

Slackware lezing

door Eric Hameleers

<http://slackware.com/~alien/>
alien@slackware.com

NLLGG bijeenkomst
Liefland College te Utrecht
http://www.nllgg.nl/bijeenkomst_20100403

03 April 2010

Overzicht van de presentatie

Slackware lezing

- Introductie
- Een beetje geschiedenis
 - Hoe het begon
 - Slackware filosofie
- Ontwikkeling
 - Het team
 - Package management
 - Package format en support tools
- Extra software installeren
 - Community repositories
 - SlackBuild scripts
- Meer bronnen

Overzicht van de presentatie

Slackware lezing

- Introductie
- Een beetje geschiedenis
 - Hoe het begon
 - Slackware filosofie
- Ontwikkeling
 - Het team
 - Package management
 - Package format en support tools
- Extra software installeren
 - Community repositories
 - SlackBuild scripts
- Meer bronnen

Deze presentatie wordt bestuurd door een Wii controller ;-)

Introductie

Eric Hameleers

- Slackware gebruiker sinds 1996
- Maakt deel uit van het Slackware core team sinds 2006



Hoe het begon

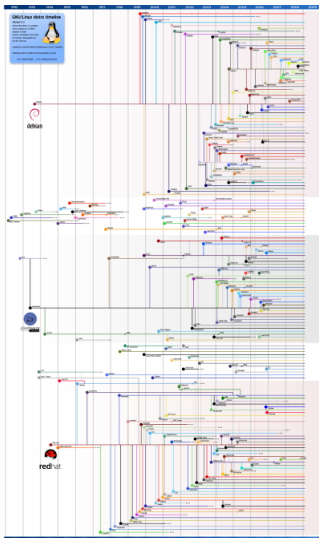
- Afgeleid van SoftLandingSystems (SLS) — Slackware is wellicht de eerste "fork" distributie.
- Begon als een serie verbeteringen en bugfixes voor SLS die nooit door Peter MacDonald werden toegevoegd aan SLS. Patrick gebruikte zijn universiteit's ftp server voor de verspreiding.
- Deze aangepaste versie van SLS ontwikkelde zich terwijl die van SLS stakte, en door de grote vraag werd het uitgebracht als "Slackware 1.0" op 16 Juli 1993. Om afstand te nemen van SLS werd de installer herschreven voor het nieuwe "Slackware", gebruikmakend van een programma genaamd "dialog" (mede door Patrick geschreven). De op dialog gebaseerde installer kwam voor het eerst voor in Slackware 1.1.0.

Tijdlijn van Slackware releases

Een paar hoogtepunten uit Slackware's geschiedenis

1.0	16/07/1993 (eerste release)
1.1.2	05/02/1994 (met dialog-gebaseerde installer)
1.2.0.1	01/04/1994 (met Linux kernel 1.0)
3.3	11/07/1997 (vroegste versie die nog te vinden is op mirrors, met linux 2.0.30)
4.0.0	17/05/1999 (eerste release met KDE)
7.1	22/06/2000 (eerste release met Gnome)
10.1	02/02/2005 (Gnome werd afgevoerd, 2.6 kernel verscheen in /testing)
11.0	01/10/2006 (laatste met gespecialiseerde 2.4 kernel boot disks, en monolitische X.Org)
12.1	02/05/2008 (ondersteuning voor HTTP/FTP netwerk installatie en LVM/LUKS disk volumes in de installer)
13.0	28/08/2009 (KDE4 vervangt KDE3, 64-bit port)

