

Het Linux Huiswerksysteem

Vanaf deze week gaan we het Linux Huiswerksysteem in gebruik nemen. Via het Linux huiswerksysteem kan ik jullie vorderingen beoordelen. Direct nadat je het huiswerk hebt gemaakt stuur je mij een e-mail met de output van het *history* commando. Om dit mogelijk te maken gaan we het *email* programma *compileren* en installeren.

Over de GNU Compiler Collection

De Linux kernel is helemaal opgebouwd uit *Free Software*. Dit houdt niet alleen in dat de software gratis is, maar als je wilt kun je zelf veranderingen aanbrengen in de software. Dat kan omdat de *broncode* van de software vrij beschikbaar is. Broncode is de leesbare versie van een computerprogramma. Nadat je wijzigingen hebt aangebracht in de broncode, kun je deze opnieuw *compileren* tot een programma dat kan draaien op je computer. Het compileren van broncode doe je met de *GNU Compiler Collection*.

Richard Stallman is de programmeur achter de GNU Software en de oprichter van de *Free Software Foundation*. Hij heeft een licentie bedacht, genaamd *Copyleft* om de rechten van de gebruiker van de software te beschermen.



Het huiswerksysteem is een goed voorbeeld van het wijzigen van broncode. Om ervoor te zorgen dat jij je huiswerk kunt insturen, zodat ik het kan nakijken, heb ik de broncode van het *email* programma een klein beetje aangepast. Het enige dat je zelf moet doen, is het programma opnieuw compileren, zodat het in jouw Linux versie kan worden geïnstalleerd.

GCC installeren

Zoals in de les behandeld is Linux een *operating system* dat gebaseerd is op vrij beschikbare broncode (*Open Source*). Om broncode te compileren tot uitvoerbare programma's heb je de GNU Compiler Collection (*gcc*) nodig. In de meeste Linux distributies wordt *gcc* niet of alleen gedeeltelijk geïnstalleerd.

Cygwin gebruikers

In Cygwin wordt *gcc* niet standaard geïnstalleerd. We gaan dit alsnog doen met het programma *apt-get*. Update eerst de package database:

```
apt-get update
```

Wacht rustig tot de installatie gereed is:

```
apt-get install make gcc-g++
```

Voor het geautomatiseerd compileren van grote software-pakketten hebben we het *make* programma nodig. Het package *gcc-g++* bevat de *GNU Compiler Collection* (*gcc*) en de bijbehorende bestanden voor object-georiënteerd programmeren (*g++*).

Andere Linux gebruikers

In bijvoorbeeld *Ubuntu* wordt *gcc* alleen gedeeltelijk geïnstalleerd. Update eerst de package database:

```
apt-get update
```

We hebben nog aanvullende broncode nodig. Deze installeren we met het *apt-get* programma:

```
apt-get install build-essential
```

Apt-get gaat nu de aanvullende broncode van *gcc* voor je neerladen en installeren. Wacht tot dit gereed is.

Map voor broncode aanmaken

De juiste locatie voor broncode in Linux is `/usr/local/src`. Kijk eerst of deze directory al bestaat. Dit doen we met het commando `ls`:

```
ls -l /usr/local/
```

Je zou ongeveer het volgende moeten kunnen zien:

```
drwxr-xr-x+ 1 daniel daniel 0 Dec 7 21:30 bin
drwxr-xr-x+ 1 daniel daniel 0 Dec 7 21:30 etc
drwxr-xr-x+ 1 daniel daniel 0 Dec 7 21:30 lib
drwxr-xr-x+ 1 daniel daniel 0 Dec 7 21:30 src
```

Dit is de output van Cygwin. In de overige Linux distributies is de lijst langer en hebben de bestanden andere toegangsrechten. Indien de directory `src` niet bestaat, maak hem dan aan met het commando `mkdir` (MaKe DIRectory):

```
mkdir /usr/local/src
```

Broncode neerladen en uitpakken

We gaan eerst de broncode voor het *email* programma neerladen. Ga in de directory voor broncode staan. Dit doen we met het commando `cd` (change directory):

```
cd /usr/local/src
```

Als je zelf programma's wilt installeren gebruik je het installatie-systeem van de betreffende distributie of je doet een zogenaamde *local install*. Het handige van een locale installatie is dat er niets aan de originele distributie wordt gewijzigd. Alles gebeurt "locaal" en je kunt altijd terugvallen op de originele standaard distributie. Locale installaties worden altijd geplaatst in de map `/usr/local`.

