

Fondation
Campus
Biotech
Geneva
+

Neuroimaging data science course

Michael Dayan, Data Scientist Manager

Methods & Data facility
Human Neuroscience Platform
Foundation Campus Biotech Geneva

Virtual machine info

WIFI for the Virtual Machine (VM) [no phone !]

- SSID: NIDS_course
- Password: reproduciblescience

START RDP CLIENT (as instructed in email / Slack):

- *Remote Desktop Connection* on Windows
- *Remote Desktop App* on Mac OS
- *Remmina* on Linux distributions (e.g. Ubuntu)

→ To get an IP, please fill the form at:

<https://tinyurl.com/IORDS2021-IP-linux1>

ANY PROBLEM? Please raise your hand or ask questions
on Slack: channel #linux

WIFI for your phone or tablets

- SSID: CAMPUS_VISITORS
- Password: welcomecampus

PLEASE CONNECT TO THE VM

- Login: brainhacker
- Password: brainhack!



GENEVA UNIVERSITY NEUROCENTER



LECTURE OBJECTIVES

Linux lectures objectives:

- Be comfortable with Linux & the command line
- Understand the notion of kernel, shell and command line
- Understand the linux file system structure
- Know how to navigate the file system
- Know how to create, delete and interact with (text) files
- Understand file permissions and know how to modify them
- Learn how to write shell scripts
 - Variables
 - For loop
 - More control flow statements (if-else statements, etc.)
 - Define and use user-provided arguments
 - Manipulate string variables

→ Write your own script to automate specific operations, which can be used by others

Linux Part 1

Linux Part 2

Linux Part 3

HISTORY OF UNIX / LINUX

UNiplexed

→ MULTICS: MUXplexed Information
and Computing Service

→ 1969: UNICS



Ken Thompson &
Dennis Ritchie



HISTORY OF UNIX / LINUX

UNiplexed

→ MULTICS: MUXplexed Information
and Computing Service

→ 1969: UNICS

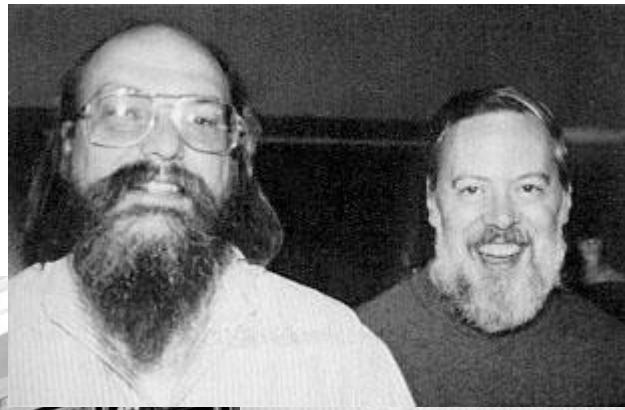


cat (concatenate files); cd (change working directory); cp (copy file); ls (list directory contents); mv (move or rename file); wc (get word count)

"The most user-hostile editor ever created" P. H. Salus



Ken Thompson &
Dennis Ritchie



HISTORY OF UNIX / LINUX

UNiplexed

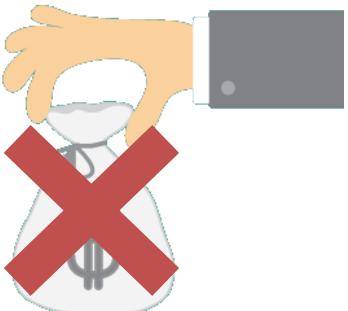
→ MULTICS: MUXplexed Information
and Computing Service

→ 1969: UNICS



The UNIX Time-Sharing System

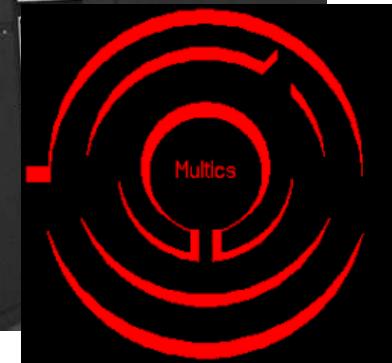
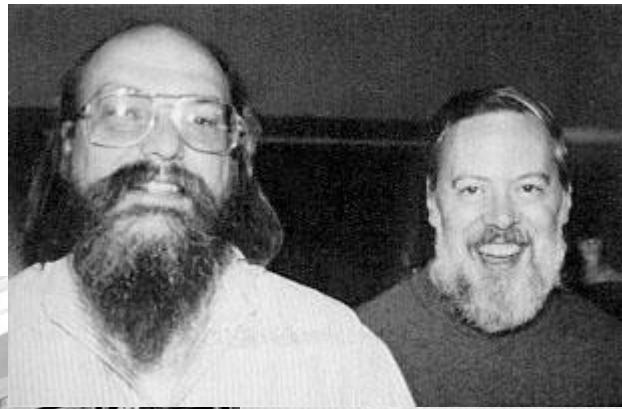
Dennis M. Ritchie and Ken Thompson
Bell Laboratories



Symposium of the Association for Computing Machinery, 1974



Ken Thompson &
Dennis Ritchie



HISTORY OF UNIX / LINUX

UNiplexed

→ MULTICS: MUXplexed Information and Computing Service



→ 1969: UNICS

→ 1977: Berkeley Unix
Berkeley Software Distribution (BSD)

[→ NeXTSTEP → MacOS]



AT&T

→ 1983: GNU's Not Unix (GNU)
General Public License (GPL)



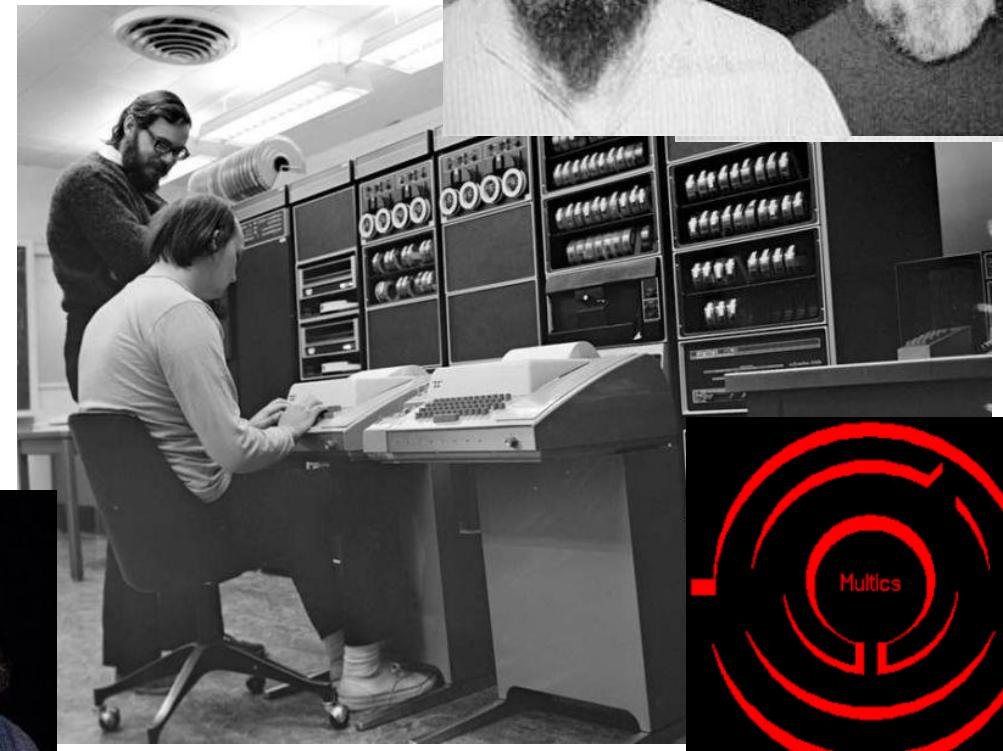
→ 1987: Mini-Unix (MINIX)

→ 1991: Linux

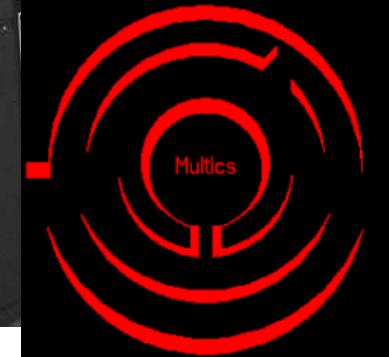


Linus Torvalds

Ken Thompson & Dennis Ritchie



Fondation
Campus
Biotech
Geneva
+

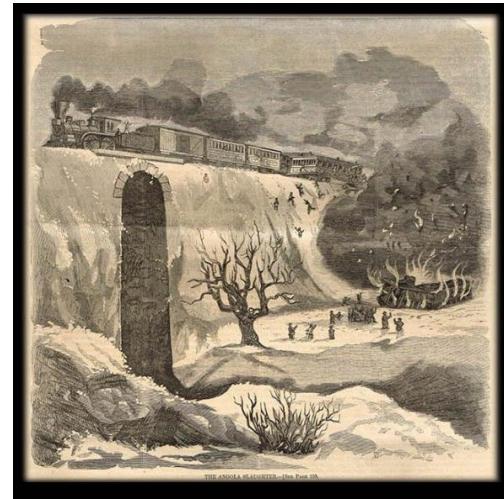


POPULARITY OF UNIX / LINUX

Linux is everywhere (servers, PCs, mobile phones, tablets), and the % of top 500 supercomputers using Linux is... 100%

- Strong community which supports OPEN software [GPL]
- Modular and interoperable approach via POSIX standards (FAIR principles)
- REPRODUCIBLE commands via the command-line

Angola horror railway disaster, December, 1867

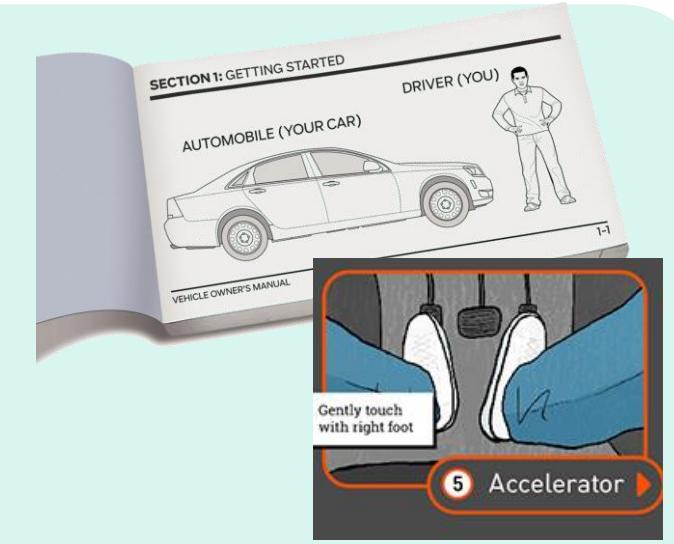


Microsoft CEO
in 2001



Microsoft CEO
now



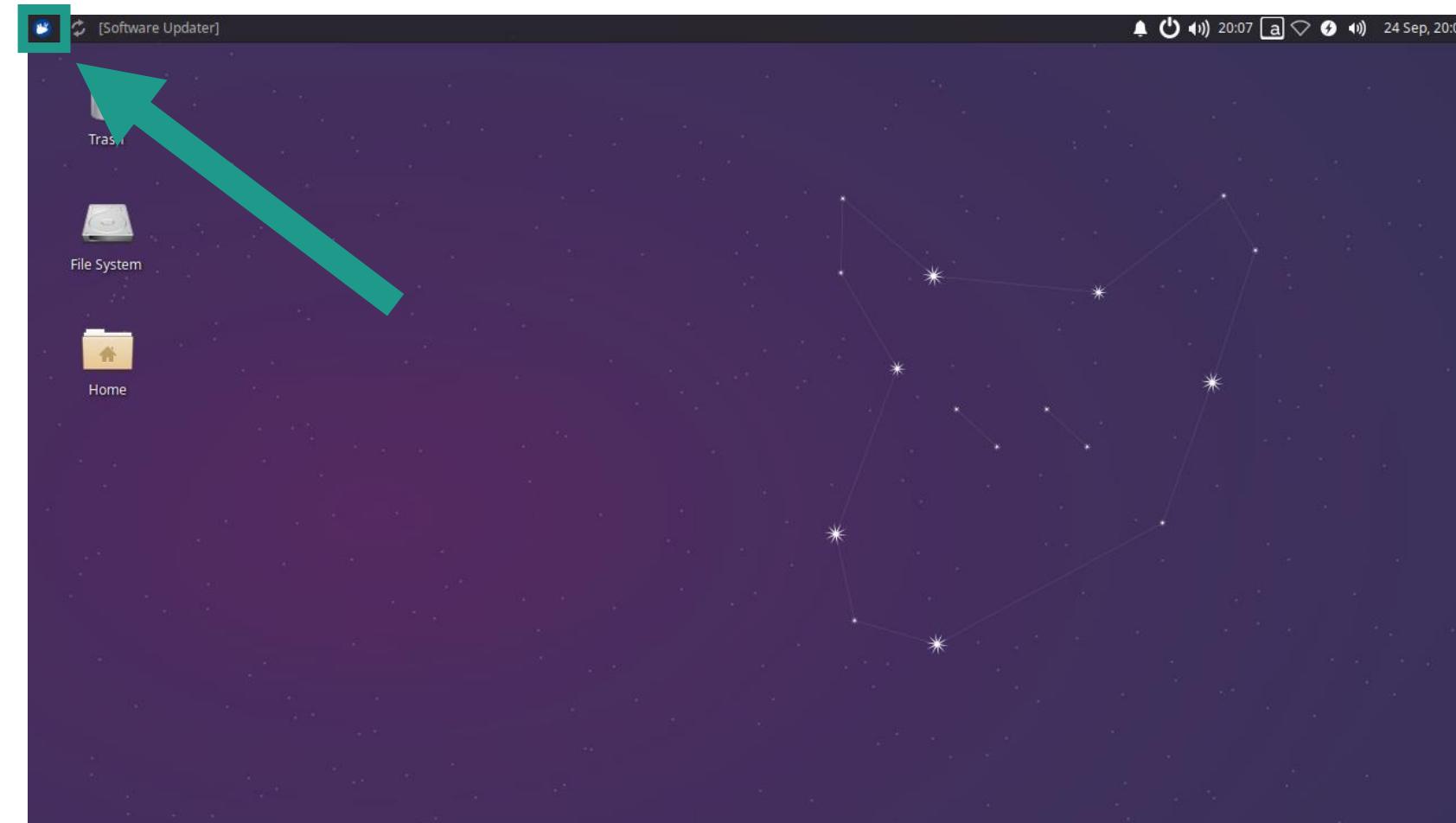


GNU Shell & Utilities
(command interpreter [user → OS] &
set of useful commands)

