

Software installeren vanaf source

Johan Swenker, Linux platform CompUsers, 27 september 2003

Tijdens de CompUsers gebruikersdag op 27 september 2003 heeft Johan Swenker een workshop geleid waarin hij liet zien hoe programma's waarvan alleen de broncode beschikbaar is moeten worden geïnstalleerd. Als voorbeeld gebruikte hij drie programma's. Deze kunt u na enig zoekwerk op het internet weliswaar zo downloaden, maar om het u gemakkelijk te maken heeft Johan een pakket met de drie programma's samengesteld. U vindt het hier onder de naam [september2003.tar](#). Het is bijna 3 MByte groot.

Hieronder staat wat Johan van plan was de deelnemers te laten doen. Hij zegt er zelf van:

Ik heb de bedoeling gehad om iets voor gevorderden te doen (gevorderden zoals in beginners, gevorderden, brons, zilver en goud bij dansles). Ik merkte dat er beginners bij waren. In de auto terug naar huis hoorde ik waarom: men dacht dat het om het installeren van Linux zou gaan en niet om het installeren van programma's. De mensen die niet goed voorgelicht waren, hebben een stoomcursus Linux op de commandline gekregen. Ik hoop dat het toch interessant geweest is.

Vooraf

gcc, X, Tcl en Tk moeten geïnstalleerd zijn. Van X, Tcl en Tk moeten ook de development rpm's geïnstalleerd zijn.

Inleiding

In deze workshop gaan we 3 programma's installeren. De programma's en de installatie is van opklimmende graad van complexiteit. Misschien dat niet eens iedereen aan het derde programma kan beginnen.

1) Hello world: world.c

vertalen met: `make world`

uitvoeren met: `./world`

nog een keer vertalen: `make world`

aanpassen met: `touch world.c`

nog een keer vertalen: `make world`

leermomenten:

- include van stdio.h, uit de systeem omgeving
- een klein beetje C
- make, compileert alleen wat veranderd is
- het make commando zonder make-file
- uitvoeren terwijl de directory niet in PATH staat
- afbreken met ^C

informatie:

Dit programma is op internet te vinden door te zoeken met google naar "hello world". De eerste hit is de Hello, World Page!

2) xsnow: xsnow-1.42.tar.gz

Controleer of de tar-file zelf een directory aanmaakt (gaat bijna nooit mis, maar als het mis gaat heb je een puinhoop):

1. LESSOPEN is gedefinieerd en het commando herkent .tar.gz files

```
less xsnow-1.42.tar.gz
```

2. met less wil het niet, dan maar met de hand:

```
tar tvzf xsnow-1.42.tar.gz
```

Uitpakken van de file:

```
mkdir klad
```

```
cd klad
```

```
tar xvzf ../xsnow-1.42.tar.gz
```

het commando tar:

Scan door de manual pagina van tar met:

```
man tar
```

Zoek naar de optie -t en de optie -x met: /t, gevolgd door een keer de letter n om het zoek commando te herhalen.

Blader even terug naar 'FUNCTION LETTERS'. Tar heeft blijkbaar een aantal functies zoals Test en eXtract

Zoek naar de opties -v, -z en -f. Geef in eigen bewoordingen aan wat deze modifiers doen. Wat stelt de default waarde voor -f voor?

Zoek naar de optie -y of -j. Een van beide lijkt op de optie -z. Welke? Waarom? Wanneer gebruik je die?

Merk op dat een opdracht zoals

```
gunzip < filenaam.tar.gz | tar xf -
```

voor Linux (met de Gnu-variant van tar) niet nodig is.

Ga naar de subdirectory xsnow-1.42:

```
cd <tab>
```

(Ik misbruik hier file name completion van bash.)

Bekijk eens welke files hier staan. Files met namen zoals README, INSTALL zijn meestal de moeite van het lezen waard.

Als je niet zeker weet waar het programma zelf voor dient, bekijk dan een directory zoals doc of html en lees files die eindigen op .1 t/m .8 Ik lees ze altijd met less; mijn lesspipe.sh herkent deze files. Lees ze anders met het man-commando:

```
man ./xsnow.man
```

(In dit geval is de .1 de geformatteerde tekst en de .man de groff source file)

Probeer maar te compileren en uit te voeren.

het commando make:

Herhaal het make proces:

```
make
```

Doet niets. Toch wil ik dat alles opnieuw gebouwd wordt.

Wis alles wat gebouwd is:

```
make clean
```

Herbouw:

```
make
```

Kun je ook echt alles wissen (zonder de file opnieuw uit te pakken)?

Ja: `make distclean`

(history: toen Mr. Proper een bekend schoonmaakmiddel was, werkte `make mrproper` voor dit doel.)

