

# Linux: Editors en Vim

D. Leeuw

1 april 2025

v.1.0.0



© 2020-2025 Dennis Leeuw

Dit werk is uitgegeven onder de Creative Commons BY-NC-SA Licentie en laat anderen toe het werk te kopiëren, distribueren, vertonen, op te voeren, en om afgeleid materiaal te maken, zolang de auteurs en uitgever worden vermeld als maker van het werk, het werk niet commercieel gebruikt wordt en afgeleide werken onder identieke voorwaarden worden verspreid.

# Over dit Document

## 0.1 Leerdoelen

Na het bestuderen van dit document heeft de lezer kennis van:

- wat een editor is
- de commando's: vi, nano, pico, vim
- de basis functionaliteit van vim: tekst invoer, knippen, plakken

Dit document sluit aan op de volgende onderdelen van de LPI:

- LPI Linux Essentials 010-160 - 6.3.3 Turning Commands into a Script (weight: 4)

## 0.2 Voorkennis

Voor een goed begrip van dit document is het van belang dat de lezer kennis heeft van:

- de shell



# Inhoudsopgave

<b>Over dit Document</b>	<b>i</b>
0.1 Leerdoelen . . . . .	i
0.2 Voorkennis . . . . .	i
<b>1 Het gebruik van een editor</b>	<b>1</b>
1.1 vi, pico, nano . . . . .	1
<b>2 vim</b>	<b>3</b>
2.1 vim opstarten . . . . .	3
2.2 Text invoer . . . . .	4
2.3 Text delete . . . . .	4
2.4 Knippen en plakken . . . . .	4
2.5 Bewegen door een tekstbestand . . . . .	5
<b>Index</b>	<b>7</b>



# Hoofdstuk 1

## Het gebruik van een editor

Op de commandline heb je geen menu's en vaak geen muis om door een applicatie te navigeren. Het maken van documenten is dan ook een stuk lastiger dan in een grafische interface. Toch zijn er oplossingen om op de commandline te werken met bestanden. Een tekstverwerker op de commandline heet een editor. Er zijn verschillende editors bedacht en in gebruik. Een van de oudste voor Unix geschreven editors is **vi**.

Een van de grote namen achter Unix is Ken Thompson. De eerste drie commando's die hij schreef voor het jonge Unix systeem waren **as** (assembler), **ed** (editor) en **sh** (shell). Dennis M. Ritchie bracht verbeteringen aan op **ed** en vanaf 1969 tot 1976 bleef dit de editor op een Unix systeem. In 1976 kwamen Billy Joy en Chuck Haley met een nieuwe editor die **ex** werd genoemd. Voor **ex** schreef Billy Joy ook een soort interface om er makkelijker mee te kunnen werken en die wrapper om **ex** noemde hij **vi** (visual interface). Vanaf 1979 werd **ex** geïntegreerd in **vi** en was er alleen nog **vi**. Later werd **vi** onderdeel van de Single Unix Specification en daarmee een editor die op bijna elk Unix systeem aanwezig is en dat is nog steeds het geval. Op bijna alle beschikbare Unix systemen, van BSD tot Linux en Mac OS X is een vorm van **vi** aanwezig. Dat is dan ook het voordeel van het aanleren van het werken met **vi**, ja kan je kennis op verschillende platformen gebruiken.

### 1.1 **vi, pico, nano**

De eerste redelijk gebruiksvriendelijke editor op Unix was **vi**. De **vi** editor kent twee modi. De eerste modus is de “edit mode” en de tweede is de “command mode”. Standaard start **vi** op in de command mode waarin je commando's kunt geven om bestanden te laden of op te slaan en waarin je functies als knippen en plakken kan uitvoeren. De edit modus is die waarin

je tekst invoert. Dit onderscheid maakt het voor beginnende gebruikers `vi` soms verwarrend.

Naast `vi` zijn er ook andere editors voor Unix-achtige systemen ontwikkeld. De meeste bekende zijn `pico` en `nano`. Pico was de oorspronkelijke editor. Nano is ontwikkeld door het GNU-project en is een vervanging van `pico` omdat `pico` een licentie had die “problematisch” was. Dat probleem is inmiddels opgelost, maar `nano` biedt zoveel extra mogelijkheden dat velen de voorkeur geven aan `nano`.

Het grote voordeel van `nano` ten opzichte van `vi` is zijn gebruiksvriendelijke interface. Nano kent geen edit en command mode zoals `vi`. Nano gebruikt control codes om commando's te geven en is direct beschikbaar voor de invoer van tekst van de gebruiker.



# Hoofdstuk 2

## vim

De naam `vim` staat voor Vi IMproved. Of wel een verbeterde versie van `vi`. Bram Molenaar, een Nederlandse software ontwikkelaar, schrijft al sinds 1991 aan de code van `vim` en zijn verbeteringen zijn zo populair dat op alle Linux systemen alleen nog `vim` geïnstalleerd wordt. Je kan op een Linux systeem `vim` opstarten als `vi` waarmee je zoveel mogelijk de functionaliteiten krijgt als het oude `vi` en je kan `vim` opstarten als `vim` waarmee je alle nieuwe toevoegingen van Bram Molenaar en zijn medeontwikkelaars krijgt.

`vim` is niet een van de makkelijkste editors maar heeft als grote voordeel, zoals eerder gezegd, dat het beschikbaar is op elk willekeurig Unix systeem.

