

# Working with the command line

## Introduction

We trust that all of you have worked with computers before in some way. Programs that you are already familiar with provide you with a nice and shiny graphical user interface (GUI) that makes the use of these programs pretty easy. Probably, one of the most widely known examples is the text editor Microsoft Word. As one part of our course, we will be working with [Linux](#) as the operating system and introduce you into the working with the [shell](#)<sup>1)</sup>. The shell facilitates a text and command line based control of the computer. Specifically we will use the [bash](#) (Fig. 1), as many tools we will be using can be called only from the command line. This is, for an average user, the most direct way of interacting with the computer. We assume that the use of the shell is a bit cryptic at the beginning. However once you got used to it you can appreciate the 'power of the shell'. But let's get started.

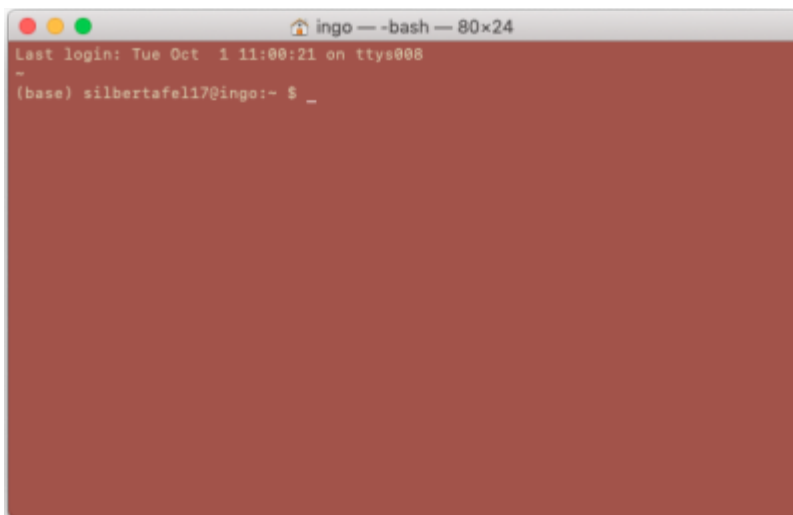


Figure 1: The *bash* represents a terminal with which you can directly interact with the computer. You can use the bash to read and write files; generate, copy, move or delete directories; as well as executing or stopping programs. There are many ways to customize the appearance of the bash. An explanation in German can be found [here](#). The default appearance of the command line prompt is something along the following lines: username@computername:~\$

## The computer environment

It will be necessary to get accustomed to the way how data and directories are organized under Linux. You will be working in a network environment with a central server architecture, local desktop computers like the one you are sitting on, and a computer cluster. Files and directories are organized in a hierarchical structure, referred to as the directory tree (Fig. 2). The computers in our work group are set up such that most parts of the directory tree, and in particular the home directories of the individual users are shared among all computers and are provided by a central file server. Thus, your data is almost never stored physically on the computer you are working on. This has the advantage that you can access your data from basically every computer in the network. Moreover, it allows to easily share information and data among people working on the same project.

