Linux-Kurs - Slackware *-current*

Nach Abschluss des Kursteils *Konfiguration* hast Du bereits ein lauffähiges, sicheres Linux-System und bist in der Lage Programme zu installieren sowie Konfigurationsänderungen vorzunehmen. Falls Du etwas ältere aber dafür ausgereifte Programme vorziehst oder wenig Aufwand in den Betrieb deines Computers stecken möchtest, bietet dir die stabile Version von Slackware die richtige Basis dafür.

Bei Slackware *-current* hingegen handelt es sich um die Entwickelervariante der Distribution. Dieser Zweig dient primär der Vorbereitung der nächsten stabilen Version. Es werden fortlaufend Pakete aktualisiert, was allerdings kurz vor dem Release der kommenden Version etwas nachlässt, da es dann primär um die Stabilisierung der Distribution geht. Viele begeisterte Slackware-Anwender nutzen den -current Zweig als Basis für ihr Linux-System. Gründe dafür gibt es viele, wie zum Beispiel die aktuellere Software und die ständige Erneuerung die dort stattfindet.

Distributionen wie *Archlinux* haben das sogenannte Rolling-Release-Verfahren zum Konzept erklärt, welches gänzlich ohne stabile Veröffentlichungsversionen auskommt. Im Gegensatz dazu ist Slackware grundsätzlich auf die Veröffentlichung von stabilen Versionen fokussiert und nicht als Rolling-Release ausgelegt. Bei einer reinen Rolling-Distribution wie *Archlinux* gibt es beispielsweise die erwähnten Stabilisierungsphasen nicht.

Für die Nutzung von -current als Basis für dein Linux-System braucht es zunächst einmal etwas Mut und den Willen dein System laufend zu aktualisieren. Als Lohn dafür erhältst Du eine sehr aktuelle Linux-Umgebung und kannst fortlaufend von diversen Weiterentwicklungen profitieren.

Im Folgenden wird beschrieben, wie Du deine bisherige Installation auf -current aktualisieren und entsprechend erweitern kannst. Es ist auch möglich eine vollständige Neuinstallation auf Basis von -current durchzuführen. Der Slackware-Entwickler alienBOB stellt entsprechende Installationsmedien zur Verfügung:
http://bear.alienbase.nl/mirrors/slackware/slackware64-current-iso/.

Die Vorgehensweise zur Installation und Einrichtung entspricht in den Grundsätzen der im Kurs beschriebenen Schritte. In -current wird der Dienst *rc.inetd* nicht mehr defaultmässig aktiviert. Daher entfallen einige der im Kursteils *Konfiguration* unter dem Abschnitt *Dienste* beschriebenen Schritte.

Voraussetzungen

Wir gehen davon aus, dass dein Linux-System dem im bisherigen Kursverlauf beschriebenen Setup entspricht und Du den Defaultkernel in der *huge*-Variante verwendest. Letzeres ist der Fall, wenn Du keine vom Kurs abweichenden Änderungen im Bereich des Kernels vorgenommen hast.

Die folgenden Arbeiten werden mit Root-Rechten ausgeführt. Starte daher bitte zunächst ein Root-Terminal:

```
su -
```

slackpkg Anpassungen

Zur Verwaltung der offiziellen Slackware-Pakete nutzen wir weiterhin den Paketmanager **slackpkg**. In der Datei */etc/slackpkg/mirrors* haben wir bisher einen Spiegelserver für die stabile Version von Slackware hinterlegt. Um auf -current zu wechseln müssen wir diesen auskommentieren und in der gleichen Datei deinen entsprechenden Mirror für -current aktivieren.

```
vi /etc/slackpkg/mirrors
```

Suche die Zeile mit dem aktiven Slackware64 Release-Mirror und füge am Zeilenanfang ein #-Zeichen ein.

