Basics of Unix

Libor Mořkovský, Václav Janoušek

What are we going to learn today?

- Move around directory structure
- Create directories
- List and explore content of directories
- View plain text files
- To copy, move and rename files
- Work with compressed files
- Use variables and lists
- To download and install software

How to connect to remote machine

```
ssh userXX@ngs-course.duckdns.org
```

Command line

To type commands (syntax):

```
name -flag(value) input > output

head -n20 file.txt > out.txt
```

What if I don’t know?

```
man head
head --help
head -h
```

Take a break and check your keyboard…

[]	squared brackets
{}	curly brackets
<>	angle brackets (smaller-than, bigger-than sign)
()	parentheses
~	tilde
/	slash
\	back slash
|	pipe
^	caret
$	dollar sign
:	colon
;	semicolon

.	dot
,	comma
#	hash
_	underscore
-	dash
*	asterisk
!	exclamation mark
?	question mark
&	ampersand
@	at sign
''	quotation mark single
""	quotation mark double

screen

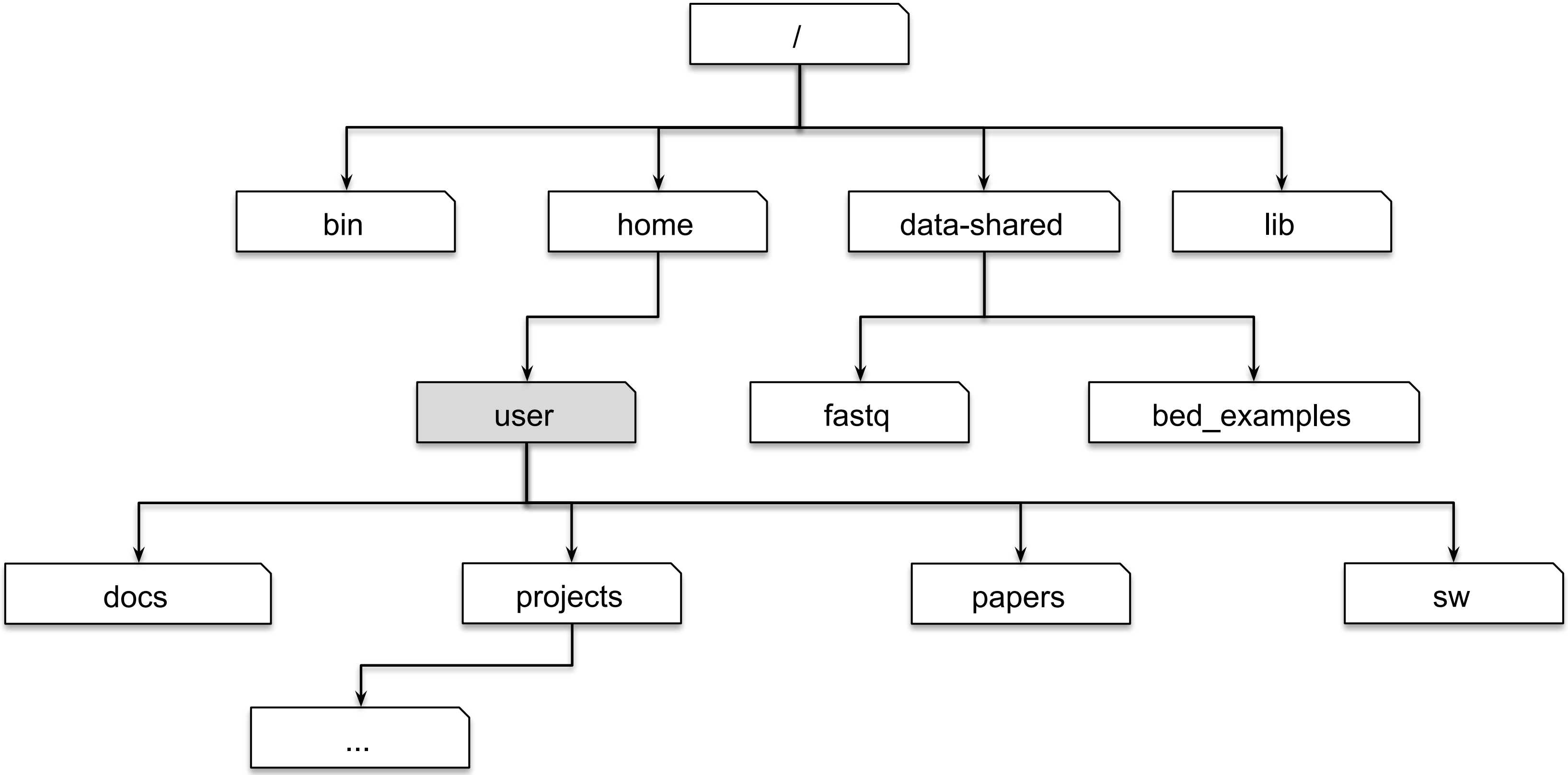
Protection from unexpected connection drop outs…

```
screen
screen -r
screen -ls
```

