NAME

BusyBox - The Swiss Army Knife of Embedded Linux

SYNTAX

```
 BusyBox <function> [arguments...]  # or
 <function> [arguments...]          # if symlinked
```

DESCRIPTION

BusyBox combines tiny versions of many common UNIX utilities into a single small executable. It provides minimalist replacements for most of the utilities you usually find in GNU coreutils, util-linux, etc. The utilities in BusyBox generally have fewer options than their full-featured GNU cousins; however, the options that are included provide the expected functionality and behave very much like their GNU counterparts.

BusyBox has been written with size-optimization and limited resources in mind. It is also extremely modular so you can easily include or exclude commands (or features) at compile time. This makes it easy to customize your embedded systems. To create a working system, just add /dev, /etc, and a Linux kernel. BusyBox provides a fairly complete POSIX environment for any small or embedded system.

BusyBox is extremely configurable. This allows you to include only the components you need, thereby reducing binary size. Run 'make config' or 'make menuconfig' to select the functionality that you wish to enable. The run 'make' to compile BusyBox using your configuration.

After the compile has finished, you should use 'make install' to install BusyBox. This will install the '/bin/busybox' binary, and will also create symlinks pointing to the '/bin/busybox' binary for each utility that you compile into BusyBox. By default, 'make install' will place these symlinks into the './_install' directory, unless you have defined 'PREFIX', thereby specifying some alternative location (i.e., 'make PREFIX=/tmp/foo install'). If you wish to install using hardlinks, rather than the default of using symlinks, you can use 'make PREFIX=/tmp/foo install-hardlinks' instead.

USAGE

BusyBox is a multi-call binary. A multi-call binary is an executable program that performs the same job as more than one utility program. That means there is just a single BusyBox binary, but that single binary acts like a large number of utilities. This allows BusyBox to be smaller since all the built-in utility programs (we call them applets) can share code for many common operations.

You can also invoke BusyBox by issuing a command as an argument on the command line. For example, entering

```
        /bin/busybox ls
```

will also cause BusyBox to behave as 'ls'.

Of course, adding '/bin/busybox' into every command would be painful. So most people will invoke BusyBox using links to the BusyBox binary.

For example, entering

```
        ln -s /bin/busybox ls
        ./ls
```

will cause BusyBox to behave as 'ls' (if the 'ls' command has been compiled into BusyBox). Generally speaking, you should never need to make all these links yourself, as the BusyBox build system will do this for you when you run the 'make install' command.

If you invoke BusyBox with no arguments, it will provide you with a list of the applets that have been compiled into your BusyBox binary.

COMMON OPTIONS

Most BusyBox commands support the **--help** argument to provide a terse runtime description of their behavior. If the CONFIG_FEATURE_VERBOSE_USAGE option has been enabled, more detailed usage information will also be available.

COMMANDS

Currently defined functions include:

```
        addgroup, adduser, adjtimex, ar, arping, ash, awk, basename,
bunzip2,
        busybox, bzcat, cal, cat, chgrp, chmod, chown, chroot, chvt,
clear, cmp,
        cp, cpio, crond, crontab, cut, date, dc, dd, deallocvt,
delgroup, deluser,
        devfsd, df, dirname, dmesg, dos2unix, dpkg, dpkg-deb, du,
dumpkmap,
        dumpleases, echo, egrep, env, expr, false, fbset, fdflush,
fdformat, fdisk,
        fgrep, find, fold, free, freeramdisk, fsck.minix, ftpget,
ftpput, getopt,
        getty, grep, gunzip, gzip, halt, hdparm, head, hexdump, hostid,
hostname,
        httpd, hush, hwclock, id, ifconfig, ifdown, ifup, inetd, init,
insmod,
        install, ip, ipcalc, iplink, iproute, iptunnel, kill, killall,
klogd, lash,
        last, length, linuxrc, ln, loadfont, loadkmap, logger, login,
logname,
        logread, losetup, ls, lsmod, makedevs, md5sum, mesg, mkdir,
mkfifo,
        mkfs.minix, mknod, mkswap, mktemp, modprobe, more, mount, msh,
mt, mv,
        nameif, nc, netstat, nslookup, od, openvt, passwd, patch, pidof,
ping,
        ping6, pipe_progress, pivot_root, poweroff, printf, ps, pwd,
rdate,
        readlink, realpath, reboot, renice, reset, rm, rmdir, rmmod,
route, rpm,
        rpm2cpio, run-parts, rx, sed, seq, setkeycodes, sha1sum, sleep,
sort,
        start-stop-daemon, strings, stty, su, sulogin, swapoff, swapon,
sync,
        sysctl, syslogd, tail, tar, tee, telnet, telnetd, test, tftp,
time, top,
        touch, tr, traceroute, true, tty, udhcpc, udhcpd, umount, uname,
        uncompress, uniq, unix2dos, unzip, uptime, usleep, uudecode,
uuencode,
        vconfig, vi, vlock, watch, watchdog, wc, wget, which, who,
whoami, xargs,
        yes, zcat
```

COMMAND DESCRIPTIONS

addgroup

addgroup [**-g** GID] group_name [user_name]

Adds a group to the systemOptions:

```
        -g GID          specify gid
```

----

adduser

adduser [OPTIONS] user_name

Adds a user to the systemOptions:

```
        -h DIR          Assign home directory DIR
        -g GECOS        Assign gecos field GECOS
        -s SHELL        Assign login shell SHELL
        -G              Add the user to existing group GROUP
        -S              create a system user (ignored)
        -D              Do not assign a password (logins still
possible via ssh)
        -H              Do not create the home directory
```

----

adjtimex

adjtimex [**-q**] [**-o** offset] [**-f** frequency] [**-p** timeconstant] [**-t** tick]

Reads and optionally sets system timebase parameters. See adjtimex(2).

Options:

```
        -q              quiet mode - do not print
        -o offset       time offset, microseconds
        -f frequency    frequency adjust, integer kernel units
(65536 is 1ppm)
                        (positive values make the system clock run
fast)
        -t tick         microseconds per tick, usually 10000
        -p timeconstant
```

----

ar

ar [**-o**] [**-v**] [**-p**] [**-t**] [**-x**] ARCHIVE FILES

Extract or list FILES from an ar archive.

Options:

```
        -o              preserve original dates
        -p              extract to stdout
        -t              list
        -x              extract
        -v              verbosely list files processed
```

----

arping

arping [**-fqbDUA**] [**-c** count] [**-w** timeout] [**-I** device] [**-s** sender] target

Ping hosts by ARP requests/replies.

Options:

```
        -f              Quit on first ARP reply
        -q              Be quiet
        -b              Keep broadcasting, don't go unicast
        -D              Duplicated address detection mode
        -U              Unsolicited ARP mode, update your
neighbours
        -A              ARP answer mode, update your neighbours
        -c count        Stop after sending count ARP request
packets
        -w timeout      Time to wait for ARP reply, in seconds
        -I device       Outgoing interface name, default is eth0
        -s sender       Set specific sender IP address
        target          Target IP address of ARP request
```

----

ash

ash [FILE]... or: ash **-c** command [args]...

The ash shell (command interpreter)

----

awk

awk [OPTION]... [program-text] [FILE ...]

Options:

```
        -v var=val              assign value 'val' to variable
'var'
        -F sep          use 'sep' as field separator
        -f progname             read program source from file
'progname'
```

----

basename

basename FILE [SUFFIX]

Strips directory path and suffixes from FILE. If specified, also removes any trailing SUFFIX.

Example:

```
$ basename /usr/local/bin/foo
foo
$ basename /usr/local/bin/
bin
$ basename /foo/bar.txt .txt
bar
```

----

bunzip2

bunzip2 [OPTION]... [FILE]

Uncompress FILE (or standard input if FILE is '-' or omitted).

Options:

```
-c      Write output to standard output
-f      Force
```

----

bzcat

bzcat FILE

Uncompress to stdout.

----

cal

cal [**-jy**] [[month] year]

Display a calendar.

Options:

```
-j      Use julian dates.
-y      Display the entire year.
```

----

cat

cat [**-u**] [FILE]...

Concatenates `FILE(s)` and prints them to stdout.

Options:

```
-u      ignored since unbuffered i/o is always used
```

Example:

```
        $ cat /proc/uptime
        110716.72 17.67
```

----

chgrp

chgrp [OPTION]... GROUP FILE...

Change the group membership of each FILE to GROUP.

Options:

```
        -R      Changes files and directories recursively.
```

Example:

```
        $ ls -l /tmp/foo
        -r--r--r--    1 andersen andersen        0 Apr 12 18:25 
/tmp/foo
        $ chgrp root /tmp/foo
        $ ls -l /tmp/foo
        -r--r--r--    1 andersen root            0 Apr 12 18:25 
/tmp/foo
```

----

chmod

chmod [**-R**] MODE[,MODE]... FILE...

Each MODE is one or more of the letters ugoa, one of the symbols +-= and one or more of the letters rwxst.

Options:

```
        -R      Changes files and directories recursively.
```

Example:

```
        $ ls -l /tmp/foo
        -rw-rw-r--    1 root     root            0 Apr 12 18:25 
/tmp/foo
        $ chmod u+x /tmp/foo
        $ ls -l /tmp/foo
        -rwxrw-r--    1 root     root            0 Apr 12 18:25 
/tmp/foo*
        $ chmod 444 /tmp/foo
        $ ls -l /tmp/foo
        -r--r--r--    1 root     root            0 Apr 12 18:25
/tmp/foo
```

----

chown

chown [**-Rh**]... OWNER[<.|:>[GROUP]] FILE...

Change the owner and/or group of each FILE to OWNER and/or GROUP.

Options:

```
        -R      Changes files and directories recursively.
        -h      Do not dereference symbolic links.
```

Example:

```
        $ ls -l /tmp/foo
        -r--r--r--    1 andersen andersen        0 Apr 12 18:25
/tmp/foo
        $ chown root /tmp/foo
        $ ls -l /tmp/foo
        -r--r--r--    1 root     andersen        0 Apr 12 18:25
/tmp/foo
        $ chown root.root /tmp/foo
        ls -l /tmp/foo
        -r--r--r--    1 root     root            0 Apr 12 18:25
/tmp/foo
```

----

chroot

chroot NEWROOT [COMMAND...]

Run COMMAND with root directory set to NEWROOT.

Example:

```
        $ ls -l /bin/ls
        lrwxrwxrwx    1 root     root           12 Apr 13 00:46
/bin/ls -> /BusyBox
        # mount /dev/hdc1 /mnt -t minix
        # chroot /mnt
        # ls -l /bin/ls
        -rwxr-xr-x    1 root     root        40816 Feb  5 07:45
/bin/ls*
```

----

chvt

chvt N

Changes the foreground virtual terminal to /dev/ttyN

----

clear

clear

Clear screen.

