1 Unix utilities and shell builtins

The GNU Core Utilities are the basic file, shell and text manipulation utilities (marked here with a star*) of the GNU operating system.

1.1 File system

cat* concatenates and prints files (**tac** does the same in reverse, **rev** reverses lines characterwise):

- `A` shows nonprinting characters,
- `b` numbers nonempty output lines,
- `n` numbers all output lines,
- `s` suppresses repeated empty output lines.

chmod changes permissions of a file. `ugoa` controls whose permissions will be changed: the owner (u), the group (g), other users (o) or all users (a). + adds selected file mode bits, - removes them, = adds them and removes the unmentioned bits. `rwx` selects file mode bits for the affected users: read (4), write (2), execute or search for directories (1).

chown changes file owner and group, **chgrp** changes only group ownership.

cksum prints CRC checksum with byte count and **shasum** computes or checks SHA message digests:

- `a` algorithm: 1, 224, 256, 384, 512, 512224 or 512256,
- `b` reads in binary mode,
- `c` reads SHA sums from the „files" and checks them.

cp* copies files and directories:

- `a` never follows symlinks, preserves all attributes,
- `b` makes a backup of each existing destination file,
- `d` nevers follows symlinks in „source",
- `f` removes an existing destination file if needed,
- `i` prompts before overwrite,
- `r` copies directories recursively,
- `l` hard links files instead,
- `s` makes symbolic links instead,
- `t` copies all „source" arguments into „directory",
- `u` copies only when the source file is newer than the destination file.

dd if=file of=file bs=bytes count=n converts and copies a file:

- `if` reads from a file instead of stdin
- `of` writes to a file instead of stdout
- `bs` reads and writes up to „bytes" bytes at a time
- `count` copies only „n" input blocks

df reports file system disk space usage.

du* estimates file space usage:

- `a` writes counts for all files, not just directories,
- `h` prints sizes in human readable format,
- `s` diplays only a total.

file determines file type.

fsck checks and repairs a Linux filesystem.

fuser identifies processes using files or sockets.

ln* makes hard links between files (only in the same file system, does not work with directories): `ln -s` makes symbolic links instead.

ls lists directory contents:

- `a` does not ignore entries starting with dot
- `F` appends indicator to entries
- `h` prints human readable sizes
- `i` prints the index number of each file
- `l` prints permissions, number of hard links, owner, group, size, last-modified date as well.
- `r` reverses order while sorting
- `R` lists subdirectories recursively
- `S` sorts by file size (largest first)
- `t` sorts by modification time (newest first)

mkdir makes directories (`mkdir p`: with parents as needed, no error if existing).

mount mounts a filesystem.

mv* moves (renames) files:

- `f` does not prompt before overwriting,
- `i` prompts before overwriting.

pwd* prints name of current directory.

rm* removes files or directories:

- `f` never prompts,
- `i` always prompts,
- `r` removes directories and their contents.

rmdir removes (empty) directories.

split* splits a file into pieces:

- `b` puts „size" bytes per output file,
- `n` generates „chunks" output files.

tar stores and extracts files from a tape or disk archive.

- `c` creates a new archive,
- `x` extracts files from an archive,
- `t` lists the contents of an archive,
- `f` uses archive file or device
- `z` uses zip/gzip
- `j` bzip2 compression
- `k` does not replace existing files when extracting

tee duplicates pipe content.

touch changes file timestamps.

umask set file mode creation mask.

1.2 Processes

chroot changes root directory.

at schedules commands to be executed once, at a particular time in the future.

bg resumes suspended jobs in the background.

cron is a daemon to execute scheduled commands.

fg resumes suspended jobs in the foreground.

kill sends a `TERM` signal to a process.

killall kills processes by name.

nice changes process priority.

pgrep, **pkill** looks up or signals processes based on name and other attributes.

ps reports a snapshot of the current processes.

pstree displays a tree of processes.

time runs programs and summarizes system resource usage.

top displays linux processes.