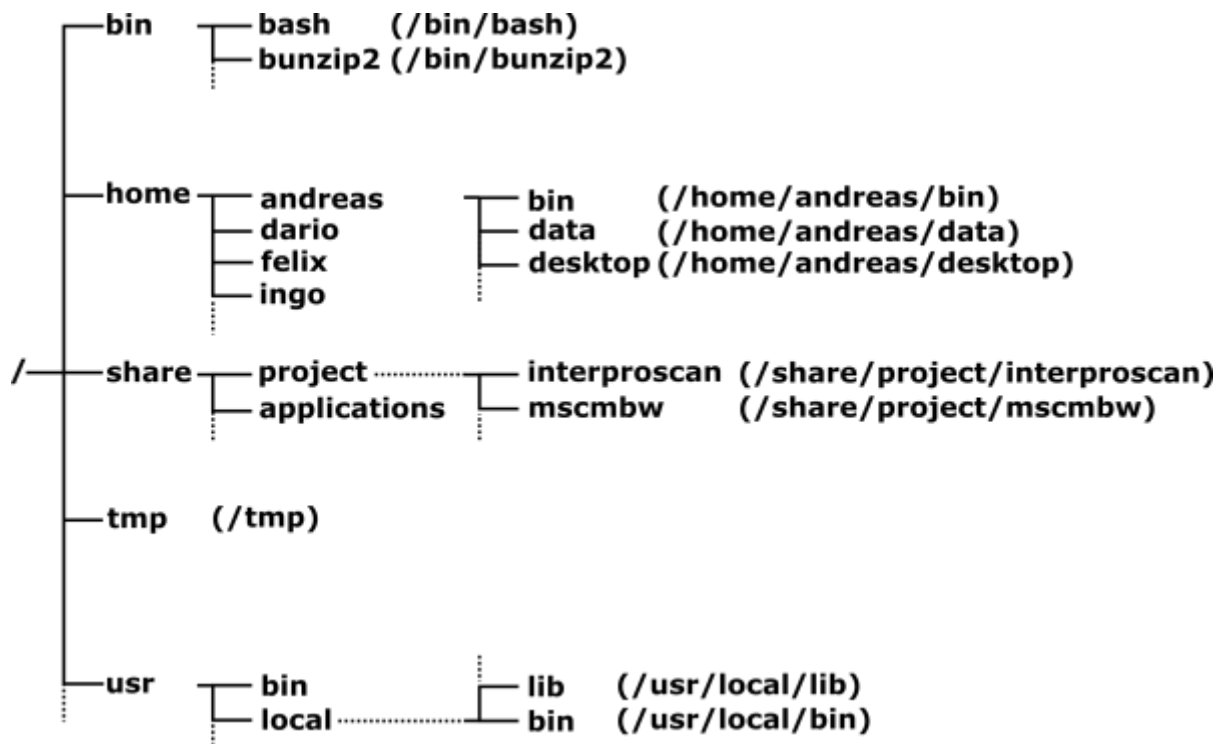



Figure 2: Section of the directory tree in the AK Applied Bioinformatics. The root of the tree is on the left side and is represented by the '/'. The absolute [paths](#) leading to the individual terminal files and directories are given in parentheses.

- A directory is just like a basket that can contain zero to many things. This includes subdirectories and files.
- A file is like a document. It contains information such as text or program code.
- If a directory contains sub-directories, then a tree-like structure emerges.
- Each file and each directory can be addressed by a unique path.
- Absolute paths start at the  root of the directory tree. Thus, an absolute path **always** starts with a '/'.
  - Relative paths start at the current working directory. Thus, they never start with a '/'.  - Directories in a path are separated by a '/'.  - Directories in a path are separated by a '.'.

## Tips and conventions

- File types are typically indicated by appending a '.' and an informative suffix, e.g. 'myinfo.txt'.
- Directory names do not contain any special characters. You will make your life substantially easier when you avoid characters such as '!', '?', '\$' and the like.
- On linux-based systems it is advisable to **not use white spaces** in file or directory names. Use underscores ('\_') instead.

## Code documentation

Sometimes we will provide you with short pieces of code that you can copy and paste into your terminal. Commands like that will appear in grey boxes like this:

```
# this is only a comment
$ mkdir <dirname>
```

- Lines starting with a hashtag **#** are comments and are not part of the code
- The **\$** sign represents the prompt in your shell (see the example above). Please **do not** copy it, when trying to copy-past commands from the DokuWiki to the command line
- Words in-between **<these signs>** are placeholders and need to be exchanged for something when typing the command.

---

## Task List

Once you know how to [open a BASH<sup>2\)</sup>](#) on the system you are using, it is time to learn how to use it. We have compiled a selection of resources and exercises for you after which you will be comfortably working with the command line in no time.

### 1. Command Line Bootcamp

If you are not familiar with the command line, fear not, the online command line bootcamp is a nice way to get started. You can spend some time together with the tutor - and/or individually - walking through this tutorial

1. if you are within the AppliedBionformaticsFrankfurt network you have access to an [interactive environment](#)
2. from outside the network, please open the [static command line bootcamp](#) together with a [shell on your system](#)

Memorize the individual commands, and it might be good idea to generate yourself short wiki pages that outline the individual functions together with the most relevant options. See the following pages as an example:

- Changing directories: [cd](#)
- locating your position in the directory tree: [pwd](#)
- looking into files: [less](#)
- linking files: [ln](#)

Remember that the DokuWiki is a shared resource and that you can use to work together when creating these notes.

### 2. Bash puzzles

- We will start by opening the shell. Copy the directory with the bash puzzles into your own

```
cp -r /home/ruben/Desktop/How-to/bash_miniguide ~/
```

[Alternatively, you can download them from](#)

here

- Each exercise contains a README file with instructions. Hints and useful commands are found in the help section.

