

# OWL Shell Command Guide

2020-04-03

# 1. Contents

OWL Shell Command Guide .....	1	2.30. gunzip.....	19
1. Contents.....	2	2.31. gzip.....	20
2. Commands.....	4	2.32. head.....	20
2.1. arp.....	4	2.33. hwclock.....	21
2.2. awk.....	4	2.34. chmod.....	21
2.3. backup.....	5	2.35. ifconfig.....	22
2.4. brctl.....	5	2.36. io.....	23
2.5. cat.....	6	2.37. ip.....	23
2.6. cd.....	7	2.38. iptables.....	24
2.7. cdmaat.....	7	2.39. kill.....	25
2.8. cdmapwr.....	8	2.40. killall.....	26
2.9. cmp.....	8	2.41. led.....	26
2.10. conntrack.....	8	2.42. ln.....	27
2.11. cp.....	10	2.43. logger.....	27
2.12. curl.....	10	2.44. lpm.....	28
2.13. date.....	11	2.45. ls.....	28
2.14. defaults.....	11	2.46. mac.....	29
2.15. dhcrelay.....	12	2.47. mkdir.....	30
2.16. dmesg.....	13	2.48. mount.....	30
2.17. echo.....	14	2.49. mv.....	31
2.18. email.....	14	2.50. nc.....	31
2.19. ethtool.....	15	2.51. netstat.....	32
2.20. find.....	15	2.52. nohup.....	32
2.21. free.....	16	2.53. ntpdate.....	33
2.22. fwupdate.....	16	2.54. openssl.....	33
2.23. grep.....	17	2.55. passwd.....	34
2.24. gsmat.....	17	2.56. pidof.....	34
2.25. gsmat2.....	18	2.57. ping.....	34
2.26. gsminfo.....	18	2.58. portd.....	35
2.27. gsmpwr.....	18	2.59. ps.....	35
2.28. gsmpwr2.....	19	2.60. pse.....	36
2.29. gsmsms.....	19	2.61. pwd.....	36

2.62.	<i>reboot</i> .....	36	2.78.	<i>stty</i> .....	45
2.63.	<i>report</i> .....	37	2.79.	<i>sync</i> .....	46
2.64.	<i>restore</i> .....	37	2.80.	<i>tail</i> .....	46
2.65.	<i>rm</i> .....	37	2.81.	<i>tar</i> .....	47
2.66.	<i>rmdir</i> .....	38	2.82.	<i>tcpdump</i> .....	47
2.67.	<i>route</i> .....	38	2.83.	<i>telnet</i> .....	48
2.68.	<i>scp</i> .....	39	2.84.	<i>top</i> .....	48
2.69.	<i>sed</i> .....	40	2.85.	<i>touch</i> .....	49
2.70.	<i>service</i> .....	41	2.86.	<i>traceroute</i> .....	49
2.71.	<i>sleep</i> .....	41	2.87.	<i>umount</i> .....	50
2.72.	<i>slog</i> .....	41	2.88.	<i>umupdate</i> .....	50
2.73.	<i>snmpget</i> .....	42	2.89.	<i>vi</i> .....	50
2.74.	<i>snmpset</i> .....	42	2.90.	<i>wc</i> .....	51
2.75.	<i>snmptrap</i> .....	43	2.91.	<i>wget</i> .....	51
2.76.	<i>status</i> .....	44	2.92.	<i>xargs</i> .....	52
2.77.	<i>split</i> .....	45			
3.	<i>Examples of Scripts</i> .....	52			
3.1.	<i>HTTP POST Method for a Remote Execution</i> .....	52			
3.2.	<i>Send SMS Remotely via Router's Web Page by HTTP POST Method</i> .....	52			
3.3.	<i>SMS Handling Using /var/scripts/sms</i> .....	53			
3.4.	<i>Send SMS to E-mail</i> .....	53			
3.5.	<i>SMS Command 1</i> .....	54			
3.6.	<i>SMS Command 2</i> .....	54			
3.7.	<i>Send Information Email 1</i> .....	55			
3.8.	<i>Send Information SMNP Trap 1</i> .....	55			
3.9.	<i>Send Information Email 2</i> .....	55			
3.10.	<i>Send Information SMNP Trap 2</i> .....	56			
3.11.	<i>Automatic Reboot</i> .....	56			
3.12.	<i>Switch Between WAN and PPP</i> .....	56			
3.13.	<i>How to Use Unsupported FTDI Chip</i> .....	57			

## 2. Commands

### 2.1. arp

The arp program displays and modifies the Internet-to-Ethernet address translation tables used by the address resolution protocol.

#### Synopsis:

```
arp [-a <hostname>] [-s <hostname> <hw_addr>] [-d <hostname>] [-v] [-n] [-i <if>] [-D <hostname>] [-A ] [-f <filename>]
```

#### Options:

Option	Description
-a	The entries will be displayed in alternate (BSD) style.
-s	Manually create an ARP address mapping entry for host hostname with hardware address set to hw_addr.
-d	Remove any entry for the specified host.
-v	Tell the user what is going on by being verbose.
-n	Shows numerical addresses instead of trying to determine symbolic host, port or user names.
-i	Select an interface.
-D	Use the interface ifa's hardware address.
-f	Similar to the -s option, only this time the address info is taken from file filename set up. The name of the data file is very often <i>/etc/ethers</i> , but this is not official. If no filename is specified <i>/etc/ethers</i> is used as default. The format of the file is simple; it only contains ASCII text lines with a hardware address and a hostname separated by whitespace. Additionally the pub, temp and netmask flags can be used.

Table 1: arp options

With no flags, the program displays the current ARP entry for hostname. The host may be specified by name or by number, using Internet dot notation. For detail description of this command, visit Linux manual pages.

#### Examples:

View arp table without translating IP addresses to domain names

```
arp -n
```

### 2.2. awk

*Awk* program scans each input file for lines that match any of a set of patterns specified literally in program-text or in one or more files specified as -f progfile.

**Synopsis:**

```
awk [-v] [-F] [-f] ... [<program-text>] [<file> ...]
```

**Options:**

Option	Description
-v	Assign the value <i>val</i> to the variable <i>var</i> , before execution of the program begins. Such variable values are available to the BEGIN block of an AWK program.
-F	Use for the input field separator (the value of the FS predefined variable).
-f	Read the AWK program source from the file <i>program-file</i> , instead of from the first command line argument. Multiple -f (or --file) options may be used.

Table 2: awk options

**Examples:**

Show IP address of Gateway

```
route -n | awk '/^01.01.01.0/ { print $2 }'
```

## 2.3. backup

This program can be used to backup configuration of the router. The configuration is written to the standard output and can be redirected to a file, see example below. The stored configuration can be restored from the file by *restore* command.

**Synopsis:**

```
backup [<options>] [<filename>]
```

**Options:**

Option	Description
-c	Backup configuration excluding data of user accounts.
-u	Backup configuration data of user accounts only.
-a	Backup the whole configuration.

**Examples:**

Table 3: backup options

Backup the whole configuration to the file */tmp/my.cfg*.

```
backup -a > /tmp/my.cfg
```

## 2.4. brctl

*Brctl* command can be used to set up, maintain, and inspect the Ethernet

bridge configuration in the Linux kernel.

An Ethernet bridge is a device commonly used to connect different networks of Ethernets together, so that these Ethernets will appear as one Ethernet to the participants.

Each of the Ethernets being connected corresponds to one physical interface in the bridge. These individual Ethernets are bundled into one bigger ('logical') Ethernet, this bigger Ethernet corresponds to the bridge network interface.

### Synopsis:

```
brctl [<commands>]
```

### Options:

Command	Parameters	Description
addbr	<bridge>	Add bridge
delbr	<bridge>	Delete bridge
addif	<bridge> <device>	Add interface to bridge
delif	<bridge> <device>	Delete interface from bridge
setageing	<bridge> <time>	Set ageing time
setbridgepri	<bridge> <prio>	Set bridge priority
setfd	<bridge> <time>	Set bridge forward delay
sethello	<bridge> <time>	Set hello time
setmaxage	<bridge> <time>	Set max message age
setpathcost	<bridge> <port> <cost>	Set path cost
setportprio	<bridge> <port> <prio>	Set port priority
show		Show list of bridges
showmacs	<bridge>	Show list of mac address
showstp	<bridge>	Show bridge stp info
stp	<bridge> {on   off}	Turn stp on/off

Table 4: brctl commands

### Examples:

Create bridge between eth0 and eth1.

```
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 eth1
```

## 2.5. cat

This command concatenates files and print on the standard output.

**Synopsis:**

```
cat [-u] [<file>] ...
```

**Options:**

Option	Description
-u	Ignored since unbuffered I/O is always used.

Table 5: cat options

**Examples:**

View the contents of file `/proc/tty/driver/spear_serial` (info about serial ports of v2 routers).

```
cat /proc/tty/driver/spear_serial
```

Copy the contents of the router configuration files in `/tmp/my.cfg`.

```
cat /etc/settings.* > /tmp/my.cfg
```

## 2.6. cd

This command can be used to change the current working directory.

