18.16 The `/proc` filesytem

- Acts as an interface to internal data structures
- Use:
  - ♢ To obtain information about the system
  - ♢ To change certain kernel parameters at runtime
- Also contains one subdirectory for each process running on the system
- Named after the process id (PID) of the process
- Contents of `/proc` can change with different kernel versions
  - ♢ Shouldn't write programs that rely on it

18.17 Process specific subdirectories

- Each process subdirectory has following entries:

`cmdline`	Command line arguments
`cwd`	Link to the current working directory
`environ`	Values of environment variables
`exe`	Link to the executable of this process
`fd`	Directory containing all open file descriptors
`maps`	Memory maps (what memory the process has mapped from files)
`mem`	Memory held by this process
`root`	Link to the root directory of this process
`stat`	Process status
`statm`	Process memory status information
`status`	Process status in human readable form

18.18 Process Status

- To get the status of a process, just read `/proc/PID/status`:

```
$ cat /proc/14502/status
Name:   httpd
State:  S (sleeping)
Pid:    14502
PPid:   381
Uid:    500     500     500     500
Gid:    500     500     500     500
VmSize:     1716 kB
VmLck:         0 kB
VmRSS:       864 kB
VmData:      304 kB
VmStk:        32 kB
VmExe:       300 kB
VmLib:       864 kB
SigPnd: 00000000
SigBlk: 00000000
SigIgn: 00000000
SigCgt: 0040766b
```

- Shows almost the same information as `ps` because `ps` gets its info from `proc`

18.19 Process Memory Usage (`statm`)

- The `statm` file details process memory usage
- Its values have the following meanings:

`size`	total program size
`resident`	size of in memory portions
`shared`	number of the pages that are shared
`trs`	number of pages that are 'code'
`drs`	number of pages of data/stack
`lrs`	number of pages of library
`dt`	number of dirty pages

- Ratio `trs/data/library` is only approximate

18.20 Kernel data

- The following subdirectories give info on the running kernel
- Not all present on every system
  - ♢ depends on kernel config and loaded modules

`apm`	Advanced power management info
`cmdline`	Kernel command line
`cpuinfo`	Info about the CPU
`devices`	Available devices (block and character)
`dma`	Used DMA channels
`filesystems`	Supported filesystems
`interrupts`	Interrupt usage
`ioports`	I/O port usage
`kcore`	Kernel core image
`kmsg`	Kernel messages
`ksyms`	Kernel symbol table
`loadavg`	Load average
`locks`	Kernel locks
`meminfo`	Memory info
`misc`	Miscellaneous
`modules`	List of loaded modules
`mounts`	Mounted filesystems
`partitions`	Table of partitions known to the system
`rtc`	Real time clock
`slabinfo`	Slab pool info
`stat`	Overall statistics
`swaps`	Swap space utilization
`uptime`	System uptime
`version`	Kernel version

18.21 Interrupts In Use

- See `/proc/interrupts` to:
  - ♢ Check which interrupts are currently in use
  - ♢ Check what they are used for/by
- For example:

```
$ cat /proc/interrupts
 0:  310441744   timer
 1:        442   keyboard
 2:          0   cascade
 3:     259823 + serial
 4:       1974 + serial
 5:    5618945 + serial
 8:          1 + rtc
10:  140843983   3c509
13:          1   math error
14:  272072142 + ide0
15:  307923561 + ide1
```


18.22 IDE Devices (/proc/ide)

- Details all IDE devices known to the kernel
- One subdirectory for each device
- Each directory containing these files:

cache	The cache
capacity	Capacity of the medium
driver	Driver and version
geometry	Physical and logical geometry
identify	Device identify block
media	Media type
model	Device identifier
settings	Device setup
smart_thresholds	IDE disk management thresholds
smart_values	IDE disk management values

