

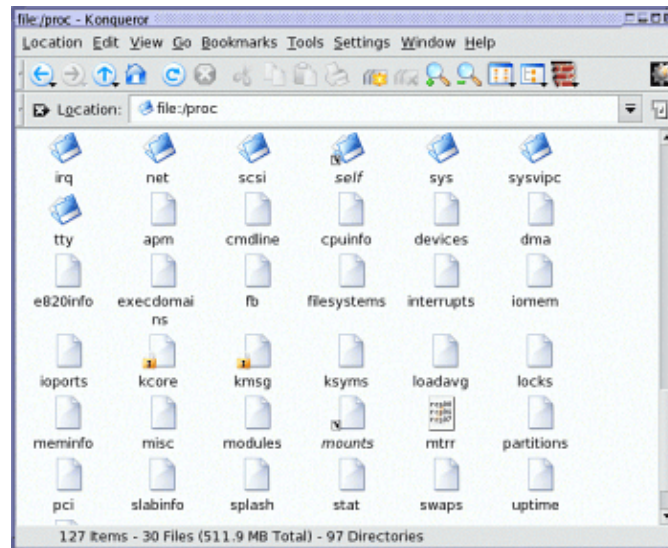
Das Proc File System verstehen



by Sandeep Grover
 <sandeep@at>Magma-DA.com

About the author:

Sandeep Grover arbeitet für Magma Design Automation, India – die am schnellsten wachsende EDA (Electronic Design Automation)–Firma. In seiner Freizeit beschäftigt er sich mit den Internas von Linux, in der Hoffnung, eines Tages zum Linuxkernel beitragen zu können!



Abstract:

Der Linuxkernel enthält einen Mechanismus, um Zugang zu seinen internen Datenstrukturen möglich zu machen und um Kerneleinstellungen im laufenden Betrieb über das /proc–Dateisystem zu ändern. Wir werden hier das /proc–Dateisystem in Bezug auf die Intel x86–Architektur behandeln – das zugrunde liegende Konzept für Linux ist jedoch auf allen Plattformen gleich.

/proc – ein virtuelles Dateisystem

Das /proc–Dateisystem ist ein Mechanismus, um Informationen an die Prozesse des Kernels und die der Kernelmodule zu schicken (daher der Name /proc). Dieses Pseudo–Dateisystem erlaubt uns, auf die interne Datenstruktur des Kernels einzuwirken, nützliche Informationen über die Prozesse herauszufinden und um Einstellungen während des Betriebs zu ändern (durch Änderung der Kernelparameter). /proc ist im Hauptspeicher untergebracht, während die anderen Dateisysteme auf der Festplatte gespeichert sind. Betrachten wir die Datei /proc/mounts (die alle gemounteten Dateisysteme auflistet, wie z.B. den "mount"–Befehl), sollten wir eine Zeile finden, die etwa so aussieht:

```
grep proc /proc/mounts
```

```
/proc /proc proc rw 0 0
```

/proc wird durch den Kernel gehandhabt und hat keine eigenes Device. Da die Datei hauptsächlich vom Kernel gesteuerte Zustandsinformationen enthält, ist es logisch, diese im Hauptspeicher unterzubringen, der vom Kernel gehandhabt wird. Der Befehl 'ls-l' – in Bezug auf /proc – zeigt uns, dass die meisten Dateien eine Größe von 0 Byte haben. Schauen wir uns die Datei jedoch näher an, finden wir eine ganze Menge Informationen. Wie ist das möglich? Das geschieht, weil sich das /proc-Dateisystem – wie alle normalen Dateisysteme – in der Virtual File System-Schicht (VFS) registriert. Bei Aufrufen des VFS auf I-nodes für Dateien/Verzeichnisse generiert das /proc-Dateisystem diese Dateien/Verzeichnisse dynamisch aus Informationen vom Kernel. Diese Verzeichnisstruktur entsteht also erst wenn man darauf zugreift.

Das proc-Dateisystem mounten

Falls dieses noch nicht in Ihrem System gemountet ist, kann das durch folgenden Befehl geschehen –

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

Mit diesem Befehl sollten Sie Ihr proc-Dateisystem erfolgreich gemountet haben. Nähere Informationen sind in der mount-Manpage zu finden.

