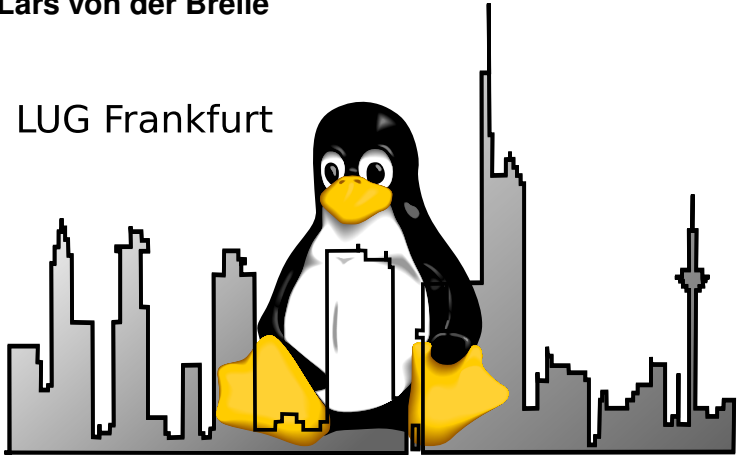


Slackware

Das Paketmanagement

Lars von der Brellie

LUG Frankfurt





Übersicht I

Titel

Allgemeine Vorbemerkungen

Entwicklungszweige

Paketquellen

Administration

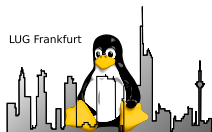
Paketbau

Lösungen zu verschiedenen Aufgaben

Fragen und Antworten

Links

IRC



Allgemeine Vorbemerkungen

Über diese Präsentation

Diese Präsentation ist bewußt und gewollt so angelegt, das man sie im Nachgang als Tutorium durcharbeiten kann, um die behandelten Inhalte später erneut nachzuvollziehen.

Nichts desto trotz beinhaltet sie bestimmt einige Fehler. Der Author bittet hier um Nachsicht.

Allgemeine Vorbemerkungen

Alles ist vanilla!



In Slackware wird alle Software unverändert kompiliert und installiert. Das gilt auch und im besonderen, für den Kernel.

Ausnahmen sind patches, ohne die die Software in der gegebenen Infrastruktur nicht lauffähig, instabil oder unsicher sind. Patchen bleibt aber die Ausnahme.



Allgemeine Vorbemerkungen

Automatische Auflösung der Abhängigkeiten

Gibt es eine automatische Auflösung der abhängigkeiten in Slackware?

nein

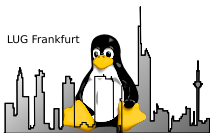


Allgemeine Vorbemerkungen

Automatische Auflösung der Abhängigkeiten

... und das ist gut so!

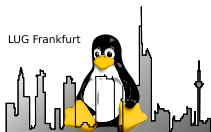
Die meisten Programme benötigen „ihre zwei Bibliotheken“, die sie einbinden. Ein System zur Auflösung der Abhängigkeiten läuft nach der Erfahrung des Authors irgendwann aus dem Ruder und installiert übermäßig viele Pakete, die eigentlich gar nicht gebraucht werden.



Allgemeine Vorbemerkungen

Automatische Auflösung der Abhängigkeiten

Die Abhängigkeiten der Programme, die man benutzt, eruiert man im Idealfall nur ein mal.



Entwicklungszweige

stable

Slackware unterhält zwei verschiedene Zweige in den Repositories.

Der **stable** Zweig, in dem die entsprechende Version veröffentlicht ist. Hier werden Sicherheitsupdates durch Paketupdates eingepflegt, aber keine neuen Programme eingepflegt oder andere Änderungen der Infrastruktur vorgenommen.



Entwicklungszweige

current

Current ist der Entwicklungszweig. Hier wird das System weiterentwickelt. Ein auf current gepflegtes System sollte nach einem Update immer stabil laufen. Das funktioniert auch normalerweise.



Paketquellen

Offizieller Datenträger

Die offiziellen releases bekommt man am besten von <http://www.slackware.com/getslack/>. Von dort navigiert man weiter und lädt sich ein Image herunter oder man lädt sich den Verzeichnisbaum mit den Installationsdateien¹.

¹Hier kann man sich im Anschluss einfach einen Installer auf einem USB-Stick bauen



Paketquellen

[Slackbuilds.org](https://slackbuilds.org)

Das ist das Repository, dass von der Community gefüllt und gepflegt wird.



Paketquellen

AlienBob (Eric Hameleers)

AlienBob ist eine feste Größe in der Slackware Community und hat unglaublich viel Arbeit in die Distribution gesteckt. Er hält unter <http://www.slackware.com/~alien/slackbuilds/> verschiedene Pakete vor.