18.23 Networking (/proc/net)

- The files and their meanings:

arp	Kernel ARP table
dev	Network devices with statistics
dev_mcast	Lists the Layer2 multicast groups a device is listening to (interface index, label, number of references, number of bound addresses).
dev_stat	Network device status
ip_fwchains	Firewall chain linkage
ip_fwnames	Firewall chains
ip_masq	Directory containing the masquerading tables
ip_masquerade	Major masquerading table
netstat	Network statistics
raw	Raw device statistics
route	Kernel routing table
rpc	Directory containing rpc info
rt_cache	Routing cache
snmp	SNMP data
sockstat	Socket statistics
tcp	TCP sockets
tr_rif	Token ring RIF routing table
udp	UDP sockets
unix	UNIX domain sockets
wireless	Wireless interface data (Wavelan etc)
igmp	IP multicast addresses, which this host joined
psched	Global packet scheduler parameters
netlink	List of PF_NETLINK sockets
ip_mr_vifs	List of multicast virtual interfaces
ip_mr_cache	List of multicast routing cache
udp6	UDP sockets (IPv6)
tcp6	TCP sockets (IPv6)
raw6	Raw device statistics (IPv6)
igmp6	IP multicast addresses, which this host joineed (IPv6)
if_inet6	List of IPv6 interface addresses
ipv6_route	Kernel routing table for IPv6
rt6_stats	global IPv6 routing tables statistics
sockstat6	Socket statistics (IPv6)
snmp6	Snmp data (IPv6)

18.24 Networking 2 (/proc/net)

- Use /proc/net to see:
  - ♢ The network devices available in your system
  - ♢ How much traffic is routed over them
- For example:

```
$  cat /proc/net/dev
Inter-|   Receive                   |  Transmit
 face |packets errs drop fifo frame|packets errs drop fifo colls carrier
    lo:20664014    0    0    0    0 20664014    0    0    0     0    0
  eth0:93964796 618018 618018 616771 618018 83123181    1    0    0 49304   39
eth0:0:    141    0    0    0    0        1    0    0    0     0    0
eth0:1:    333    0    0    0    0        4    0    0    0     0    0
eth0:2:      0    0    0    0    0        0    0    0    0     0    0
eth0:3:      0    0    0    0    0        0    0    0    0     0    0
eth0:4:    212    0    0    0    0        3    0    0    0     0    0
```


18.25 SCSI info (/proc/scsi)

- To see a list of all recognized SCSI devices in /proc/scsi:

```
$ cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 00
  Vendor: QUANTUM  Model: XP34550W         Rev: LXY4
  Type:   Direct-Access                    ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 01 Lun: 00
  Vendor: SEAGATE  Model: ST34501W         Rev: 0018
  Type:   Direct-Access                    ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 02 Lun: 00
  Vendor: SEAGATE  Model: ST34501W         Rev: 0017
  Type:   Direct-Access                    ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 04 Lun: 00
  Vendor: ARCHIVE  Model: Python 04106-XXX Rev: 703b
  Type:   Sequential-Access                ANSI SCSI revision: 02
```

- One file for each adapter found in the system
- Info on controller, IRQ used, IO address range:

```
$ cat /proc/scsi/ncr53c8xx/0
General information:
 Chip NCR53C875, device id 0xf, revision id 0x4
 IO port address 0xec00, IRQ number 11
 Synchronous period factor 12, max commands per lun 4
```


18.26 Parallel Port (/proc/parport)

- Info on parallel ports
- One subdirectory for each port
- named after the port number (0,1,2, . . .)
- Contains four files:

autoprobe	Autoprobe results of this port
devices	Connected device modules
hardware	Port type, io-port, DMA, IRQ, etc
irq	Used interrupt, if any

18.27 Kernel Parameters (`/proc/sys`)

- Displays parameters within the kernel
- Allows you to *change* them
- Can tune and monitor kernel operation
- Be very careful, a reboot may be the only option after a mistake
- To change a value `echo` the new value into the file (see file handles example below)
- Superuser permission is required
- Can be automated via the init scripts
  - ♢ Should check kernel documentation when upgrading kernel to check the `/proc` information you use has not changed

18.28 File system data (/proc/fs)

- Info on file handles, inodes, dentry and quotas
- /proc/sys/fs currently contains these files:

dentry-state	Status of the directory cache
dquot-nr	Number of allocated and free disk quota entries
dquot-max	Maximum number of cached disk quota entries
file-nr	Number of allocated, used and maximum number of file handles
file-max	Maximum number of file handles that the Linux kernel will allocate
inode-state	Contains three actual numbers and four dummy values. Actual numbers are nr_inodes (inodes allocated), nr_free_inodes (free inodes), and preshrink (nonzero when the nr_inodes > inode-max and system needs to reduce inode list instead of allocating more)
inode-nr	Contains the first two items from inode-state
inode-max	Maximum number of inode handlers. Should be 3-4x > file-max, since stdin, stdout, and network sockets also need an inode struct to handle them
super-nr	Number of currently allocated super block handlers
super-max	Maximum number of super block handlers. Every mounted file system needs one, so more mounts need more of them