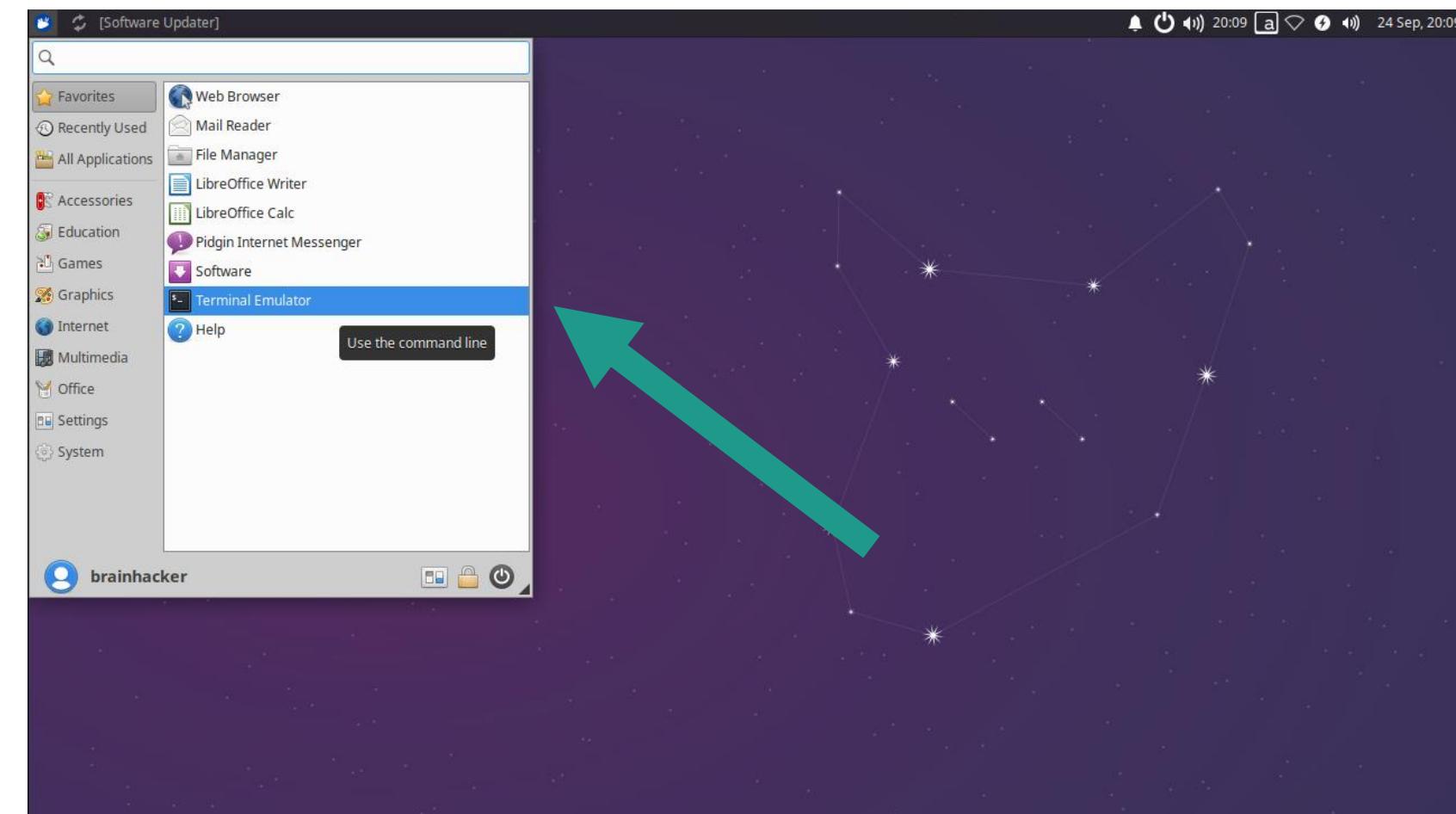
LINUX Kernel
(control computer resources & schedule jobs)

Hardware (devices)

LINUX OS, SHELL & TERMINAL



LINUX OS, SHELL & TERMINAL



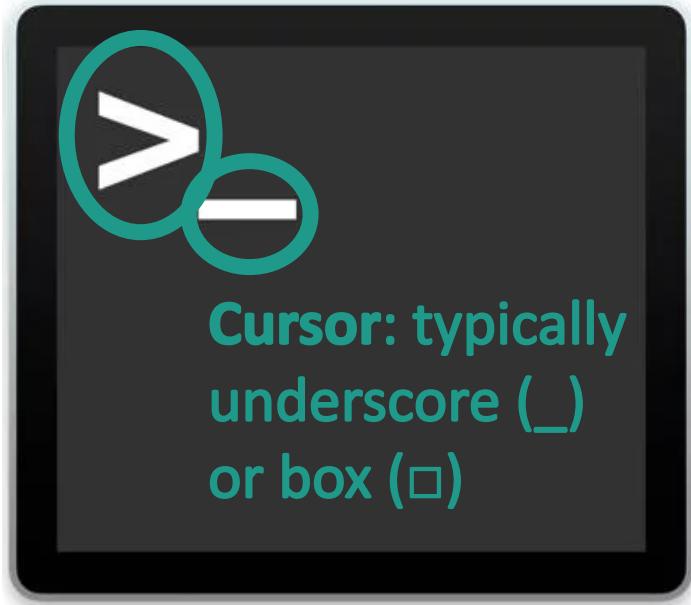
OR

Ctrl + Alt + T

A screenshot of a terminal window showing a command-line interface. The screen is filled with numerous lines of green text, which appear to be command-line output or log files. The text is too dense to read individual lines but shows typical command-line syntax.

TERMINAL & COMMAND LINE

Prompt: typically greater sign (>),
dollar sign (\$) or hashmark (#)



> **Command**  Enter ↲
> **fortune**  Enter ↲

To erase what you just typed on
the command line:

Ctrl + **U**

> **cmatrix**  Enter ↲

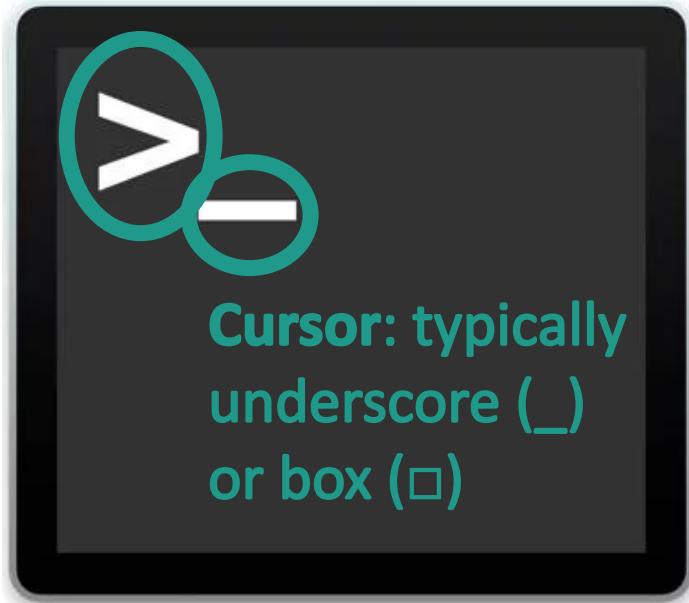
Ctrl + **C**

**KEEP
CALM
AND
PRESS
CTRL C**



TERMINAL & COMMAND LINE

Prompt: typically greater sign (>),
dollar sign (\$) or hashmark (#)

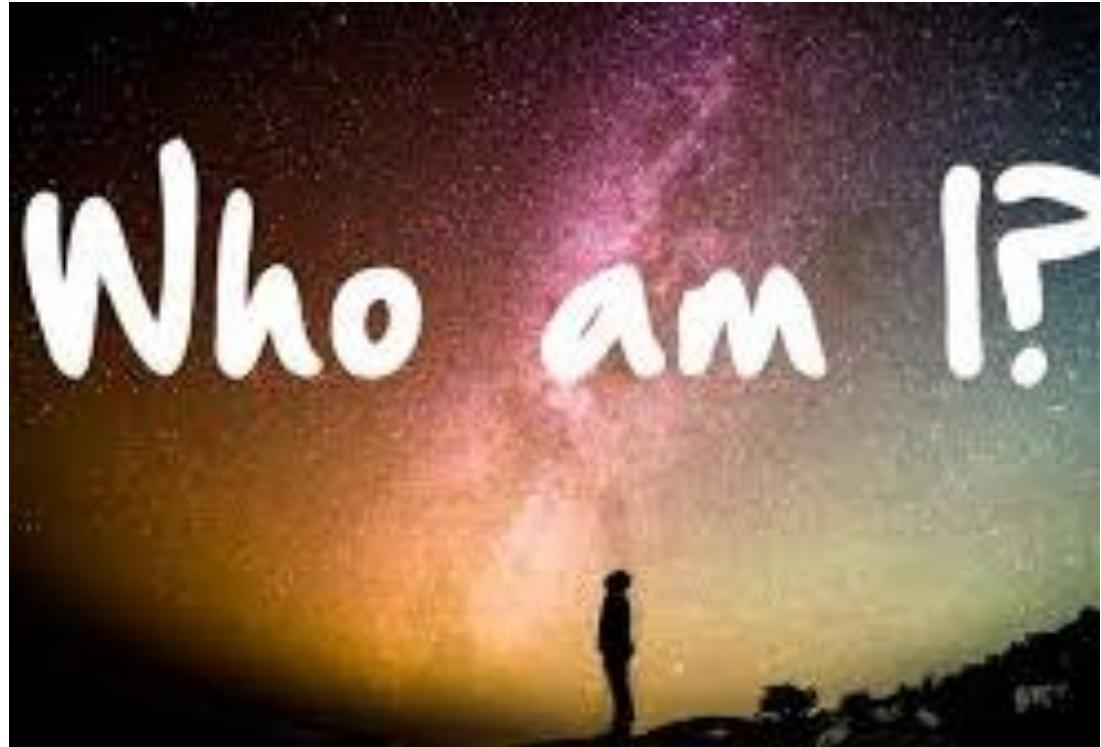


> *command* 
> *fortune* 

→ Where am i?
→ Who am i?



TERMINAL & COMMAND LINE



> whoami



> whe~~x~~ami

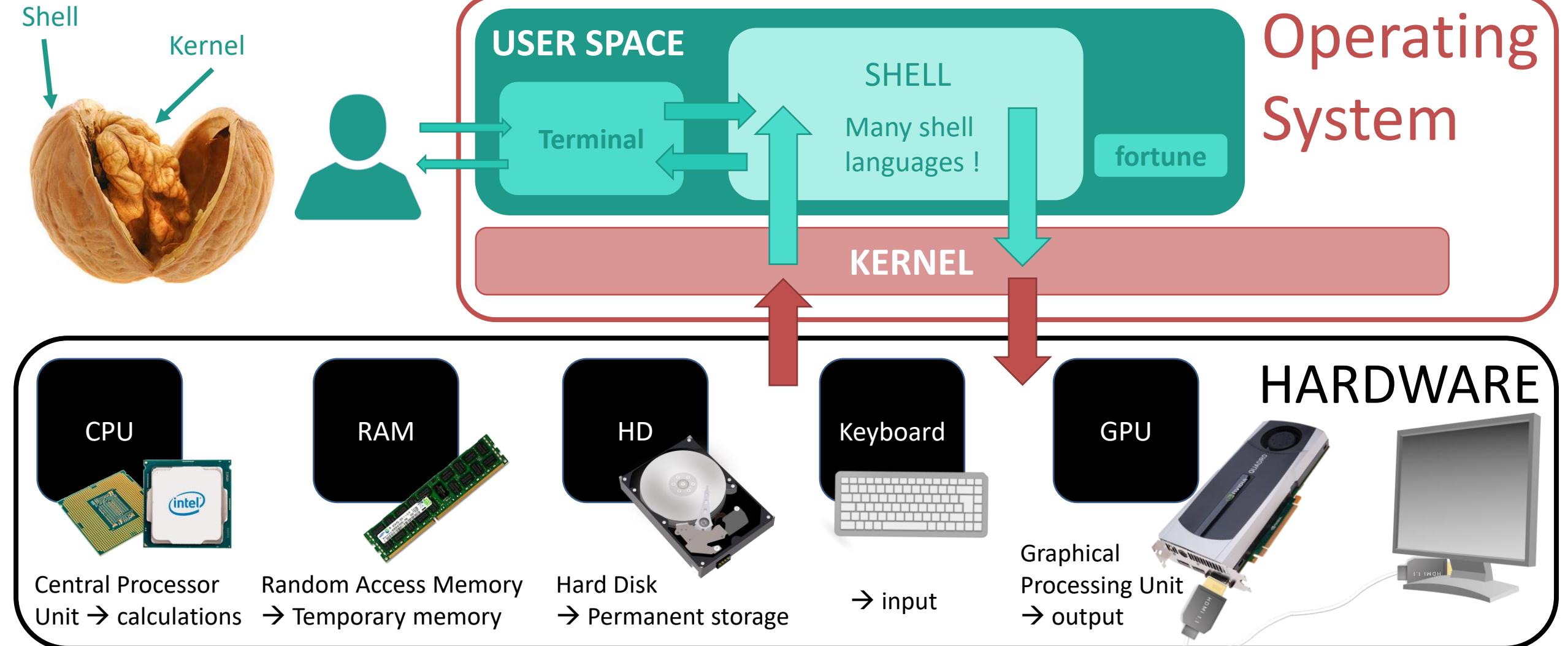
> pwd
(print working
directory)