Linux Distro Tijdlijn - <http://futurist.se/gldt/>



Stamboom van
Linux distributies

Slackware Filosofie

- Slackware is nog steeds een erg “traditionele” linux distributie — dat kon verwacht worden dankzij de reputatie de meest “UNIX-like” distributie van linux te zijn.
- Stabiliteit en gebruiksgemak zijn de hoofd-doelstellingen van Slackware
 - Waar mogelijk wordt ongemodificeerde source code gebruikt
 - Niet gebonden aan een tevoren afgesproken “release schedule” — release master wordt gemaakt wanneer de tijd rijp is, met het doel minimaal een release per jaar uit te brengen
 - Transparante configuratie — goed gedocumenteerde configuratie bestanden
 - Iedere applicatie wordt onafhankelijk geconfigureerd zoals de “upstream” ontwikkelaars bedoelen — de documentatie van de ontwikkelaars moet correct en compleet zijn
 - Geen applicatie-patches voor Slackware-specifieke configuratie — Slackware heeft geen “globaal” configuratie-gereedschap

Slackware Filosofie

- Conservatief ontwikkelmodel — enkel met goede redenen zal stabiele en functionele software vervangen worden door nieuwere, ongeteste en wellicht onstabiele software (denk X.Org, KDE4)

Dat betekent niet dat Slackware vol zit met OUDE software!

Slackware is over het algemeen net zo “up to date” (zo niet **meer** up to date) als iedere andere distributie

- Slackware is niet bedoeld als een Windows vervanger — er zijn andere linux distributies die dat veel beter kunnen ;-)
- Slackware biedt een systeem dat intuïtief is en eenvoudig te gebruiken voor ervaren UNIX beheerders, maar desondanks niet moeilijk te leren is voor de nieuwe linux gebruiker

Ontwikkeling

Hoe ziet Slackware's ontwikkelmodel er uit?

- Commerciële distributie, *enkel* gefinancierd door online CD/DVD verkoop. Aan de andere kant, altijd vrijelijk te downloaden.
- Gesloten ontwikkeling
 - Geen bugzilla, geen publieke code repository, geen code contributors
 - Geen procedures om ontwikkelaar te "worden"
- "Benevolent Dictatorship" (iets dat Pat bestrijdt :-)
Het eindoordeel over iedere wijziging aan Slackware berust bij Patrick Volkerding. Maar... hij staat open voor discussie.
- Geen een-mans show!
Slackware heeft een "core team" dat bijdraagt aan:
 - Niet-publieke tests van nieuwe, of gewijzigde, pakketten
 - Wijzigingen in de Slackware kern scripts (opstarten, netwerk, installatie, ...)
 - Nieuw toe te voegen pakketten
 - Niet-publiek discussie-forum waar over de toekomstige koers wordt gedebatteerd

Kijkend naar de toekomst

Uitdagingen

- Met zelf-gekozen restricties (klein team, ontwerp filosofie) zal Slackware nooit het grote publiek aanspreken. Een echte “out of the box” ervaring vereist een groter team maar dat past sowieso niet bij de “minimalistische” aanpak
- Server versus desktop
Ideaal platform voor software ontwikkeling, dankzij de aanwezigheid van grotendeels ongewijzigde software, maar biedt standaard ook “server” diensten. Niettemin ook een prima desktop dankzij het werk dat door de KDE, XOrg and andere teams wordt verricht.
- Hardware platforms
Na het toevoegen van x86_64 aan de lijst van ondersteunde platforms, iets dat zorgde voor het herschrijven van zowat ieder SlackBuild script, is het nu de beurt aan de ARM en S390 ports om opnieuw gesynchroniseerd te worden met de basis.
Denk aan de komende ARM gebaseerde netbooks.

Core Team

Patrick Volkerding — volkerdi@
Eric Hameleers — alien@
Piter PUNK — piterpunk@
Robby Workman — rworkman@
Stuart Winter — mozes@
Mark Post — markkp@
Vincent Batts — vbatts@

Fred Emmott — fred@
Alan Hicks — alan@
John Jenkins — mrgoblin@
Amritpal Bath — amrit@
Erik Jan Tromp — alphageek@
Karl Magnus Kolstø — karlmag@
Leopold Midha — netrixtardis@

En meer...