Bouw alles opnieuw: `xkcmf && make`

Probeer te installeren: `make install`

Dit hoort niet te werken. Je hoort niet als root aangemeld te zijn. Dus je mag niet in de systeem directories schrijven. Mijn advies:

Voorkeur:

configureer en gebruik `sudo`: `sudo make install`

Tweede keuze:

open een tweede window waar je wel als root bent aangemeld.

het commando gcc:

Lees de uitvoer van het `make` commando:

- aanroep van `gcc`, met als parameters o.a.

```
... -I/usr/X11R6/include ... -c -o xsnow.o xsnow.c
```

Op magische wijze weet `gcc` waar de include files staan. Sommige komen uit `/usr/include`, zoals `<stdio.h>` van `world.c`. Sommige komen uit de current directory, zoals `xsnow.h`.

Sommige komen uit een systeem directory die niet-standaard is. Dan moet je die met `-I` opgeven. Normaal probeert `gcc` een complete executable op te leveren. Nu moeten twee C-files gecompileerd en gelinkt worden. Daarom de optie `-c`, d.w.z. doe het linken (nog) niet. Daarom ook de optie `-o xsnow.o`, d.w.z. zet de uitvoer in de file `xsnow.o`.

- nog zo'n zelfde aanroep van `gcc`, nu voor `toon_root.c`
- aanroep van `gcc`, met als parameters o.a.

```
... -L/usr/X11R6/lib xsnow.o toon_root.o -lXext ...
```

Op magische wijze weet de loader van `gcc` waar de biblioeken (libraries) staan. Soms moet je een handje helpen met de optie `-L<directory>`. Je moet uiteraard aangeven welke objects samengevoegd moeten worden. Je moet blijkbaar (`-lXext`) ook nog losse bibliotheken opgeven. Zoek eens in `/usr/X11R6/lib` naar `Xext`:

```
ls -l /usr/X11R6/lib/*Xext*
```

Blijkbaar heet de file `libXext.*`. Waarbij ik kan verklappen dat `libXext.so.6` gebruikt wordt.

leermomenten:

- tarmet de opties `-j`, `-y`, `-z`
- zelf een subdirectory maken
- README's zijn er om te lezen
- manual pagina's
- `xkcmf` (X-windows MaKe makefile)
- enige standaard `make` aanroepen
- rootrechten voor `make install`
- `gcc` met include files en libraries

informatie:

Ik heb `xsnow` gekozen uit nostalgische overwegingen. Ik herinner `mexsnow` als eerste programma wat ik zelf vanaf source installeerde. Het world wide web bestond nog niet. Ik kreeg de source uit een nieuwsgroep zoals `alt.sources.x` (een nieuwsgroep die inmiddels al jaren leeg is).

Het was toendertijd geen tar-file maar een shar-file. Een shar-file is een SHell-ARchief, een bestand wat je onder andere kon uitpakken door het gewoon door de shell te laten uitvoeren.

Linux was toen ook lang zo populair niet als nu. Ik was dol-enthousiast dat iets wat ik van usenet plukte in een keer werkte, terwijl ik `xkcmf` nooit eerder gebruikt had.

3) ttt, een programma voor netwerk analyse, vergelijkbaar met tcpdump

Uitpakken: `mkdir work ; cd work ; tar xvzf ../ttt-1.7.tar.gz`

Lezen wat je moet doen: `cd ttt-1.7 ; less README`

Dit doen: `./configure ; make ; make install`

Oeps waarom werkt dit niet?

het script configure:

Bekijk de file configure met een gewone editor. Waaraan kun je zien dat het een shell script is?

Heeft configure een help optie? Zoek `help'. Op regel 148 staat het antwoord.

Zoek de letterlijke foutmelding van `./configure`. Op regel 2000 staat deze. Wat betekent `exit 1' in regel 2001? Haal die eens weg en kijk of je nu wel door `./configure` heen komt. Werkt de `make` nu?

Eerst BLT2.4u installeren.

Dus: `pushd ../..`

(`pushd` doet een `cd` maar onthoud waar we vandaan kwamen)

Ergens BLT2.4u.tar.gz vandaan toveren.

Uitpakken: `cd work; tar xvzf ../BLT2.4u.tar.gz`

Lezen wat je moet doen: `cd BLT2.4u ; less README`

Dit doen: `./configure && make`

We besluiten dus om `_niet_` te installeren.

Daarna (indien nodig) `libpcap-0.7.1` installeren.

Tenslotte verder gaan waar we gebleven waren:

`popd` (dit is het omgekeerde van de `pushd`)

`./configure --with-bl=../blt2.4u/src/` (deze parameter is nodig omdat we BLT niet geïnstalleerd hebben.)

`make && make install`

leermomenten:

- `configure`
- optie `--help`
- een van de vele `--with-` opties
- standaard riedel: `./configure && make && make install`
- het equivalent van de `dll-hell`: BLT was nodig
- handmatig aanpassen van files kan soms nodig zijn.
Ik heb wel C-files en gegenereerde Makefiles moeten aanpassen :(

informatie:

Ik vind netwerksoftware interessant, daardoor kwam ik deze tegen. Zelfs op mijn machine met heel veel geïnstalleerde software, kwam ik een afhankelijkheid tegen die ik wilde demonstreren.