Tips and Tricks with the proc filesystem

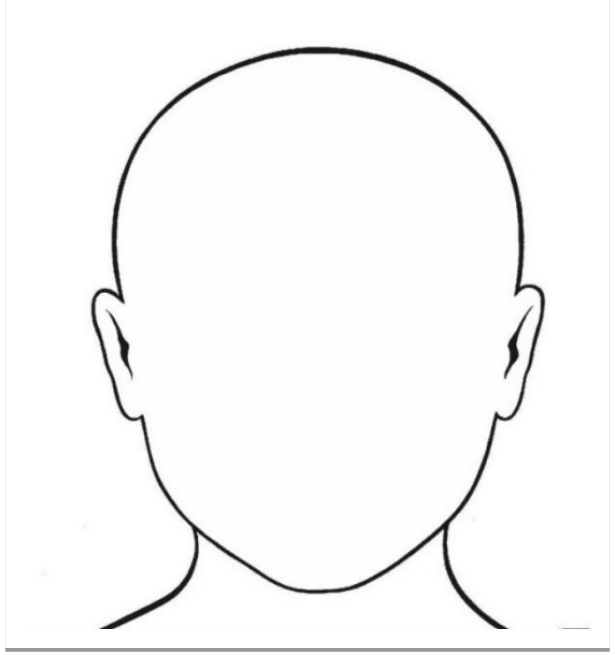
Luxoft | training
A DXC Technology Company

INTRODUCTION

Julius Schwartzenberg

jschwartzenberg@luxoft.com

http://luxoft-training.ru
http://luxoft-training.com



TRAINING ROADMAP: STRUCTURE & PREPARATION

- **single 1-2 hour session**
- **Practical, please open a terminal to a GNU/Linux system**
- **Use the lower right area of the screen**
- **Example daemon, mini_httpd: https://github.com/jschwartzenberg/mini_httpd**
- **Example media files: https://media.xiph.org/**

PRACTICAL PROBLEMS AND HOW PROCFS WILL HELP YOU

Your terminal here

Preparation

- **Get a copy of mini_httpd**

Git clone URLs:

- **git@github.com:jschwartzenberg/mini_httpd.git**
- **https://github.com/jschwartzenberg/mini_httpd.git**

Should build with a simple `make`

Introduction

- **Introduced with UNIX (8th edition), further enhanced in Plan 9, BSD 4.4**
- **Essential part of Solaris and Linux**
- **Continues on the principle**

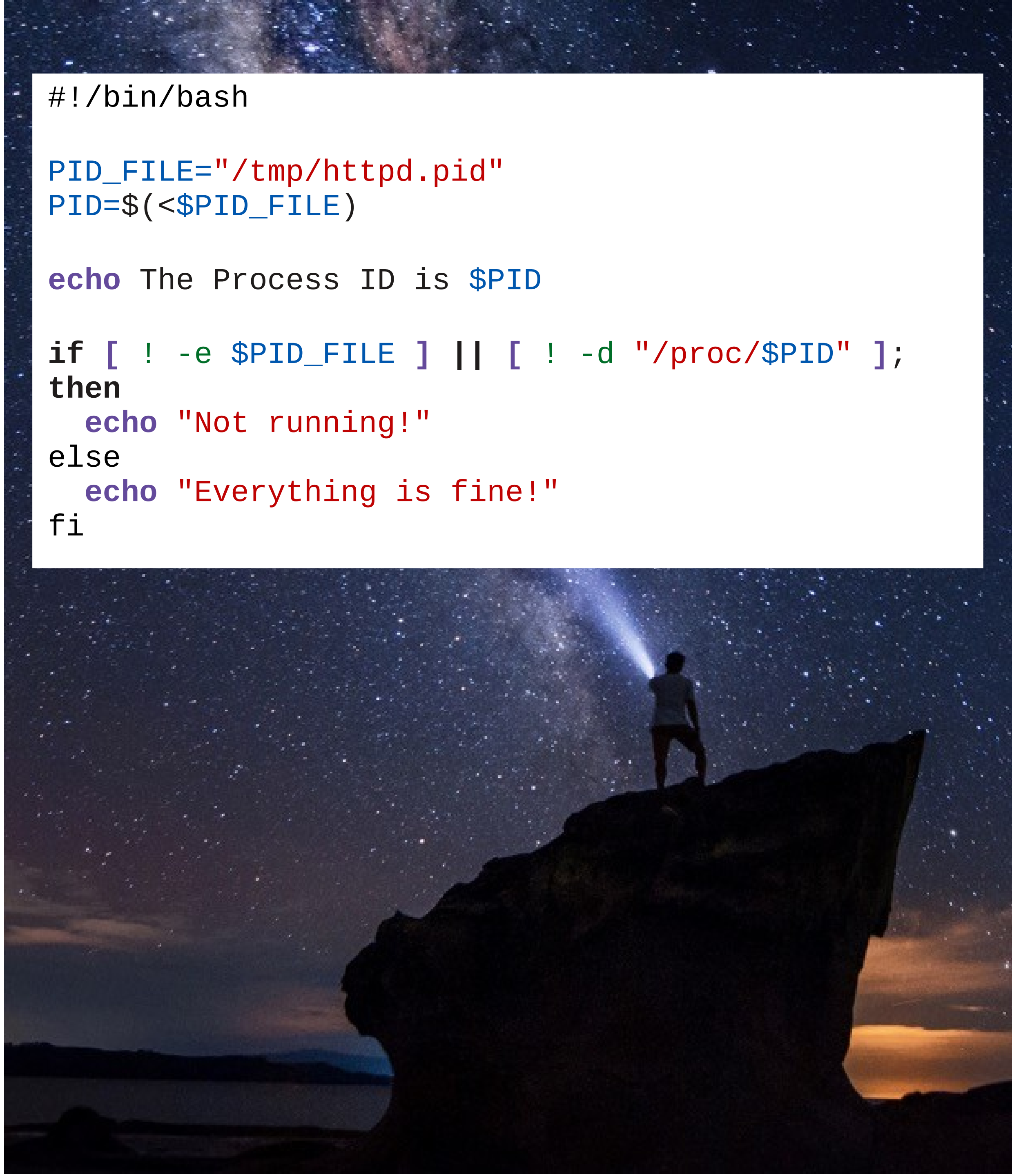
  "everything is a file"
- **Information about the system and individual processes**

Problem 1: Is my process running? (1/2)

- **Daemon crashes sometimes**
- **Daemon should be started only once**

Problem 1: Is my process running? (2/2)

- **Every running process is represented below /proc**
- **Daemons should create a pid file on start-up**



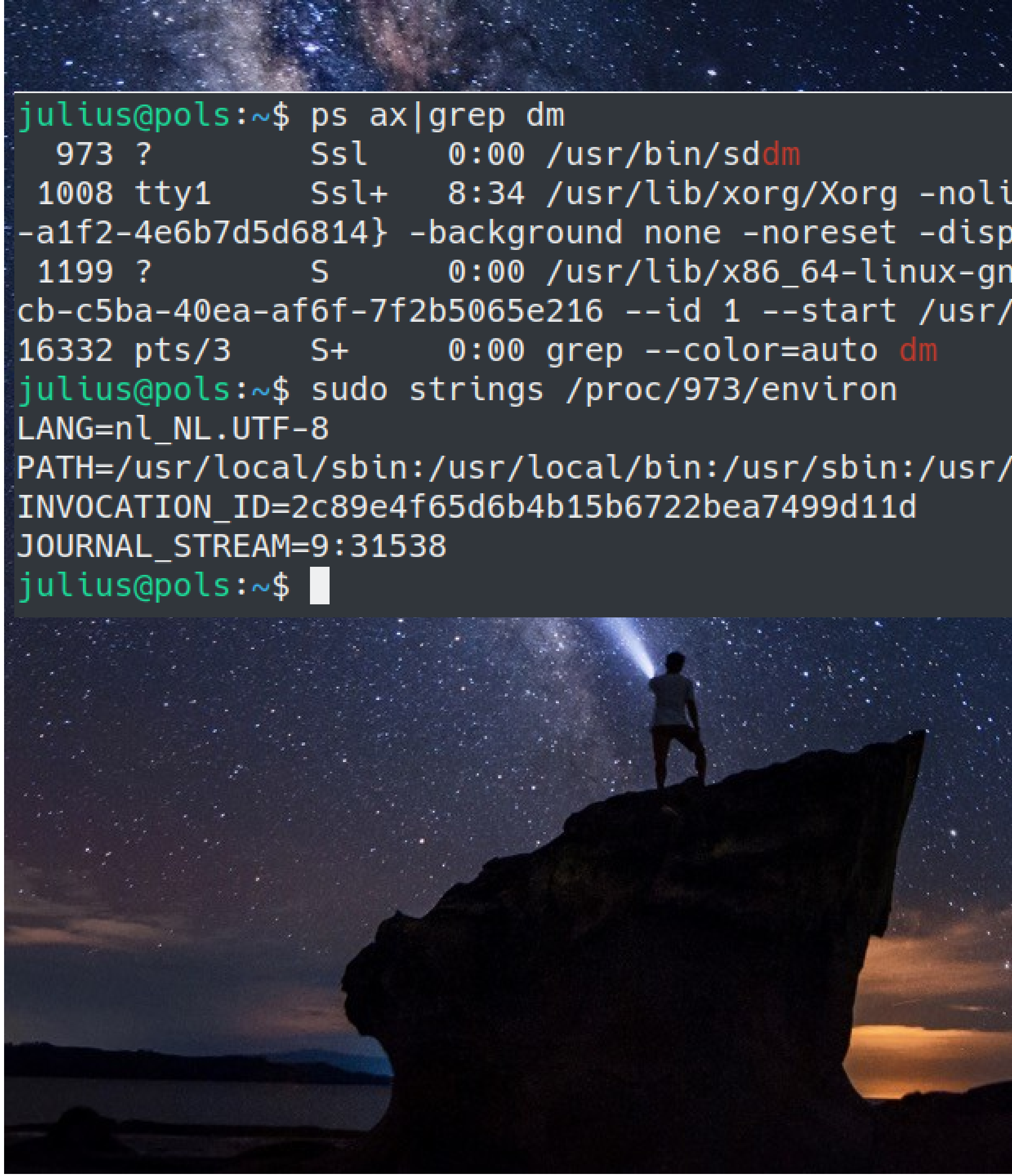
```
#!/bin/bash

PID_FILE="/tmp/httpd.pid"
PID=$(<$PID_FILE)

echo The Process ID is $PID

if [ ! -e $PID_FILE ] || [ ! -d "/proc/$PID" ];
then
  echo "Not running!"
else
  echo "Everything is fine!"
fi
```

