Vim Color Editor HOW-TO (Vi Improved met syntax color highlighting)

Al Dev (Alavoor Vasudevan) *alavoor@yahoo.com*
Vertaald door: Ellen Bokhorst *bokkie@nl.linux.org*

v5.0, 04 maart 2000

Dit document is een leidraad voor het snel instellen van de Vim-kleuren editor onder Linux- of Unix-systemen. De informatie hierin zal de productiviteit van een programmeur verhogen aangezien de Vim-editor ondersteuning biedt voor syntax color highlighting en bold fonts die de "leesbaarheid" van programmeercode verbetert. De productiviteit van programmeurs wordt met 2 tot 3 keer verbeterd door gebruik te maken van een kleureneditor zoals Vim. De informatie in dit document geldt voor alle besturingssystemen waaronder Vim werkt, dat zijn - Windows 95/NT en alle soorten Unix zoals Linux, Solaris, HPUX, AIX, SCO, Sinix, BSD, SCO, enz..

Inhoudsopgave

1 Introductie

Vim editor staat voor 'Vi IMproved'. Vi is de populairste en krachtigste editor in de Unix-wereld. De **Vi** is een afkorting voor **Vi@"** ***Visueelëditor. Vroeger was de eerste editor op deze planeet een regeleditor genaamd 'ed' (en 'ex'). De Visuele editor zoals Vi was een reusachtige verbetering in vergelijking met regeleditors als 'ed' (of 'ex'). De editors 'ed' en 'ex' zijn nog steeds beschikbaar onder Linux, zie 'man ed' en 'man ex'.***

Een goede editor zal de productiviteit van de programmeur verbeteren. Vim biedt ondersteuning voor color syntax highlighting van programma-code en tevens benadrukking door gebruik te maken van verschillende fonts als normaal, vet of cursief. Een kleureneditor als Vim zal de productiviteit bij het programmeren met een factor 2 tot 3 verhogen!! Programmeurs kunnen de code veel sneller lezen aangezien de syntax van de code in kleur en opgelicht wordt weergegeven.

1.1 Installatie van Vim onder Redhat Linux

Installeer de volgende RPM-packages om Vim onder Redhat Linux te installeren -

```
        rpm -i vim*.rpm

OF gebruik dit -
        rpm -i vim-enhanced*.rpm
        rpm -i vim-X11*.rpm
        rpm -i vim-common*.rpm
        rpm -i vim-minimal*.rpm
```

Je kan een lijst te zien krijgen van de bestanden die het rpm-package van vim installeert met -

```
        rpm -qa | grep ^vim | awk '{print "rpm -ql " $1 }' | /bin/sh | less
```

en hier doorheen bladeren met j,k, CTRL+f, CTRL+D, CTRL+B, CTRL+U of door gebruik te maken van de pijltjestoetsen, page up/down toetsen. Zie 'man less'.

1.2 Installatie van Vim onder GNU Debian Linux

Voor de installatie van Vim onder Debian Linux (GNU Linux) log je in als root en wanneer je bent verbonden met het internet type je:

```
apt-get install vim vim-rt
```

Hiermee zal de laatste versie van vim worden gedownload, geïnstalleerd en geconfigureerd en zal het 't .deb bestand dat werd gedowload, worden verwijderd. Het eerst opgesomde package is vim, de standaard editor, gecompileerd met X11-ondersteuning, vim-rt is de vim runtime, het bevat alle syntax- en helpbestanden.

1.3 Vim onder diverse Unix-en installeren

Download voor andere varianten van unix, zoals Solaris, HPUX, AIX, Sinix, SCO het broncode bestand

```
zcat vim.tar.gz | tar -xvf -
cd vim-5.5/src
./configure --enable-gui=motif
make
make install
```

1.4 Vim installeren onder Microsoft Windows 95/NT

Voor Windows 95/NT moet je TWEE zip-bestanden downloaden.

- ***Runtime ondersteuningsbestand vim*rt.zip***
- ***Vim opdrachtenbestand vim*56.zip. Hier is de Vim-versie 5.6***

Haald deze twee zipbestanden op vanaf -

- ***De homepage van vim:*** `<http://www.vim.org>`
- ***Een mirror site in US:*** `<http://www.us.vim.org>`

Pak de zip-bestanden uit met behulp van Winzip `<http://www.winzip.com>`***. Beide zip-bestanden (vim*rt.zip en vim*56.zip) moeten in dezelfde directory, bv in c:\vim worden uitgepakt.***

Stel de omgeving VIM in in autoexec.bat en plaats daarin deze regel -

```
set VIM=c:\vim\vim56
```

Vim zou daarnaar moeten verwijzen daar waar je de vim56 directory installeerde. Je kunt je PATH ook zo instellen dat het pad naar gvim.exe erin voorkomt.

Mogelijk moet je uitloggen en weer inloggen om je omgeving in te stellen. Typ achter de MS-DOS prompt -

```
set vim
```

En je zou moeten zien: VIM=c:\vim\vim56

Maakt een snelkoppeling aan op je desktop door te klikken en slepen vanuit c:\vim\vim56\gvim.exe. Kopieer het bestand gvimrc_example naar $VIM_gvimrc. In mijn geval is dit c:\vim\vim56_gvimrc.

2 Stel de init-bestanden van gvim in

Je MOET het bestand gvimrc naar je homedirectory kopiëren om de syntax color highlighting te activeren. Hierdoor zal ook het "Syntax"Menu met de gvim-opdracht worden geplaatst. Je kunt op het Syntax-Menu klikken en de van toepassing zijnde talen selecteren, zoals C++, Perl, Java, SQL, ESQL enz..

```
cd $HOME
cp /usr/doc/vim-common-5.3/gvimrc_example  ~/.gvimrc
```

Commentaarregels beginnen in .gvimrc met dubbele aanhalingstekens ("). Je kunt gvim aanpassen door het wijzigen van het bestand $HOME/.gvimrc en daar de volgende regels in plaatsen -

```
"Deze regel is commentaar .... één die met dubbele aanhalingstekens begint
" Het beste is het bold-font, probeer ze allemaal en kies er één uit....
set guifont=9x15bold
"set guifont=8x13bold
"set guifont=7x14bold
"set guifont=7x13bold
"
"Beslist aan te raden de tabtoetsen op 4 in te stellen.
set tabstop=4
set shiftwidth=4
"Het tegengestelde is 'set wrapscan' bij het zoeken naar strings....
set nowrapscan
"
"Het tegengestelde is noignorecase
set ignorecase
```

Het word je ten zeerste aanbevolen de tabstop en shiftwidth op 4 in te stellen.

De tabstop is het aantal spaties dat door TAB zal worden geplaatst als je met gvim aan het werken bent. De shiftwidth is het aantal spaties waarmee de regels zullen worden verschoven bij gebruik van de ">>öf "<<öpdrachten van vi. Raadpleeg de tutorials van Vi 8 () voor meer details.

Zie de opdracht xlsfonts voor een lijst met beschikbare fonts onder Linux/Unix. Typ

```
        bash$ xlsfonts | less
        bash$ xlsfonts | grep -i bold | grep x
        bash$ man xlsfonts
```

2.1 Voorbeeld gvimrc bestand

Je kunt de instellingen, zoals kleuren, bold/normale fonts in het $HOME/.gvimrc bestand wijzigen. Het wordt ten zeerste aanbevolen dat je de achtergrondkleur op `lichtgeel` ***of*** `wit` ***met een*** `zwarte` ***voorgrond instelt. Ergonomen zeggen dat de beste achtergrondkleur*** `lichtgeel` ***of*** `wit` ***met een zwarte voorgrond is. Wijzig daarom als volgt de variabele 'guibg':***

```
highlight Normal guibg=lightyellow
```

Het voorbeeld gvimrc vanuit /usr/doc/vim-common-5.3/gvimrc_ example luidt als volgt (vertaalde versie):

```
" Vim
" Een voorbeeld gvimrc bestand.
" De opdrachten hierin worden uitgevoerd wanneer de GUI wordt opgestart.
"
" Kopieer het voor gebruik naar
"     voor Unix en OS/2:  ~/.gvimrc
"            voor Amiga:  s:.gvimrc
"  voor MS-DOS en Win32:  $VIM\_gvimrc

" Zorg dat externe opdrachten via een pipe in plaats van een pseudo-tty werken
"set noguipty

" stel het te gebruiken X11-font in. Zie onder unix/linux 'man xlsfonts'.
" set guifont=-misc-fixed-medium-r-normal--14-130-75-75-c-70-iso8859-1
set guifont=8x13bold
"set guifont=9x15bold
"set guifont=7x14bold
"set guifont=7x13bold
"
"Beslist aan te raden de tabtoetsen op 4 in te stellen.
set tabstop=4
set shiftwidth=4
"Het tegengestelde is 'set wrapscan' bij het zoeken naar strings....
set nowrapscan
"
"Het tegengestelde is noignorecase
set ignorecase

" Stel de opdrachtregel in op 2 regels
set ch=2

" shift-insert werkt net als in Xterm
map <S-Insert> <MiddleMouse>
map! <S-Insert> <MiddleMouse>

" Doe dit alleen voor Vim versie 5.0 en later.
if version >= 500

  " Ik vind 't prettig als strings in C commentaar wordt opgelicht
  let c_comment_strings=1

  " Schakel syntax highlighting in.
  syntax on

" Schakel zoekpatroon highlighting in.
```

```
set hlsearch

" Voor Win32 versie, "K" zoekt het sleutelwoord in een helpbestand
"if has("win32")
"  let winhelpfile='windows.hlp'
"  map K :execute "!start winhlp32 -k <cword> " . winhelpfile <CR>
"endif

" Verberg de muisaanwijzer tijdens het typen
set mousehide

" Stel fraaie kleuren in
" achtergrond voor normale tekst is lichtgrijs
" Tekst onder de laatste regel is donkergrijs
" Cursor is groen
" Constanten worden niet onderstreept maar hebben een iets lichtere achtergrond
  highlight Normal guibg=grey90
  highlight Cursor guibg=Green guifg=NONE
  highlight NonText guibg=grey80
  highlight Constant gui=NONE guibg=grey95
  highlight Special gui=NONE guibg=grey95

endif
```

Zie ook het voorbeelbestand vimrc in /usr/doc/vim-common-5.3/vimrc_example dat voor onder de console wordt gebruikt

2.2 Xdefaults parameters

Je kunt een aantal eigenschappen van Vim in het bestand Xdefaults plaatsen.