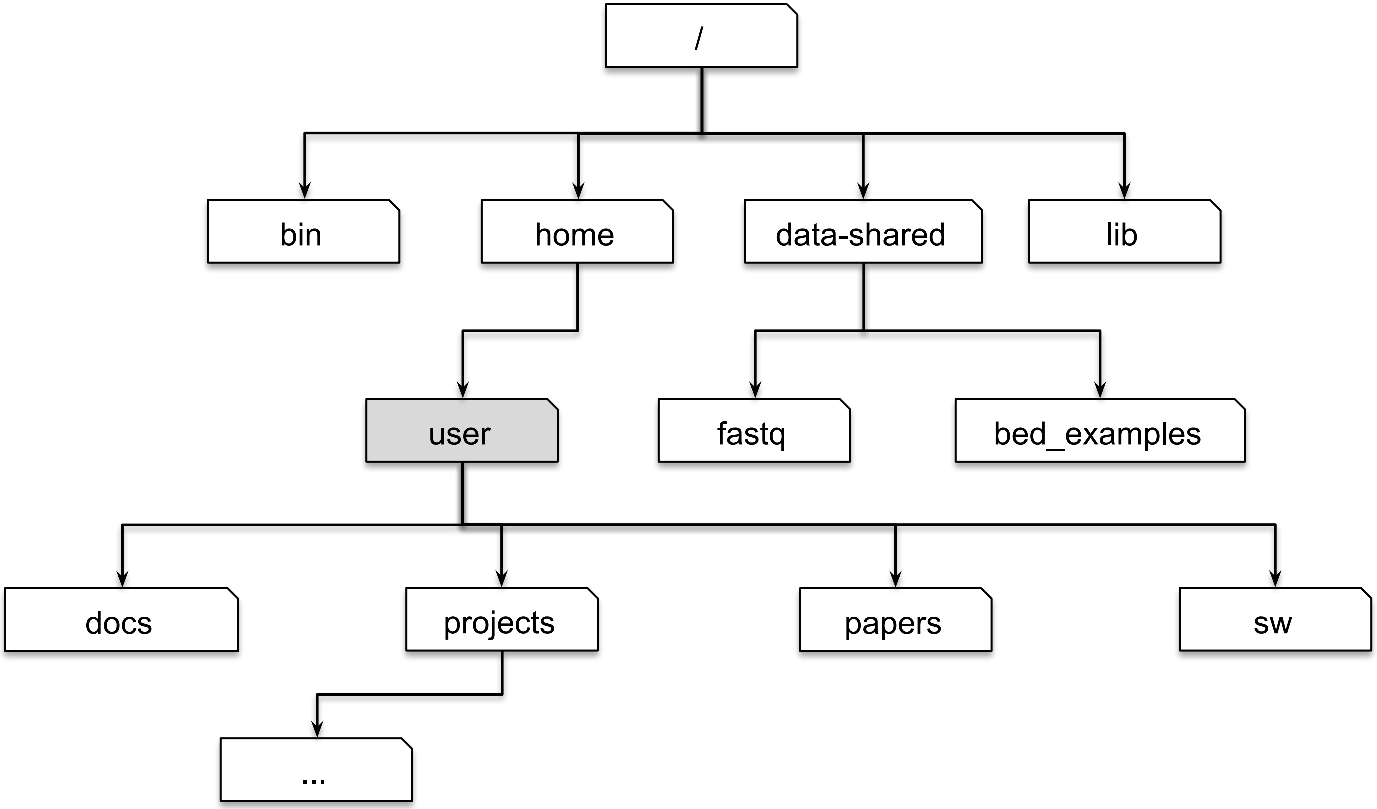
```
# inside screen
ctrl+a c          # new window
ctrl+a space      # switch between multiple windows
ctrl+a d          # detach from the screen
```

