Unix - Basics

Course on Unix and Genomic Data
Prague, January 2017

Libor Mořkovský, Václav Janoušek,
Anastassiya Zidkova, Anna Přistoupilová, Filip Sedlák

http://ngs-course.readthedocs.org/en/praha-january-2017/

```
user@localhost:~$
```

This is where all begins...

Command line

To type commands (syntax):

```
name (-flag(=flag-parameter)) (input) (output)
```

```
head -n 20 file.txt > file-out.txt
```

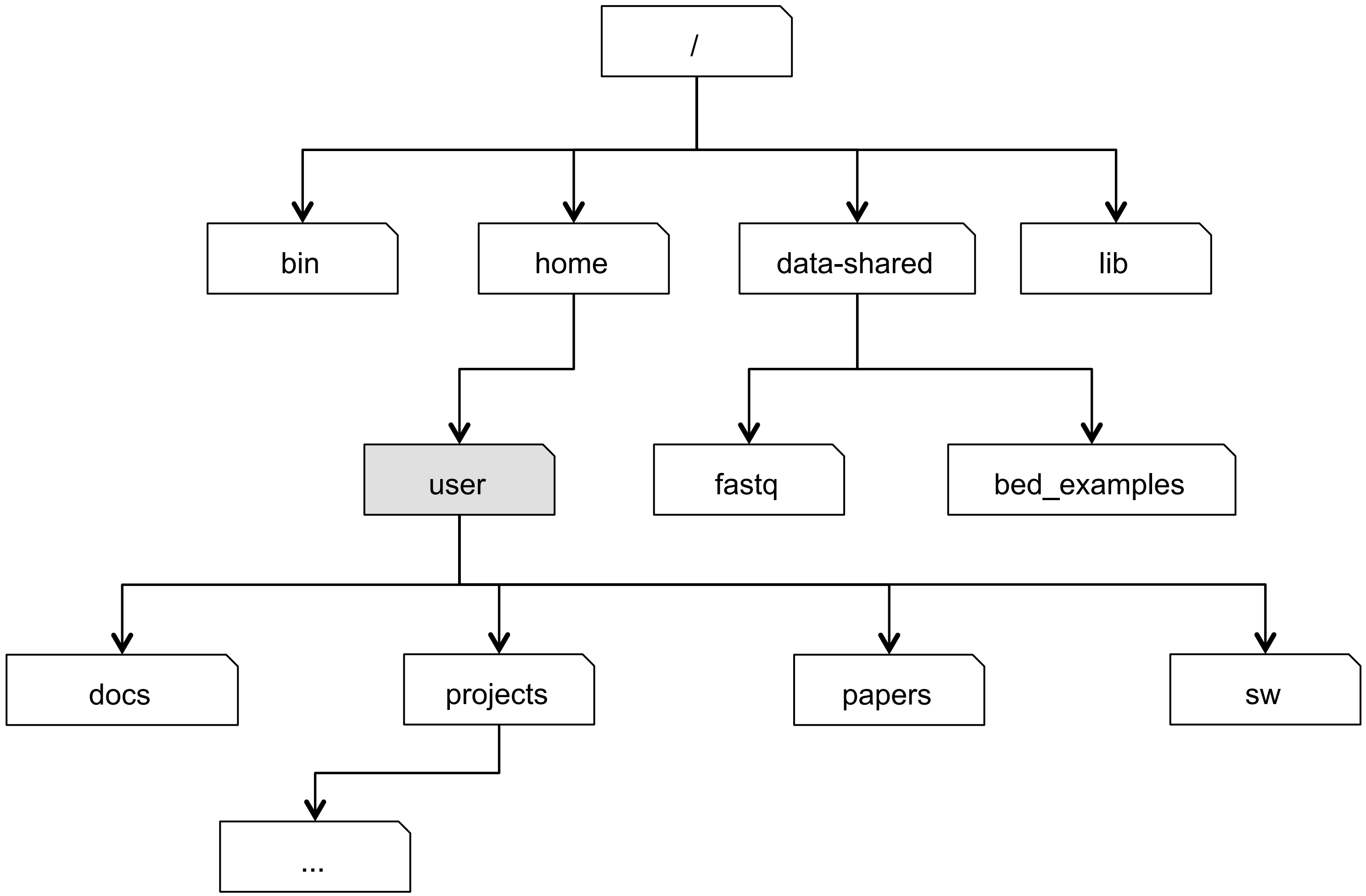
** man command || google it

Take a break and check your keyboard

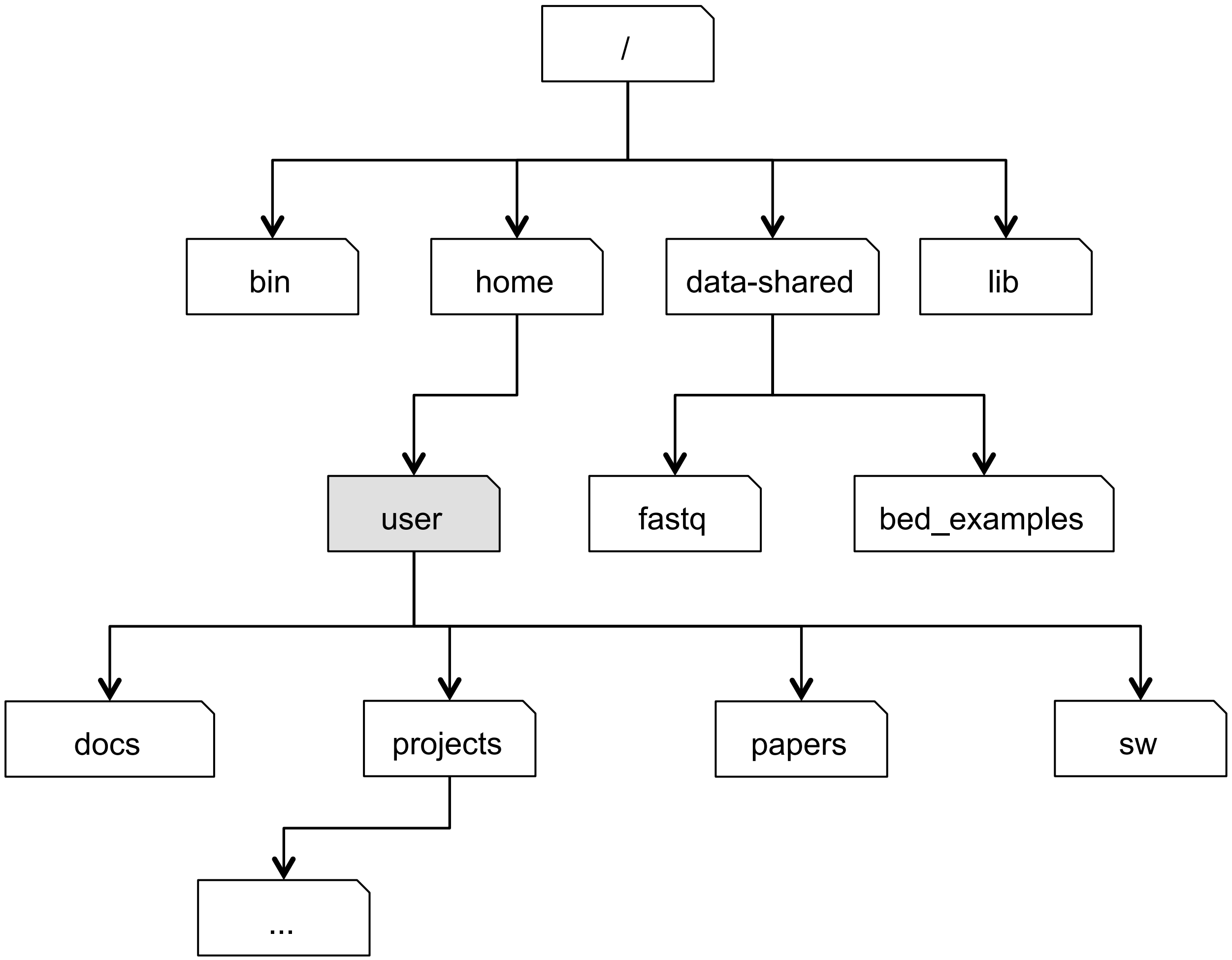
```
[] - squared brackets

{} - 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
```

