Contents

ls . . . 1
mkdir . . . 2
cp . . . 2
mv . . . 3
rm . . . 3
ln . . . 3
cat . . . 4
pr . . . 4
fmt . . . 5
lp . . . 5
wc . . . 5
diff . . . 5
sort . . . 6
cut . . . 7
paste . . . 7
tr . . . 7
tee . . . 8
head . . . 8
tail . . . 9
perl . . . 9


```
for i in {1..5}; do
  ...
done
while : ; do echo "hi"; done
python -c "print('hi')"
python -e "print 'hi';"
cat < file.txt # y?
```

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

ls

```
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).

  -a, --all                  do not ignore entries starting with .
  -A, --almost-all           do not list implied . and ..
  -C                         list entries by columns
      --color[=WHEN]         colorize the output; WHEN can be 'always' (default
                               if omitted), 'auto', or 'never'; more info below
  -d, --directory            list directories themselves, not their contents
  -D, --dired                generate output designed for Emacs' dired mode
  -f                         do not sort, enable -aU, disable -ls --color
  -F, --classify             append indicator (one of */=>@|) to entries
      --file-type            likewise, except do not append '*'
  -g                         like -l, but do not list owner
  -h, --human-readable       with -l and -s, print sizes like 1K 234M 2G etc.
      --si                   likewise, but use powers of 1000 not 1024
  -i, --inode                print the index number of each file
  -I, --ignore=PATTERN       do not list implied entries matching shell PATTERN
  -l                         use a long listing format
```

```
  -L, --dereference          when showing file information for a symbolic
                               link, show information for the file the link
                               references rather than for the link itself
  -m                         fill width with a comma separated list of entries
  -o                         like -l, but do not list group information
  -p, --indicator-style=slash
                             append / indicator to directories
  -q, --hide-control-chars   print ? instead of nongraphic characters
      --show-control-chars   show nongraphic characters as-is (the default,
                               unless program is 'ls' and output is a terminal)
  -Q, --quote-name           enclose entry names in double quotes
      --quoting-style=WORD   use quoting style WORD for entry names:
                               literal, locale, shell, shell-always,
                               shell-escape, shell-escape-always, c, escape
                               (overrides QUOTING_STYLE environment variable)
  -r, --reverse              reverse order while sorting
  -R, --recursive            list subdirectories recursively
  -s, --size                 print the allocated size of each file, in blocks
  -S                         sort by file size, largest first
      --sort=WORD            sort by WORD instead of name: none (-U), size (-S),
                               time (-t), version (-v), extension (-X)
      --time=WORD            with -l, show time as WORD instead of default
                               modification time: atime or access or use (-u);
                               ctime or status (-c); also use specified time
                               as sort key if --sort=time (newest first)
      --time-style=TIME_STYLE  time/date format with -l; see TIME_STYLE below
  -t                         sort by modification time, newest first
  -U                         do not sort; list entries in directory order
  -v                         natural sort of (version) numbers within text
  -w, --width=COLS           set output width to COLS.  0 means no limit
  -X                         sort alphabetically by entry extension
  -1                         list one file per line.  Avoid '\n' with -q or -b
```

mkdir

```
Usage: mkdir [OPTION]... DIRECTORY...
Create the DIRECTORY(ies), if they do not already exist.

  -m, --mode=MODE   set file mode (as in chmod), not a=rwx - umask
  -p, --parents     no error if existing, make parent directories as needed
  -v, --verbose     print a message for each created directory
```