----

cmp

cmp [**-l**] [**-s**] FILE1 [FILE2]

Compare files. Compares FILE1 vs stdin if FILE2 is not specified.

Options:

```
        -l      Write the byte numbers (decimal) and values
(octal)
                  for all differing bytes.
        -s      quiet mode - do not print
```

----

cp

cp [OPTION]... SOURCE DEST

Copies SOURCE to DEST, or multiple `SOURCE(s)` to DIRECTORY.

```
        -a      Same as -dpR
        -d      Preserves links
        -p      Preserves file attributes if possible
        -f      force (implied; ignored) - always set
        -i      interactive, prompt before overwrite
        -R,-r   Copies directories recursively
```

----

cpio

cpio -[dimtuv][F cpiofile]

Extract or list files from a cpio archive Main operation mode:

```
        d               make leading directories
        i               extract
        m               preserve mtime
        t               list
        u               unconditional overwrite
        F               input from file
```

----

crond

crond **-d**[#] **-c** <crondir> **-f -b**

```
        -d [#] -l [#] -S -L logfile -f -b -c dir
        -d num  debug level
        -l num  log level (8 - default)
        -S      log to syslod (default)
        -L file log to file
        -f      run in fordeground
        -b      run in background (default)
        -c dir  working dir
```

----

crontab

crontab [**-c** dir] {file|-}|[**-u**|**-l**|**-e**|**-d** user]

```
        file <opts>  replace crontab from file
        -    <opts>  replace crontab from stdin
        -u user      specify user
        -l [user]    list crontab for user
        -e [user]    edit crontab for user
        -d [user]    delete crontab for user
        -c dir       specify crontab directory
```

----

cut

cut [OPTION]... [FILE]...

Prints selected fields from each input FILE to standard output.

Options:

```
        -b LIST         Output only bytes from LIST
        -c LIST         Output only characters from LIST
        -d CHAR         Use CHAR instead of tab as the field
delimiter
        -s              Output only the lines containing delimiter
        -f N            Print only these fields
        -n              Ignored
```

Example:

```
        $ echo "Hello world" | cut -f 1 -d ' '
        Hello
        $ echo "Hello world" | cut -f 2 -d ' '
        world
```

----

date

date [OPTION]... [MMDDhhmm[[CC]YY][.ss]] [+FORMAT]

Displays the current time in the given FORMAT, or sets the system date.

Options:

```
        -R              Outputs RFC-822 compliant date string
        -d STRING       Displays time described by STRING, not
`now'
        -I[TIMESPEC]    Outputs an ISO-8601 compliant date/time
string.
                        TIMESPEC=`date' (or missing) for date
only,
                        `hours', `minutes', or `seconds' for date
and,
                        time to the indicated precision.
        -s              Sets time described by STRING
        -u              Prints or sets Coordinated Universal Time
```

Example:

```
        $ date
        Wed Apr 12 18:52:41 MDT 2000
```

----

dc

dc expression ...

This is a Tiny RPN calculator that understands the following operations: +, add, -, sub, *, mul, /, div, %, mod, **, exp, and, or, not, eor. For example: 'dc 2 2 add' -> 4, and 'dc 8 8 * 2 2 + /' -> 16.

Options: p - Prints the value on the top of the stack, without altering the stack. f - Prints the entire contents of the stack without altering anything. o - Pops the value off the top of the stack and uses it to set the output radix.

```
    Only 10 and 16 are supported.
```

Example:

```
        $ dc 2 2 +
        4
        $ dc 8 8 * 2 2 + /
```

```
16
$ dc 0 1 and
0
$ dc 0 1 or
1
$ echo 72 9 div 8 mul | dc
64
```

----

dd

dd [if=FILE] [of=FILE] [bs=N] [count=N]
[skip=N]
[seek=N]
[conv=notrunc|noerror|sync]

Copy a file, converting and formatting according to options

```
if=FILE         read from FILE instead of stdin
of=FILE         write to FILE instead of stdout
bs=N            read and write N bytes at a time
count=N         copy only N input blocks
skip=N          skip N input blocks
seek=N          skip N output blocks
conv=notrunc    don't truncate output file
conv=noerror    continue after read errors
conv=sync       pad blocks with zeros
```

Numbers may be suffixed by c (x1), w (x2), b (x512), kD (x1000), k (x1024), MD (x1000000), M (x1048576), GD (x1000000000) or G (x1073741824).

Example:

```
$ dd if=/dev/zero of=/dev/ram1 bs=1M count=4
4+0 records in
4+0 records out
```

----

deallocvt

deallocvt [N]

Deallocate unused virtual terminal /dev/ttyN

----

delgroup

delgroup GROUP

Deletes group GROUP from the system

----

deluser

deluser USER

Deletes user USER from the system

----

devfsd

devfsd mntpnt [**-v**][**-fg**][**-np**]

Optional daemon for managing devfs permissions and old device name symlinks.

Options:

```
        mntpnt  The mount point where devfs is mounted.
        -v      Print the protocol version numbers for devfsd
                and the kernel-side protocol version and exits.
        -fg     Run the daemon in the foreground.
        -np     Exit  after  parsing  the configuration file
                and processing synthetic REGISTER events.
                Do not poll for events.
```

----

df

df [**-hmk**] [FILESYSTEM ...]

Print the filesystem space used and space available.

Options:

```
        -h      print sizes in human readable format (e.g., 1K
243M 2G )
        -m      print sizes in megabytes
        -k      print sizes in kilobytes(default)
```

Example:

```
        $ df
        Filesystem           1k-blocks      Used Available Use%
Mounted on
        /dev/sda3              8690864   8553540    137324  98% /
        /dev/sda1                64216     36364     27852  57%
/boot
        $ df /dev/sda3
        Filesystem           1k-blocks      Used Available Use%
Mounted on
        /dev/sda3              8690864   8553540    137324  98% /
```

----

dirname

dirname FILENAME

Strips non-directory suffix from FILENAME

Example:

```
        $ dirname /tmp/foo
        /tmp
        $ dirname /tmp/foo/
        /tmp
```

----

dmesg

dmesg [**-c**] [**-n** LEVEL] [**-s** SIZE]

Prints or controls the kernel ring buffer

Options:

```
        -c              Clears the ring buffer's contents after
printing
        -n LEVEL        Sets console logging level
        -s SIZE         Use a buffer of size SIZE
```

----

dos2unix

dos2unix [option] [FILE]

Converts FILE from dos format to unix format. When no option is given, the input is converted to the opposite output format. When no file is given, uses stdin for input and stdout for output.

Options:

```
        -u      output will be in UNIX format
        -d      output will be in DOS format
```

----

dpkg

dpkg [**-iCPru**] package_name

dpkg is a utility to install, remove and manage Debian packages.

Options:

```
        -i      Install the package
        -C      Configure an unpackaged package
        -P      Purge all files of a package
        -r      Remove all but the configuration files for a
package
        -u      Unpack a package, but dont configure it
```

----

dpkg-deb

dpkg-deb [**-cefItxX**] FILE [argument]

Perform actions on Debian packages (.debs)

Options:

```
        -c      List contents of filesystem tree
        -e      Extract control files to [argument] directory
        -f      Display control field name starting with
[argument]
        -I      Display the control filenamed [argument]
        -t      Extract filesystem tree to stdout in tar format
        -x      Extract packages filesystem tree to directory
        -X      Verbose extract
```

Example:

```
        $ dpkg-deb -X ./busybox_0.48-1_i386.deb /tmp
```

----

du

du [**-aHLdclsxhmk**] [FILE]...

Summarizes disk space used for each FILE and/or directory. Disk space is printed in units of 1024 bytes.

Options:

```
        -a      show sizes of files in addition to directories
        -H      follow symbolic links that are FILE command line
args
        -L      follow all symbolic links encountered
        -d N    limit output to directories (and files with -a) of
depth < N
        -c      output a grand total
        -l      count sizes many times if hard linked
        -s      display only a total for each argument
        -x      skip directories on different filesystems
        -h      print sizes in human readable format (e.g., 1K
243M 2G )
        -m      print sizes in megabytes
        -k      print sizes in kilobytes(default)
```

Example:

```
        $ du
        16      ./CVS
        12      ./kernel-patches/CVS
        80      ./kernel-patches
        12      ./tests/CVS
        36      ./tests
        12      ./scripts/CVS
        16      ./scripts
        12      ./docs/CVS
        104     ./docs
        2417    .
```

----

dumpkmap

dumpkmap > keymap

Prints out a binary keyboard translation table to standard output.

Example:

```
        $ dumpkmap > keymap
```

----

dumpleases

dumpleases [**-r**|**-a**] [**-f** LEASEFILE]

Displays the DHCP leases granted by udhcpd.

Options:

```
        -f,     --file=FILENAME Leases file to load
        -r,     --remaining     Interepret lease times as time
remaing
        -a,     --absolute      Interepret lease times as expire
time
```

----

echo

echo [**-neE**] [ARG ...]

Prints the specified ARGs to stdout

Options:

```
        -n      suppress trailing newline
        -e      interpret backslash-escaped characters (i.e.,
\t=tab)
        -E      disable interpretation of backslash-escaped
characters
```

Example:

```
        $ echo "Erik is cool"
        Erik is cool
        $  echo -e "Erik\nis\ncool"
        Erik
        is
        cool
        $ echo "Erik\nis\ncool"
        Erik\nis\ncool
```

----

env

env [**-iu**] [-] [name=value]... [command]

Prints the current environment or runs a program after setting up the specified environment.

Options:

```
        -, -i   start with an empty environment
        -u      remove variable from the environment
```

----

expr

expr EXPRESSION

Prints the value of EXPRESSION to standard output.

EXPRESSION may be:

```
        ARG1 |  ARG2    ARG1 if it is neither null nor 0,
otherwise ARG2
        ARG1 &  ARG2    ARG1 if neither argument is null or 0,
otherwise 0
        ARG1 <  ARG2    ARG1 is less than ARG2
        ARG1 <= ARG2    ARG1 is less than or equal to ARG2
        ARG1 =  ARG2    ARG1 is equal to ARG2
        ARG1 != ARG2    ARG1 is unequal to ARG2
        ARG1 >= ARG2    ARG1 is greater than or equal to ARG2
        ARG1 >  ARG2    ARG1 is greater than ARG2
        ARG1 +  ARG2    arithmetic sum of ARG1 and ARG2
        ARG1 -  ARG2    arithmetic difference of ARG1 and ARG2
        ARG1 *  ARG2    arithmetic product of ARG1 and ARG2
        ARG1 /  ARG2    arithmetic quotient of ARG1 divided by
ARG2
        ARG1 %  ARG2    arithmetic remainder of ARG1 divided by
ARG2
        STRING : REGEXP             anchored pattern match of
REGEXP in STRING
        match STRING REGEXP         same as STRING : REGEXP
        substr STRING POS LENGTH    substring of STRING, POS
counted from 1
        index STRING CHARS          index in STRING where any
CHARS is found,
                                    or 0
        length STRING               length of STRING
        quote TOKEN                 interpret TOKEN as a string,
even if
                                    it is a keyword like `match'
or an
                                    operator like `/'
        ( EXPRESSION )              value of EXPRESSION
```

Beware that many operators need to be escaped or quoted for shells. Comparisons are arithmetic if both ARGs are numbers, else lexicographical. Pattern matches return the string matched between \(and \) or null; if \(and \) are not used, they return the number of characters matched or 0.

