

SO WHAT IS /PROC ANYWAYS?

I've been using Linux for years, but I've never really known what `/proc` was or why certain commands used it. I'm not sure why I've never looked it up in the past, but I recently did and I thought I'd share.

`/proc` isn't a "real" directory, in the sense that it doesn't exist on disk. It's not an in memory file system either. It's a **virtual file system** that exposes information from the Linux kernel. `/proc` uses the `procfs` file system driver and is mounted to `/proc` at boot time. A lot of system utilities are simply wrappers around reading files from `/proc`.

For example, run the following command on your Linux machine (it may look familiar):

```
cat /proc/cpuinfo
```

The above command should spit out all sorts of information about your CPU. Very useful.

`/proc` contains all sorts of info, so I recommend looking around. `/proc/meminfo` contains more info about the kernel's memory usage than you even knew existed. `/proc/cmdline` shows the options that Grub used to boot the kernel.

Process Information

I find the process specific information very useful on a day-to-day basis. Process information is available under `/proc/<pid>`.

For example, `/proc/<pid>/limits` will show you the ulimits of that process as they are right now.

`/proc/<pid>/oom_adj` is a writable file that can adjust priorities for the out-of-memory killer, to prevent your process from being killed.

`/proc/<pid>/status` is the human readable status of that process.

Writing to `/proc`

You can also write to some proc files to change configuration options in the Linux kernel. For example, you can manipulate swappiness via `/proc/sys/vm/swappiness`.

Next time you see a command that redirects output to `/proc`, you'll know it's changing a kernel configuration option.

Just remember, `/proc` represents the current kernel configuration, and changes will not persist through a reboot.

Useful files in `/proc`

There are many useful files in `/proc`, so I recommend just looking around.

However, the following are among the more useful.

`/proc/cmdline`

Kernel command line arguments

`/proc/cpuinfo`

CPU information such as make, model, cores, etc

`/proc/devices`

List of devices (block/character devices)

`/proc/filesystems`

List of supported filesystem drivers

`/proc/meminfo`

Information about memory usage, both physical and swap

`/proc/modules`

Kernel modules that are currently loaded

`/proc/mounts`

List of mounted filesystems, mount points, and mount arguments

`/proc/swaps`

Swap space devices and utilization

`/proc/sys`

Lots of kernel parameters, most of which are writable to allow changes to kernel parameters without rebooting/recompiling

`/proc/version`

Kernel version

A great source of information is the Linux Documentation Project.

Source: <http://brandonwamboldt.ca/so-what-is-proc-anyways-1510/>