* ***tmux*** *- a bit more user friendly and powerful alternative to* `screen`

Basic directory structure of unix

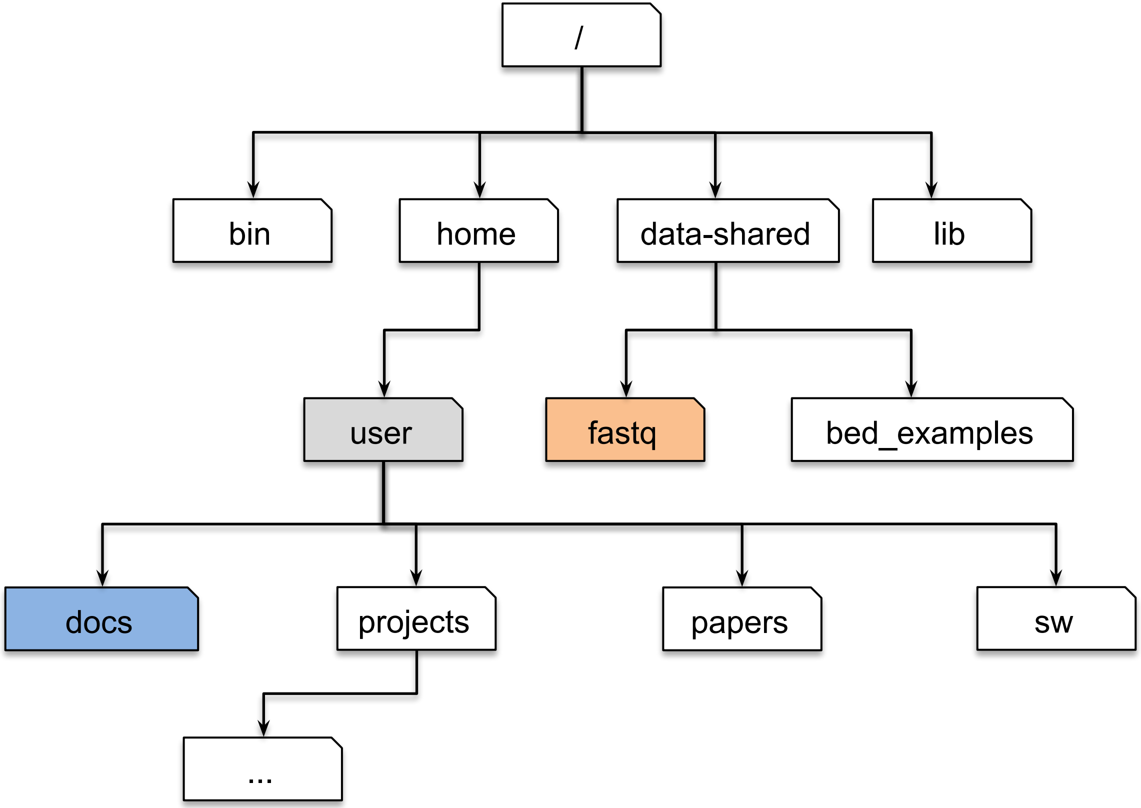


Moving around



```
pwd
ls
ls ~
ls /
ls ..