----

false

false

Return an exit code of FALSE (1).

Example:

```
        $ false
        $ echo $?
        1
```

----

fbset

fbset [options] [mode]

Show and modify frame buffer settings

Example:

```
        $ fbset
        mode "1024x768-76"
                # D: 78.653 MHz, H: 59.949 kHz, V: 75.694 Hz
                geometry 1024 768 1024 768 16
                timings 12714 128 32 16 4 128 4
                accel false
                rgba 5/11,6/5,5/0,0/0
        endmode
```

----

fdflush

fdflush DEVICE

Forces floppy disk drive to detect disk change

----

fdformat

fdformat [**-n**] DEVICE

Low-level formats a floppy disk

Options:

```
        -n      Don't verify after format
```

----

fdisk

fdisk [**-l**] [**-v**] [**-b** SSZ] [**-u**] DISK

Change partition table Options:

```
        -l  List partition table(s)
        -u  Give Start and End in sector (instead of cylinder)
units
        -s PARTITION  Give partition size(s) in blocks
        -b 2048: (for certain MO disks) use 2048-byte sectors
        -v  Give fdisk version
```

----

find

find [PATH...] [EXPRESSION]

Search for files in a directory hierarchy. The default PATH is the current directory; default EXPRESSION is '**-print**'

EXPRESSION may consist of:

```
        -follow         Dereference symbolic links.
        -name PATTERN   File name (leading directories removed)
matches PATTERN.
        -print          Print (default and assumed).
        -type X         Filetype matches X (where X is one of:
f,d,l,b,c,...)
        -perm PERMS     Permissions match any of (+NNN); all of (-
NNN);
                        or exactly (NNN)
        -mtime TIME     Modified time is greater than (+N); less
than (-N);
                        or exactly (N) days
        -newer FILE     Modified time is more recent than FILE's
        -inum N         File has inode number N
```

Example:

```
        $ find / -name passwd
        /etc/passwd
```

----

fold

fold [**-bsw**] [FILE]

Wrap input lines in each FILE (standard input by default), writing to standard output.

Options:

```
        -b      count bytes rather than columns
        -s      break at spaces
        -w      use WIDTH columns instead of 80
```

----

free

free

Displays the amount of free and used system memory

Example:

```
        $ free
```

```
                      total         used         free       shared
buffers
          Mem:       257628       248724         8904        59644
93124
         Swap:       128516         8404       120112
        Total:       386144       257128       129016
```

----

freeramdisk

freeramdisk DEVICE

Frees all memory used by the specified ramdisk.

Example:

```
        $ freeramdisk /dev/ram2
```

----

fsck.minix

fsck.minix [**-larvsmf**] /dev/name

Performs a consistency check for MINIX filesystems.

Options:

```
        -l      Lists all filenames
        -r      Perform interactive repairs
        -a      Perform automatic repairs
        -v      verbose
        -s      Outputs super-block information
        -m      Activates MINIX-like "mode not cleared" warnings
        -f      Force file system check.
```

----

ftpget

ftpget [options] remote-host local-file remote-file

Retrieve a remote file via FTP.

Options:

```
        -c, --continue         Continue a previous transfer
        -v, --verbose          Verbose
        -u, --username         Username to be used
        -p, --password         Password to be used
        -P, --port             Port number to be used
```

----

ftpput

ftpput [options] remote-host remote-file local-file

Store a local file on a remote machine via FTP.

Options:

```
        -v, --verbose          Verbose
        -u, --username         Username to be used
        -p, --password         Password to be used
        -P, --port             Port number to be used
```

----

getopt

getopt [OPTIONS]...

Parse command options

```
        -a, --alternative               Allow long options
starting with single -
        -l, --longoptions=longopts      Long options to be
recognized
        -n, --name=progname             The name under which
errors are reported
        -o, --options=optstring Short options to be recognized
        -q, --quiet                     Disable error reporting by
getopt(3)
        -Q, --quiet-output              No normal output
        -s, --shell=shell               Set shell quoting
conventions
        -T, --test                      Test for getopt(1) version
        -u, --unqote                    Do not quote the output
```

Example:

```
        $ cat getopt.test
        #!/bin/sh
        GETOPT=`getopt -o ab:c:: --long a-long,b-long:,c-long:: \
               -n 'example.busybox' -- "$@"`
        if [ $? != 0 ] ; then  exit 1 ; fi
        eval set -- "$GETOPT"
        while true ; do
         case $1 in
           -a|--a-long) echo "Option a" ; shift ;;
           -b|--b-long) echo "Option b, argument `$2'" ; shift 2
;;
           -c|--c-long)
             case "$2" in
               "") echo "Option c, no argument"; shift 2 ;;
```

```
                *)  echo "Option c, argument `$2'" ; shift 2 ;;
              esac ;;
            --) shift ; break ;;
            *) echo "Internal error!" ; exit 1 ;;
          esac
        done
```

----

getty

getty [OPTIONS]... baud_rate,... line [termtype]

Opens a tty, prompts for a login name, then invokes /bin/login

Options:

```
        -h              Enable hardware (RTS/CTS) flow control.
        -i              Do not display /etc/issue before running
login.
        -L              Local line, so do not do carrier detect.
        -m              Get baud rate from modem's CONNECT status
message.
        -w              Wait for a CR or LF before sending
/etc/issue.
        -n              Do not prompt the user for a login name.
        -f issue_file   Display issue_file instead of /etc/issue.
        -l login_app    Invoke login_app instead of /bin/login.
        -t timeout      Terminate after timeout if no username is
read.
        -I initstring   Sets the init string to send before
anything else.
        -H login_host   Log login_host into the utmp file as the
hostname.
```

----

grep

grep [**-ihHnqvs**] PATTERN [FILEs...]

Search for PATTERN in each FILE or standard input.

Options:

```
        -H      prefix output lines with filename where match was
found
        -h      suppress the prefixing filename on output
        -i      ignore case distinctions
        -l      list names of files that match
        -n      print line number with output lines
        -q      be quiet. Returns 0 if result was found, 1
otherwise
        -v      select non-matching lines
        -s      suppress file open/read error messages
```

Example:

```
        $ grep root /etc/passwd
        root:x:0:0:root:/root:/bin/bash
        $ grep ^[rR]oo. /etc/passwd
        root:x:0:0:root:/root:/bin/bash
```

----

gunzip

gunzip [OPTION]... FILE

Uncompress FILE (or standard input if FILE is '-').

Options:

```
        -c      Write output to standard output
        -t      Test compressed file integrity
```

Example:

```
        $ ls -la /tmp/BusyBox*
        -rw-rw-r--    1 andersen andersen   557009 Apr 11 10:55
/tmp/BusyBox-0.43.tar.gz
        $ gunzip /tmp/BusyBox-0.43.tar.gz
        $ ls -la /tmp/BusyBox*
        -rw-rw-r--    1 andersen andersen  1761280 Apr 14 17:47
/tmp/BusyBox-0.43.tar
```

----

gzip

gzip [OPTION]... [FILE]...

Compress `FILE(s)` with maximum compression. When FILE is '-' or unspecified, reads standard input. Implies **-c**.

Options:

```
        -c      Write output to standard output instead of FILE.gz
        -d      decompress
```

Example:

```
        $ ls -la /tmp/busybox*
        -rw-rw-r--    1 andersen andersen  1761280 Apr 14 17:47
/tmp/busybox.tar
        $ gzip /tmp/busybox.tar
        $ ls -la /tmp/busybox*
        -rw-rw-r--    1 andersen andersen   554058 Apr 14 17:49
/tmp/busybox.tar.gz
```

----

halt

halt

Halt the system.

----

hdparm

hdparm [options] [device] ..

Options: **-a** get/set fs readahead

```
        -A   set drive read-lookahead flag (0/1)
        -b   get/set bus state (0 == off, 1 == on, 2 == tristate)
        -B   set Advanced Power Management setting (1-255)
        -c   get/set IDE 32-bit IO setting
        -C   check IDE power mode status
        -d   get/set using_dma flag
        -D   enable/disable drive defect-mgmt
        -f   flush buffer cache for device on exit
        -g   display drive geometry
        -h   display terse usage information
        -i   display drive identification
        -I   detailed/current information directly from drive
        -Istdin  similar to -I, but wants /proc/ide/*/hd?/identify
as input
        -k   get/set keep_settings_over_reset flag (0/1)
        -K   set drive keep_features_over_reset flag (0/1)
        -L   set drive doorlock (0/1) (removable harddisks only)
        -m   get/set multiple sector count
        -n   get/set ignore-write-errors flag (0/1)
        -p   set PIO mode on IDE interface chipset (0,1,2,3,4,...)
        -P   set drive prefetch count
        -q   change next setting quietly
        -Q   get/set DMA tagged-queuing depth (if supported)
        -r   get/set readonly flag (DANGEROUS to set)
        -R   register an IDE interface (DANGEROUS)
        -S   set standby (spindown) timeout
        -t   perform device read timings
        -T   perform cache read timings
        -u   get/set unmaskirq flag (0/1)
        -U   un-register an IDE interface (DANGEROUS)
        -v   defaults; same as -mcudkrag for IDE drives
        -V   display program version and exit immediately
        -w   perform device reset (DANGEROUS)
        -W   set drive write-caching flag (0/1) (DANGEROUS)
        -x   tristate device for hotswap (0/1) (DANGEROUS)
```

```
-X   set IDE xfer mode (DANGEROUS)
-y   put IDE drive in standby mode
-Y   put IDE drive to sleep
-Z   disable Seagate auto-powersaving mode
-z   re-read partition table
```

----

head

head [OPTION]... [FILE]...

Print 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 -, read standard input.

Options:

```
-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
```

Example:

```
$ head -n 2 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
```

----

hexdump

hexdump [-[bcdefnosvx]] [OPTION] FILE

The hexdump utility is a filter which displays the specified files, or the standard input, if no files are specified, in a user specified format

```
-b              One-byte octal display
-c              One-byte character display
-d              Two-byte decimal display
-e FORMAT STRING
-f FORMAT FILE
-n LENGTH       Interpret only length bytes of input
-o              Two-byte octal display
-s OFFSET       Skip offset byte
-v              display all input data
-x              Two-byte hexadecimal display
```

----

hostid

hostid

Print out a unique 32-bit identifier for the machine.