WAARSCHUWING!! WAARSCHUWING: Plaats hierin niet de Vim*geometry; het zal het menu van gvim opbreken, gebruik hier voor in de plaats Vim.geometry.

Wijzig het bestand $HOME/.Xdefaults en plaats hierin de volgende regels:

```
! GVim great Colors.
Vim*useSchemes:         all
Vim*sgiMode:            true
Vim*useEnhancedFSB:     true
Vim.foreground:         Black
!Vim.background:        lightyellow2
Vim*background:         white
! Gebruik GEEN Vim*geometry, dit zal de menu's opbreken, gebruik in plaats
! daarvan Vim.geometry. Een asteriks tussen Vim en geometry is niet toegestaan.
! Vim.geometry: widthxheight
Vim.geometry:           88x40
!Vim*font:              -misc-fixed-medium-r-normal--20-200-75-75-c-100-iso8859-15-*5
Vim*menuBackground: yellow
Vim*menuForeground: black
```

Dit heeft effect na het intikken van -

```
        xrdb -merge $HOME/.Xdefaults
        man xrdb
```

Je kunt ook het bestand ~/.gvimrc wijzigen om de achtergrondkleuren te veranderen.

```
        gvim $HOME/.gvimrc
De beste achtergrondkleur is lichtgeel of wit.
highlight Normal guibg=lightyellow
```

3 Kleuren Syntax init bestanden

In plaats van gebruik te maken van het "Syntax"menu kun je het syntax-bestand ook handmatig bewerken. Wijzig het bestand met gvim en achter de : (dubbele punt) opdracht geef je de 'so' opdracht. Bijvoorbeeld -

```
        gvim foo.pc
        :so $VIM/syntax/esqlc.vim
```

De syntax source-bestanden staan in /usr/share/vim/syntax/*.vim. Vim ondersteunt meer dan 120 verschillende syntax-bestanden voor verschillende talen zoals C++, PERL, VHDL, Javascript,... enzovoort!!

Ieder syntaxbestand ondersteunt een standaardnaam voor de bestandsextensie, het syntaxbestand voor Javascript bijvoorbeeld, ondersteunt de extensie *.js. Mocht het je gebeuren dat je een andere extensie gebruikt die een conflict veroorzaakt met een ander standaard syntaxbestand zoals bij het plaatsen van Javascript in een *.html bestand, dan MOET je de opdracht :so $VIM/syntax/javascript.vim gebruiken. De beste methode is een softlink aan te maken zoals

```
ln -s $VIM/syntax/javascript.vim js
gvim foo.html (... dit bestand bevat javascript functies en HTML)
:so js
```

4 VIM Gebruik

Je kunt Vim in twee modi gebruiken - één met GUI en de andere zonder GUI. Om de GUI te gebruiken geef je de volgende opdracht -

```
        gvim foo.cpp
```

Om de niet-gui modus te gebruiken tik je in -

```
        vim foo.cpp
OF plain vanilla modus
        vi foo.cpp
```

Het is ten zeerste aan te raden dat je altijd gebruik maakt van gvim in plaats van vim, aangezien de GUI mode met kleuren je productiviteit beslist zal verbeteren.

De GUI mode voorziet in het volgende -

- *Je kunt de tekst met de muis markeren om te knippen, kopiëren, en plakken.*
- *Je kunt de Menubalk met de File, Edit, Window, Tools, Syntax en Help knoppen items gebruiken.*
- *In de nabije toekomst van gvim zal een tweede menubalk de lijst met de te wijzigen bestanden tonen, en kun je tussen bestanden schakelen door op de bestandsnamen te klikken, tot dan kun je gebruik maken van de vi-opdrachten -:e#, :e#1, :e#3, :#4, ...enzovoort om de bestanden te selecteren.*

5 Vi bondgenoten

In het algemeen wordt Vim in samenwerking met andere krachtige tools zoals ctags en gdb gebruikt. De ctags is voor zeer snelle navigatie door miljoenen regels "C/C++"code en gdb voor het debuggen van de "C/C++"code. Een beknopte introductie van deze twee onmisbare opdrachten zal in dit hoofdstuk worden gegeven.

De ctags is de krachtigste beschikbare opdracht voor het coderen in C, C++, Java, Perl, Korn/Bourne shell scripts of Fortran. Ontwikkelaars gebruiken ctags zeer uitgebreid om "naar en vanuit"duizende functies C/C++ programma's te navigeren. Zie onder Unix 'man ctags'. Het is zeer belangrijk dat je MOET leren hoe deze ctags te gebruiken om programma's in C of C++, Java, enz.. te ontwikkelen. Navigatie is de belangrijkste taak tijdens het ontwikkelen in C of C++ code. Met behulp van ctags kun je de code zeer snel lezen door te springen van de aanroepende regel naar de functie en dieper in geneste functie-aanroepen gaan en naar het begin teruggaan. Je kunt zeer snel heen en weer van functie naar functie gaan.

Zonder NAVIGATIE zal je waarschijnlijk verloren zijn! ctags is als de magnetische KOMPAS naald voor de programmeurs.

Gebruik van ctags :

```
ctags *.cpp
gvim -t foo_function
gvim -t main
```

Hiermee zal het C++ programmabestand met daarin de functie foo_funtion() worden gewijzigd en zal de cursor automatisch op de eerste regel van de funtie foo_funtion() worden geplaatst. De tweede opdracht brengt je naar de regel met de main() functie definitie.

Binnen de Vim editor, kun je naar een functie springen door het intikken van een : (dubbele punt) tag < functie naam > als hieronder -

```
:tag voorbeeld_functie
```

Hiermee zal de cursor op de eerste regel van voorbeeld_functie() worden geplaatst.

Als je binnen de functie naar een regel met de naam van de functie wilt springen, plaats je de cursor direct voor de naam van de functie en druk je op CTRL+] (druk tegelijkertijd

op de control-toets en linker blokhaak), hierdoor zal je automatisch direct rechts van de regel terechtkomen waar de functie begint!!

```
// voorbeeldcode
switch(id_number) {
        Case 1:
                if ( foo_function( 22, "abcef") == 3 )
                    ^
                    |
                    |
                    |
  Plaats de cursor hier (direct voor foo_function) en druk op CTRL+]
  Hierdoor zal je naar de functie met de naam "foo_function" gaan.
  Druk op CTRL+t om naar deze regel terug te keren.
```

Druk op CTRL+t (Control-toets en letter 't' samen) om terug te gaan naar de aanroepende regel. Je blijft de CTRL+t indrukken om terug te gaan en naar de eerste regel te gaan vanwaar je de navigatie begon. Dat wil zeggen dat je de CTRL+] in kan blijven drukken en dan de CTRL+t om terug te gaan. Je kunt dit zovaak herhalen als je wilt voor volledige navigatie door de C of C++ functies.

5.1 Ctags voor ESQL

Aangezien ctags de Embedded SQL/C (ESQL) taal niet direct ondersteunt, kan het volgende script worden gebruikt om tags voor esql aan te maken. ESQL/C zijn database SQL opdrachten embedded binnen de "C"programma's. Oracle's ESQL/C wordt Pro*C genoemd en Sybase, Informix heeft ESQL/C en PostgreSQL heeft het produkt ëcpg".

