



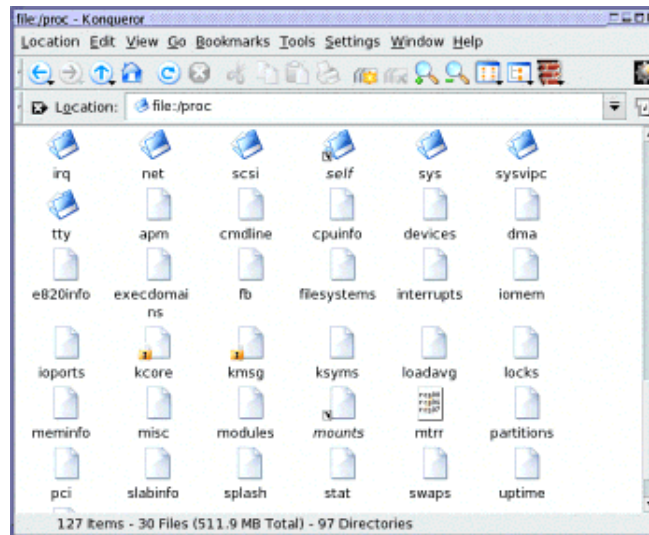
door Sandeep Grover
<sandeep<at>Magma-DA.com>

Over de auteur:

Sandeep Grover werkt voor Magma Design Automation in India, het snelst groeiende EDA (Electronic Design Automation) bedrijf. In zijn vrije tijd verkent hij de interne keuken van Linux -- ooit hoopt hij er zelf ook zijn steentje toe bij te dragen.

Vertaald naar het Nederlands door:
Floris Lambrechts
<nothanks.floris(at)linuxfocus.org>

Het Proc bestandssysteem leren begrijpen



Kort:

Met het /proc bestandssysteem kun je de onderliggende datastructuren van de Linux kernel bekijken en kernel parameters van een draaiend systeem veranderen. We zullen het /proc systeem beschrijven voor de Intel x86 architectuur, maar de achterliggende concepten zijn hetzelfde voor alle Linux kernels ongeacht het platform waarop ze draaien.

/proc - een Virtueel Bestandssysteem

Het /proc bestandssysteem (file system) is een mechanisme dat door de kernel en zijn modules gebruikt wordt om informatie aan te bieden aan processen (vandaar de naam /proc). Dit pseudo bestandssysteem laat je toe om de interne datastructuren van de kernel te beïnvloeden en om instellingen (kernel parameters) 'on the fly' te veranderen. In tegenstelling tot andere bestandssystemen, die op de schijf worden opgeslagen, bevindt /proc zich in het geheugen. Als je het bestand /proc/mounts bekijkt (dat alle gemounte bestandssystemen weergeeft - cfr. het commando 'mount'), dan zou je iets moeten zien zoals:

```
grep proc /proc/mounts
/proc /proc proc rw 0 0
```

/proc staat onder controle van de kernel en is niet verbonden met een fysiek apparaat (device). Omdat het vooral status-informatie bevat die beheerd wordt door de kernel, is het voor de hand liggend dat deze data in het RAM geheugen wordt bijgehouden. Een 'ls -l' opdracht op /proc toont dan ook aan dat de meeste bestanden 0 bytes groot zijn. Nochtans blijken ze ze veel informatie te bevatten als je ze geopend hebt. Hoe is dit mogelijk? Dit valt te verklaren doordat het /proc bestandssysteem, zoals elk ander bestandssysteem, zichzelf registreert bij de Virtual File System layer (VFS). Wanneer VFS aanvragen doet aan /proc dan vraagt het eigenlijk i-nodes op voor bestanden of directories. De truc is nu dat die i-nodes eigenlijk fysiek helemaal niet bestaan; het /proc systeem maakt ze dynamisch aan aan de hand van informatie uit de kernel.

Het proc File System mounten

Voor zover dat niet reeds gebeurd is, kun je het proc bestandssysteem mounten met de volgende opdracht

```
mount -t proc proc /proc
```

De bovenstaande opdracht zou een gemount proc systeem moeten opleveren. Voor meer informatie kan je de man page van mount nalezen.