Im weiteren Verlauf der Datei findest Du entsprechende Mirrors für die Slackware64 -current Variante. Entferne vor einem Spiegelserver deiner Wahl das Rautezeichen. Zusammengefasst kann die Konfiguration wie folgt aussehen:

```
# GERMANY (DE)
# ftp://ftp.gwdg.de/pub/linux/slackware/slackware64-14.2/
# http://ftp.gwdg.de/pub/linux/slackware/slackware64-14.2/
# ftp://ftp.tu-chemnitz.de/pub/linux/slackware/slackware64-14.2/
# http://ftp.tu-chemnitz.de/pub/linux/slackware/slackware64-14.2/
# ftp://sunsite.informatik.rwth-aachen.de/pub/comp/Linux/slackware/slackware64-14.2/
# http://sunsite.informatik.rwth-aachen.de/ftp/pub/comp/Linux/slackware/slackware64-14.2/
#
# ...
#
#------------------------------------------------------------
# Slackware64-current
#------------------------------------------------------------
# USE MIRRORS.SLACKWARE.COM (DO NOT USE FTP - ONLY HTTP FINDS A NEARBY MIRROR)
# https://mirrors.slackware.com/slackware/slackware64-current/
#
#
# Here are some individual mirrors that can be used instead of the
# redirector at mirrors.slackware.com if necessary ; note that this
# list is not guaranteed to be up-to-date
#
# ...
#
# GERMANY (DE)
# ftp://ftp.fu-berlin.de/unix/linux/slackware/slackware64-current/
ftp://ftp.gwdg.de/pub/linux/slackware/slackware64-current/
# http://ftp.gwdg.de/pub/linux/slackware/slackware64-current/
# ftp://ftp.tu-chemnitz.de/pub/linux/slackware/slackware64-current/
# http://ftp.tu-chemnitz.de/pub/linux/slackware/slackware64-current/
# ftp://sunsite.informatik.rwth-aachen.de/pub/comp/Linux/slackware/slackware64-current/
# http://sunsite.informatik.rwth-aachen.de/ftp/pub/comp/Linux/slackware/slackware64-current/
# ftp://wrz1013.rz.uni-wuerzburg.de/pub/MIRROR/slackware/slackware64-current/
# http://wrz1013.rz.uni-wuerzburg.de/pub/MIRROR/slackware/slackware64-current/
# GREECE (GR)
# ftp://ftp.cc.uoc.gr/mirrors/linux/slackware/slackware64-current/
# http://ftp.cc.uoc.gr/mirrors/linux/slackware/slackware64-current/
# ftp://ftp.otenet.gr/pub/linux/slackware/slackware64-current/
# http://ftp.otenet.gr/linux/slackware/slackware64-current/
```

Hinweis: Es darf nur ein Mirror aktiv konfiguriert sein.

Blacklist

Es empfiehlt sich zu diesem Zeitpunkt die mittels sbopkg erstellten Pakete sowie sbopkg selbst zu blacklisten (das Paket endet auf *_wsr*).

Füge dazu die folgenden Zeilen zu am Ende der Konfigurationsdatei */etc/slackpkg/blacklist* ein:

```
[0-9]+_SBo
[0-9]+_wsr
```

`vi /etc/slackpkg/blacklist`

```
# You can blacklist using regular expressions.
#
# Don't use *full* regex here, because all of the following
# will be checked for the regex: series, name, version, arch,
# build and fullname.
#
# This one will blacklist all SBo packages:
[0-9]+_SBo
[0-9]+_wsr
```

Führe nach den Anpassungen eine Aktualisierung der Repository-Metadaten und ein Update von **slackpkg** selbst durch:

```
slackpkg update
```

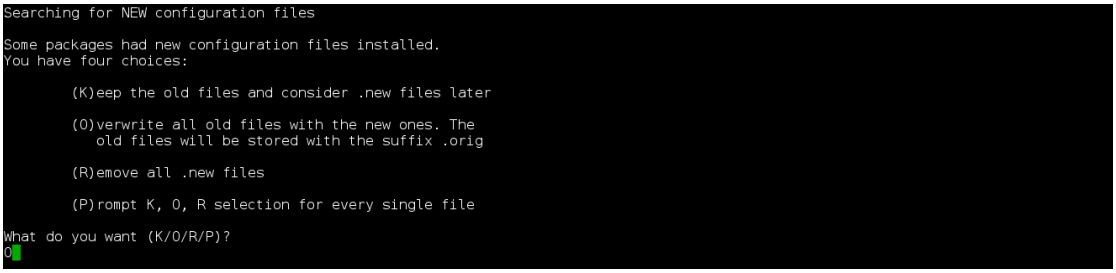
slackpkg erkennt, dass Du auf die Entwicklerversion gewechselt hast und fordert zu einer Bestätigung auf. Gebe **Y** ein um fortzufahren:

```
root@darkstar:~# slackpkg update

You have selected a mirror for Slackware -current in /etc/slackpkg/mirrors,
but Slackware version 14.2 appears to be installed.

Slackware -current is the development (i.e. unstable) tree.

Is this really what you want?

To confirm your choice, press Y, else press N. Then, press Enter: Y
```

```
slackpkg upgrade slackpkg
```

Wenn ein Paket bei einem Upgrade neue Konfigurationsdateien mit sich bringt wirst Du wie üblich gefragt, ob du die alten Versionen behalten möchtest oder die neue Version der jeweiligen Konfigurationsdatei übernommen werden soll.

