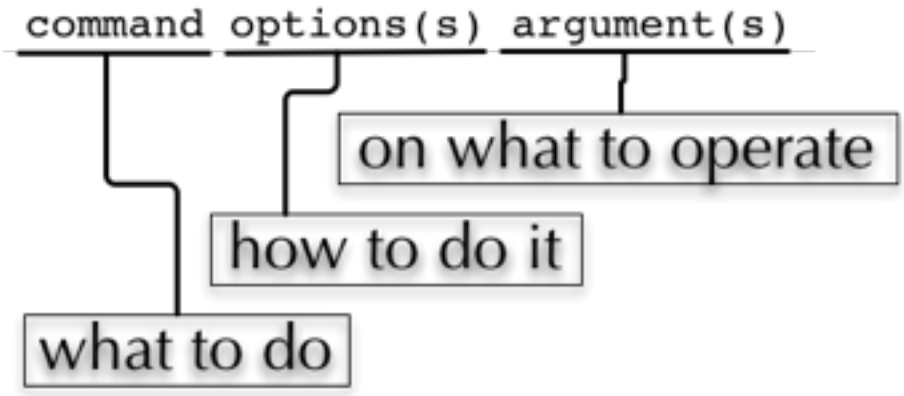
Software Carpentry Course
EMBL Heidelberg, Sep 19-21, 2016

Automating tasks with the Unix shell

Frank Thommen, Theoretical Bioinformatics (B080), DKFZ Heidelberg and HD-HuB (de.NBI)
(based on previous work with Holger Dinkel, EMBL Heidelberg, see acknowledgements)

General Structure of Linux Commands



Commandline options are sometimes called commandline switches.

Most common form of commandline switches:

- Short form: `-h`
- Short form with additional parameter: `-f myfile`
- Long form: `--help`
- Long form with additional parameter: `--file=myfile`

Mixed forms and parameters without leading dash can also be encountered.

Getting Help

Help option: `cmd -h / cmd --help`

Manual page of a command: `man command`

List manpages containing a keyword in their description: `apropos keyword`

Files in /usr/share/doc

Who and Where am I?

Print your username: `whoami`
Print the name of the computer: `hostname`
Print the current working directory: `pwd`
Print current date and time: `date`

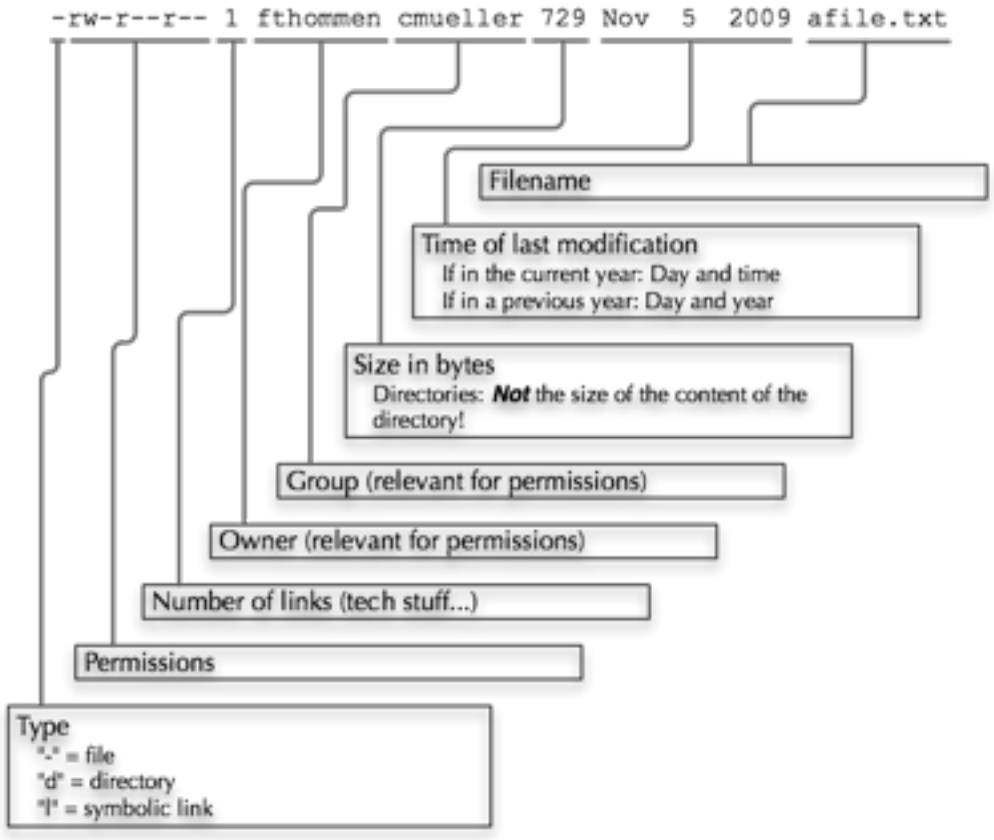
Moving Around

Change the working directory: `cd [new_directory]`

See What's Around

List directory contents: `ls [options] [file(s) or dir/s]`

-l (Long) / -1 (one column) / -a (all files) / -A (almost all files) / -F (show filetypes) / -d (directory information instead of directory content) / -t (most recent on top)



On shell globs

Files and folders can't only be referred to with their full name, but also with so-called "Shell Globs", which are a kind of simple pattern to address groups of files and folders. Instead of explicit names you can use the following placeholders:

?:	Any single character
*:	Any number of any character (including no character at all)
[…]:	One of the characters included in the brackets. Use "-" to define ranges of characters
[!…]:	*Not* one of the characters included in the brackets.
[^…]:	ditto.
~:	User's home directory (only if first element of a path)

On filenames

- Stick with lowercase names
- Don't use blanks in filenames
- Don't use other characters then alphabetic characters, numbers, "-", "." and "_"

Organizing Files and Folders

Remove files and directories: `rm [options] file(s)`

`rm -r [options] directory/ies`

-i (interactive) / -r (recursion) / -f (force)

Dangerous command!

