

# { UNIX CHEAT SHEET }

MARCIN SZEWCZYK

NOTES 2007 · VERSION 2026

## USER AND ENVIRONMENT

### - Account information:

`passwd` – (password) change the user’s login password  
`quota -v` – (quota – limit) display disk space limits assigned to the account  
`id` – (user identity) display information about the current user’s name and group  
`whoami` – display the name of the current user  
`who / w` – display logged-in users (and their activity)

### - Session and help:

`man polecenie` – (manual) display the manual page for a command  
`history` – command history  
`logout` – log out the user

### - Current directory:

`pwd` – (print working directory) display the name of the current directory

### - Shell configuration:

`~/.zshrc` – file with instructions executed at login (note: errors may prevent access to the account)  
`alias polecenie='nowe polecenie'` – assign a new definition to a command (recommended to place in `.zshrc`)  
`alias ls='ls -F'` – mark directories with `/` and executable files with `*`  
`alias pico='pico -w -z' --w` (disable word wrap), `-z` (enable suspend)  
`alias cp='cp -i'` – confirm file overwrite  
`alias mv='mv -i'` – confirm file overwrite  
`umask` – display default permissions for new files/directories  
`umask ijk` – set new permission value (analogous to `chmod`)

### - Running programs and PATH:

Programs not in the `PATH` variable require the full path.

`printenv PATH / echo $PATH` – display the `PATH` variable  
`which polecenie` – path to the executable program (from `PATH`)

### - Utility programs:

`nano` – terminal text editor (e.g. `^O` save, `^X` exit, `^W` search)  
`vim` – advanced text editor (normal/insert mode, `:w` save, `:q` exit)  
`g++` – C++ compiler (e.g. `g++ plik.cpp -o program`)  
`make` – automate project builds based on a Makefile  
`git` – version control system (e.g. `git clone`, `git commit`, `git push`)  
`ssh user@host` – remote login via SSH  
`tar -cvf / -xvf` – create and extract archives

## FILES AND DIRECTORIES

### - Navigation:

`ls` – (list) display directory contents (`ls -al` – all information)  
`cd` – (change directory) go to the home directory  
`cd nazwa` – go to the specified directory  
`.` – current directory  
`..` – parent directory  
`/` – root directory  
`~` – home directory  
`~user` – specified user’s home directory

### - File and directory operations:

`mkdir nazwa` – create a directory  
`touch plik` – create an empty file (or update modification time)  
`cp nazwa1 nazwa2 - (copy) copy file/directory nazwa1 to nazwa2`  
`mv nazwa1 nazwa2 - (move) move/rename nazwa1 to nazwa2`  
`ln nazwa1 nazwa2` – create a hard link  
`ln -s cel link` – create a symbolic link  
`rm nazwa` – (remove) delete a file  
`rmdir nazwa` – remove an empty directory  
`rm -r nazwa` – remove a directory with its contents (warning: no confirmation)

### - Permissions:

`chmod ijk nazwa` – (change the permissions mode) change permissions (execute=1, write=2, read=4) for owner/group/others; also: `chmod -R`

### - Disk and search:

`du -h` – (disk usage) directory size  
`df -h` – filesystem usage  
`find ścieżka -name 'wzorzec'` – search for files/directories

## STREAMS, PIPES AND PROCESSES

### - Streams and pipes (I/O):

`polecenie > plik` – write command output to a file  
`polecenie » plik` – append output to the end of a file  
`polecenie < plik` – read input from a file instead of the keyboard  
`polecenie1 | polecenie2` – pass output of `polecenie1` to `polecenie2`  
`tee plik` – write output to a file and simultaneously to the screen  
`xargs` – build command arguments from a pipe

### - Stream tools:

`grep tekst plik` – search for text in a file  
`cat plik` – display file contents  
`less plik` – view file page by page  
`head plik` – beginning of file  
`tail plik` – end of file  
`tail -f plik` – follow appended lines (e.g. logs)  
`wc plik` – (word count) number of lines, words and characters in a file  
`sort plik` – sort file lines  
`uniq plik` – remove adjacent duplicate lines (often used after `sort`)

### - Processes:

`ps / ps aux` – process list  
`top` – dynamic process viewer  
`kill PID` – terminate process with given PID