----

hostname

hostname [OPTION] {hostname | **-F** FILE}

Get or set the hostname or DNS domain name. If a hostname is given (or FILE with the **-F** parameter), the host name will be set.

Options:

```
        -s      Short
        -i      Addresses for the hostname
        -d      DNS domain name
        -f      Fully qualified domain name
        -F FILE Use the contents of FILE to specify the hostname
```

Example:

```
        $ hostname
        sage
```

----

httpd

httpd [**-c** <conf file>] [**-p** <port>] [**-u** user] [**-r** <realm>] [**-m** pass] [**-d**/**-e** <string>]

Listens for incoming http server requests.

Options:

```
        -c FILE         Specifies configuration file. (default
httpd.conf)
        -p PORT Server port (default 80)
        -u USER Set uid to USER after listening privilegies port
        -r REALM        Authentication Realm for Basic
Authentication
        -m PASS         Crypt PASS with md5 algorithm
        -e STRING       Html encode STRING
        -d STRING       URL decode STRING
```

----

hwclock

hwclock [**-r**|**--show**] [**-s**|**--hctosys**] [**-w**|**--systohc**] [**-l**|**--localtime**] [**-u**|**--utc**]

Query and set the hardware clock (RTC)

Options:

```
        -r      read hardware clock and 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
```

----

id

id [OPTIONS]... [USERNAME]

Print information for USERNAME or the current user

Options:

```
        -c      prints only the security context
        -g      prints only the group ID
        -u      prints only the user ID
        -n      print a name instead of a number
        -r      prints the real user ID instead of the effective
ID
```

Example:

```
        $ id
        uid=1000(andersen) gid=1000(andersen)
```

----

ifconfig

ifconfig [**-a**] <interface> [<address>]

configure a network interface

Options: [add <address>[/<prefixlen>]] [del <address>[/<prefixlen>]]

```
        [[-]broadcast [<address>]]  [[-]pointopoint [<address>]]
        [netmask <address>]  [dstaddr <address>]
        [outfill <NN>] [keepalive <NN>]
        [hw ether <address>]  [metric <NN>]  [mtu <NN>]
        [[-]trailers]  [[-]arp]  [[-]allmulti]
        [multicast]  [[-]promisc]  [txqueuelen <NN>]  [[-]dynamic]
        [mem_start <NN>]  [io_addr <NN>]  [irq <NN>]
        [up|down] ...
```

----

ifdown

ifdown <**-ahinv**> <ifaces...>

ifdown <options> <ifaces...>

Options:

```
        -h      this help
        -a      de/configure all interfaces automatically
        -i FILE use FILE for interface definitions
        -n      print out what would happen, but don't do it
                (note that this option doesn't disable mappings)
        -v      print out what would happen before doing it
        -m      don't run any mappings
        -f      force de/configuration
```

----

ifup

ifup <**-ahinv**> <ifaces...>

ifup <options> <ifaces...>

Options:

```
        -h      this help
        -a      de/configure all interfaces automatically
        -i FILE use FILE for interface definitions
        -n      print out what would happen, but don't do it
                        (note that this option doesn't disable
mappings)
        -v      print out what would happen before doing it
        -m      don't run any mappings
        -f      force de/configuration
```

----

inetd

inetd [**-q** len] [conf]

Listens for network connections and launches programs

Option:

```
        -q      Sets the size of the socket listen queue to
                the specified value. Default is 128.
```

----

init

init

Init is the parent of all processes.

This version of init is designed to be run only by the kernel.

BusyBox init doesn't support multiple runlevels. The runlevels field of the /etc/inittab file is completely ignored by BusyBox init. If you want runlevels, use sysvinit.

BusyBox init works just fine without an inittab. If no inittab is found, it has the following default behavior:

```
        ::sysinit:/etc/init.d/rcS
        ::askfirst:/bin/sh
        ::ctrlaltdel:/sbin/reboot
        ::shutdown:/sbin/swapoff -a
        ::shutdown:/bin/umount -a -r
        ::restart:/sbin/init
```

if it detects that /dev/console is _not_ a serial console, it will also run:

```
        tty2::askfirst:/bin/sh
        tty3::askfirst:/bin/sh
        tty4::askfirst:/bin/sh
```

If you choose to use an /etc/inittab file, the inittab entry format is as follows:

```
        <id>:<runlevels>:<action>:<process>
        <id>:
                WARNING: This field has a non-traditional meaning
for BusyBox init!
                The id field is used by BusyBox init to specify
the controlling tty for
                the specified process to run on.  The contents of
this field are
                appended to "/dev/" and used as-is.  There is no
need for this field to
                be unique, although if it isn't you may have
strange results.  If this
                field is left blank, the controlling tty is set to
the console.  Also
                note that if BusyBox detects that a serial console
is in use, then only
                entries whose controlling tty is either the serial
console or /dev/null
                will be run.  BusyBox init does nothing with utmp.
We don't need no
                stinkin' utmp.
        <runlevels>:
                The runlevels field is completely ignored.
```

```
    <action>:
        Valid actions include: sysinit, respawn, askfirst,
wait,
        once, restart, ctrlaltdel, and shutdown.
        The available actions can be classified into two
groups: actions
        that are run only once, and actions that are re-
run when the specified
        process exits.
        Run only-once actions:
            'sysinit' is the first item run on boot.
init waits until all
            sysinit actions are completed before
continuing.  Following the
            completion of all sysinit actions, all
'wait' actions are run.
            'wait' actions, like  'sysinit' actions,
cause init to wait until
            the specified task completes.  'once'
actions are asynchronous,
            therefore, init does not wait for them to
complete.  'restart' is
            the action taken to restart the init
process.  By default this should
            simply run /sbin/init, but can be a script
which runs pivot_root or it
            can do all sorts of other interesting
things.  The 'ctrlaltdel' init
            actions are run when the system detects
that someone on the system
           console has pressed the CTRL-ALT-DEL key
combination.  Typically one
           wants to run 'reboot' at this point to
cause the system to reboot.
            Finally the 'shutdown' action specifies
the actions to taken when
           init is told to reboot.  Unmounting
filesystems and disabling swap
           is a very good here
        Run repeatedly actions:
            'respawn' actions are run after the 'once'
actions.  When a process
            started with a 'respawn' action exits,
init automatically restarts
            it.  Unlike sysvinit, BusyBox init does
not stop processes from
            respawning out of control.  The 'askfirst'
actions acts just like
            respawn, except that before running the
specified process it
            displays the line "Please press Enter to
activate this console."
            and then waits for the user to press enter
before starting the
            specified process.
        Unrecognized actions (like initdefault) will cause
init to emit an
```

```
                error message, and then go along with its
business.  All actions are
                run in the order they appear in /etc/inittab.
        <process>:
                Specifies the process to be executed and it's
command line.
```

Example /etc/inittab file:

```
        # This is run first except when booting in single-user
mode.
        #
        ::sysinit:/etc/init.d/rcS

        # /bin/sh invocations on selected ttys
        #
        # Start an "askfirst" shell on the console (whatever that
may be)
        ::askfirst:-/bin/sh
        # Start an "askfirst" shell on /dev/tty2-4
        tty2::askfirst:-/bin/sh
        tty3::askfirst:-/bin/sh
        tty4::askfirst:-/bin/sh

        # /sbin/getty invocations for selected ttys
        #
        tty4::respawn:/sbin/getty 38400 tty4
        tty5::respawn:/sbin/getty 38400 tty5


        # Example of how to put a getty on a serial line (for a
terminal)
        #
        #::respawn:/sbin/getty -L ttyS0 9600 vt100
        #::respawn:/sbin/getty -L ttyS1 9600 vt100
        #
        # Example how to put a getty on a modem line.
        #::respawn:/sbin/getty 57600 ttyS2

        # Stuff to do when restarting the init process
        ::restart:/sbin/init

        # Stuff to do before rebooting
        ::ctrlaltdel:/sbin/reboot
        ::shutdown:/bin/umount -a -r
        ::shutdown:/sbin/swapoff -a
```

----

insmod

insmod [OPTION]... MODULE [symbol=value]...

Loads the specified kernel modules into the kernel.

Options:

```
        -f      Force module to load into the wrong kernel
version.
        -k      Make module autoclean-able.
        -v      verbose output
        -L      Lock to prevent simultaneous loads of a module
        -m      Output load map to stdout
        -o NAME Set internal module name to NAME
        -x      do not export externs
```

----

install

install [**-cgmops**] [sources] <dest|directory>

Copies files and set attributes

Options:

```
        -c      copy the file, default
        -g      set group ownership
        -m      set permission modes
        -o      set ownership
        -p      preserve date
        -s      strip symbol tables
```

----

ip

ip [OPTIONS] { address | link | route | tunnel } { COMMAND | help }

ip [OPTIONS] OBJECT { COMMAND | help } where OBJECT := { link | addr | route | tunnel } OPTIONS := { **-f**[amily] { inet | inet6 | link } | **-o**[neline] }

----

ipaddr

ipaddr { {add|del} IFADDR dev STRING |
{show|flush}

[dev STRING] [to PREFIX
] }

ipaddr {add|del} IFADDR dev STRING ipaddr {show|flush} [dev STRING] [scope SCOPE-ID]

```
                        [ to PREFIX ] [ label PATTERN ]
                        IFADDR := PREFIX | ADDR peer PREFIX
                        [ broadcast ADDR ] [ anycast ADDR ]
                        [ label STRING ] [ scope SCOPE-ID ]
                        SCOPE-ID := [ host | link | global |
NUMBER ]
```

----

ipcalc

ipcalc [OPTION]... <ADDRESS>[[/]<NETMASK>] [NETMASK]

Calculate IP network settings from a IP address

Options:

```
        -b      --broadcast     Display calculated broadcast
address.
        -n      --network       Display calculated network
address.
        -m      --netmask       Display default netmask for IP. X
        -p      --prefix        Display the prefix for IP/NETMASK.
-h      --hostname      Display first resolved host name.
        -s      --silent        Don't ever display error messages.
```

----

iplink

iplink

iplink set DEVICE { up | down | arp { on | off } |

```
                        dynamic { on | off } |
                        mtu MTU }
        iplink show [ DEVICE ]
```

----

iproute

iproute { list | flush | { add | del | change | append |
replace | monitor } ROUTE }

iproute { list | flush } SELECTOR iproute get ADDRESS [from ADDRESS iif STRING]

```
                        [ oif STRING ]  [ tos TOS ]
        iproute { add | del | change | append | replace | monitor
} ROUTE
                        SELECTOR := [ root PREFIX ] [ match PREFIX
] [ proto RTPROTO ]
                        ROUTE := [ TYPE ] PREFIX [ tos TOS ] [
proto RTPROTO ]
```

----

iptunnel

iptunnel { add | change | del | show } [NAME]

[mode { ipip | gre | sit }]
[remote ADDR] [local ADDR] [ttl TTL]

iptunnel { add | change | del | show } [NAME]

```
                        [ mode { ipip | gre | sit } ] [ remote ADDR ] [ local ADDR ]
                        [ [i|o]seq ] [ [i|o]key KEY ] [ [i|o]csum ]
                        [ ttl TTL ] [ tos TOS ] [ [no]pmtudisc ] [ dev PHYS_DEV ]
```

----

kill

kill [**-signal**] process-id [process-id ...]