cp

```
Usage: cp [OPTION]... [-T] SOURCE DEST
  or:  cp [OPTION]... SOURCE... DIRECTORY
  or:  cp [OPTION]... -t DIRECTORY SOURCE...
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

  -a, --archive                same as -dR --preserve=all
      --attributes-only        don't copy the file data, just the attributes
      --backup[=CONTROL]       make a backup of each existing destination file
  -b                           like --backup but does not accept an argument
      --copy-contents          copy contents of special files when recursive
```

```
  -d                            same as --no-dereference --preserve=links
  -f, --force                   if an existing destination file cannot be
                                  opened, remove it and try again (this option
                                  is ignored when the -n option is also used)
  -i, --interactive             prompt before overwrite (overrides a previous -n
                                   option)
  -H                            follow command-line symbolic links in SOURCE
  -l, --link                    hard link files instead of copying
  -L, --dereference             always follow symbolic links in SOURCE
  -n, --no-clobber              do not overwrite an existing file (overrides
                                  a previous -i option)
  -R, -r, --recursive           copy directories recursively
      --reflink[=WHEN]          control clone/CoW copies. See below
      --remove-destination      remove each existing destination file before
                                  attempting to open it (contrast with --force)
      --sparse=WHEN             control creation of sparse files. See below
      --strip-trailing-slashes  remove any trailing slashes from each SOURCE
                                  argument
  -s, --symbolic-link           make symbolic links instead of copying
  -S, --suffix=SUFFIX           override the usual backup suffix
  -t, --target-directory=DIRECTORY  copy all SOURCE arguments into DIRECTORY
  -T, --no-target-directory     treat DEST as a normal file
  -v, --verbose                 explain what is being done
  -x, --one-file-system         stay on this file system
```

mv

```
Usage: mv [OPTION]... [-T] SOURCE DEST
  or:  mv [OPTION]... SOURCE... DIRECTORY
  or:  mv [OPTION]... -t DIRECTORY SOURCE...
Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.

      --backup[=CONTROL]       make a backup of each existing destination file
  -b                           like --backup but does not accept an argument
  -f, --force                  do not prompt before overwriting
  -i, --interactive            prompt before overwrite
  -n, --no-clobber             do not overwrite an existing file
If you specify more than one of -i, -f, -n, only the final one takes effect.
      --strip-trailing-slashes  remove any trailing slashes from each SOURCE
                                  argument
  -S, --suffix=SUFFIX          override the usual backup suffix
  -t, --target-directory=DIRECTORY  move all SOURCE arguments into DIRECTORY
  -T, --no-target-directory    treat DEST as a normal file
  -u, --update                 move only when the SOURCE file is newer
                                 than the destination file or when the
                                 destination file is missing
  -v, --verbose                explain what is being done
```

rm

```
Usage: rm [OPTION]... [FILE]...
Remove (unlink) the FILE(s).

  -f, --force           ignore nonexistent files and arguments, never prompt
```

```
  -i                    prompt before every removal
  -I                    prompt once before removing more than three files, or
                          when removing recursively; less intrusive than -i,
                          while still giving protection against most mistakes
      --no-preserve-root  do not treat '/' specially
      --preserve-root[=all]  do not remove '/' (default);
                              with 'all', reject any command line argument
                              on a separate device from its parent
  -r, -R, --recursive   remove directories and their contents recursively
  -d, --dir             remove empty directories
  -v, --verbose         explain what is being done
```

ln

```
Usage: ln [OPTION]... [-T] TARGET LINK_NAME
  or:  ln [OPTION]... TARGET
  or:  ln [OPTION]... TARGET... DIRECTORY
  or:  ln [OPTION]... -t DIRECTORY TARGET...
In the 1st form, create a link to TARGET with the name LINK_NAME.
In the 2nd form, create a link to TARGET in the current directory.
In the 3rd and 4th forms, create links to each TARGET in DIRECTORY.
Create hard links by default, symbolic links with --symbolic.
By default, each destination (name of new link) should not already exist.
When creating hard links, each TARGET must exist.  Symbolic links
can hold arbitrary text; if later resolved, a relative link is
interpreted in relation to its parent directory.

      --backup[=CONTROL]      make a backup of each existing destination file
  -b                          like --backup but does not accept an argument
  -d, -F, --directory         allow the superuser to attempt to hard link
                                directories (note: will probably fail due to
                                system restrictions, even for the superuser)
  -f, --force                 remove existing destination files
  -i, --interactive           prompt whether to remove destinations
  -L, --logical               dereference TARGETs that are symbolic links
  -n, --no-dereference        treat LINK_NAME as a normal file if
                                it is a symbolic link to a directory
  -P, --physical              make hard links directly to symbolic links
  -r, --relative              create symbolic links relative to link location
  -s, --symbolic              make symbolic links instead of hard links
  -S, --suffix=SUFFIX         override the usual backup suffix
  -v, --verbose               print name of each linked file
```

