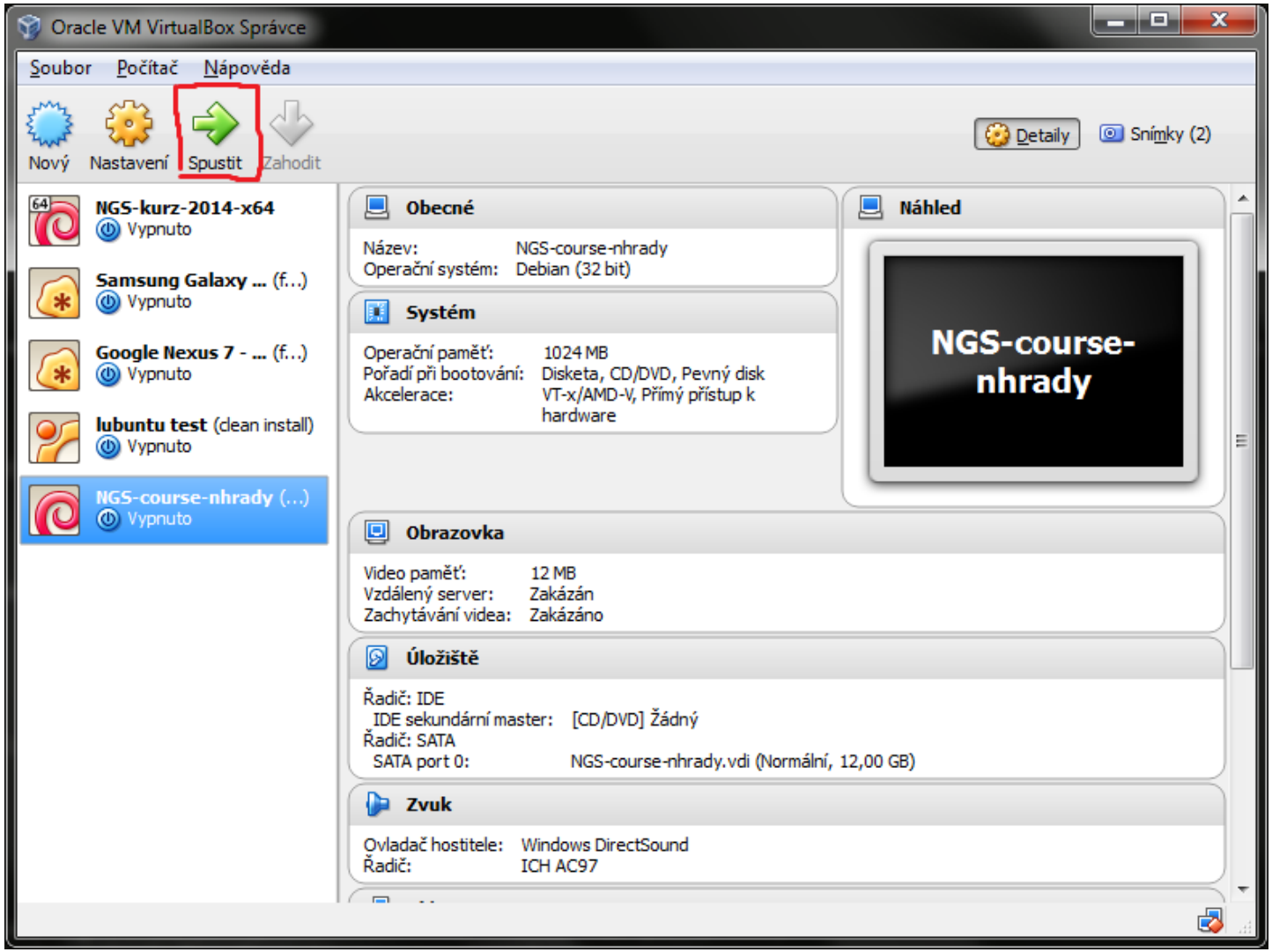
Unix - Basics

Course on Unix and Genomic Data
Prague, January 2016

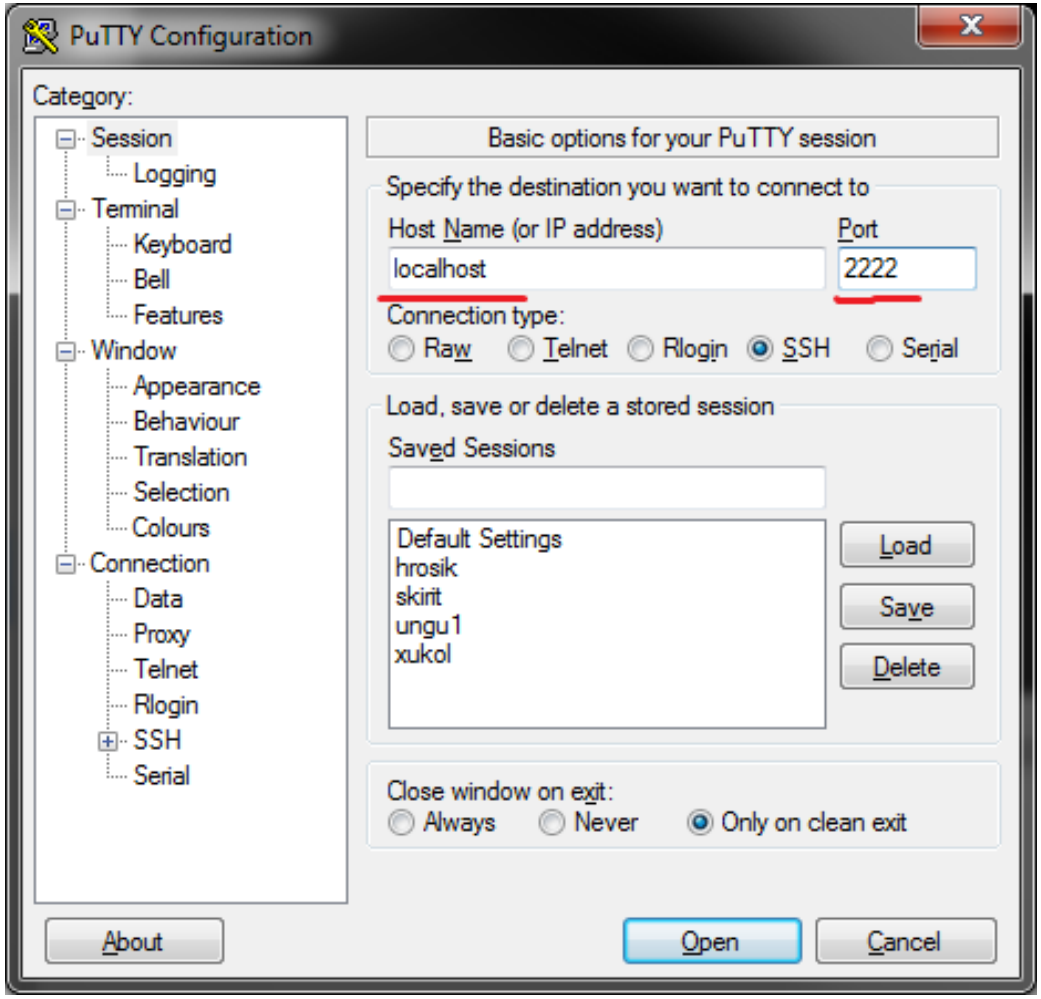
Václav Janoušek, Libor Mořkovský

http://ngs-course.readthedocs.org/en/praha-january-2016/

Start Virtual Machine



Get connected: PuTTY



OS X/Linux (command line):