Send a signal (default is SIGTERM) to the specified process(es).

Options:

```
        -l      List all signal names and numbers.
```

Example:

```
        $ ps | grep apache
        252 root     root     S [apache]
        263 www-data www-data S [apache]
        264 www-data www-data S [apache]
        265 www-data www-data S [apache]
        266 www-data www-data S [apache]
        267 www-data www-data S [apache]
        $ kill 252
```

----

killall

killall [**-signal**] process-name [process-name ...]

Send a signal (default is SIGTERM) to the specified process(es).

Options:

```
        -l      List all signal names and numbers.
```

Example:

```
        $ killall apache
```

----

klogd

klogd [**-c** n] [**-n**]

Kernel logger. Options:

```
        -c n    Sets the default log level of console messages to
n.
        -n      Run as a foreground process.
```

----

lash

lash [FILE]... or: sh **-c** command [args]...

The BusyBox LAme SHell (command interpreter)

This command does not yet have proper documentation.

Use lash just as you would use any other shell. It properly handles pipes, redirects, job control, can be used as the shell for scripts, and has a sufficient set of builtins to do what is needed. It does not (yet) support Bourne Shell syntax. If you need things like ``if-then-else'', ``while'', and such use ash or bash. If you just need a very simple and extremely small shell, this will do the job.

----

last

last

Shows listing of the last users that logged into the system

----

length

length STRING

Prints out the length of the specified STRING.

Example:

```
        $ length Hello
        5
```

----

ln

ln [OPTION] TARGET... LINK_NAME|DIRECTORY

Create a link named LINK_NAME or DIRECTORY to the specified TARGET

You may use '--' to indicate that all following arguments are non-options.

Options:

```
        -s      make symbolic links instead of hard links
        -f      remove existing destination files
        -n      no dereference symlinks - treat like normal file
```

Example:

```
        $ ln -s BusyBox /tmp/ls
        $ ls -l /tmp/ls
        lrwxrwxrwx    1 root     root            7 Apr 12 18:39 ls
-> BusyBox*
```

----

loadfont

loadfont < font

Loads a console font from standard input.

Example:

```
        $ loadfont < /etc/i18n/fontname
```

----

loadkmap

loadkmap < keymap

Loads a binary keyboard translation table from standard input.

Example:

```
        $ loadkmap < /etc/i18n/lang-keymap
```

----

logger

logger [OPTION]... [MESSAGE]

Write MESSAGE to the system log. If MESSAGE is omitted, log stdin.

Options:

```
        -s      Log to stderr as well as the system log.
        -t TAG  Log using the specified tag (defaults to user
name).
        -p PRIORITY     Enter the message with the specified
priority.
                This may be numerical or a ``facility.level''
pair.
```

Example:

```
        $ logger "hello"
```

----

login

login [OPTION]... [username] [ENV=VAR ...]

Begin a new session on the system

Options:

```
        -f      Do not authenticate (user already authenticated)
        -h      Name of the remote host for this login.
        -p      Preserve environment.
```

----

logname

logname

Print the name of the current user.

Example:

```
        $ logname
        root
```

----

logread

logread [OPTION]...

Shows the messages from syslogd (using circular buffer).

Options:

```
        -f              output data as the log grows
```

----

losetup

losetup [OPTION]... LOOPDEVICE FILE or: losetup [OPTION]... **-d** LOOPDEVICE

Associate LOOPDEVICE with FILE.

Options:

```
        -d              Disassociate LOOPDEVICE.
        -o OFFSET       Start OFFSET bytes into FILE.
```

----

ls

ls [**-1AacCdeFilnpLRrSsTtuvwxXhkK**] [filenames...]

List directory contents

Options:

```
        -1      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
        -F      append indicator (one of */=@|) to entries
        -i      list the i-node for each file
        -l      use a long listing format
        -n      list numeric UIDs and GIDs instead of names
        -p      append indicator (one of /=@|) to entries
        -L      list entries pointed to by symbolic links
        -R      list subdirectories recursively
        -r      sort the listing in reverse order
        -S      sort the listing by file size
        -s      list the size of each file, in blocks
        -T NUM  assume Tabstop every NUM columns
        -t      with -l: show modification time
        -u      with -l: show access time
        -v      sort the listing by version
        -w NUM  assume the terminal is NUM columns wide
        -x      list entries by lines instead of by columns
        -X      sort the listing by extension
        -h      print sizes in human readable format (e.g., 1K
243M 2G )
        -k      print security context
        -K      print security context in long format
```

----

lsmod

lsmod

List the currently loaded kernel modules.

----

makedevs

makedevs NAME TYPE MAJOR MINOR FIRST LAST [s]

Creates a range of block or character special files

TYPEs include:

```
        b:      Make a block (buffered) device.
        c or u: Make a character (un-buffered) device.
        p:      Make a named pipe. MAJOR and MINOR are ignored for
named pipes.
```

FIRST specifies the number appended to NAME to create the first device. LAST specifies the number of the last item that should be created. If 's' is the last argument, the base device is created as well.

For example:

```
        makedevs /dev/ttyS c 4 66 2 63   ->  ttyS2-ttyS63
        makedevs /dev/hda b 3 0 0 8 s    ->  hda,hda1-hda8
```

Example:

```
        # makedevs /dev/ttyS c 4 66 2 63
        [creates ttyS2-ttyS63]
        # makedevs /dev/hda b 3 0 0 8 s
        [creates hda,hda1-hda8]
```

----

md5sum

md5sum [OPTION] [FILEs...] or: md5sum [OPTION] **-c** [FILE]

Print or check MD5 checksums.

Options: With no FILE, or when FILE is -, read standard input.

```
        -c      check MD5 sums against given list
```

The following two options are useful only when verifying checksums:

```
        -s      don't output anything, status code shows success
        -w      warn about improperly formated MD5 checksum lines
```

Example:

```
        $ md5sum < busybox
        6fd11e98b98a58f64ff3398d7b324003
        $ md5sum busybox
        6fd11e98b98a58f64ff3398d7b324003  busybox
        $ md5sum -c -
        6fd11e98b98a58f64ff3398d7b324003  busybox
        busybox: OK
        ^D
```

----

mesg

mesg [y|n]

mesg controls write access to your terminal

```
        y       Allow write access to your terminal.
        n       Disallow write access to your terminal.
```

----

minit

minit [**-spPrRC**]

A small replacement for SysV init

----

mkdir

mkdir [OPTION] DIRECTORY...

Create the `DIRECTORY(ies)` if they do not already exist

Options:

```
        -m      set permission mode (as in chmod), not rwxrwxrwx -
umask
```

```
        -p      no error if existing, make parent directories as
needed
```

Example:

```
        $ mkdir /tmp/foo
        $ mkdir /tmp/foo
        /tmp/foo: File exists
        $ mkdir /tmp/foo/bar/baz
        /tmp/foo/bar/baz: No such file or directory
        $ mkdir -p /tmp/foo/bar/baz
```

----

mkfifo

mkfifo [OPTIONS] name

Creates a named pipe (identical to 'mknod name p')

Options:

```
        -m      create the pipe using the specified mode (default
a=rw)
```

----

mkfs.minix

mkfs.minix [**-c** | **-l** filename] [**-nXX**] [**-iXX**] /dev/name [blocks]

Make a MINIX filesystem.

Options:

```
        -c              Check the device for bad blocks
        -n [14|30]      Specify the maximum length of filenames
        -i INODES       Specify the number of inodes for the
filesystem
        -l FILENAME     Read the bad blocks list from FILENAME
        -v              Make a Minix version 2 filesystem
```

----

mknod

mknod [OPTIONS] NAME TYPE MAJOR MINOR

Create a special file (block, character, or pipe).

Options:

```
        -m      create the special file using the specified mode
(default a=rw)
```

TYPEs include:

```
        b:      Make a block (buffered) device.
        c or u: Make a character (un-buffered) device.
        p:      Make a named pipe. MAJOR and MINOR are ignored for
named pipes.
```

Example:

```
        $ mknod /dev/fd0 b 2 0
        $ mknod -m 644 /tmp/pipe p
```

----

mkswap

mkswap [**-c**] [**-v0**|**-v1**] device [block-count]

Prepare a disk partition to be used as a swap partition.

Options:

```
        -c              Check for read-ability.
        -v0             Make version 0 swap [max 128 Megs].
        -v1             Make version 1 swap [big!] (default for
kernels >
                        2.1.117).
        block-count     Number of block to use (default is entire
partition).
```

----

mktemp

mktemp [**-q**] TEMPLATE

Creates a temporary file with its name based on TEMPLATE. TEMPLATE is any name with six `Xs' (i.e., /tmp/temp.XXXXXX).

Example:

```
        $ mktemp /tmp/temp.XXXXXX
        /tmp/temp.mWiLjM
        $ ls -la /tmp/temp.mWiLjM
        -rw-------    1 andersen andersen        0 Apr 25 17:10
/tmp/temp.mWiLjM
```

----

modprobe

modprobe [FILE ...]

Used for high level module loading and unloading.

Example:

```
        $ modprobe cdrom
```

----

more

more [FILE ...]

More is a filter for viewing FILE one screenful at a time.

Example:

```
        $ dmesg | more
```

----

mount

mount [flags] DEVICE NODE [**-o** options,more-options]

Mount a filesystem. Autodetection of filesystem type requires the /proc filesystem be already mounted.

Flags:

```
        -a:             Mount all filesystems in fstab.
        -f:             "Fake" Add entry to mount table but don't
mount it.
        -n:             Don't write a mount table entry.
        -o option:      One of many filesystem options, listed
below.
        -r:             Mount the filesystem read-only.
        -t fs-type:     Specify the filesystem type.
        -w:             Mount for reading and writing (default).
```

Options for use with the ``**-o**" flag:

```
        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.
        loop:           Mounts a file via loop device.
        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:           Use the linux 2.4.x "bind" feature.
```

There are EVEN MORE flags that are specific to each filesystem. You'll have to see the written documentation for those filesystems.