cat

```
Usage: cat [OPTION]... [FILE]...
Concatenate FILE(s) to standard output.

  -b, --number-nonblank    number nonempty output lines, overrides -n
  -e                       equivalent to -vE
  -E, --show-ends          display $ at end of each line
  -n, --number             number all output lines
  -s, --squeeze-blank      suppress repeated empty output lines
  -t                       equivalent to -vT
```

```
  -T, --show-tabs          display TAB characters as ^I
  -v, --show-nonprinting   use ^ and M- notation, except for LFD and TAB
```

pr

```
Usage: pr [OPTION]... [FILE]...
Paginate or columnate FILE(s) for printing.

  +FIRST_PAGE[:LAST_PAGE], --pages=FIRST_PAGE[:LAST_PAGE]
                   begin [stop] printing with page FIRST_[LAST_]PAGE
  -COLUMN, --columns=COLUMN
                   output COLUMN columns and print columns down,
                   unless -a is used. Balance number of lines in the
                   columns on each page
  -a, --across     print columns across rather than down, used together
                   with -COLUMN
  -d               double space the output
  -h, --header=HEADER
                   use a centered HEADER instead of filename in page header,
                   -h "" prints a blank line, don't use -h""
  -l, --length=PAGE_LENGTH
                   set the page length to PAGE_LENGTH (66) lines
                   (default number of lines of text 56, and with -F 63).
                   implies -t if PAGE_LENGTH <= 10
  -m, --merge      print all files in parallel, one in each column,
                   truncate lines, but join lines of full length with -J
  -n[SEP[DIGITS]], --number-lines[=SEP[DIGITS]]
                   number lines, use DIGITS (5) digits, then SEP (TAB),
                   default counting starts with 1st line of input file
```

fmt

```
Usage: fmt [-WIDTH] [OPTION]... [FILE]...
Reformat each paragraph in the FILE(s), writing to standard output.
The option -WIDTH is an abbreviated form of --width=DIGITS.

  -s, --split-only          split long lines, but do not refill
  -u, --uniform-spacing     one space between words, two after sentences
  -w, --width=WIDTH         maximum line width (default of 75 columns)
```

lp

```
Usage: lp [options] [--] [file(s)]
Options:
-d destination          Specify the destination
-o nofilebreak          don't jump on new page after finishing a file
-o length=n             duh
-o width=n              duh
-P page-list            Specify a list of pages to print
-w                      Write a message on exit - not found on Manjaro
```

wc

```
Usage: wc [OPTION]... [FILE]...
Print newline, word, and byte counts for each FILE, and a total line if
```

```
more than one FILE is specified.  A word is a non-zero-length sequence of
characters delimited by white space.

The options below may be used to select which counts are printed, always in
the following order: newline, word, character, byte, maximum line length.
  -c, --bytes            print the byte counts
  -m, --chars            print the character counts
  -l, --lines            print the newline counts
  -w, --words            print the word counts
```

diff

```
Usage: diff [OPTION]... FILES
Compare FILES line by line.

  -q, --brief                   report only when files differ
  -c, -C NUM, --context[=NUM]   output NUM (default 3) lines of copied context
  -r, --recursive                 recursively compare any subdirectories found
  -i, --ignore-case               ignore case differences in file contents
```