Bewaar dit bestand als "sqltags.shën pas een a+rx tags_ gen.sh toe.

```
#!/bin/sh

# Programma om ctags voor ESQL, C++ en C bestanden aan te maken
ESQL_EXTN=pc
tag_file1=tags_file.1
tag_file2=tags_file.2

which_tag=ctags

rm -f $tag_file1 $tag_file2 tags

aa=`ls *.$ESQL_EXTN`
#echo $aa
for ii in $aa
do
        #echo $ii
        jj=`echo $ii | cut -d'.' -f1`
        #echo $jj

        if [ ! -f $jj.cpp ]; then
```

```
                echo " "
                echo " "
                echo "************************************************"
                echo "ESQL *.cpp bestanden bestaan niet.. "
                echo "Je moet de *.cpp vanuit het *.pc bestand genereren"
                echo "met behulp van de Oracle Pro*C pre-compiler of Sybase"
                echo "of Informix esql/c pre-compiler."
                echo "En vervolgens deze opdracht weer opnieuw uitvoeren"
                echo "************************************************"
                echo " "
                exit
        fi

        rm -f tags
        $which_tag $jj.cpp
        kk=s/$jj\.cpp/$jj\.pc/g

        #echo $kk > sed.tmp
        #sed -f sed.tmp tags >> $tag_file1

        #sed -e's/sample\.cpp/sample\.pc/g' tags >> $tag_file1
        sed -e $kk tags >> $tag_file1
done

# Handel nu alle C++/C bestanden af - sluit de ESQL *.cpp bestanden uit
rm -f tags $tag_file2
bb=`ls *.cpp *.c`
aa=`ls *.$ESQL_EXTN`
for mm in $bb
do
        ee=`echo $mm | cut -d'.' -f1`
        file_type="NOT_ESQL"
        # Sluit de ESQL *.cpp en *.c bestanden uit
        for nn in $aa
        do
                dd=`echo $nn | cut -d'.' -f1`
                if [ "$dd" = "$ee" ]; then
                        file_type="ESQL"
                        break
                fi
        done

        if [ "$file_type" = "ESQL" ]; then
                continue
        fi

        rm -f tags
        $which_tag $mm
        cat tags >> $tag_file2
done
```

```
mv -f $tag_file2 tags
cat  $tag_file1 >> tags
rm -f $tag_file1

# Moet tags bestand sorteren wil het juist werken ....
sort tags > $tag_file1
mv $tag_file1 tags
```

5.2 Ctags voor Javascript programma's, Korn, Bourne shells

Het hieronder gegeven shell-script kan worden gebruikt om tags voor een zeer grote variëteit aan programma's geschreven in variety of programs written in JavasScript, PHP/FI scripts, Korn shell, C shell, Bourne shell en vele anderen te genereren. Dit is een zeer algemene module.

Bewaar dit bestand als tags_gen.sh en pas een chmod a+rx tags_gen.sh toe.

```
#!/bin/sh

tmp_tag=tags_file
tmp_tag2=tags_file2

echo " "
echo " "
echo " "
echo " "
echo " "
echo "Genereer tags voor ...."
while :
do
        echo "    Voer de bestandsextensie in waarvoor je tags wilt genereren."
        echo -n "  Bestandsextensie zou iets moeten zijn als sh, js, ksh, enz... : "
        read ans

        if [ "$ans" == "" ]; then
                echo " "
                echo "Onjuiste invoer. Probeer opnieuw!"
        else
                break
        fi
done

\rm -f $tmp_tag

aa=`ls *.$ans`

for ii in $aa
do
        jj=`echo $ii | cut -d'.' -f1`
```

```
        #echo $jj
        cp $ii $jj.c
        ctags $jj.c
        echo "s/$jj.c/$ii/g" > $tmp_tag2
        sed -f $tmp_tag2 tags >> $tmp_tag
        \rm -f tags $jj.c
done

sort $tmp_tag > tags

\rm -f $tmp_tag $tmp_tag2
```

5.3 Debugger gdb

Je zou samen met Vi uitgebreid gebruik moeten maken van gdb. Debuggen is het belangrijkste aspect van programmeren aangezien de belangrijkste kosten van een softwareproject in het debuggen en testen gaan zitten.

Gebruik de too 'gdb' voor het debuggen van C++/C programma's. Zie 'man gdb'. Je moet je programma's met de optie -g3 compileren als

```
        gcc -g3 foo.c foo_another.c sample.c
```

Voor het eenvoudig instellen van aliassen -

```
    Stel een alias in in je ~/.bash_profile
        alias gdb='gdb -directory=/home/src -directory=/usr/myname/src '
    Give -
        gdb foo.cpp
        gdb> dir /hom2/another_src
        Dit zal aan het bestandszoekpad worden toegevoegd
        gdb> break 'some_class::func<TAB><TAB>
    Hiermee zal de functienaam worden aangevuld om je tijd te sparen bij het typen... en zal uitvoer produceren als -
        gdb> break 'some_class::function_foo_some_where(int aa, float bb)'
```

Het tweemaal indrukken van de TAB-toets is de voltooiïng van de opdrachtregel, wat je heel veel tijd zal besparen bij het typen. Dit is één van de belangrijkste technieken bij het gebruik van gdb.

Voor online hulp -

```
        gdb> help
    Geeft online hulp
        gdb> help breakpoints
    Geeft meer details over breakpoints.
```

Het instellen van breakpoints en het debuggen

```
        unixprompt> gdb exe_filename
        gdb> b main
   Hiermee zal een breakpoint in de main() functie worden geplaatst
        gdb> b 123
   Hiermee zal een breakpoint in regel 123 van het huidige bestand worden geplaatst.
        gdb> help breakpoints
   Geeft meer details over breakpoints.
```

Voor het analyseren van de core dumps

```
        unixprompt> gdb exe_filename  core
        gdb> bt
    Geeft backtrace van functies en regelnummers waar het programma in gebreke bleef
        gdb> help backtrace
    Geeft meer details over backtrace.
```

Je kunt ook de GUI-versie van gdb, genaamd xxgdb gebruiken.

Memory leak tools -

- ***Freeware Electric Fence on linux cd,***
- ***Commercial tools Purify*** `<http://www.rational.com>`
- ***Insure++*** `<http://www.insure.com>`

6 Online VIM help

Zie de online man pages. Typ achter de unix shell-prompt man vim en man gvim.

Of in de gvim sessie tik je :help in voor de helppagina. Zie ook de 8 ()Vim Tutiorial.

```
                            VIM - hoofd helpbestand

          Manoeuvreren:  Gebruik de cursortoetsen, of "h" om naar links te gaan,
                         "j" om naar beneden te gaan, "k" om naar boven te gaan,
                         "l" om naar rechts te gaan.
                         ":1" ga naar 1e regel van pagina
                         ":n" ga naar de n-de regel van pagina
                         "<SHIFT>g" ga naar de onderkant van pagina
                         ":/eenwoord/ zoek naar "eenwoord" in doc

     Sluit dit venster:  Gebruik ":q<Enter>".

     Ga naar onderwerp:  Positioneer de cursor op een tag tussen |<tag>| en druk op
                         CTRL-].

           Met de muis:  ":set mouse=a" om de muis te activeren (in xterm of GUI).
                         Dubbelklik met de linkermuisknop op een tag tussen |<tag>|.

              ga terug:  Typ CTRL-T of CTRL-O.

       Specifieke hulp:  Het is mogelijk direct naar een bepaald hulponderwerp
                         te gaan, door een argument aan de ":help" opdracht
                         |:help| op te geven.
                         Het is mogelijk de context verder te specificeren:
                               WAT                    ERVOOR     VOORBEELD    ~
                         Normale modus opdrachten    (niets)    :help x
                         Visual modus opdrachten       v_       :help v_u
                         Insert modus opdrachten       i_       :help i_<Esc>
                         command-line opdrachten       :        :help :quit
                         command-line editing          c_       :help c_<Del>
                         Vim opdrachtargumenten        -        :help -r
```

```
                         opties                       '        :help 'textwidth'

     lijst met documentatie-bestanden:

     |howto.txt|     hoe het meest algemene te doen
     |intro.txt|     introductie Vim
     |index.txt|     alfabetische index voor iedere mode
     |autocmd.txt|   automatisch uitvoeren van opdrachten bij een actie
     |change.txt|    verwijderen en vervangen van tekst
```

7 Vim Home page en Vim links

De homepage van vim is te vinden bij `<http://www.vim.org>` ***en de mirror-site in de US*** `<http://www.us.vim.org>`

Vim FAQ is bij `<http://www.grafnetix.com/~laurent/vim/faq.html>` ***en bij*** `<http://www.vim.org/faq>`

Eli's Vim Page op `<http://www.netusa.net/~eli/src/vim.html>`

De Vi Lovers Home Page `<http://www.cs.vu.nl/~tmgil/vi.html>`

Vim Reference Guide op `<http://scisun.sci.ccny.cuny.edu/~olrcc/vim/>`

Vim mailing list op `<http://www.findmail.com/listsaver/vimannounce.html>` ***en*** `<http://www.vim.org/mail.html>`

Mailing list archieven worden bewaard op:

- `<http://www.egroups.com/group/vim>`
- `<http://www.egroups.com/group/vimdev>`
- `<http://www.egroups.com/group/vimannounce>`

Vim macro's `<http://www.grafnetix.com/~laurent/vim/macros.html>`

8 Vim Tutorial

8.1 Vim Hands-on Tutorial

Kijk voor de tutorial op een Linux-systeem in /usr/doc/vim-common-5.*/tutor, ga onder andere unix-systemen naar de directory waarin vim is geïnstalleerd en zoek naar de doc directory.

```
bash$ cd /usr/doc/vim-common*/tutor
bash$ less README.txt
bash$ cp tutor $HOME
bash$ cd $HOME
bash$ less tutor
```

8.2 Vi Tutorials op het Internet

- ***Purdue University*** <http://ecn.www.ecn.purdue.edu/ECN/Documents/VI/>
- ***Quick Vi tutorial*** <http://linuxwww.db.erau.edu/LUG/node165.html>
- ***Advanced Vi tutorial*** <http://www.yggdrasil.com/bible/bible-src/user-alpha-4/guide/node171.html>
- ***Tutorials*** <http://www.cfm.brown.edu/Unixhelp/vi_.html>
- ***Tutorials*** <http://www.linuxbox.com/~taylor/4ltrwrd/section3_4.html>
- ***Unix world online vi tutorial*** <http://www.networkcomputing.com/unixworld/unixhome.html>
- ***Univ of Hawaii tutorial*** <http://www.eng.hawaii.edu/Tutor/vi.html>
- ***InfoBound*** <http://www.infobound.com/vi.html>
- ***Cornell Univ*** <http://www.tc.cornell.edu/Edu/Tutor/Basics/vi/>
- ***Vi Lovers home page:*** <http://www.cs.vu.nl/~tmgil/vi.html>
- ***Zal na Sept 2000, verplaatst worden naar*** <http://www.thomer.com/thomer/vi/vi.html>
- ***Beginner's Guide to vi*** <http://www.cs.umr.edu/unixinfo/general/packages/viguide.html>
- ***vi Help file*** <http://www.vmunix.com/~gabor/vi.html>
- ***vim FAQ*** <http://www.math.fu-berlin.de/~guckes/vim/faq/> ***Er bestaan vele Vi Tutorials op het internet. Tik bij Yahoo (Lycos, excite of Hotbot) "Vi Tutorialïn het zoekveld en de zoekmachine zal vele verwijzingen retourneren.***

9 Vi Tutorial

In deze tutorial beschrijven we een aantal "gevorderde"vi concepten en opdrachten, zodat je de kracht van vi kunt leren waarderen en dus beslissen hoe je je kennis van vi opdrachten kunt uitbreiden. Bijna alle vi referenties sommen de beschikbare opdrachten op, maar veel gaan er niet op in hoe de opdrachten met elkaar in verband staan; dit onderwerp is het hoofddoel van deze tutorial.