**Synopsis:**

```
cd [-P] [-L] [<directory>]
```

**Options:**

Option	Description
-P	Do not follow symbolic links
-L	Follow symbolic links (default)

Table 6: cd options

**Examples:**

Move to home directory (`/root`).

```
cd
```

Move to directory `/mnt`.

```
cd /mnt
```

## 2.7. cdmaat

This program can be used for sending AT command to CDMA module if available (equivalent of the `gsmat` command).

**Synopsis:**

```
cdmaat <AT command>
```

## 2.8. cdmapwr

The program can be used to control the supply of CDMA module if available (equivalent of the gsmpr command).

### Synopsis:

```
cdmapwr [on | off]
```

## 2.9. cmp

The *cmp* utility compares two files of any type and writes the results to the standard output.

### Synopsis:

```
cmp [-l] [-s] <file1> <file2> [<skip1> [<skip2>]]
```

### Options:

Option	Description
-l	Print the byte number (decimal) and the differing byte values (octal) for each difference.
-s	Print nothing for differing files; return exit status only.

Table 7: cmp options

By default, *cmp* is silent if the files are the same; if they differ, the byte and line number at which the first difference occurred is reported. Bytes and lines are numbered beginning with one. If *<file2>* is not specified, standard input is used instead.

The optional arguments *<skip1>* and *<skip2>* are the byte offsets from the beginning of *<file1>* and *<file2>* respectively, where the comparison will begin. The offset is decimal by default, but may be expressed as a hexadecimal or octal value.

## 2.10. conntrack

This program is the user interface to the netfilter connection tracking system.

### Synopsis:

```
conntrack [commands] [option]
```

### Options:

Command	Description
-L [table] [option]	List conntrack or expectation table
-G [table]	Get conntrack or expectation
-D [table]	Delete conntrack or expectation



-I [table]	Create a contrack or expectation
-U [table]	Update a contrack
-E [table]	Show events
-F [table]	Flush table

Table 8: contrack comands

Table	Description
contrack	This is the default table. It contains a list of all currently tracked connections through the system.
expect	This is the table of expectations. Connection tracking expectations are the mechanism used to "expect" RELATED connections to existing ones.

Table 9: contrack tables

Option	Description
-n <ip>	Source NAT ip
-g <ip>	Destination NAT ip
-m <mark>	Set mark
-e <eventmask>	Event mask, eg. NEW,DESTROY
-z	Zero counters while listing
-o <type[...]>	Output format, eg. xml

Table 10: contrack options

Option	Description
--tuple-src <ip>	Source address in expect tuple
--tuple-dst <ip>	Destination address in expect tuple
--mask-src <ip>	Source mask address
--mask-dst <ip>	Destination mask address

Table 11: expectation options

Option	Description
-s <ip>	Source address from original direction
-d <ip>	Destination address from original direction
-r <ip>	Source adres from reply direction
-q <ip>	Destination address from reply direction
-p <proto>	Layer 4 Protocol, eg. 'tcp'
-f <proto>	Layer 3 Protocol, eg. 'ipv6'
-t <timeout>	Set timeout
-u <status>	Set status, eg. ASSURED

Table 12: contrack and expectation options

### Examples:

Display content of contrack table.

```
contrack -L
```

Delete content of contrack table.

```
contrack -F
```

## 2.11. cp

This command can be used to copy files and directories.

### Synopsis:

```
cp [<option>] <source> <dest>
```

### Options:

Option	Description
-a	Preserve the all attributes
-d, -P	Never follow symbolic links
-H, -L	Follow command-line symbolic links
-p	Preserve the mode, ownership, timestamps attributes
-f	If an existing destination file cannot be opened, remove it and try again
-i	Prompt before overwrite
-R, -r	Copy directories recursively

Table 13: cp options

### Examples:

Copy the system log to directory /mnt.

```
cp /var/log/messages* /mnt
```

Copy configuration profile "Alternative 1" to profile "Standard".

```
cp -r /etc/alt1/* /etc
```

## 2.12. curl

Curl (transfer a URL) is a tool to transfer data from or to a server, using one of the supported protocols (DICT, FILE, FTP, FTPS, GOPHER, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMTP, SMTPS, TELNET and TFTP). It is an alternative to `wget` (see chapter 1.92).

### Synopsis:

```
curl [options...] <url>
```

### Options:

Type `curl --help` for options to show in the command line or visit online manual page at <http://curl.haxx.se/docs/manpage.html>

## 2.13. date

This command can be used to display the current time in the given FORMAT, or set the system date (and time).

### Synopsis:

```
date [-R] [-d <string>] [-s] [-r <file>] [-u] [MMDDhhmm[[CC]YY][.ss]]
```

### Options:

Option	Description
-R	Output date and time in RFC 2822 format
-d <string>	Display time described by STRING, not 'now'
-s	Set time described by STRING
-r <file>	Display the last modification time of FILE
-u	Print or set Coordinated Universal Time

Table 14: date options

### Examples:

Display the current date and time.

```
date
```

Setting the date and time on December 24, 2011 20:00.

```
date 122420002011
```

## 2.14. defaults

The script can be used to restore the default configuration.

### Synopsis:

```
defaults
```

## 2.15. df

This command can be used to view report file system disk space usage. FORMAT, or set the system date (and time).

### Synopsis:

```
df [-k] [<filesystem> ...]
```

### Options:

Option	Description
-k	Print sizes in kilobytes

Table 15: df options

## 2.16. dhcrelay

The Dynamic Host Configuration Protocol (DHCP) Relay Agent, `dhcrelay`, provides a means for relaying DHCP and BOOTP requests from a subnet to which no DHCP server is directly connected to one or more DHCP servers on other subnets. It supports both DHCPv4/BOOTP and DHCPv6 protocols (v3 routers only).

### Synopsis:

```
dhcrelay [-4] [-d] [-q] [-a] [-D] [-A <length>] [-c <hops>] [-p <port>] [-pf <pid-
file>] [--no-pid] [-m append|replace|forward|discard] [-i interface0 [ ... -i
interfaceN] server0 [ ... serverN]
```

```
dhcrelay -6 [-d] [-q] [-I] [-c <hops>] [-p <port>] [-pf <pid-file>] [--no-
pid] -l lower0 [ ... -l lowerN] -u upper0 [ ... -u upperN]
```

### Options:

Option	Description
-a	Append an agent option field to each request before forwarding it to the server. Agent option fields in responses sent from servers to clients will be stripped before forwarding such responses back to the client.
-A <length>	Specify the maximum packet size to send to a DHCPv4/BOOTP server. This might be done to allow sufficient space for addition of relay agent options while still fitting into the Ethernet MTU size.
-D	Drop packets from upstream servers if they contain Relay Agent Information options that indicate they were generated in response to a query that came via a different relay agent.
-i <ifname>	Listen for DHCPv4/BOOTP queries on interface ifname. Multiple interfaces may be specified by using more than one -i option. If no interfaces are specified on the command line, <code>dhcrelay</code> will identify all network interfaces, eliminating non-broadcast interfaces if possible, and attempt to listen on all of them.
-m <option>	Control the handling of incoming DHCPv4 packets which already contain relay agent options.

Table 16: `dhcrelay` options available in DHCPv4 mode only

Option	Description
-c <hops>	Maximum hop count. When forwarding packets, <code>dhcrelay</code> discards packets which have reached a hop count of <hops>. Default is 10. Maximum is 255.
-d	Force <code>dhcrelay</code> to run as a foreground process.

-p <port>	Listen and transmit on port <port>. Default is port 67 for DHCPv4, or port 547 for DHCPv6.
-q	Quiet mode.

Table 17: dhcrelay options available for both DHCPv4 and DHCPv6

Option	Description
-I	Force use of the DHCPv6 Interface-ID option. This option is automatically sent when there are two or more downstream interfaces in use, to disambiguate between them.
-l	Specifies the "lower" network interface for DHCPv6 relay mode: the interface on which queries will be received from clients or from other relay agents.
-u	Specifies the "upper" network interface for DHCPv6 relay mode: the interface to which queries from clients and other relay agents should be forwarded.

Table 18: dhcrelay options available in DHCPv6 mode only:

For help in command line type `dhcrelay -h`. For more information on this command, look up the man page on the Internet.

It is necessary to set at least **two interfaces** for sending/listening – see an Example below.

### Examples:

The DHCP Relay Agent listens for DHCPv4 or DHCPv6 queries from clients on interfaces eth0 and eth1, passing them along to "upstream" server <server ip>. When a reply is received from upstream, it is multicast or unicast back downstream to the source of the original request. `dhcrelay -i eth1 -i eth0 <server ip>`

## 2.17. dmesg

This command can be used to display the Kernel log messages.

### Synopsis:

```
dmesg [-R] [-T] [-c]
```

### Options:

Option	Description
-R	Display relative time since the router booted in <i>sec.nanosec</i> format.
-T	Display human-readable timestamp in <i>YYYY-MM-DD hh:mm:ss</i> format.
-c	Read and clear all messages.

Table 19: dmesg options

**Examples:**

Display latest Kernel log messages and subsequent deletion of the Kernel ring buffer.

```
dmesg -c
```

Display latest Kernel log messages including the human-readable timestamp.

```
dmesg -T
```

## 2.18. echo

This command prints the strings to standard output.

