

Korte uitleg: Processen

Omdat we het veel over processen zullen hebben moeten we weten wat een proces in Linux eigenlijk is.

Een proces is een programma in uitvoering op de computer. Het zelfde programma (bijv. Kwrite) kan tegelijkertijd meermalen in uitvoering zijn. Voor dezelfde of voor verschillende gebruikers. Dat zijn allemaal verschillende processen.

Uiteraard regelt de Kernel alles rondom een proces. Een programma krijgt van de Kernel plek in het geheugen. Als het programma op die plek is ingelezen door de Kernel start de Kernel het proces op het startpunt. Het proces gaat dan "lopen" of "draaien", d.w.z. dat het een CPU gebruikt. De Kernel onderbreekt het programma af en toe zodat ook andere processen ook aan de beurt komen. De Kernel onthoudt dan waar het proces was gebleven.

De Kernel weet welke gebruiker onder welke groep het proces hebben gestart en dus eigenaar is van het proces.

Ieder proces heeft een nummer om het uniek te maken. Dit is de Process Identifier of PID.

Een proces wordt altijd gestart op verzoek (aan de Kernel) van een ander proces. Dat proces is de "ouder" of parent en zijn PID wordt in dat verband de PPID genoemd. De Kernel zelf heeft PID 0. Als je dit begrijpt, begrijp je ook dat alle processen in het systeem op een bepaald moment een boomstructuur vormen beginnend bij de Kernel (0), die overigens niet wordt vermeld, maar die boom start bij PID 1 in onderstaande commando's:

```
ps -ef
```

```
pstree
```

Een proces heeft een extra stukje geheugen met daarin de "process environment". Dat zijn dingen die door de Kernel aan het proces worden meegegeven. Het proces kan die lezen en sommige zaken ook wijzigen. Een deel van dit environment bevat de "environment variables" (zie [Korte uitleg: Environment variabelen \(zoals PATH, DISPLAY, LANG, ...\)](#)). De meeste daarvan worden door het ouder proces doorgegeven, een manier om van alles aan je kind te vertellen. Let op! omgekeerd werkt dit niet, een proces kan de omgeving van zijn ouder niet wijzigen. Om een idee te krijgen van de environment variables zoals ze nu in je shell bekend zijn:

```
env
```

Als een proces stopt wordt via Kernel een "return code" afgegeven aan het

ouder proces. De afspraak is dat return code 0 betekent dat alles "goed is gegaan". Alle andere waardes betekenen dat er iets niet helemaal goed was. Wat dat is staat als het goed is in de documentatie van het programma. Bijv. uit de `man` pagina van `ls`:

Exit status:

- 0 if OK,
- 1 if minor problems (e.g., cannot access subdirectory),
- 2 if serious trouble (e.g., cannot access command-line argument).