## Enjoy your new power

---

### 3. The final boss

Professor Ebersberger prepared a heartwarming message, but the tutor decided to corrupt it:

[message.txt](#)

```
ajtxdfsdbrxKKmbbbbbbbbnjyfvylxdm
aliushsjwnKlpqk$jjw4iutjsmrwxwprt
u$dkslncwdikbmjlznm4$xjwhhuws$zsg
vpdlrdK$scfztmzmmnlhhjhkuccjhyy
gywolhzkdjchKj$vdbs_wqjufqdkkhfKs
gzkppuzcrnwugnzunKj9opsfikwzmc$tv
cgbuwuvychKkynuwm$vtnrqvlh$fixdl
mcubfbmvrozsxuytkcp4mzyzsfglqbftv
qiyghrtyrnbmplftdflyjhcyzqbcruw$
rKmxzizjztdbsfyqnaaaaaaapcjgksbrvl
cutxtszqgiuKhjcqKgl2ikipinzKinmnq
mopnjprmyrhrwgkptxmw$ykkqgtKlwckS
nmrhj$hyniqmdksgszl7fqKumtzinykvt
kwcyjfokfhmsvdrKlr0xvgtqjv$btghs
lubdtrlqkumhoxmmosl7ntiwbv$jtjoK
lydvjKzyfpw$uwtkzkw_nruofvtthk$ug
bixcokiyngorzjlgKhwxofKuzkvrpjh
glKqdudq$ypipnydbbd8gzdc$ogkoxrtK
nrzjh$qly$uirKcmg6Kuzujhjwfmzql
ntKycbgzxvhK$$qknjqKhndfxhdvl$$ks
nviqqzsktwhryqkjbzlKmqkotfghfdycn
nwbhtfdtqokgookdwgx3uzuhKn$zwKwsc
d$xgKm$nguicrxftptuKmdvlivq$ktniy
ehlfmxdqpbxlvkbmqcm_pwzyqtyonroki
fnzxbglvgxjtfjyptqh$plthdtlbibhj
otzvubvbrmhbtKKprdhkcsymgpxrvvsu
pbobvqvlswgdbwhqkuKqflquqwpwfdiK
qcyhmzjdbncqnxccccccKrKqmbfuwwq
gmxmcnlsfxpjijjiwu6Kshyzysbwfnud
tliryqipzmrurhjypj5dhxgzvwfjhwdw
txyvvnvfjrmziyjiuhK2v$cuxtKjgvutg
tzcxpskunxhoxhyjnjf2uqzowqzyksblK
h$gyhbmsckritgdddddd$oblpdmppqy
i$dcwssvps$qirwfymy0o$xgzpxuibcpm
idxgdqrk$jjxkotxccliKKut$rpjlylxb
```

```
jbbkrjgKfvu$voggbtccojkwhwdydkzc$
kqrbkgfucu cmfglzwutKhnyucpduKwsKq
vwsfnmssyzucKwtxvfb5tzx$qmzpgqKyy
vykspgktfgKqlufqtlr2hqdioqsfppfo
wdqnfq$orlbongwsrlz1vmhzc pdvj tKsy
wmlrylbmnfwzvohyokl3ctmbvvpysndu
wpfgkgs cKpvifmiqpmp9j$cmmwtocx lci
xojuilwkykvh$hjjjig8fpzthxyKdhykp
xovyzjukwdsycjxkwqr3gbsywd rjrvufy
yKprfKy ydsgnvzwqfnp5tsrKqiqyzKjuw
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

### codex.txt

```
aaaaaaa rkydfrp
bbbbbbb vgk khbw
ccccccc lskhrnt
ddddddd hmKlcnd
xxxxxxx xxxxxxx
```

In order to decode the secret message, the following will be needed:

- Files will be downloaded to your “~/Downloads”.
- Use the codex to restore the words from the first column, into the second one. (Make a backup. Try a combination of while-read + sed -i “”)

```
cat codex.txt | while read n k; do sed -i "s/???/???" $n ; done
```

- Get rid of the lines with numbers
- Instead of capital letter **K**, we need the letter **e**
- Instead of **\$**, we need the letter **a**
- Sort the lines
- The message should be in the 20th column.
- Read them in one line

[← Back to main](#)

[← Back to workplan](#)

1)

Sometimes people refer to the shell also as *terminal* or *command line*

2)

also called 'terminal'

Last  
update:  
2025/01/15 13:26 general:computerenvironment:bashbasics <https://applbio.biologie.uni-frankfurt.de/teaching/wiki/doku.php?id=general:computerenvironment:bashbasics>

---

From:  
<https://applbio.biologie.uni-frankfurt.de/teaching/wiki/> - **Teaching**

Permanent link:  
<https://applbio.biologie.uni-frankfurt.de/teaching/wiki/doku.php?id=general:computerenvironment:bashbasics>

Last update: **2025/01/15 13:26**