1.3 User environment

clear clears the terminal screen.

env runs a program in a modified environment.

exit terminates the calling process.

finger is a user information lookup program.

history displays the history list with line numbers.

logname prints user's login name.

mesg displays (or does not display) messages from other users.

passwd changes user password:

- `d` deletes an account's password (makes it empty),
- `e` expires an account's password,
- `n` sets the minimum number of days between password changes,
- `w` sets the number of days of warning before a password change is required,
- `x` sets the maximum number of days a password remains valid.

su changes user ID or becomes superuser.

sudo executes a command as another user.

tput initializes a terminal or queries terminfo database.

uname prints system information:

- `a` all information, in the following order:
- `s` the kernel name,
- `n` the network node hostname,
- `r` the kernel release,
- `v` the kernel version,
- `m` the machine hardware name,
- `p` the processor type,
- `i` the hardware platform,
- `o` the operating system.

uptime tells how long the system has been running.

wall writes a message to all users, **write** sends a message to another user.

who shows who is logged on, **w** shows who is logged on and what they are doing, **whoami** prints effective userid.

1.4 Text processing

awk is a pattern scanning and processing language.

basename strips directory and suffix from filenames, **dirname** strip last component from file name.

comm compares two sorted files line by line.

csplit splits a file into sections determined by context lines.

join joins lines of two files on a common field. **paste** merges lines of files. **cut*** removes sections from each line of files:

- `d` uses „delim" instead of Tab for field delimeter,
- `f` selects only these fields.

diff compares files line by line.

fmt is a simple optimal text formatter, **fold** wraps each input line to fit in specified width.

head* outputs the first part of files:

- `c` the first „num" bytes,
- `n` the first „num" lines.

iconv converts text from one character encoding to another.

less is opposite of **more**, a file perusal filter for crt viewing.

nl* numbers lines of files:

- `s` adds „string" after line number,
- `w` uses „number" columns for line numbers.

printf formats and prints data.

sed is a stream editor for filtering and transforming text.

shuf* generates random permutations:

- `e` treats each „arg" as an input line,
- `i` treats each number .. through .. as an input line,
- `n` outputs at most „count" lines,
- `r` output lines can be repeated (with `-n`).

sort* sorts lines of text files:

- `g` compares general numerical values,
- `h` compares human readable numbers,
- `n` compares string numerical values,
- `r` reverses the results.

strings prints the strings of printable characters in files.

tail* outputs the last part of files:

- `c` the last „num" bytes,
- `f` outputs appended data as the file grows,
- `n` the last „num" lines.

tr* translates or deletes characters:

1. `tr abc xyz` changes `a` to `x`, . . .,
- `c` uses the complement of „set1",
- `d` deletes characters, does not translate,
- `s` replaces each sequence of a repeated character that is listed in the last specified „set" with a single occurrence of that character.

uniq* omits repeated lines:

- `c` prefixes lines by the number of occurences
- `d` only prints duplicate lines, one for each group
- `f` avoids comparing first fields
- `i` ignores differences in case
- `s` avoids comparing first characters
- `w` compares no more than n characters

vim a programmers text editor.

wc* prints newline, word and byte counts:

- `c` prints the byte counts,
- `l` prints the newline counts,
- `m` prints the character counts,
- `w` prints the word counts.

xargs builds and executes command lines from standard input.

yes outputs a string repeatedly until killed.

1.5 Shell builtins

alias . . . . . .

cd changes the shell working directory.

- `-` to the previous directory.

echo* displays a line of text:

- `e` enables interpretation of backslash escapes,
- `n` does not output the trailing newline.

test checks file types and compares values.

unset . . . . . .

wait waits for process to change state.

1.6 Networking

curl transfers a URL.

dig is a DNS lookup utility (domain information groper).

- `x` simplified reverse lookups.

host is a DNS lookup utility.

ifconfig configures a network interface.

inetd is a super-server daemon that provides Internet services.

netcat: arbitrary TCP and UDP connections and listens.

netstat prints network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.

nslookup queries Internet name servers interactively.

ping tests the reachability of a host on an IP network by sending ICMP ECHO_REQUEST:

- `c` stops after sending „count" packets,
- `n` numeric output only, avoids to lookup symbolic names for host addresses.

rdate sets the system's date from a remote host.

rlogin is an OpenSSH SSH client (remote login program)

route shows and manipulates the IP routing table.

ssh is an OpenSSH SSH client (remote login program).