In Falle von slackpkg ist es empfehlenswert die neuen Konfigurationsdateien zu übernehmen, da diese auch neue Mirrorinformationen enthalten:



Dies hat allerdings zur Folge, dass deine zuvor gemachten Änderungen wieder überschrieben werden und du den Mirror für -current erneut in der */etc/slackpkg/mirrors* aktivieren musst. Gehe dazu wie zuvor beschrieben vor.

Hinweis: Alle Pakete die nicht offiziell Bestandteil der Distribution sind müssen nach der Aktualisierung auf -current neu erstellt beziehungsweise installiert werden.

Aktualisierung auf -current

Die Aktualisierung sollte sicherheitshalber in einer tty-durchgeführt werden. Du kannst diese beispielsweise mit Hilfe von `Ctrl + Alt + F1` öffnen. Melde dich dort als Root-Benutzer an.

Zunächst muss slackpkg mit dem `update` Parameter ausgeführt werden, um die Anwendung über die Änderungen zu informieren:

```
slackpkg update
```

Neue Versionen von Slackware werden in der Regel mit einer aktuelleren Version der *GNU C libraries* ausgeliefert. Viele der enthaltenen Pakete wurden gegen diese aktualisierte *glibc* Version compiliert. Um Fehler bei der Paketinstallation zu vermeiden sollte daher bei einem Wechsel auf -current zuerst *glibc* aktualisiert werden:

```
slackpkg upgrade glibc-solibs
```

Nach der *glibc* Aktualisierung kannst du mit Hilfe des *install-new* Parameters alle neu hinzugekommenen Pakete installieren, die bisher nicht Bestandteil der Distribution waren:

```
slackpkg install-new
```

Das eigentlich Upgrade startest du mit dem *upgrade-all* Parameter:

```
slackpkg upgrade-all
```

Die Aktualisierung kann je nach Umfang deines Systems einige Zeit in Anspruch nehmen.

Wie bekannt wirst Du nach der Installation der Pakete gefragt, ob die bisherigen Konfigurationsdateien beibehalten oder durch die neuen Standardkonfiguration ersetzt werden sollen. Mit **Prompt** wirst Du für jede neue Konfigurationsdatei gefragt ob Du die neue Datei übernehmen möchtest oder die bisherige behalten willst. Es ist auch möglich die Dateien zu vergleichen oder zu *mergen*, also zusammenzuführen. Mit **Overwrite** würdest Du alle neuen Konfigurationsdateien übernehmen, was zur Folge hätte, dass Du die bisherigen Einstellungen, wie zum Beispiel den default Runlevel, die NTP-Konfiguration oder die Locales-Einstellungen erneut konfiguriert werden müssen. Dafür kannst Du dir sicher sein, dass alle Neuerungen in Konfigurationsdateien auch übernommen werden.

Führe nach jeder Aktualisierung des Kernels `lilo` aus um den Bootloader neu zu schreiben:

```
lilo
```

Mit Hilfe des *clean-system* Parameters entfernst Du abschliessend alle Pakete die nicht mehr Bestandteil der Distribution sind:

```
slackpkg clean-system
```

Nach der Aktualisierung kannst Du dein Linux-System neu starten.

Slackpkg+

Ein alternativer Ansatz zur Verwaltung von Zusatzrepositories ist die slackpkg-Erweiterung mit dem Namen *Slackpkg+*. Du findest sie auf der Homepage des Entwicklers:

http://slakfinder.org/slackpkg+.html

Zur Installation klicke auf den Download-Link und wähle aus der Liste die neueste verfügbare Version aus. Speichere das Paket im Download-Ordner deines Benutzers ab. Öffne ein Benutzer-Terminal und installiere das Paket mit Hilfe des `upgradepkg` Kommandos:

```
sudo upgradepkg --install-new ~/Downloads/slackpkg+*
```

Wechsle danach zurück in das Root-Terminal.

Multilib

Falls Du wie im Kursteil *Programme* beschrieben *Multilib* verwendest, hast Du bisher das Script */usr/local/sbin/alien-multilib-rsync.sh* zur Synchronisation und Installation der Multilib-Pakete genutzt. Es prüft den Mirror jeweils auf Veränderungen. Sollte neue Pakete vorhanden sein, wird mit Hilfe des **rsync** Kommandos ein Abgleich durchgeführt. Dabei werden nur die geänderten Pakete heruntergeladen. Installiert beziehungsweise aktualisiert werden hingegen jeweils alle Pakete, auch wenn sich nur wenige Änderungen ergeben haben. Da die Frequenz der Änderungen im Multilib-Repository unter Slackware -current deutlich höher ist, ist die Nutzung des Scriptes für diesen Distributionszweig nicht empfehlenswert. Sinnvoller ist die Verwendubg von *Slackpkg". Das bisherige Sync-Script kannst Du mit Hilfe des folgenden Kommandos löschen:

```
\rm /usr/local/sbin/alien-multilib-rsync.sh
```

Hinweis: Das `\`-Zeichen vor dem `rm` unterdrückt die Nutzung des von uns gesetzten Aliases und löscht die Datei somit ohne Nachfrage.