Een aantal leden van het team kiezen er voor om anoniem te blijven.

De Community van gebruikers

Natuurlijk wordt **VEEL** van het harde werk, vooral het testen en melden (en oplossen!) van bugs, door de gebruikers community gedaan — met dank!

Slackware Package Management

Slackware Package Management

- Slackware heeft **echt wel** een package manager.
 - `pkgtool(8)`, `installpkg(8)`, `upgradepkg(8)`, en `removepkg(8)` doen een verdienstelijke job in het beheren van **pakketten**.

Slackware Package Management

- Slackware heeft **echt wel** een package manager.
 - `pkgtool(8)`, `installpkg(8)`, `upgradepkg(8)`, en `removepkg(8)` doen een verdienstelijke job in het beheren van **pakketten**.
- Slackware's eigen package management gereedschap probeert echter **NIET** om de **afhankelijkheden** van pakketten te beheren.
 - Pakketten worden niet gesplitst in onderdelen als *app-bin*, *app-lib*, *app-devel*, *app-doc*, enzovoort — ieder bestand dat door een `make install` wordt neergezet belandt in het Slackware pakket.
 - Disk ruimte is tegenwoordig geen probleem meer, dus een volledige installatie geniet de voorkeur.
 - In een volledige installatie worden alle afhankelijkheden van Slackware's eigen pakketten vervuld.

Slackware Package Management

Stel, je wilt informatie over pakketten die op je systeem staan.

Gebruik `ls(1)` voor een lijst van pakketten

```
# ls /var/log/packages
a2ps-4.14-i486-4
aaa_base-13.0-noarch-2
aaa_elflibs-12.34-i486-1
...
```

Gebruik `cat(1)` om de inhoud van een pakket te zien

```
# cat /var/log/packages/a2ps-4.14-i486-4
PACKAGE NAME: a2ps-4.14-i486-4
COMPRESSED PACKAGE SIZE: 748 K
UNCOMPRESSED PACKAGE SIZE: 4650 K
...
usr/share/psutils/md68_0.ps
usr/share/psutils/md71_0.ps
```

Hoe werkt Slackware's pakket-beheer?

- Slackware pakketten zijn in feite gecomprimeerde *tar* archieven. Ze worden in de `/` directory uitgepakt tenzij een alternatieve root wordt gespecificeerd
- Als de omgevingsvariabele `“ROOT”` gezet is, zullen de pakketbeheer programma's diens waarde als de echte `/` opvatten.
- Misschien heb je gezien dat alle bestanden in het voorbeeld pakket met een relatief pad werden getoond — dat zorgt ervoor dat ze op de juiste plaats belanden bij het uitpakken van het pakket.
- Na het uitpakken van de bestanden in het pakket wordt het `doinst.sh` (post-install) script uitgevoerd vanuit de `$ROOT` directory (meestal is dat `/`).

Voorbeeld van de inhoud van een pakket

FILELIST

```
./
etc/
etc/hejaz.conf.new
usr/
usr/bin/
usr/bin/makehejaz
usr/doc/
usr/doc/makehejaz-1.0/
usr/doc/makehejaz-1.0/README
usr/doc/makehejaz-1.0/COPYING
usr/include/
usr/include/hejaz.h
usr/lib
usr/lib/libhejaz.so.1.0
usr/man
usr/man/man1/
usr/man/man1/makehejaz.1.gz
install/
install/doinst.sh
install/slack-desc
```

makehejaz doinst.sh inhoud

```
# cat install/doinst.sh
```

```
config() {
    NEW="$1"
    OLD="$(dirname $NEW)/$(basename $NEW .new)"
    # If there's no config file by that name, move it over:
    if [ ! -r $OLD ]; then
        mv $NEW $OLD
    elif [ "$(cat $OLD | md5sum)" = "$(cat $NEW | md5sum)" ]; then
        # toss the redundant copy
        rm $NEW
    fi
    # Otherwise, we leave the new copy for the admin to consider...
}

config etc/hejaz.conf.new

( cd usr/lib ; rm -rf libhejaz.so.1 )
( cd usr/lib ; ln -s libhejaz.so.1.0 libhejaz.so.1 )
( cd usr/lib ; rm -rf libhejaz.so )
( cd usr/lib ; ln -s libhejaz.so.1 libhejaz.so )
```

Omgaan met configuratiebestanden in het pakket

Let op dat het configuratiebestand “etc/hejaz.conf” met een “.new” extensie werd ge-installeerd.