Basic Structure

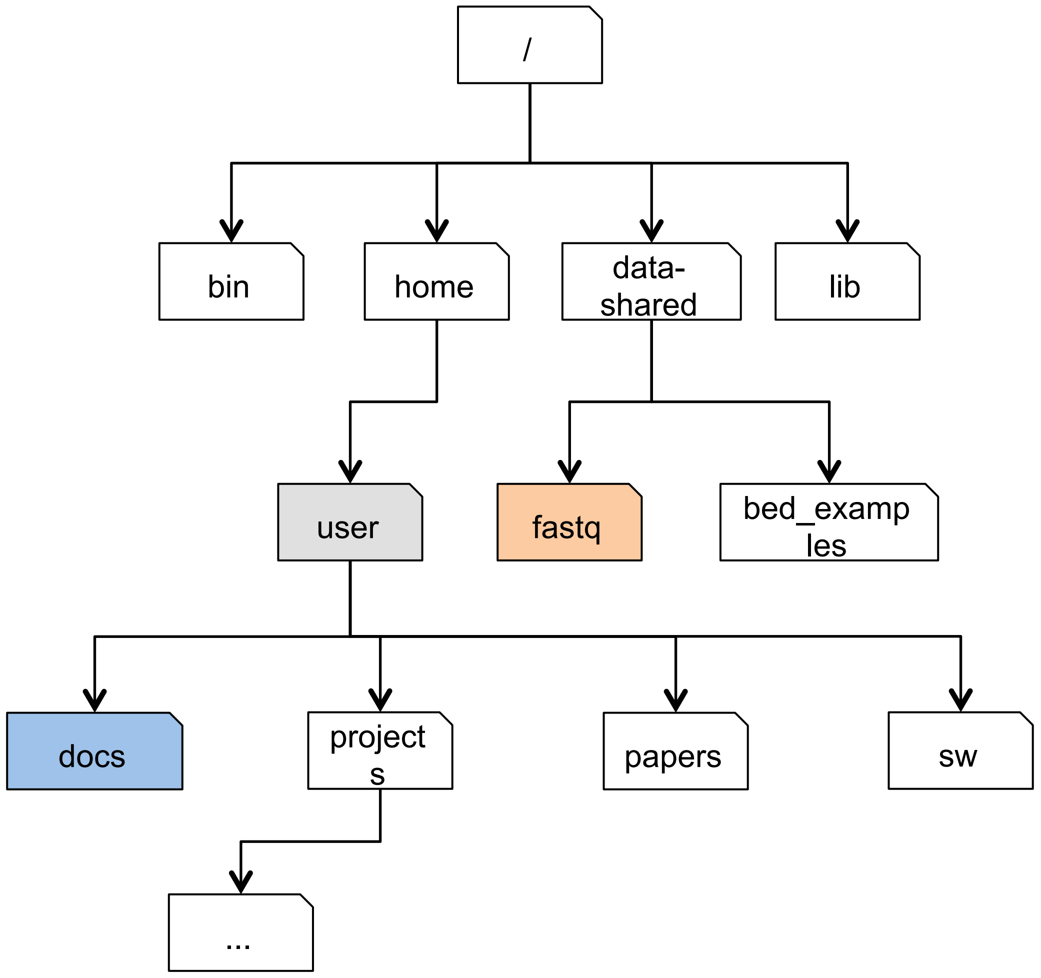


Moving Around



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

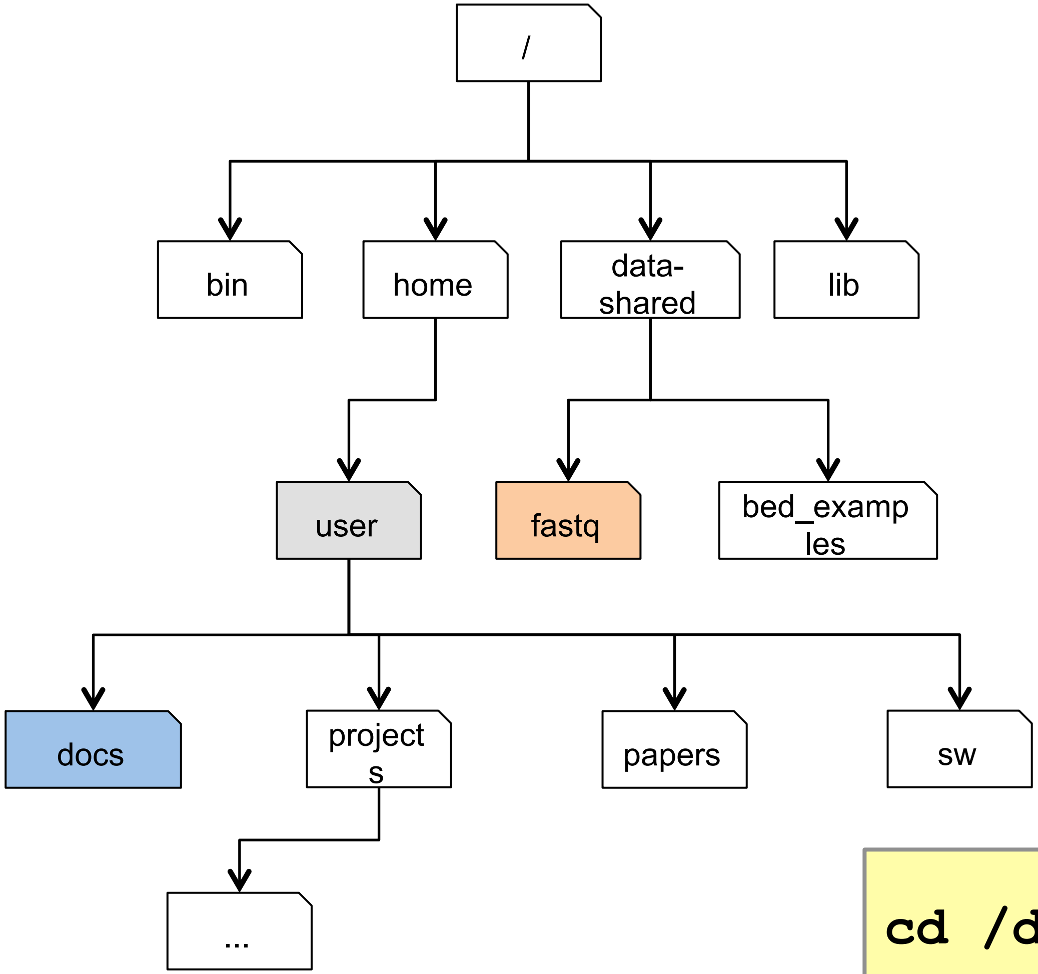
Absolute vs. Relative Path



Exercise:

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

Absolute vs. Relative Path

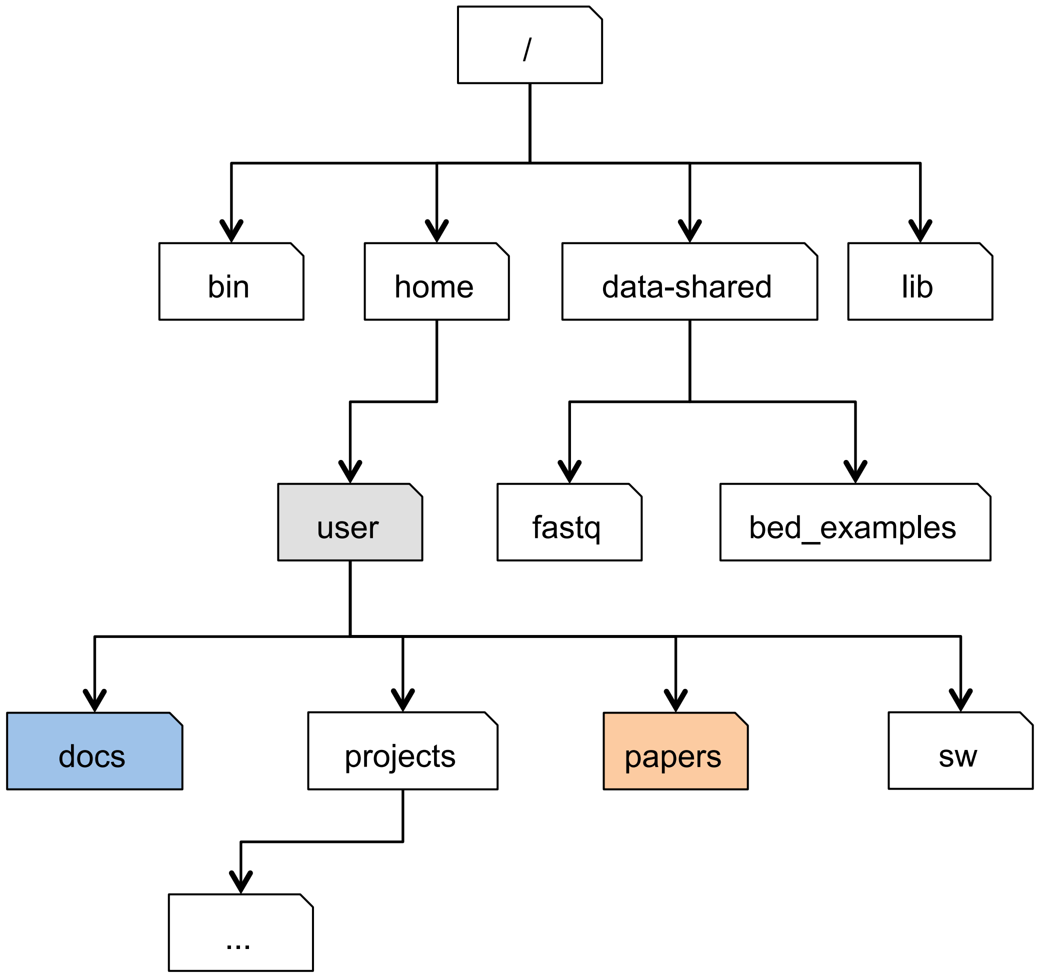


Exercise:

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

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

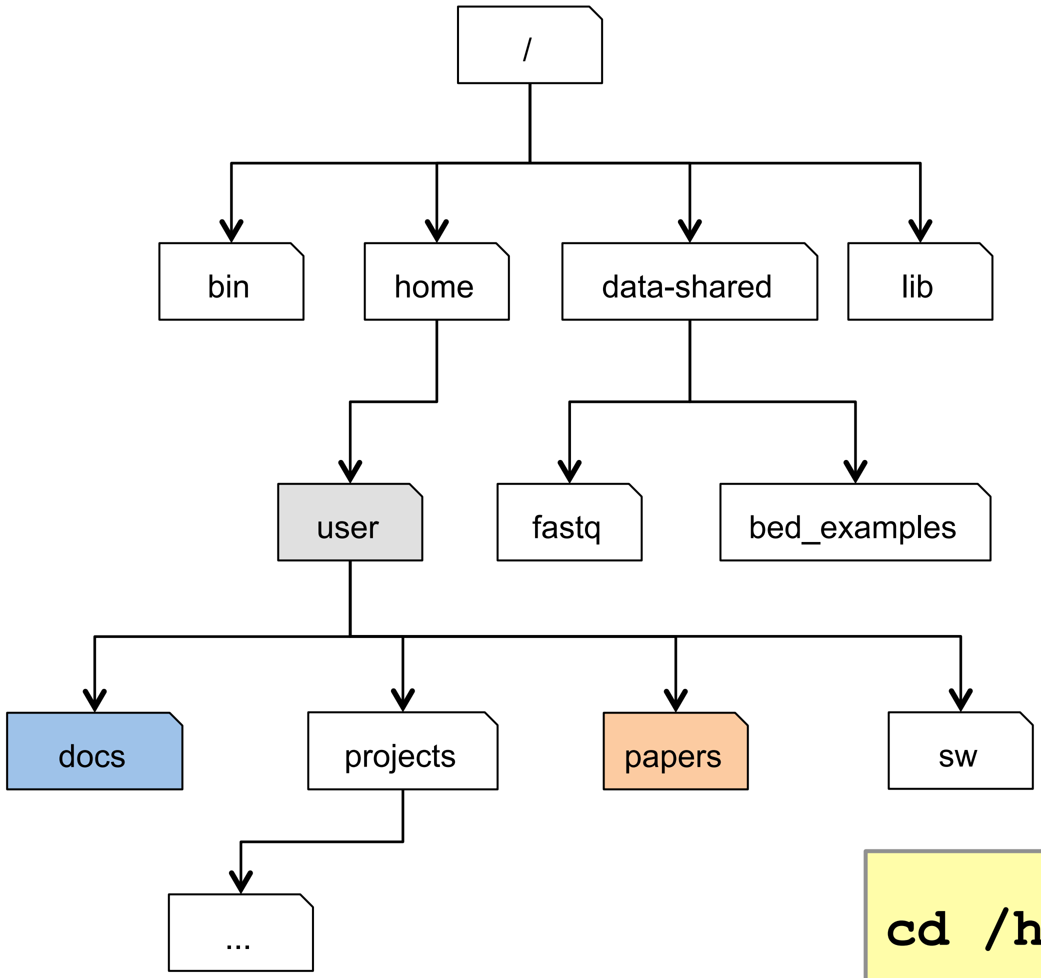
Absolute vs. Relative Path



Exercise:

Use absolute and relative path in 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/coping files/directories

Try these tools to:

- make new files/(sub)directories
- move/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)
```

Moving/coping files/directories

Prepare FASTQ data file:

```
cd ## Go to home directory
mkdir projects/fastq ## Make a new dir
## Copy a fastq file to the new dir:
cp /data/fastq/fastq.tar.gz projects/fastq/.
cd projects/fastq
tar -zxvf fastq.tar.gz
ls
```

Viewing compressed data/ Uncompressing data

```
tar -xzvf fastq.tar.gz ## tarball archive + gzip
(multiple files into one archive)

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

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

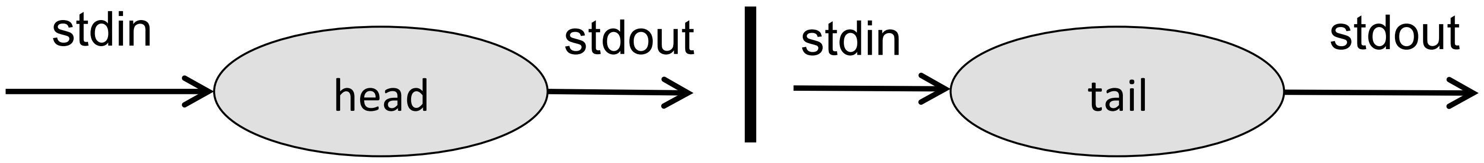
Viewing plain text file content

```
less -SN
tail -n 8
head -n 8
cat
nano
```

** TAB completion

Pipes '|'

Chaining standard input and output:



```
head -8 file.fastq | tail -4 | less

< file.fastq head -8 | tail -4 | less
```

Pipes '|' - Exercise

How many reads are there?

What does exactly 'wc' command do?

```
cd ~/projects/fastq
cat HRTMUOC01.RL12.00.fastq | wc -l

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

Globbing/Wildcards (*, ?, [class])

What if I need to choose multiple files?

```
cat file.*.fastq | wc -l

cat file.0?.fastq | wc -l