Die Konfiguration von Slackpkg+ erfolgt in der Datei */etc/slackpkg/slackpkgplus.conf*.

```
vi /etc/slackpkg/slackpkgplus.conf
```

Um Multilib mit Slackpkg+ zu verwalten, entferne das #-Kommentarzeichen vor der Zeile:

```
MIRRORPLUS['multilib']=http://bear.alienbase.nl/mirrors/people/alien/multilib/current/
```

Füge des weiteren *multilib* zur REPOPLUS Variable hinzu:

```
REPOPLUS=( slackpkgplus multilib )
```

Slackpkg+ nutzt ein Prioritäten-System zur Verwaltung der Repositories. Du kannst das multilib-Repository bevorzugen, indem Du es in der Variable PKGS_PRIORITY hinterlegst:

```
PKGS_PRIORITY=( multilib )
```

```
#PKGS_PRIORITY=( myrepo )
#
# if you have two reporitories to give priority you must set both in the same line
#PKGS_PRIORITY=( myrepo restricted:vlc )
#
# if you want to install 'ktown' repository you must set it here
#PKGS_PRIORITY=( ktown )
# and DO NOT MISS to read special instruction on /usr/doc/slackpkg+-*/repositories.txt
#
# If you want a multilib system, uncomment the multilib repository and set:
PKGS_PRIORITY=( multilib )
#
# (Use /usr/doc/slackpkg+-*/setupmultilib.sh to setup a multilib configuration)
#
# For both multilib and ktown set
#PKGS_PRIORITY=( multilib ktown )

# Otherwise you can try to upgrade a package from a repository that contains a package with the
# same tag of the already installed package. Tipically that means to upgrade a package from the
# same author of the already installed package.
# Note that this method may not works properly where two repositories contains a package with the
# same tag.
# Set TAG_PRIORITY to 'on' to enable this function
TAG_PRIORITY=off

# List repositories you want to use (defined below)
# remember to launch 'slackpkg update' if you modify that row.
#REPOPLUS=( slackpkgplus restricted alienbob slacky )
REPOPLUS=( slackpkgplus multilib )


# Define mirrors (uncomment one or more mirror; remember to add it to REPOPLUS)
# GPG Note: after adding/renaming a repository, you must to run 'slackpkg update gpg'
#           some repositories as salixos, have a partial GPG support;
#           for that repositories you may need to run slackpkg with 'slackpkg -checkgpg=off ...'

# Slackware 14.2 - x86_64
#MIRRORPLUS['multilib']=http://bear.alienbase.nl/mirrors/people/alien/multilib/14.2/
#MIRRORPLUS['alienbob']=http://bear.alienbase.nl/mirrors/people/alien/sbrepos/14.2/x86_64/
#MIRRORPLUS['restricted']=http://bear.alienbase.nl/mirrors/people/alien/restricted_sbrepos/14.2/x86_64/
#MIRRORPLUS['slacky']=http://repository.slacky.eu/slackware64-14.2/

# use this to keep the slackpkg+ package updated to the latest stable release
MIRRORPLUS['slackpkgplus']=http://slakfinder.org/slackpkg+/

# use the development branch to use the mainline version and help develop by reporting bugs.
#MIRRORPLUS['slackpkgplus']=http://slakfinder.org/slackpkg+dev/


# Slackware current - x86_64
MIRRORPLUS['multilib']=http://bear.alienbase.nl/mirrors/people/alien/multilib/current/
```

Falls Du *alien* und *compat32* in der Datei */etc/slackpkg/blacklist* aufgeführt hast, musst Du die entsprechenden Einträge vor der Nutzung von Multilib mit Slackpkg+ entfernen oder auskommentieren:

```
vi /etc/slackpkg/blacklist
```

```
# This will blacklist compat32 and alienBOB's packages
# [0-9]+alien
# [0-9]+compat32
```

Nach Änderungen an den zu verwendeten Repositories, müssen die GPG-Entwicklerschlüssel und die Repository-Metadaten aktualisiert werden:

```
slackpkg update gpg
slackpkg update
```

Mit dem `upgrade-all` slackpkg Parameter werden die *gcc* Pakete aus dem Slackware-Repository durch die Multilib Pakete ersetzt. Daraufhin kannst Du alle compat32 aus dem Multilib-Repository installieren:

```
slackpkg upgrade-all
slackpkg install multilib
```