Computer name
> hostname

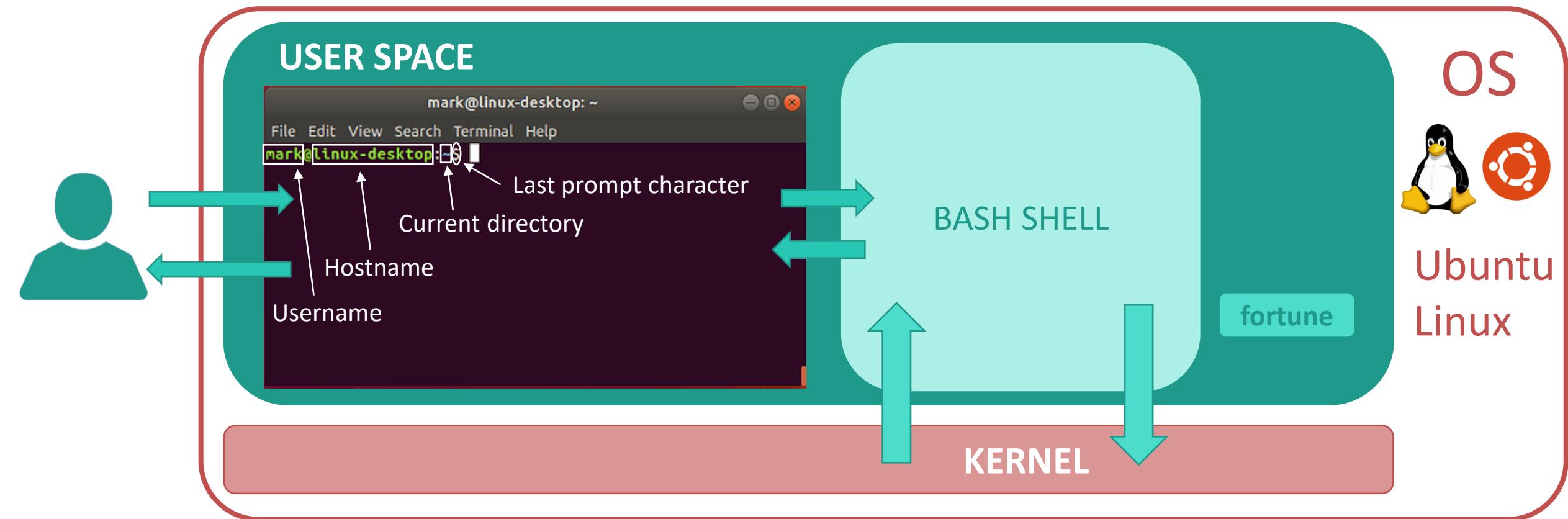
Very useful on a network



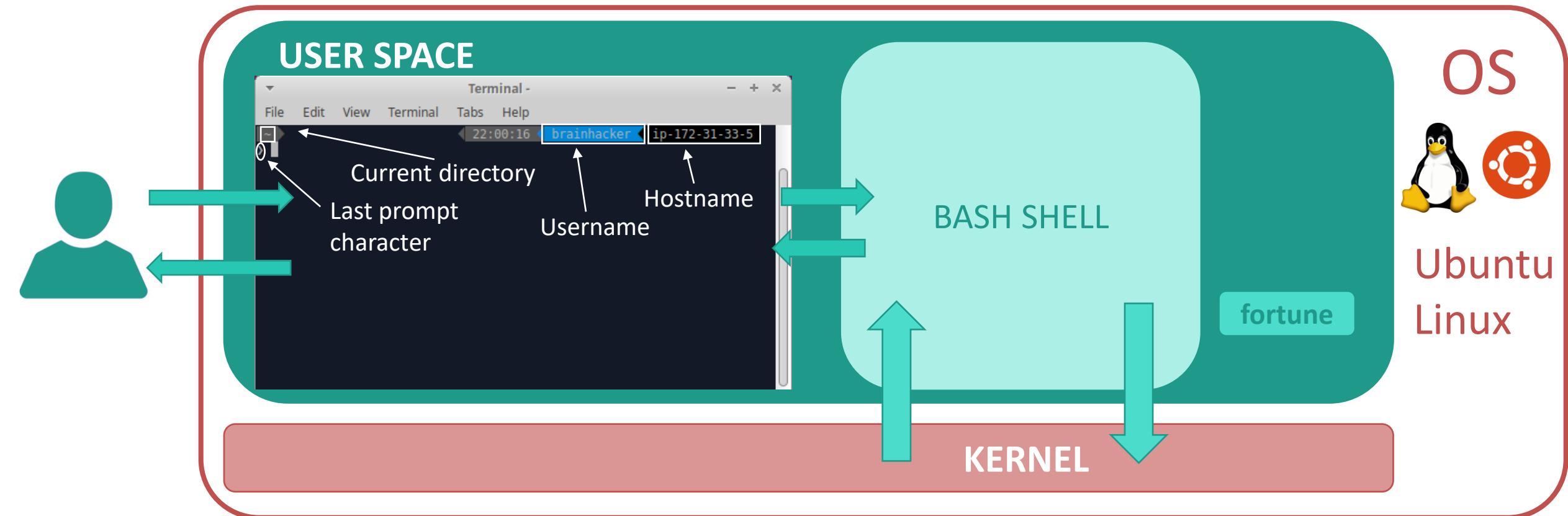
OS, SHELL & TERMINAL



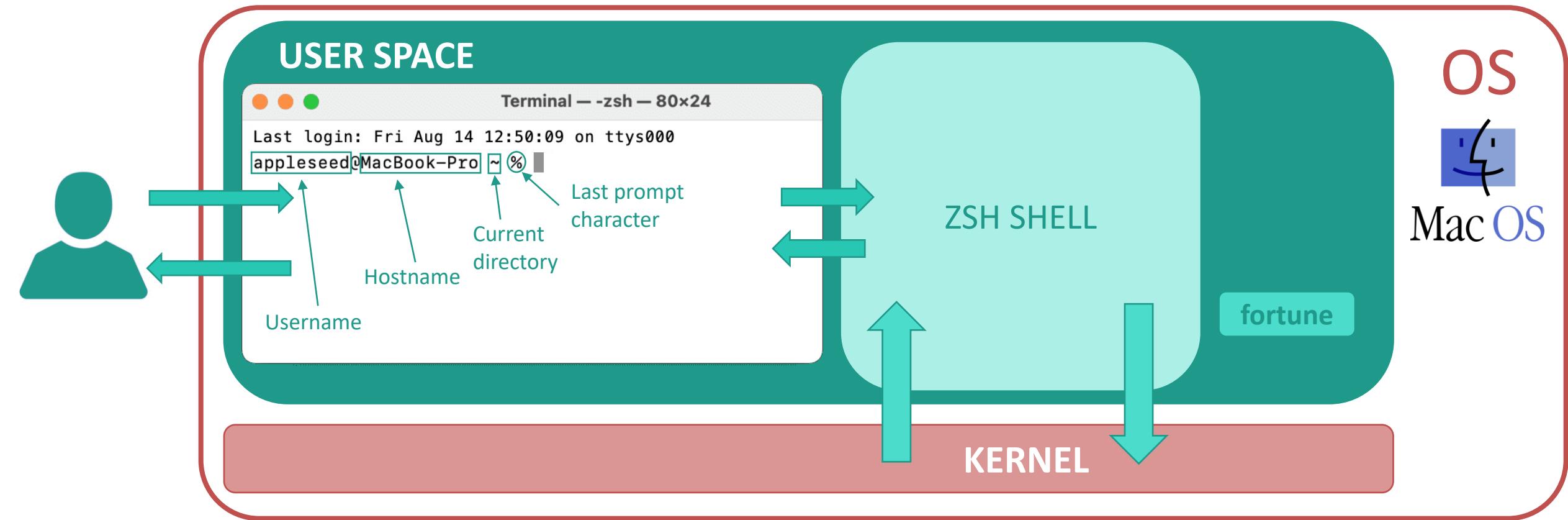
LINUX OS, SHELL & TERMINAL



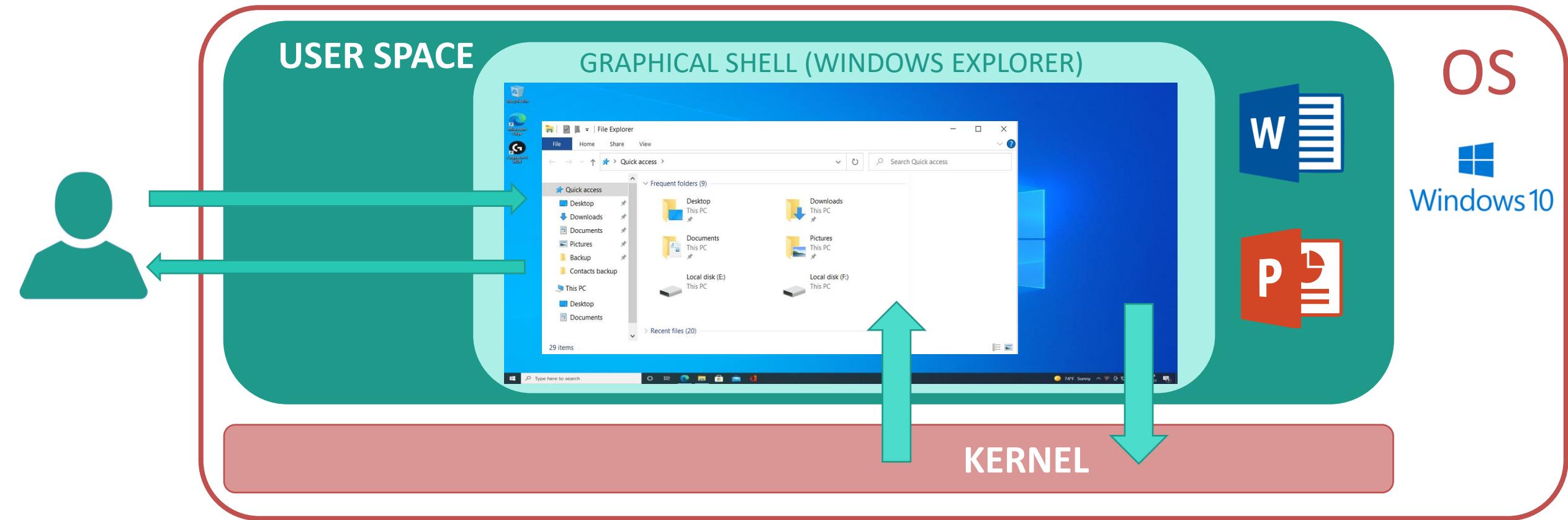
LINUX OS, SHELL & TERMINAL



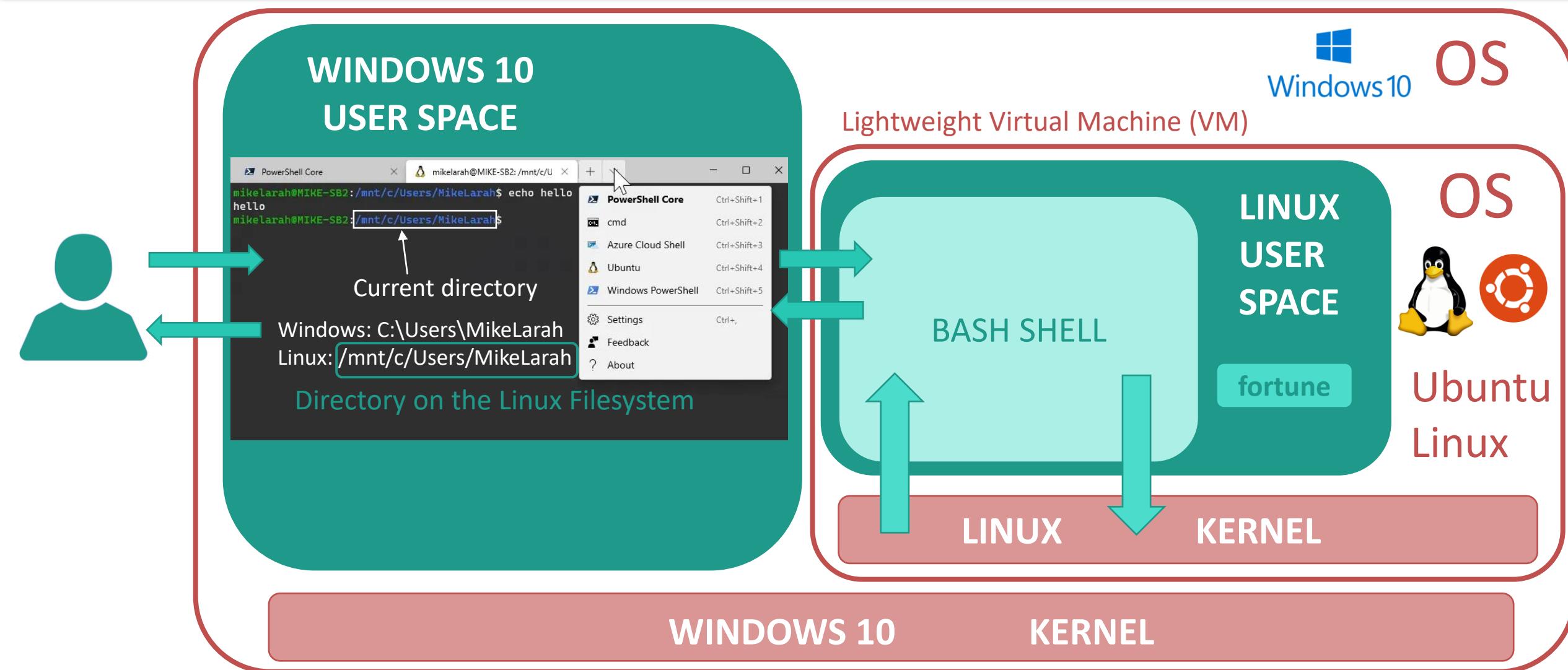
MAC OS, SHELL & TERMINAL



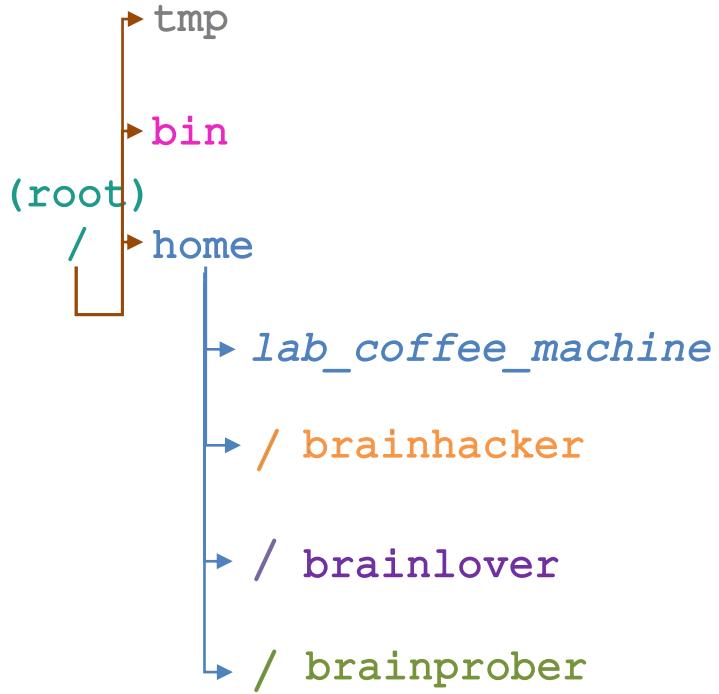
WINDOWS OS & GRAPHICAL SHELL



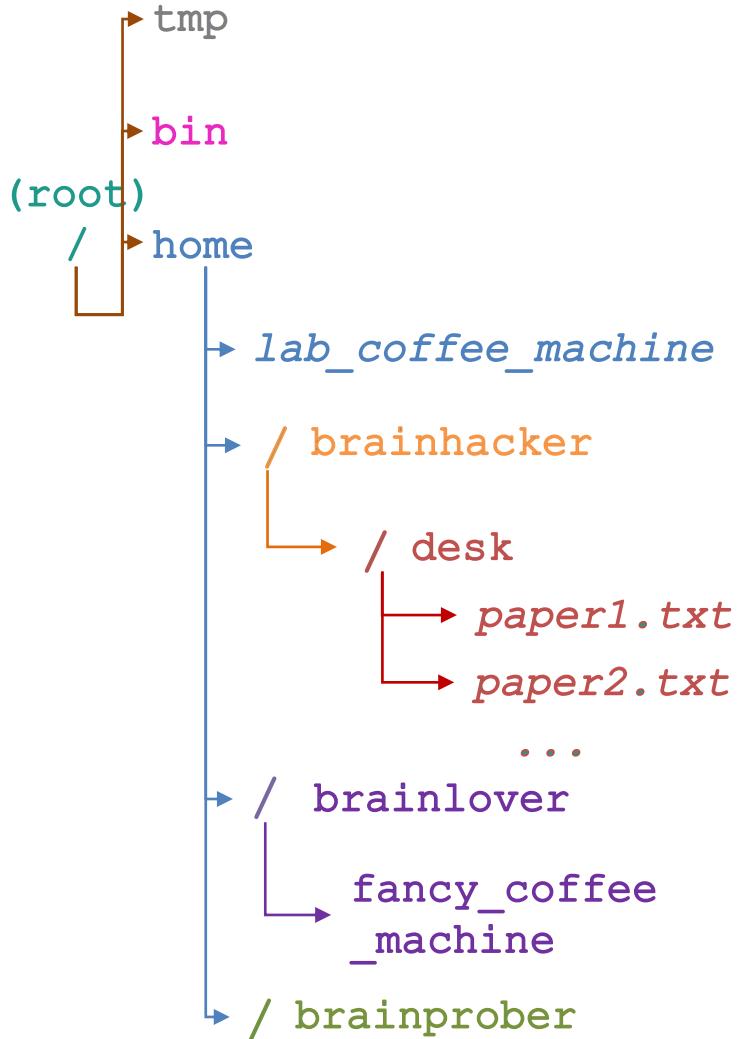
WINDOWS SUBSYSTEM FOR LINUX (WSL) 2



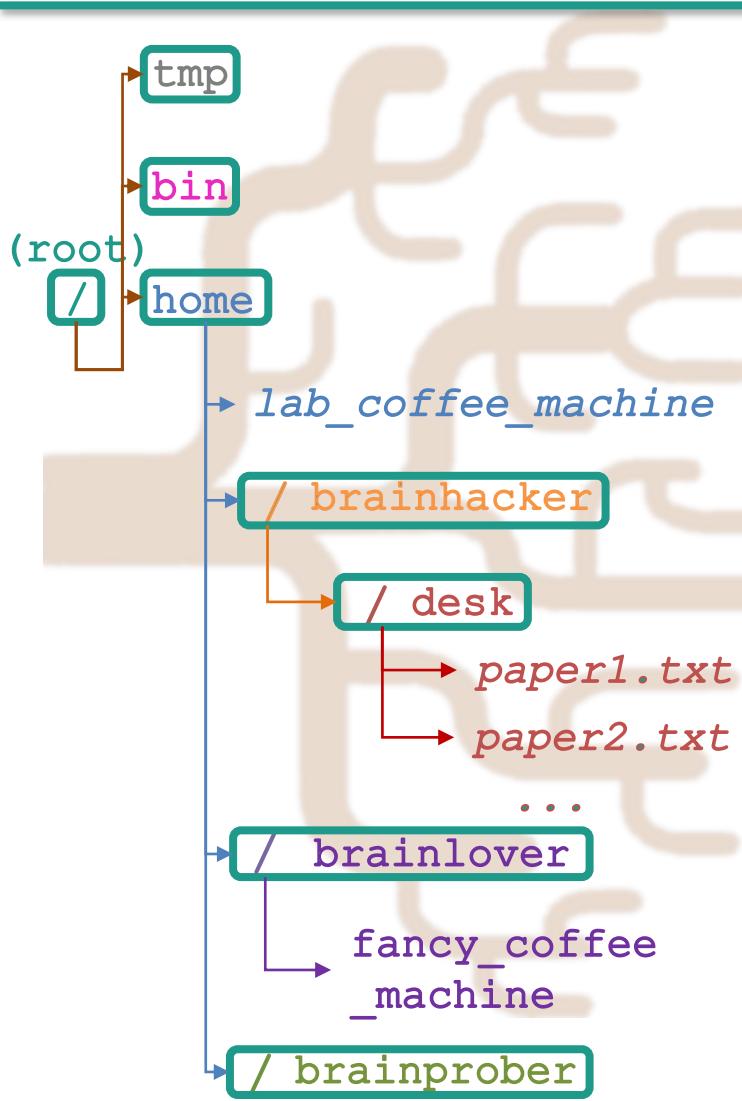
LINUX FILESYSTEM



LINUX FILESYSTEM

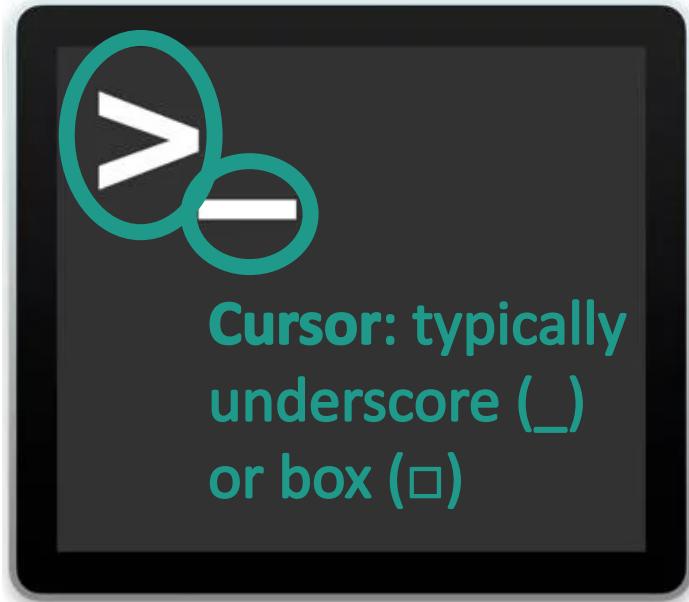


LINUX FILESYSTEM



BASH SHELL (command line interpreter)

Prompt: typically greater sign (>), dollar sign (\$) or hashmark (#)



Cursor: typically underscore (_) or box (□)

> command arg1 arg2 ...  

A new line is a special character: “\n” (\r\n in win)