ssh -p 2222 user@localhost

```
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

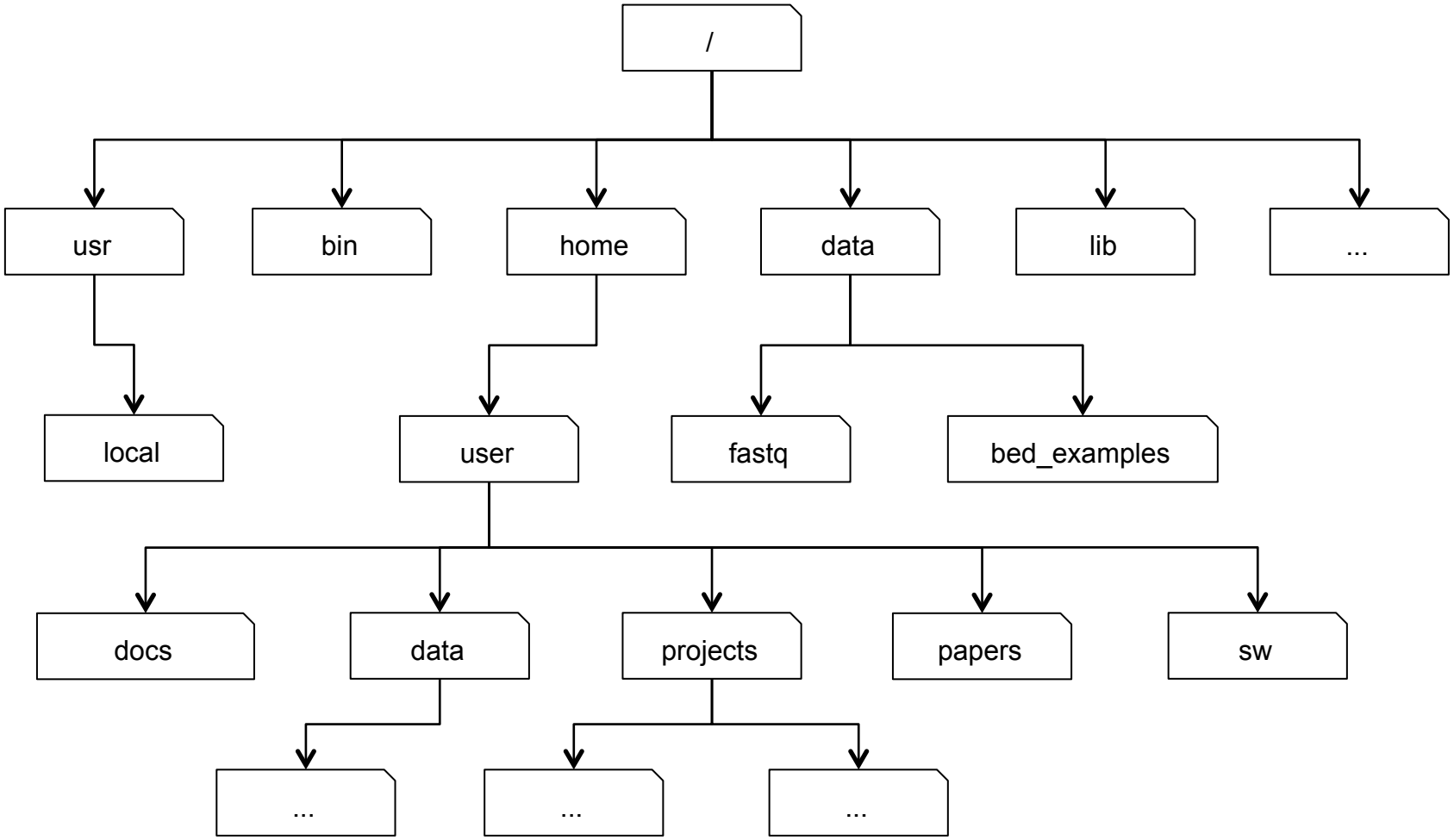
& - ampersand

@ - at sign