Anschaun der /proc-Dateien

Durch die /proc-Dateien erhalten wir Zugang zu Informationen über den Zustand des Kernels, Attribute des Rechners, den Status der laufenden Prozesse, usw. Die meisten Dateien im /proc-Verzeichnis liefern uns einen aktuellen Einblick in die technische Umgebung des Systems. Obwohl die /proc-Dateien virtuell sind, können sie mittels Dateieditor oder durch Programme – wie 'more', 'less' oder 'cat' – eingesehen werden. Für jeden Editor, der eine virtuelle Datei öffnen will, wird diese umgehend aus aktuellen Informationen des Kernels erzeugt. Hier sind ein paar interessante Ergebnisse über mein System

```
$ 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

```

Das ist das Ergebnis für eine Zwei-CPU-Maschine. Die meisten der Informationen bedürfen keiner Erklärung, sie liefern nützliche Hardwareinformationen über dieses System. Einige der Angaben der /proc-Dateien sind kodiert, es bestehen verschiedene Utilities, welche die kodierten Informationen interpretieren und in lesbarer Form ausgeben. Einige der Utilities sind: top', 'ps', 'apm', usw.

Brauchbare Informationen über den Kernel zu erzeugen

Das Proc-File-System kann nützliche Angaben über das System und den laufenden Kernel sammeln. Einige der wichtigen Dateien finden wir in der folgenden Liste

- /proc/cpuinfo – Informationen über die CPU (Model, Familie, Cachegrösse, usw.)
- /proc/meminfo – Informationen über vorhandenes RAM, Swap space, usw.
- /proc/mounts – erzeugt eine Liste der gemounteten Dateisysteme
- /proc/devices – erzeugt eine Liste der vorhandenen Geräte
- /proc/filesystems – unterstützte Dateisysteme
- /proc/modules – Liste der geladenen Module
- /proc/version – Kernelversion
- /proc/cmdline – Parameters, die beim Start an den Kernel übergeben werden

/proc enthält viel mehr Dateien als die vorgehende Liste. Der aufmerksame Leser sollte 'more' für jede Datei im /proc-Verzeichnis durchführen oder in [1] für mehr Informationen zu den Dateien im /proc-Verzeichnis nachlesen. Ich empfehle 'more' anstelle von 'cat' zu benutzen, bis Sie das Dateisystem etwas genauer kennen, denn einige Dateien (z.B. kcore) können sehr umfangreich sein.

Informationen über laufende Prozesse

Mit dem `/proc`-Dateisystem kann man Informationen über jeden laufenden Prozess aufspüren. Es gibt einige nummerierte Unterverzeichnisse im `/proc`. Jedes nummerierte Unterverzeichnis bezieht sich auf eine Prozess-ID (PID). Das bedeutet, für jeden laufenden Prozess gibt es ein Unterverzeichnis innerhalb von `/proc`, bezeichnet mit der zugehörigen PID. Innerhalb dieser Unterverzeichnisse finden wir Dateien mit wichtigen Details über den Status und die Umgebung der Prozesse. Probieren wir die Suche nach einem laufenden Prozess.

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

Der obige Befehl deutet auf einen laufenden Prozess von Mozilla mit der PID 32558. Infolgedessen sollte in `/proc` ein Unterverzeichnis mit der Nummer 32558 zu finden sein.

```
$ 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
```

Die Datei "cmdline" enthält den Befehl, der aufgerufen wird, um den Prozess zu starten. Die "environ"-Datei enthält die Umgebungsvariablen für den Prozess. "status" enthält die Statusinformationen des Prozesses, einschliesslich Benutzer (UID)- und Gruppen (GID)-Identifikation des Benutzers, der den Prozess ausführen lässt, sowie die Parentprozess-ID (PPID) – welche die PID instanziierte – und den gegenwärtigen Status des Prozesses, wie "Sleeping" oder "Running". Jedes Prozessverzeichnis hat einige symbolische Links. "cwd" ist ein Link zum aktuellen aktiven Verzeichnis, "exe" zur ausführbaren Datei des aktiven Prozesses, "root" ist ein Link zum Verzeichnis, welches der Prozess als sein root-Verzeichnis erkennt (meistens "/"). Das Verzeichnis "fd" enthält Links zu den Dateideskriptoren, die der Prozess benutzt. Der "cpu"-Eintrag erscheint nur bei SMP-Linuxkerneln. Er enthält Angaben über die Dauer des Prozesses für jedes CPU.