9.1 Cursor verplaatsingsopdrachten

De vi cursorverplaatsingsopdrachten maken het mogelijk de cursor met een minimum aantal toetsaanslagen efficiënt in het bestand en/of op het scherm te positioneren. Er zijn heel veel cursorverplaatsingsopdrachten - probeer ze niet allemaal tegelijkertijd te onthouden! Later zullen we zien dat veel van de kracht van vi komt door het mixen van cursorverplaatsingsopdrachten met andere opdrachten om tekst te verwijderen, wijzigen, yanken (kopiëren), en filteren.

Wijzig alsjeblieft een groot tekstbestand (zoals bijvoorbeeld wknight) zodat je met iedere opdracht kunt experimenteren op 't moment dat 't wordt beschreven. Houd in gedachten dat deze opdrachten alleen in de Commando-modus werken, niet in de Insert Modus; als je -opdrachtenïn je tekst krijgt, druk dan op de ESC-toets om naar de Commando-modus terug te keren.

- *cursortoetsen : Zoals we hebben gezien, kunnen we met de cursorpijlen een enkel teken naar links, beneden, boven en rechts opschuiven. Verplaatsing voorbij de bovenkant, de onderkant, rechts van het einde van een regel, of links van het begin van een regel van het bestand is niet toegestaan (geen regelomslag).*
- *hjkl : Toen vi werd geschreven (rond 1978), hadden veel terminals op UNIX-systemen geen cursortoetsen! h, j, k, en l werden gekozen als opdrachten om respectievelijk naar links, beneden, boven, en naar rechts te gaan. Probeer ze uit! De meeste vi fanaten geven de voorkeur aan deze cursortoetsen omdat*
  - *(a) ze op alle toetsenborden op dezelfde plaats voorkomen, en*
  - *(b) ze prettig onder de vingers passen, in tegenstelling tot de meeste cursortoetsen, die in een box of "Töf één of andere niet lineaire vorm zijn gearrangeerd.*

  Waarom h, j, k, en l? In de ASCII character set, staat CTRL-H voor de backspace (verplaatsing naar links), CTRL-J voor linefeed (verplaatsing naar beneden), en, k en l bevinden zich direct naast h en j, dus zoals je kunt zien zijn ze makkelijk te onthouden.
- *0 : ("nul", niet öh") Ga naar het begin van de huidige regel. (gebruik de cursortoetsen of h j k l om naar een ingesprongen tekstregel met een paar ë"tekens erin te gaan, om dit en de volgende paar opdrachten uit te proberen. Als je geen ingesprongen regel in je bestand kunt vinden, maak er dan één aan door een paar spaties aan het begin van een regel in te voegen).*
- *^ : Ga naar het eerste niet-blanco teken van de huidige regel. (Voor een ingesprongen regel, hebben 0 en ^ een verschillende betekenis).*
- *$: Ga naar het laatste teken van de huidige regel.*
- *tC : Ga naar (vlak ervoor) het volgende teken c in de huidige regel. (Druk op 0, en tik dan 'te' in. Hiermee zal naar de eerste 'e' in de huidige regel worden gegaan).*
- *fC : Zoek naar (plaatst cursor onder teken) volgende teken 'c' in de huidige regel. (Tik 'fe' in, en de cursor zal de volgende 'e' vinden - dat wil zeggen, verplaatst worden naar - de volgende 'e' in de huidige regel).*
- *TC : Ga naar (direct erachter) het vorige teken c in de huidige regel (Druk op $, dan Te).*
- *FC : Zoek (plaatst cursor onder teken) het vorige teken c in de huidige regel. (Druk op Fe.)*
- *n| : Ga naar kolom n in de huidige regel. (Probeer 20 |. De cijfers 2 en 0 zullen, als je ze intikt, niet worden getoond, maar als je de | intikt, zal de cursor naar kolom 20 gaan).*

  Probeer wat met t f T F | te experimenteren. Als je iets niet goed doet, zal vi een beep genereren.
- *w : Voorwaarts naar het begin van het volgende "kleine"woord (een "klein"woord bestaat uit een onafgebroken reeks alfanumerieke tekens of leestekens, maar niet gemengd alfanumeriek en leestekens). Probeer een aantal keren de w - merk op wat er gebeurt bij een leesteken.*
- *W : Voorwaarts naar het begin van het volgende "grote"woord (gemengd alfanumeriek en leestekens). Probeer een aantal dozijn maal de W.*

- ***b** : Terugwaarts naar het begin van een "klein"woord.*
- ***B** : Terugwaarts naar het begin van een "groot"woord.*
- ***e** : Voorwaarts naar het einde van een "klein"woord.*
- ***E** : Voorwaarts naar het einde van een "groot"woord.*
- ***+ Return** : Ga naar het eerste niet-blanco teken op de volgende regel. (+ en de Return toets hebben hetzelfde effect).*
- ***-** : Ga naar het eerste niet-blanco teken op de vorige regel.*
- ***)** : Ga naar het einde van de zin. (Een zin eindigt of met een lege regel of met een punt of een uitroepteken gevolgd door twee spaties of op het einde van een regel. Een punt of uitroepteken gevolgd door een spatie betekent niet het einde van een zin; dit gedrag is juist, overeenkomstig traditionele regels van hoe zinnen in getypte documenten zouden moeten verschijnen, maar lijkt vaak onjuist voor degenen die nooit typeles hebben gehad).*
- ***(** : Ga naar het begin van de zin.*
- **}** *: Ga naar het einde van de paragraaf. (Paragrafen worden gescheiden door lege regels, volgens de definitie van vi).*
- **{** *: Ga naar het begin van de paragraaf.*
- ***H** : Ga naar de home-positie (bovenste regel) op het scherm.*
- ***M** : Ga naar de middelste regel op het scherm.*
- ***L** : Ga naar de laatste regel op het scherm.*
- ***nG** : Ga naar regel n. Als n niet wordt opgegeven, ga dan naar de laatste regel in het bestand. (Probeer bijvoorbeeld 15G om naar regel 15 te gaan. De CTRL-G opdracht toont de naam van het bestand, wat statusinformatie en het huidige regelnummer. Om naar het begin van het bestand te gaan: 1G)*
- ***CTRL-d** : Scroll een half scherm omlaag (zie noot).*
- ***CTRL-u** : Scroll een half scherm omhoog (zie noot).*
- ***CTRL-f** : Ga een scherm verder (zie noot).*
- ***CTRL-b** : Ga een scherm terug (zie noot).*
- ***Noot** : Deze vier scrolling/paging opdrachten kunnen niet met de verwijder-, wijzigings-, kopieer- of filteropdrachten worden gebruikt.*
- ***/reg_exp** : Ga naar het volgende voorkomen van de reguliere expressie reg_exp. Als je de / intikt, wordt de cursor naar de linkeronderkant van het scherm geplaatst en wacht totdat je de reguliere expressie typt. Druk op de Return toets om de invoer te bevestigen; vi zoekt dan in voorwaartse richting naar het volgende voorkomen van de reguliere expressie. Bijvoorbeeld, tik in /de gevolgd door een druk op de Return. Hiermee wordt naar het volgende voorkomen van 'de' gegaan, misschien onderdeel uitmakend van een wat langer woord (ander, weder, enz.). Als je de / intikt, gevolgd door de Return, zoekt vi naar de laatst opgegeven reguliere expressie waarnaar werd gezocht.*
- ***n** : Heeft hetzelfde effect als het indrukken van de / en de Return; d.w.z., zoekt naar het volgende voorkomen van de laatste reguliere expressie waarnaar je zocht.*

- *?reg_exp : Zoek voorwaarts, in plaats van terugwaarts. Als er geen reg_exp werd opgegeven, zoekt het naar de laatste reguliere expressie die werd ingevoerd. Het zoeken met / en ? "na"de onderkant en "voor"de bovenkant van het bestand is toegestaan.*
- *N : Hetzelfde als het indrukken van ? en dan op Return.*

9.2 Herhalingen

Veel van de verplaatsingsopdrachten die hiervoor werden besproken kunnen worden voorafgegaan door een getal waarmee wordt aangegeven hoeveel maal de erna volgende opdracht moet worden herhaald; de verplaatsing wordt gewoon het gegeven aantal malen herhaald:

- *3w : Ga drie woorden vooruit*
- *5k : Ga vier tekens naar boven*
- *3fa : Zoek de derde opeenvolgende a in de huidige regel*
- *6+ : Ga zes regels naar beneden*

Voor een aantal opdrachten hebben de herhalingen een speciale betekenis:

- *4H : Ga naar regel 4 op het scherm (home plus 3)*
- *8L : Ga naar de achtste regel vanaf de onderkant van het scherm.*
- *3$: Ga naar het einde van de derde regel van onder*

Bij een aantal opdrachten (b.v., ^) wordt de herhaling genegeerd; bij anderen (b.v., / en ?) is het niet toegestaan

9.3 Verwijderen van Tekst

We hebben gezien dat dd de huidige regel verwijdert. Dit kan met een herhaling worden gebruikt: 3dd verwijdert drie regels, de huidige regel en de 2 daarop volgende regels.