- `D` (bind address)
- `p` (port)
- `X` (X11 forwarding)

traceroute is a computer network diagnostic tool for displaying the route (path) and measuring transit delays of packets across an Internet Protocol (IP) network.

wget is a non-interactive network downloader.

- `A, R` specifies lists of file suffixes or patterns (when wildcard characters appear) to accept or reject,
- `b` goes to background immediately after startup,
- `c` continues getting a partially-downloaded file,
- `m` turns on options suitable for mirroring: infinite recursion and time-stamping,
- `np` does not ever ascend to the parent directory when retrieving recursively,
- `U` identifies as „agent-string" to the HTTP server.
- `w` waits the specified number of seconds between the retrievals (see also `--random-wait`).

1.7 Searching

find searches for files in a directory hierarchy.

grep prints lines matching a pattern.

locate finds files by names.

whatis displays one-line manual page description.

whereis locates the binary, source, and manual page files for a command.

1.8 Documentation

apropos searches the manual page names and descriptions.

man is an interface to the online reference manuals.

1.9 Miscellaneous

bc is an arbitrary precision calculator language.

1. `echo 'obase=16;255' | bc` prints `FF`,
2. `echo 'ibase=2;obase=A;10' | bc` prints `2`,
3. `scale=10` (after `bc -l`) sets working precision.

dc is a reverse-polish desk calculator. One of the oldest Unix utilities, predating even the invention of the C programming language.

cal, **ncal** displays a calendar and the date of Easter.

- `e` displays date of Easter,
- `j` displays Julian days,
- `m` displays the specified month,
- `w` prints the numbers of the weeks,
- `y` displays a calendar for the specified year,
- `3` displays the previous, current and next month.

date prints or set the system date and time.

lp prints files.

od dumps files in octal.

sleep delays for a specified amount of time.

true, **false** does nothing, (un)successfully.

1.10 Other commands

base32 base64 chcon date dir expand factor groups hostid hostname id install link md5sum mkfifo mknod mktemp nohup nproc numfmt pathchk pr printenv readlink realpath runcon shred stat stdbuf stty sum sync timeout touch truncate tsort tty unexpand unlink users vdir

2 Keyboard shortcuts

1. `Ctrl+A` cursor to start of line
2. `Ctrl+C` halts current command
3. `Ctrl+D` log out of current session current command
4. `Ctrl+E` cursor to end of line
5. `Ctrl+L` clear the terminal
6. `Ctrl+U` erases whole line
7. `Ctrl+W` erase one word in current line
8. `Ctrl+Z` stops current command

3 Programming in Bash

The shebang (`#!`) at the head of a script indicates an interpreter for execution, as in `#!/bin/bash`. Lines starting with a `#` (with the exception of shebang) are comments and thus won't be executed.

There are always three default files open: *stdin* (the keyboard, file descriptor 0), *stdout* (the screen, file descriptor 1) and *stderr* (error messages output to the screen, file descriptor 2).

These **streams** can be **redirected**: `cmd > file` redirects to a file (overwrites), `cmd >> file` appends instead, `m>n` (or `m>&n`) redirects a file descriptor to a file (or another file descriptor), `&>file` redirects both stdout and stderr to a file; `:> file` truncates file to zero length and `|` (pipe) serves as a command chaining tool.

Variables are case sensitive and capitalized by default. Variables can also contain digits and underscores, but a name starting with a digit is not allowed. Example: `var=value; echo $vars` prints `values`. Special variables:

1. `$0`, `$1`, ...: name of the script itself, the first, second, etc. argument.
2. `$*` and `$@` denote all the positional parameters.
3. `$#`: the number of positional parameters
4. `$?`: exit status of the most recently executed command.
5. `$$`: the process ID of the shell.
6. `$!`: the process ID of the most recently executed command.

Popular classes:

1. `[:alnum:]`, letters and digits,
2. `[:alpha:]`, alphabetic characters
3. `[:digit:]`, digits,
4. `[:lower:]`,
5. `[:punct:]`, punctuation characters,
6. `[:upper:]`.

3.1 Regular expressions