Paketquellen

Eigenbau Pakete

Wie man eigene Pakete kreiert ist Gegenstand dieses Vortrages.



Paketquellen

Sonstige

Es gibt noch weitere Seiten, die slackware Pakete anbieten. Diese wechseln allerdings stetig und unaufhörlich, so dass es nicht lohnt sie hier aufzuzeigen.



Administration

Die Installation, Pflege und Entfernung von Paketen auf dem System erfolgt mittels Kommandozeilentools. Alle diese Tools haben man-pages, die bereitwillig über alle Vorgänge der diesbezüglichen Aktionsmöglichkeiten Auskunft geben.

Der Author empfiehlt, vor der Anwendung die Vorschläge der manpages wohlwollend zu diskutieren.



Administration

Installation

Die Installation eines Paketes erfolgt mittels dem Befehl:

```
root@rechner ~# installpkg /pfad/zur/paketdatei.tgz
```




Administration

Upgrade

Das upgraden eines Paketes erfolgt mittels dem Befehl:

```
root@rechner ~# upgradepkg /pfad/zur/paketdatei.txz
```



Administration

Deinstallation

Das Entfernen eines Paketes erfolgt mittels dem Befehl:

```
root@rechner ~# removepkg paketname
```



Administration

Entpacken

Das Entpacken eines Paketes erfolgt mittels dem Befehl:

```
root@rechner ~# explodepkg /pfad/zur/paketdatei.txz
```



Administration

Systemweite Aktionen

Mittels `slackpkg` sind Funktionen, wie Systemupgrades und ähnliches gelöst. Mit diesem Programm sind auch weitreichendere Aktionen, als einfaches Installieren oder Deinstallieren möglich.

```
root@rechner ~# slackpkg
```



Administration

pkgtool

pkgtool ist ein Menübasiertes Programm zur Paketverwaltung

```
root@rechner ~# pkgtool
```



Administration

pkgdata

pkgdata ist ein Programm zur Zusammenfassung von Daten

```
root@rechner ~# pkgdata
```



Administration

pkgdiff

pkgdiff ist ein Programm zum Vergleichen zweier Pakete

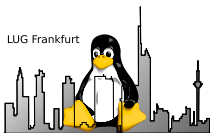
```
root@rechner ~# pkgdiff <Paket 1> <Paket 2> | less
```



Paketbau

Inhalt

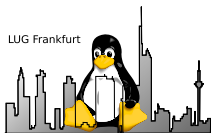
Der Inhalt eines exemplarischen Paketes wird im folgenden aufgezeigt.



Paketbau

ed

Gewählt ist das Paket `ed`. `ed` ist ein Editor, der schon lange sein Dasein in der unixoiden Welt fristet.



ed
source Paket

Es gibt für die offiziellen Quellen immer ein Quellpaket.

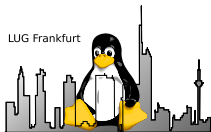


ed

Source Dateien

ed

- ed-1.17.tar.lz
- ed-1.17.tar.lz.sig
- ed.slackbuild
- slack-desc



ed

Source Dateien

[ed-1.17.tar.lz](#) : Gepackter source code

[ed-1.17.tar.lz.sig](#) : Prüfsumme

[ed.slackbuild](#) : bash script zum erstellen des Paketes

[slack-desc](#) : Eine Datei mit der Beschreibung des Paketes

[doinst.sh](#) : Ein Skript, das **nach** der Installation ausgeführt wird (nicht in diesem Paket)

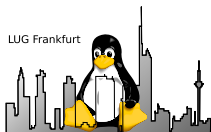


Source Dateien

ed.slackbuild

Die Datei [\[hier-könnte-ihr-paketname-stehen\].slackbuild](#) oder einfach nur das slackbuild ist ein bash skript, dass aus dem sourcen das Paket automatisch aufbaut.

Es entpackt den Code, kompiliert und linkt die Quellen und die Dokumentation, baut einen temporären Dateibaum auf und packt das alles dann in einem Paket zusammen.



Source Dateien

slack-desc

Eine Textdatei, die eine Beschreibung des Programmes enthält und ein paar Formalien unterliegt.

Mehr dazu später.



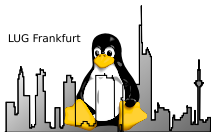
Source Dateien

`doinst.sh`

Ein bash skript, das nach der Installation ausgeführt wird um Aktionen auszuführen, die erst nach der Installation durchgeführt werden sollten.