De d opdracht kan als een "voorvoegsel"worden gebruikt met de meeste verplaatsingsopdrachten hierboven om bijna willekeurige stukken tekst te verwijderen. De verplaatsingsopdrachten worden "target specifiers"genoemd, als ze in combinatie met d worden gebruikt. Aan d kan een herhaling worden opgegeven. (Als je hiermee gaat experimenteren, denk er dan aan na iedere opdracht de u in te tikken om de verwijdering ongedaan te maken).

- *dw : Verwijder "klein"woord voorwaarts*
- *d3w : Verwijder drie "kleine"woorden voorwaarts*
- *3dw : Drie keer, verwijder "klein"woord voorwaarts*
- *3d3w : Drie keer, verwijder drie "kleine"woorden voorwaarts (dat wil zeggen, verwijder negen "kleine"woorden voorwaarts)*
- *d+ : Verwijder huidige en daaropvolgende regel*
- *d/de : Verwijder vanaf het huidige teken tot aan, maar niet inclusief het volgende voorkomen van het patroon 'de'*

- *d$: Verwijder tot aan het einde van de regel*
- *d0 : Verwijder tot aan het begin van de regel*
- *d30G : Verwijder vanaf de huidige regel tot en met regel 30*
- *dG : Verwijder vanaf de huidige regel tot en met de laatste regel*
- *d1G : Verwijder vanaf de huidige regel tot en met regel 1*

Gebruik x om enkele tekens te verwijderen. aan x kan een herhaling worden opgegeven:

- *15x : Verwijder het huidige en de 14 daaropvolgende 14 tekens*

x is in feite gewoon een afkorting van d1; dat wil zeggen, verwijder één teken rechts.

9.4 Wijzigen van Tekst

De c opdracht is vergelijkbaar met d, behalve dat het vi naar Insert mode laat schakelen, waarbij het mogelijk wordt de oorspronkelijke (ongewenste) tekst in iets anders te veranderen.

Plaats bijvoorbeeld de cursor aan het begin van een woord (druk op w om naar het begin van het volgende woord te gaan). Druk dan op cw om dat woord te wijzigen. Op het scherm zal het laatste teken in het woord dat wordt gewijzigd, worden vervangen door een $-symbool om de grens van de wijziging aan te geven; typ een nieuw woord in (je zult het oorspronkelijke woord op het scherm overschrijven) en druk de ESC-toets in als je klaar bent. Wat je invoert mag langer of korter zijn dan het woord dat wordt gewijzigd.

Plaats de cursor aan het begin van een regel bestaande uit op z'n minst drie woorden, en probeer c3w om drie woorden te wijzigen. Probeer c$ om tot het einde van de huidige regel te wijzigen. In alle gevallen wordt de grens aangegeven met $ daar waar de wijziging alleen effect heeft op de huidige regel.

Als een wijziging op meer dan alleen de huidige regel effect heeft, verwijdert vi de oorspronkelijke tekst van het scherm en schakelt over naar Insert-mode. Probeer bijvoorbeeld eens met c3+ de huidige en 3 volgende regels te wijzigen; vi verwijdert de vier oorspronkelijke regels van het scherm en schakelt over naar Insert-mode op een nieuwe blanco regel. Druk zoals gewoonlijk op de ESC-toets wanneer je klaar bent met het invoeren van je nieuwe tekst.

Een aantal andere wijzigingsopdrachten:

- *cc : Wijzig huidige regel*
- *5cc : Wijzig vijf regels (huidige en de volgende vier)*
- *c/de : Wijzig vanaf huidige teken tot aan, maar niet inclusief het volgende voorkomen van het volgende patroon dat voldoet aan 'de'*
- *c$: Wijzig tot aan het einde van de regel*
- *c30G : Wijzig vanaf de huidige regel tot en met Regel 30*
- *cG : Wijzig vanaf de huidige regel tot en met de laatste regel*
- *c1G : Wijzig vanaf de huidige regel tot en met Regel 1*

9.5 Yanken (kopiëren) van Tekst

De y opdracht kopieert een kopie van tekst naar een buffer; de gekopieerde tekst kan dan met p of P elders in het bestand worden geplaatst (of geplakt).

De eenvoudigste vorm van yank is yy om de huidige regel te kopiëren; probeer na een yy een kopie van de gekopieerde regel na de cursor te plaatsen. Na een yy, kun je zoveel kopieën van de gekopieerde regel maken als je wilt door naar boven en naar beneden in het bestand te gaan en op p te drukken.

Probeer voor het kopiëren van meerdere regels bijvoorbeeld 5yy (kopieer de huidige en de volgende vier regels). p plaatst een kopie van de gekopieerde regels na de cursor; de reeks 5yyp "werkt"maar het is waarschijnlijk niet wat je wilt. De P opdracht is vergelijkbaar met p, maar het plaatst een kopie van de gekopieerde tekst voor de cursor; probeer de reeks 5yyP.

Andere kopieer-opdrachten:

- *y3w : Kopieer drie woorden*
- *y$: Kopieer tot aan het einde van de huidige regel*
- *y1G : Kopieer vanaf de huidige regel tot en met Regel 1*

9.6 Filteren van tekst

De filteropdracht !, verwacht de naam van een UNIX-opdracht (wat een filter zou moeten zijn), en geeft de geselecteerde regels dan door aan het filter, waarbij de geselecteerde regels in de vi-buffer dan worden vervangen door de uitvoer van het filteropdracht. vi's vermogen om bijna willekeurige stukken tekst via iedere UNIX-filter door te geven voegt, zonder ëxtra kostenäan vi een ongelofelijke flexibliteit toe.

Ter illustratie zal een aantal voorbeelden van hulp zijn. Maak in het bestand een regel met slechts het woord 'who' en absoluut geen andere tekst. Plaats de cursor op deze regel, en druk op !!. Deze opdracht is analoog aan dd, cc, of yy, maar in plaats van het verwijderen, wijzigen of kopiëren van de huidige regel, filtert het de huidige regel. Als je de tweede ! indrukt, gaat de cursor naar de linkerhoek aan de onderkant van het scherm en wordt een enkele ! getoond, daarbij wachtend totdat je de naam van een filter opgeeft. Typ als filternaam sh en druk op de Return-toets. sh (de Bourne shell) is een filter! Het leest van standaarinvoer, verwerkt zijn invoer (dat wil zeggen, voert opdrachten uit) en stuurt de uitvoer (de uitvoer van die opdrachten) naar standaarduitvoer. Het filteren van de regel met who via sh zorgt ervoor dat de regel met who wordt vervangen door een lijst met de huidige gebruikers op het systeem - direct in je bestand!

Probeer dit proces te herhalen met date. Dat wil zeggen, maak een regel met niets anders dan het woord date, plaats de cursor dan op deze regel, druk op !!sh en de Return-toets. De regel met date wordt vervangen door de uitvoer van de date opdracht.

Plaats de cursor op de eerste regel van de uitvoer van who. Tel het aantal regels. Stel bijvoorbeeld dat het aantal zes is. Selecteer dan deze zes regels om via sort te laten filteren; druk op 6!!sort en vervolgens de Return-toets. De zes regels zullen aan sort worden doorgegeven, en de zes oorspronkelijke regels worden door de uitvoer van sort vervangen.

De filteropdracht kan alleen op gehele regels worden toegepast, niet op tekens of woorden.

Nog wat andere filter-opdrachten (hier betekent, $< CR >$ een druk op de Return-toets):

- *!/de < CR > sort < CR > : Sorteer vanaf de huidige regel tot en met de volgende regel met daarin het woord ‘de’.*
- *!1Ggrep de < CR > : Vervang vanaf de huidige regel tot en met Regel 1 door slechts die regels met daarin het woord ‘de’.*
- *`!Gawk '{print $1}'` < CR > : Vanaf de huidige regel tot aan het einde van het bestand, het vervangen van iedere regel door slechts het eerste woord.*

9.7 Markeren van Regels en Tekens

Je kunt regels en tekens markeren om te worden gebruikt als doel voor verplaatsing, verwijdering, wijziging, ze te kopiëren, of filteren met behulp van de opdracht mc, waar c staat voor een kleine letter.

Plaats de cursor bijvoorbeeld in het midden van een woord en druk op ma. Hiermee wordt het teken onder de cursor met a gemarkeerd.

Verplaats de cursor nu naar elders, weg van het gemarkeerde teken en naar een andere regel (gebruik de cursorpijlen, CTRL-u of iets dergelijks). Om naar de gemarkeerde regel terug te keren, tik je 'a in. (dat wil zeggen, een enkel aanhalingsteken en dan een a). Hierdoor wordt de cursor naar het eerste niet-blanco teken in de regel gemarkeerd met a verplaatst.