Haikus are easy
But sometimes they don't make sense

Refrigerator

Haikus are easy 

But sometimes they don't make sense 

Refrigerator

TASK

Print on screen:

One

Two

Three

> command  

BASH SHELL (command line interpreter)

echo "One" argument to echo

echo "Two"

move cursor left/right

↑ back in history

↓ forward in history

← → ← Backspace delete character(s)

echo "Three"

echo "One\nTwo\nThree"

Space bar: display more, q: quit

man echo

echo -e "One\nTwo\nThree"

arguments to echo

history #to see all past commands

> command arg1 arg2 ... 

A new line is a special character: "\n" (\r\n in win)

Haikus are easy
But sometimes they don't make sense

Refrigerator

Haikus are easy 
But sometimes they don't make sense 
Refrigerator

TASK

Print on screen:
One
Two
Three

To display manual about a command:
man command_name

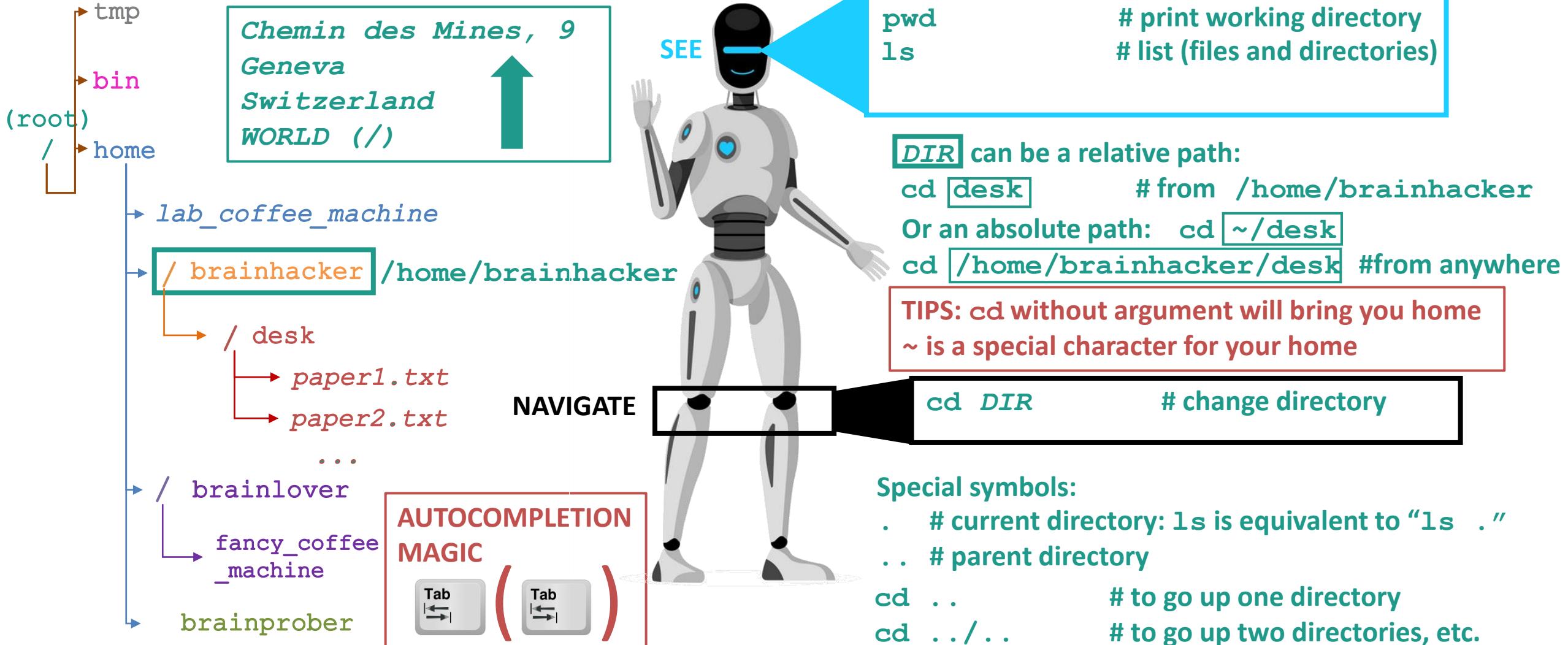
tldr for quick ex.