**Synopsis:**

```
echo [-n] [-e] [-E] [<string> ...]
```

**Options:**

Option	Description
-n	Do not output the trailing newline
-e <level>	Enable interpretation of backslash escapes
-E <size>	Disable interpretation of backslash escapes (default)

Table 20: echo options

**Examples:**

Switch profile to "Standard".

```
echo "PROFILE="
> /etc/settings
reboot
```

Switch profile to "Alternative 1".

```
echo "PROFILE=alt1"
> /etc/settings
reboot
```

Send a sequence of bytes 0x41,0x54,0x0D,0x0A to serial line (write data in octal).

```
echo -n -e "\101\124\015\012" > /dev/ttyS0
```

## 2.19. email

The program can be used for sending email.

**Synopsis:**

```
email -t <to> [-s <subject>] [-m <message>] [-a <attachment>] [-r
<retries>]
```

**Options:**

Option	Description
-t	Email of recipient
-s	Subject of email
-m	Message of email
-a	Attachment of email
-r	Number of retries

Table 21: email options

**Examples:**

Send system logs to the address john.doe@email.com.

```
email -t john.doe@email.com -s "System Log" -a /var/log/messages
```

## 2.20. ethtool

This command can be used to display or change Ethernet card settings.

**Synopsis:**

```
ethtool [<option>...] <devname> [<commands>]
```

**Options:**

For detail description this command, visit Linux manual pages.

**Examples:**

View the status of the interface eth0.

```
ethtool eth0
```

Switch interface eth0 to mode 10 Mbit/s, half duplex.

```
ethtool -s eth0 speed 10 duplex half autoneg off
```

Turn on autonegacion on the interface eth0.

```
ethtool -s eth0 autoneg on
```

## 2.21. find

Command to search for files in a directory hierarchy.

**Synopsis:**

```
find [<path>...] [<expression>]
```

**Options:**

The default path is the current directory, default expression is '-print'. Type `find --help` for help or look up online man page for more detailed description. Expression may consist of:

Option	Description
-follow	Dereference symbolic links

<code>-name &lt;pattern&gt;</code>	File name (leading directories removed) matches <code>&lt;pattern&gt;</code>
<code>-print</code>	Print (default and assumed)
<code>-type X</code>	Filetype matches X (where X is one of: f,d,l,b,c,...)
<code>-perm &lt;perms&gt;</code>	Permissions match any of (+NNN); all of (-NNN); or exactly (NNN)
<code>-mtime &lt;days&gt;</code>	Modified time is greater than (+N); less than (-N); or exactly (N) days
<code>-mmin &lt;mins&gt;</code>	Modified time is greater than (+N); less than (-N); or exactly (N) minutes
<code>-exec &lt;cmd&gt;</code>	Execute command with all instances of {} replaced by the files matching <code>&lt;expression&gt;</code>

Table 22: find expressions

### Examples:

Search for files in your home directory which have been modified in the last twenty-four hours.

```
find $HOME -mtime 0
```

Search for files which have read and write permission for their owner, and group, but which other users can read but not write to.

```
find . -perm 664
```

## 2.22. free

This command can be used to display information about free and used memory.

### Synopsis:

```
free
```

## 2.23. fwupdate

This program can be used for router's firmware update.

### Synopsis:

```
fwupdate [-i <filename> [-h] [-n]] [-f]
```

### ptions:

Option	Description
<code>-i</code>	File of the new firmware, filename has to be specified
<code>-h</code>	HTML output (used when called from web configuration)
<code>-n</code>	Do not reboot after firmware update
<code>-f</code>	finish update procedures, called by default

Table 23: fwupdate options



## 2.24. grep

*Grep* program searches the named input FILEs (or standard input if no files are named, or the file name – is given) for lines containing a match to the given PATTERN. By default, *grep* prints the matching lines.

### Synopsis:

```
grep [<options> ...] <pattern> [<file> ...]
```

### Options:

Option	Description
-H	Print the filename for each match
-h	Suppress the prefixing of filenames on output when multiple files are searched
-i	Ignore case distinctions
-l	Suppress normal output; instead print the name of each input file from which output would normally have been printed
-L	Suppress normal output; instead print the name of each input file from which no output would normally have been printed
-n	Prefix each line of output with the line number within its input file
-q	Quiet; do not write anything to standard output. Exit immediately with zero status if any match is found, even if an error was detected. Also see the -s or --no-messages option.
-v	Invert the sense of matching, to select non-matching lines
-s	Suppress error messages about nonexistent or unreadable files
-c	Suppress normal output; instead print a count of matching lines for each input file
-f	Obtain patterns from FILE, one per line
-e	Use PATTERN as the pattern; useful to protect patterns beginning with –
-F	Interpret PATTERN as a list of fixed strings, separated by new lines, any of which is to be matched

Table 24: grep options

### Examples:

See all lines of system log in which occurs the word "error".

```
grep error /var/log/messages
```

View all processes whose name the contents of the string "ppp".

```
ps | grep ppp
```

## 2.25. gsmat

This program can be used for sending AT command to GSM module.

**Synopsis:**

```
gsmat <AT command>
```

**Examples:**

Determine the type and firmware version of GSM module.

```
gsmat ATI
```

Determine the IMEI code of module.

```
gsmat "AT+GSN"
```

## 2.26. gsmat2

This program can be used for sending AT command to second GSM module if available.

**Synopsis:**

```
gsmat2 <AT command>
```

## 2.27. gsminfo

This program can be used to display information about the signal quality.

**Synopsis:**

```
gsminfo
```

**Options:**

Option	Description
PLMN	Code of operator
Cell	The cell to which the router is connected
Channel	The channel on which the router communicates
Level	The signal quality of the selected cell
Neighbours	Signal quality of neighboring hearing cells
Uptime	Time to establish PPP connection

Table 25: Description of GSM information

## 2.28. gsmpwr

This program can be used to control the power supply of the cellular module.

**Synopsis:**

```
gsmpwr [on | off | shutdown]
```

**Examples:**

Turn on the power for the cellular module.

```
gsmprw on
```

Turn off the power for the cellular module.

```
gsmprw off
```

Shutdown the cellular module by an AT command and turn off the power for it. Please note that execution of this command may take a few seconds.

```
gsmprw shutdown
```

It is highly recommended to use the *gsmprw shutdown* command prior to use the *gsmprw off* command to increase the lifetime of the cellular module.

## 2.29. gsmprw2

This program can be used to control the power supply of the second cellular module in case this module is installed. For usage examples see the [gsmprw](#) command.

Synopsis:

```
gsmprw2 [on | off | shutdown]
```

It is highly recommended to use the *gsmprw2 shutdown* command prior to use the *gsmprw2 off* command to increase the lifetime of the cellular module.

## 2.30. gsmSMS

This program can be used to send SMS message.

Synopsis:

```
gsmSMS <phone number> <text>
```

### Examples:

Send SMS "Hello word" on telephone number +420123456789.

```
gsmSMS +420123456789 "Hello word"
```

## 2.31. gunzip

This program can be used to decompress FILE (or standard input if filename is '-').

Synopsis:

```
gunzip [-c] [-f] [-t] <filename>
```

Options:

Option	Description
-c	Write output on standard output
-f	Force decompression even if the file has multiple links or the corresp. file already exists, or if the compressed data is read from or written to a terminal.
-t	Test. Check the compressed file integrity.

Table 26: gunzip options

### Examples:

Decompression of file test.tar.gz (creates file test.tar).

```
gunzip test.tar.gz
```

## 2.32. gzip

This program can be used to compress FILE with maximum compression.

Synopsis:

```
gzip [-c] [-d] [-f] <filename>
```

Options:

Option	Description
-c	Write output on standard output
-d	Decompress
-f	Force compression even if the file has multiple links or the corresponding file already exists, or if the compressed data is read from or written to a terminal

Table 27: gzip options

### Examples:

Compression of file test.tar (creates file test.tar.gz).

```
gzip test.tar
```

## 2.33. head

This program prints first 10 lines of each file to standard output. With more than one file, precede each with a header giving the file name. With no file, or when file is a dash ("-"), read standard input.

Synopsis:

```
head [<option(s)>] [<file(s)>]
```

Options:

Option	Description
--------	-------------

-n NUM	Print first NUM lines instead of first 10.
-c NUM	Output the first NUM bytes.
-q	Never output headers giving file names.
-v	Always output headers giving file names.

Table 28: head options

## 2.34. hwclock

This program can be used to query and set the hardware clock (RTC).

### Synopsis:

```
hwclock [-r] [-s] [-w] [-u] [-l]
```

### Options:

Option	Description
-r	Read hardware clock a print result
-s	Set the System Time from the Hardware Clock
-w	Set the Hardware Clock to the current System Time
-u	The hardware clock is kept in coordinated universal time
-l	The hardware clock is kept in local time

Table 29: hwclock options

### Examples:

Set the hardware clock to the current system time.

```
hwclock -w -u
```

## 2.35. chmod

This command can be used to change file mode bits.

### Synopsis:

```
chmod [-R] <mode> <filename>
```

### Options:

Option	Description
-R	Change files and directories recursively

Table 30: chmod options

### Examples:

Settings rights (permit execution) of script /tmp/script.

```
chmod 755 /tmp/script
```

## 2.36. ifconfig

This command can be used to configure a network interface.

