

# being a wise guy

## L<sup>A</sup>T<sub>E</sub>X met één toets vanuit vi

Sven A. Bovin  
K.U.Leuven  
departement Scheikunde  
afdeling Kwantumchemie  
Celestijnenlaan 200F  
B-3001 Heverlee

### abstract

De ontstaansgeschiedenis van een shellscript om vanuit vi met één toets L<sup>A</sup>T<sub>E</sub>X op te roepen en, indien nodig, xdvi te starten voor het previewen.

### keywords

shell script, vi, L<sup>A</sup>T<sub>E</sub>X, Unix

Het begon allemaal met een tip op de tex-nl-discussielijst. Deze tip werd gepost door Hendri Hondorp en was afkomstig van de faculteit Informatica van de Universiteit Twente. Het is een manier om vanuit vi met één toetsaanslag L<sup>A</sup>T<sub>E</sub>X te laten lopen op je bestand en tegelijk je xdvi venster te laten verversen. Hiertoe wordt een executable script do\_latex aangemaakt met volgende inhoud:

```
#!/bin/sh
echo =====
echo q | latex $1 | tail -10
kill -USR1 `ps -a | grep xdvi | awk '{print $1}'`
```

Het script voert L<sup>A</sup>T<sub>E</sub>X uit en laat toont de laatste 10 regels van de output naar std out; indien nodig wordt L<sup>A</sup>T<sub>E</sub>X met q in batch-modus gezet na de eerste foutmelding. Om dit script uit te voeren vanuit vi, wordt een key mapping voorgesteld (in de .vimrc- of .virc-file): map l :w^M:!do\_latex %^M. Hierin staat ^M voor ctrl-M, in vi in te geven als Ctrl-V, Ctrl-M. Hierdoor krijgt de toetsaanslag l de betekenis 'sla het bestand op (w) en voer het externe commando do\_latex uit (het scriptje) met als parameter de naam van het huidige bestand'.

Wybo Dekker merkte op dat de toets pijl-rechts bij hem ook een l genereert. Dit hangt samen met het default gedrag van vi, waarin naar rechts bewogen wordt m.b.v. de toets l. Hij stelt daarom voor een andere toets te mappen. Wybo vraagt bovendien of het mogelijk is xdvi te starten indien het programma nog niet loopt.

Johan Wevers antwoordt hierop:

```
#!/bin/sh
echo =====
test `ps -ax | grep xdvi ` || xdvi $1
```

maar helaas is het niet zo simpel. ps -ax vangt ook xdvi's van andere gebruikers, en er moet bovendien gefilterd worden om het juiste proces te pakken te krijgen. Wybo stelt nu het volgende script voor:

```
#!/bin/sh
echo =====
file=${1%.tex}
echo q | latex $file | tail -10
pid=`ps x | grep xdvi | sed -e '/grep/d; s/^ *\[0-9\][0-9]*\).*\/\1/;q`
if [ x$pid == x ]; then
  xdvi $file&
else
  kill -USR1 $pid
fi
```

file=\${1%.tex}\$ slaat de eerste buffer van vi, dankzij de key mapping is dat de naam van het huidige bestand, op in file, s/^ \*\([0-9][0-9]\*\) .\*/\1/ isoleert het pid en de q in dezelfde regel zorgt ervoor dat je maar één xdvi ziet.

Bert Frederiks kan het nog mooier, en stelt

```
#!/bin/sh
echo =====
file=${1%.tex}
echo q | latex $file | tail -10
pid='ps x | grep " [x]dvi *$file" | awk '{print $1}'`
if [ -z "$pid" ]
then
    xdvi $file&
else
    kill -USR1 $pid
fi
```

Waarmee Wybo het gedeeltelijk eens is. Hij vindt echter

```
#!/bin/sh
echo =====
file=${1%.tex}
echo q | latex $file | tail -10
pid='ps x | grep " [x]dvi $file" `
if [ -z "$pid" ]
then
    xdvi $file &
else
    kill -USR1 ${pid:0:5}
fi
```

leuker.