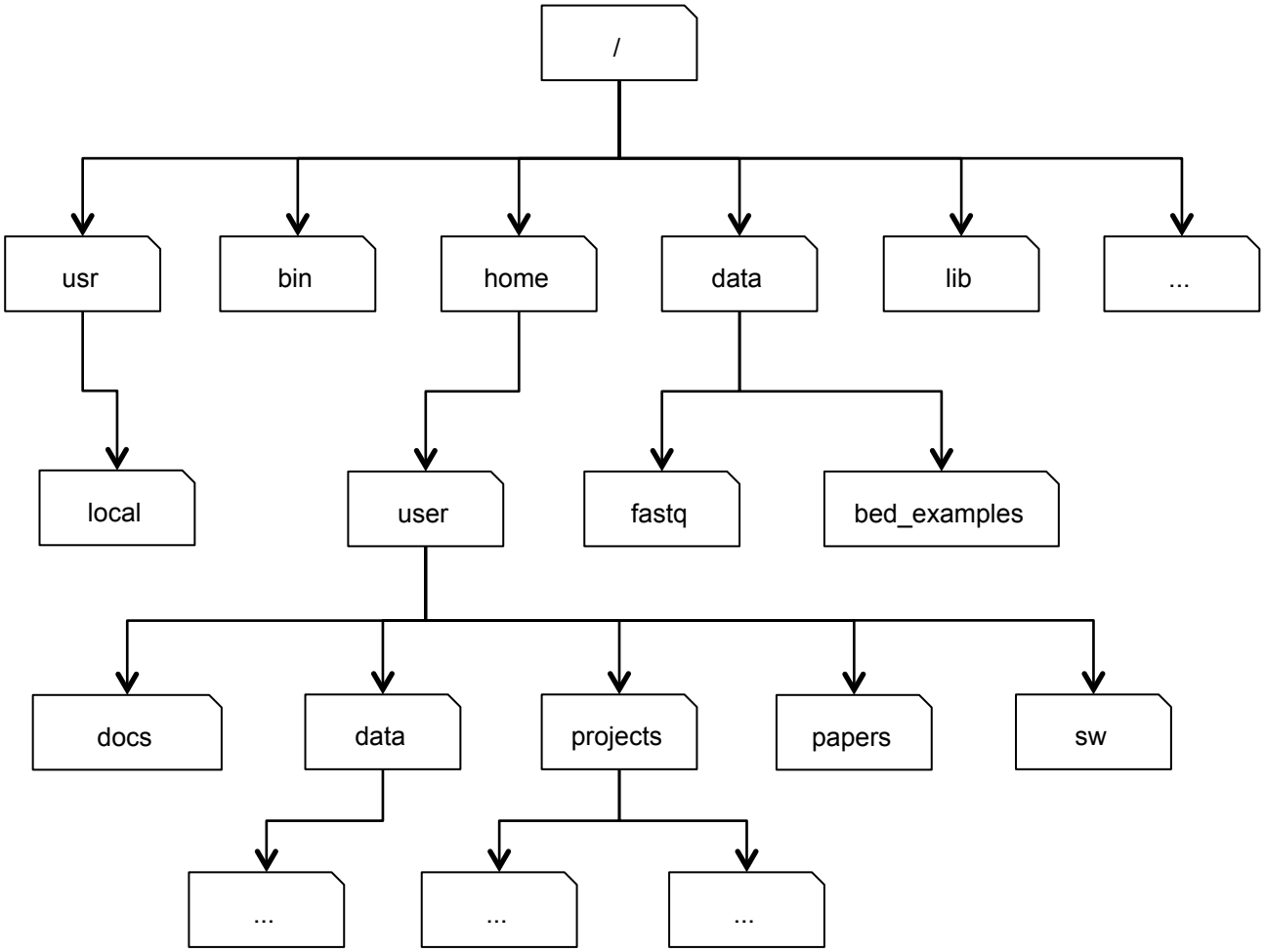
'' - quotation mark single

"" - quotation mark double

Basic Structure

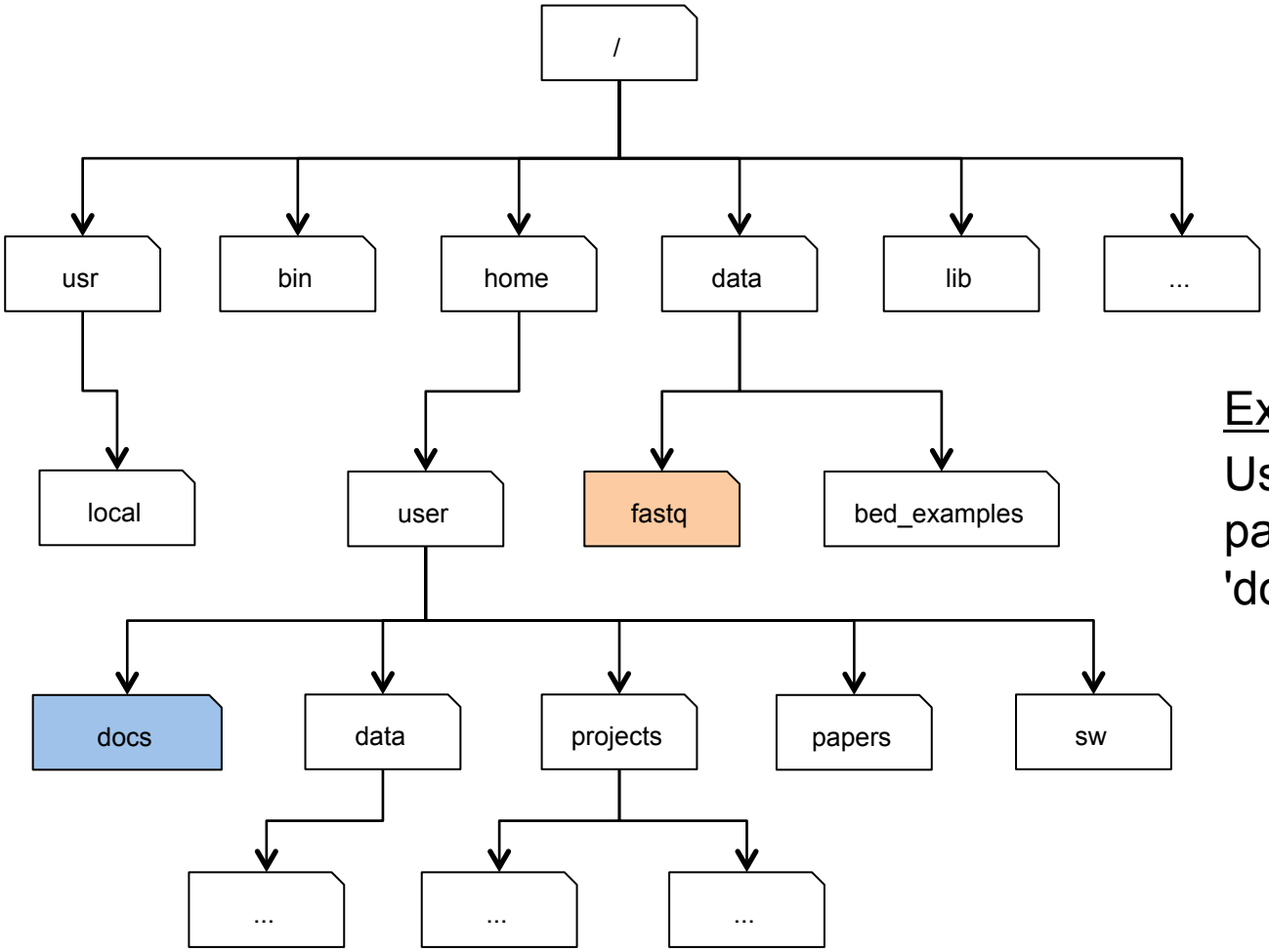


Moving Around