Problem 2: How was my process started? (1/2)

- **You are looking at an unknown system**
- **Analyse its processes**
- **Discover its configuration**

Problem 2: How was my process started? (2/2)

- **Every running process has its own environment**
- **It can be read through /proc**
- **Individual variables are terminated with a NUL**
- **The command line used to start a process is also available**



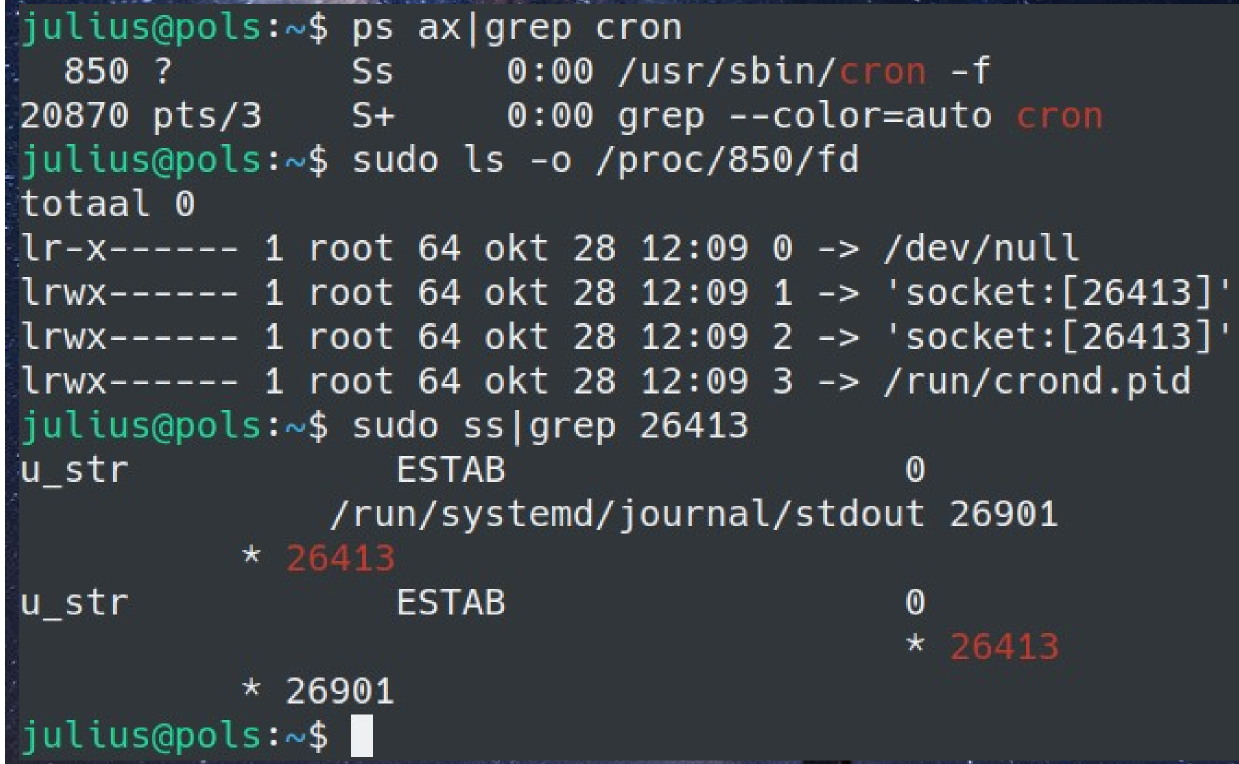
```
julius@pols:~$ ps ax|grep dm
  973 ?        Ssl    0:00 /usr/bin/sddm
 1008 tty1     Ssl+   8:34 /usr/lib/xorg/Xorg -noli
-a1f2-4e6b7d5d6814} -background none -noreset -disp
 1199 ?        S      0:00 /usr/lib/x86_64-linux-gn
cb-c5ba-40ea-af6f-7f2b5065e216 --id 1 --start /usr/
16332 pts/3    S+     0:00 grep --color=auto dm
julius@pols:~$ sudo strings /proc/973/environ
LANG=nl_NL.UTF-8
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/
INVOCATION_ID=2c89e4f65d6b4b15b6722bea7499d11d
JOURNAL_STREAM=9:31538
julius@pols:~$ 
```

Problem 3: Where is that process logging? (1/2)

- **You are looking at an unknown system**
- **No idea if any logging is in place**

Problem 3: Where is that process logging? (2/2)

- **Every running process has stdio: stdin, stdout, stderr**
- **Logging to stdout & stderr**
- **File descriptors are visible below /proc**
- **stdin: 0; stdout: 1; stderr: 2**
- **ss — tool to investigate sockets**



```
julius@pols:~$ ps ax|grep cron
  850 ?        Ss     0:00 /usr/sbin/cron -f
20870 pts/3    S+     0:00 grep --color=auto cron
julius@pols:~$ sudo ls -o /proc/850/fd
totaal 0
lr-x------ 1 root 64 okt 28 12:09 0 -> /dev/null
lrwx------ 1 root 64 okt 28 12:09 1 -> 'socket:[26413]'
lrwx------ 1 root 64 okt 28 12:09 2 -> 'socket:[26413]'
lrwx------ 1 root 64 okt 28 12:09 3 -> /run/crond.pid
julius@pols:~$ sudo ss|grep 26413
u_str            ESTAB                  0
              /run/systemd/journal/stdout 26901
          * 26413
u_str            ESTAB                  0
                                        * 26413
          * 26901
julius@pols:~$ 
```

Problem 4: Does the running process match the executable on disk? (1/3)

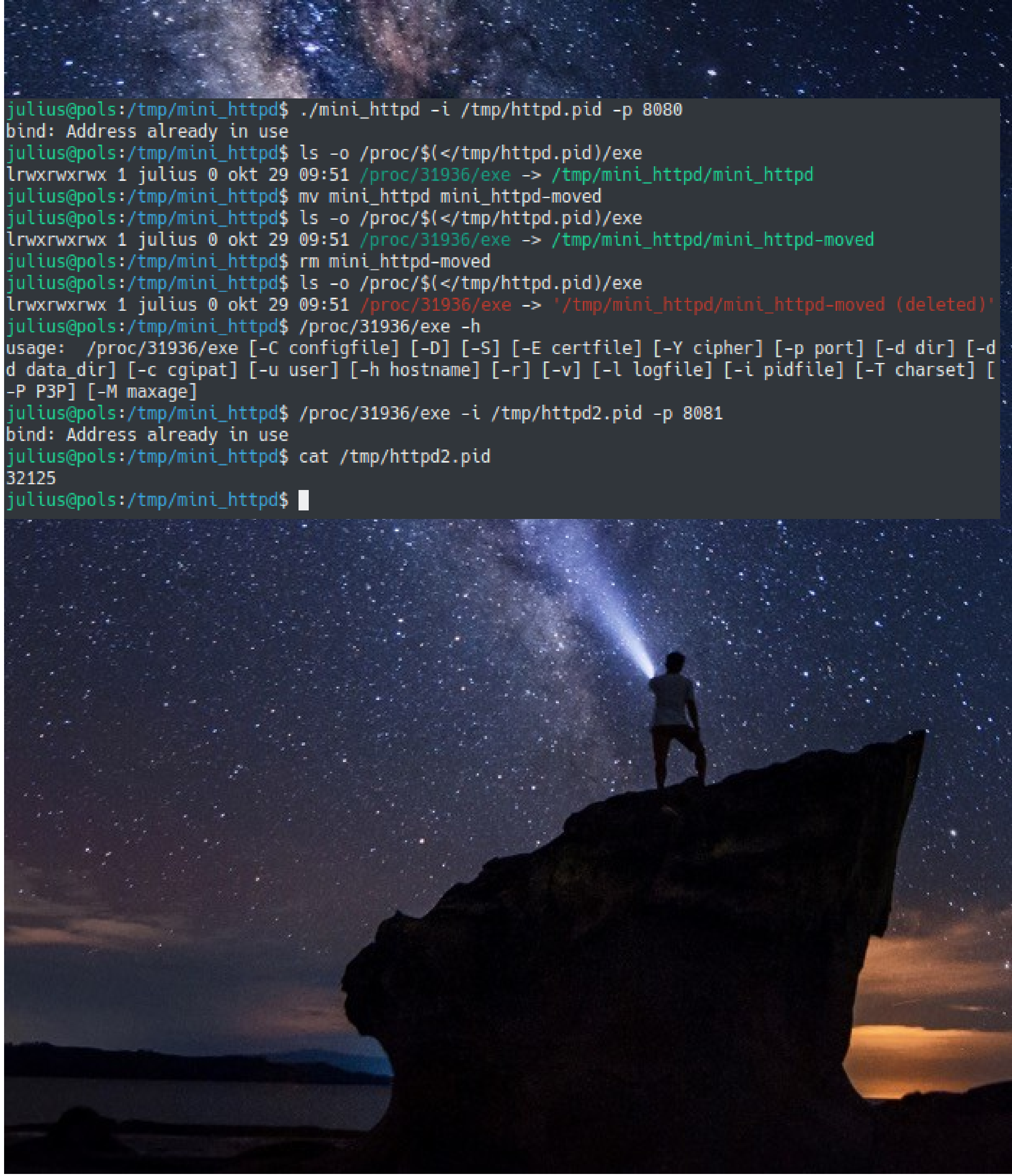
- **Updates were installed**
- **Version on disk might be different from the running one**
- **The previous version is not referenced anymore from the filesystem**
- **It is still somewhere**