To search manual for a keyword:
man -k keyword_name

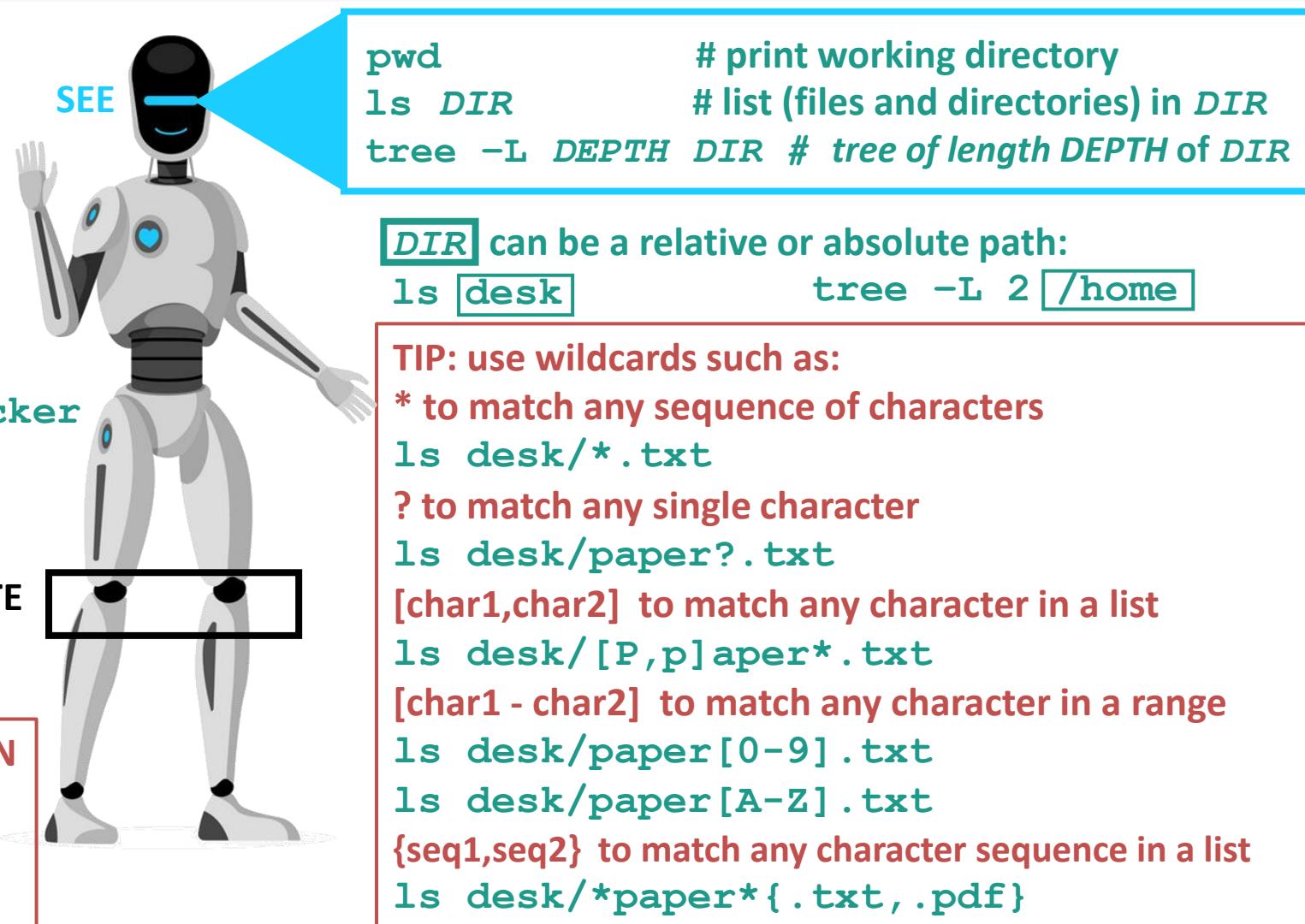
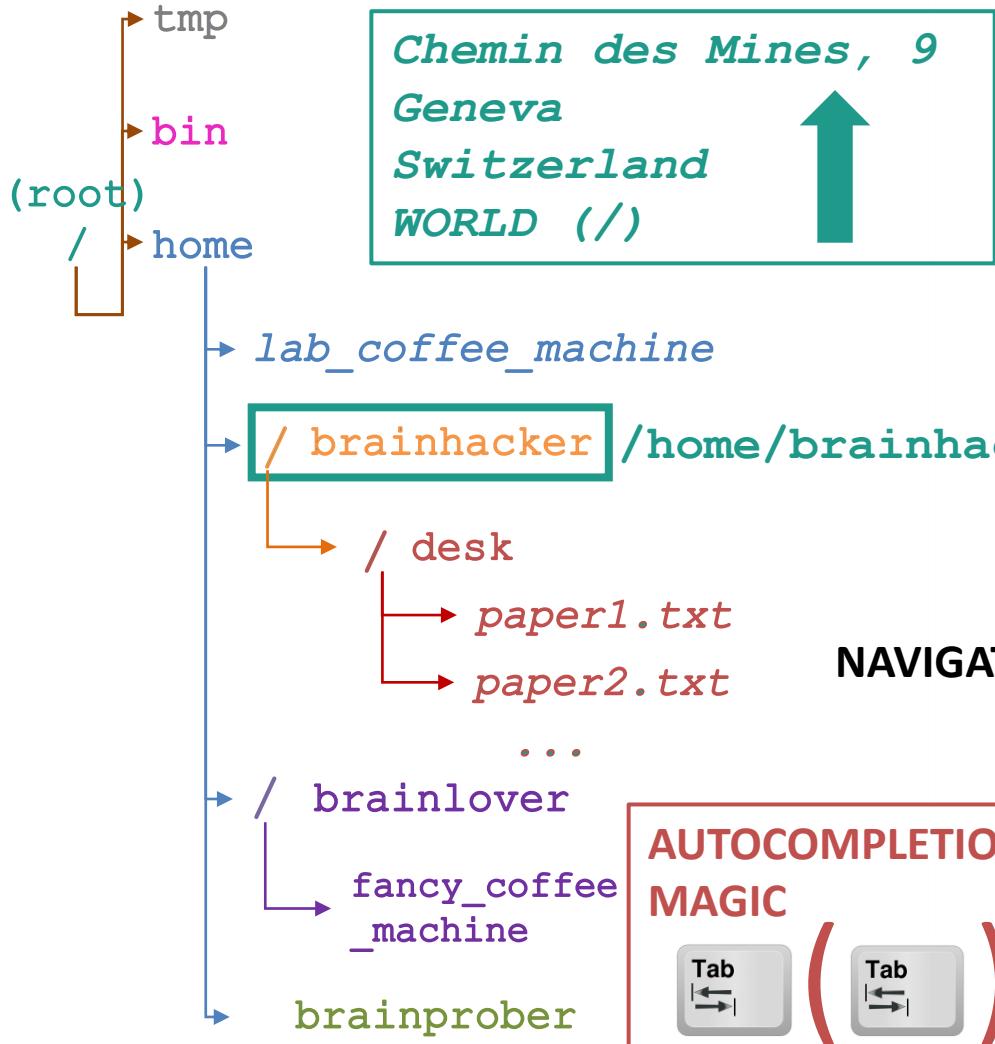
man -k who



LINUX FILESYSTEM



LINUX FILESYSTEM



Find all papers and abstracts
(with only one BASH command)

Find all numbered notepads
(only those ending with numbers)



LINUX FILESYSTEM

cd *DIR*

change directory

TASKS

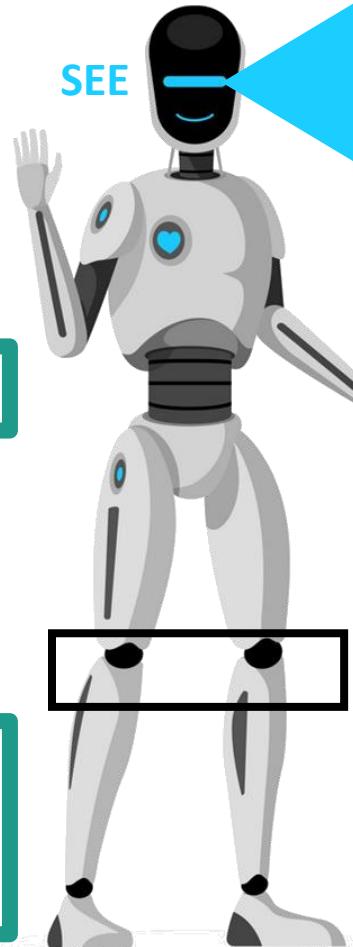
Go to the `edu_ranking` directory in your home and print the content of `timesData.csv` with `cat`

Do the same with `more` (press q to QUIT)

Find the column names with `head`
Bonus: fine the “female/male ratio” of the 8th and 9th best ranked universities

Find the name of the last but one university (i.e. end - 1) using `tail`
Note: last ones are in the 601-800 group

Using `grep`, check if some familiar Swiss universities are there (TIP: use the name of the cities they belong to)



`pwd` # print working directory
`ls DIR` # list (files and directories) in *DIR*
`cat FILE` # print *FILE* to standard output
(can concatenate files, cf. later)
`more FILE` # view *FILE* (press SPACE for more)
`less FILE` # similar but can also go backward
(PAGE UP and PAGE DOWN keys)
`head FILE` # view first 10 lines of *FILE*
`head -n X FILE` # view first *X* lines
`tail FILE` # view last 10 lines
`tail -n X FILE` # view last *X* lines
`grep STRING FILE` # find *STRING* in *FILE*
`wc FILE` # count of *FILE* chars, words, lines
`wc -l FILE` # count of lines in *FILE*

How many lines in `timesData.csv` ?



LINUX FILESYSTEM



"Write programs that do one thing and do it well. Write programs to work together."

Programs can be combined with the pipe operator: |

```
cowsay "hello"  
fortune | cowsay
```

TASKS

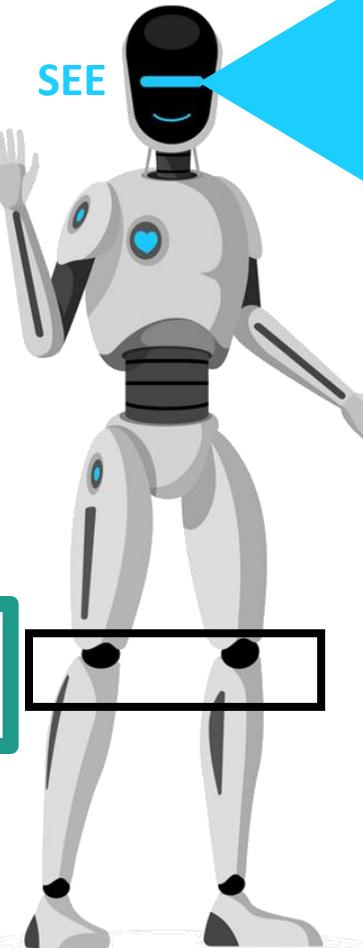
using `~/edu_ranking/timesData.csv`

Output the top 15 universities across all years
(repeated names for different years OK)

TIP: first sort the data, then select the first 15 lines
(solution can have one or two pipe operators)

```
sort -n timesData.csv | head -n 15
```

```
cat timesData.csv | sort -n | head -n 15
```



SEE

<code>pwd</code>	# print working directory
<code>ls DIR</code>	# list (files and directories) in <i>DIR</i>
<code>cat FILE</code>	# print <i>FILE</i> to standard output
<code>more FILE</code>	# (can concatenate files, cf. later)
<code>less FILE</code>	# view <i>FILE</i> (press SPACE for more)
<code>head FILE</code>	# similar but can also go backward
<code>head -n X FILE</code>	# (PAGE UP and PAGE DOWN keys)
<code>tail FILE</code>	# view first 10 lines of <i>FILE</i>
<code>tail -n X FILE</code>	# view first <i>X</i> lines
<code>tail -n X FILE</code>	# view last 10 lines
<code>grep STRING FILE</code>	# view last <i>X</i> lines
<code>wc FILE</code>	# find <i>STRING</i> in <i>FILE</i>
<code>wc -l FILE</code>	# count of <i>FILE</i> chars, words, lines
<code>wc -l FILES</code>	# count of lines in <i>FILE</i>
<code>sort FILE</code>	# count of lines in all <i>FILES</i>
<code>sort -n FILE</code>	# sort <i>FILE</i> by alphabetical order
<code>sort -n FILE</code>	# sort <i>FILE</i> by numerical value



LINUX FILESYSTEM (HOMEWORK)



"Write programs that do one thing and do it well. Write programs to work together."