18.29 Example: Increase Maximum Filehandles

- Kernel allocates file handles dynamically, but doesn't free them while processes still run
- The default value maximum (`file-max`) is 4096
- To change it, just write a new number into the file:

```
# cat /proc/sys/fs/file-max
4096
# echo 8192 > /proc/sys/fs/file-max
# cat /proc/sys/fs/file-max
8192
```

- Useful for all customizable kernel parameters
- N.B. There is still a per process limit of open files (1024 by default) — can't be easily changed [2]

[2] To change it, edit the files `limits.h` and `fs.h` in the directory `/usr/src/linux/include/linux`. Change the definition of NR_OPEN and recompile the kernel.

18.30 General Kernel Parameters (/proc/sys/kernel)

- There are many general prarameters here and they vary from system to system
- The most commonly utilised covers the behaviour of `ctrl-alt-del`
  - ♢ When = 0, `ctrl-alt-del` is trapped and sent to `init(1)` to handle a graceful restart
  - ♢ When > 0, Linux produces an immediate reboot, without syncing dirty buffers
  - ♢ Occasionally `ctrl-alt-del` won't reach the kernel (e.g. intercepted by `dosemu`)
- Other files you might see, include:
  - ♢ `acct`
  - ♢ `domainname and hostname`
  - ♢ `osrelease, ostype and version`
  - ♢ `panic`
  - ♢ `sg-big-buff`
  - ♢ `modprobe`

18.31 Virtual Memory Subsystem (/proc/sys/vm)

- Typically used to set rather than read parameters
- Used for low-level tuning of the kernel's virtual memory (VM) subsystem
- Generally for wizards, i.e. supra-guru

18.32 Device Specific Parameters (/proc/sys/dev)

- A newish feature
- May not even exist on some systems
- Currently only support for CDROM drives
- Only one read-only file on CD-ROM drives attached to the system, e.g.

```
$ cat /proc/sys/dev/cdrom/info
CD-ROM information

drive name:           sr0  hdc
drive speed:           0    6
drive # of slots:      1    0
Can close tray:        1    1
Can open tray:         1    1
Can lock tray:         1    1
Can change speed:      1    1
Can select disk:       0    1
Can read multisession: 1    1
Can read MCN:          1    1
Reports media changed: 1    1
Can play audio:        1    1
```

- Example shows two drives, `sr0` and `hdc` with their features

18.33 Remote Procedure Calls (/proc/sys/sunrpc)

- Contains four files, enabling or disabling debugging for the RPC functions:
  - ♢ NFS
  - ♢ NFS-daemon
  - ♢ RPC
  - ♢ NLM
- Default values are 0
- Can be set to 1 to turn debugging on

18.34 Networking (/proc/sys/net)

- The interface to the networking parts of the kernel is located in /proc/sys/net
- Contains literally hundreds of parameters which can be read or set
- This table shows all possible subdirectories, some will not appear on every system:

core	General parameter	appletalk	Appletalk protocol
unix	Unix domain sockets	netrom	NET/ROM
802	E802 protocol	ax25	AX25
ethernet	Ethernet protocol	rose	X.25 PLP layer
ipv4	IP version 4	x25	X.25 protocol
ipx	IPX	token-ring	IBM token ring
bridge	Bridging	decnet	DEC net
ipv6	IP version 6		

- No time to discuss them all here

18.35 IPV4 settings (/proc/sys/net/ipv4)

- ICMP settings:
  - ♢ icmp_echo_ignore_all and icmp_echo_ignore_broadcasts

    Turn on (1) or off (0). First ignores ping of your host. Second ignores pings of your network. Can help tackle denial of service packet flooding attacks

  - ♢ icmp_destunreach_rate icmp_echoreply_rate icmp_paramprob_rate icmp_timeexeed_rate

    Set limits for sending ICMP packets to specific targets, depending on icmp type, i.e. can stop packet flooding *from* your host
- There are dozens of other IP and TCP settings . . . too many to discuss here
- See /usr/src/linux/Documentation/proc.txt for details

18.36 Special Topics Exercises

1. *Configuring LILO*

   (a) Put a copy of your existing Linux kernel on a floppy, then configure `lilo` to boot your machine from it. N.B. Do NOT do the next question until you are sure your boot disk works!