ls ../..
cd
cd ~
cd /
cd ..
cd -
```

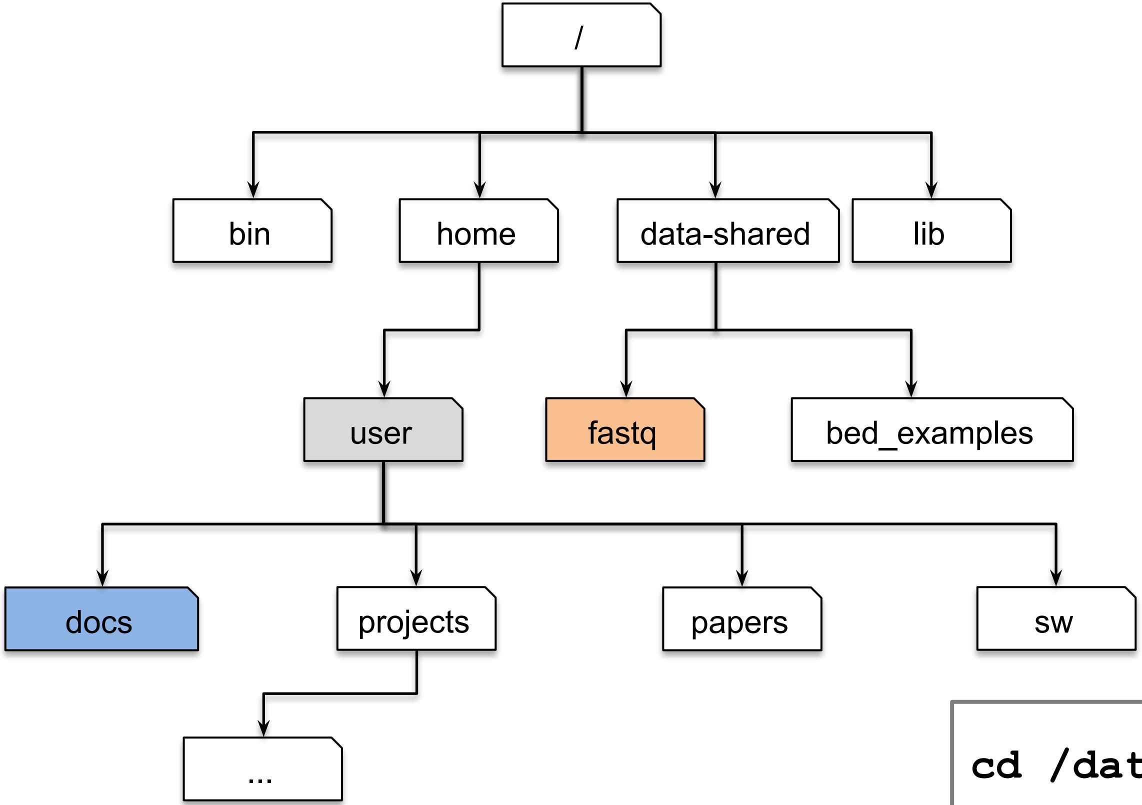
Absolute vs. relative path



<u>Exercise:</u>

Use absolute and relative path to move from 'docs' (blue) to 'fastq' (red)