Example:

```
        $ mount
        /dev/hda3 on / type minix (rw)
        proc on /proc type proc (rw)
        devpts on /dev/pts type devpts (rw)
        $ mount /dev/fd0 /mnt -t msdos -o ro
        $ mount /tmp/diskimage /opt -t ext2 -o loop
```

----

msvc

msvc -[udorspchaitkx] service

[option] service Where option is one of

```
        -u      Up.  If the service is not running, start it.  If
the service stops, restart it.
        -d      Down.  If the service is running, stop it, do not
restart it.
        -o      Once.  If the service is not running, start it.
Do not restart it if it stops.
        -r      Tell supervise that the service is normally
running; this affects status messages.
        -s      Tell supervise that the service is normally
stopped; this affects status messages.
        -p      Pause.  Send the service a STOP signal.
        -c      Continue.  Send the service a CONT signal.
        -h      Hangup.  Send the service a HUP signal.
        -a      Alarm.  Send the service an ALRM signal.
        -i      Interrupt.  Send the service an INT signal.
        -t      Terminate.  Send the service a TERM signal.
        -k      Kill.  Send the service a KILL signal.
        -x      Exit.  supervise will quit as soon as the service
is down.
```

----

mt

mt [**-f** device] opcode value

Control magnetic tape drive operation

Available Opcodes:

bsf bsfm bsr bss datacompression drvbuffer eof eom erase fsf fsfm fsr fss load lock mkpart nop offline ras1 ras2 ras3 reset retension rewind rewoffline seek setblk setdensity setpart tell unload unlock weof wset

----

mv

mv SOURCE DEST or: mv SOURCE... DIRECTORY

Rename SOURCE to DEST, or move `SOURCE(s)` to DIRECTORY.

Example:

```
        $ mv /tmp/foo /bin/bar
```

----

nameif

nameif [**-s**] [**-c** FILE] [{IFNAME MACADDR}]

Nameif renaming network interface while it in the down state.

Options:

```
        -c FILE         Use configuration file (default is
/etc/mactab)
        -s              Use syslog (LOCAL0 facility).
        IFNAME MACADDR  new_interface_name interface_mac_address
```

Example:

```
        $ nameif -s dmz0 00:A0:C9:8C:F6:3F
         or
        $ nameif -c /etc/my_mactab_file
```

----

nc

nc [OPTIONS] [IP] [port]

Netcat opens a pipe to IP:port

Options:

```
        -l              listen mode, for inbound connects
        -p PORT         local port number
        -i SECS         delay interval for lines sent
        -e PROG         program to exec after connect (dangerous!)
```

Example:

```
        $ nc foobar.somedomain.com 25
        220 foobar ESMTP Exim 3.12 #1 Sat, 15 Apr 2000 00:03:02 -
0600
        help
        214-Commands supported:
        214-    HELO EHLO MAIL RCPT DATA AUTH
        214     NOOP QUIT RSET HELP
        quit
        221 foobar closing connection
```

----

netstat

netstat [**-laenrtuwx**]

Netstat displays Linux networking information.

Options:

```
        -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
```

----

nslookup

nslookup [HOST] [SERVER]

Queries the nameserver for the IP address of the given HOST optionally using a specified DNS server

Example:

```
        $ nslookup localhost
        Server:     default
        Address:    default

        Name:       debian
        Address:    127.0.0.1
```

----

od

od [**-aBbcDdeFfHhIiLlOovXx**] [FILE]

Write an unambiguous representation, octal bytes by default, of FILE to standard output. With no FILE, or when FILE is -, read standard input.

----

openvt

openvt <vtnum> <COMMAND> [ARGS...]

Start a command on a new virtual terminal

Example:

```
        openvt 2 /bin/ash
```

----

passwd

passwd [OPTION] [name]

Change a user password. If no name is specified, changes the password for the current user. Options:

```
        -a      Define which algorithm shall be used for the 
password.
                        (Choices: des, md5 
PASSWORD_ALG_TYPES(", sha1") )
        -d      Delete the password for the specified user 
account.
        -l      Locks (disables) the specified user account.
        -u      Unlocks (re-enables) the specified user account.
```

----

patch

patch [**-p**<num>]

[**-p**<num>]

Example:

```
        $ patch -p1 <example.diff
```

----

pidfilehack

pidfilehack [daemon.pid] [daemon]

service /var/run/daemon.pid /usr/sbin/daemon args...

----

pidof

pidof process-name [process-name ...]

Lists the PIDs of all processes with names that match the names on the command line

Example:

```
        $ pidof init
        1
```

----

ping

ping [OPTION]... host

Send ICMP ECHO_REQUEST packets to network hosts.

Options:

```
        -c COUNT        Send only COUNT pings.
        -s SIZE         Send SIZE data bytes in packets
(default=56).
        -q              Quiet mode, only displays output at start
                        and when finished.
```

Example:

```
        $ ping localhost
        PING slag (127.0.0.1): 56 data bytes
        64 bytes from 127.0.0.1: icmp_seq=0 ttl=255 time=20.1 ms

        --- debian ping statistics ---
        1 packets transmitted, 1 packets received, 0% packet loss
        round-trip min/avg/max = 20.1/20.1/20.1 ms
```

----

ping6

ping6 [OPTION]... host

Send ICMP ECHO_REQUEST packets to network hosts.

Options:

```
        -c COUNT        Send only COUNT pings.
        -s SIZE         Send SIZE data bytes in packets
(default=56).
        -q              Quiet mode, only displays output at start
                        and when finished.
```

Example:

```
        $ ping6 ip6-localhost
        PING ip6-localhost (::1): 56 data bytes
        64 bytes from ::1: icmp6_seq=0 ttl=64 time=20.1 ms

        --- ip6-localhost ping statistics ---
        1 packets transmitted, 1 packets received, 0% packet loss
        round-trip min/avg/max = 20.1/20.1/20.1 ms
```

----

pivot_root

pivot_root NEW_ROOT PUT_OLD

Move the current root file system to PUT_OLD and make NEW_ROOT the new root file system.

----

poweroff

poweroff

Halt the system and request that the kernel shut off the power.

----

printf

printf FORM AT [ARGUMENT...]

Formats and prints `ARGUMENT(s)` according to FORMAT, Where FORMAT controls the output exactly as in C printf.

Example:

```
        $ printf "Val=%d\n" 5
        Val=5
```

----

ps

ps

Report process status

```
        This version of ps accepts no options.
Options:
        -c      show SE Linux context
```

Example:

```
        $ ps
          PID  Uid      Gid State Command
            1 root     root     S init
            2 root     root     S [kflushd]
            3 root     root     S [kupdate]
            4 root     root     S [kpiod]
            5 root     root     S [kswapd]
          742 andersen andersen S [bash]
          743 andersen andersen S -bash
          745 root     root     S [getty]
         2990 andersen andersen R ps
```

----

pwd

pwd

Print the full filename of the current working directory.

Example:

```
        $ pwd
        /root
```

----

raid_start

raid_start MD_DEVICE DISK_DEVICE

Start MD_DEVICE, taking superblock from DISK_DEVICE. Example: raid_start /dev/md0 /dev/sdb

----

rdate

rdate [**-sp**] HOST

Get and possibly set the system date and time from a remote HOST.

Options:

```
        -s      Set the system date and time (default).
        -p      Print the date and time.
```

----

readlink

readlink

Displays the value of a symbolic link.

----

realpath

realpath pathname ...

Returns the absolute pathnames of given argument.

----

reboot

reboot

Reboot the system.

----

renice

renice priority pid [pid ...]

Changes priority of running processes. Allowed priorities range from 20 (the process runs only when nothing else is running) to 0 (default priority) to **-20** (almost nothing else ever gets to run).

----

reset

reset

Resets the screen.

----

rm

rm [OPTION]... FILE...

Remove (unlink) the FILE(s). You may use '--' to indicate that all following arguments are non-options.

Options:

```
        -i              always prompt before removing each
destination
        -f              remove existing destinations, never prompt
        -r or -R        remove the contents of directories
recursively
```

Example:

```
        $ rm -rf /tmp/foo
```

----

rmdir

rmdir [OPTION]... DIRECTORY...

Remove the DIRECTORY(ies), if they are empty.

Example:

```
        # rmdir /tmp/foo
```

----

rmmod

rmmod [OPTION]... [MODULE]...

Unloads the specified kernel modules from the kernel.

Options:

```
        -a      Remove all unused modules (recursively)
```

Example:

```
        $ rmmod tulip
```

----

route

route [{add|del|delete}]

Edit the kernel's routing tables.

Options:

```
        -n              Dont resolve names.
        -e              Display other/more information.
        -A inet{6}      Select address family.
```

----

rpm

rpm **-i -q**[ildc]p package.rpm

Manipulates RPM packages

Options:

```
        -i Install package
        -q Query package
        -p Query uninstalled package
        -i Show information
        -l List contents
        -d List documents
        -c List config files
```

----

rpm2cpio

rpm2cpio package.rpm

Outputs a cpio archive of the rpm file.

----

run-parts

run-parts [**-t**] [**-a** ARG] [**-u** MASK] DIRECTORY

Run a bunch of scripts in a directory.

Options:

```
        -t      Prints what would be run, but does not actually
run anything.
        -a ARG  Pass ARG as an argument for every program invoked.
        -u MASK Set the umask to MASK before executing every
program.
```

----

rx

rx FILE

Receive a file using the xmodem protocol.

Example:

```
        $ rx /tmp/foo
```

----

sed

sed [**-nef**] pattern [files...]

Options:

```
        -n              suppress automatic printing of pattern
space
        -e script       add the script to the commands to be
executed
        -f scriptfile   add script-file contents to the
                        commands to be executed
        -i              Edit files in-place
```

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.

Example:

```
        $ echo "foo" | sed -e 's/f[a-zA-Z]o/bar/g'
        bar
```

----

seq

seq [first [increment]] last

Print numbers from FIRST to LAST, in steps of INCREMENT. FIRST, INCREMENT default to 1 Arguments:

```
        LAST
        FIRST   LAST
        FIRST   INCREMENT       LAST
```

----

setkeycodes

setkeycodes SCANCODE KEYCODE ...

Set entries into the kernel's scancode-to-keycode map, allowing unusual keyboards to generate usable keycodes.

SCANCODE may be either xx or e0xx (hexadecimal), and KEYCODE is given in decimal

Example:

```
        $ setkeycodes e030 127
```

----

sha1sum

sha1sum [OPTION] [FILEs...] or: sha1sum [OPTION] **-c** [FILE]

Print or check SHA1 checksums.

Options: With no FILE, or when FILE is **-**, read standard input.

```
        -c      check SHA1 sums against given list
```

The following two options are useful only when verifying checksums:

```
        -s      don't output anything, status code shows success
        -w      warn about improperly formated SHA1 checksum lines
```

----

sleep

sleep [N]...

```
                 Pause for a time equal to the total of the args
given, where each arg can
                have an optional suffix of (s)econds, (m)inutes,
(h)ours, or (d)ays.
```

Example:

```
        $ sleep 2
        [2 second delay results]
        $ sleep 1d 3h 22m 8s
        [98528 second delay results]
```

----

sort

sort [**-nru**] [FILE]...