Ga weer weg van deze regel. Druk op ‘a (dat wil zeggen, een aanhalingsteken achterwaarts en dan een a) om naar het gemarkeerde teken terug te keren. Hierdoor wordt de cursor naar het teken gemarkeerd met a verplaatst.

Markeringen worden meestal gebruikt in combinatie met verwijderingen, wijzigingen, kopieer- of filteropdrachten. Verplaats de cursor bijvoorbeeld naar een andere regel dan die met de markering a en druk dan op d'a (d, een enkel aanhalingsteken en een a). Hiermee wordt de huidige regel tot en met de regel gemarkeerd met a verwijderd.

Plaats de cursor in het midden van een ander woord en druk op mb om de markering b in te stellen. Verplaats de cursor nu, weg van dat woord (maar slechts een paar regels, zodat je makkelijk kunt zien wat we aan het doen zijn), en druk dan op d‘b (d, aanhalingsteken achterwaarts, b). Hierdoor wordt het huidige teken tot en met het teken gemarkeerd met b verwijderd.

Een ander voorbeeld is om de uitvoer van who te sorteren, markeer de eerste regel (ma), verplaats de cursor dan naar de laatste regel en druk op !'asort en dan de Return-toets.

Als je naar een markering springt en besluit terug te gaan naar waar je vandaan sprong, kun je ” intikken (ga terug naar regel ..) of “ (ga terug naar het teken ..).

9.8 Benoemen van Buffers

Als je tekst verwijdert, wijzigt of kopieert, wordt de oorspronkelijke tekst (tot de volgende verwijdering, wijziging of kopieeropdracht) in een onbenoemde buffer opgeslagen van waaruit het kan worden opgehaald met p of P. Door gebruik te maken van de onbenoemde buffer, kan alleen de meest recente verwijderde, gewijzigde of gekopieerde tekst worden hersteld.

Als je meerdere delen van je tekst wilt verwijderen, wijzigen of kopiëren en je wilt ze allemaal onthouden (tot aan een maximum van 26), kun je voor de wijzigings- of kopieeropdracht aan een buffer een naam toekennen. De naam van een buffer heeft de vorm "c (dubbel aanhalingsteken, kleine letter c).

Tik bijvoorbeeld äyy in om de huidige regel naar buffer a te kopiëren. Verplaats je vervolgens naar een andere regel en druk op "byy om die regel naar buffer b te kopiëren. Verplaats de cursor nu naar elders in het bestand en tik in äp en "bp om kopieën van de tekst te plaatsen die in de buffers a en b zijn geplaatst.

Nog wat andere benoemde bufferopdrachten:

- *ä6yy : Kopieer zes regels (de huidige en de volgende vijf) naar buffer a*
- *"bd1G : Verwijder vanaf de huidige regel tot en met Regel 1, waarbij de verwijderde regel in buffer b wordt opgeslagen*
- *"cy'c : Kopieer vanaf de huidige regel gemarkeerd als c naar buffer c (markeringen en buffers zijn verschillend, en kunnen dezelfde naam hebben zonder dat vi daarvan in de war raakt).*

9.9 Substituties

Gebruik de :s opdracht om een stuk tekst door een ander stuk tekst in regels door je hele bestand te vervangen. Wat substitueer voorbeelden:

- *:1,$s/de/DE/g Vervang van regel 1 tot aan de laatste regel (regel $), het woord 'de' door 'DE' doe dit globaal in iedere regel waar 'de' voorkomt*
- *:'a,.s/.*/ha ha/ Vervang in de regel gemarkeerd met 'a' tot aan de huidige regel (regel .) alles op de regel door de tekst 'ha ha'*

9.10 Diverse "Commando's beginnend met een dubbele punt"

Alle volgende opdrachten beginnen met een dubbele punt; als je de dubbele punt intikt, gaat de cursor naar de linkeronderkant van het scherm en wordt er een prompt, bestaande uit een dubbele punt, weergegeven, die wacht totdat je gereed bent met het invoeren van je opdracht.

Een aantal belangrijke voorbeelden:

- *:w Schrijf de inhoud van de buffer naar het bestand zonder vi te verlaten*
- *:w abc Schrijf de inhoud van de buffer naar het bestand 'abc' (waarbij 'abc' wordt aangemaakt als het niet bestaat of de huidige inhoud overschrijft als het wel bestaat) zonder vi te verlaten*
- *:1,10w abc Schrijf de regels 1 tot en met 10 naar het bestand 'abc'*
- *:'a,$w abc Schrijf vanaf de regel gemarkeerd als a tot aan de laatste regel naar het bestand 'abc'*
- *:e abc Wijzig bestand 'abc', in plaats van het huidige bestand. vi drukt een foutmelding af als er aan het huidige bestand wijzigingen zijn aangebracht die nog niet met :w zijn opgeslagen*
- *:e! abc Wijzig het bestand 'abc', waarbij eventuele wijzigingen die zijn aangebracht aan het huidige bestand worden verworpen*
- *:e # Wijzig het vorige gewijzigde bestand (opeenvolgende :e# opdrachten schakelen tussen twee bestanden)*

- *:f abc Wijzig de bestandsnaam voor de huidige vi buffer in abc*
- *:q Beëindig, tenzij niet opgeslagen wijzigingen zijn gemaakt*
- *:q! Beëindig, waarbij eventuele wijzingingen die zijn aangebracht, worden verworpen*
- *:r abc Lees het bestand 'abc' in de huidige vi buffer, na de regel waarop de cursor staat (probeer :r croc om een kopie van het croc bestand in te lezen)*
- *:!cmd Voer de opdracht cmd uit (who, sort, ls, enz.)*

9.11 Opties Instellen

Diverse opties hebben effect op de "feel"van vi. Je kunt alle diverse opties die kunnen worden ingesteld tonen met de opdracht :set all. Je kunt set ook gebruiken om opties te wijzigen.

Als je bijvoorbeeld regelnummers wilt zien voor de regel in het bestand wat je aan het wijzigen bent, gebruik je de opdracht :set number. Voor het uitzetten van de regelnummering, gebruik je de opdracht :set nonumber. De meeste opties kunnen worden afgekort. :set nu zet de regelnummering aan en :set nonu zet de regelnummering uit.

Met een :set nomagic, wordt de speciale betekenis van de reguliere expressie tekens (punt, asterisk, blokhaak, enz.) uitgeschakeld. Gebruik :set magic om de speciale betekenissen weer te herstellen.

Een aantal opties verwacht een waarde. :set tabstop=4 bijvoorbeeld zorgt ervoor dat tabs als vier spatie-tekens worden weergegeven in plaats van de gebruikelijke acht.

Als je bemerkt dat je bepaalde opties altijd op een bepaalde manier instelt, kun je de set opdrachten die je wilt in een bestand met de naam .exrc plaatsen, of je kunt de omgevingsvariabele EXINIT instellen om de gewenste opties in te stellen.

Als je login-shell bijvoorbeeld de Bourne-shell is, zou deze regel in je .profile bestand geplaatst kunnen worden:

```
EXINIT='set nomagic nu tabstop=4'; export EXINIT
```

Als je login-shell een C-shell is, zou deze regel in je .login bestand geplaatst kunnen worden:

```
setenv EXINIT 'set nomagic nu tabstop=4'
```

9.12 Key Mappings

Als je bemerkt dat je keer op keer een serie eenvoudige opdrachten aanroept, kun je met de opdracht :map de serie opdrachten aan een ongebruikte opdrachttoets toekennen. Als in de toekenning controle-tekens zoals de Return-toets (CTRL-M in ASCII) of de ESC (CTRL-[in ASCII), moeten worden opgenomen, laat dergelijke tekens dan voorafgaan door een CTRL-v om hun speciale betekenis te onderdrukken.

In bijvoorbeeld de hierna volgende opdracht wordt aan de opdracht CTRL-A toegekend dat de cursor 55 regels verder wordt geplaatst, dan weer terug wordt gegaan naar de regel met de minste tekens, waarbij die blanco regel wordt gewijzigd in een formfeed (CTRL-L) gevolgd door drie blanco regels. Dat wil zeggen dat iedere CTRL-A de volgende pagina zal pagineren, zonder dat paragrafen op meerdere pagina's worden gesplitst.

Noot: In deze opdracht wordt ieder controle-teken getoond als ˆC, C is hierbij een hoofdletter. CTRL-M wordt bijvoorbeeld getoond als ˆM. Wanneer je deze opdracht invoert, zul je ook de CTRL-v tekens zoals getoond, niet zien: iedere CTRL-v onderdrukt louter de gebruikelijke speciale betekenis van het volgende controle-teken, dus als je de reeks ˆVˆM intikt, zul je op het scherm slechts ˆM zien. In deze opdracht, is ˆM de Return-toets en ˆ[is de ESC-toets.

```
:map ^A  55+?^$^V^Mcc^V^L^V^M^V^M^V^M^V^[
```

9.13 Wijzigen van Meerdere Bestanden

Je kunt met vi meerdere bestanden wijzigen, door als argumenten op de opdrachtregel, meerdere bestandsnamen op te geven:

```
vi croc fatherw  wknight
```

Twee opdrachten beginnend met een dubbele punt worden gebruikt om je tussen de meerdere bestanden te verplaatsen:

- ***:n Ga naar het volgende bestand in de argumentenlijst (je moet je wijzigingen opslaan met :w anders zal vi een foutmelding weergeven)***
- ***:rew Ga terug en ga weer verder met het eerste bestand in de lijst met argumenten***

De :n en :rew opdrachten zijn wat onhandig, maar er zijn een aantal belangrijke voordelen: de inhoud van benoemde buffers (ä, "b, "c, etc.) worden door de bestanden heen onthouden, dus je kunt :n en :rew met p en P gebruiken om tekst heen en weer tussen bestanden te kopiëren. Ook wordt de meest recente zoekstring voor de / en ? opdrachten door de bestanden heen onthouden, dus je kunt nogal makkelijk herhalende zoekopdrachten in meerdere bestanden gebruiken.