   (b) Configure `lilo` to boot your machine from a new Linux kernel on your hard drive.

   Ideally you should do this with a distinctively new kernel, such as the one made for the Kernel Internals module, but you could simply copy your current kernel with a new name.

2. *Using RPMs*

   (a) Use `rpm` from the command line to:

      i. Install a package
      ii. Update a package
      iii. Uninstall a package

   (b) If you have a distribution CD available:

      i. Find the main directory containing RPMs.
      ii. Work out and use the command string to put a complete list of all the packages' summary information and filenames into a file called `rpmlist.txt`

   (c) Verify your `setup` RPM.

   (d) With a colleague, draw up a list of other RPM packages containing files which have probably changed since installation. Verify them.

   (e) Imagine you suspect a system break-in has occurred. Use `rpm` to check:

      i. Whether such a break-in has occurred
      ii. How your files have been affected

   (f) Depending on what you have on your system, find out which packages are required to run `fvwm2` or another window manager

3. *Building And Installing Applications From Sources*

   (a) Install an application from sources provided, or indicated, by your tutor

4. *Using the `/proc` filesystem*

   (a) Print (to screen) simple info from `/proc` on:

      i. memory usage
      ii. cpu usage

   (b) Use `/proc` to get status info on the following processes:

      i. The shell you are currently working in
      ii. `syslogd`
      iii. `crond`

   (c) Use `/proc` to enable/disable:

      i. IP forwarding

ii. ICMP packet flooding from your host
iii. ICMP packet flooding of your network

(d) Pass parameters to the running kernel to:

i. Increase the maximum number of file handles available
ii. Change your hostname

N.B. Change back to your original hostname as soon as you have succeeded. Many other exercises on your course may depend on it.

18.37 Special Topics Solutions

1. *Configuring LILO*

   (a) Put a boot image on the floppy, then add something like the following to `lilo.conf`, before running `lilo` and rebooting:

   ```
   image=/boot/bzlinuz
           label=floppylinux
           root=/dev/fd0
           read-only
   ```

   (b) Put a boot image in the `boot` directory of your hard disk, then add something like the following to `lilo.conf`, before running `lilo` and rebooting:

   ```
   image=/boot/newlinuz
           label=newlinux
           root=/dev/hda1
           read-only
   ```

2. *Using RPMs*

   (a) Use something like the following commands:

      i. `$ rpm -i` *package*
      ii. `$ rpm -U` *package*
      iii. `$ rpm -e` *package*

   (b) If you have a distribution CD available:

      i. On Red Hat distributions it will usually be `/mnt/cdrom/RedHat/RPMS/`
      ii. `$ rpm -qilp *.rpm > rpmlist.txt`

   (c) `$ rpm -V setup`

   (d) Potentially hundreds of correct answers to this one. Dependent on host setup. On any system, the following files should really have changed:

      - `passwd`
      - `group`
      - `hosts.allow`
      - `hosts.deny`

      Find out which package these belong to using:

      `$ rpm -qf` *filename*

   (e) `$ rpm -Va`

   (f) `$ rpm -R` *package*

3. *Building And Installing Applications From Sources*

   There are several possible methods, but the most popular procedure does the following in the source directory:

   ```
   $ ./configure
   ...
   $ make
   ...
   $ su
   Password:
   $ make install
   ```

4. *Using the* `/proc` *filesystem*

   (a) i. `$ cat /proc/meminfo`
   ii. `$ cat /proc/cpuinfo`

   (b) Use `ps` or `top` to get the appropriate process IDs, then:

   `$ cat /proc/`*`PID`*`/status`

   (c) i.
   - On: `$ echo 1 > /proc/sys/net/ipv4/ip_forward`
   - Off `echo 0 > /proc/sys/net/ipv4/ip_forward`

   ii. See tutor
   iii. See tutor

   (d) E.g.

   i. `$ echo 8192 > /proc/sys/fs/file-max`
   ii.
   - Change: `$ echo` *`newname`* `> /proc/sys/kernel/hostname`
   - Undo: `$ echo` *`orignalname`* `> /proc/sys/kernel/hostname`