### Synopsis:

```
ifconfig [-a] <interface> [<option> ...]
```

### Options:

Option	Description
<code>broadcast &lt;addr.&gt;</code>	If the address argument is given, set the protocol broadcast address for this interface.
<code>pointtopoint &lt;ad.&gt;</code>	This keyword enables the point-to-point mode of an interface, meaning that it is a direct link between two machines with nobody else listening on it.
<code>netmask &lt;address&gt;</code>	Set the IP network mask for this interface.
<code>dstaddr &lt;address&gt;</code>	Set the remote IP address for a point-to-point link (such as PPP).
<code>metric &lt;NN&gt;</code>	This parameter sets the interface metric.
<code>mtu &lt;NN&gt;</code>	This parameter sets the Maximum Transfer Unit of an interface.
<code>trailers</code>	This flag used to cause a non-standard encapsulation of inet packets on certain link levels.
<code>arp</code>	Enable or disable the use of the ARP protocol on this interface.
<code>allmulti</code>	Enable or disable all-multicast mode. If selected, all multicast packets on the network will be received by the interface.
<code>multicast</code>	Set the multicast flag on the interface. This should not normally be needed as the drivers set the flag correctly them-selves.
<code>promisc</code>	Enable or disable the promiscuous mode of the interface. If selected, all packets on the network will be received by the interface.
<code>txqueuelen &lt;NN&gt;</code>	Set the length of the transmit queue of the device.
<code>up   down</code>	This flag causes the interface to be activated.   This flag causes the driver for this interface to be shut down.

Table 31: ifconfig options

### Examples:

View the status of all interfaces.

```
ifconfig
```

Activation of loopback with IP address 127.0.0.1/8.

```
ifconfig lo up
```

Activation of virtual interface eth0:0 with IP address 192.168.2.1/24.

```
ifconfig eth0:0 192.168.2.1 netmask 255.255.255.0 up
```

## 2.37. io

This program can be used to read binary inputs and to control binary outputs of the router. If installed, it also supports an expansion ports of the router.

### Synopsis:

```
io [get <pin>] | [set <pin> <value>]
```

### Options:

Option	Description
get	Get the state of input
set	Set the state of output

Table 32: io options

### Examples:

Get the state of digital input BIN0.

```
io get bin0
```

Get the state of analog input AN1 on expansion port XC-CNT.

```
io get an1
```

Get the state of counter input CNT1 on expansion port XC-CNT.

```
io get cnt1
```

Set the state of binary output OUT0 to 1.

```
io set out0 1
```

## 2.38. ip

This command can be used to configure a network interface or show the current configuration. Type `ip -help` for help in the terminal.

The SPECTRE v3 routers support more ip options and commands (options: `-d[etails]`, `-t[imestamp]`, `-b[atch] <filename>`, `-rc[vbuf]`; objects: `addrlabel`, `ntable`, `tuntap`, `mrule`, `netns`, `l2tp`, `tcp_metrics`, `token`). For information how to use, type `ip <object> help`, for detailed description of all options, visit Linux manual pages or look up them online.

### Synopsis:

```
ip [ <options> ] <object> { <command> | help }
```

### Options:

Option	Description
<code>-V[ersion]</code>	Print the version of the ip utility and exit
<code>-s[tatistics]</code>	Output more information. If the option appears twice or more, the amount of information increases.

<code>-r[esolve]</code>	use the system's name resolver to print DNS names instead of host addresses
<code>-f[amily] &lt;family&gt;</code>	Specifies the protocol family to use. The protocol family identifier can be one of <code>inet</code> , <code>inet6</code> , <code>bridge</code> , <code>ipx</code> , <code>dnet</code> or <code>link</code> .
<code>-o[neline]</code>	output each record on a single line, replacing line feeds with the <code>'\'</code> character

Table 33: ip options

Object	Description
<code>link</code>	network device
<code>addr</code>	protocol (IP or IPv6) address on a device
<code>route</code>	routing table entry
<code>rule</code>	rule in routing policy database
<code>neigh</code>	manage ARP or NDISC cache entries
<code>tunnel</code>	tunnel over IP
<code>maddr</code>	multicast address
<code>mroute</code>	multicast routing cache entry
<code>monitor</code>	watch for netlink messages
<code>xfrm</code>	manage IPsec policies

Table 34: ip objects

### Examples:

View the status of all interfaces.

```
ip link show
```

View the route table.

```
ip route list
```

Add routing networks 192.168.3.0/24 through interface eth0.

```
ip route add 192.168.3.0/24 dev eth0
```

Add routing IP address 192.168.3.1 through gateway 192.168.1.2.

```
ip route add 192.168.3.1 via 192.168.1.2
```

Add default gateway 192.168.1.2.

```
ip route add default via 192.168.1.2
```

## 2.39. iptables

This command can be used as an administration tool for IP packets filtering and NAT.

### Synopsis:

```
iptables [<options>]
```

### Options:

For detail description of this command type `iptables -h` or visit Linux manual pages.



- "DSCP" target and "dscp" match extension is supported – it is possible to configure and use QoS based on marked packets.
- "CONNMARK" target and "connmark" match extension is supported – it sets the netfilter mark value associated with a connection unlike MARK target which is used to set the netfilter mark value associated with the packet. With the CONNMARK target you can mark all the packets of a connection or related to a connection with the same mark. Another usefull use of CONNMARK is that you can mark packets using the criteria that only matches with the first packet.
- "string" match extension is supported. This modules matches a given string by using some pattern matching strategy. See man pages for iptables for more information.
- "statistic" module is supported. This module matches packets based on some statistic condition. It supports the "nth" and "random" distinct modes settable with the `--mode` option. Example: `iptables -I INPUT -p icmp -m statistic --mode nth --every 2 --packet 0 -j DROP`. See more in iptables man pages.

### Examples:

Redirect incoming TCP connections to port 8080 on IP address 192.168.1.2 and port 80.

```
/sbin/iptables -t nat -A prenat -p tcp --dport 8080 -j DNAT --
to-destination 192.168.1.11:80
```

```
/sbin/iptables -t mangle -A pre\_nat -p tcp --dport 8080 -j ACCEPT
```

#### Example of using DSCP with iptables:

```
iptables -t mangle -I POSTROUTING -p tcp --dport 81 -j DSCP -
-set-dscp 0x0a iptables -t mangle -I POSTROUTING -m dscp --
dscp 0x0a -j MARK --set-mark 81
```

## 2.40. kill

This command can be used to terminate process.

### Synopsis:

```
kill [ -<signal> ] <process-id> [
<process-id> ...] kill -l
```

### Options:

Option	Description
-l	Print a list of signal names. These are found in /usr/include/linux/signal.h

Table 35: kill options

### Examples:

End the process with PID 1234 by sending signal SIGTERM.

```
kill 1234
```

End the process with PID 1234 by sending signal SIGKILL.

```
kill -9 1234
```

## 2.41. killall

This command can be used to kill all process with process name.

Synopsis:

```
killall [ -q ] [ -<signal> ] <process-name> [<process-name> ...]
```

Options:

Option	Description
-l	Print a list of signal names. These are found in /usr/include/linux/signal.h
-q	Do not complain if no processes were killed

Table 36: killall options

**Examples:**

End the all processes with name pppd by sending signal SIGTERM.

```
killall pppd
```

End the all processes with name pppd by sending signal SIGKILL.

```
killall -9 pppd
```

## 2.42. led

This program can be used to control the USR or PWR LED on the front panel of the router.

Synopsis:

```
led [-p] [-u] <command>
```

Options:

Option	Description
-p	Control of PWR LED
-u	Control of USR LED (default)

Table 37: led options

Commands:

Command	Description
on	Power on the LEDn
off	Power off the LED
slow	Start blinking the LED slowly

fast	Start blinking the LED fast
------	-----------------------------

Table 38: led commands

**Examples:**

Turn on the USR LED.

```
led on
```

Start blinking slowly with the USR LED.

```
led slow
```

## 2.43. ln

This program can be used to make links between files.

**Synopsis:**

```
ln [ option ] < target > ...< link_name > | < directory >
```

**Options:**

Option	Description
-s	Make symbolic links instead of hard links
-f	Remove existing destination files
-n	No dereference symlinks – treat like normal file
-b	Make a backup of the target (if exists) before link operation
-S	Use suffix instead of ~ when making backup files

Table 39: ln options

**Examples:**

Creating a symbolic link to file /var/log/messages called my.log.

```
ln -s /var/log/messages my.log
```

## 2.44. logger

This program makes entries in the system log. It provides a shell command interface to the system log module.

**Synopsis:**

```
logger [ option ] [ message ...]
```

**Options:**

Option	Description
-i	Log the process id of the logger process with each line
-s	Log the message to standard error, as well as the system log

-f <file>	Log the specified file
-p <priority>	Enter the message with the specified priority. The priority may be specified numerically or as a facility.level pair.
-t <tag>	Mark every line in the log with the specified tag
-u <socket>	Write to socket as specified with socket instead of builtin syslog routines
-d	Use a datagram instead of a stream connection to this socket

Table 40: logger options

### Examples:

Send the message System rebooted to the syslogd daemon.

```
logger System rebooted
```

Send the message System going down immediately!!! to the syslog daemon, at the emerg level and user facility.

```
logger -p user.emerg "System going down immediately!!!"
```

## 2.45. lpm

This program can be used to switch the router into the Low Power Mode. The router than can be awoken up by an event according to the specified parameters. The first option is to wake up the router at the time the specified time period has expired and the second option is to wake up the router by activating the binary input. If both parameters were specified, the router will wake up at the time when the first event comes.