Sorts lines of text in the specified files

Options:

```
        -u      suppress duplicate lines
        -r      sort in reverse order
        -n      sort numerics
```

Example:

```
        $ echo -e "e\nf\nb\nd\nc\na" | sort
        a
        b
        c
        d
        e
        f
```

----

start-stop-daemon

start-stop-daemon [OPTIONS] [**--start**|**--stop**] ... [-- arguments...]

Program to start and stop services.

Options:

```
        -S|--start                      start
        -K|--stop                       stop
        -a|--startas <pathname>         starts process specified
by pathname
        -b|--background                 force process into
background
        -u|--user <username>|<uid>      stop this user's processes
        -x|--exec <executable>          program to either start or
check
        -m|--make-pidfile <filename>    create the -p file and
enter pid in it
        -n|--name <process-name>        stop processes with this
name
        -p|--pidfile <pid-file>         save or load pid using a
pid-file
        -q|--quiet                      be quiet
        -s|--signal <signal>            signal to send (default
TERM)
```

----

strings

strings [**-afo**] [**-n** length] [file ...]

Display printable strings in a binary file.

Options:

```
        -f      Precede each string with the name of the file
where it was found.
        -n N    Specifies that at least N characters forms a
sequence (default 4)
        -o      Each string is preceded by its decimal offset in
the file.
```

----

stty

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

Without arguments, prints baud rate, line discipline, and deviations from stty sane.

Options:

```
        -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
```

----

su

su [OPTION]... [-] [username]

Change user id or become root. Options:

```
        -p      Preserve environment
```

----

sulogin

sulogin [OPTION]... [tty-device]

Single user login Options:

```
        -f      Do not authenticate (user already authenticated)
        -h      Name of the remote host for this login.
        -p      Preserve environment.
```

----

swapoff

swapoff [OPTION] [DEVICE]

Stop swapping virtual memory pages on DEVICE.

Options:

```
        -a      Stop swapping on all swap devices
```

----

swapon

swapon [OPTION] [DEVICE]

Start swapping virtual memory pages on DEVICE.

Options:

```
        -a      Start swapping on all swap devices
```

----

sync

sync

Write all buffered filesystem blocks to disk.

----

sysctl

sysctl [OPTIONS]... [VALUE]...

sysctl - configure kernel parameters at runtime

Options:

```
        -n      Use this option to disable printing of the key
name when printing values.
        -e      Use this option to ignore errors about unknown
keys.
        -w      Use this option when you want to change a sysctl
setting.
        -p      Load in sysctl settings from the file specified or
/etc/sysctl.conf if none given.
        -a      Display all values currently available.
        -A      Display all values currently available in table
form.
```

Example:

```
        sysctl [-n] [-e] variable ...
        sysctl [-n] [-e] -w variable=value ...
        sysctl [-n] [-e] -a
        sysctl [-n] [-e] -p <file>      (default /etc/sysctl.conf)
        sysctl [-n] [-e] -A
```

----

syslogd

syslogd [OPTION]...

Linux system and kernel logging utility. Note that this version of syslogd ignores /etc/syslog.conf.

Options:

```
        -m MIN          Minutes between MARK lines (default=20,
0=off)
        -n              Run as a foreground process
        -O FILE         Use an alternate log file
(default=/var/log/messages)
        -s SIZE         Max size (KB) before rotate
(default=200KB, 0=off)
        -b NUM          Number of rotated logs to keep (default=1,
max=99, 0=purge)
        -R HOST[:PORT]  Log to IP or hostname on PORT (default
PORT=514/UDP)
        -L              Log locally and via network logging
(default is network only)
        -C [size(KiB)]  Log to a circular buffer (read the buffer
using logread)
```

Example:

```
        $ syslogd -R masterlog:514
        $ syslogd -R 192.168.1.1:601
```

----

tail

tail [OPTION]... [FILE]...

Print last 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 -, read standard input.

Options:

```
        -c N[kbm]       output the last N bytes
        -n N[kbm]       print last N lines instead of last 10
        -f              output data as the file grows
        -q              never output headers giving file names
        -s SEC          wait SEC seconds between reads with -f
```

-v	always output headers giving file names

If the first character of N (bytes or lines) is a '+', output begins with the Nth item from the start of each file, otherwise, print the last N items in the file. N bytes may be suffixed by k (x1024), b (x512), or m (1024^2).

Example:

```
$ tail -n 1 /etc/resolv.conf
nameserver 10.0.0.1
```

----

tar

tar -[czjZxtvO] [**-X** FILE][**-f** TARFILE] [**-C** DIR] [FILE(s)] ...

Create, extract, or list files from a tar file.

Options:

c	create
x	extract
t	list

Archive format selection:

z	Filter the archive through gzip
j	Filter the archive through bzip2
Z	Filter the archive through compress

File selection:

f	name of TARFILE or "-" for stdin
O	extract to stdout
exclude	file to exclude
X	file with names to exclude
C	change to directory DIR before operation
v	verbosely list files processed

Example:

```
$ zcat /tmp/tarball.tar.gz | tar -xf -
$ tar -cf /tmp/tarball.tar /usr/local
```

----

tee

tee [OPTION]... [FILE]...

Copy standard input to each FILE, and also to standard output.

Options:

```
        -a      append to the given FILEs, do not overwrite
        -i      ignore interrupt signals (SIGINT)
```

Example:

```
        $ echo "Hello" | tee /tmp/foo
        $ cat /tmp/foo
        Hello
```

----

telnet

telnet HOST [PORT]

Telnet is used to establish interactive communication with another computer over a network using the TELNET protocol.

----

telnetd

telnetd [OPTION]

Telnetd listens for incoming TELNET connections on PORT. Options:

```
        -p PORT listen for connections on PORT (default 23)
        -l LOGIN        exec LOGIN on connect (default /bin/sh)
```

----

test

test EXPRESSION or [EXPRESSION]

Checks file types and compares values returning an exit code determined by the value of EXPRESSION.

Example:

```
        $ test 1 -eq 2
        $ echo $?
        1
        $ test 1 -eq 1
        $ echo $?
        0
        $ [ -d /etc ]
        $ echo $?
        0
        $ [ -d /junk ]
        $ echo $?
        1
```

----

tftp

tftp [OPTION]... HOST [PORT]

Transfers a file from/to a tftp server using ``octet" mode.

Options:

```
        -l FILE Local FILE.
        -r FILE Remote FILE.
        -g      Get file.
        -p      Put file.
        -b SIZE Transfer blocks of SIZE octets.
```

----

time

time [OPTION]... COMMAND [ARGS...]

Runs the program COMMAND with arguments ARGS. When COMMAND finishes, COMMAND's resource usage information is displayed

Options:

```
        -v      Displays verbose resource usage information.
```

----

top

top [**-d** <seconds>]

top provides an view of processor activity in real time. This utility reads the status for all processes in /proc each <seconds> and shows the status for however many processes will fit on the screen. This utility will not show processes that are started after program startup, but it will show the EXIT status for and PIDs that exit while it is running.

----

touch

touch [**-c**] FILE [FILE ...]

Update the last-modified date on the given FILE[s].

Options:

```
        -c      Do not create any files
```

Example:

```
        $ ls -l /tmp/foo
        /bin/ls: /tmp/foo: No such file or directory
        $ touch /tmp/foo
        $ ls -l /tmp/foo
        -rw-rw-r--    1 andersen andersen        0 Apr 15 01:11
/tmp/foo
```

----

tr

tr [**-cds**] STRING1 [STRING2]

Translate, squeeze, and/or delete characters from standard input, writing to standard output.

Options:

```
        -c      take complement of STRING1
        -d      delete input characters coded STRING1
        -s      squeeze multiple output characters of STRING2 into
one character
```

Example:

```
        $ echo "gdkkn vnqkc" | tr [a-y] [b-z]
        hello world
```

----

traceroute

traceroute [**-dnrv**] [**-m** max_ttl] [**-p** port#] [**-q** nqueries]
[**-s** src_addr] [**-t** tos] [**-w** wait] host
[data size]

trace the route ip packets follow going to ``host" Options:

```
        -d      set SO_DEBUG options to socket
        -n      Print hop addresses numerically rather than
symbolically
```

```
        -r      Bypass the normal routing tables and send directly
to a host
        -v      Verbose output
        -m max_ttl      Set the max time-to-live (max number of
hops)
        -p port#        Set the base UDP port number used in
probes
                (default is 33434)
        -q nqueries     Set the number of probes per ``ttl'' to
nqueries
                (default is 3)
        -s src_addr     Use the following IP address as the source
address
        -t tos  Set the type-of-service in probe packets to the
following value
                (default 0)
        -w wait Set the time (in seconds) to wait for a response
to a probe
                (default 3 sec.).
```

----

true

true

Return an exit code of TRUE (0).

Example:

```
        $ true
        $ echo $?
        0
```

----

tty

tty

Print the file name of the terminal connected to standard input.

Options:

```
        -s      print nothing, only return an exit status
```

Example:

```
        $ tty
        /dev/tty2
```

----

udhcpc

udhcpc [**-fbnqv**] [**-c** CLIENTID] [**-H** HOSTNAME] [**-i** INTERFACE] [**-p** pidfile] [**-r** IP] [**-s** script]

```
        -c,     --clientid=CLIENTID     Client identifier
        -H,     --hostname=HOSTNAME     Client hostname
        -h,                             Alias for -H
        -f,     --foreground    Do not fork after getting lease
        -b,     --background    Fork to background if lease cannot
be immediately negotiated.
        -i,     --interface=INTERFACE   Interface to use (default:
eth0)
        -n,     --now   Exit with failure if lease cannot be
immediately negotiated.
        -p,     --pidfile=file  Store process ID of daemon in file
        -q,     --quit  Quit after obtaining lease
        -r,     --request=IP    IP address to request (default:
none)
        -s,     --script=file   Run file at dhcp events (default:
/usr/share/udhcpc/default.script)
        -v,     --version       Display version
```

----

udhcpd

udhcpd [configfile]

----

umount

umount [flags] FILESYSTEM|DIRECTORY

Unmount file systems

Flags:

```
        -a      Unmount all file systems in /etc/mtab
        -n      Don't erase /etc/mtab entries
        -r      Try to remount devices as read-only if mount is
busy
        -f      Force umount (i.e., unreachable NFS server)
        -l      Do not free loop device (if a loop device has been
used)
```

Example:

```
        $ umount /dev/hdc1
```

----

uname

uname [OPTION]...

Print certain system information. With no OPTION, same as **-s**.

Options:

```
        -a      print all information
        -m      the machine (hardware) type
        -n      print the machine's network node hostname
        -r      print the operating system release
        -s      print the operating system name
        -p      print the host processor type
        -v      print the operating system version
```

Example:

```
        $ uname -a
        Linux debian 2.4.23 #2 Tue Dec 23 17:09:10 MST 2003 i686
GNU/Linux
```

----

uncompress

uncompress [**-c**] [**-f**] [name ...]