Probeer bijvoorbeeld het volgende experiment: Ga eerst uit vi, start vi dan op met croc en wknight als argumenten:

```
$ vi croc wknight
```

Zoek in croc naar 'de'

/de < CR >

Kopieer deze regel naar buffer a:

äyy

Ga nu naar het volgende bestand (je hebt geen wijzigingen aangebracht aan croc, dus dit werkt):

:n < CR >

Zoek naar de volgende regel met 'de', zonder het opnieuw intikken van de zoekstring:

n

Plaats een kopie van buffer a na de huidige regel in wknight:

äp

Ga twee regels naar beneden, en kopieer de huidige regel naar buffer b:

jj"byy

Sla de wijzigingen in wknight op

:w < CR >

Ga nu terug naar croc

:rew < CR >

Zoek nogmaals, en plaats een kopie van buffer b na de gevonden regel:

n"bp

Sla de wijzigingen op en beëindig vi

ZZ

9.14 Laatste Opmerkingen

Deze tutorial was bedoeld om wat van de mogelijkheden van vi te introduceren waar je in het systeemhandboek van vi misschien overheen zou kijken of die wellicht niet in het handboek worden vermeld (op de verschillende systemen zijn de handboeken van zeer verschillende kwaliteit).

Je zult na het lezen van deze tutorial geen vi-expert zijn, maar je kunt de mogelijkheden van vi dan wel juist inschatten. Slechts tijd en moeite kunnen een vi-expert van je maken. Maar de efficiëntie en alomvattenheid betalen dit op de lange termijn terug.

Mogelijk heb je besloten dat je vi haat. Het zij zo! Maar wees je er bewust van dat vi de standaard UNIX teksteditor blijft - de enige editor waar je op kunt rekenen dat het op ieder UNIX-systeem beschikbaar is - dus zelfs als je de voorkeur geeft om dagelijks iets anders te gebruiken, wordt je geadviseerd om bekend te zijn met het absolute minimum aan het vi-materiaal dat in deze tutorial is behandeld.

10 Vim Referentiekaart

10.1 Vi status

Vi heeft 3 modi:

1. *opdrachtenmodus - Normale en initiële status; anderen keren hier naar terug (gebruik ESC om een gedeeltelijk getypte opdracht af te breken)*

2. *invoer modus - ingegaan door specieke opdrachten a i A I o O c C s S R en beëindigd door ESC of abnormaal door een interrupt*

3. *regel modus - d.w.z. het wachten op invoer na een : , / , ? of een ! opdracht (beëindigd met CR, af te breken met CTRL-c). CTRL is de control-toets: CTRL-c betekent "control c"*

10.2 Shell Commando's

1. *TERM= code Plaatst een code-naam voor je terminal in de variabele TERM*

2. *export TERM Draagt de waarde van TERM over (de terminal code) naar ieder UNIX-systeemprogramma dat terminal afhankelijk is.*
3. *tput init Initialiseert de terminal zodat het juist zal functioneren met diverse UNIX-systeemprogramma's.*
4. *vi filename Geeft toegang tot de schermeditor vi zodat je een specifiek bestand kunt wijzigen.*
5. *vi file1 file2 file3 Laadt drie bestanden in de vi buffer om te worden gewijzigd. Die bestanden zijn file1, file2, en file3.*
6. *view file Roep de vi-editor aan met het bestand file in read-only modus*
7. *vi -R file Roep de vi-editor aan met het bestand file in read-only mode*
8. *vi -r file Herstel het bestand file en recente wijzigingen na een systeemcrash*
9. *vi -r file Herstel het bestand file en recente wijzigingen na een systeemcrash*

10.3 Opties Instellen

1. *:set optie Activeer optie*
2. *:set optie=waarde Wijs waarde toe aan optie*
3. *:set no optie Deactiveer optie*
4. *:set Toon door gebruiker ingestelde opties*
5. *:set all Toon lijst met alle huidige opties, zowel standaard als die zijn ingesteld door de gebruiker*
6. *:set optie? Toon waarden van optie*

10.4 Gebruikte Notaties

Notaties:

1. *CTRL-c CTRL is de control-toets: CTRL-c betekent "control c"*
2. *CR is een Carriage return (ENTER toets)*

10.5 Onderbreken, annuleren

- *ESC beëindig invoeg-modus of onvolledige opdracht*
- *CTRL-? CTRL is de control-toets: CTRL-? betekent "control ?"verwijder of rubout interrupts*
- *CTRL-l reprint/ververs scherm als CTRL-? het door de war gooit*

10.6 Bestandsmanipulatie

- *ZZ Bewaar bestand en beëindig vi*
- *:wq Bewaar het bestand en beëindig vi*
- *:w Sla het huidige bestand op*
- *:w! Forceer schrijven naar het huidige bestand, als het bestand alleen lezen is*
- *:wname Schrijf naar bestand name*
- *:q Beëindig vi*
- *:q! Forceer beëindiging vi (verwerp wijzigingen)*
- *:e name Wijzig bestandsnaam*
- *:e! opnieuw wijzigen, verwerp veranderingen*
- *:e + name wijzig bestand name, beginnend aan het eind*
- *:e + n begin met wijzigen op regeln*
- *:e # wijzig andere bestand*
- *:n wijzig volgende bestand in arglist*
- *:args geef lijst met bestanden in huidige bestandenlijst*
- *:rew wijzig eerste bestand in bestandenlijst*
- *:n args specificeer nieuwe arglist*
- *:f toon huidige bestand en regel*
- *CTRL-G synoniem voor :f , toon huidige bestand en regel*
- *:ta tag naar tag file entry tag*
- *CTRL-] :ta, volgende woord is tag*

10.7 Verplaatsen

- *Cursorpijlen Verplaats de cursor*
- *CTRL-d Scroll halve pagina omlaag*
- *CTRL-u Scroll halve pagina omhoog*
- *CTRL-f Scroll een volledige pagina omlaag*
- *CTRL-b Scroll een volledige pagina omhoog*
- *:0 Ga naar begin van het bestand*
- *:n Ga naar regelnummer n*
- *:$ Ga naar einde bestand*
- *0 Ga naar begin van de regel*
- *^ Ga naar eerste niet-blanco teken*

- *$ Ga naar einde regel*
- *CR Ga naar het begin van de volgende regel*
- *- Ga naar het begin van de vorige regel*
- *% Zoek naar overeenkomende blokhaak*
- *G ga naar regel (standaard de laatste regel)*
- *]] volgende sectie/functie*
- *[[vorige sectie/functie*

10.8 Regel positionering

- *H Linksbovenaan venster*
- *L Laatste regel van venster*
- *M Middelste regel van venster*
- *+ Volgende regel, naar eerst niet-blanco*
- *- Vorige regel, naar eerste niet-blanco*
- *CR return, zelfde als +*
- *j volgende regel, zelfde kolom*
- *k vorige regel, zelfde kolom*

10.9 Positioneren van de cursor

- *0 begin van de regel*
- *$ einde van de regel*
- *h vooruit*
- *l terug*
- *SPACE zelfde als l*
- *fx zoek x voorwaarts*
- *Fx zoek x terugwaarts*
- *; herhaal laatste f F*
- *, inverse van ;*
- *| naar aangegeven kolom*
- *% zoek overeenkomende { of }*

10.10 Woorden, zinnen, paragrafen

- *w Woord vooruit*
- *b Woord achteruit*
- *e Einde woord*
- *) Naar volgende zin*
- *(Zin terug*
- } *Naar volgende paragraaf*
- { *Paragraaf terug*
- *W Volgende woord*
- *B W terug*
- *E Naar einde van W*

10.11 Markeren en terugkeren

- *“ (druk tweemaal op de ‘ toets) Vorige context*
- *” (druk tweemaal op de ‘ toets) Vorige context op eerste niet-blanco teken in regel*
- *mx Markeer positie met letter x*
- *‘x (‘ en letter x) ga naar markering x*
- *’x ga naar markering x op eerste niet-blanco teken in regel*

10.12 Correcties tijdens het invoegen

- *CTRL-h Verwijder laatste teken*
- *CTRL-w Verwijder laatste woord*
- *erase Druk op de DELETE toets, zelfde als CTRL-h*
- *kill Je kill-toets, verwijder invoer deze regel*
- \ *Escapes CTRL-h, DELETE en kill*
- *ESC Beëindig het invoegen, terug naar opdrachtenmodus*
- *CTRL-? Interrupt, beëindig invoer*
- *CTRL-d Terugtab over autoindent*
- *CTRL-v Quote niet-afdrukbaar teken*

10.13 Aanpassen van het scherm

- *CTRL-l Maak het scherm schoon en bouw het opnieuw op*
- *CTRL-r typ opnieuw, elimineer @lines*
- *z-CR ververs scherm, huidige regel bovenkant van venster*
- *z- ververs scherm, huidige regel onderkant van venster*
- *z. ververs scherm, huidige regel in het midden van het venster*
- */pat/z- pat regel onderaan*
- *tn Gebruik n regel venster*
- *CTRL-e Scroll venster 1 regel naar beneden*
- *CTRL-y Scroll venster 1 regel naar boven*

10.14 Verwijderen

- *x Verwijder het teken onder de cursor*
- *X Verwijder het teken voor de cursor*
- *D Verwijder tot aan het einde van de regel*
- *d^ Verwijder terug tot aan het begin van de regel*
- *dd Verwijder de huidige regel*
- *ndd Verwijder n regel te beginnen met de huidige regel*
- *dnw Verwijder n woorden te beginnen vanaf de cursor*

10.15 Invoegen, wijzigen

- *i Ga naar invoermode en voeg in voor de cursor*
- *I Ga naar invoermode en voeg in voor het eerste niet-blanco teken*
- *a Ga naar invoermode en voeg in na de cursor*
- *A Ga naar invoermode en voeg in na het einde van de regel*
- *o Voeg een nieuwe regel in na de regel onder de huidige regel en ga naar invoermode*
- *O Voeg een nieuwe regel in boven de huidige regel en gan naar invoermode*
- *r Vervang het teken onder de cursor (ga NIET naar invoermode)*
- *R Ga naar invoermode en vervang tekens*
- *C shift-c. Wijzig rest van regel*
- *D shift-d. Verwijder rest van regel*
- *s Vervang tekens*
- *S Vervang regels*
- *J Voeg regels samen*
- *J Voeg regels samen*

10.16 Kopiëren en Plakken

De ijank buffer"wordt met IEDERE delete opdracht gevuld, of expliciet door Y en yy.