Programs can be combined with the pipe operator: |

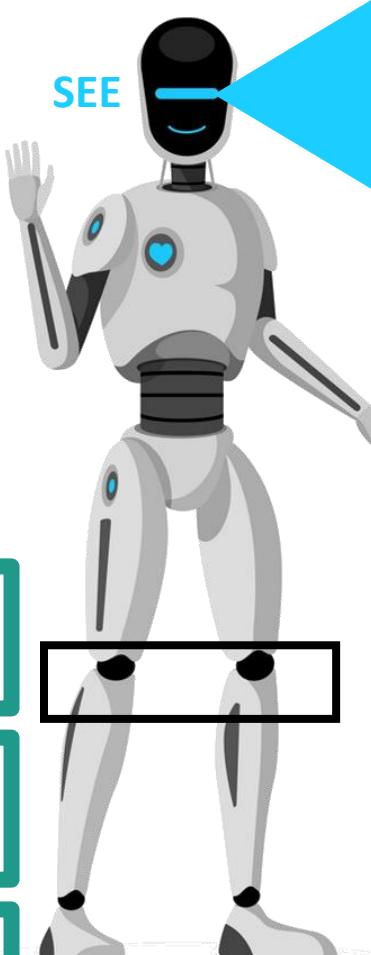
```
cowsay "hello"  
fortune | cowsay
```

TASKS starting in ~ /inflammation_data

Count number of lines in one inflammation CSV file of your choice

Using wildcards, print number of lines in all inflammation CSV files in a single command

Use pipe operator(s) with the previous command to find the 3 smallest CSV files



pwd	# print working directory
ls <i>DIR</i>	# list (files and directories) in <i>DIR</i>
cat <i>FILE</i>	# print <i>FILE</i> to standard output
	# (can concatenate files, cf. later)
more <i>FILE</i>	# view <i>FILE</i> (press SPACE for more)
less <i>FILE</i>	# similar but can also go backward # (PAGE UP and PAGE DOWN keys)
head <i>FILE</i>	# view first 10 lines of <i>FILE</i>
head -n <i>X</i> <i>FILE</i>	# view first <i>X</i> lines
tail <i>FILE</i>	# view last 10 lines
tail -n <i>X</i> <i>FILE</i>	# view last <i>X</i> lines
grep <i>STRING</i> <i>FILE</i>	# find <i>STRING</i> in <i>FILE</i>
wc <i>FILE</i>	# count of <i>FILE</i> chars, words, lines
wc -l <i>FILE</i>	# count of lines in <i>FILE</i>
wc -l <i>FILES</i>	# count of lines in all <i>FILES</i>
sort <i>FILE</i>	# sort <i>FILE</i> by alphabetical order
sort -n <i>FILE</i>	# sort <i>FILE</i> by numerical value

LINUX FILESYSTEM



"Write programs that do one thing and do it well. Write programs to work together."

Programs can be combined with the pipe operator: |

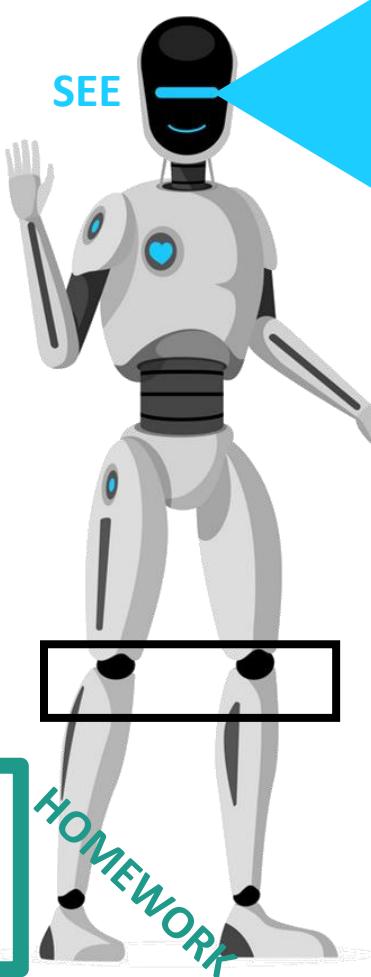
```
cowsay "hello"  
fortune | cowsay
```

Standard output can be redirected to write (>) or appended (>>) to a file:

```
echo "hello there" > ~/greetings.txt  
echo "welcome!" >> ~/greetings.txt  
cat ~/greetings.txt
```

Use output redirection to save the output of the previous homework command to a file called `smallest_groups.txt`

Bonus if you can first add the line "`n_subjects filenames`"



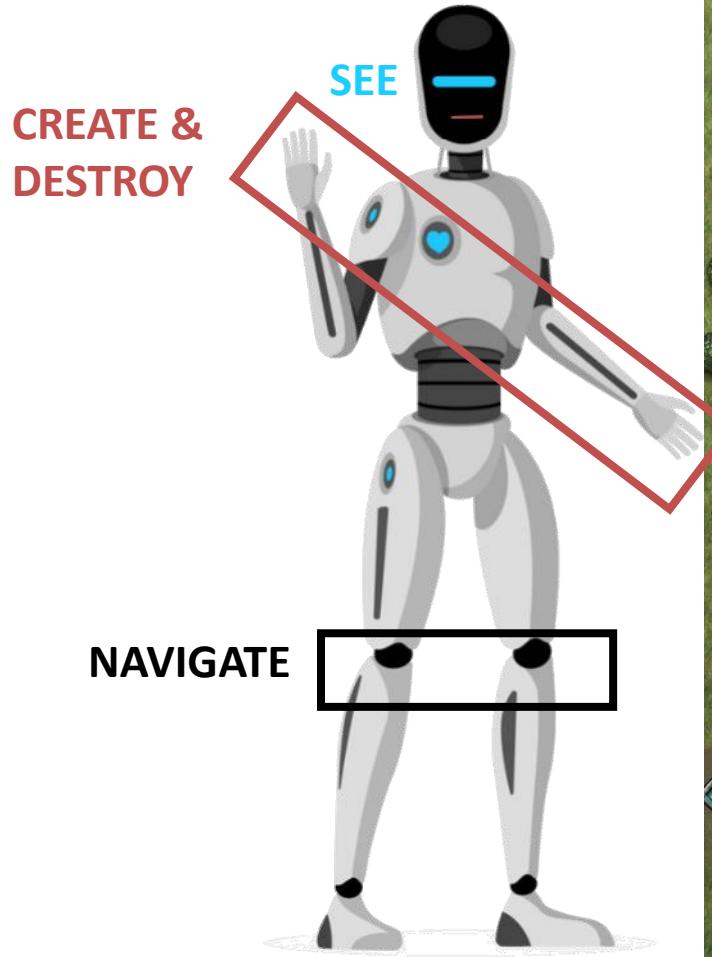
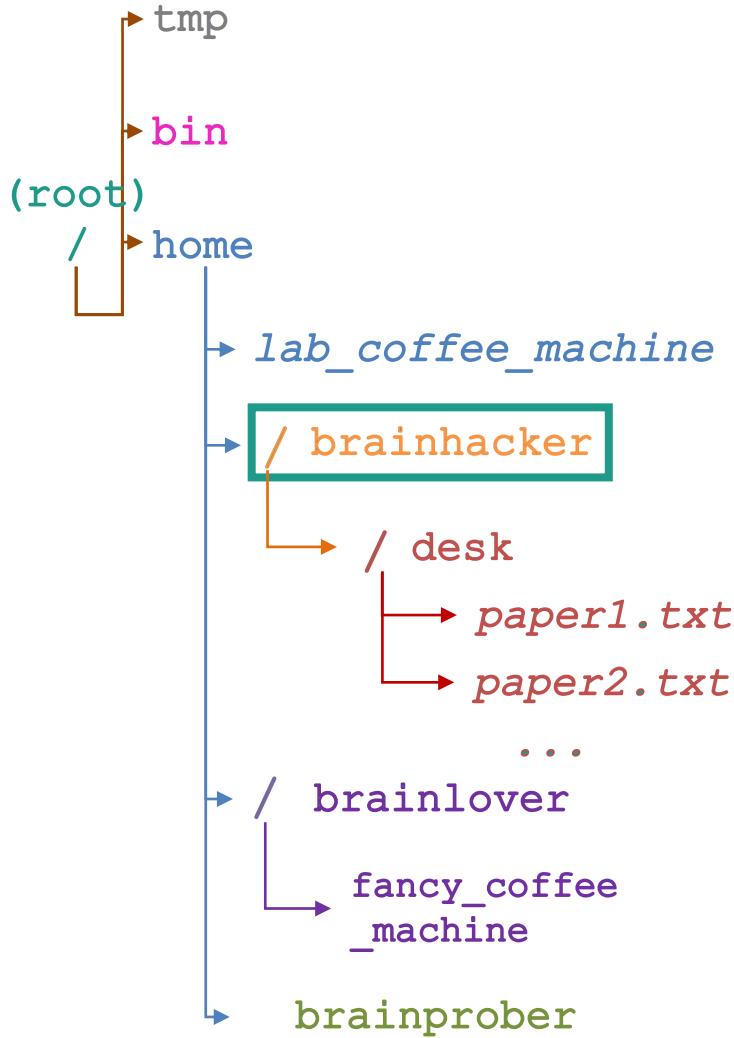
pwd
ls DIR
cat FILE

more FILE
less FILE

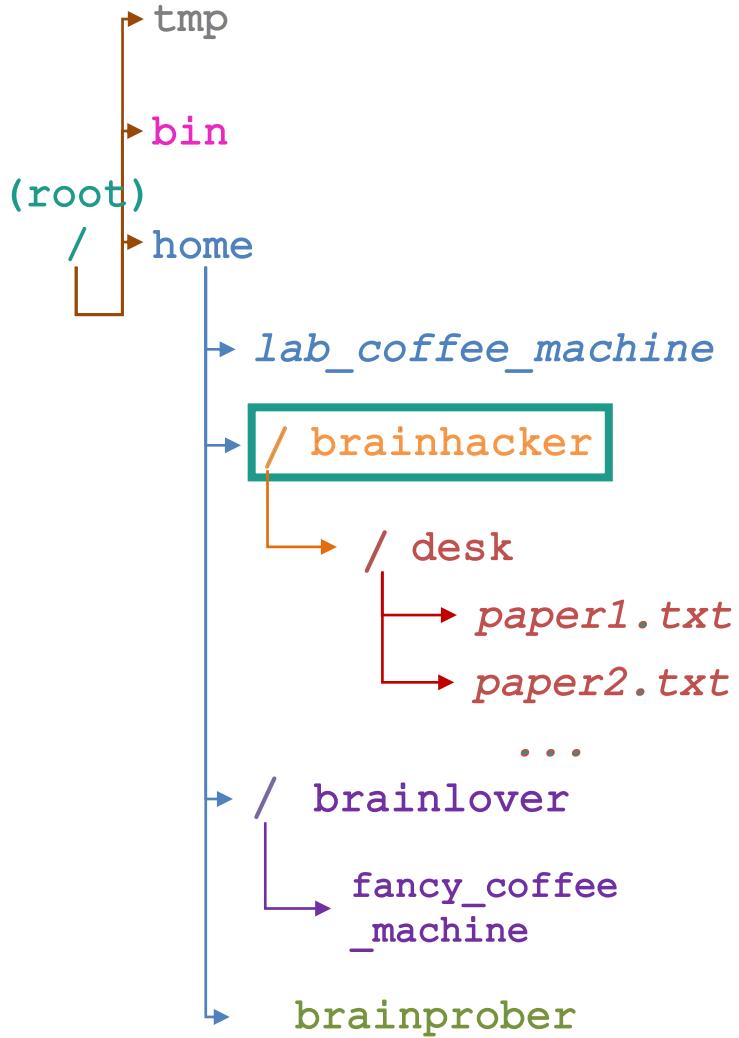
head FILE
head -n X FILE # view first x lines
tail FILE # view last 10 lines
tail -n X FILE # view last x lines
grep STRING FILE # find *STRING* in *FILE*
wc FILE # count of *FILE* chars, words, lines
wc -l FILE # count of lines in *FILE*
wc -l FILES # count of lines in all *FILES*
sort FILE # sort *FILE* by alphabetical order
sort -n FILE # sort *FILE* by numerical value



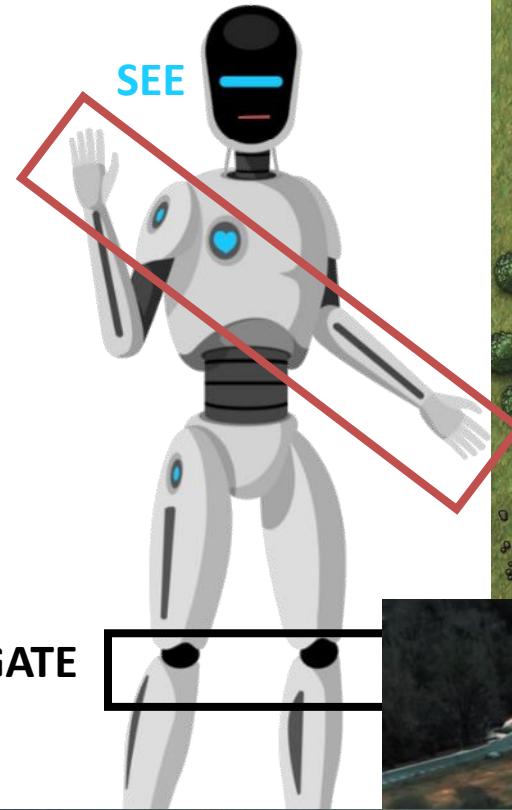
LINUX FILESYSTEM



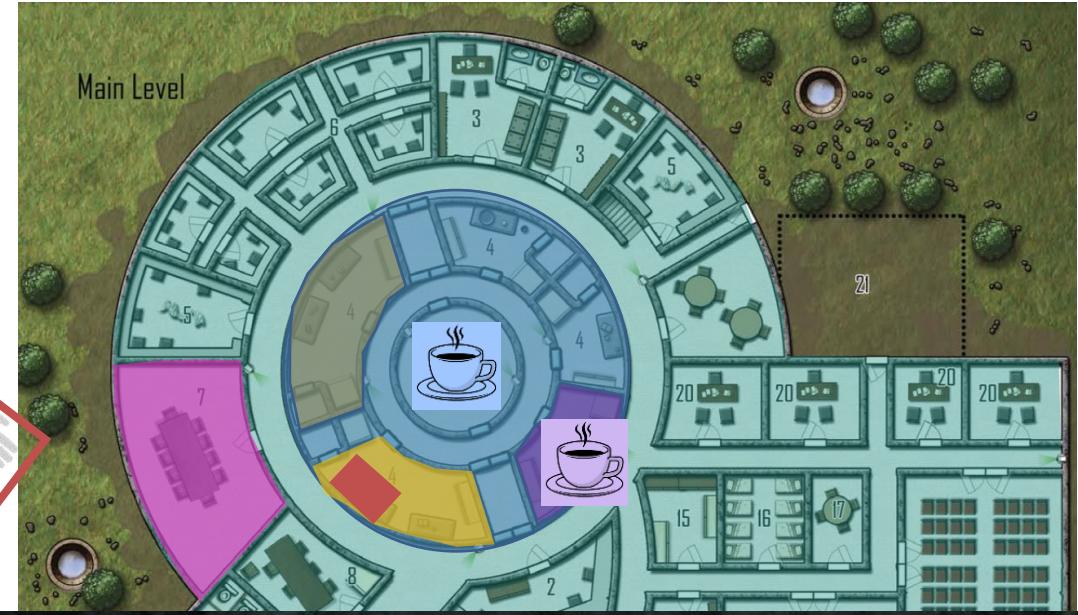
LINUX FILESYSTEM



CREATE &
DESTROY



NAVIGATE



SCP
Secure. Contain. Protect.