Absolute vs. relative path

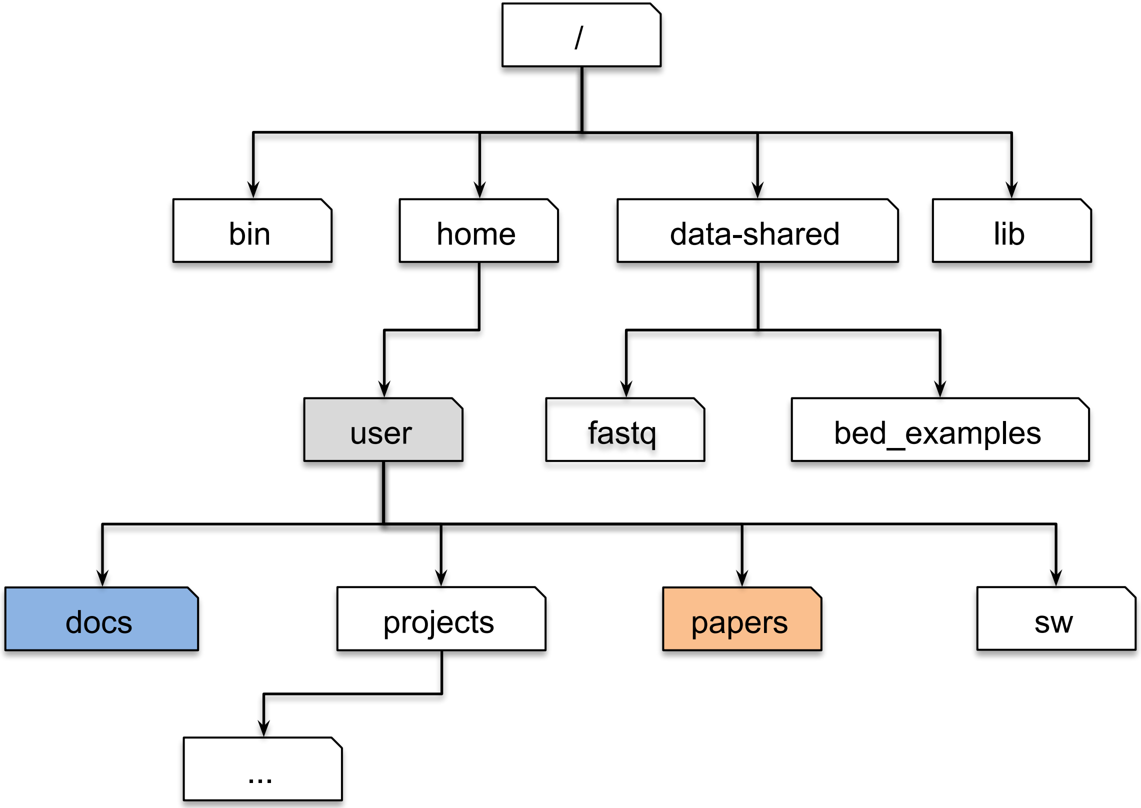


Exercise:

Use absolute and relative path to move from 'docs' (blue) to 'fastq' (red)

```
cd /data/fastq
cd ../../../data/fastq
```

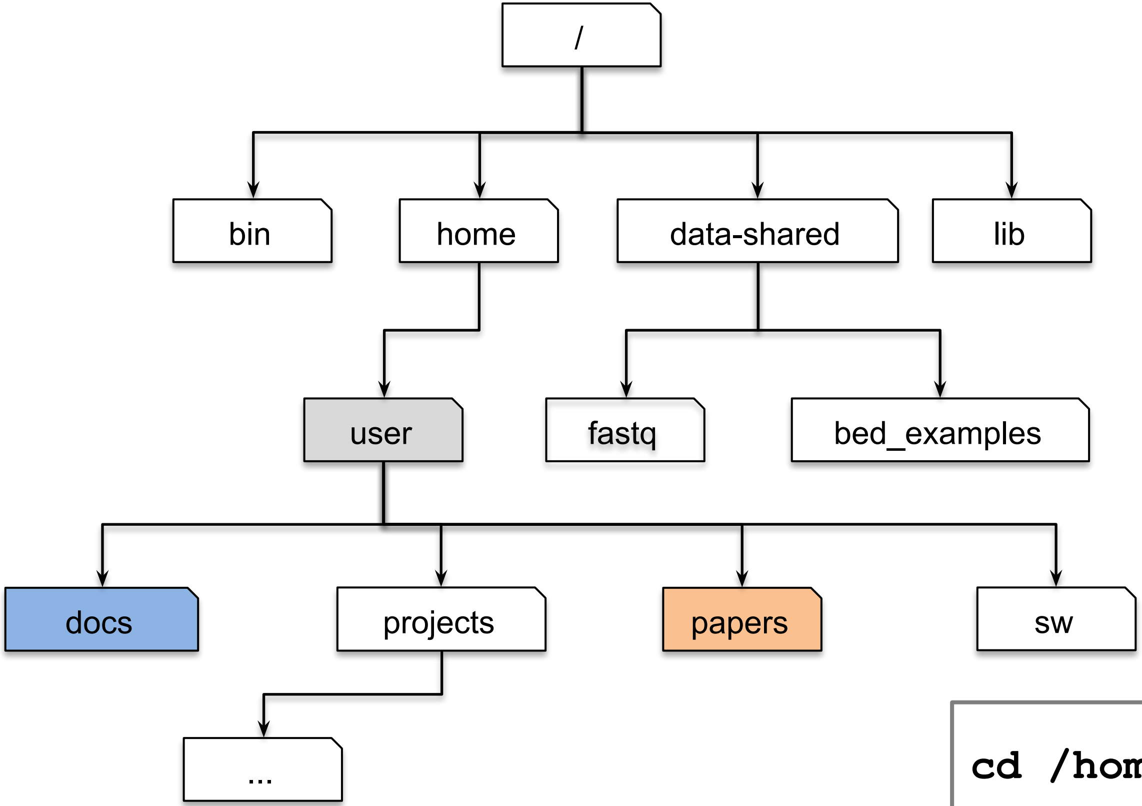
Absolute vs. relative path



Exercise:

Use absolute and relative path to move from 'docs' (blue) to 'projects' (red)