De /proc bestanden bekijken

/proc bestanden kunnen je veel leren over de toestand van de kernel, de configuratie van de machine, de status van de draaiende processen, enzoverder. De meeste bestanden in /proc bieden een actuele blik op de fysieke toestand van een systeem. Alhoewel deze /proc bestanden virtueel zijn, kun je ze wel gewoon openen in een editor of in programma's zoals 'more', 'less' of 'cat'. Wanneer een programma een proc bestand probeert te openen, wordt dit bestand 'on the fly' aangemaakt met informatie uit de kernel. Hier volgen een paar interessante resultaten van op mijn eigen systeem:

```
$ ls -l /proc/cpuinfo -r--r--r-- 1 root root 0 Dec 25 11:01 /proc/cpuinfo
```

```
$ file /proc/cpuinfo
/proc/cpuinfo: empty
```

```
$ cat /proc/cpuinfo
```

```
processor      : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 8
model name    : Pentium III (Coppermine)
stepping      : 6
cpu MHz       : 1000.119
cache size    : 256 KB
```

```

fdiv_bug      : no
hlt_bug       : no
sep_bug       : no
f00f_bug     : no
coma_bug     : no
fpu           : yes
fpu_exception : yes
cpuid level   : 2
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 mmx fxsr xmm
bogomips      : 1998.85

processor     : 3
vendor_id    : GenuineIntel
cpu family   : 6
model        : 8
model name   : Pentium III (Coppermine)
stepping     : 6
cpu MHz      : 1000.119
cache size   : 256 KB
fdiv_bug     : no
hlt_bug      : no
sep_bug      : no
f00f_bug    : no
coma_bug    : no
fpu          : yes
fpu_exception : yes
cpuid level  : 2
wp           : yes
flags        : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 mmx fxsr xmm
bogomips     : 1992.29

```

Deze resultaten zijn afkomstig van een PC met twee processoren. De meeste variabelen spreken voor zich en geven nuttige informatie over het systeem. Sommige informatie in /proc bestanden is gecodeerd en er bestaan verschillende hulpmiddelen om deze om te zetten in een meer leesbare vorm. Enkele voorbeelden: 'top', 'ps', 'apm', ...

Nuttige systeem- en kernel informatie opvragen

Je kunt via het Proc File System informatie opvragen over het systeem en de kernel die erop draait. We vermelden enkele belangrijke bestanden.

- /proc/cpuinfo - informatie over de processor (CPU): model, familie, cache-grootte etc.
- /proc/meminfo - informatie over fysiek RAM-geheugen, swap space etc.
- /proc/mounts - lijst van gemounte bestandssystemen
- /proc/devices - lijst van beschikbare apparaten
- /proc/filesystems - ondersteunde bestandssystemen
- /proc/modules - lijst van ingeladen modules
- /proc/version - kernel versie
- /proc/cmdline - parameters gegeven aan de kernel tijdens het opstarten

Er zijn uiteraard nog veel meer bestanden te vinden in /proc. De nieuwsgierige lezer kan alle bestanden in /proc een keer bekijken, of [1] lezen voor meer informatie over de verschillende bestanden. Om wat rond te neuzen raad ik aan om 'more' te gebruiken in de plaats van 'cat' want sommige bestanden (bvb. kcore) kunnen zeer groot zijn.

Informatie over draaiende processen

Het /proc bestandssysteem bevat informatie over elk draaiend proces. In /proc zul je een aantal genummerde subdirectories terugvinden, die elk overeenkomen met een process id (PID) van een lopend programma. Van elk lopend proces kun je via /proc dus de belangrijkste kenmerken opvragen. Voor een voorbeeld gaan we eerst op zoek naar een lopend proces:

```
$ ps -aef | grep mozilla
root 32558 32425 8 22:53 pts/1 00:01:23 /usr/bin/mozilla
```

Het bovenstaande leert ons dat mozilla een draaiend proces is met een PID van 32558. In /proc zouden we dus een directory moeten vinden met het nummer 32558.

```
$ ls -l /proc/32558
total 0
-r--r--r-- 1 root root 0 Dec 25 22:59 cmdline
-r--r--r-- 1 root root 0 Dec 25 22:59 cpu
lrwxrwxrwx 1 root root 0 Dec 25 22:59 cwd -> /proc/
-r----- 1 root root 0 Dec 25 22:59 environ
lrwxrwxrwx 1 root root 0 Dec 25 22:59 exe -> /usr/bin/mozilla*
dr-x----- 2 root root 0 Dec 25 22:59 fd/
-r--r--r-- 1 root root 0 Dec 25 22:59 maps
-rw----- 1 root root 0 Dec 25 22:59 mem
-r--r--r-- 1 root root 0 Dec 25 22:59 mounts
lrwxrwxrwx 1 root root 0 Dec 25 22:59 root -> //
-r--r--r-- 1 root root 0 Dec 25 22:59 stat
-r--r--r-- 1 root root 0 Dec 25 22:59 statm
-r--r--r-- 1 root root 0 Dec 25 22:59 status
```