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

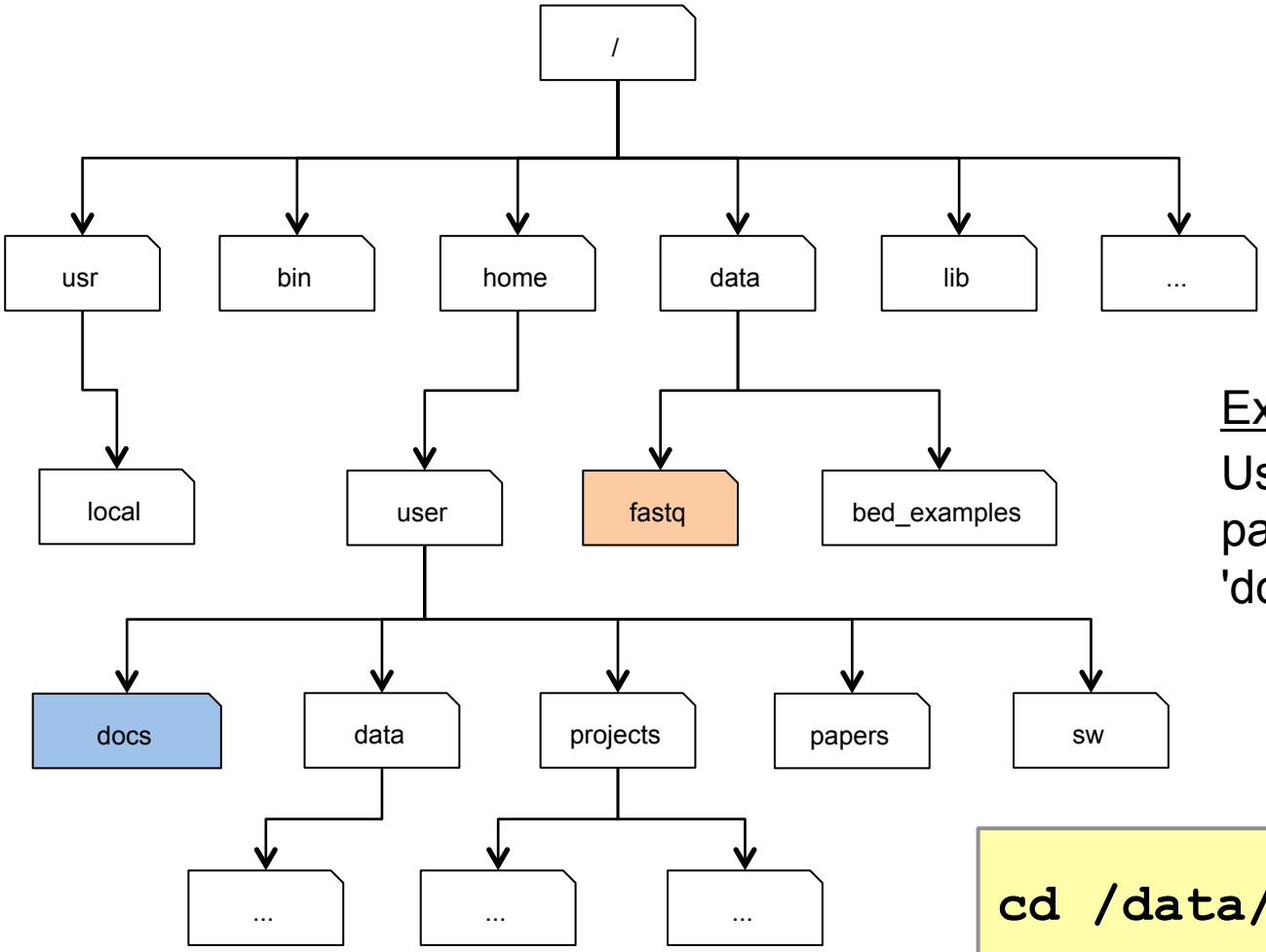
Absolute vs. Relative Path



Exercise:

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