### Synopsis:

```
lpm [-b] [-i <interval>]
```

### Options:

Option	Description
-b	Wake up the router by activating the binary input. On <i>SmartFlex</i> routers on input BIN1, on <i>SmartStart</i> and <i>ICR-3200</i> series routers on input BIN0.
-i	Wake up the router at the time when the specified time period has expired. The time interval is specified in seconds.

Table 41: lpm options

## 2.46. ls

This program can be used to list directory contents.

### Synopsis:

```
ls [ option ] < filename > ...
```

Options:

Option	Description
-l	List files in a single column
-A	Do not list implied . and ..
-a	Do not hide entries starting with .
-C	List entries by columns
-c	With -l: show ctime
-d	List directory entries instead of contents
-e	List both full date and full time
-i	List the i-node for each file
-l	Use a long listing form
-n	List numeric UIDs and GIDs instead of names
-L	List entries pointed to by symbolic links
-r	Sort the listing in reverse order
-S	Sort the listing by file size
-s	List the size of each file, in blocks
-t	With -l: show modification time
-u	With -l: show access time
-v	Sort the listing by version
-x	List entries by lines instead of by columns
-X	Sort the listing by extension

Table 42: ls options

### Examples:

View detailed content of directory /mnt.

```
ls -l /mnt
```

View list contents of actually directory.

```
ls
```

## 2.47. mac

This program can be used to display the MAC address of eth0.

### Synopsis:

```
mac [<separator>]
```

### Examples:

Display the MAC address of eth0. Will be used as the separator character "-" instead of ":".

```
mac -
```

## 2.48. mkdir

This program used to make directories.

### Synopsis:

```
mkdir [<option>] directory ...
```

### Options:

Option	Description
-m	Set permission mode (as in chmod), not rwxrwxrwx – umask
-p	No error if existing, make parent directories as needed

Table 43: mkdir options

### Examples:

Create directory /tmp/test/example.

```
mkdir -p /tmp/test/example
```

## 2.49. mount

This program can be used to mount a file system.

### Synopsis:

```
mount [-a] [-o] [-r] [-t] [-w] <DEVICE> <NODE> [ -o <option>, ...]
```

### Options:

Flag	Description
-a	Mount all filesystems in fstab
-o	One of many filesystem options, listed below
-r	Mount the filesystem read-only
-t	Specify the filesystem type
-w	Mount for reading and writing (default)

Table 44: mount flags

Option	Description
async/sync	Writes are asynchronous / synchronous
atime/noatime	Enable / disable updates to inode access times
dev/nodev	Allow use of special device files / disallow them
exec/noexec	Allow use of executable files / disallow them
suid/nosuid	Allow set-user-id-root programs / disallow them
remount	Re-mount a mounted filesystem, changing its flags
ro/rw	Mount for read-only / read-write

bind	Bind a directory to an additional location
move	Relocate an existing mount point

Table 45: mount options

For detail description this command, visit Linux manual pages.

### Examples:

Connect a contents of USB flash drive to the directory /mnt.

```
mount -t vfat /dev/sda1 /mnt
```

## 2.50. mv

This program can be used to move or rename files.

### Synopsis:

```
mv [-f] [-i] <source> ...<dest>
```

### Options:

Option	Description
-f	Don't prompt before overwriting
-i	Interactive, prompt before overwrite

Table 46: mv options

### Examples:

Rename file abc.txt na def.txt.

```
mv abc.txt def.txt
```

Move all files with the extension txt to the directory /mnt.

```
mv *.txt /mnt
```

## 2.51. nc

This program (Netcat) can be used to open a pipe to IP:port.

### Synopsis:

```
nc [<options>] [<ip>] [<port>]
```

### Options:

Option	Description
-l	listen mode, for inbound connects
-p <port>	local port number
-i <secs>	delay interval for lines sent
-w <secs>	timeout for connects and final net reads

Table 47: nc options

**Example:**

Open a TCP connection to port 42 of 192.168.3.1, using port 31337 as the source port, with a timeout of 5 seconds:

```
nc -p 31337 -w 5 192.168.3.1 42
```

## 2.52. netstat

This program can be used to display the networking information.

**Synopsis:**

```
netstat [<options>]
```

**Options:**

Option	Description
-l	display listening server sockets
-a	display all sockets (default: connected)
-e	display other/more information
-n	don't resolve names
-r	display routing table
-t	tcp sockets
-u	udp sockets
-w	raw sockets
-x	unix sockets

Table 48: netstat options

## 2.53. nohup

Nohup is short for “No Hang-up”. Nohup is a supplemental command that tells the system not to stop another command once it has started. That means it'll keep running until it's done, even if the user that started it logs out.

**Synopsis:**

```
nohup <program> [<arguments>]
```

**Options:**

Option	Description
program	Program to be run immune to hang-ups with output to a non-tty.
arguments	Arguments for the program.

Table 49: nohup options



## 2.54. ntpdate

This program can be used to set the system time from NTP server.

### Synopsis:

```
ntpdate [-p <probes>] [-t <timeout>] <server>
```

### Options:

Option	Description
-p	Specify the number of samples to be acquired from each server as the integer samples, with values from 1 to 8 inclusive.
-t	Specify the maximum time waiting for a server response as the value timeout, in seconds and fraction.

Table 50: ntpdate options

### Examples:

Set the system time according to the NTP server time.windows.com.

```
ntpdate time.windows.com
```

## 2.55. openssl

The openssl program is a command line tool for using the various cryptography functions of OpenSSL's crypto library from the shell. It can be used for:

- *Creation of RSA, DH and DSA key parameters*
- *Creation of X.509 certificates, CSRs and CRLs*
- *Calculation of Message Digests*
- *Encryption and Decryption with Ciphers*
- *SSL/TLS Client and Server Tests*
- *Handling of S/MIME signed or encrypted mail*

### Synopsis:

```
openssl [<option> ...]
```

### Options:

For detail description this command, visit Linux manual pages.

### Examples:

Generate a new key for the SSH server.

```
openssl genrsa -out /etc/certs/ssh_rsa_key512
```

Generate a new certificate for the HTTPS server.

```
openssl req -new -out /tmp/csr -newkey rsa:1024 -nodes -keyout
/etc/certs/https_key
openssl x509 -req -setstart 700101000000Z -setend 400101000000Z
-in /tmp/csr -signkey /etc/certs/https_key -out
/etc/certs/https_cert
```

## 2.56. passwd

This program can be used to change password for user root.

### Synopsis:

```
passwd
```

## 2.57. pidof

This program lists the PIDs of all processes with names that match the names on the command line.

### Synopsis:

```
pidof <process-name> [<option>] [<process-name>...]
```

### Options:

Option	Description
-s	Display only a single PID.

Table 51: pidof options

## 2.58. ping

This program can be used to send ICMP echo request to network host.

### Synopsis:

```
ping [-c <count>] [-s <size>] [-I <interface>] [-q] <host>
```

### Options:

Option	Description
-c	Send only COUNT pings.
-s	Send SIZE data bytes in packets (default = 56).
-I	Source interface or IP address.
-q	Quiet mode, only displays output at start and when finished.

Table 52: ping options

### Examples:

Send one ICMP packet Echo Request with size 500 B on IP address 10.0.0.1.

```
ping -c 1 -s 500 10.0.0.1
```

## 2.59. portd

The program can be used for transparent transfer of data from the serial line by TCP or UDP.

### Synopsis:

```
portd -c <device> [-b <baudrate>] [-d <databits>] [-p <parity>]
[-s <stopbits>] [-l <split timeout>] [-4] [-h <hostname>] [-o
<proto>] -t <port> [-k <keepalive time>] [-i <keepalive
interval>] [-r <keepalive probes>] [-x] [-z]
```

### Options:

Option	Description
-c	Serial line device
-b	Baudrate
-d	Number of data bits
-p	Parity – even, odd or none
-s	Number of stop bits
-l	Split timeout
-4	Forced detection Expansion port 485
-h	Hostname
-o	Protocol TCP or UDP
-t	TCP or UDP port
-k	Keepalive time
-i	Keepalive interval
-r	Keepalive probes
-x	Use signal CD as indicator of the TCP connection
-z	Use DTR as control TCP connection

Table 53: portd options

### Examples:

Running a TCP server listening on port 1000th After a TCP connection, the program transparently transmit data from the serial port settings 115200 bit/s, 8N1.

```
portd -c /dev/ttyS0 -b 115200 -t 1000 &
```

## 2.60. ps

This program can be used to view report process status.

Synopsis:

```
ps
```

## 2.61. pse

This program can be used to manage PoE PSE board.

Synopsis:

```
pse [eth0 | eth1] [on | off]
```

**Example:**

Enable PoE PSE on *eth1* interface.

```
pse eth1 on
```

## 2.62. pwd

This program can be used to view current directory.