SBo

Die meisten Pakete die Du zuvor mittels sbopkg aus den SlackBuild Scripten erstellt hast, werden nun nicht mehr funktionieren. Die Unterstützung von -current in sbopkg ist experimentell. Um diese zu aktivieren, öffne ein Root-Terminal und bearbeite die Datei */etc/sbopkg/sbopkg.conf*.

```
su -
vi /etc/sbopkg/sbopkg.conf
```

Entscheidend sind die Variablen: *REPO_BRANCH* und *REPO_NAME*. Diese müssen wie folgt konfiguriert werden:

```
REPO_BRANCH=${REPO_BRANCH:-current}
REPO_NAME=${REPO_NAME:-SBo-git}
```

```
# Other variables:
CLEANUP=${CLEANUP:-NO}
DEBUG_UPDATES=${DEBUG_UPDATES:-0}
KEEPLOG=${KEEPLOG:-YES}
MKDIR_PROMPT=${MKDIR_PROMPT:-YES}
NICE=${NICE:-10}
REPO_BRANCH=${REPO_BRANCH:-current}
REPO_NAME=${REPO_NAME:-SBo-git}
```

Führe danach mit Hilfe des `-r` Parameters eine Synchronisation des Repositories aus:

`sbopkg -r`

Beim ersten Aufruf nach den Anpassungen wirst Du darauf hingewiesen, dass möglicherweise einige Verzeichnisse noch nicht existieren. Bestätige die Erstellung durch Eingabe von **C**:

```
root@darkstar:~# sbopkg -r

The following directories do not exist:

Variable                  Assignment
--------                  ----------
REPO_{ROOT,NAME} --------> /var/lib/sbopkg/,SBo-git
SRCDIR -----------------> /var/cache/sbopkg
TMP --------------------> /tmp/SBo

You can have sbopkg create them or, if these values are incorrect, you can
abort to edit your config files or pass different flags.

(C)reate or (A)bort?: C
```

sqg

Auch das im sbopkg Paket enthaltene Tool **sqg** zur Erstellung der Queuefiles muss angepasst werden.

`vi /usr/sbin/sqg`

Die anzupassenden Variablen sind hier ebenfalls *REPO_NAME* und *REPO_BRANCH*:

```
REPO_NAME=${REPO_NAME:-SBo-git}
REPO_BRANCH=${REPO_BRANCH:-current}
```

```
#QUEUEDIR=${QUEUEDIR:-/var/lib/sbopkg/queues}
REPO_ROOT=${REPO_ROOT:-/var/lib/sbopkg}
REPO_NAME=${REPO_NAME:-SBo-git}
REPO_BRANCH=${REPO_BRANCH:-current}
#SKIP_EMPTY=${SKIP_EMPTY:-NO}

### NO CHANGES SHOULD BE NECESSARY BELOW THIS LINE ###
```

Nach der Umstellung des Repositories müssen die Queuefiles neu erstellt werden. Dies kann entweder durch die Angabe einzelner Queues geschehen, wie zum Beispiel für transmission:

`sqg -p transmission`

oder wie bekannt mit Hilfe des `-a` Parameters für alle verfügbaren Queuefiles:

`sqg -a`

Hinweis: Dieser Vorgang kann sehr lange dauern.

Wie Du vielleicht am Namen *SBo-git* erkannt hast, basiert die sbopkg-Version für -current auf einem GIT-Repository. Dieses wird bei grösseren Änderungen vollständig gelöscht und neu erstellt, was dazu führen kann, dass das git-Programm welches sbopkg im Hintergrund verwendet, mit diesen Änderungen nicht umgehen kann.

Daher wird empfohlen vor jedem Abgleich des Repositories mittels `sbopkg -r` folgenden Befehl auszuführen:

```
rm -fR /var/lib/sbopkg/SBo-git
sbopkg -r
```

Auch die Queuefiles sollten regelmässig aktualisiert werden. Vor jeder Erstellung eines Paketes anhand eines Queuefiles, sollte letzeres mittels `sqg -p $PAKETNAME` auf den aktuellen Stand gebracht werden.

SBo Pakete ermitteln

Um eine Liste aller Pakete die Du mit sbopkg installiert hast zu erstellen, kannst Du wie folgt vorgehen:

```
find /var/log/packages/*_SBo -printf "%f\n" | awk -F'-[0-9]' '{print $1}' > /root/mysbopackages
```

Der **find** Befehl sucht im Verzeichnis */var/log/packages* alle Dateien die mit *_SBo* enden und gibt diese aus. Die Ausgabe würde ohne weitere Parameter den gesamten Pfad der gefundenen Dateien anzeigen. Mit Hilfe von `-printf "%f\n"` schränken wir die Ausgabe auf den tatsächlichen Dateinamen ein (`%f`) und fügen nach jeder ausgegebenen Zeile mit Hilfe von `\n` einen Zeilenumbruch hinzu.