LEVEL 3

Giving the key card to personnel with an insufficient security clearance is strictly forbidden

SCP
Secure. Contain. Protect.

LEVEL 4

Giving the key card to personnel with an insufficient security clearance is strictly forbidden

SCP
Secure. Contain. Protect.

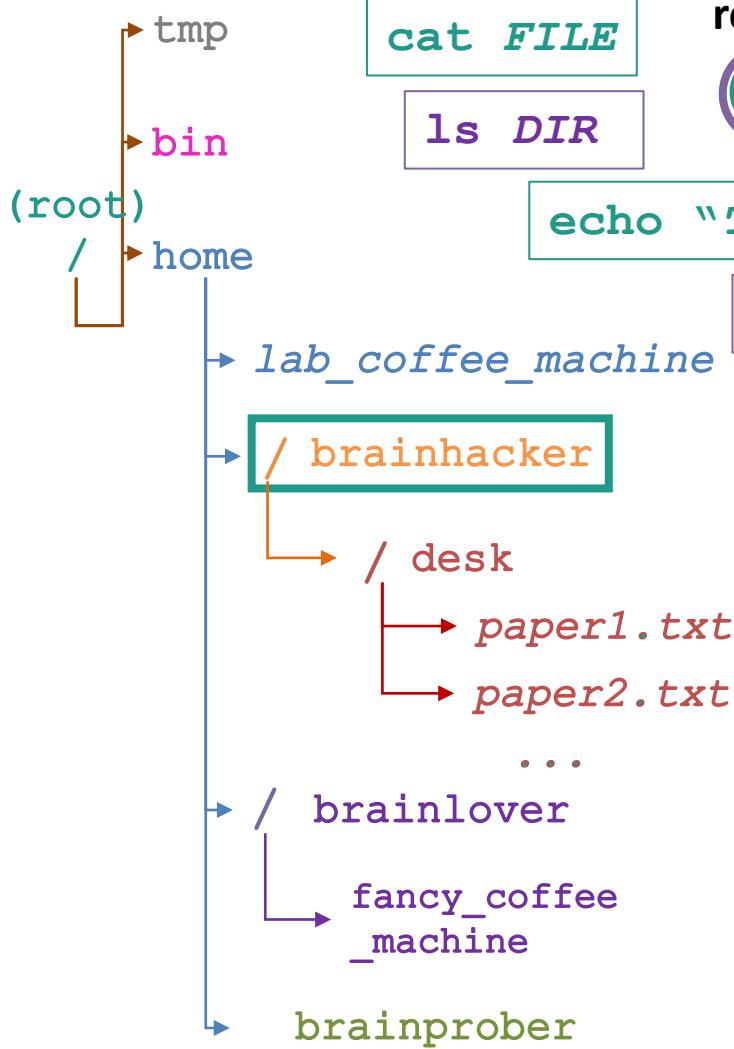
LEVEL 5

Giving the key card to personnel with an insufficient security clearance is strictly forbidden



METHODS
& DATA

LINUX FILESYSTEM



`cat FILE`
`ls DIR`

read write execute
r w x

`echo "TEXT" >> FILE`

`fortune`

`cd DIR`

`rm FILE`

`echo "TEXT" > DIR/FILE`

user (owner)	group	other
rwx	r - x	r - x

Which groups are you part of ?

groups

Permissions of a file *FILE* ?

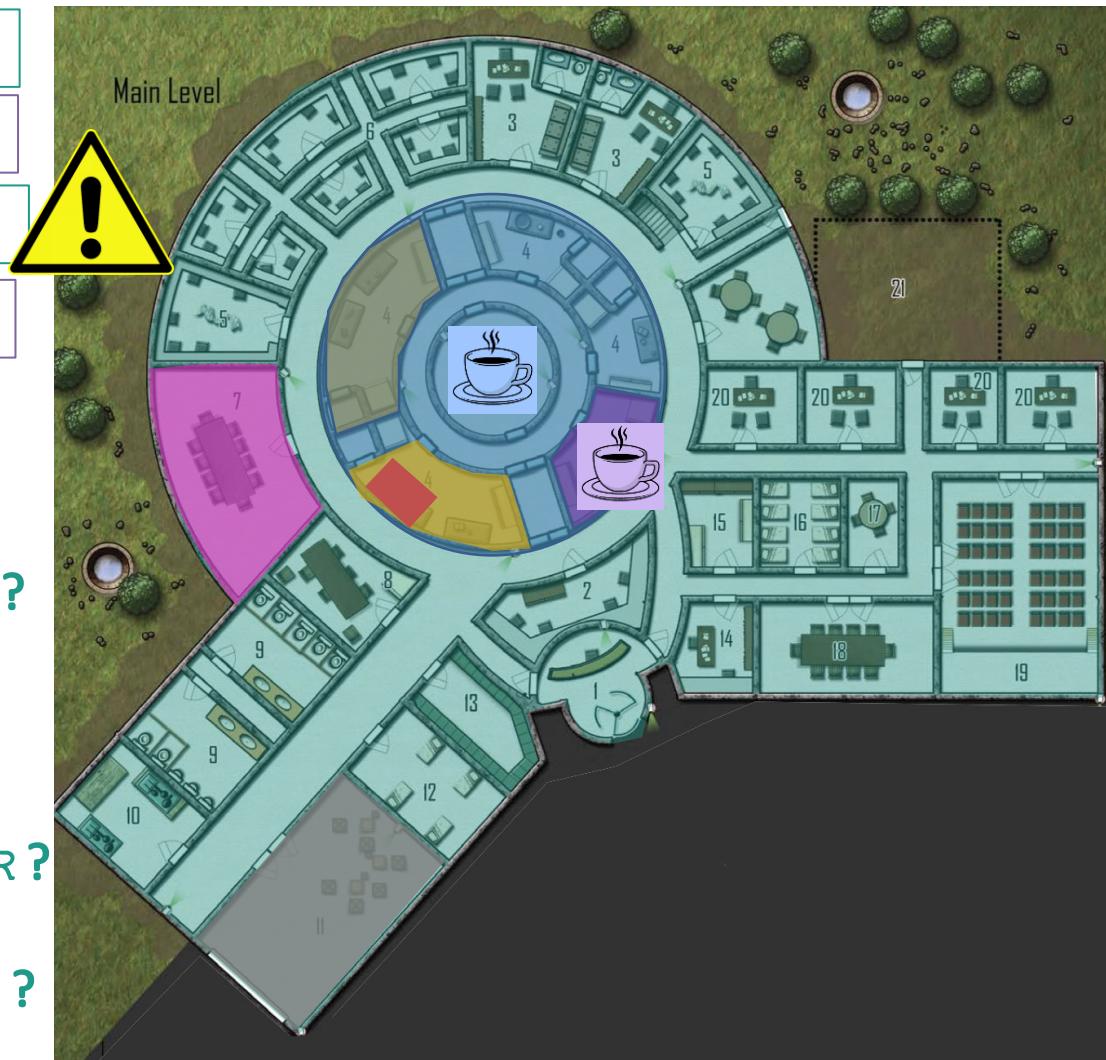
`ls -l FILE`

Permissions of a directory *DIR* ?

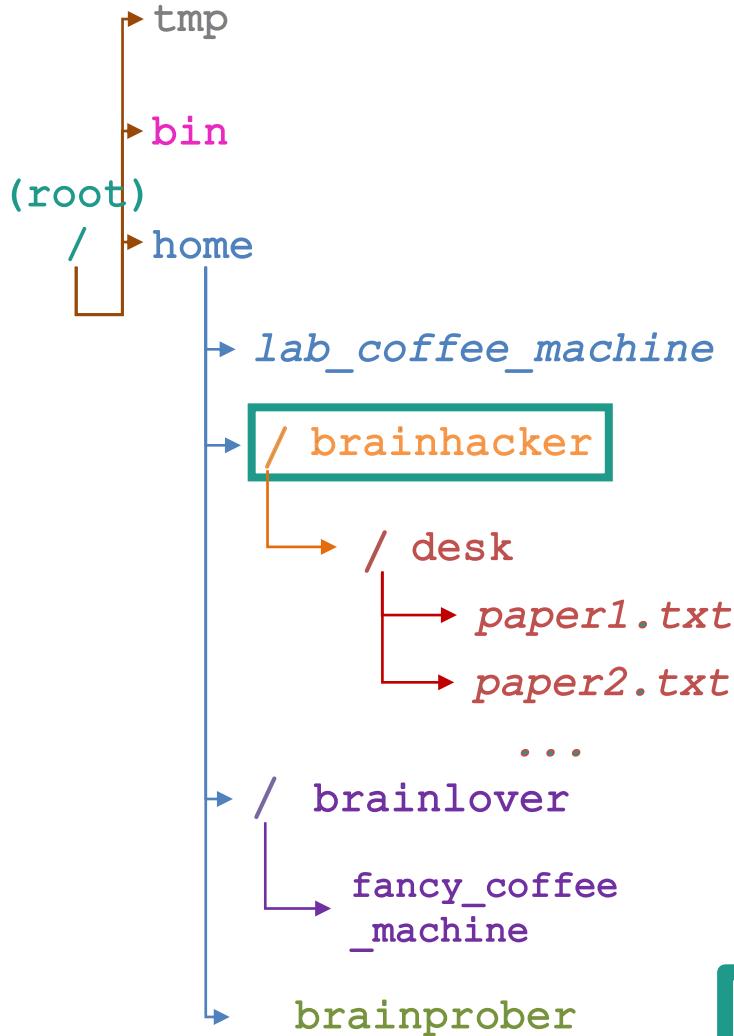
`ls -l -d DIR`

Permissions of all files in *DIR* ?

`ls -l DIR`



LINUX FILESYSTEM



read write execute: r w x

user (owner)	group	other
root rwx	root r - x	r - x
root rwx	neuro r - x	r --

Execute lab coffee machine:

➤ absolute path:

/home/lab_coffee_machine

➤ relative path:

cd /home

./lab_coffee_machine

Destroying lab coffee machine:

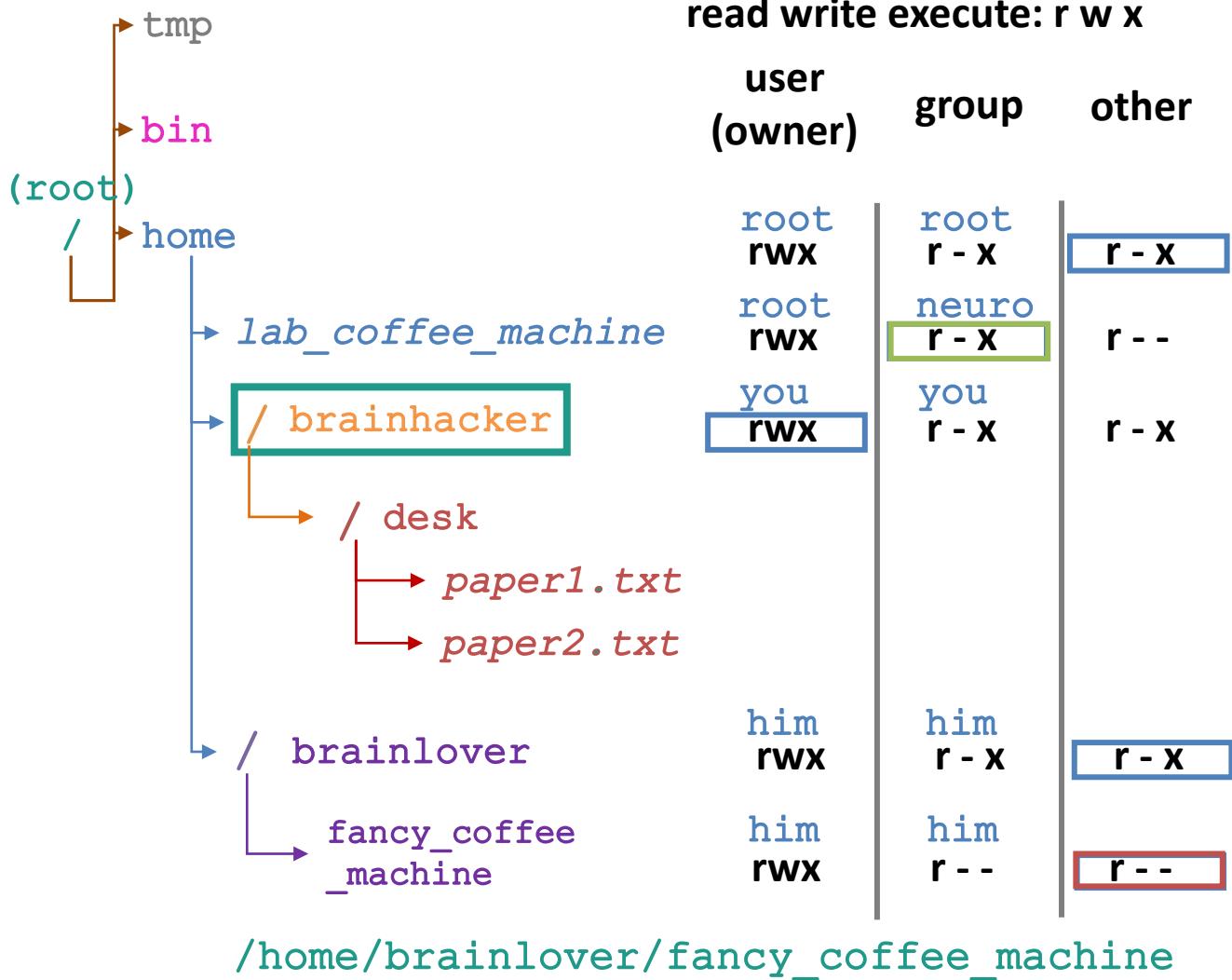
cd /home

rm lab_coffee_machine

Destroy all notepads in your desk



LINUX FILESYSTEM



LINUX FILESYSTEM

TASK

Create `/cosy_corner` directory

(at root level)

How to do what you are not allowed to do?