- ***Y Kopieer de huidige regel naar de yank buffer***
- ***nyy Kopieer n regels te beginnen vanaf de huidige regel naar de yank buffer***
- ***p "Plak"de inhoud van de yank buffer na de cursor (of onder de huidige regel)***
- ***P "Plak"de yank buffer voor de cursor (of boven de huidige regel)***
- ***@p "Plak"vanuit buffer x***
- ***@y Yank (kopieer) naar buffer x***
- ***@d Verplaats naar buffer x***

10.17 Operators (gebruik dubbel om effect te hebben op regels)

- ***d delete (verwijder)***
- ***c change (wijzig)***
- ***< linker verschuiving***
- ***> rechter verschuiving***
- ***! filter door opdracht***
- ***= inspringing voor LISP***
- ***y yank (kopieer) tekst naar buffer***

10.18 Zoeken en vervangen

- ***/text Zoek voorwaarts naar text***
- ***?text Zoek terugwaarts naar text***
- ***n Herhaal de laatste zoekactie in dezelfde richting***
- ***N Herhaal de laatste zoekactie in tegengestelde richting***
- ***/ Herhaal de laatste zoekactie voorwaarts***
- ***? Herhaal de laatste zoekactie terugwaarts***
- ***[addr] s/van/naar/ [g] Zoek naar het voorkomen van van en vervang het in de huidige regel of in de range addr (twee regelnummers gescheiden door een komma; 1,$ is het gehele bestand) door naar. Vervang één voorkomen per regel, of alle voorkomendheden als g is gespecificeerd. Bijvoorbeeld, :3,20s/zoekwoord/anderwoord/g zal "zoekwoord"vervangen door änderwoord"te beginnen bij regel 3 tot aan regel 20. 'g' voor globaal betekent dat alle keren dat het "zoekwoord"voorkomt, het moet worden vervangen.***

10.19 Algemeen

- *:sh Roept een shell aan (kan worden beëindigd met CTRL-d)*
- *:!opdracht Roept een shell aan om de opdracht uit te voeren.*
- *:set number Schakel regelnummering in*
- *:set nonumber Schakel regelnummering uit*

10.20 Regeleditor-opdrachten

- *: Vertelt vi dat de volgende opdrachten die je uitvoert, regeleditor-opdrachten zullen zijn.*
- *:sh Ga tijdelijk naar de shell om een aantal shell-opdrachten uit te voeren zonder vi te verlaten.*
- *CTRL-d Beëindig de tijdelijke shell en keer terug naar vi zodat je het huidige venster kunt wijzigen.*
- *:n Ga naar de n-de regel in de buffer.*
- *:x,z w filename Schrijf regel vanaf het nummer x tot aan het nummer z naar het nieuwe bestand genaamd filename.*
- *:$ Verplaats de cursor naar het begin van de laatste regel in de buffer.*
- *:.,$d Verwijder alle regels vanaf de huidige regel tot aan de laatste regel.*
- *:r filename Voegt de inhoud van het bestand filename toe onder de huidige regel van de buffer.*
- *:s/text/new_text/ Vervangt het eerste voorkomen van text op de huidige regel door new_text*
- *:s/text/new_text/g Vervangt ieder voorkomen van text op de huidige regel door new_text*
- *:g/text/s//new_text/g Wijzigt ieder voorkomen van text in de buffer naar new_text.*

10.21 Andere opdrachten

- *u Maak de laatste wijziging ongedaan*
- *U Herstel de huidige regel*
- *~ Wijzig grote letter in kleine letter en vice versa*
- *J Voeg de huidige regel samen met de volgende regel*
- *. Herhaal laatste tekstwijzigingsopdracht*
- *CTRL-g Toon bestandsnaam en regelnummer*

11 Gerelateerde URL's

Aan VIM gerelateerde URL's zijn te vinden bij -

- ***C and C++ Beautifer*** <http://www.metalab.unc.edu/LDP/HOWTO/C-C++Beautifier-HOWTO.html>
- ***Linux goodies*** <http://www.aldev.8m.com> ***or bij*** <http://www.aldev.webjump.com>

12 Andere Formaten van dit Document

Dit document is in 10 verschillende formaten gepubliceerd, te weten - DVI, Postscript, Latex, LyX, GNU-info, HTML, RTF(Rich Text Format), Plain-text, Unix man pages en SGML.

- ***Je kunt dit HOWTO-document als een enkel tar-archief ophalen in HTML, DVI, Postscript of SGML-formaat vanaf -*** <ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/other-formats/> ***of*** <ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/other-formats/>
- ***Plain tekstformaat is in:*** <ftp://metalab.unc.edu/pub/Linux/docs/HOWTO> ***of*** <ftp://metalab.unc.edu/pub/Linux/docs/HOWTO>
- ***Vertalingen naar 't Frans, Duits, Spaans, Chinees, Japans staan in*** <ftp://metalab.unc.edu/pub/Linux/docs/HOWTO> ***of*** <ftp://metalab.unc.edu/pub/Linux/docs/HOWTO> ***Alle hulp bij het vertalen naar andere talen is welkom.***

Het document is geschreven door gebruik te maken van een tool genaamd "SGML tool"welke kan worden opgehaald vanaf - <http://www.xs4all.nl/~cg/sgmltools/> ***Bij het compileren van de source krijg je met de volgende opdrachten***

- ***sgml2html vim-howto.sgml (een html-bestand)***
- ***sgml2rtf vim-howto.sgml (een RTF-bestand)***
- ***sgml2latex vim-howto.sgml (een latex-bestand)***

Dit document is te vinden op -

- <http://metalab.unc.edu/LDP/HOWTO/Vim-HOWTO.html>

Je kunt dit document ook vinden op de volgende mirror-sites -

- <http://www.caldera.com/LDP/HOWTO/Vim-HOWTO.html>
- <http://www.WGS.com/LDP/HOWTO/Vim-HOWTO.html>
- <http://www.cc.gatech.edu/linux/LDP/HOWTO/Vim-HOWTO.html>
- <http://www.redhat.com/linux-info/ldp/HOWTO/Vim-HOWTO.html>
- ***Andere mirror-sites in de buurt (netwerk-adres-wijsheid) zijn te vinden op*** <http://metalab.unc.edu/LDP/hmirrors.html> ***selecteer een site en ga naar de directory /LDP/HOWTO/Vim-HOWTO.html***

Gebruik het xdvi-programma om dit document in het dvi-formaat te bekijken. Het xdvi-programma is te vinden in het tetex-xdvi*.rpm package in Redhat Linux wat kan worden gevonden via de ControlPanel | Applications | Publishing | TeX menu buttons.

```
Geef de volgende opdracht om het dvi-document te lezen -
        xdvi -geometry 80x90 howto.dvi
En pas de grootte van het scherm met de muis aan.
Zie de manpage over xdvi.
Gebruik de pijltjestoetsen, Page Up, Page Down keys, om door
het document te manoeuvreren, je kunt ook de
'f', 'd', 'u', 'c', 'l', 'r', 'p', 'n' letter
toetsen gebruiken om naar boven, onder, het midden, de volgende
pagina, vorige pagina, enz te gaan.
Druk op 'x' om het expert-menu uit te schakelen.
```

Je kunt het postscript-bestand lezen met behulp van het programma 'gv' (ghostview) of 'ghostscript'. Het ghostscript programma is te vinden in het ghostscript*.rpm package en het gv programma is te vinden in het gv*.rpm package in Redhat Linux welke kan worden gelokaliseerd via de ControlPanel | Applications | Graphics menu buttons. Het gv programma is veel gebruikersvriendelijker dan ghostscript. Ghostscript en gv zijn ook voor andere platformen beschikbaar zoals OS/2, Windows 95 en NT.

```
Geef de volgende opdracht om het postscript document te lezen -
        gv howto.ps

Met gebruik van ghostscript geef je op -
        ghostscript howto.ps
```

Je kunt het HTML-formaat document lezen met Netscape Navigator, Microsoft Internet explorer, Redhat Baron Web browser of enige andere webbrowsers.

Je kunt de latex, LyX uitvoer lezen met LyX een "X-Windows"front end naar latex.

13 Copyright Notice