Nu kunnen we de broncode voor het *email* programma neerladen. Dit doen we met het `wget` (www get document) programma:

```
wget -N http://www.commandlinerevolution.nl/Source/eMail-3.2.3-git_CR.tar.bz2
```

Het programma `wget` maakt een kopie van het neergeladen bestand met een volgnummer als het bestand al een keer is neergeladen. Dit is niet handig. Via de optie `N` (newer) geven we aan dat `wget` het eventueel bestaande bestand vervangt als het nieuwer is.

De broncode is ingepakt in het *tar* (tape archive) formaat. Pak het bestand uit middels het *tar* commando:

```
tar -xvf eMail-3.2.3-git_CR.tar.bz2
```

Optie `x` (extract) betekent dat we het bestand willen uitpakken. Middels optie `v` (verbose) geven we aan dat we het uitpak-proces willen zien. Met optie `f` geven we aan dat het om een bestand gaat.

Je ziet nu dat het *tar* programma de bestanden uitpakt naar de map *eMail*. Wacht tot dit klaar is.

Broncode compileren

Nu kunnen we beginnen met het compileren van de broncode. Ga eerst naar de nieuwe map *eMail* met het commando `cd`:

```
cd eMail
```

Compileren gaat in drie stappen: configureren, compileren en installeren.

Starten *configure* script

Omdat alle Linux-software is geschreven als platform-onafhankelijke software, moeten er extra tests op je systeem worden uitgevoerd. Het *configure* script is een zogenaamd *shell script* waarmee deze

tests worden uitgevoerd. Dit script wordt in alle software-distributies meegeleverd. Het *configure* script kan als volgt worden gestart:

```
./configure --prefix=/usr --sysconfdir=/etc
```

Via de optie *-prefix* (voorvoegsel) geven we aan dat het programma in de */usr* directory moet worden geplaatst. Met de optie *-sysconfdir* geven we aan dat de configuratiebestanden in de map */etc* moet worden geplaatst.

Je ziet nu dat er een hele serie van controles en tests op je systeem wordt uitgevoerd. Wacht rustig af tot dit gereed is:

```
checking for strerror... yes
checking for strrchr... yes
checking for uname... yes
configure: creating ./config.status
config.status: creating Makefile
config.status: creating include/config.h
```

Compileren broncode

De tweede stap is het daadwerkelijk compileren van de broncode. Dit doen we met het *make* commando:

```
make
```

Je ziet nu opnieuw allerlei codes voorbij komen. De output toont hoofdzakelijk het uitvoeren van het *gcc* commando (de compiler), gevolgd door een heleboel opties. Wacht af tot dit gereed is:

```
gcc -g -O2 -Wall -W -DUSE_GNU_STRFTIME -I/usr/local/src/email-3.1.3 -
I/usr/local/src/email-3.1.3/src -I/usr/local/src/email-3.1.3/include -I../include -
I/usr/local/src/email-3.1.3/dlib/include -I../dlib/include -DEMAIL_VERSION='"3.1.3"'
-D_COMPILE_DATE='"10/31/2010-01:13:35PM_EDT"' -DEMAIL_DIR='"usr/etc/email"' -
DHAVE_CONFIG_H -o email email.o addr_parse.o addy_book.o conf.o error.o execgpg.o
file_io.o message.o mimeutils.o processmail.o progress_bar.o remotessmtp.o sig_file.o
smtpcommands.o utils.o ../dlib/libdlib.a
make[1]: Map '/usr/local/src/email-3.1.3/src' wordt verlaten
```

Installeren binaries

We hebben nu een zogenaamde *binary* (uitvoerbaar programma) gemaakt. De derde en laatste stap is het installeren van de nieuwe bestanden. De binary, inclusief help- en configuratiebestand moeten nu in de juiste locaties geïnstalleerd worden:

Type bestanden	locatie
binaries (programma-bestanden)	/usr/bin
man pages (help-bestanden)	/usr/share/man
conf files (instellingen)	/etc/email

Dit kunnen we ook weer automatisch doen met het *make* commando:

```
make install
```

Via de optie *install* geven we aan dat het zogenaamde *install target* moet worden uitgevoerd.