Problem 4: Does the running process match the executable on disk? (2/3)

- **There is a pointer to the executable that is running below /proc**
- **Multiple references can point to the same filesystem object**
- **An object is only removed from the filesystem when there are no references left**

Problem 4: Does the running process match the executable on disk? (3/3)

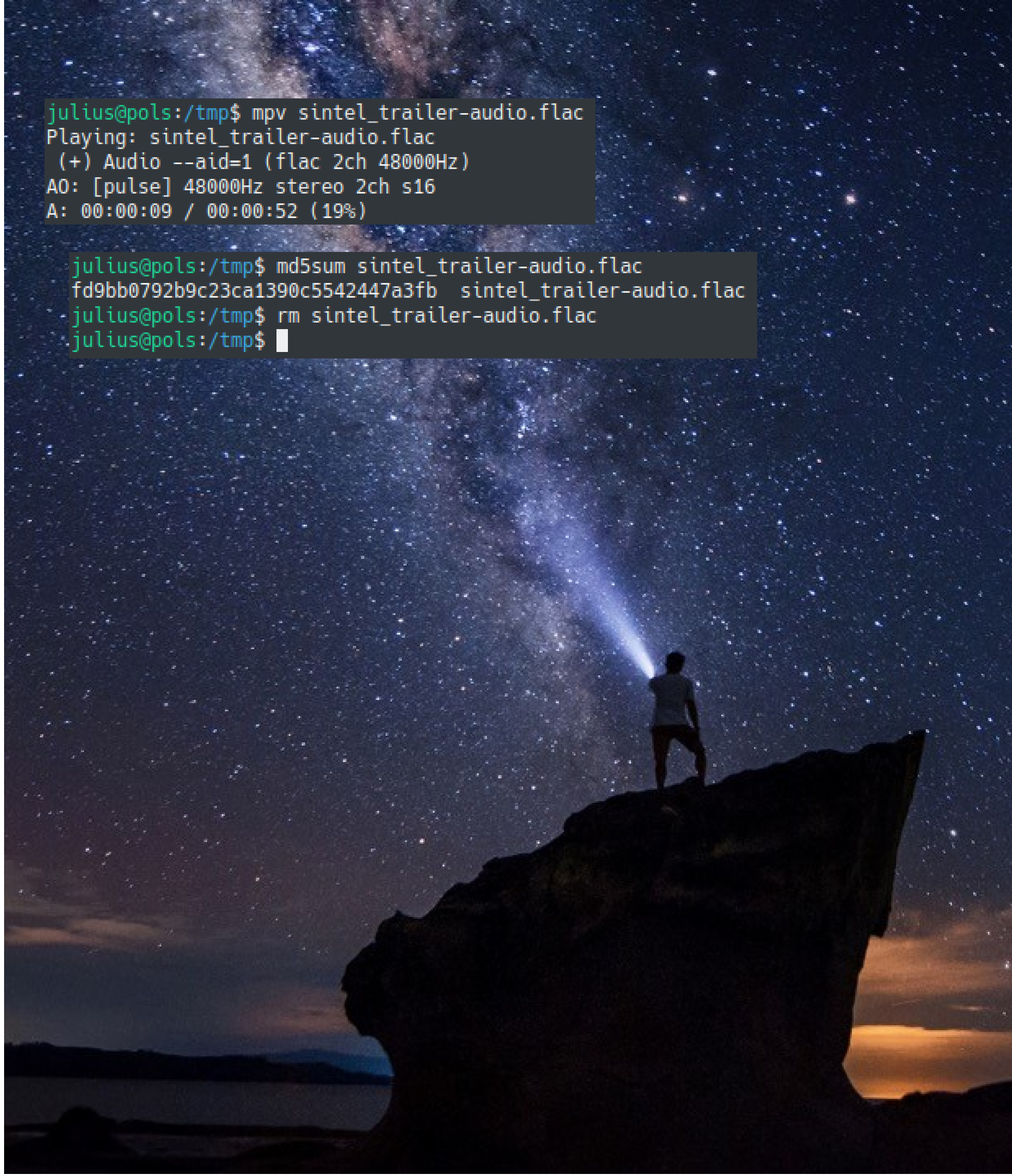
- **Start mini_httpd**
- **Find it below /proc**
- **What happens with mv?**
- **What happens with rm?**
- **What happens when replacing the executable?**
- **Executing from inside /proc?**



```
julius@pols:/tmp/mini_httpd$ ./mini_httpd -i /tmp/httpd.pid -p 8080
bind: Address already in use
julius@pols:/tmp/mini_httpd$ ls -o /proc/$(</tmp/httpd.pid)/exe
lrwxrwxrwx 1 julius 0 okt 29 09:51 /proc/31936/exe -> /tmp/mini_httpd/mini_httpd
julius@pols:/tmp/mini_httpd$ mv mini_httpd mini_httpd-moved
julius@pols:/tmp/mini_httpd$ ls -o /proc/$(</tmp/httpd.pid)/exe
lrwxrwxrwx 1 julius 0 okt 29 09:51 /proc/31936/exe -> /tmp/mini_httpd/mini_httpd-moved
julius@pols:/tmp/mini_httpd$ rm mini_httpd-moved
julius@pols:/tmp/mini_httpd$ ls -o /proc/$(</tmp/httpd.pid)/exe
lrwxrwxrwx 1 julius 0 okt 29 09:51 /proc/31936/exe -> '/tmp/mini_httpd/mini_httpd-moved (deleted)'
julius@pols:/tmp/mini_httpd$ /proc/31936/exe -h
usage:  /proc/31936/exe [-C configfile] [-D] [-S] [-E certfile] [-Y cipher] [-p port] [-d dir] [-d
d data_dir] [-c cgipat] [-u user] [-h hostname] [-r] [-v] [-l logfile] [-i pidfile] [-T charset] [
-P P3P] [-M maxage]
julius@pols:/tmp/mini_httpd$ /proc/31936/exe -i /tmp/httpd2.pid -p 8081
bind: Address already in use
julius@pols:/tmp/mini_httpd$ cat /tmp/httpd2.pid
32125
julius@pols:/tmp/mini_httpd$ 
```

Problem 5: File recovery for open files (1/2)

- **You accidentally deleted a song.**
- **But it is still playing!**
- **How we get to it?**

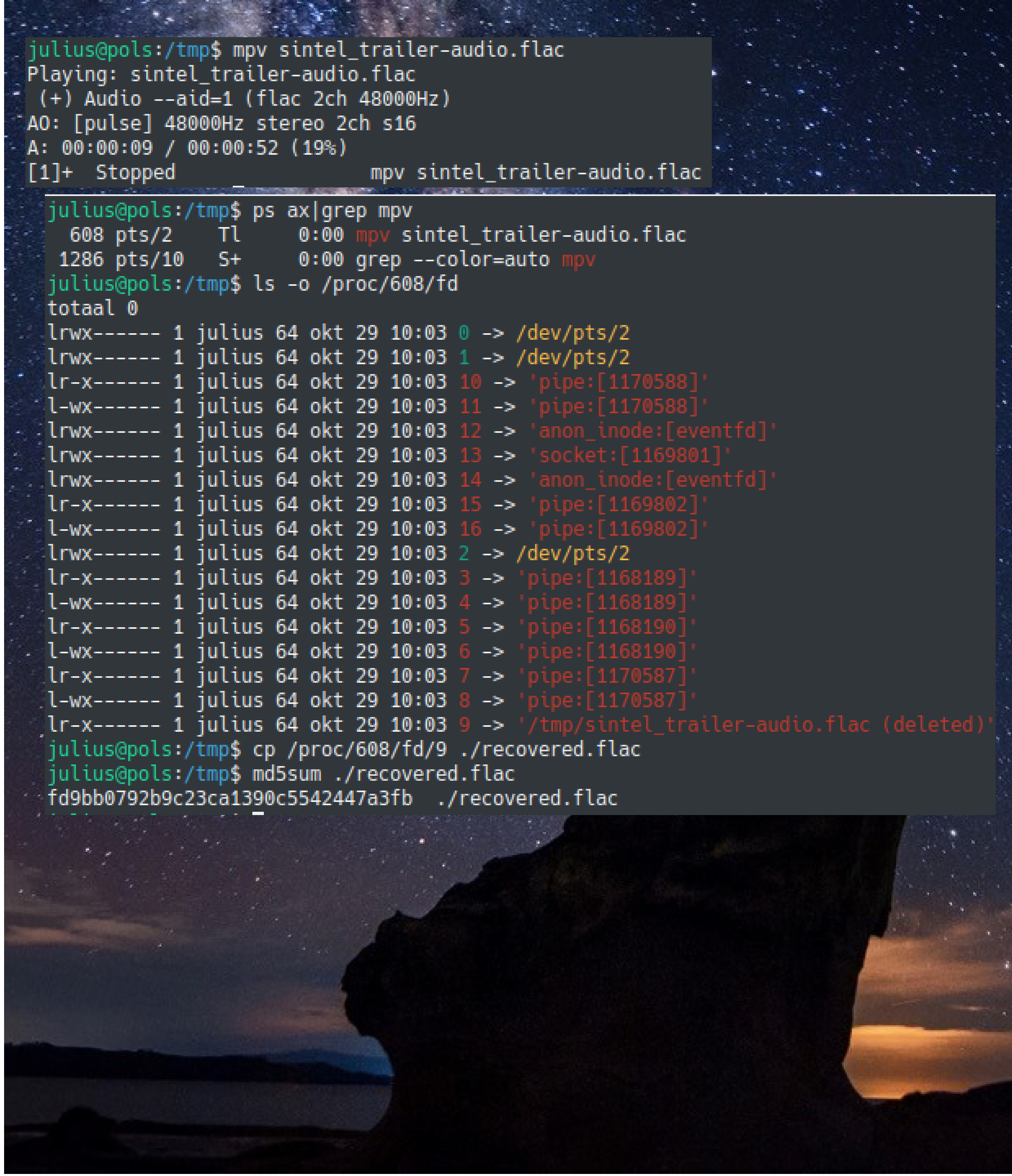


```
julius@pols:/tmp$ mpv sintel_trailer-audio.flac
Playing: sintel_trailer-audio.flac
 (+) Audio --aid=1 (flac 2ch 48000Hz)
AO: [pulse] 48000Hz stereo 2ch s16
A: 00:00:09 / 00:00:52 (19%)
```

```
julius@pols:/tmp$ md5sum sintel_trailer-audio.flac
fd9bb0792b9c23ca1390c5542447a3fb  sintel_trailer-audio.flac
julius@pols:/tmp$ rm sintel_trailer-audio.flac
julius@pols:/tmp$ 
```

