

Working with Command Line Interface

Directories and Files

Types

There are two general types of directories/files:

- Hidden
 - On Unix-like -> start with a dot character (.)
 - Will not be displayed on usual listings (such as plain ls)
 - Often related to configurations or information
- Normal/Regular
 - Shows up on usual listings (such as plain ls)

To display regular directories/files you can use:

```
ls
```

To also display hidden directories/files:

```
ls -a
```

To give the output as a structured list:

```
ls -la
```

To navigate through directories you can use:

```
cd /path/to/dir
```

To directly go to your home directory:

```
cd
```

Creating & Removing directories/files

To create a new file you can use:

```
touch <filename>
```

To create a new directory you can use:

```
mkdir <dirname>
```

To remove a file you can use:

```
rm <filename>
```

To remove a directory recursively you can use:

```
rm -r <dirname>
```

To remove just a directory:

```
rmdir <dirname>
```

Editing files

Common terminal editors are:

- nano
- vi/vim/neovim

By using editors you can skip the touch step when creating a file, and pick a filename when you are saving a file or when you are entering the command to edit a file:

```
nano <filename>
vim <filename>
```

You can also use non terminal editors from the terminal by using the same pattern:

Visual Studio Code (usually):

```
code <filename>
```

Sublime Text:

```
subl <filename>
```

Nano:

- To save a file in nano you can CTRL^X -> Y -> Enter/type file name + Enter
- Nano has a short list of keybinds at the bottom

Vim family:

- To save a file you need to be in “normal mode” (usually reachable by ESC), and then type :w (write) or :wq (write and quit)
- To start typing you have to enter “insert mode”, this can be done by pressing i (insert) or a (append)
- Typing :help in “normal mode” opens up a help page

Terminal file editors are very commonly used when SSHing into remote machines.

Viewing files

To output the entire content of a file at once:

```
cat <filename>
```

To view a file and be able to navigate through it with arrow keys:

```
less <filename>
```

You can always view the file in your favourite terminal editor:

`<terminal-editor> <filename>`

The Shell

Popular shells

A shell is a user interface that gives you access to an operating system's services. You often interact with the shell when you are using the command line interface in your terminal program.

The most common shells are:

- Bash (standard on GNU/Linux)
- Zsh (Kind of feels like a superset of bash)
- Fish
- Sh

Shells can be configured with specific dotfiles (.). For bash it is .bashrc, in which you can configure your bash shell.

Environment variables

Environment variables are dynamic variables that are used by a shell. They can be defined in the shell configuration file (e.g. .bashrc), so that they persist through sessions.

You can add an environment variable in your shell configuration file by adding:

```
export ENV_NAME=ENV_VALUE
```

You can also use this in the shell directly, but if you do that it will only be valid for the current shell session.

Aliases

Aliases are a way to create shortcuts or rename a command to something simpler.

You can add an alias to your configuration file by adding:

```
alias shortcut=action
```

You can also use this in the shell directly, but if you do that it will only be valid for the current shell session.

To check where the program you want to use is located OR what alias it actually is using you can use:

```
which <command>
```