

A Collection of Useful Shell Commands

Disclaimer

Many of the following commands can be used on all Unix platforms. Some of them only work on MacOS, though.

Basic navigation and file/directory management

| Command | Meaning |
|-----------------------------------|---|
| cd | change directory |
| ls | list [-l] [-a] |
| pwd | present working directory |
| cp | copy |
| mv | move / rename |
| cat | show file content |
| less | show file content in separate window |
| grep | grep content from file, etc. (e.g. 'cat grep -i #ignores cases') |
| awk | advanced tool to grap contents |
| echo "string" [> file] | echo string to terminal [file] |
| | overwrites existing content! |
| touch [file1] ... | create one or multiple files very fast |
| mkdir | make directory |
| rmdir | remove directory |
| chown [username:usergroup] [file] | change ownership of particular file (e.g. 'chown root:root [file]') |
| chmod | change file permissions [+x] makes file executable [+w] makes file writable [+r] makes file readable |
| rm | remove [-rf] |
| adduser [new_user_name] | create new user and group |
| nano / vim / emacs | text editor |
| apt-get install / brew install | package manager |
| | pipe from standard output to command |
| > | write left into right (overwrites!) |
| > > | append left to right (does not overwrite!) |
| find [-name] [folder_name] | find a name |
| locate [name] | list all paths that contain [name] |
| updatedb | updates file system database (better run before every search) |
| history | shows bash history from ~/.bash_history |
| clear | clears screen be shifting it up |
| uname | [-a] shows host system |
| ps | shows running processes and pids |
| shutdown | shuts down the system after a short pause [-h now] or immediately |
| poweroff | powers off immediately |
| reboot | reboots the system immediately |
| ifconfig | shows network interfaces |
| ping [url] | simple send/receive of packets to [url] |
| lpr lp | print a document to the default printer (or specify whatever printer you are connected to!) |

Misc commands

Go

```
env GOOS=windows GOARCH=amd64 CGO_ENABLED=1 go build -v FILENAME.go # cross-compile for windows
go test -coverprofile cover.out <package_name> # save test coverage profile to cover.out
go tool cover -html=cover.out -o cover.html # convert cover.out to html
```

Heroku

```
echo "web: <app_directory_name>" > Procfile # create a Procfile
```

```
go get github.com/tools/godep # install godep if not already installed
godep save # make godep save dependencies in ./vendor with json file in Godep/Godep.json
heroku create # create an application with remote origin in Heroku's github
git push heroku master # deploy application; make sure to get $PORT from env in app
heroku open # open the deployed app
heroku logs --tail # show latest app logs
```

Compiler instructions

```
g++ -std=c++17 -g main.cpp -o main
```

ffmpeg - list all devices

```
ffmpeg -f avfoundation -list_devices true -i ''
```

ffmpeg - record from a device (change device number with -i flag in bash script)

```
tmnlutils ffmpeg # not yet implemented
sh ~/Documents/CheatSheets/Terminal_Applications/ffmpeg_record.sh
# this is the actual script that gets called by tmnlutils
ffplay <input_file>
```

pandoc

```
pandoc -s -o DOC.pdf DOC.md #compile a markdown file to pdf

#download webpages for printing/offline reading
wget <url> -o logfile #recursive download is possible, too
pandoc DOC.html --pdf-engine=xelatex -o DOC.pdf #could not install the default engine on MacOS
```

imagemagick

```
convert <INPUT> -filter Lanczos -sampling-factor 1x1 -quality 90 -resize 150 <OUTPUT>
#resizes a jpg to a lower resolution image with a decent result
convert <INPUT> -crop 1200x300+0+250 <OUTPUT>
#crop an image to size 1200x300 with cropping window starting at x=0, y=250
```

Environment variables and how to get help

```
env #list all environment variables (similarly, you can do that for go!) OR just:
echo $VAR_NAME #e.g. $HOME, $GOPATH, ...

man <command>
<command> --help
<command> -h
```