Problem 5: File recovery for open files (2/2)

- **Whenever a process opens a file, this is handled through a file descriptor**
- **If your process might complete, you might want to suspend it: Ctrl+Z**



```
julius@pols:/tmp$ mpv sintel_trailer-audio.flac
Playing: sintel_trailer-audio.flac
 (+) Audio --aid=1 (flac 2ch 48000Hz)
AO: [pulse] 48000Hz stereo 2ch s16
A: 00:00:09 / 00:00:52 (19%)
[1]+  Stopped                 mpv sintel_trailer-audio.flac
```

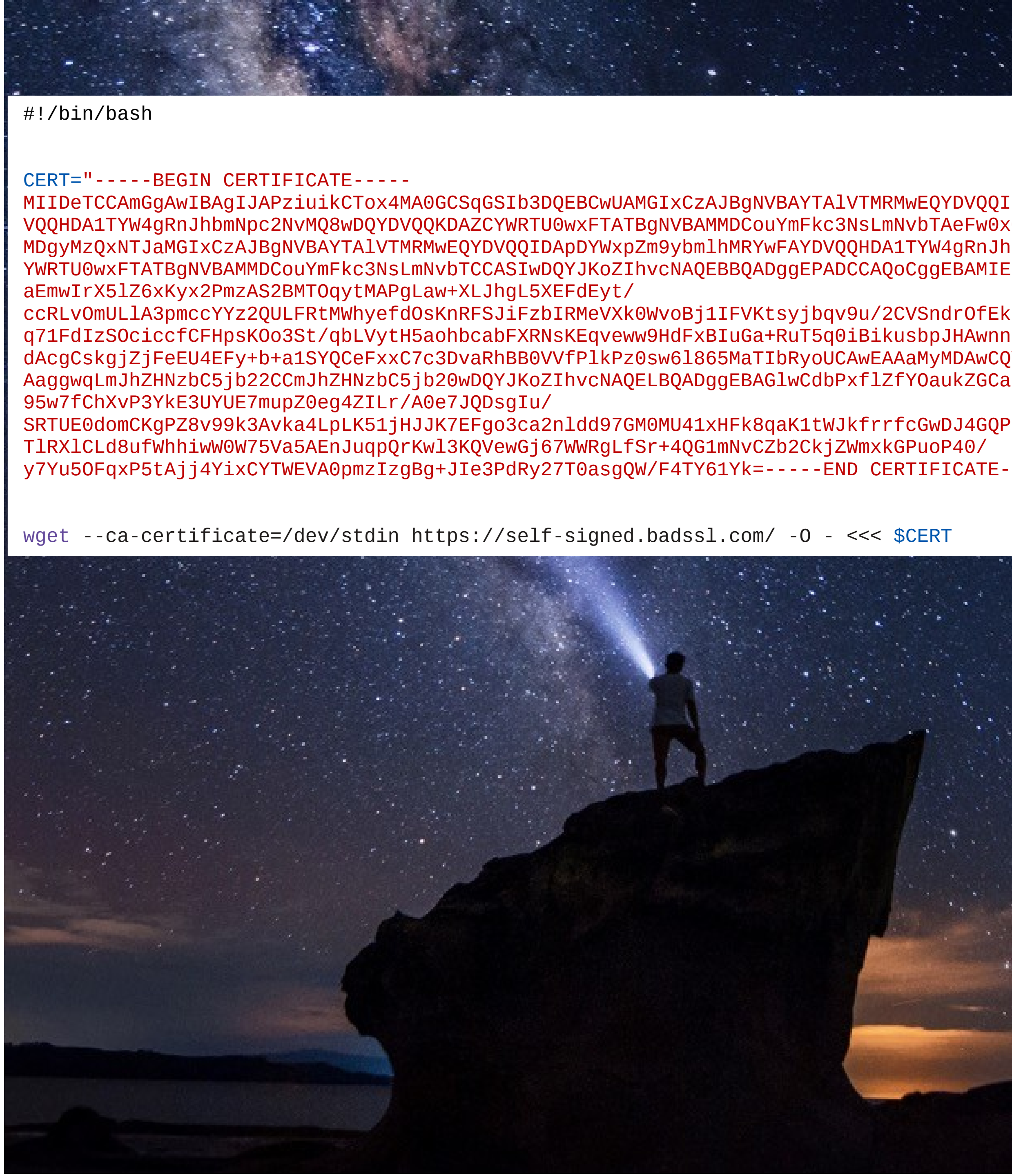
```
julius@pols:/tmp$ ps ax|grep mpv
  608 pts/2    Tl     0:00 mpv sintel_trailer-audio.flac
 1286 pts/10   S+     0:00 grep --color=auto mpv
julius@pols:/tmp$ ls -o /proc/608/fd
totaal 0
lrwx------ 1 julius 64 okt 29 10:03 0 -> /dev/pts/2
lrwx------ 1 julius 64 okt 29 10:03 1 -> /dev/pts/2
lr-x------ 1 julius 64 okt 29 10:03 10 -> 'pipe:[1170588]'
l-wx------ 1 julius 64 okt 29 10:03 11 -> 'pipe:[1170588]'
lrwx------ 1 julius 64 okt 29 10:03 12 -> 'anon_inode:[eventfd]'
lrwx------ 1 julius 64 okt 29 10:03 13 -> 'socket:[1169801]'
lrwx------ 1 julius 64 okt 29 10:03 14 -> 'anon_inode:[eventfd]'
lr-x------ 1 julius 64 okt 29 10:03 15 -> 'pipe:[1169802]'
l-wx------ 1 julius 64 okt 29 10:03 16 -> 'pipe:[1169802]'
lrwx------ 1 julius 64 okt 29 10:03 2 -> /dev/pts/2
lr-x------ 1 julius 64 okt 29 10:03 3 -> 'pipe:[1168189]'
l-wx------ 1 julius 64 okt 29 10:03 4 -> 'pipe:[1168189]'
lr-x------ 1 julius 64 okt 29 10:03 5 -> 'pipe:[1168190]'
l-wx------ 1 julius 64 okt 29 10:03 6 -> 'pipe:[1168190]'
lr-x------ 1 julius 64 okt 29 10:03 7 -> 'pipe:[1170587]'
l-wx------ 1 julius 64 okt 29 10:03 8 -> 'pipe:[1170587]'
lr-x------ 1 julius 64 okt 29 10:03 9 -> '/tmp/sintel_trailer-audio.flac (deleted)'
julius@pols:/tmp$ cp /proc/608/fd/9 ./recovered.flac
julius@pols:/tmp$ md5sum ./recovered.flac
fd9bb0792b9c23ca1390c5542447a3fb  ./recovered.flac
```

Problem 6: Read stdin when a file is expected (1/2)

- **Your want to embed a file in your shell script and pass it as input to your program**
- **Many programs can read stdin, but not all**
- **Example: wget --ca-certificate=<file>**

Problem 6: Read stdin when a file is expected (2/2)