Synopsis:

```
pwd
```

## 2.63. reboot

This program can be used to reboot the router.

Synopsis:

```
reboot [-d <delay>] [-n <nosync>] [-f <force>]
```

Options:

Option	Description
-d	Delay interval for rebooting
-n	No call to sync()
-f	Force reboot, do not call shutdown

Table 54: reboot options

**Examples:**

Reboot router after 10 second.

```
reboot -d 10
```

## 2.64. report

This command can be used for creating of report from the command line.

Sensitive data from the report are filtered out for security reasons.

### Synopsis:

```
report [<options>]
```

### Options:

Option	Description
<i>no option</i>	Report all sections except the configuration one.
-a	Report all sections.
-s	Report status section.
-m	Report user modules section.
-l	Report log section.
-c	Report configuration section.

Table 55: report options

## 2.65. restore

This program can be used to restore configuration of the router from a file containing router's configuration. The backup of router's configuration can be generated into a file by *backup* command, see chapter 1.3.

### Synopsis:

```
restore <filename>
```

### Examples:

Restore configuration from file */tmp/my.cfg*.

```
restore /tmp/my.cfg
```

## 2.66. rm

This program can be used to remove files or directories.

### Synopsis:

```
rm [-i] [-f] [-r] <file> ...
```

Options:

Option	Description
-i	Always prompt before removing each destination
-f	Remove existing destinations, never prompt
-r	Remove the contents of directories recursively

Table 56: rm options

### Examples:

Remove all files with extension txt in the current directory.

```
rm *.txt
```

Remove directory /tmp/test and all subdirectories.

```
rm -rf /tmp/test
```

## 2.67. rmdir

This program can be used to remove empty directories.

Synopsis:

```
rmdir <filename>
```

### Examples:

Remove empty directory /tmp/test.

```
rmdir /tmp/test
```

## 2.68. route

This program can be used to show and manipulate the IP routing table.

Synopsis:

```
route [ -n ] [ -e ] [ -A ] [ add | del | delete ]
```

Options:

Option	Description
-n	Don't resolve names
-e	Display other/more information
-A	Select address family

Table 57: route options

For detail description this command, visit Linux manual pages.

### Examples:

View the routing table without translating IP addresses to domain names.

```
route -n
```

Add routing networks 192.168.3.0/24 through eth0.

```
route add -net 192.168.3.0/24 dev eth0
```

Add routing IP addresses 192.168.3.1 through 192.168.1.2 gateway.

```
route add -host 192.168.3.1 gw 192.168.1.2
```

Add default gateway 192.168.1.2

```
route add default gw 192.168.1.2
```

## 2.69. scp

This program can be used for secure file transferring between hosts on a network. It uses *ssh* protocol for data transfer with the same authentication and security.

### Synopsis:

```
scp [-12346BCpqr] [-c cipher] [-F ssh_config] [-i
identity_file] [-l limit] [-o ssh_option] [-P
port] [-S program] [[user@]host1:]file1 ...
                [[user@]host2:]file2
```

### Options:

Option	Description
-1	Forces scp to use protocol 1.
-2	Forces scp to use protocol 2.
-4	Forces scp to use IPv4 addresses only.
-6	Forces scp to use IPv6 addresses only.
-B	Selects batch mode (prevents asking for passwords or passphrases).
-C	Compression enable. Passes the -C flag to ssh to enable compression.
-c cipher	Selects the cipher to use for encrypting the data transfer. This option is directly passed to ssh.
-F ssh_config	Specifies an alternative per-user configuration file for ssh. This option is directly passed to ssh.
-i identity_file	Selects the file from which the identity (private key) for public key authentication is read. This option is directly passed to ssh.
-l limit	Limits the used bandwidth, specified in Kbit/s.
-o ssh_option	Can be used to pass options to ssh in the format used in ssh_config.
-P port	Specifies the port to connect to on the remote host.
-p	Preserves modification times, access times, and modes from the original file.
-q	Quiet mode: disables the progress meter as well as warning and diagnostic messages from ssh.
-r	Recursively copy entire directories. Note that scp follows symbolic links encountered in the tree traversal.

-S program	Name of program to use for the encrypted connection. The program must understand ssh options.
-v	Verbose mode. Causes scp and ssh to print debugging messages about their progress.

Table 58: scp options

The scp utility exits 0 on success, and >0 if an error occurs.

### Examples:

Copy the file "/etc/version" from remote host "remotehost.edu" to the local host, into subdirectory "myFolder" in user's home directory.

```
scp root@remotehost.edu:/etc/version ~/myFolder
```

Copy the file "/etc/version" from the local host to remote host "remotehost.edu", into user's home directory.

```
scp /etc/version root@remotehost.edu:~/
```

Copy the directory "/home/user" from the local host to a remote host's "/tmp/bar" directory.

```
scp -r /home/user root@remotehost.edu:/tmp/bar
```

## 2.70. sed

This program can be used for filtering and transforming text.

### Synopsis:

```
sed [ -e ] [ -f ] [ -i ] [ -n ] [ -r ] pattern [ -files ]
```

### Options:

Option	Description
-e	Add the script to the commands to be executed
-f	Add script-file contents to the commands to be executed
-i	Edit files in place (makes backup if extension supplied)
-n	Suppress automatic printing of pattern space
-r	Use extended regular expression syntax

Table 59: sed options

If no -e or -f is given, the first non-option argument is taken as the sed script to interpret. All remaining arguments are names of input files; if no input files are specified, then the standard input is read. Source files will not be modified unless -i option is given.

### Examples:

Change parameter PPP\_APN in file /etc/settings.ppp to value "internet".

```
sed -e "s/\(PPP_APN=\).*\/\1internet/" -i /etc/settings.ppp
```



## 2.71. service

This program can be used to start, stop or restart specified service.

**Synopsis:**

```
service <service_name> <start | stop | restart>
```

**Examples:**

Start service cron.

```
service cron start
```

Restart service ppp.

```
service ppp restart
```

## 2.72. sleep

This program can be used to delay for a specified amount of time.

**Synopsis:**

```
sleep <time>
```

**Examples:**

Sleep for 30 second.

```
sleep 30
```

## 2.73. slog

This script can be used to show system log (file /var/log/message).

**Synopsis:**

```
slog [-n <number>] [-f]
```

**Options:**

Option	Description
-n	Print last N lines instead of last 10
-f	Output data as the file grows

Table 60: slog options

**Examples:**

Continuous listing the system log. Listing stops when reaching the maximum number of lines of log.

```
slog -f
```

## 2.74. snmpget

*snmpget* is an SNMP application that uses the SNMP GET request to query for information on a network entity. One or more object identifiers (OIDs) may be given as arguments on the command line.

### Synopsis:

```
snmpget [OPTIONS] [-Cf] OID [OID]...
```

### Options:

Option	Description
-h, -help	Display the help.
-H	Display configuration file directives understood.
-v 1 2c 3	Specifies SNMP version to use.
-V, -version	Display package version number.
-Cf	If -Cf is not specified, some applications ( <i>snmpdelta</i> , <i>snmpget</i> , <i>snmpgetnext</i> and <i>snmpstatus</i> ) will try to fix errors returned by the agent that you were talking to and resend the request. The only time this is really useful is if you specified a OID that didn't exist in your request and you're using SNMPv1 which requires "all or nothing" kinds of requests.
<i>other</i>	<i>For other options see the command help.</i>

Table 61: snmpget options

### Examples:

Retrieve the variable `system.sysDescr.0` from the host `zeus` using the community string `public`.

```
snmpget -c public zeus system.sysDescr.0
```

## 2.75. snmpset

*snmpset* is an SNMP application that uses the SNMP SET request to set information on a network entity. One or more object identifiers (OIDs) must be given as arguments on the command line. A type and a value to be set must accompany each object identifier.

### Synopsis:

```
snmpset [OPTIONS] OID TYPE VALUE [OID TYPE VALUE]...
```

### Options:

Option	Description
-h, -help	Display the help.
-H	Display configuration file directives understood.
-v 1 2c 3	Specifies SNMP version to use.
-V, -version	Display package version number.

*other* For other options see the command help.

Table 62:

snmpset options The TYPE is a single

Character	Description
i	integer
u	unsigned
s	string
x	hex string
d	decimal string
n	nullobj
o	objid
t	timeticks
a	ipaddress
b	bits

## 2.76. snmptrap

This program can be used to send a SNMP trap.

### Synopsis:

```
snmptrap [-c <community>] [-g <generic>] [-s
<specific>] <hostname> [<oid> <type> <value>]
```

### Options:

Option	Description
-c	Community
-g	Specifies generic trap types: <ul style="list-style-type: none"> <li>• 0 – coldStart</li> <li>• 1 – warmStart</li> <li>• 2 – linkDown</li> <li>• 3 – linkUp</li> <li>• 4 – authenticationFailure</li> <li>• 5 – egpNeighborLoss</li> <li>• 6 – enterpriseSpecific</li> </ul>
-r	Sends MAC address of eth0 interface
-s	Specifies user definition trap types in the enterpriseSpecific

Table 64: snmptrap options

### Examples:

Send TRAP with info about the status of a digital input BIN0 to the IP address 192.168.1.2.

```
snmptrap 192.168.1.2 1.3.6.1.4.1.30140.2.3.1.0 u 'io get bin0'
```

Send TRAP "warm start" to the IP address 192.168.1.2.

```
snmptrap -g 1 192.168.1.2
```

## 2.77. status

This program writes out the status of the router's interfaces or system. It is equivalent to information available at *General Status* and *Mobile WAN Status* in the router's web administration.