Absolute vs. relative path



Exercise:

Use absolute and relative path in to move from 'docs' (blue) to 'projects' (red)

```
cd /home/user/projects
cd ../projects
```

Moving and copying files or directories

Try these tools to:

- make new files/(sub)directories
- move and rename them
- remove them

```
touch         # make empty file(s)
mv            # move/rename files
cp (-r)       # copy files (-r directories)
mkdir (-p)    # make directory (-p subdirectory)
rm (-r)       # remove file (-r non-empty directory)
```

Viewing plain text file content

```
less -SN
tail -n8
head -n8
cat
nano
```

Work with compressed data

```
# only gzipped (only one file)
gunzip file.gz

# view content of a compressed file
zcat fastq.gz | less

# gzipped tarball archive
tar -xzvf fastq.tar.gz
```

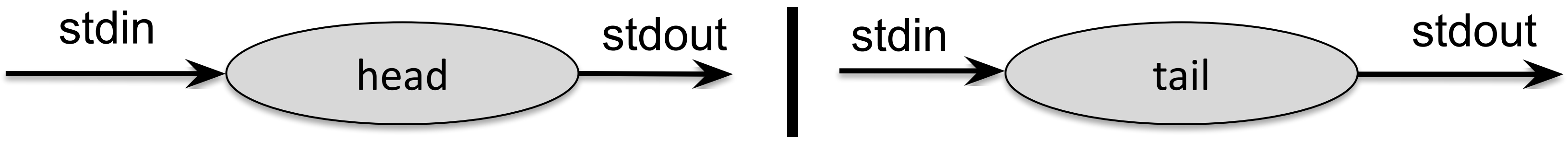
Exercise

Prepare FASTQ data file:

```
# go to home directory
cd
# make a new dir
mkdir projects/fastq && cd projects/fastq
# copy a fastq tarball to the new dir
cp /data-shared/fastq/fastq.tar.gz .
# decompress files
tar –zxvf fastq.tar.gz
# list files
ls -sh
```

Pipes '|'

Chaining standard input and output:



```
head -8 HRTMUOC01.RL12.00.fastq | tail -4 | less

# Neater way to structure pipelines
< HRTMUOC01.RL12.00.fastq head -8 | tail -4 | less
```

Globbing & wildcards (*, ?, [class])

What if I need to choose multiple files?

```
cd ~/projects/fastq

ls *.fastq # choose all fastq files

ls HRTMUOC01.RL12.0?.fastq # one character

ls HRTMUOC01.RL12.0[1-9].fastq # one numerical character
```

Exercise

How many reads are in all fastq files?

```
cd ~/projects/fastq
cat *.fastq | wc -l

expr XXXX / 4 ## Or
echo $((XXXX/4))
```

Variables

Variable: storage location paired with an associated symbolic name

```
CPU=4
echo $CPU

FILE=~/projects/fastq/HRTMUOC01.RL12.00.fastq
echo $FILE
```

Loops

Loop over set of parameter values:

```
PARAM=$(echo {0..9})

for v in $PARAM
do
    echo $v;
Done

# one line syntax
For v in $PARAM; do echo $v; done
```

Installing software in Unix

- *The easiest way is to use package manager (apt-get)*

```
sudo apt-get install htop
sudo apt-get install tmux
```

Installing software in Unix

- *Otherwise we have to download the source code and compile it on its own (canonical way in Unix):*

```
# Downloading compressed source code
wget –O - ..url.. | tar xvz

# Cloning from Git repository
git clone ..url..

# Compilation of binaries
cd ...directory..
./configure
make
sudo make install
```

bedtools2

- *See our website*

```
# Download the compressed source code
wget https://github.com/arq5x/bedtools2/releases/download/v2.25.0/bedtools-2.25.0.tar.gz
tar -zxvf bedtools-2.25.0.tar.gz

# Or clone Git repository
git clone https://github.com/arq5x/bedtools2

# Compile binaries
cd bedtools2
make
```

What have we learned today?

- Move around directory structure
- Create directories
- List and explore content of directories
- View plain text files
- To copy, move and rename files
- Read compressed files
- Use variables and lists
- To download and install software

Why should we learn unix shell? ChatGPT…

- You are welcome to use it as it is common these days among programmers but be aware of errors - it is a good servant requiring instant double-check

That's all for today...