Absolute vs. Relative Path

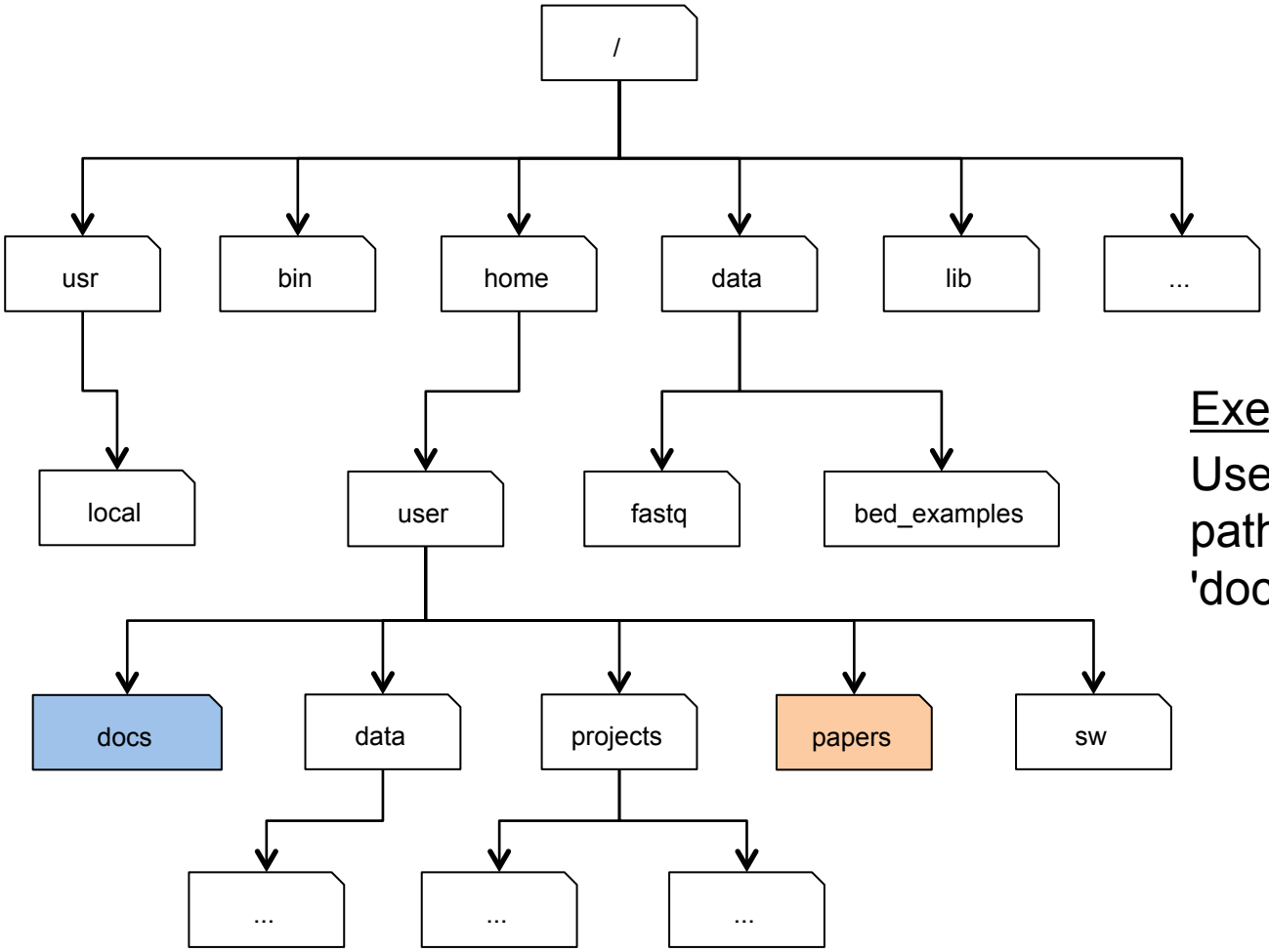


Exercise:

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