Z. B. Softlinks in `/usr/local/bin` anlegen und ähnliches.

Im Paket `ed` ist diese Datei nicht in den Quellen vorhanden. Aber im Paket dann schon. Auch hier später mehr davon.



ed
Paket

Das Paket heißt:

`ed-1.17-x86_64-3.tgz`

und entspricht damit vollumfänglich der Konvention zur Benennung von Slackware Paketen.

Diese ist:

`$Paketname-$Programmversion-$Architektur-$Build.tgz`



Paket

Kompression

Es gibt mehrere Möglichkeiten, ein Paket zu komprimieren.

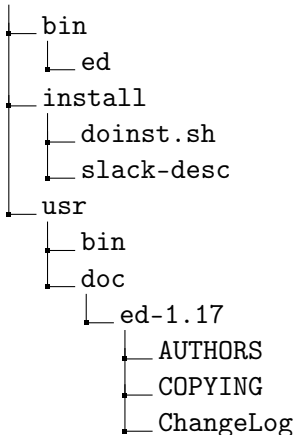
Erweiterung	Kompression per
tbz	bzip2
tlz	lzip
tgz	gzip
txz	xz (heute gängig)



Paket I

Inhalt

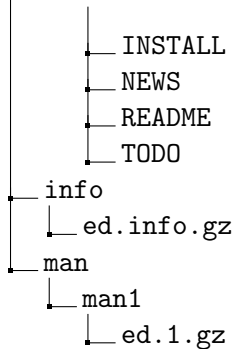
ed-1.17-x86_64-3.txz





Paket II

Inhalt





Paket

Inhalt

Besonders hervorzuheben sind hier die Dateien:

`install/doinst.sh`

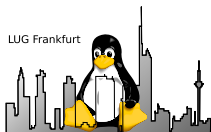
`install/slack-desc`



Inhalt

doinst.sh

Die Funktion wurde vorhin schon besprochen. Gezeigt werden kann sie trotzdem!



Inhalt

slack-desc

Auch diese Datei wurde vorhin einmal angeschnitten. Sie erfüllt folgende Formalien:

- # leitet eine Kommentarzeile ein
- Nach dem Kommentar (Hier steht in der Regel schon die Anleitung für diese Datei) kommt der `handy-ruler`. Dieser ist nur eine Hilfe
- Jetzt folgen genau 11 Zeilen, von denen jede gemäß `paketname: PaketbeschreibungstextMitMaximal79ZeichenProZeile` aufgebaut ist.



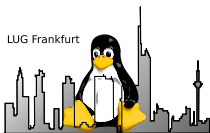
Inhalt

slack-desc von ed

HOW TO EDIT THIS FILE:

The "handy ruler" below makes it easier to edit a package description. Line
up the first '|' above the ':' following the base package name, and the '|'
on the right side marks the last column you can put a character in. You must
make exactly 11 lines for the formatting to be correct. It's also
customary to leave one space after the ':'.

```
|-----handy-ruler-----|  
ed: ed (text editor)  
ed:  
ed: GNU ed is an 8-bit clean, more or less POSIX-compliant implementation  
ed: of the standard Unix line editor. These days, full-screen editors  
ed: have rendered 'ed' mostly of historical interest. Nonetheless, it  
ed: appeals to a handful of aging programmers who still believe that  
ed: "Small is Beautiful".  
ed:  
ed:  
ed:  
ed:
```



slack-desc von ed

Best practices

- Die erste Zeile der Beschreibung enthält den Paketnamen mit einer wirklich extrem rudimentären Kurzbeschreibung
- Dann eine Leerzeile
- Dann eine Beschreibung des Paketes
- Sollte das Programmpaket eine Internetadresse sein eigen nennen dürfen, so wird dieselbe gerne in der letzten Zeile vermerkt.

Es gibt aber auch Helferlein, die valide [slack-desc](#) Dateien erstellen und prüfen.



Inhalt

slackbuild

Wie vorhin schon erwähnt ist das slackbuild ein bash Skript, das das Paket baut.

@Lars: Zeige ed's slackbuild auf der Leinwand!



Pakete

Source oder binary

Da stellt sich die Frage: Sourcepakete, wie Gentoo oder Fertigkompilat wie bei arch?

Beides!

Die Pakete aus dem offiziellen Repo werden binär gepackt, die Pakete aus dem Repo von SlackBuilds.org (SBo ist das community Repo von slackware) werden kompiliert, verlinkt und paketiert (alles mittels der slackbuild).

Nur Pakete, die im Bau sehr . . . streitbar sind, werden bei SBo als binärpaket angeboten (z. B. libreoffice).



Administration

Paketdaten im System

Die Daten zu den installierten- und nicht mehr installierten Paketen werden in

- `/var/log/packages/`
- `/var/log/pkgtools/removed_packages/`
- `/var/log/pkgtools/removed_skrpts/`

vorgehalten. In Textdateien.