Um eine Paketliste zu erhalten, benötigen wir allerdings nur die Paketnamen ohne die Versionsnummern. Dies erreichen wir indem wir die Ausgabe des find Befehls mit dem **awk** Kommando bearbeiten. Der Parameter `-F'-[0-9]'` gibt an, welche Zeichen von awk als Trennzeichen interpretiert werden sollen. Anhand dieser Informationen teilt awk die Ausgabe in verschiedene durchnummerierte Bereiche auf. In diesem Fall soll das Kommando ein `-` Zeichen gefolgt von einer beliebigen Zahl als Trennelement nutzen. `'{print $1}` gibt den ersten Block der aufgetrennten Elemente aus. `$0` würde für die gesamte unbearbeitete Ausgabe stehen.

Die Ausgabe leiten wir mit Hilfe vom `> /root/mysbopackages` in eine Datei um.

Diese kannst du dir beispielsweise mit dem **cat** Kommando anzeigen lassen:

```
cat /root/mysbopackages
```

Die Paketliste kann dir Hinweise geben, welche Paket Du neu erstellen und installieren musst.

Alte SBo Pakete bereinigen

Nach Erstellung der Paketliste wird empfohlen alle alten mittels **sbopkg** installierten Pakete zu entfernen und danach wie bekannt neu zu erstellen und zu installieren. Zum Entfernen kannst Du den Blacklist Eintrag `[0-9]+_SBo` in der Datei */etc/slackpkg/blacklist* löschen und danach slackpkg mit dem `clean-system` Parameter ausführen:

```
slackpkg clean-system
```

Es sollten alle Pakete mit der Endung *_SBo* zur Deinstallation angeboten werden.

Nun kannst Du die zuvor unter der stabilen Variante von Slackware mittels **sbopkg** installierten Pakete neu erstellen.

Blacklist für SBo-git

Damit **slackpkg clean-system** die SBo-git Pakete nicht zur Deinstallation vorschlägt, muss ein Blacklist Eintrag in der */etc/slackpkg/blacklist* Datei hinterlegt werden:

[0-9]+ponce

```
vi /etc/slackpkg/blacklist
```

```
# You can blacklist using regular expressions.
#
# Don't use *full* regex here, because all of the following
# will be checked for the regex: series, name, version, arch,
# build and fullname.
#
# This one will blacklist all SBo packages:
[0-9]+ponce
[0-9]+_wsr
```

Besonderheiten

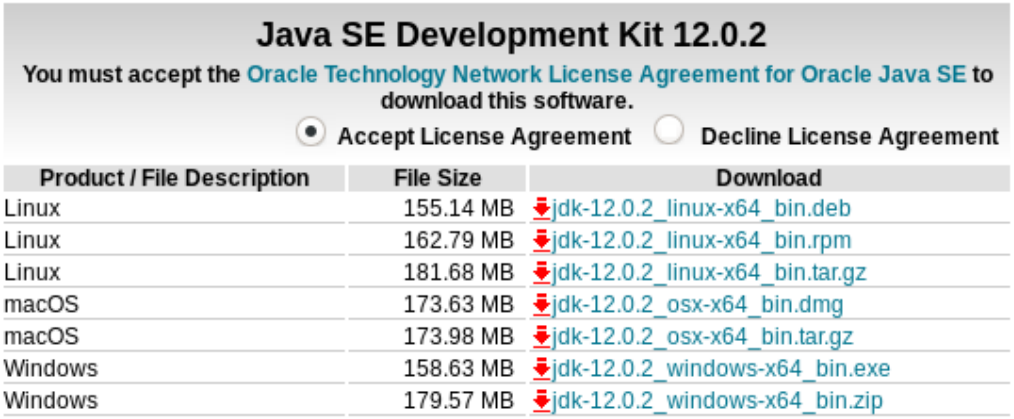
Einige der SlackBuilds werden möglicherweise unter -current nicht mehr funktionieren. Die Entwickler und Betreiber des Repositories geben sich allerdings sehr viel Mühe, möglichst alle SlackBuilds lauffähig zu halten. Für manche Pakete ergeben sich allerdings Änderungen im Gegensatz zur stabilen Variante:

LibreOffice

Das *LibreOffice* SlackBuild-Script erstellt die Bürosoftware vollständig aus dem Quelltext. In -current wird als Abhängigkeit nicht mehr *openjdk7* sondern das offizielle JDK von Oracle aufgeführt. Die Installation des *jdk* Paketes ist nicht ohne weiteres möglich, da die Sourcen erst nach einem Login und dem Akzeptieren der Nutzungsbedingungen herunterladbar sind. Dieser Prozess lässt sich nicht mittels sbopkg automatisieren und erfordert manuelle Vorarbeit.