`/proc/self` ist ein interessantes Unterverzeichnis, es ermöglicht einem Programm Informationen über seinen eigenen Prozess zu finden. Die Angabe `/proc/self` ist ein symbolischer Link zum `/proc`-Verzeichnis – er bezieht sich auf den Prozess der gerade das `/proc`-Verzeichnisses liest.

Interaktion mit dem Kernel durch `/proc`

Die meisten oben genannten Dateien in `/proc` sind schreibgeschützt. Das `/proc`-Dateisystem ermöglicht jedoch durch Lese-/Schreibdateien – innerhalb von `/proc` – auf den Kernel einzuwirken. In diese Dateien schreiben kann den Zustand des Kernels ändern, deshalb sollten Änderungen dieser Dateien mit Vorsicht durchgeführt

werden. Das `/proc/sys`-Verzeichnis enthält Lese-/Schreib-Dateien, es kann daher benutzt werden, um das Verhalten des Kernel zu ändern.

`/proc/sys/kernel` – Diese Verzeichnis enthält Informationen zum allgemeinen Kernelverhalten.
`/proc/sys/kernel/{ domainname, hostname }` enthält den Domainnamen und den Hostnamen der Maschine/ des Netzwerks. Diese Dateien können für die Änderung der Namen benutzt werden.

```
$ hostname
machinename.domainname.com

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

$ cat /proc/sys/kernel/hostname
machinename

$ echo "new-machinename" > /proc/sys/kernel/hostname

$ hostname
new-machinename.domainname.com
```

Wie gesagt durch Änderung der Dateien innerhalb des `/proc`-Dateisystems können wir den Hostnamen wechseln. Viele andere konfigurierbare Dateien sind im `/proc/sys/kernel` zu finden. Es ist jedoch nicht möglich alle diese Dateien hier aufzuführen, die Leser können die Details dieses Verzeichnisses einfach einsehen.

Eine weitere konfigurierbare Datei ist `/proc/sys/net`. Dateien innerhalb dieses Verzeichnisses können für Änderungen von Netzwerkeigenschaften von Maschine/Netzwerk benutzt werden. Ein Beispiel: einfach durch Änderung einer Datei können Sie Ihre Maschine im Netzwerk unsichtbar machen.

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

Das versteckt Ihre Maschine im Netzwerk, indem es die Antwort auf `icmp_echos` verweigert. Der Host wird nicht auf ping-Aufrufe von anderen Hosts antworten.

```
$ ping machinename.domainname.com
no answer from machinename.domainname.com
```

Um zur Grundeinstellung zurückzugelangen, geben wir ein

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

Es gibt viele andere Unterverzeichnisse im `/proc/sys`, in denen durch Konfiguration die Eigenschaften des Kernels geändert werden können. Siehe [1], [2] für weitere Angaben.

Zusammenfassung

Das `/proc`-Dateisystem bietet eine Schnittstelle zum Linuxkernel. Es hilft, den Status und die Konfigurationen vieler Einrichtungen und Prozesse Ihres Systems zu ermitteln. Das Verständnis und das angewandte Wissen dieses Dateisystems ist deshalb der Schlüssel, um das Beste aus Ihrem Linuxsystem zu machen.

Literaturverzeichnis

- [1] Dokumentation über das Linux-proc-Dateisystem kann hier gefunden werden:
</usr/src/linux/Documentation/filesystems/proc.txt>
- [2] RedHat Guide: The /proc File System:
<http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/ref-guide/ch-proc.html>

<p><u>Webpages maintained by the LinuxFocus Editor team</u></p>	<p>Translation information: en --> -- : Sandeep Grover <sandeep<at>Magma-DA.com> en --> de: Jürgen Pohl <sept.sapins(at)verizon.net></p>
<p>© Sandeep Grover</p>	
<p>"some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	

2005-01-11, generated by lfparsr_pdf version 2.51