cat file.0[1-9].fastq | wc -l
```

Variables/Lists

Variable: storage location paired with an associated symbolic name

```
CPU=4
echo $CPU

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

Variables/Lists

```
echo file{1..9}.txt

LST=$( echo file{1..9}.txt )
echo $LST

LST2=$( ls ~/projects/fastq/*.fastq )
echo $LST2
```

Loops

Repeat a command (set of commands) multiple times:

```
LST=$( echo file{1..9}.txt )

for I in $LST
do
   echo $I;
done
```

Loops

Repeat a command (set of commands) multiple times:

```
FILES=$(ls ~/my_data/fastq/*.fastq)

for I in $FILES
do
   echo $I;
   head -n 1 $I | wc -c;
done
```

Multiple Windows in Unix

Yes, you can have them...

+ protection from unexpected network fails

```
screen
screen -ls
screen -r XXXX.NNNNNN.XXXX
screen -X -S XXXX.NNNNNN.XXXX quit
```

```
ctrl+a c
ctrl+a space
ctrl+a d
```

Installing Software in Unix

- *The easiest way is to use package manager (apt-get)*
- *Otherwise one needs to download the source code and compile it on its own (canonical way in Unix):*

```
wget -O - ..url.. | tar xvz
cd ..unpacked directory..
./configure # configuration of MAKE file based
on the OS
make # actual compilation of source code
sudo make install # installation of binaries
```

htop

- *Package manager*

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

bedtools2

- *wget*

```
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
cd bedtools2
make
```

** if you need the most recent (development) version – use 'git clone'

That's all for today…