Öffne zunächst im Internetbrowser die Downloadseite von Oracle:
https://www.oracle.com/technetwork/java/javase/downloads/index.html

Wähle unterhalb von *Oracle JDK* den Punkt **Download** und klicke dort auf *Accept License Agreement*:



Java SE Development Kit 12.0.2

You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.

(•) Accept License Agreement ( ) Decline License Agreement

Product / File Description	File Size	Download
Linux	155.14 MB	jdk-12.0.2_linux-x64_bin.deb
Linux	162.79 MB	jdk-12.0.2_linux-x64_bin.rpm
Linux	181.68 MB	jdk-12.0.2_linux-x64_bin.tar.gz
macOS	173.63 MB	jdk-12.0.2_osx-x64_bin.dmg
macOS	173.98 MB	jdk-12.0.2_osx-x64_bin.tar.gz
Windows	158.63 MB	jdk-12.0.2_windows-x64_bin.exe
Windows	179.57 MB	jdk-12.0.2_windows-x64_bin.zip

An dieser Stelle musst Du nichts weiter tun.

Suche nun auf SlackBuilds.org nach dem *jdk* Paket:

https://slackbuilds.org/result/?search=jdk&sv=

Wähle dort *Source Downloads (64bit)*. Der angegebene Link verweist auf die im SlackBuild hinterlegte JDK-Version.

Für den Download benötigst Du seit einiger Zeit zusätzlich einen Oracle-Account. Zur Erstellung empfehlen wir die Verwendung einer *Wegwerf Emailadresse* wie zum Beispiel:
https://10minutemail.com

Speichere die Datei in deinem Download-Ordner ab und verschiebe sie daraufhin mit Root-Rechten in das Verzeichnis */var/cache/sbopkg*. Falls **sbopkg** in dem Pfad bereits die heruntergeladenen Quellen vorfindet, werden diese verwendet und kein Downloadversuch gestartet.

```
sudo mv ~/Downloads/jdk-* /var/cache/sbopkg
```

Daraufhin solltest Du LibreOffice anhand des Queuefiles erstellen können:

```
sbopkg -k -i LibreOffice
```

System aktualisieren

Die Vorgehensweise zur Aktualisierung deines -current Systems entspricht den zuvor geschilderten Schritten. Diese solltest Du regelmässig durchführen. Alle Kommandos müssen mit Root-Rechten ausgeführt werden.

Lese dir zunächst den Changelog durch. Dort findest Du wichtige Hinweise zu Veränderungen. Öffne dazu die folgende URL im Firefox-Browser:

ftp://ftp.osuosl.org/pub/slackware/slackware64-current/ChangeLog.txt

Führe daraufhin mir Root-Rechten die Aktualisierung aus, welche aus den folgenden Kommandos besteht:

```
slackpkg update
slackpkg install-new
slackpkg upgrade-all
slackpkg clean-system
```

Solltest Du Multilib verwenden führe nach dem `slackpkg upgrade-all` folgenden Befehl aus um mögliche neue Pakete aus dem Multilib-Repository zu installieren:

`slackpkg install multilib`

Falls der Kernel aktualisert worden ist, führe zusätlich `lilo` aus um den Bootloader neu zu schreiben.

`lilo`

Auf diese Weise kannst Du dein System aktuell halten und von den Neuerungen in -current profitieren.

Schriftdarstellung

In Slackware -current hat sich die Darstellung der Schriften mittels des Renderes *FreeType* geändert. Des weiteren wurden andere Standardschriftarten gewählt. Die neuen Schriften sind sehr schmal geschnitten und teilweise schlecht lesbar. Um die Darstellung zu verbessern kannst Du die Standardschriftarten von *Liberation* auf *DejaVu* umstellen.

Erstelle zunächst eine Kopie der Schriften-Konfigurationsdatei für *Latin*-Fonts:

`sudo cp /etc/fonts/conf.d/60-latin.conf /etc/fonts/conf.d/60-latin.conf_liberation`

Bearbeite daraufhin die Konfigurationsdatei mit dem Editor vim:

`sudo vi /etc/fonts/conf.d/60-latin.conf`

Die Schriften werden in der Datei in Abschnitte gruppiert und von oben nach unten abgearbeitet. Das heisst der erste Eintrag in einem Abschnitt definiert die Standardschrift. Wir möchten die bevorzugte Schriftart für *serif*, *sans-serif* und *monospace* anpassen.

