

Show Process Limits Using /proc Filesystem

```
root@redhat:/ # cat /proc/5130/limits
Limit                Soft Limit           Hard Limit           Units
Max cpu time         unlimited            unlimited            seconds
Max file size        unlimited            unlimited            bytes
Max data size        unlimited            unlimited            bytes
Max stack size       8388608             unlimited            bytes
Max core file size   unlimited            unlimited            bytes
Max resident set     unlimited            unlimited            bytes
Max processes        127372              127372              processes
Max open files       1024                4096                files
Max locked memory    16777216            16777216            bytes
Max address space    unlimited            unlimited            bytes
Max file locks       unlimited            unlimited            locks
Max pending signals  127372              127372              signals
Max msgqueue size    819200              819200              bytes
Max nice priority    0                   0
Max realtime priority 0                   0
Max realtime timeout unlimited            unlimited            us
root@redhat:/ # █
```

Show process limits with /proc filesystem

I think I mentioned the special **/proc filesystem** before, it's available in Linux distros and helps you obtain system and process information via normal files created in a special structure. Today I'd like to show you another cool trick **/proc** has.

Show Process Info Using /proc

Just to remind you, here's what I mean: on my Red Hat PC I have this **sshd** daemon process running:

```
root@redhat:/ # ps -aef | grep [o]penssh
root          5130          1  0  Oct03  ?        00:00:00 /usr/sbin/sshd
```

```

-D -oCiphers=aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-ctr,aes256-cbc,aes128-gcm@openssh.com,aes128-ctr,aes128-cbc -oMACs=hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha1,umac-128@openssh.com,hmac-sha2-512 -oGSSAPIKexAlgorithms=gss-gex-sha1-,gss-group14-sha1- -oKexAlgorithms=curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group14-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1 -oHostKeyAlgorithms=rsa-sha2-256,ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384,ecdsa-sha2-nistp384-cert-v01@openssh.com,rsa-sha2-512,ecdsa-sha2-nistp521,ecdsa-sha2-nistp521-cert-v01@openssh.com,ssh-ed25519,ssh-ed25519-cert-v01@openssh.com,ssh-rsa,ssh-rsa-cert-v01@openssh.com -oPubkeyAcceptedKeyTypes=rsa-sha2-256,ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384,ecdsa-sha2-nistp384-cert-v01@openssh.com,rsa-sha2-512,ecdsa-sha2-nistp521,ecdsa-sha2-nistp521-cert-v01@openssh.com,ssh-ed25519,ssh-ed25519-cert-v01@openssh.com,ssh-rsa,ssh-rsa-cert-v01@openssh.com

```

So the **sshd** process ID (PID) is 5130. That means I can use **/proc** filesystem to learn quite a bit about the process:

```

root@redhat:/ # cd /proc/5130
root@redhat:/proc/5130 # ls
attr          cmdline      environ      io           mem
ns            pagemap     sched        smaps_rollup syscall
wchan
autogroup     comm         exe          limits       mountinfo
numa_maps     patch_state schedstat    stack        task
auxv          coredump_filter fd           loginuid     mounts
oom_adj       personality  sessionid    stat         timers
cgroup        cpuset      fdinfo       map_files    mountstats
oom_score     projid_map  setgroups    statm
timerslack_ns

```

```
clear_refs  cwd          gid_map  maps      net
oom_score_adj  root          smaps    status    uid_map
```

Each file or directory in this /proc/5130 location shows some information specific to this PID 5130.

For instance, if we list files in the **fd** directory there, we'll see all the files and sockets open by **sshd** at the moment:

```
root@redhat:/proc/5130 # ls -al fd/*
lr-x-----. 1 root root 64 Oct  3 14:10 fd/0 -> /dev/null
lrwx-----. 1 root root 64 Oct  3 14:10 fd/1 ->
'socket:[39555] '
lrwx-----. 1 root root 64 Oct  3 14:10 fd/2 ->
'socket:[39555] '
lr-x-----. 1 root root 64 Oct  3 14:10 fd/3 -> /dev/urandom
lr-x-----. 1 root root 64 Oct  3 14:10 fd/4 ->
/var/lib/sss/mc/passwd
lrwx-----. 1 root root 64 Oct  3 14:10 fd/5 ->
'socket:[45446] '
lrwx-----. 1 root root 64 Oct  3 14:10 fd/6 ->
'socket:[45450] '
lr-x-----. 1 root root 64 Oct  3 14:10 fd/7 ->
/var/lib/sss/mc/group
lrwx-----. 1 root root 64 Oct  3 14:10 fd/8 ->
'socket:[45452] '
```

TODO: I'll be sure to write a separate post on the /proc filesystem with more thorough walkthrough.

Show Process Limits Using /proc

One of the files in /proc subdirectories is file called limits, and it's super useful for confirming the current OS limits applied to the process in question.

So for the sshd process with PID 5130, here's what we can see:

```
root@redhat:/proc/5130 # cat limits
```

| Limit | Soft Limit | Hard Limit |
|--------------------------------|------------|------------|
| Units | | |
| Max cpu time seconds | unlimited | unlimited |
| Max file size bytes | unlimited | unlimited |
| Max data size bytes | unlimited | unlimited |
| Max stack size bytes | 8388608 | unlimited |
| Max core file size bytes | unlimited | unlimited |
| Max resident set bytes | unlimited | unlimited |
| Max processes processes | 127372 | 127372 |
| Max open files files | 1024 | 4096 |
| Max locked memory bytes | 16777216 | 16777216 |
| Max address space bytes | unlimited | unlimited |
| Max file locks locks | unlimited | unlimited |
| Max pending signals signals | 127372 | 127372 |
| Max msgqueue size | 819200 | 819200 |

```
bytes
Max nice priority      0          0
Max realtime priority 0          0
Max realtime timeout   unlimited  unlimited
us
```

Basically, this confirms that I haven't fine-tuned anything on this new desktop just yet – open files count of 1024 is small cause it's not a server that requires serving multiple files simultaneously.

Hope you find this useful!

See Also

- [ps command](#)
- [lsof](#)
- [Show TCP connections with lsof](#)
- [Process tree](#)
- [How To Use lsof command](#)
- [Basic Unix Commands](#)
- [Using /proc/mdstat](#)
- [Monitor processes with htop](#)