@Lars: Zeig das mal!



Paket erstellen

Manuell

Im folgenden wird ein Paket komplett manuell erstellt.

Das Paket ist:

HALLO WELT!

LUG Frankfurt



Paket erstellen

Manuell

yippieh!



Paket erstellen

Programm

Der Quellcode des wirklich aufwendigen und gut strukturierten, sowie ausgiebig von diversen Experten getestet und auf Fehler kontrollierten Programmes ist:

```
#!/bin/sh  
echo "Hallo_Welt!"
```



Programm

Debugging

1. Die Datei `/root/Temp/hw.sh` erstellen und testen.
2. Die Datei `/root/Temp/slack-desc` erstellen und testen.



Hallo Welt!

Buildstage

Das Programm wird in eine buildstage eingebettet:

```
/root/Temp/buildstage
```

```
├── bin
│   └── hw.sh
├── install
└── slack-desc
```




HALLO WELT!

Packetieren

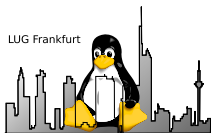
Das Paketieren erfolgt anschliessend mit `tar`, gepackt mittels `lz`.

Ein Befehl hierzu lautet dann:

```
root@rechner ~/Temp/# tar -cJ --xz -C buildstage/ -f ./hw-1.0-x86_64-1.txz  
bin install
```

Cave: Der Aufbau des Dateinamens wurde vorhin schon erklärt.

@Lars: Mach ma Paket, drüben! ... unn installier!



Paketieren

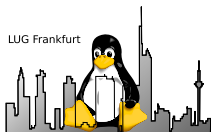
makepkg

Die Paketierung per `makepkg` ist einfacher und auch vorzuziehen, da verschiedene Prüfungen durchgeführt werden.

Also:

```
root@rechner ~/Temp # cd buildstage
```

```
root@rechner ~/Temp/buildstage # makepkg ../hw-1.0-X64_64-2.txz
```



Paketieren

Slacktrack

Die Paketierung mittels Slacktrack erfolgt über ein buildscript. Da die offiziellen Slackware Pakete auf diesen Buildscripts basieren, kann man sich jedes Sourcepaket herausgreifen und dieses inspizieren. Es wird genutzt, um den originalen Quellcode zu kompilieren, zu linkern, die Dokumentation auf die verschiedenen Verzeichnisse zu sortieren usw.

Kleiner Spoiler: Gegen Ende des Buildscriptes wird **makepkg** aufgerufen.



Hacks

Dateien eines Paketes finden

In installierten Paketen:

```
less /var/log/packages/[paketname]
```

In allen offiziellen Paketen:

Die Datei [MANIFEST.bz2](#) auf dem Installationsmedium im Verzeichnis [slackware64](#) enthält alle Dateien aller Pakete der Distribution.



Hacks

Liste aller installierten Pakete ausgeben

```
ls /var/log/packages/
```



Hacks

Liste aller deinstallierten Pakete ausgeben

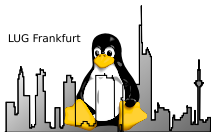
```
ls /var/log/removed_packages/
```



Fragen und Antworten

Sind noch Fragen?

Hier werden noch Fragen beantwortet, wenn möglich.



Links I

FraLUG: <http://www.lugfrankfurt.de>

Slackware: <http://www.slackware.com/>

ChangeLog: <http://www.slackware.com/changelog>

System downloaden: <http://www.slackware.com/getslack/>

Mirrorliste: <https://mirrors.slackware.com/>

Packages: <http://packages.slackware.com/>

Dokumentation: <http://docs.slackware.com/>

Dokumentation Pakete: [http://docs.slackware.com/slackware:
package_management](http://docs.slackware.com/slackware:package_management)



Links II

Offizielles Forum: <https://www.linuxquestions.org/questions/forumdisplay.php?forumid=14>

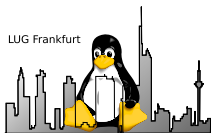
SlackBuilds: <https://slackbuilds.org/>

SlackBuilds git: <https://git.slackbuilds.org/slackbuilds/>

SlackBuilds ftp: <ftp://ftp.slackbuilds.org/>

Pakete von Eric Hameleers (AlienBob):

<http://www.slackware.com/~alien/slackbuilds/>



IRC

Folgende Kanäle kann man per IRC konsultieren:

Server: Libera.Chat

1. `##slackware`
2. `##slackware.de`
3. `#slackdocs`
4. Weitere Kanäle in weiteren Sprachen

Danke
fürs
Mitnehmen
und
Tschüß