- **stdin in is represented as a file**



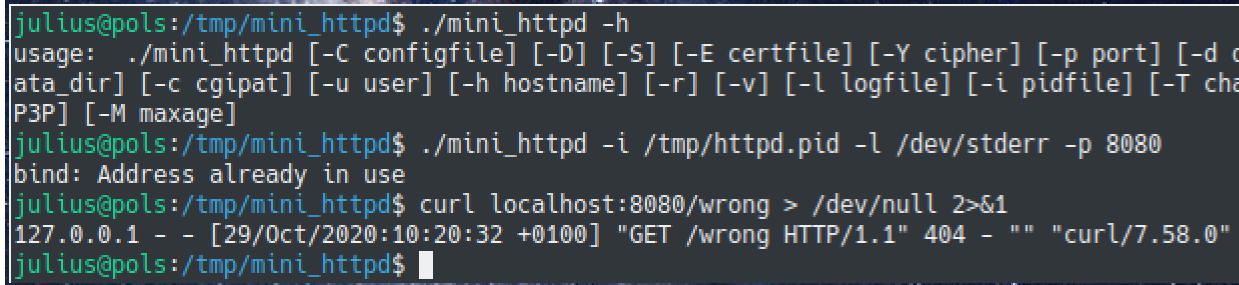
```
#!/bin/bash

CERT="-----BEGIN CERTIFICATE-----
MIIDeTCCAmGgAwIBAgIJAPziuikCTox4MA0GCSqGSIb3DQEBCwUAMGIxCzAJBgNVBAYTAlVTMRMwEQYDVQQI
VQQHDA1TYW4gRnJhbmNpc2NvMQ8wDQYDVQQKDAZCYWRTU0wxFTATBgNVBAMMDCouYmFkc3NsLmNvbTAeFw0x
MDgyMzQxNTJaMGIxCzAJBgNVBAYTAlVTMRMwEQYDVQQIDApDYWxpZm9ybmlhMRYwFAYDVQQHDA1TYW4gRnJh
YWRTU0wxFTATBgNVBAMMDCouYmFkc3NsLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMIE
aEmwIrX5lZ6xKyx2PmzAS2BMTOqytMAPgLaw+XLJhgL5XEFdEyt/
ccRLvOmULlA3pmccYYz2QULFRtMWhyefdOsKnRFSJiFzbIRMeVXk0WvoBj1IFVKtsyjbqv9u/2CVSndrOfEk
q71FdIzSOciccfCFHpsKOo3St/qbLVytH5aohbcabFXRNsKEqveww9HdFxBIuGa+RuT5q0iBikusbpJHAwnn
dAcgCskgjZjFeEU4EFy+b+a1SYQCeFxxC7c3DvaRhBB0VVfPlkPz0sw6l865MaTIbRyoUCAwEAAaMyMDAwCQ
AaggwqLmJhZHNzbC5jb22CCmJhZHNzbC5jb20wDQYJKoZIhvcNAQELBQADggEBAGlwCdbPxflZfYOaukZGCa
95w7fChXvP3YkE3UYUE7mupZ0eg4ZILr/A0e7JQDsgIu/
SRTUE0domCKgPZ8v99k3Avka4LpLK51jHJJK7EFgo3ca2nldd97GM0MU41xHFk8qaK1tWJkfrrfcGwDJ4GQP
TlRXlCLd8ufWhhiwW0W75Va5AEnJuqpQrKwl3KQVewGj67WWRgLfSr+4QG1mNvCZb2CkjZWmxkGPuoP40/
y7Yu5OFqxP5tAjj4YixCYTWEVA0pmzIzgBg+JIe3PdRy27T0asgQW/F4TY61Yk=-----END CERTIFICATE-

wget --ca-certificate=/dev/stdin https://self-signed.badssl.com/ -O - <<< $CERT
```

Problem 7: Write stdout/stderr when a file is expected (1/2)

- **You want to see the output in your terminal**
- **mini_httpd -l <logfile>**

Problem 7: Write stdout/stderr when a file is expected (2/2)

- **Solution is similar to the one from problem 6**



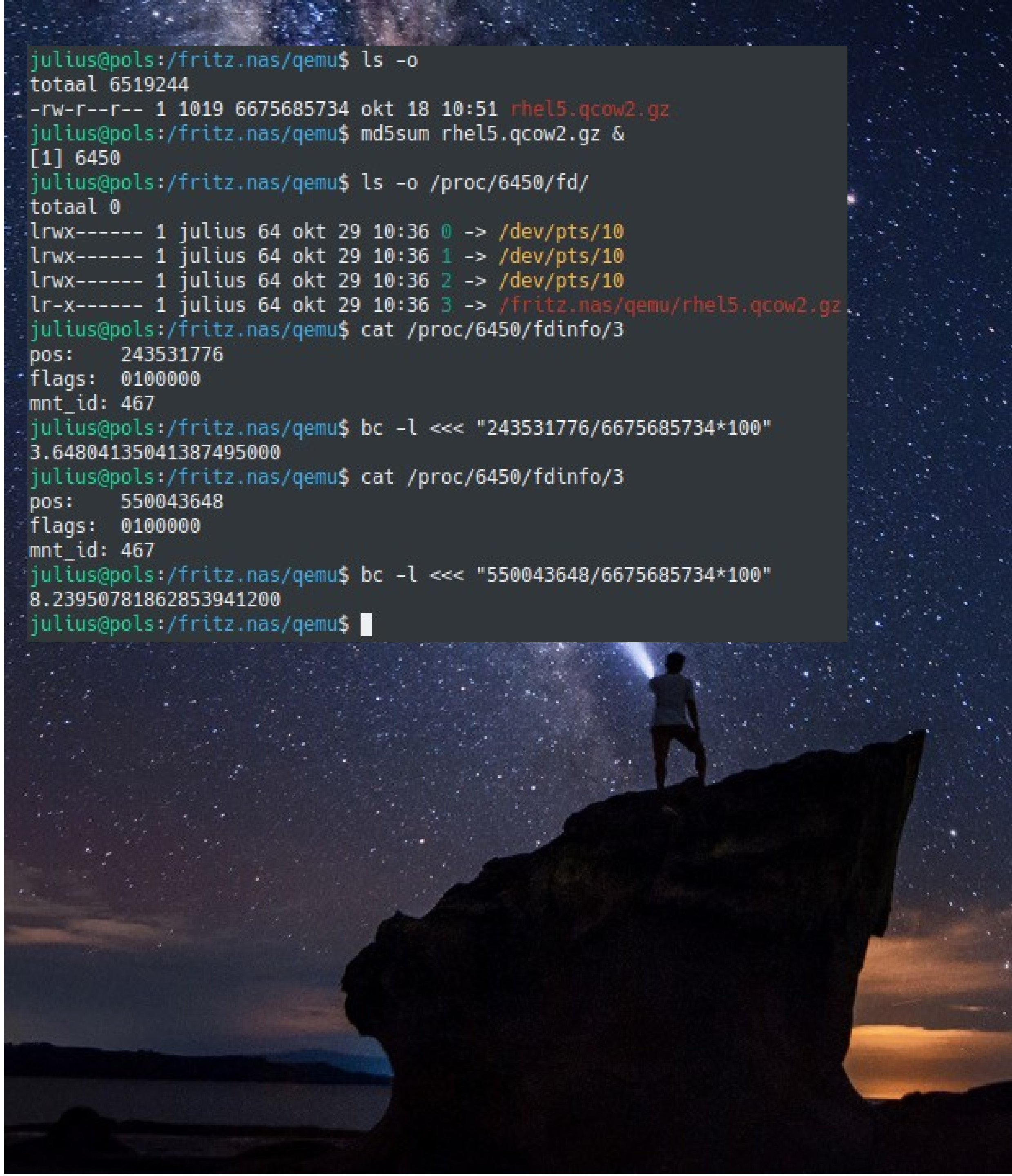
```
julius@pols:/tmp/mini_httpd$ ./mini_httpd -h
usage:  ./mini_httpd [-C configfile] [-D] [-S] [-E certfile] [-Y cipher] [-p port] [-d d
ata_dir] [-c cgipat] [-u user] [-h hostname] [-r] [-v] [-l logfile] [-i pidfile] [-T cha
P3P] [-M maxage]
julius@pols:/tmp/mini_httpd$ ./mini_httpd -i /tmp/httpd.pid -l /dev/stderr -p 8080
bind: Address already in use
julius@pols:/tmp/mini_httpd$ curl localhost:8080/wrong > /dev/null 2>&1
127.0.0.1 - - [29/Oct/2020:10:20:32 +0100] "GET /wrong HTTP/1.1" 404 - "" "curl/7.58.0"
julius@pols:/tmp/mini_httpd$
```

Problem 8: Determine progress of a running program (1/2)

- **A process is running over a large file that will take a long time**
- **You have no idea how much time is left**
- **Usually processes read a file from start to end**

Problem 8: Determine progress of a running program (2/2)

- **In addition to /proc/<pid>/fd, there is also /proc/<pid>/fdinfo**
- **Look for the value of "pos"**