### Synopsis:

```
status [-hv] [lan | mobile | module | mwan | ports | sys | wifi ap | wifi sta]
```

### Options:

Option	Description
-h	Generates html output (used when called by web interface).
-v	The verbose mode that writes out more detailed information. Units of data amount are just <b>in bytes</b> , not in dynamic units (KB, MB, etc.).
lan	Status of primary LAN. Can be lan 1, lan 2, etc. if available.
mobile	Status of mobile WAN.
module	Status of mobile module. Can be module 1, module 2, etc. if available.
mwan	Status of mobile connection.
ports	Status of available peripheral ports.
sys	System information.
wifi ap	Status of WiFi access point.
wifi sta	Status of WiFi station.

Table 65: status options

### Examples:

Show verbose status of mobile connection:

```
# status -v mobile Registration      : Home Network Operator  : VodafoneCZ

Technology   : LTE
PLMN        : 23003
Cell        : 10A802
LAC         : 947C
Channel     : 275
SignalStrength : -90 dBm
SignalQuality  : -12 dB
CSQ         : 11
```

## 2.78. split

This program splits a single file (INPUT) into multiple files. A custom PREFIX for the name of the output files can be specified.

### Synopsis:

```
split [OPTIONS] [INPUT [PREFIX]]
```

### Options:

Option	Description
-b N[k m]	Split input file by N (kilo mega)bytes.
-l N	Split input file by N lines (by default 1000 lines).
-a N	Use N letters as suffix for the name of the output files (by default 2 letters).

### Examples:

Table 66: split options

Split file *file.img* into files (*xaa*, *xab*, *xac*, ...) every with size of 50 kB:

```
split -b 50k file.img
```

Split file *file.txt* into files (*file\_a*, *file\_b*, *file\_c*, ...) every containing of 200 lines:

```
split -l 200 -a 1 file.txt file_
```

## 2.79. stty

This program can be used to print or to change terminal characteristics.

### Synopsis:

```
stty [-a|g] [-F DEVICE] [SETTING]...
```

### Options:

Option	Description
-F DEVICE	Open device instead of stdin
-a	Print all current settings in human-readable form
-g	Print in stty-readable form
[SETTING]	See manpage

### Examples:

Table 67: stty options

To get current parameters of the first UART serial port.

```
stty -F /dev/ttyS0
```

To only get actual speed of the second UART serial port.

```
stty -F /dev/ttyS1 speed
```

To set parameters of the first UART serial port to:

- *speed to 1200 bps*
- *character size to 7 bits*
- *2 stop bits*
- *disable software output flow control*
- *reset parameters to system default raw mode*

```
stty -F /dev/ttyS0 1200 cs7 cstopb -ixon raw
```

## 2.80. sync

This command forces an immediate transfer of buffered data blocks in memory or in FILES to the disk.

Synopsis:

```
sync [OPTIONS] [FILEs]...
```

Options:

Option	Description
-d	Avoid syncing metadata.
-f	Sync filesystems underlying FILEs.

Table 68: sync options

## 2.81. tail

This program can be used to output the last part of files.

Synopsis:

```
tail [ -n <number> ] [ -f ]
```

Options:

Option	Description
-n	Print last N lines instead of last 10
-f	Output data as the file grows

Examples:

Table 69: tail options

Show last 30 lines of /var/log/messages.

```
tail -n 30 /var/log/messages
```

## 2.82. tar

This program can be used to create, extract or list files from a tar file.

### Synopsis:

```
tar -[czxtv0] [ -f tarfile ] [ -C dir ] [ file ] ...
```

### Options:

Option	Description
c	Create
x	Extract
t	List
z	Filter the archive through gzip
-f	Name of TARFILE or "-" for stdin
0	Extract to stdout
-C	Change to directory DIR before operation
v	Verbosely list files processed

### Examples:

#### Table 70: tar options

Creating log.tar archive that contains files from the directory /var/log.

```
tar -cf log.tar /var/log
```

Extract files from the archive log.tar.

```
tar -xf log.tar
```

## 2.83. tcpdump

This program can be used to dump traffic on a network.

### Synopsis:

```
tcpdump [-AdDeflLnNOpqRStuUvxxX] [-c <count>] [-C <file size>]  
[-E algo:secret] [-F <file>] [-i <interface>] [-r <file>] [-s  
<snaplen>] [-T type] [-w <file>] [-y <datalinktype>]  
[expression]
```

### Options:

For detail description this command, visit Linux manual pages.

### Examples:

View traffic on interface ppp0.

```
tcpdump -n -i ppp0
```

View traffic on interface eth0 except protocol Telnet.

```
tcpdump -n not tcp port 23
```

View UDP traffic on interface eth0.

```
tcpdump -n udp
```

View HTTP traffic on interface eth0.

```
tcpdump -n tcp port 80
```

View all traffic from/to IP address 192.168.1.2.

```
tcpdump -n host 192.168.1.2
```

View traffic from/to IP address 192.168.1.2 except protocol Telnet.

```
tcpdump -n host 192.168.1.2 and not tcp port 23
```

## 2.84. telnet

This program can be used to establish interactive communication with another computer over a network using the TELNET protocol.

### Synopsis:

```
telnet <host> [<port>]
```

### Examples:

Connect to 192.168.1.2 by protocol Telnet.

```
telnet 192.168.1.2
```

## 2.85. top

This program provides a dynamic real-time view of a running system. It can display system summary information, as well as a list of processes or threads currently being managed by the kernel.

### Synopsis:

```
top [-b] [-nCOUNT] [-dSECONDS]
```

### Options:

Option	Description
-b	Batch mode - could be useful for sending output from <i>top</i> to other programs or to a file. In this mode, <i>top</i> will not accept input and runs until the iterations limit you've set with the '-n' command-line option, or until killed.
-nCOUNT	Exit after N iterations.
-dSECONDS	Delay between updates in secs format.

Table 71: touch options



Keys:

Key	Description
n/m/p/t	Sort by pid/mem/cpu/time
r	Reverse sort
q, ^C	Exit

Table 72: touch keys

### Examples:

Run *top* program in batch mode and exit after 5 iterations.

```
top -b -n5
```

Run *top* program and update the output every 10 seconds. Exit by *q* key.

```
top -d10
```

## 2.86. touch

This program can be used to update timestamp of file.

Synopsis:

```
touch [-c] <file> [<file> ...]
```

Options:

Option	Description
-c	Do not create any files

Table 73: touch options

### Examples:

Create a file, respectively update timestamp of file */tmp/test*.

```
touch /tmp/test
```

## 2.87. traceroute

This program can be used to track the route to a network host.

Synopsis:

```
traceroute [-FIldnrv] [-f<1st_ttl>] [-m<max_ttl>] [-p<port#>] [-q
<nqueries>]
[-s<src_addr>] [-t<tos>] [-w<wait>] [-g<gateway>] [-i<iface>] [-
z<pausesecs>] host [data size]
```

Options:

Table 74: traceroute options

**Examples:**

Start traceroute on IP address 10.0.0.1 (without translation IP addresses to domain names).

```
traceroute -n 10.0.0.1
```

## 2.88. umount

This program can be used to unmount file systems.

**Synopsis:**

```
umount [-a] [-r] [-l] [-f] <file system> | <directory>
```

**Options:**

Option	Description
-a	Unmount all file systems
-r	Try to remount devices as read-only if mount is busy
-l	Lazy unmount (detach filesystem)
-f	Force unmount (i.e. unreachable NFS server)

Table 75: umount options

**Examples:**

Disconnecting the disc connected to the directory /mnt.

```
umount /mnt
```

## 2.89. umupdate

This program can be used for adding or deleting of a user module from the command line.

**Synopsis:**

```
umupdate [-a <filename>] [-d <name>]
```

**Options:**

Option	Description
-a	Add new or update installed user module. Enter path to the installation file.
-d	Delete an installed user module with the specified name. List of installed user modules can be obtained by <code>service module list</code> command.

Table 76: umupdate options

## 2.90. vi

This program can be used to edit and read text file.

**Synopsis:**

```
vi [-R] [<file> ...]
```

**Options:**

Option	Description
-R	Read only, do not write to the file

Table 77: vi options

**Examples:**

Open file `/etc/rc.local` in the text editor `vi`.

```
vi /etc/rc.local
```

## 2.91. wc

Print newline, word, and byte counts for each file, and a total line if more than one file is specified. With no file, or when file is a dash ("-"), read standard input.

**Synopsis:**

```
wc [<option(s)>] [<file(s)>]
```

**Options:**

Option	Description
-c	Print the byte counts.
-l	Print the newline counts.
-L	Print the length of the longest line.
-w	Print the word counts.

Table 78: wc options

## 2.92. wget

This program can be used to retrieve files via HTTP or FTP.