sort

```
Usage: sort [OPTION]... [FILE]...
  or:  sort [OPTION]... --files0-from=F
Write sorted concatenation of all FILE(s) to standard output.

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

Ordering options:

  -b, --ignore-leading-blanks  ignore leading blanks
  -d, --dictionary-order      consider only blanks and alphanumeric characters
  -f, --ignore-case           fold lower case to upper case characters
  -g, --general-numeric-sort  compare according to general numerical value
  -i, --ignore-nonprinting    consider only printable characters
  -M, --month-sort            compare (unknown) < 'JAN' < ... < 'DEC'
  -h, --human-numeric-sort    compare human readable numbers (e.g., 2K 1G)
  -n, --numeric-sort          compare according to string numerical value
  -R, --random-sort           shuffle, but group identical keys.  See shuf(1)
      --random-source=FILE    get random bytes from FILE
  -r, --reverse               reverse the result of comparisons
      --sort=WORD             sort according to WORD:
                                general-numeric -g, human-numeric -h, month -M,
                                numeric -n, random -R, version -V
  -V, --version-sort          natural sort of (version) numbers within text

Other options:

      --batch-size=NMERGE   merge at most NMERGE inputs at once;
                            for more use temp files
  -c, --check, --check=diagnose-first  check for sorted input; do not sort
  -C, --check=quiet, --check=silent  like -c, but do not report first bad line
      --compress-program=PROG  compress temporaries with PROG;
                               decompress them with PROG -d
```

```
      --debug               annotate the part of the line used to sort,
                              and warn about questionable usage to stderr
      --files0-from=F       read input from the files specified by
                            NUL-terminated names in file F;
                            If F is - then read names from standard input
  -k, --key=KEYDEF          sort via a key; KEYDEF gives location and type
  -m, --merge               merge already sorted files; do not sort
  -o, --output=FILE         write result to FILE instead of standard output
  -s, --stable              stabilize sort by disabling last-resort comparison
  -S, --buffer-size=SIZE    use SIZE for main memory buffer
  -t, --field-separator=SEP  use SEP instead of non-blank to blank transition
  -T, --temporary-directory=DIR  use DIR for temporaries, not $TMPDIR or /tmp;
                              multiple options specify multiple directories
      --parallel=N          change the number of sorts run concurrently to N
  -u, --unique              with -c, check for strict ordering;
                              without -c, output only the first of an equal run
  -z, --zero-terminated     line delimiter is NUL, not newline
```

cut

```
Usage: cut OPTION... [FILE]...
Print selected parts of lines from each FILE to standard output.

  -b, --bytes=LIST        select only these bytes
  -c, --characters=LIST   select only these characters
  -d, --delimiter=DELIM   use DELIM instead of TAB for field delimiter
  -f, --fields=LIST       select only these fields;  also print any line
                            that contains no delimiter character, unless
                            the -s option is specified
  -n                      (ignored)
      --complement        complement the set of selected bytes, characters
                            or fields
  -s, --only-delimited    do not print lines not containing delimiters
      --output-delimiter=STRING  use STRING as the output delimiter
                            the default is to use the input delimiter
  -z, --zero-terminated    line delimiter is NUL, not newline
      --help     display this help and exit
      --version  output version information and exit

Use one, and only one of -b, -c or -f.  Each LIST is made up of one
range, or many ranges separated by commas.  Selected input is written
in the same order that it is read, and is written exactly once.
Each range is one of:

  N     N'th byte, character or field, counted from 1
  N-    from N'th byte, character or field, to end of line
  N-M   from N'th to M'th (included) byte, character or field
  -M    from first to M'th (included) byte, character or field
cut f -b 1-10
```

paste

```
Usage: paste [OPTION]... [FILE]...
Write lines consisting of the sequentially corresponding lines from
```

```
each FILE, separated by TABs, to standard output.

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

Mandatory arguments to long options are mandatory for short options too.
  -d, --delimiters=LIST   reuse characters from LIST instead of TABs
  -s, --serial            paste one file at a time instead of in parallel
  -z, --zero-terminated    line delimiter is NUL, not newline
```

tr

```
Usage: tr [OPTION]... SET1 [SET2]
Translate, squeeze, and/or delete characters from standard input,
writing to standard output.

  -c, -C, --complement    use the complement of SET1
  -d, --delete            delete characters in SET1, do not translate
  -s, --squeeze-repeats   replace each sequence of a repeated character
                            that is listed in the last specified SET,
                            with a single occurrence of that character
  -t, --truncate-set1     first truncate SET1 to length of SET2
      --help     display this help and exit
      --version  output version information and exit

SETs are specified as strings of characters.  Most represent themselves.
Interpreted sequences are:

  \NNN            character with octal value NNN (1 to 3 octal digits)
  \\              backslash
  \a              audible BEL
  \b              backspace
  \f              form feed
  \n              new line
  \r              return
  \t              horizontal tab
  \v              vertical tab
  CHAR1-CHAR2     all characters from CHAR1 to CHAR2 in ascending order
  [CHAR*]         in SET2, copies of CHAR until length of SET1
  [CHAR*REPEAT]   REPEAT copies of CHAR, REPEAT octal if starting with 0
  [:alnum:]       all letters and digits
  [:alpha:]       all letters
  [:blank:]       all horizontal whitespace
  [:cntrl:]       all control characters
  [:digit:]       all digits
  [:graph:]       all printable characters, not including space
  [:lower:]       all lower case letters
  [:print:]       all printable characters, including space
  [:punct:]       all punctuation characters
  [:space:]       all horizontal or vertical whitespace
  [:upper:]       all upper case letters
  [:xdigit:]      all hexadecimal digits
  [=CHAR=]        all characters which are equivalent to CHAR
```