### 2.1 vim opstarten

Het kan zijn dat `vim` nog niet geïnstalleerd is. Mocht dat het geval zijn, installeer dan `vim` via de package manager voor jouw systeem. Voor Debian systemen is dat:

```
$ sudo apt-get install vim
```

De meeste gebruikelijke manier om `vim` op te starten is door aan `vim` meteen een bestandsnaam mee te geven:

```
$ vim bestand.txt
```

Als je klaar bent met het toevoegen van tekst kan je met `:wq` afsluiten. Dit slaat het document op (w) en sluit af (q). Wil je de editor verlaten zonder de gemaakte wijzigingen op te slaan, dan gebruikt je `:q!`. Het doet een quit (q) zonder verdere vragen stellen.

Een andere manier om `vim` op te starten is door geen bestandsnaam mee te geven:

```
$ vim
```

de editor weet nu niet onder welke bestandsnaam een bestand opgeslagen moet worden. Bij de write (w) moet je nu dus de bestandsnaam meegeven: **:w bestand.txt** slaat het bestand dat je gemaakt hebt op als bestand.txt. Na deze opdracht kan je met **:q** vim afsluiten.

## 2.2 Text invoer

Om tekst toe te voegen of te wijzigen in vi moet je vanuit de command mode naar de edit mode gaan. Hiervoor zijn verschillende commando's beschikbaar. De meest gebruikte zijn **i** van insert of **a** van add. Om de edit mode te verlaten gebruik je de ESC toets.

Met het gebruik van het **i** commando voeg je tekst in vóór de plek van de cursor. Door gebruik te maken van **a** voeg je tekst in na de positie van de cursor.

## 2.3 Text delete

Vanuit de command modus kan je het character waarop je staat verwijderen door gebruik te maken van het **x** commando, eigenlijk is dit het knippen commando, maar kan gebruikt worden om snel wat characters te verwijderen. Het **D** commando verwijdert de text vanaf de plek waarop je staat tot het einde van de regel. Het einde van de regel is tot de plek waar je ENTER gegeven hebt, dus niet tot het einde van de regel op het scherm.

Het verwijderen van een complete regel van begin tot eind doe je met **dd**. Dit commando kan je uitbreiden met aantallen regels door tussen de d's een getal op te geven. Bijvoorbeeld **d2d** verwijdert 2 regels vanaf de plek waar je staat. En zo kan je ook met **dl** letters verwijderen. De combinatie **d5l** verwijdert vanaf de positie van de cursor 5 characters. Om hele woorden weg te gooien gebruik je **dw** en ook daar geldt de mogelijkheid om meerdere woorden in één keer weg te gooien met **d3w** gooi je 3 woorden achter elkaar weg.

## 2.4 Knippen en plakken

Met het **x** commando kan je één enkel character knippen.

De traditionele manier om een kopie van een stuk tekst te maken is het gebruik van het **y** commando. Het **yl** commando kopieert een character, **yw** kopieert een woord en **yy** kopieert regels.

Het **p** commando kan gebruikt worden om text te plakken. Het **p** commando is plakken achter de positie van de cursor en **P** is plakken voor de positie van de cursor.

Een speciale functie van vim en niet van vi is het gebruik van **v** om een visuele selectie maken, met de pijltjes toetsen kan je nu bepalen hoe groot de selectie worden moet. Het commando **v** geeft je de mogelijkheid om de selectie op character niveau te maken. Met **V** maak je selecties per regel, hier gebruik je de omhoog en omlaag pijltjes toetsen om je selectie groter of kleiner te maken. Als je de selectie gemaakt hebt gebruik je **y** om te kopieëren of **d** om het stuk te verwijderen.

Met **u** kan je een wijziging ongedaan maken.

## 2.5 Bewegen door een tekstbestand

Met de pijltjes toetsen kun je in de edit mode bewegen door een bestand. De pijltjes links en rechts bewegen de cursor een character per keer naar links of rechts, de op en neer pijltjes bewegen de cursor een regel op en neer.

In de command mode kan je met **^** bewegen naar het begin van een regel en met **\$** naar het einde van de regel. Een grotere sprong is met **gg** naar het begin van een document en met **G** naar het einde van het document.

Met **w** spring je een woord vooruit.

Je kan ook springen naar een bepaalde regel. Hiervoor gebruik je bijvoorbeeld **:10**. Dit commando sprint naar regel 10 van boven.

Met zoek opdrachten kan je ook door een document bewegen. Het **/** teken geeft het begin van een zoekopdracht aan. Dit werkt alleen in de command-modus. Met **/zoeken** zoek je naar het woord zoeken in de tekst. Met de letter **n** kan je naar het volgende zoekresultaat springen en met **N** naar het voorgaande.



# Index

- command mode, 4
- ed, 1
- edit mode, 4
  - verlaten, 4
- editor, 1
- ex, 1
- nano, 2
- pico, 2
- tekstverwerker, 1
- vi, 1
- Vi IMproved, 3
- vim, 3
  - add, 4
  - afsluiten zonder opslaan, 3
  - bestand schrijven, 4
  - copy, 5
  - cut character, 4
  - delete character, 4
  - delete een regel, 4
  - delete een word, 4
  - delete tot einde regel, 4
  - insert, 4
  - installeren, 3
  - opslaan, 3
  - opslaan en afsluiten, 3
  - paste, 5
  - starten met bestandsnaam, 3
  - starten zonder bestandsnaam, 3
  - visual selection, 5
- visual interface, 1