Uncompress .Z file[s] Options:

```
        -c      extract to stdout
        -f      force overwrite an existing file
```

----

uniq

uniq [OPTION]... [INPUT [OUTPUT]]

Discard all but one of successive identical lines from INPUT (or standard input), writing to OUTPUT (or standard output).

Options:

```
        -c      prefix lines by the number of occurrences
        -d      only print duplicate lines
        -u      only print unique lines
        -f N    skip the first N fields
        -s N    skip the first N chars (after any skipped fields)
```

Example:

```
        $ echo -e "a\na\nb\nc\nc\na" | sort | uniq
        a
        b
        c
```

----

unix2dos

unix2dos [option] [FILE]

Converts FILE from unix format to dos format. When no option is given, the input is converted to the opposite output format. When no file is given, uses stdin for input and stdout for output. Options:

```
        -u      output will be in UNIX format
        -d      output will be in DOS format
```

----

unzip

unzip [**-opts**[modifiers]] file[.zip] [list] [**-x** xlist] [**-d** exdir]

Extracts files from ZIP archives.

Options:

```
        -l      list archive contents (short form)
        -n      never overwrite existing files (default)
        -o      overwrite files without prompting
        -p      send output to stdout
        -q      be quiet
        -x      exclude these files
        -d      extract files into this directory
```

----

uptime

uptime

Display the time since the last boot.

Example:

```
        $ uptime
          1:55pm  up  2:30, load average: 0.09, 0.04, 0.00
```

----

usleep

usleep N

Pause for N microseconds.

Example:

```
        $ usleep 1000000
        [pauses for 1 second]
```

----

uudecode

uudecode [FILE]...

Uudecode a file that is uuencoded.

Options:

```
        -o FILE direct output to FILE
```

Example:

```
        $ uudecode -o busybox busybox.uu
        $ ls -l busybox
        -rwxr-xr-x   1 ams      ams        245264 Jun  7 21:35
busybox
```

----

uuencode

uuencode [OPTION] [INFILE] REMOTEFILE

Uuencode a file.

Options:

```
        -m      use base64 encoding per RFC1521
```

Example:

```
        $ uuencode busybox busybox
        begin 755 busybox
        <encoded file snipped>
        $ uudecode busybox busybox > busybox.uu
        $
```

----

vconfig

vconfig COMMAND [OPTIONS] ...

vconfig lets you create and remove virtual ethernet devices.

Options:

```
        add              [interface-name] [vlan_id]
        rem              [vlan-name]
        set_flag         [interface-name] [flag-num]       [0 | 1]
        set_egress_map   [vlan-name]      [skb_priority]
[vlan_qos]
        set_ingress_map  [vlan-name]      [skb_priority]
[vlan_qos]
        set_name_type    [name-type]
```

----

vi

vi [OPTION] [FILE]...

edit FILE.

Options:

```
        -R      Read-only- do not write to the file.
```

----

vlock

vlock [OPTIONS]

Lock a virtual terminal. A password is required to unlock Options:

```
        -a      Lock all VTs
```

----

watch

watch [**-n** <seconds>] COMMAND...

Executes a program periodically. Options:

```
        -n      Loop period in seconds - default is 2.
```

Example:

```
        $ watch date
        Mon Dec 17 10:31:40 GMT 2000
        Mon Dec 17 10:31:42 GMT 2000
        Mon Dec 17 10:31:44 GMT 2000
```

----

watchdog

watchdog [**-t** <seconds>] DEV

Periodically write to watchdog device DEV. Options:

```
        -t      Timer period in seconds - default is 30.
```

----

wc

wc [OPTION]... [FILE]...

Print line, word, and byte counts for each FILE, and a total line if more than one FILE is specified. With no FILE, read standard input.

Options:

```
        -c      print the byte counts
        -l      print the newline counts
        -L      print the length of the longest line
        -w      print the word counts
```

Example:

```
        $ wc /etc/passwd
             31      46    1365 /etc/passwd
```

----

wget

wget [**-c**|**--continue**] [**-q**|**--quiet**] [**-O**|**--output**-document file]

[**--header** 'header: value'] [**-Y**|**--proxy** on/off] [**-P** DIR] url

wget retrieves files via HTTP or FTP

Options:

```
        -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')
```

----

which

which [COMMAND ...]

Locates a COMMAND.

Example:

```
        $ which login
        /bin/login
```

----

who

who

Prints the current user names and related information

----

whoami

whoami

Prints the user name associated with the current effective user id.

----

xargs

xargs [COMMAND] [OPTIONS] [ARGS...]

Executes COMMAND on every item given by standard input.

Options:

```
        -p      Prompt the user about whether to run each command
        -r      Do not run command for empty readed lines
        -x      Exit if the size is exceeded
        -0      Input filenames are terminated by a null character
        -t      Print the command line on stderr before executing
it.
```

Example:

```
        $ ls | xargs gzip
        $ find . -name '*.c' -print | xargs rm
```

----

yes

yes [OPTION]... [STRING]...

Repeatedly outputs a line with all specified STRING(s), or 'y'.

----

zcat

zcat FILE

Uncompress to stdout.

LIBC NSS

GNU Libc (glibc) uses the Name Service Switch (NSS) to configure the behavior of the C library for the local environment, and to configure how it reads system data, such as passwords and group information. This is implemented using an /etc/nsswitch.conf configuration file, and using one or more of the /lib/libnss_* libraries. BusyBox tries to avoid using any libc calls that make use of NSS. Some applets however, such as login and su, will use libc functions that require NSS.

If you enable CONFIG_USE_BB_PWD_GRP, BusyBox will use internal functions to directly access the /etc/passwd, /etc/group, and /etc/shadow files without using NSS. This may allow you to run your system without the need for installing any of the NSS configuration files and libraries.

When used with glibc, the BusyBox 'networking' applets will similarly require that you install at least some of the glibc NSS stuff (in particular, /etc/nsswitch.conf, /lib/libnss_dns*, /lib/libnss_files*, and /lib/libresolv*).

Shameless Plug: As an alternative, one could use a C library such as uClibc. In addition to making your system significantly smaller, uClibc does not require the use of any NSS support files or libraries.

MAINTAINER

Erik Andersen <andersen@codepoet.org>

AUTHORS

The following people have contributed code to BusyBox whether they know it or not. If you have written code included in BusyBox, you should probably be listed here so you can obtain your bit of eternal glory. If you should be listed here, or the description of what you have done needs more detail, or is incorect, please send in an update.

Emanuele Aina <emanuele.aina@tiscali.it>

run-parts

Erik Andersen <andersen@codepoet.org>

```
Tons of new stuff, major rewrite of most of the
core apps, tons of new apps as noted in header files.
Lots of tedious effort writing these boring docs that
nobody is going to actually read.
```

Laurence Anderson <l.d.anderson@warwick.ac.uk>

```
rpm2cpio, unzip, get_header_cpio, read_gz interface, rpm
```

Jeff Angielski <jeff@theptrgroup.com>

```
ftpput, ftpget
```

Edward Betts <edward@debian.org>

```
expr, hostid, logname, whoami
```

John Beppu <beppu@codepoet.org>

```
du, nslookup, sort
```

Brian Candler <B.Candler@pobox.com>

```
tiny-ls(ls)
```

Randolph Chung <tausq@debian.org>

```
fbset, ping, hostname
```

Dave Cinege <dcinege@psychosis.com>

```
more(v2), makedevs, dutmp, modularization, auto links file,
various fixes, Linux Router Project maintenance
```

Jordan Crouse <jordan@cosmicpenguin.net>

```
    ipcalc
```

Magnus Damm <damm@opensource.se>

```
tftp client insmod powerpc support
```

Larry Doolittle <ldoolitt@recycle.lbl.gov>

```
pristine source directory compilation, lots of patches and fixes.
```

Glenn Engel <glenne@engel.org>

```
httpd
```

Gennady Feldman <gfeldman@gena01.com>

```
Sysklogd (single threaded syslogd, IPC Circular buffer support,
logread), various fixes.
```

Karl M. Hegbloom <karlheg@debian.org>

```
cp_mv.c, the test suite, various fixes to utility.c, &c.
```

Daniel Jacobowitz <dan@debian.org>

```
mktemp.c
```

Matt Kraai <kraai@alumni.cmu.edu>

```
documentation, bugfixes, test suite
```

Stephan Linz <linz@li-pro.net>

```
        ipcalc, Red Hat equivalence
```

John Lombardo <john@deltanet.com>

```
    tr
```

Glenn McGrath <bug1@optushome.com.au>

```
    ar, dpkg, dpkg-deb
```

Manuel Novoa III <mjn3@codepoet.org>

```
    cat, head, mkfifo, mknod, rmdir, sleep, tee, tty, uniq, usleep, wc,
yes,
    mesg, vconfig, make_directory, parse_mode, dirname, mode_string,
    get_last_path_component, simplify_path, and a number trivial libbb
routines
    also bug fixes, partial rewrites, and size optimizations in
    ash, basename, cal, cmp, cp, df, du, echo, env, ln, logname, md5sum,
mkdir,
    mv, realpath, rm, sort, tail, touch, uname, watch, arith,
human_readable,
    interface, dutmp, ifconfig, route
```

Vladimir Oleynik <dzo@simtreas.ru>

```
    cmdedit; xargs(current), httpd(current);
    ports: ash, crond, fdisk, inetd, stty, traceroute, top;
    locale, various fixes
    and irreconcilable critic of everything not perfect.
```

Bruce Perens <bruce@pixar.com>

```
    Original author of BusyBox in 1995, 1996. Some of his code can
    still be found hiding here and there...
```

Tim Riker <Tim@Rikers.org>

```
    bug fixes, member of fan club
```

Kent Robotti <robotti@metconnect.com>

```
    reset, tons and tons of bug reports and patches.
```

Chip Rosenthal <chip@unicom.com>, <crosenth@covad.com>

```
    wget - Contributed by permission of Covad Communications
```

Pavel Roskin <proski@gnu.org>

```
    Lots of bugs fixes and patches.
```

Gyepi Sam <gyepi@praxis-sw.com>

```
    Remote logging feature for syslogd
```

Linus Torvalds <torvalds@transmeta.com>

```
    mkswap, fsck.minix, mkfs.minix
```

Mark Whitley <markw@codepoet.org>

```
    grep, sed, cut, xargs(previous),
    style-guide, new-applet-HOWTO, bug fixes, etc.
```

Charles P. Wright <cpwright@villagenet.com>

```
    gzip, mini-netcat(nc)
```

Enrique Zanardi <ezanardi@ull.es>

```
    tarcat (since removed), loadkmap, various fixes, Debian maintenance
```

Tito Ragusa <farmatito@tiscali.it>

```
        devfsd and size optimizations in strings, openvt and deallocvt.
```