tee

```
Usage: tee [OPTION]... [FILE]...
Copy standard input to each FILE, and also to standard output.

  -a, --append              append to the given FILEs, do not overwrite
  -i, --ignore-interrupts   ignore interrupt signals
  -p                        diagnose errors writing to non pipes
      --output-error[=MODE]   set behavior on write error.  See MODE below
```

head

```
Usage: head [OPTION]... [FILE]...
Print the 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.

Mandatory arguments to long options are mandatory for short options too.
  -c, --bytes=[-]NUM       print the first NUM bytes of each file;
                             with the leading '-', print all but the last
                             NUM bytes of each file
  -n, --lines=[-]NUM       print the first NUM lines instead of the first 10;
                             with the leading '-', print all but the last
                             NUM lines of each file
  -q, --quiet, --silent    never print headers giving file names
  -v, --verbose            always print headers giving file names
  -z, --zero-terminated    line delimiter is NUL, not newline
```

tail

```
Usage: tail [OPTION]... [FILE]...
Print the 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.

  -c, --bytes=[+]NUM       output the last NUM bytes; or use -c +NUM to
                             output starting with byte NUM of each file
  -f, --follow[={name|descriptor}]
                           output appended data as the file grows;
                             an absent option argument means 'descriptor'
  -F                       same as --follow=name --retry
  -n, --lines=[+]NUM       output the last NUM lines, instead of the last 10;
                             or use -n +NUM to output starting with line NUM
      --max-unchanged-stats=N
                           with --follow=name, reopen a FILE which has not
                             changed size after N (default 5) iterations
                             to see if it has been unlinked or renamed
                             (this is the usual case of rotated log files);
                             with inotify, this option is rarely useful
      --pid=PID            with -f, terminate after process ID, PID dies
  -q, --quiet, --silent    never output headers giving file names
      --retry              keep trying to open a file if it is inaccessible
  -s, --sleep-interval=N   with -f, sleep for approximately N seconds
```

```
                              (default 1.0) between iterations;
                              with inotify and --pid=P, check process P at
                              least once every N seconds
  -v, --verbose            always output headers giving file names
  -z, --zero-terminated    line delimiter is NUL, not newline
```

perl

```
Usage: perl [switches] [--] [programfile] [arguments]
  -0[octal]         specify record separator (\0, if no argument)
  -a                autosplit mode with -n or -p (splits $_ into @F)
  -C[number/list]   enables the listed Unicode features
  -c                check syntax only (runs BEGIN and CHECK blocks)
  -d[:debugger]     run program under debugger
  -D[number/list]   set debugging flags (argument is a bit mask or alphabets)
  -e program        one line of program (several -e's allowed, omit programfile)
  -E program        like -e, but enables all optional features
  -f                don't do $sitelib/sitecustomize.pl at startup
  -F/pattern/       split() pattern for -a switch (//'s are optional)
  -i[extension]     edit <> files in place (makes backup if extension supplied)
  -Idirectory       specify @INC/#include directory (several -I's allowed)
  -l[octal]         enable line ending processing, specifies line terminator
  -[mM][-]module    execute "use/no module..." before executing program
  -n                assume "while (<>) { ... }" loop around program
  -p                assume loop like -n but print line also, like sed
  -s                enable rudimentary parsing for switches after programfile
  -S                look for programfile using PATH environment variable
  -t                enable tainting warnings
  -T                enable tainting checks
  -u                dump core after parsing program
  -U                allow unsafe operations
  -v                print version, patchlevel and license
  -V[:variable]     print configuration summary (or a single Config.pm variable)
  -w                enable many useful warnings
  -W                enable all warnings
  -x[directory]     ignore text before #!perl line (optionally cd to directory)
  -X                disable all warnings

Run 'perldoc perl' for more help with Perl.
```