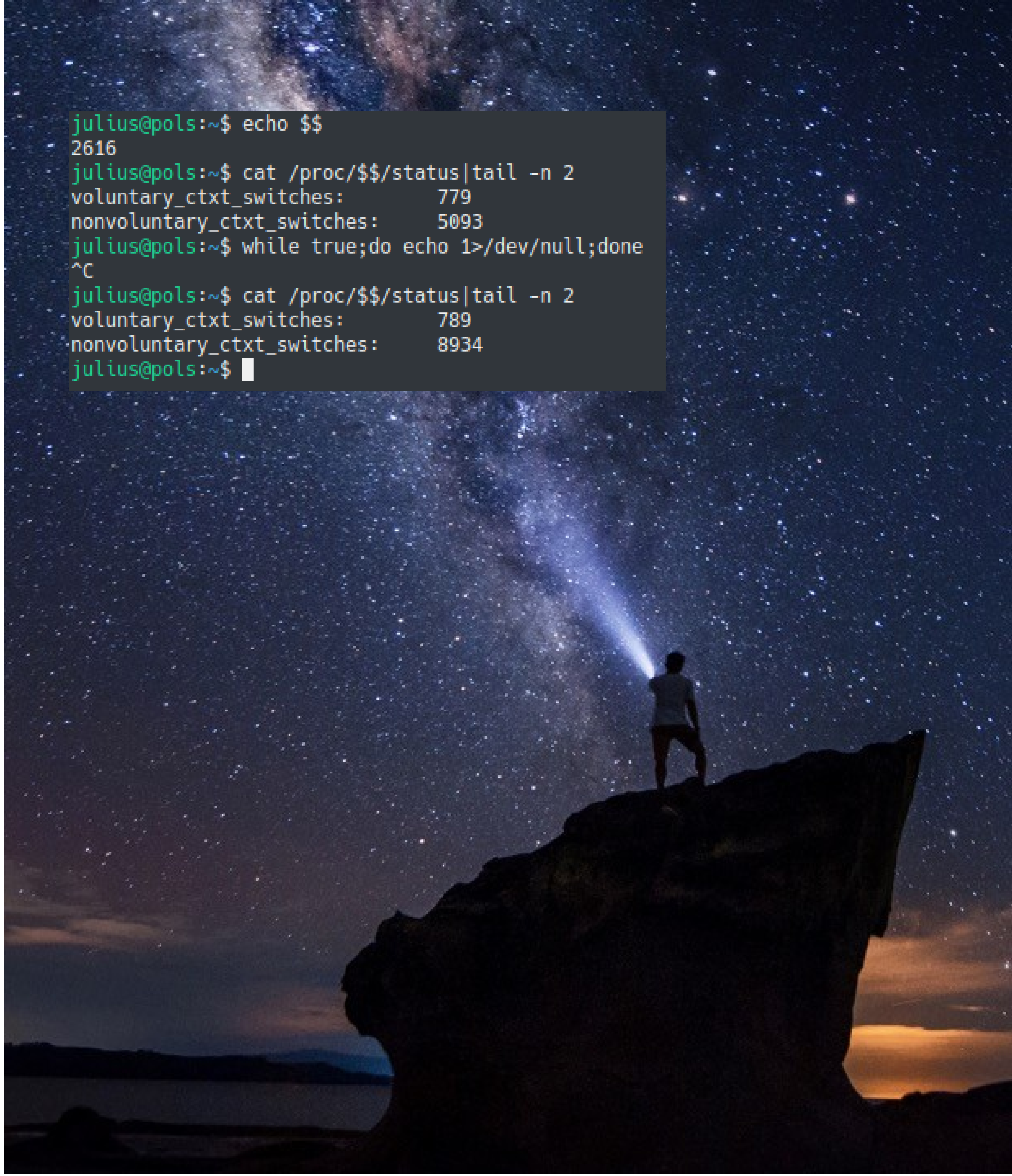
```
julius@pols:/fritz.nas/qemu$ ls -o
totaal 6519244
-rw-r--r-- 1 1019 6675685734 okt 18 10:51 rhel5.qcow2.gz
julius@pols:/fritz.nas/qemu$ md5sum rhel5.qcow2.gz &
[1] 6450
julius@pols:/fritz.nas/qemu$ ls -o /proc/6450/fd/
totaal 0
lrwx------ 1 julius 64 okt 29 10:36 0 -> /dev/pts/10
lrwx------ 1 julius 64 okt 29 10:36 1 -> /dev/pts/10
lrwx------ 1 julius 64 okt 29 10:36 2 -> /dev/pts/10
lr-x------ 1 julius 64 okt 29 10:36 3 -> /fritz.nas/qemu/rhel5.qcow2.gz
julius@pols:/fritz.nas/qemu$ cat /proc/6450/fdinfo/3
pos:    243531776
flags:  0100000
mnt_id: 467
julius@pols:/fritz.nas/qemu$ bc -l <<< "243531776/6675685734*100"
3.64804135041387495000
julius@pols:/fritz.nas/qemu$ cat /proc/6450/fdinfo/3
pos:    550043648
flags:  0100000
mnt_id: 467
julius@pols:/fritz.nas/qemu$ bc -l <<< "550043648/6675685734*100"
8.23950781862853941200
julius@pols:/fritz.nas/qemu$ 
```

Problem 9: Is a process CPU- or IO-bound? (1/2)

- **A process runs slow**
- **Where is the bottleneck?**

Problem 9: Is a process CPU- or IO-bound? (2/2)

- **Processes that use IO intensively will make more system calls**
- **A system call implies a context switch**
- **Processes that do only calculation will not request a context switch**
- **Context switches are exposed via /proc/<pid>/status**



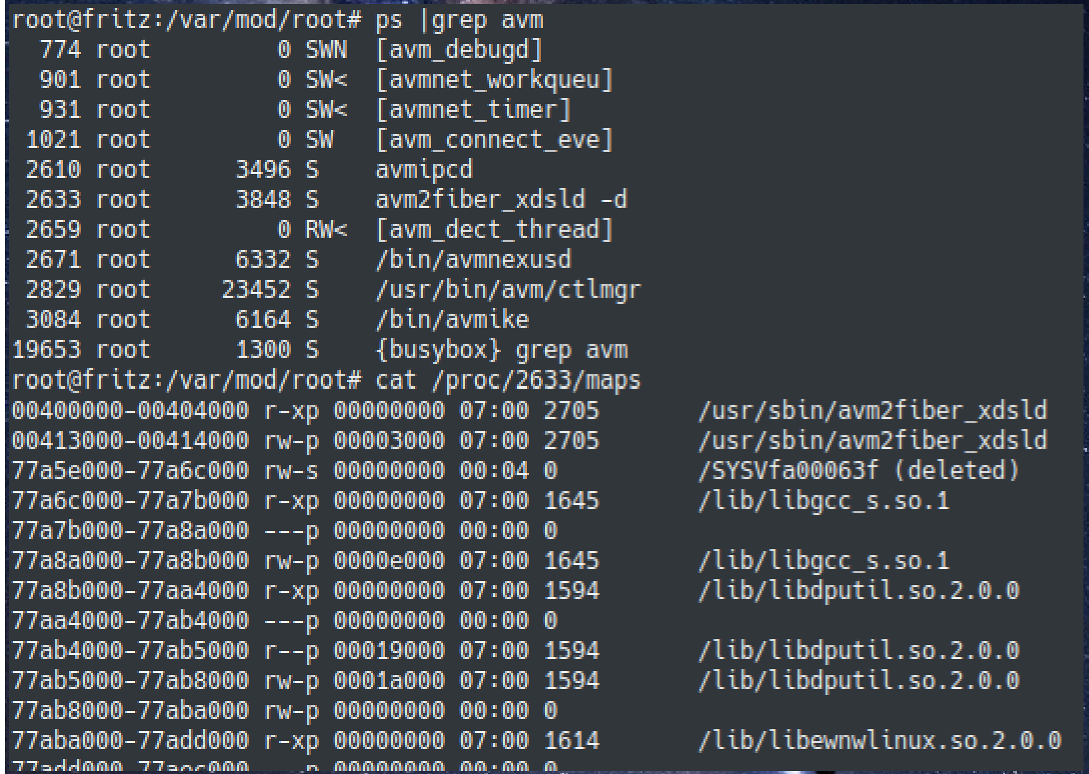
```
julius@pols:~$ echo $$
2616
julius@pols:~$ cat /proc/$$/status|tail -n 2
voluntary_ctxt_switches:        779
nonvoluntary_ctxt_switches:     5093
julius@pols:~$ while true;do echo 1>/dev/null;done
^C
julius@pols:~$ cat /proc/$$/status|tail -n 2
voluntary_ctxt_switches:        789
nonvoluntary_ctxt_switches:     8934
julius@pols:~$ 
```

Problem 10: Which libraries are loaded? (1/2)

- **Why is the program crashing?**
- **Might be be loading incompatible libraries at runtime?**

Problem 10: Which libraries are loaded? (2/2)

- **/proc provides insight in the memory map**
- **It shows which libraries are loaded**
- **Take a look at /proc/<pid>/maps**



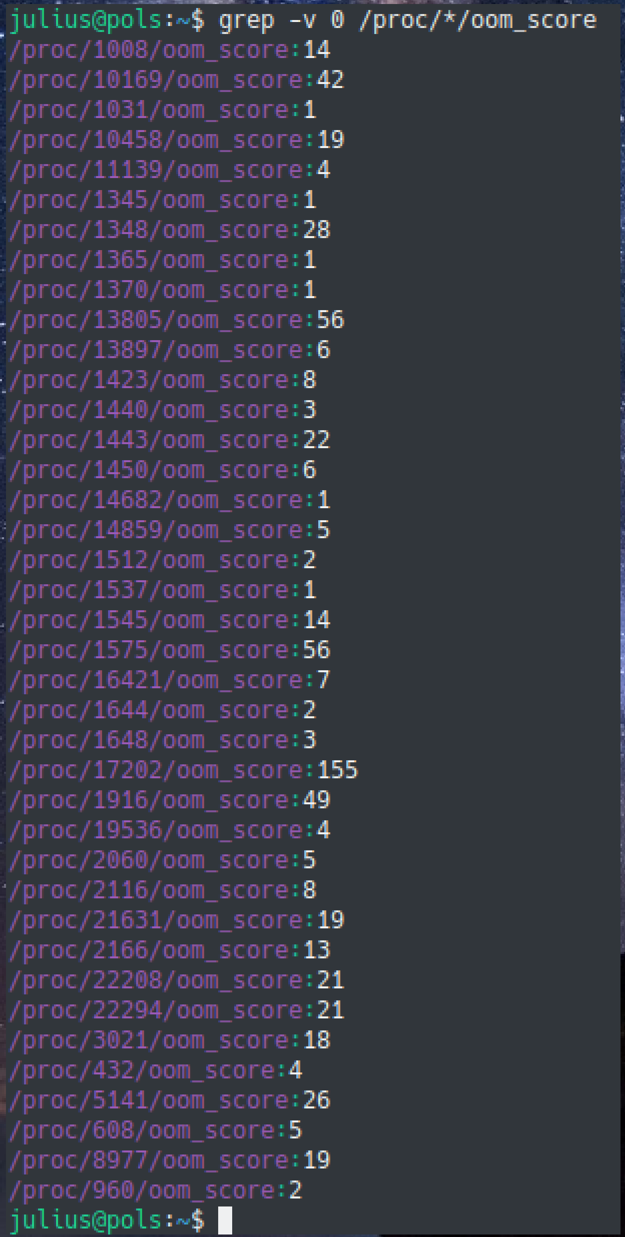
```
root@fritz:/var/mod/root# ps |grep avm
  774 root         0 SWN  [avm_debugd]
  901 root         0 SW<  [avmnet_workqueu]
  931 root         0 SW<  [avmnet_timer]
 1021 root         0 SW   [avm_connect_eve]
 2610 root      3496 S    avmipcd
 2633 root      3848 S    avm2fiber_xdsld -d
 2659 root         0 RW<  [avm_dect_thread]
 2671 root      6332 S    /bin/avmnexusd
 2829 root     23452 S    /usr/bin/avm/ctlmgr
 3084 root      6164 S    /bin/avmike
19653 root      1300 S    {busybox} grep avm
root@fritz:/var/mod/root# cat /proc/2633/maps
00400000-00404000 r-xp 00000000 07:00 2705       /usr/sbin/avm2fiber_xdsld
00413000-00414000 rw-p 00003000 07:00 2705       /usr/sbin/avm2fiber_xdsld
77a5e000-77a6c000 rw-s 00000000 00:04 0          /SYSVfa00063f (deleted)
77a6c000-77a7b000 r-xp 00000000 07:00 1645       /lib/libgcc_s.so.1
77a7b000-77a8a000 ---p 00000000 00:00 0
77a8a000-77a8b000 rw-p 0000e000 07:00 1645       /lib/libgcc_s.so.1
77a8b000-77aa4000 r-xp 00000000 07:00 1594       /lib/libdputil.so.2.0.0
77aa4000-77ab4000 ---p 00000000 00:00 0
77ab4000-77ab5000 r--p 00019000 07:00 1594       /lib/libdputil.so.2.0.0
77ab5000-77ab8000 rw-p 0001a000 07:00 1594       /lib/libdputil.so.2.0.0
77ab8000-77aba000 rw-p 00000000 00:00 0
77aba000-77add000 r-xp 00000000 07:00 1614       /lib/libewnwlinux.so.2.0.0
77add000-77aec000 ---p 00000000 00:00 0
```

