



# COMMANDLINE TIPS

## Job control

In jouw desktopomgeving open je gewoon meerdere terminalvensters om te multitasken op de commandline. Maar de shell bevat zelf ook functies om meerdere processen tegelijk te draaien. Op deze pagina leggen we je de basis van job control uit. **Filip Vervloesem**

### 1 OUTPUT

Soms wil je de output van een commando al bekijken, terwijl het nog aan het draaien is. Je hebt wellicht al gemerkt dat dat niet zo eenvoudig is. Net wanneer je in de vroegere output iets interessants gevonden hebt, komt er nieuwe output en verspringt je terminal weer helemaal naar het einde. In de meeste terminals kan je dat gedrag wel uitschakelen, maar dat doe je beter niet. Over het algemeen is het automatisch scrollen naar nieuwe output immers veel handiger dan manueel scrollen. De volgende truc biedt een eenvoudige oplossing: pauzeer tijdelijk de output van het commando met Ctrl-S. Je kan dan rustig de oudere output lezen en eventueel kopiëren. Ben je klaar, druk dan op Ctrl-Q om de output te hervatten.

### 2 SUSPEND

Sommige processen duren even om uit te voeren: denk aan iso-bestanden downloaden, afbeeldingen converteren of bestanden zoeken. Misschien wil je intussen wel verder werken in jouw terminal. Dat doe je door het actieve proces te suspenden met Ctrl-Z. Dus eerst start je bijvoorbeeld een download met curl. Via Ctrl-Z komt je opdrachtprompt weer vrij en start je convert. Je gebruikt nogmaals Ctrl-Z om ook dat commando te suspenden en het find-commando te starten.

### 3 VOOR- OF ACHTERGROND

In tip 2 zag je hoe je meerdere commando's tegelijk start in een terminal. Maar let op: via Ctrl-Z suspended commando's zijn in feite gepauzeerd en draaien dus niet verder! Wil je het commando laten doorwerken in de achtergrond? Met het jobs-commando toont bash je alle commando's die gestart waren in de huidige shell. In de eerste kolom zie je het id van de job, in de tweede de huidige status en in de derde het commando en de argumenten. Jobs die als Stopped gemarkeerd staan, zet je op de achtergrond weer in gang met de

### LISTING 1

```
$ tar czf archive.tar.gz * > /dev/null &
```

opdracht **bg %<job id>**, bijvoorbeeld **bg %1**. De status verandert dan in Running. Het is uiteraard mogelijk om meerdere jobs tegelijk in de achtergrond te draaien. Wil je één bepaalde job weer op de voorgrond draaien, omdat die bijvoorbeeld input verwacht? Gebruik dan **fg %<job id>** in plaats van **bg %<job id>**.

### 4 OUTPUT

Het nadeel van background jobs is dat hun output nog steeds in je terminal getoond wordt. Dat is soms lastig als je intussen wilt verder werken en er steeds andere output tussen jouw werk tevoorschijn komt. Background jobs start je het beste op met output redirection naar een bestand of naar /dev/null (als je er niet in geïnteresseerd bent). Errors toon je beter wél op het scherm, anders weet je niet of iets is misgelopen of niet. Jobs meteen opstarten in de achtergrond doe je door een & aan het einde van de commandline toe te voegen (zie **Listing 1**).

Zo hoeft je geen Ctrl-Z en bg uit te voeren na het commando.

### 5 JOB CONTROL

Soms wil je een proces voortijdig afsluiten, bijvoorbeeld omdat je het met de verkeerde opties hebt opgestart. Het actieve proces in je terminal breek je af met Ctrl-C of Ctrl-\ als Ctrl-C niet werkt. Lukt zelfs dat niet, dan suspend je het proces eerst en gebruik je vervolgens het bekende kill-commando, gevolgd door het job id. Probeer het proces te killen met verschillende signalen: eerst **kill -15** (SIGTERM, de standaardoptie) en als dat niet lukt **kill -9** (SIGKILL). Maak je veelvuldig gebruik van achtergrondprocessen? Let dan goed op wanneer je een terminal wilt sluiten! Zodra je een shell sessie wilt afsluiten met het logout-commando of Ctrl-D, geeft bash mogelijk een waarschuwing "There are stop-

ped jobs". Dat betekent dat er nog achtergrondjobs zijn met status Stopped. Bash zal die processen afsluiten zodra je de shell sluit. De jobs met status Running blijven wel in de achtergrond draaien, maar je kan ze niet meer naar de voorgrond brengen. Eventuele output van het commando ben je ook kwijt, tenzij je het had opgestart met output redirection naar een bestand.

### 6 NOHUP

Om zeker te zijn dat een proces blijft draaien in de achtergrond, gebruik je het nohup-commando. Dat is met name nodig als je via ssh inlogt op een ander systeem en daar een proces wilt starten dat blijft draaien nadat je weer uitlogt. Voeg gewoon nohup vóór je commando en een & achter het commando toe en het proces blijft gewoon draaien. De output vind je nadien terug in het bestand nohup.out in de directory waar je het commando gestart had. Nohup werkt enkel voor commando's die geen keyboard input verwachten. Wil je een interactief commando laten draaien op een server, dan start je het beter in een screen-sessie.