Zuoberst wurde dort jeweils die *Liberation* Schriftart definiert. Wir möchten erreichen, dass stattdessen *DejaVu* verwendet wird.

Du könntest die entsprechenden Einträge an erster Stelle in den jeweiligen Abschnitten hinterlegen und die darunterliegende Definition von *DejaVu* löschen. Mit dem Editor vi geht die Bearbeitung jedoch einfacher.

Gehe mit dem Cursor im ersten Abschnitt *serif* auf die Zeile *DejaVu Serif* und drücke im vi nacheinander folgende Tasten `Esc dd`. Mit *Esc* gelangst du wie gewohnt in den Befehlsmodus. *dd* löscht die Zeile und nimmt sie gleichzeitig in den Buffer auf. Platziere den Cursor daraufhin in der Zeile **unterhalb** der Du die ausgeschnittene Zeile einfügen möchtest, in diesem Falle im Abschnitt *serif* und drücke `p`.

Gehe für die *DejaVu* Einträge in den Abschnitten *sans-serif* und *monospace* analog vor. Die bearbeiteten Abschnitte sollten nun wie folgt aussehen:

```
<description>Set preferable fonts for Latin</description>
      <alias>
              <family>serif</family>
              <prefer>
                      <family>DejaVu Serif</family>
                      <family>Liberation Serif</family>
                      <family>DejaVu Serif</family>
                      <family>Bitstream Vera Serif</family>
                      <family>Times New Roman</family>
                      <family>Thorndale AMT</family>
                      <family>Luxi Serif</family>
                      <family>Nimbus Roman No9 L</family>
                      <family>Nimbus Roman</family>
                      <family>Times</family>
              </prefer>
      </alias>
      <alias>
              <family>sans-serif</family>
              <prefer>
                      <family>DejaVu Sans</family>
                      <family>Liberation Sans</family>
                      <family>Bitstream Vera Sans</family>
                      <family>Verdana</family>
                      <family>Arial</family>
                      <family>Albany AMT</family>
                      <family>Luxi Sans</family>
                      <family>Nimbus Sans L</family>
                      <family>Nimbus Sans</family>
                      <family>Helvetica</family>
                      <family>Lucida Sans Unicode</family>
                      <family>BPG Glaho International</family> <!-- lat,cyr,arab,geor -->
                      <family>Tahoma</family> <!-- lat,cyr,greek,heb,arab,thai -->
              </prefer>
      </alias>
      <alias>
              <family>monospace</family>
              <prefer>
                      <family>DejaVu Sans Mono</family>
                      <family>Liberation Mono</family>
                      <family>Bitstream Vera Sans Mono</family>
                      <family>Inconsolata</family>
                      <family>Andale Mono</family>
                      <family>Courier New</family>
                      <family>Cumberland AMT</family>
                      <family>Luxi Mono</family>
                      <family>Nimbus Mono L</family>
                      <family>Nimbus Mono</family>
                      <family>Nimbus Mono PS</family>
                      <family>Courier</family>
              </prefer>
      </alias>
```

Hinweis: vim bietet auch die Möglichkeit einzelne oder mehrere Zeilen zu kopieren. Der Befehl lautet `Esc 1 yy` zum kopieren und `p` zum einfügen, wobei die `1` für die Anzahl der zu kopierenden Zeilen steht.

Um das Rendering-Verhalten von Freetype umzustellen, kannst Du die Datei */etc/profile.d/freetype.sh* bearbeiten und dort das #-Kommentarzeichen vor der Interpreter-Version 35 entfernen:

```
sudo vi /etc/profile.d/freetype.sh
```

```
export FREETYPE_PROPERTIES="truetype:interpreter-version=35"
```

```
#!/bin/sh
# Configure Freetype properties. Here this is used to set the default mode
# for font hinting. Other controllable properties are listed in the section
# 'Controlling FreeType Modules' in the reference's table of contents.
#
# Three hinting settings are available:

# This is the classic hinting mode used in Freetype 2.6.x:
export FREETYPE_PROPERTIES="truetype:interpreter-version=35"

# This is Infinality mode, which was never enabled by default. It is slower
# than the new subpixel hinting mode, but said to be more accurate:
#export FREETYPE_PROPERTIES="truetype:interpreter-version=38"

# This is the new default subpixel hinting mode used in Freetype 2.7.x. It is
# derived from the Infinality code base stripped to the bare minimum with all
# configurability removed in the name of speed and simplicity:
#export FREETYPE_PROPERTIES="truetype:interpreter-version=40"
```