- Dit is het mechanisme om bij het opwaarderen van pakketten het per ongeluk overschrijven van bestaande configuratiebestanden te vermijden.
- In het voorbeeld pakket is het de `config()` functie in het `doinst.sh` script die controleert of het bestand *hejaz.conf* al bestaat als `$/ROOT/etc/hejaz.conf`.
 - Als dat niet bestaat, wordt het *hejaz.conf.new* bestand hernoemd tot *hejaz.conf*.
 - Als het al zou bestaan, dan wordt de md5sum van het oude en nieuwe bestand vergeleken.
 - Als de md5sums gelijk zijn, dan zijn de twee bestanden identiek, en wordt het nieuwe verwijderd.
 - Als de md5sums verschillen, wordt het nieuwe bestand (met de *.new* extensie) behouden zodat de systeembeheerder het kan controleren.

Omgaan met symbolische links in een pakket

Misschien heb je gemerkt dat het voorbeeldpakket een enkel “library” bestand bevatte — `libhejaz.so.1.0` — maar geen bestanden of symlinks genaamd `libhejaz.so.1` of `libhejaz.so` die er naar verwijzen.

Dat komt doordat Slackware’s `makepkg(8)` programma de symlinks verwijdert tijdens het maken van het pakket — waarna deze worden opgenomen in het `doinst.sh` script zodat ze tijdens de installatie van het pakket opnieuw kunnen worden aangemaakt.

Additionele Software installeren in Slackware

Slackware's officiële pakket omvang is voldoende voor veel gebruikers. Er bestaat veel andere software die evengoed de moeite waard is of onmisbaar voor het dagelijks werk.

Het traditionele “configure && make && make install” werkt heel goed in Slackware, maar het levert potentieel problemen op...

- Mogelijke conflicten met Slackware pakketten
- Moeilijk om bij te houden wat ge-installeerd is
- Mogelijke problemen bij opwaarderen of verwijderen
- Ervaren gebruikers zullen hier geen moeite mee hebben, maar anderen worden uiteindelijk vaak met een onstabiel systeem opgezadeld.

Er bestaat een klein aantal bekende community pakket repositories...

- Veel gebruikers willen niet eens pakketten installeren van derde partijen
- Diegenen die geen moeite hebben met pakketten van derden hechten waarde aan de **betrouwbaarheid** van deze bronnen.
- Betrouwbaarheid wordt door de community bepaald.

Installeren van Additionele Software

Betrouwbare bronnen van pakketten — waar zijn die, en hoe weet je of ze betrouwbaar zijn?

- Is de “packager” iemand met een goede reputatie in de community?
- Controleer de inhoud van het pakket
 - Worden de correcte bestandslocaties gebruikt?
 - Wordt rekening gehouden met configuratiebestanden?
 - Worden manual pagina's gecomprimeerd?
 - Bevat het pakket de documentatie (indien van toepassing)?
 - Wordt het pakket gecompileerd op een “schone” Slackware installatie?
 - Zijn alle pakket-afhankelijkheden gedocumenteerd?
- Biedt de packager de volledige broncode aan en documenteert hij hoe het pakket werd gebouwd (zoals een “build script”)?

Open Source Software licenties vereisen vaak dat de broncode wordt aangeboden samen met de binaries!