Move and rename files and folders: `mv [options] source dest`

`mv [options] file(s) dir`

-i (interactive)

Dangerous command!

Create a new directory: `mkdir [options] directory`

Remove an empty directory: `rmdir directory`

Copy files and folders: `cp [options] src dest`

`cp [options] src(s) destdir`

-r (recursion) / -i (interactive) / -p (preserve)

View Files

Print files on terminal (concatenate): `cat [options] file(s)`

View and navigate files: `less [options] file(s)`

Extracting Informations from Files

Find lines matching a pattern in files: `grep [options] pattern file(s)`
-v (not matching lines) / -i (case insensitive) / -l (list filenames) / -L (list iles w/o matches) / -c (print counts of matching lines)

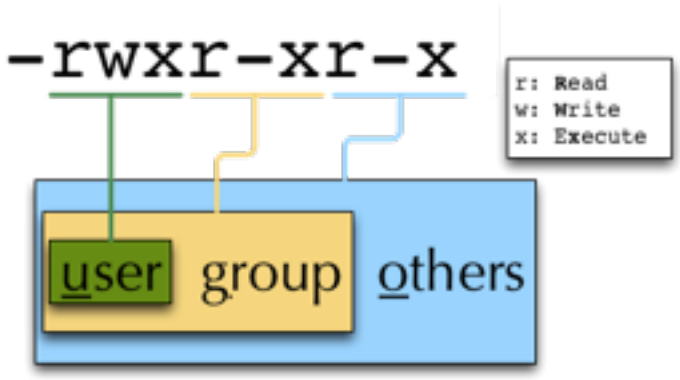
Print first lines of a textfile: `head [options] file(s)`
-n *num*

Print last lines of a textfile: `tail [options] file(s)`
-n *num* / -f ("follow")

Useful Filetools

Search/find files in any given directory: `find [start] [filter]`
find is very powerful. Refer to the manual page for possible filters

Permissions



List Permissions: `ls -l file / ls -ld directory`

Set Permissions: `chmod [options] mode(s) files(s)`

The mode is composed of

Who	What	Which permission
u: user/owner g: group o: other a: all	+: add this permission -: remove this permission =: set exactly this permission	r: read w: write x: execute

IO and Redirections

Redirect the output of one program into e.g. a file: (**Caution**: you can easily overwrite files by doing this!)

```
# date > file_containing_date
```

Append something to a file (rather than overwriting it):

```
# date >> file_containing_date
```

Feed the output of one program into the next progam

```
# ls -al | wc -l
```

Varia:

Display text on screen: `echo text`

for loop

```
for variable in list
do
  commands
done
```

if statement

```
if condition1
then
  commands
elif condition2
then
  more commands
[…]
else
  even more commands
fi
```

File conditions

`-e file`	*file* exists
`-f file`	*file* exists and is a regular file
`-d file`	*file* exists and is a directory
`-r file`	*file* exists and is readable
`-w file`	*file* exists and is writeable
`-x file`	*file* exists and is executable
`-s file`	*file* exists and has a size > 0

String Comparisons

`-n s1`	String *s1* has non-zero length
`-z s1`	String *s1* has zero length

`s1 = s2`	Strings *s1* and *s2* are identical
`s1 != s2`	Strings *s1* and *s2* differ
`string`	String *string* is not null

Integer Comparisons

`n1 –eq n2`	*n1* equals *n2*
`n1 –ge n2`	*n1* is greater than or equal to *n2*
`n1 –gt n2`	*n1* is greater than *n2*
`n1 –le n2`	*n1* is less than or equal to *n2*
`n1 –lt n2`	*n1* is less than *n2*
`n1 –ne n2`	*n1* it not equal to *n2*

Links and Further Information

- A full 500 page book about the Linux commandline for free(!): LinuxCommand.org (http://linuxcommand.org/)
- Another nice introduction: "A beginner's guide to UNIX/Linux" (http://www.mn.uio.no/astro/english/services/it/help/basic-services/linux/guide.html)
- The "commandline starter" chapter of an O'Reilly book: Learning Debian GNU/Linux – Issuing Linux Commands (http://oreilly.com/openbook/debian/book/ch04_01.html)
- A nice introduction to Linux/UNIX file permissions: "chmod Tutorial" (http://catcode.com/teachmod/)
- Linux Cheatsheets (http://www.cheat-sheets.org/#Linux)
- For the technically interested:
  Linux Filesystem Hierarchy Standard (http://www.pathname.com/fhs/) and Linux Standard Base (http://www.linuxfoundation.org/collaborate/workgroups/lsb)

Acknowledgements

- These pages are a very, very abbreviated version of a more detailed documentation of a Linux course given between 2013 and 2015 by Holger Dinkel and Frank Thommen at EMBL Heidelberg in the frame of the Bio-IT project. You can find the most recent handouts of this course at https://git.embl.de/dinkel/linuxcommandline in the "_build/latex" section of each course directory
- Graphic of the Linux Filesystem on page 3 from the SuSE 9.2 manual © Novell Inc.
- All other graphics © Frank Thommen, EMBL Heidelberg, 2012