

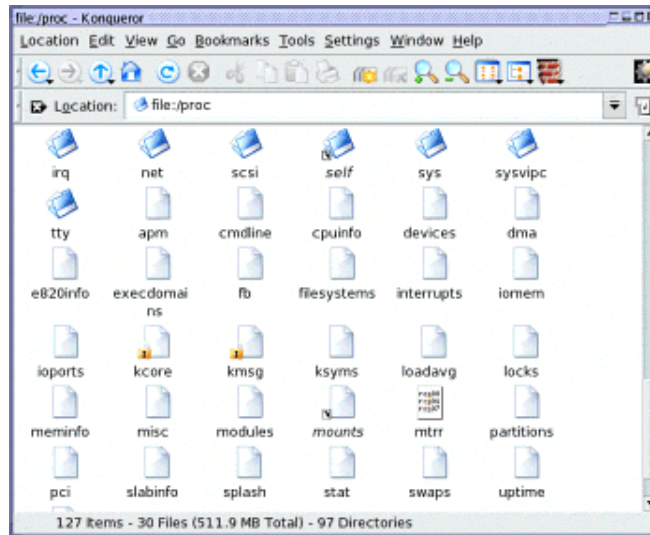
## Understanding the Proc File System



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

### *About the author:*

Sandeep Grover works for Magma Design Automation, India -- the fastest growing EDA (Electronic Design Automation) company. In his free time, he explores the internals of Linux and some day hopes to contribute to the Linux Kernel!



### *Abstract:*

The Linux kernel provides a mechanism to access its underlying internal data-structures and also to change its kernel settings at run-time through /proc file system. We will be discussing the /proc file system here targeted to the Intel x86 architecture; though the basic concepts will remain the same for Linux on any platform.

---

## /proc - a Virtual File System

The /proc file system is a mechanism that is used for the kernel and kernel modules to send information to processes (hence the name /proc). This pseudo file system allows you to interact with the internal data-structure of the kernel, get useful information about the processes, and to change settings (by modifying the kernel parameters) on the fly. /proc is stored in memory, unlike other file-systems, which are stored on disk. If you look at the file /proc/mounts (which lists all the mounted file systems, like "mount" command), you should see a line in it like:

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

/proc is controlled by the kernel and does not have an underlying device. Because it contains mainly state information controlled by the kernel, the most logical place to store the information is in memory controlled by the kernel. Doing a 'ls -l' on /proc reveals that most of the files are 0 bytes in size; Yet when the file is viewed, quite a bit of information is seen. How is this possible? This happens because the /proc file-system, like any other regular file-system registers itself to the Virtual File System layer (VFS). However, when VFS make calls to it requesting i-nodes for files/directories, the /proc file system creates those files/directories from information within the kernel.

## Mounting the proc File System

If already not mounted on your system, proc file system can be mounted on your system by running the following command -

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

The above command should successfully mount your proc file system. Please read the mount man page for more details.

## Viewing the /proc files

/proc files can be used to access information about the state of the kernel, the attributes of the machine, the state of the running processes etc. Most of the files in the /proc directory provide the latest glimpse of a system's physical environment. Although these /proc files are virtual, yet they can be viewed using any file editor or programs like 'more', 'less' or 'cat'. When any editor program tries to open a virtual file, the file is created on the fly from information within the kernel. Here are some interesting results which I got on my 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

```

This is the result for a two-CPU machine. Most of the information above is self-explanatory and gives useful hardware information about the system. Some of the information in /proc files is encoded and various utilities are built that interpret this encoded information and output it in a human readable format. Some of these utilities are: 'top', 'ps', 'apm' etc.

## Getting useful system/kernel information

The Proc File System can be used to gather useful information about the system and the running kernel. Some of the important files are listed below

- /proc/cpuinfo - information about the CPU (model, family, cache size etc.)
- /proc/meminfo - information about the physical RAM, Swap space etc.
- /proc/mounts - list of mounted file systems
- /proc/devices - list of available devices
- /proc/filesystems - supported file systems
- /proc/modules - list of loaded modules
- /proc/version - Kernel version
- /proc/cmdline - parameters passed to the kernel at the time of starting

There are much more files in /proc than listed above. An alert reader is expected to do a 'more' on every file in /proc directory or read [1] for more information about the files present in /proc directory. I

suggest to use 'more' and not 'cat' until you know the filesystem a bit because some files (e.g kcore) can be very large.

## Information about running processes

The /proc file system can be used to retrieve information about any running process. There are couple of numbered sub-directories inside /proc. Each numbered directory corresponds to a process id (PID). Thus, for each running process, there is a sub-directory inside /proc named by its PID. Inside these sub-directories are files that provide important details about the state and environment of a process. Lets try to search for a running process.

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

The above command shows that there is a running process of mozilla with PID 32558. Correspondingly, there should be a directory in /proc with number 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
```

The file "cmdline" contains the command invoked to start the process. The "environ" file contains the environment variables for the process. "status" has status information on the process, including the user (UID) and group (GID) identification for the user executing the process, the parent process ID (PPID) that instantiated the PID, and the current state of the process, such as "Sleeping" or "Running." Each process directory also has a couple of symbolic links. "cwd" is a link to the current working directory for the process, "exe" to the executable program of the running process, "root" is a link to the directory, which the process sees as its root directory (usually "/"). The directory "fd" contains links to the file descriptors that the process is using. "cpu" entry appears only on SMP Linux kernels. It contains a breakdown of process time by CPU.

/proc/self is an interesting sub-directory that makes it easy for a program to use /proc to find information about its own process. The entry /proc/self is a symbolic link to the /proc directory corresponding to the process accessing the /proc directory.

# Interacting with Kernel via /proc

Most of the files in /proc discussed above are read-only. However, the /proc file system provides provision to interact with kernel via read-write files inside /proc. Writing to these files can change the state of the kernel and therefore changes to these files should be made with caution. The /proc/sys directory is the one that hosts all the read-write files and thus can be used to change the kernel behavior.

`/proc/sys/kernel` - This directory contains information that reflects general kernel behavior. `/proc/sys/kernel/{domainname, hostname}` holds the domain-name and hostname for the machine/network. These files can be configured to modify these names.

```
$ 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
```

Thus, by modifying the file inside /proc file system, we are able to modify the hostname. Lots of other configurable files exists inside /proc/sys/kernel/. Again, its impossible to list down every file here, so readers are expected to go through this directory in detail.

Another configurable directory is `/proc/sys/net`. Files inside this directory can be modified to change the networking properties of the machine/network. E.g. By simply modifying a file, you can hide your machine in the network.

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

This will hide your machine in the network as it disables answers to icmp\_echos. The host will not respond to ping queries from other hosts.

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

To turn it back to default behavior, do

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

There are lots of other sub-directories in /proc/sys which can be configured to change the kernel properties. See [1], [2] for detailed information.

## Conclusion

The /proc File System provides a file-based interface to the Linux internals. It assists in determining the state and configuration of various devices and processes on a system. Understanding and applied knowledge of this file-system is therefore the key to making the most out of your Linux system.

## Bibliography

- [1] Documentation about the Linux proc filesystem can be found at:  
</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>

---

Webpages maintained by the LinuxFocus Editor team © Sandeep Grover "some rights reserved" see <a href="http://linuxfocus.org/license/">linuxfocus.org/license/</a> <a href="http://www.LinuxFocus.org">http://www.LinuxFocus.org</a>	Translation information: en --> -- : Sandeep Grover <sandeep<at>Magma-DA.com>
--	--