Het bestand "cmdline" bevat het commando waarmee het proces opgestart werd. Het bestand "environ" bevat de omgevingsvariabelen voor dit programma. "status" heeft status-informatie over het proces, waaronder de user (UID) en group (GID) identificatie van de gebruiker met wiens rechten het proces draait. Ook het parent process ID (PPID; het programma dat het huidige programma heeft opgestart) en de huidige toestand van het proces, zoals "Sleeping" of "Running", vind je hierin terug. Daarnaast bevat elke directory ook nog een paar symbolische links. "cwd" verwijst naar de huidige werk-directory van het proces, "exe" naar het uitvoerbare programma-bestand van het proces. "root" is een link naar de directory die door het proces gezien wordt als de root directory (meestal "/"). De map "fd" bevat links naar de file descriptors die het proces in gebruik heeft. De "cpu" link verschijnt enkel op SMP Linux kernels voor systemen met meerder processoren. Het bevat een opdeling van verlopen proces-tijd verdeeld per processor.

/proc/self is een interessante subdirectory: hierin kunnen processen informatie opzoeken over zichzelf. Dit wordt bereikt doordat /proc/self is een symbolische link is die verwijst naar de /proc directory met het PID van het proces dat de link volgt.

De kernel beïnvloeden via /proc

De meeste hierboven vermelde /proc bestanden zijn read-only: je kunt ze niet veranderen. Er zijn echter /proc bestanden die dat wel toelaten. Door te schrijven naar deze bestanden kun je parameters in de kernel veranderen - onnodig te zeggen dat voorzichtigheid hier geboden is. Alle schrijfbaar bestanden bevinden zich in /proc/sys.

/proc/sys/kernel - Deze directory bevat informatie over het gedrag van de kernel.
/proc/sys/kernel/{domainname, hostname} bevat de domein- en hostnaam voor je machine of netwerk.
Hier kun je deze namen dus aanpassen.

```
$ hostname
machinenaam.domeinnaam.com

$ cat /proc/sys/kernel/domainname
domeinnaam.com

$ cat /proc/sys/kernel/hostname
machinenaam

$ echo "nieuwe-machinenaam" > /proc/sys/kernel/hostname

$ hostname
nieuwe-machinenaam.domeinnaam.com
```

Dus, door een bestand aan te passen in /proc, veranderen we de hostname van het systeem. Er zijn nog vele andere bestanden in /proc/sys/kernel/ die je kunt aanpassen. Ook hier weer kunnen we ze niet allemaal behandelen, dus het kan interessant zijn om ze zelf eens te bekijken.

Een andere configureerbare directory is /proc/sys/net. Via de bestanden daarin kun je de netwerkeigenschappen aanpassen. Je kunt er bijvoorbeeld heel snel voor zorgen dat je machine 'onzichtbaar' wordt in het netwerk.

```
$ echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

Hiermee wordt je machine een beetje minder zichtbaar doordat hij niet meer zal antwoorden op icmp_echo aanvragen. Met andere woorden: je machine zal niet meer reageren op ping queries van andere machines.

```
$ ping machinenaam.domeinnaam.com
no answer from machinenaam.domeinnaam.com
```

Om de oude situatie te herstellen doe je

```
$ echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

Via [1] en [2] kun je kennis maken met de vele andere subdirectories in /proc/sys waar je aanpassingen kan doen.

Conclusie

Het /proc File System biedt een blik op de interne keuken van Linux via virtuele bestanden. Het kan een hulp zijn bij het onderzoeken van de status en configuratie van de verschillende apparaten en processen op een systeem. Kennis van en praktische ervaring met dit bestandssysteem is daarom van groot belang om je Linux systeem optimaal te kunnen gebruiken.

Bibliografie

- [1] Documentatie over het Linux proc bestandssysteem is beschikbaar op:
</usr/src/linux/Documentation/filesystems/proc.txt>
- [2] Red Hat Guide: The /proc File System:
<http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/ref-guide/ch-proc.html>

Site onderhouden door het LinuxFocus editors team	Vertaling info:
--	-----------------

© Sandeep Grover "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org	en --> -- : Sandeep Grover <sandeep<at>Magma-DA.com> en --> nl: Floris Lambrechts <nothanks.floris(at)linuxfocus.org>
---	---