Problem 11: Which process is using so much memory? (1/2)

- **Memory usage on a system is very high**
- **Which process is the most likely culprit?**
- **The Out-Of-Memory (OOM) killer might kick in when it is too late**

Problem 11: Which process is using so much memory? (2/2)

- **The OOM killer keeps track**
- **The scores are available below /proc**



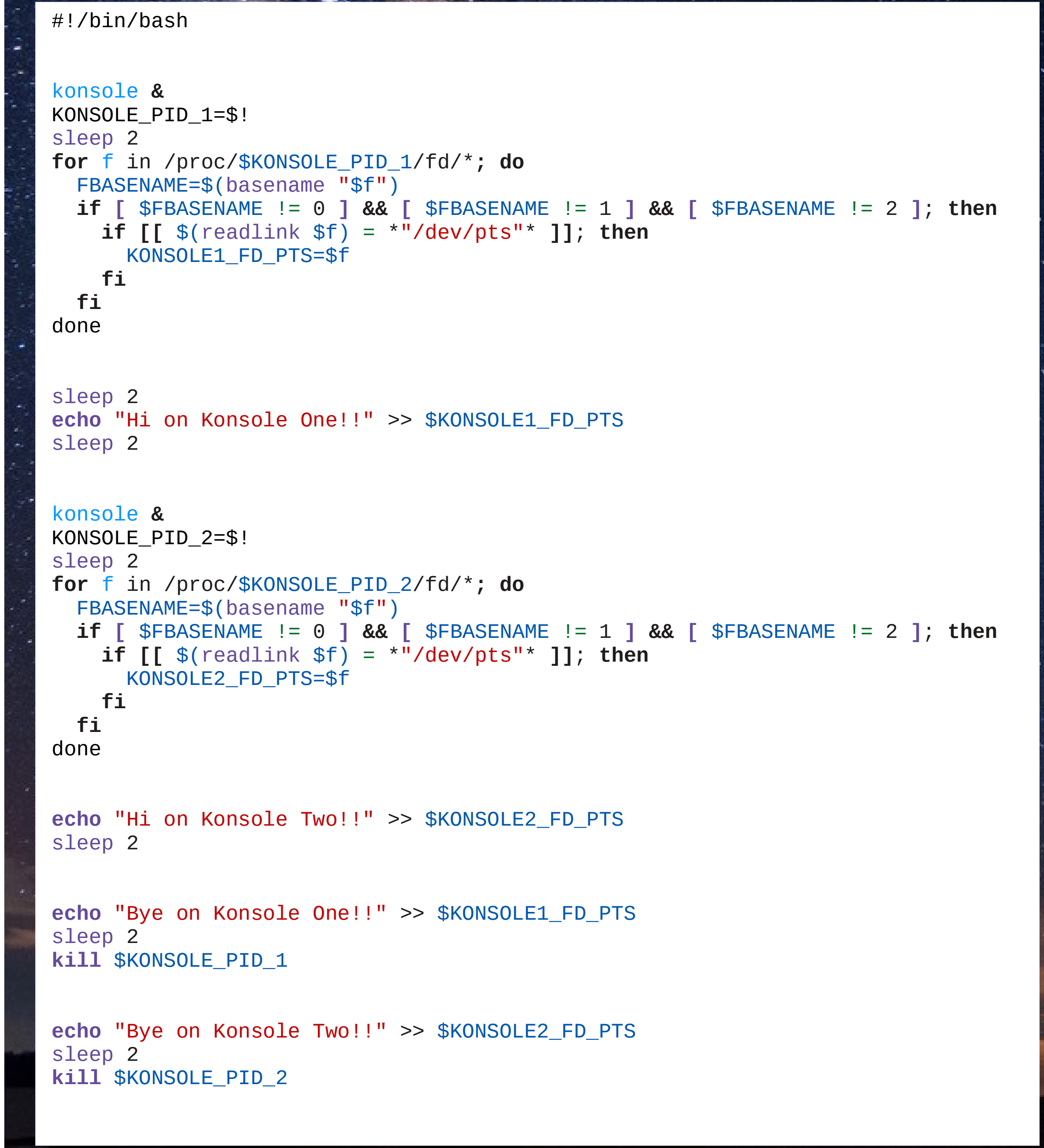
```
julius@pols:~$ grep -v 0 /proc/*/oom_score
/proc/1008/oom_score:14
/proc/10169/oom_score:42
/proc/1031/oom_score:1
/proc/10458/oom_score:19
/proc/11139/oom_score:4
/proc/1345/oom_score:1
/proc/1348/oom_score:28
/proc/1365/oom_score:1
/proc/1370/oom_score:1
/proc/13805/oom_score:56
/proc/13897/oom_score:6
/proc/1423/oom_score:8
/proc/1440/oom_score:3
/proc/1443/oom_score:22
/proc/1450/oom_score:6
/proc/14682/oom_score:1
/proc/14859/oom_score:5
/proc/1512/oom_score:2
/proc/1537/oom_score:1
/proc/1545/oom_score:14
/proc/1575/oom_score:56
/proc/16421/oom_score:7
/proc/1644/oom_score:2
/proc/1648/oom_score:3
/proc/17202/oom_score:155
/proc/1916/oom_score:49
/proc/19536/oom_score:4
/proc/2060/oom_score:5
/proc/2116/oom_score:8
/proc/21631/oom_score:19
/proc/2166/oom_score:13
/proc/22208/oom_score:21
/proc/22294/oom_score:21
/proc/3021/oom_score:18
/proc/432/oom_score:4
/proc/5141/oom_score:26
/proc/608/oom_score:5
/proc/8977/oom_score:19
/proc/960/oom_score:2
julius@pols:~$ 
```

Problem 12: Writing to other terminals (1/2)

- **Can you hijack one terminal from another?**
- **Can one script output to multiple terminal windows?**

Problem 12: Writing to other terminals (2/2)

- **A terminal is attached to a pseudo terminal device**
- **Look at its file descriptors**
- **What happens when you write into a pseudo terminal device?**



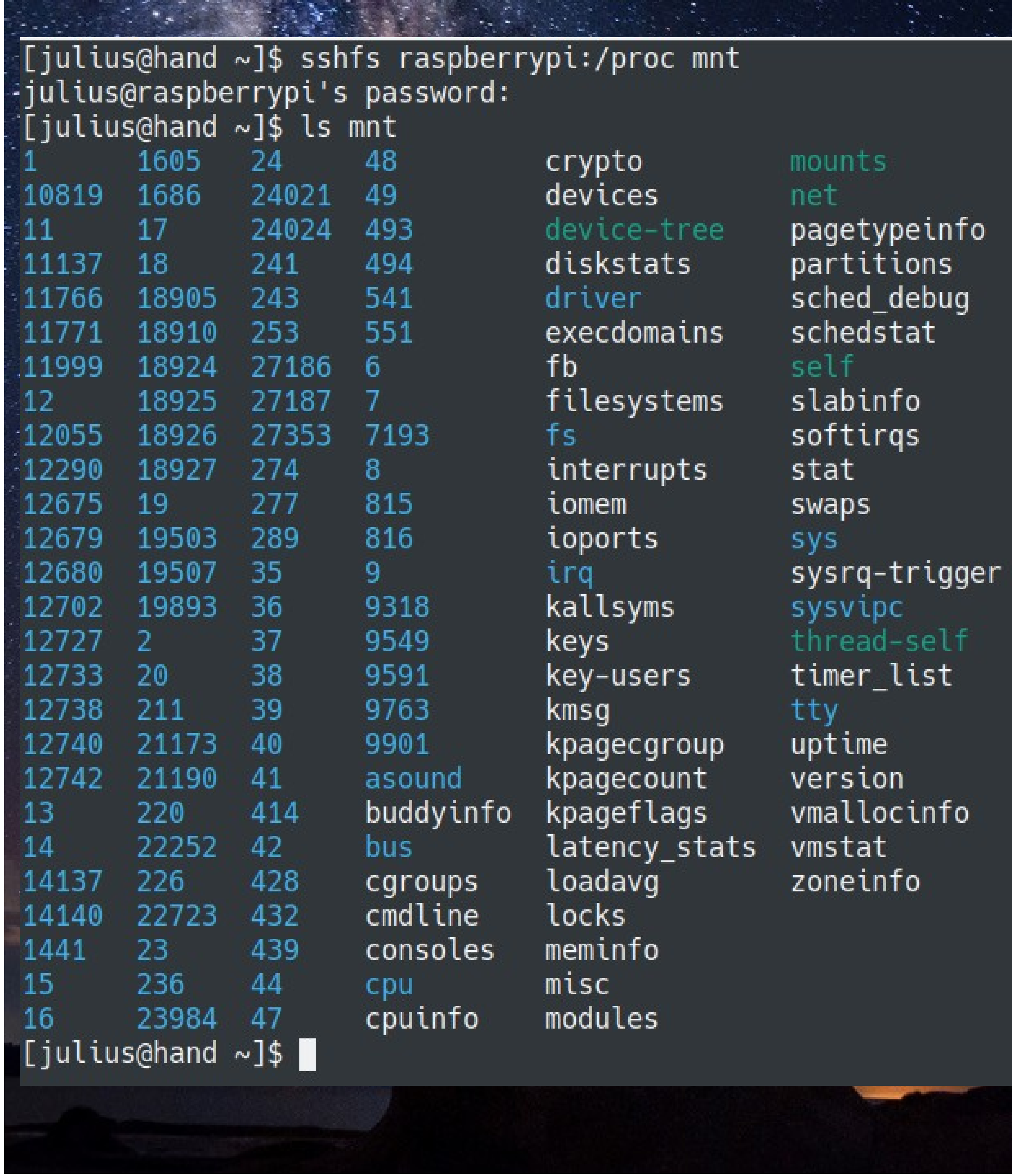
```
#!/bin/bash

konsole &
KONSOLE_PID_1=$!
sleep 2
for f in /proc/$KONSOLE_PID_1/fd/*; do
  FBASENAME=$(basename "$f")
  if [ $FBASENAME != 0 ] && [ $FBASENAME != 1 ] && [ $FBASENAME != 2 ]; then
    if [[ $(readlink $f) = *"/dev/pts"* ]]; then
      KONSOLE1_FD_PTS=$f
    fi
  fi
done

sleep 2
echo "Hi on Konsole One!!" >> $KONSOLE1_FD_PTS
sleep 2

konsole &
KONSOLE_PID_2=$!
sleep 2
for f in /proc/$KONSOLE_PID_2/fd/*; do
  FBASENAME=$(basename "$f")
  if [ $FBASENAME != 0 ] && [ $FBASENAME != 1 ] && [ $FBASENAME != 2 ]; then
    if [[ $(readlink $f) = *"/dev/pts"* ]]; then
      KONSOLE2_FD_PTS=$f
    fi
  fi
done

echo "Hi on Konsole Two!!" >> $KONSOLE2_FD_PTS
sleep 2

echo "Bye on Konsole One!!" >> $KONSOLE1_FD_PTS
sleep 2
kill $KONSOLE_PID_1

echo "Bye on Konsole Two!!" >> $KONSOLE2_FD_PTS
sleep 2
kill $KONSOLE_PID_2
```