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

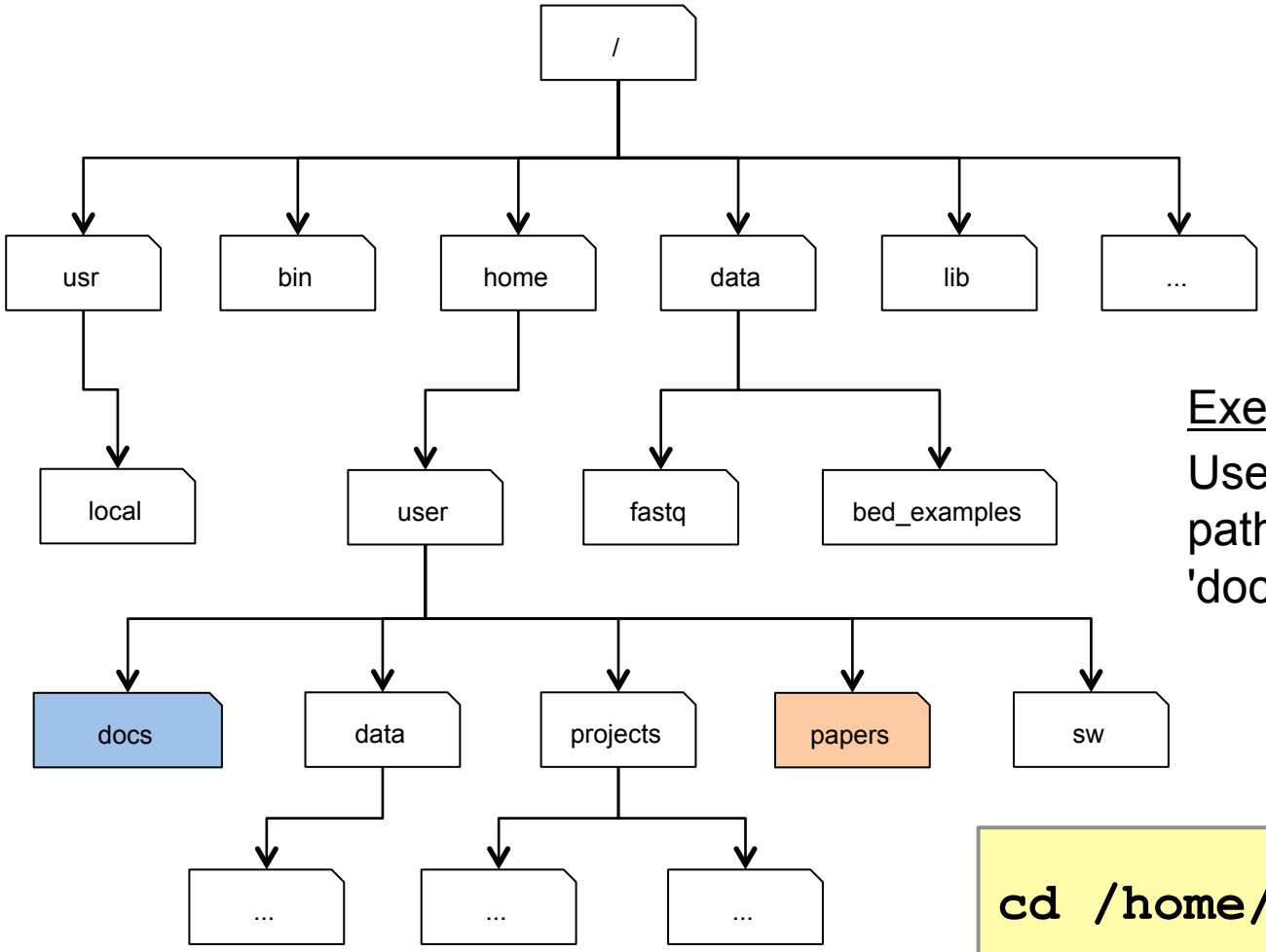
Absolute vs. Relative Path



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
```