**Synopsis:**

```
wget [-c] [-q] [-O <document file>] [--header 'header:
valu
e'] [-Y on/off] [-P <DIR>] <url>
```

**Options:**

Option	Description
-c	Continue retrieval of aborted transfers
-q	Quiet mode – do not print

-P	Set directory prefix to DIR
-O	Save to filename ('-' for stdout)
-Y	Use proxy ('on' or 'off')

Table 79: wget options

### Examples:

Download a file my.cfg from HTTP server with IP address 10.0.0.1.

```
wget http://10.0.0.1/my.cfg
```

## 2.93. xargs

This program executes the command on every item given by standard input.

### Synopsis:

```
xargs [<commands>] [<options>] [<args> ...]
```

### Options:

Option	Description
-r	Do not run command for empty readed lines
-t	Print the command line on stderr before executing it

Table 80: xargs options

### Examples:

Find files named core in or below the directory /tmp and delete them. Note that this will work incorrectly if there are any filenames containing newlines or spaces.

```
find /tmp -name core -type f -print | xargs /bin/rm -f
```

## 3. Examples of Scripts

### 3.1. HTTP POST Method for a Remote Execution

CSRF (Cross-site Request Forgery) protection is implemented in all routers from firmware version 5.3.0. To execute an action remotely, a script with *curl* command can be used. In the next chapter is example of script for sending of SMS remotely via router's web page by HTTP POST method.

### 3.2. Send SMS Remotely via Router's Web Page by HTTP POST Method

This script can be used for sending SMS remotely via router's web page by HTTP POST method. It can be executed from the router itself or from any other Linux-based machine.

**Script:**

```
#!/bin/sh ROUTER=192.168.1.1

phone=123456789

message="Hello world."

curl -k -d "username=root&password=root" --dump-header /tmp/headers

--output /dev/null https://$ROUTER/login_exec.cgi REQUEST_ID=$(curl -k -b
/tmp/headers https://$ROUTER/send.cgi |

sed -n '/hidden/s/.*name="request_id"\s\+value="\([^"]*\)\s\+/\1/p' )

curl -k -b /tmp/headers --form request_id=$REQUEST_ID --form phone="$phone" -
-form message="$message" --output /dev/null https://$ROUTER/send_exec.cgi
```

### 3.3. SMS Handling Using /var/scripts/sms

Next three Chapters show examples of using "/var/scripts/sms" script. This script located in RAM of the router has to be created in Startup Script (using EOF, see examples below), so the file is on its place in the router even after reboot.

The script "/var/scripts/sms" is called by Mobile WAN connection daemon if there is active *Enable remote control via SMS* in *Configuration Services SMS* section. The script can be used for creation of advanced (your own) control SMS commands of the router. The Mobile WAN daemon passes on following parameters to the script:

- \$0 – name of script itself (in this case "sms") – not passed
- \$1 – can be "1" or "0" (true or false). The value true ("1") is returned if mobile phone number the SMS is received from is filled in the field Phone Number X on the SMS Web configuration page. Otherwise it is false ("0").
- \$2 – mobile phone number of the SMS sender
- \$3 to \$9 – words of SMS, separated by space (maximum of seven words)

### 3.4. Send SMS to E-mail

Send incoming SMS to the email.

**Startup Script:**

```
EMAIL=john.doe@email.com

cat > /var/scripts/sms << EOF #!/bin/sh

/usr/bin/email -t $EMAIL -s "Received SMS from \"$2" -m "Authorized: \"$1, Text: \"$3
 \"$4 \"$5 \"$6 \"$7 \"$8 \"$9"
```

EOF

## 3.5. SMS Command 1

Implementation of a new SMS command "IMPULSE", which activates binary output OUT0 for 5 seconds. SMS will be processed, if it comes from one of three numbers defined on the web interface or phone number +420123456789.

Startup Script:

```
PHONE=+420123456789

cat > /var/scripts/sms << EOF #!/bin/sh

if [ "\$1" = "1" ] || [ "\$2" = "$PHONE" ]; then if [ "\$3" = "IMPULSE" ]; then

/usr/bin/io set out0 1 sleep 5

/usr/bin/io set out0 0 fi

fi EOF
```

## 3.6. SMS Command 2

*This script implements a new SMS command "PPP", which sets item **Network type**, **Default SIM card** and **Backup SIM card**. PPP command has the following structure:*

PPP <AUTO/GPRS/UMTS> <1/2>

The first parameter sets network type. If the second parameter equals 1, **Default SIM card** will be set to primary SIM card. If this parameter equals 2, **Default SIM card** will be set to secondary SIM card.

Startup Script:

```
cat > /var/scripts/sms << EOF STARTUP=#!/bin/sh

if [ "\$1" = "1" ]; then

if [ "\$3" = "PPP" ]; then

if [ "\$4" = "AUTO" ]; then

sed -e "s/\(PPP_NETTYPE=\).*\/\10/" -e "s/\(PPP_NETTYPE2=\).*\/\10/" -i

/etc/settings.ppp

elif [ "\$4" = "GPRS" ]; then
```

```
sed -e "s/\(PPP_NETTYPE=\).*\/\11/" -e "s/\(PPP_NETTYPE2=\).*\/\11/" -i
/etc/settings.ppp
elif [ "$4" = "UMTS" ];then
sed -e "s/\(PPP_NETTYPE=\).*\/\12/" -e "s/\(PPP_NETTYPE2=\).*\/\12/" -i
/etc/settings.ppp fi
if [ "$5" = "1" ];then
sed -e "s/\(PPP_DEFAULT_SIM=\).*\/\11/" -e "s/\(PPP_BACKUP_SIM=\).*\/\12/"
-i /etc/settings.ppp elif [ "$5" = "2" ]; then
sed -e "s/\(PPP_DEFAULT_SIM=\).*\/\12/" -e "s/\(PPP_BACKUP_SIM=\).*\/\11/"
-i /etc/settings.ppp fi
reboot fi
fi EOF
```

## 3.7. Send Information Email 1

Send information email about establishing of PPP connection.

Up Script:

```
EMAIL=john.doe@email.com
/usr/bin/email -t $EMAIL -s "Router has established PPP connection.
IP address: $4"
```

## 3.8. Send Information SNMP Trap 1

Send information SNMP trap about establishing of PPP connection.

Up Script:

```
SNMP_MANAGER=192.168.1.2
/usr/bin/snmptrap -g 3 $SNMP_MANAGER
```

## 3.9. Send Information Email 2

Send information email about switch binary input BIN0.

Startup Script:

```
EMAIL=john.doe@email.com MESSAGE="BIN0 is active"
```

```
while true do
/usr/bin/io get bin0 VAL=$?
if [ "$VAL" != "$OLD" ];then
[ "$VAL" = "0" ] && /usr/bin/email -t $EMAIL -s "$MESSAGE"
OLD=$VAL
fi sleep 1
done
```

## 3.10. Send Information SNMP Trap 2

Send information SNMP trap about change state of binary input BIN0.

Startup Script:

```
SNMP_MANAGER=192.168.1.2
```

```
while true do
```

```
/usr/bin/io get bin0 VAL=$?
```

```
if [ "$VAL" != "$OLD" ];then
```

```
/usr/bin/snmptrap $SNMP_MANAGER 1.3.6.1.4.1.30140.2.3.1.0 u $VAL OLD=$VAL
```

```
fi sleep 1
```

```
done
```

## 3.11. Automatic Reboot

Automatic reboot at the definition time. (23:55)

Startup Script:

```
echo "55 23 * * * root /sbin/reboot"
> /etc/crontab service cron start
```

## 3.12. Switch Between WAN and PPP

Switching between WAN and PPP. PPP connection is active, if PING on the defined IP address does not pass through.



**Startup Script:**

```
WAN_PING=192.168.2.1 WAN_GATEWAY=192.168.2.1 WAN_DNS=192.168.2.1

. /etc/settings.eth

/sbin/route add $WAN_PING gw $WAN_GATEWAY

/sbin/iptables -t nat -A PREROUTING -i eth1 -j napt

/sbin/iptables -t nat -A POSTROUTING -o eth1 -p ! esp -j MASQUERADE

LAST=1

while true do

ping -c 1 $WAN_PING PING=$?

if [ $PING != $LAST ]; then LAST=$PING

if [ $PING = 0 ]; then

/etc/init.d/ppp stop sleep 3

/sbin/route add default gw $WAN_GATEWAY

echo "nameserver $WAN_DNS" >/etc/resolv.conf
```

```
/usr/sbin/contrack -F
```

```
/etc/scripts/ip-up - - - $ETH2_IPADDR else

/etc/scripts/ip-down - - - $ETH2_IPADDR

/usr/sbin/contrack -F

/sbin/route del default gw $WAN_GATEWAY

/etc/init.d/ppp start fi

fi

sleep 1

done
```

### 3.13. How to Use Unsupported FTDI Chip

Unsupported FTDI chip can be added on the fly using:  
echo <VID> <PID> >/sys/bus/usb-serial/drivers/ftdi\_sio/new\_id



Example Startup Script for VID 0403 and PID d921:

```
echo 0403 d921>/sys/bus/usb-serial/drivers/ftdi_sio/new_id
```