dem Programm Ordner aus gestartet werden
- **`su root`**
`make install`
 - fügt das Programm auch zur Paketverwaltung hinzu.
Nicht zu empfehlen da es zu Komplikationen kommen kann.
- Fall das Programm nach einen Systemupdate nicht mehr laufen sollte, einfach das Programm nochmal neu kompilieren

Paketverwaltung unter CentOS

- **Paketmanager: `yum`**

◦ <code>yum install paketname</code>	Installation eines Paketes
◦ <code>yum remove paketname</code>	Deinstallation eines Paketes
◦ <code>yum downgrade paketname</code>	Installiertes Paket auf ältere Version bringen
◦ <code>yum search paketname</code>	Nach einem Paket suchen.

- **Es können auch Pakete über URLs installiert werden**
 - `yum install https://LinkzumPaket.com/test.rpm`
 - wichtig ist das `.rpm` am ende
- **Hinzufügen eines neuen repository (epel-release)**
 - `yum install repositoryname`
 - epel (extra packages for enterprise linux)
 - beim ersten herunterladen eines Paketes muss man bestätigen, das man dem repository vertraut. (Fingerabdruck überprüfen)

Nützliche Befehle:

<code>clear</code>	Bereinigt die Konsole
<code>strg+c</code>	Beendet ein Programm / unterbricht einen Befehl
<code>cat</code>	Erzeugt eine Ausgabe z.B. von einer Datei