Directory content and its size, disc usage

```
cd /data

ls -shaR # list content of a directory
du -sh # disc usage (by directory)
df -h # disc free
ls | wc -l # what does this command do?
locate # find a file/program
```

Moving/coping files/directories

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

Moving/coping files/directories - Exercise

Try these tools to:

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

```
touch # make empty file
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 - Exercise

Prepare FASTQ data:

```
cd ~
mkdir -p data/fastq
sudo cp -r /data/fastq/fastq.tar.gz data/fastq/.
ls data/fastq
```

Uncompressing data

```
tar -xzvf data/fastq/fastq.tar.gz # tarball
archive + gzip

gunzip file.gz # only gzipped
```

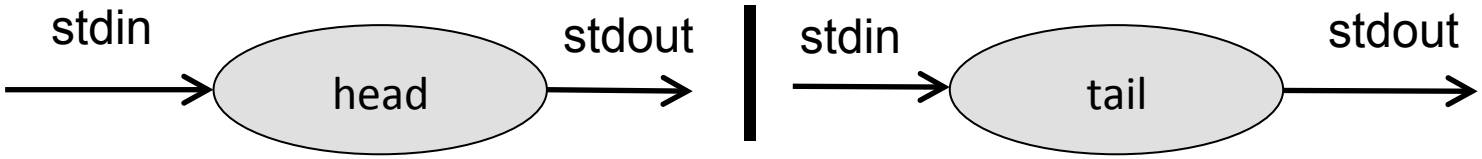
Viewing plain text file content

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

** TAB completion

Pipes '|'

Chaining standard input and output:




```
head -n 8 data/fastq/HRTMUOC01.RL12.00.fastq | \
tail -n 4 | less

< data/fastq/HRTMUOC01.RL12.00.fastq head -n8 | \
tail -n4 | less
```

Pipes '|' - Exercise

How many reads are there?

```
cd ~
cat data/fastq/HRTMUOC01.RL12.00.fastq | wc -l
expr XXXX / 4
```

Globbing

What if I need to choose multiple files?

```
cd ~
cat data/fastq/HRTMUOC01.RL12.*.fastq | wc -l
expr XXXX / 4

cat data/fastq/HRTMUOC01.RL12.0?.fastq | wc -l
expr XXXX / 4
```

Producing Lists

```
cd ~
mkdir data2 && cd data2
touch file-{1..12}.txt
ls
```

Producing Lists

Try these examples below:

```
touch file-0{1..9}.txt file-{10..20}.txt

touch 0{1..9}-{a..f}.txt {10..20}-{a..f}.txt

touch 0{1..9}-{jan,feb,mar}.txt {10..20}-
{jan,feb,mar}.txt
```

Variables

Variable: storage location paired with an associated symbolic name

```
CPU=4
echo $CPU

FILE=data/fastq/HRTMUOC01.RL12.00.fastq
echo $FILE

FILES=`ls data/fastq/*.fastq`
echo $FILES
```

** echo '$FILE'

Loops

Repeat a command (set of commands) multiple times:

```
FILE=`ls data/fastq/*.fastq`

for i in $FILE
do
   echo $i
done
```

Loops

Repeat a command (set of commands) multiple times:

```
FILE=`ls data/fastq/*.fastq`

for i in $FILE
do
   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...