SMTP Poort 25 testen

E-mail wordt verstuurd via een *SMTP* (Simple Mail Transfer Protocol) server. Normaal luistert de e-mail software op poort 25 naar binnenkomende connecties. Zowel ROC Leiden als ook de meeste providers (KPN, Ziggo) hebben deze poort geblokkeerd omdat er teveel spam overheen wordt gestuurd.

Om ervoor te zorgen dat het huiswerk-systeem werkt, moeten we dus uitwijken naar een andere poort. Het meest geschikt is dan poort 21, waar normaal *FTP* verkeer over gaat. Dit werkt alleen als de SMTP server aan de andere kant ook werkelijk luistert op poort 21.

Controleer of de server die we gaan gebruiken correct is ingesteld op poort 443. Hiervoor gebruiken we het *telnet* programma:

```
telnet boland.xs4all.nl 21
```

Let op: normaal luistert een mail-server op poort 25, maar omdat deze poort op school geblokkeerd wordt wijken we uit naar poort 21. Als het goed is krijgen we het volgende antwoord:

```
Trying 83.162.146.149...
Connected to boland.xs4all.nl.
Escape character is '^]'.
220 boland.xs4all.nl ESMTP Sendmail 8.14.0/8.14.0; Fri, 28 Jun 2019 06:05:45 +0200 (CEST)
```

We zien dat de SMTP software *Sendmail* heet en dat de server ook ESMTP (Extended SMTP) “spreekt”. De server wacht nu op een commando. Geef het commando *quit* om de sessie met de mailserver te beëindigen:

```
quit
```

We zijn nu weer terug op de Linux command line.

E-Mail instellingen

Voordat we het *email* programma kunnen gebruiken moeten we nog een aantal gegevens invullen, zoals ons afzender e-mail adres en welke server we willen gebruiken. Dit doen we door het *email* configuratiebestand te wijzigen met de *vi* text-editor:

```
vi /etc/email/email.conf
```

Je ziet nu de inhoud van het configuratiebestand. Toets **i** om naar de *INSERT* modus te gaan en breng de volgende vetgedrukte wijzigingen aan.

```
#####
# SMTP Server and Port number you use
#####
SMTP_SERVER = 'boland.xs4all.nl'
SMTP_PORT = '21'

#####
# Your email address: If you'd like To have your name to
# show in the from field instead of just your email address,
# then keep the format below and edit it to your email
# and name.
#####
MY_NAME = 'Voornaam Achternaam'
MY_EMAIL = 'xxxxx@mborijnland.nl'
#REPLY_TO = ''
```

Let op: het student-mail-adres (*MY_EMAIL*) moet je altijd invullen, anders wordt jouw e-mail als *spam* gezien. Als je mijn e-mails aan jou ergens anders wilt ontvangen, gebruik dan het veld *REPLY_TO*. Als je dit invult, verwijder dan ook de *hash-tag*. Hiermee wordt de instelling geactiveerd.

Toets **<Esc>** om uit de *INSERT* modus te komen en sla het bestand op met de commando's *w* (write) en *q* (quit), gevolgd door **<Enter>**:

```
:wq
```

Het bestand is nu gewijzigd en we zijn terug op de Linux command-line.

Je eerste huiswerk e-mail

Verstuur nu je allereerste huiswerk-email. Dit doen we met een pijpleiding tussen de commando's *history* en *email*. Zorg dat het onderwerp van de e-mail (optie *s*) tussen aanhalingstekens staat:

```
history | email -s "Mijn eerste huiswerk e-mail" docent@localhost
```

Als er geen foutmelding verschijnt, is de e-mail correct verstuurd.

Administrator uitloggen

We hebben de broncode voor het *email* programma gecompileerd, maar we zijn nog steeds ingelogd als Administrator (*root*).

Geef tot slot **<Ctrl>-D** om de Administrator uit te loggen.