Problem 13: Mounting a remote /proc? (1/2)

- **Can you mount /proc remotely?**

Problem 13: Mounting a remote /proc? (2/2)

- **NFS does not work**
- **What about SSHFS?**



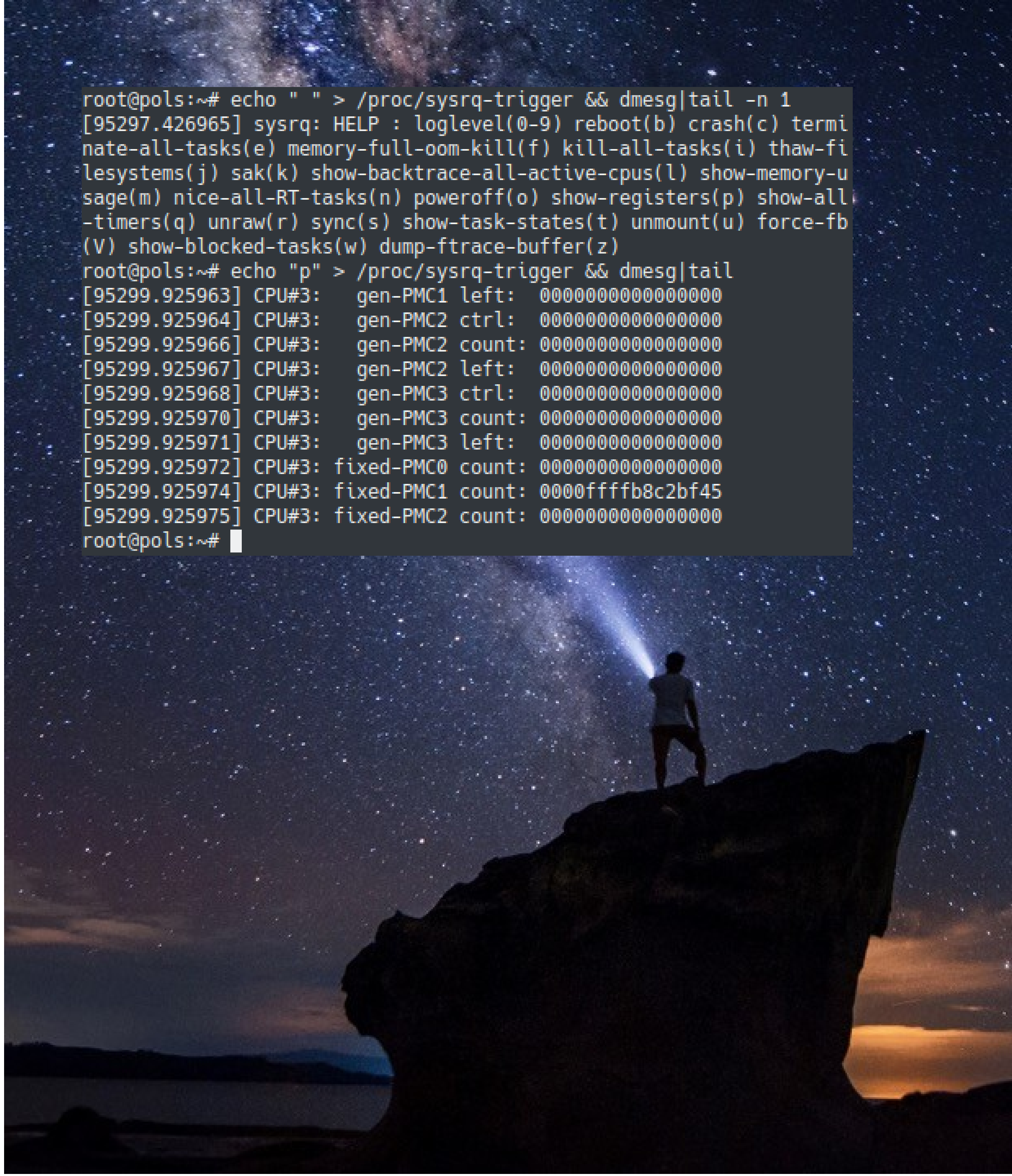
```
[julius@hand ~]$ sshfs raspberrypi:/proc mnt
julius@raspberrypi's password:
[julius@hand ~]$ ls mnt
1      1605   24     48         crypto         mounts
10819  1686   24021  49         devices        net
11     17     24024  493        device-tree    pagetypeinfo
11137  18     241    494        diskstats      partitions
11766  18905  243    541        driver         sched_debug
11771  18910  253    551        execdomains    schedstat
11999  18924  27186  6          fb             self
12     18925  27187  7          filesystems    slabinfo
12055  18926  27353  7193       fs             softirqs
12290  18927  274    8          interrupts     stat
12675  19     277    815        iomem          swaps
12679  19503  289    816        ioports        sys
12680  19507  35     9          irq            sysrq-trigger
12702  19893  36     9318       kallsyms       sysvipc
12727  2      37     9549       keys           thread-self
12733  20     38     9591       key-users      timer_list
12738  211    39     9763       kmsg           tty
12740  21173  40     9901       kpagecgroup    uptime
12742  21190  41     asound     kpagecount     version
13     220    414    buddyinfo  kpageflags     vmallocinfo
14     22252  42     bus        latency_stats  vmstat
14137  226    428    cgroups    loadavg        zoneinfo
14140  22723  432    cmdline    locks
1441   23     439    consoles   meminfo
15     236    44     cpu        misc
16     23984  47     cpuinfo    modules
[julius@hand ~]$
```

Problem 14: Rebooting or shutting down (1/2)

- **The userland shutdown and reboot binaries are missing**
- **Logged on remotely (no SYSRQ key)**

Problem 14: Rebooting or shutting down (2/2)

- **Magic SysRq is also possible via /proc/sysrq-trigger**
- **Be careful!!**
- **Alt+SysRq+Space shows available triggers**



```
root@pols:~# echo " " > /proc/sysrq-trigger && dmesg|tail -n 1
[95297.426965] sysrq: HELP : loglevel(0-9) reboot(b) crash(c) termi
nate-all-tasks(e) memory-full-oom-kill(f) kill-all-tasks(i) thaw-fi
lesystems(j) sak(k) show-backtrace-all-active-cpus(l) show-memory-u
sage(m) nice-all-RT-tasks(n) poweroff(o) show-registers(p) show-all
-timers(q) unraw(r) sync(s) show-task-states(t) unmount(u) force-fb
(V) show-blocked-tasks(w) dump-ftrace-buffer(z)
root@pols:~# echo "p" > /proc/sysrq-trigger && dmesg|tail
[95299.925963] CPU#3:   gen-PMC1 left:  0000000000000000
[95299.925964] CPU#3:   gen-PMC2 ctrl:  0000000000000000
[95299.925966] CPU#3:   gen-PMC2 count: 0000000000000000
[95299.925967] CPU#3:   gen-PMC2 left:  0000000000000000
[95299.925968] CPU#3:   gen-PMC3 ctrl:  0000000000000000
[95299.925970] CPU#3:   gen-PMC3 count: 0000000000000000
[95299.925971] CPU#3:   gen-PMC3 left:  0000000000000000
[95299.925972] CPU#3: fixed-PMC0 count: 0000000000000000
[95299.925974] CPU#3: fixed-PMC1 count: 0000ffffb8c2bf45
[95299.925975] CPU#3: fixed-PMC2 count: 0000000000000000
root@pols:~# 
```

The End: Questions?