Naar onze mening moeten packagers worden vermeden die hun compilatiescripts en broncode niet beschikbaar maken

Repositories van derden

- Diverse Slackware teamleden bieden pakketten (met broncode) aan:
 - Eric Hameleers (alienBOB) — <http://slackware.com/~alien/>
 - Robby Workman (rworkman) — <http://rlworkman.net/pkgs/>
 - Erik Jan Tromp (alphageek) — <http://alphageek.dyndns.org/>
 - Piter PUNK — <http://piterpunk.unitednerds.org/>

Dit is natuurlijk een onvolledige lijst!

<http://slackfind.net/en/> is een repository zoekmachine

De kwaliteit van pakketten die door derden worden aangeboden kan variëren — sommige staan bekend vanwege het niet aanbieden van broncode en/of compilatie instructies, of compileren hun pakketten op systemen zonder een schone Slackware installatie (wat ongedocumenteerde afhankelijkheden oplevert).

Daarom komt het steeds meer voor dat men zijn eigen pakketten bouwt met hulp van kwaliteit-gecontroleerde SlackBuild scripts die van <http://slackbuilds.org> gehaald zijn.

SlackBuild Scripts Gebruiken

Uiteindelijk komt iedereen uit bij de situatie dat een applicatie nodig is, waarvoor:

- nog geen pakket bestaat, of
- waarvoor je niet een van de bestaande pakketten wilt installeren

Een betrouwbaar SlackBuild script biedt dan uitkomst.

Een SlackBuild script is niet meer dan een klein shell script, dat:

- het broncode archief uitpakt,
- de voorbereidingen treft voor het compileren van de broncode
- de object-code compileert,
- deze in een tijdelijke “staging” directory installeert,
- verdere acties uitvoert die bij een goed pakket horen (toevoegen van documentatie en pakketbeschrijving, comprimeren van manual pagina's enzovoort)
- `makepkg(8)` aanroept om een Slackware pakket van de applicatie te maken

Tot slot

Tot zover deze tour langs Slackware Linux!

Wie interesse heeft in Slackware Linux moet eens langskomen in het `##slackware` Freenode IRC netwerk. Een actieve en behulpzame community is aanwezig in het LinuxQuestions forum:

<http://www.linuxquestions.org/questions/slackware-14/>

Andere bronnen

- Interview met Patrick Volkerding in LinuxJournal (1994)
<http://www.linuxjournal.com/article/2750>
- Audio interview met Patrick door Linux Today (1999)
<http://www.linuxtoday.com/developer/1999120500405NWSM>
- Telefoon interview met Patrick (The Linux Link Tech Show 2006)
<http://tllts.org/dl.php?episode=164>
- Compleet overzicht van Slackware releases
http://www.nielshorn.net/slackware/slack_versions.php
- Onofficiële geschiedenis van Slackware releases
<http://www.slackdown.co.uk/history.html>

Met dank aan Robby Workman (met wie ik delen van deze lezing heb gepresenteerd op de Slackware Show 2008), en Patrick Volkerding en het team voor hun feedback.

Over de auteur

Ik ben een Linux gebruiker sinds de 0.99 kernel release en kwam in aanraking met Slackware Linux rond 1995. De eerste keer dat het noodzakelijk bleek om iets aan te passen aan Slackware was toen ik bij IBM ging werken en opgezadeld werd met een Token Ring netwerk waarvoor geen support bestond in de netwerk scripts :-)

Mijn betrokkenheid met Slackware is in de loop der jaren gegroeid - ik ben nu een van de Slackware core-team leden. Ik beheer ook een repository van onofficiële Slackware pakketten en scripts, en ben een van de admins van de SlackBuilds.org website (een community site voor op kwaliteit geteste Slackware package scripts).

Feedback en suggesties zijn welkom!

alien@slackware.com

Een kopie van deze presentatie zal te vinden zijn op:

http://www.slackware.com/~alien/nllgg_20100403/

Vragen



Slackware

Nog vragen ?