Use superuser privileges: `sudo`



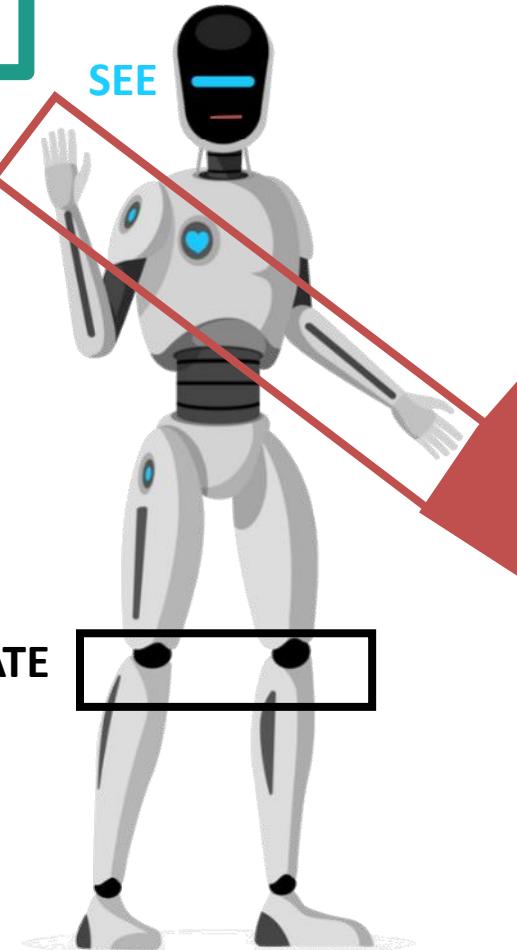
TASK

Type `sudo` before command creating `/cosy_corner` DIR

CREATE &
DESTROY

SEE

NAVIGATE



`mkdir DIRPATH`

`mkdir -p PATH`

`touch FILEPATH`

`rmdir DIRPATH`

`rm FILEPATH`

`cp FILEPATH1 FILEPATH2`

copy file to FILEPATH2, deleting file if it exists

`cp FILEPATH1 DIRPATH`

copy file to DIRPATH, (overwriting also possible)

`cp -R DIRPATH1 DIRPATH2`

copy recursively DIRPATH1 to DIRPATH2

`mv FILEPATH1 FILEPATH2`

move a file

`ln -s EXISTING_FILE SHORTCUTPATH`

create a link to EXISTING_FILE (or DIR) at SHORTCUTPATH



LINUX FILESYSTEM

With great powers come
great responsibilities: avoid
using sudo when possible

Take ownership of directory:

`chown user:user DIRPATH`

Take ownership of directory and all its files :

`chown -R user:user DIRPATH`

Change group ownership of directory :

`chgrp GROUP FILE/DIR`

Change file/directory permissions: NAVIGATE

`chmod`

u	r
g	w
o	x

 +

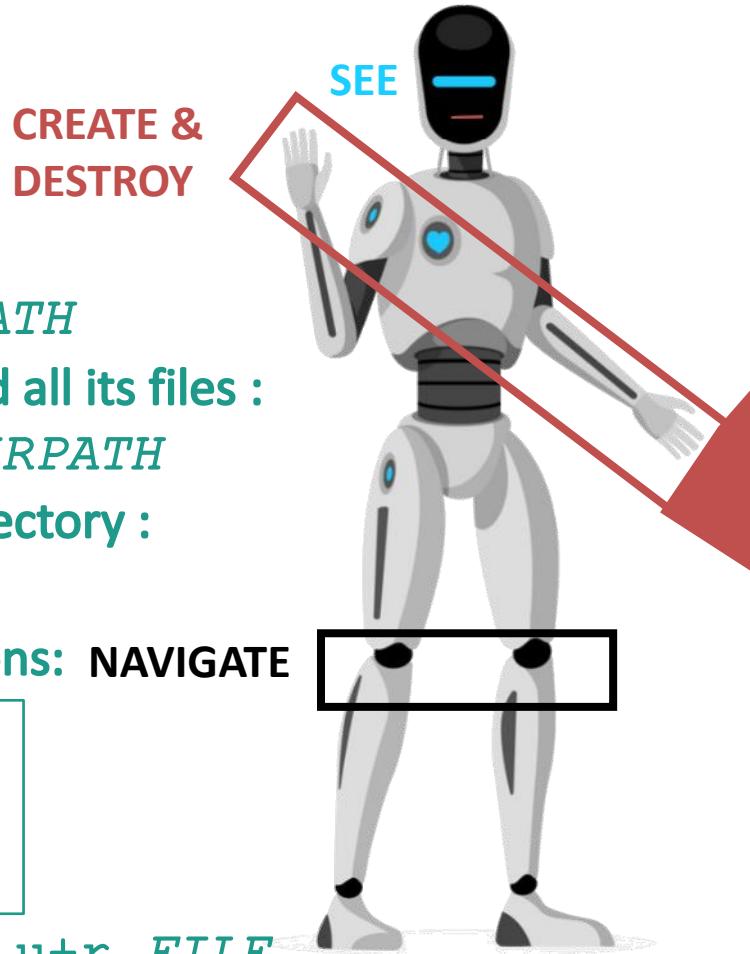
r
w
x
...

 or

rw
rx
...

Add "r" to FILE for user: `chmod u+r FILE`

Remove "wx" to all files in DIR for other: `chmod -R o-wx DIR`



`mkdir DIRPATH`

`mkdir -p PATH`

`touch FILEPATH`

`rmdir DIRPATH`

`rm FILEPATH`

`cp FILEPATH1 FILEPATH2`

copy file to FILEPATH2, deleting file if it exists

`cp FILEPATH1 DIRPATH`

copy file to DIRPATH, (overwriting also possible)

`cp -R DIRPATH1 DIRPATH2`

copy recursively DIRPATH1 to DIRPATH2

`mv FILEPATH1 FILEPATH2`

move a file

`ln -s EXISTING_FILE SHORTCUTPATH`

create a link to EXISTING_FILE (or DIR) at
SHORTCUTPATH

LINUX FILESYSTEM

With great powers come
great responsibilities: avoid
using `sudo` when possible

CREATE &
DESTROY

Take ownership of directory:

```
chown user:user DIRPATH
```

Take ownership of directory and all its files :

```
chown -R user:user DIRPATH
```

Change group ownership of directory :

```
chgrp GROUP FILE/DIR
```

Change file/directory permissions: NAVIGATE

chmod

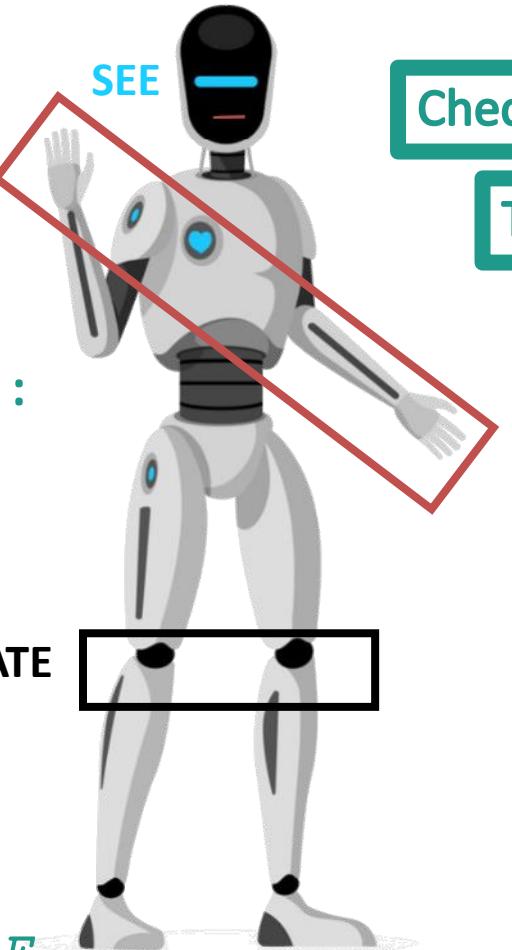
u	r
g	w
+	
-	x
o	

 or

rw
rx
...

Add "r" to *FILE* for user: `chmod u+r FILE`

Remove "wx" to all files in *DIR* for other: `chmod -R o-wx DIR`



TASKS

Check ownership of `/cosy_corner` (`ls -ld DIR`)

Take ownership of `/cosy_corner` directory

Check ownership of `/cosy_corner` again

Change group ownership to `neuro`

Check new group owner

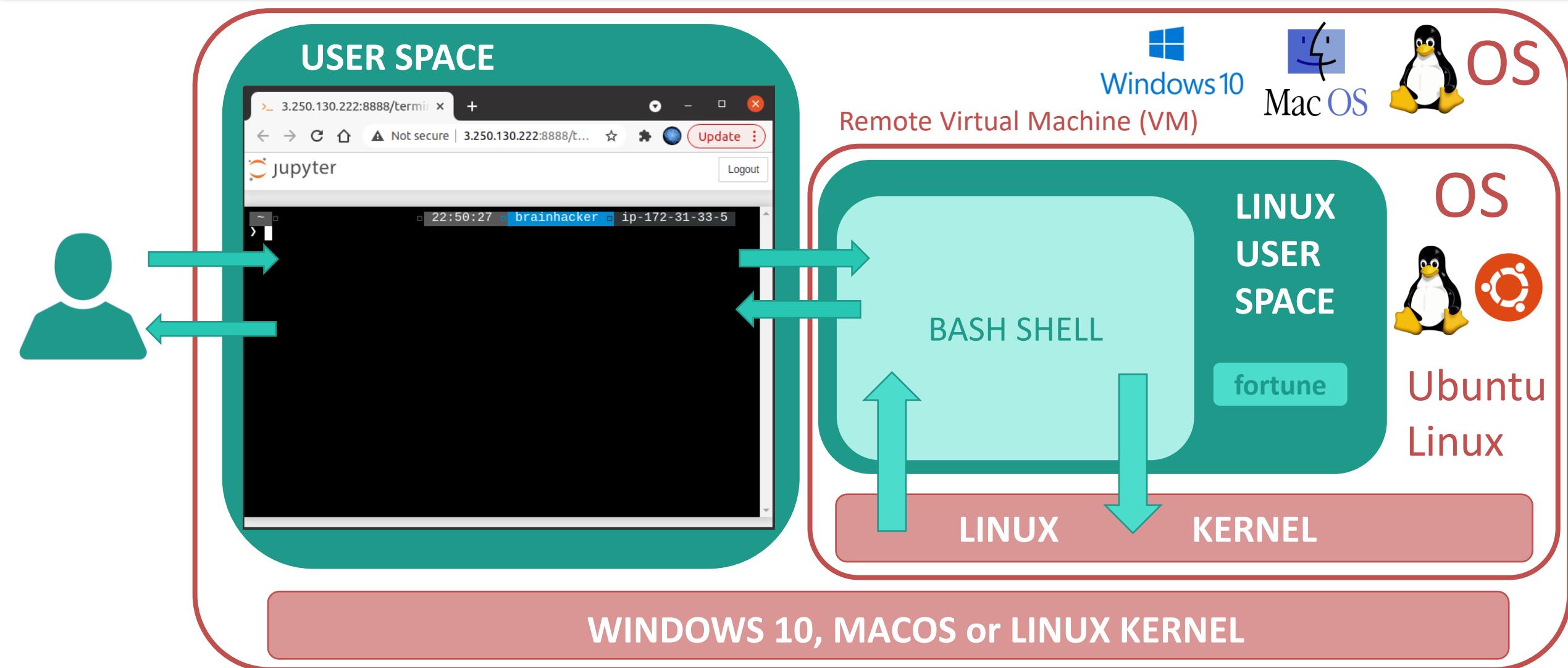
Copy (`cp`) `/home/brainlover/fancy_coffee_machine` to `/cosy_corner`

Provide permissions to any member of the `neuro` group to run the coffee machine

Run the coffee machine at `/cosy_corner`



TERMINAL ON REMOTE VM VIA BROWSER



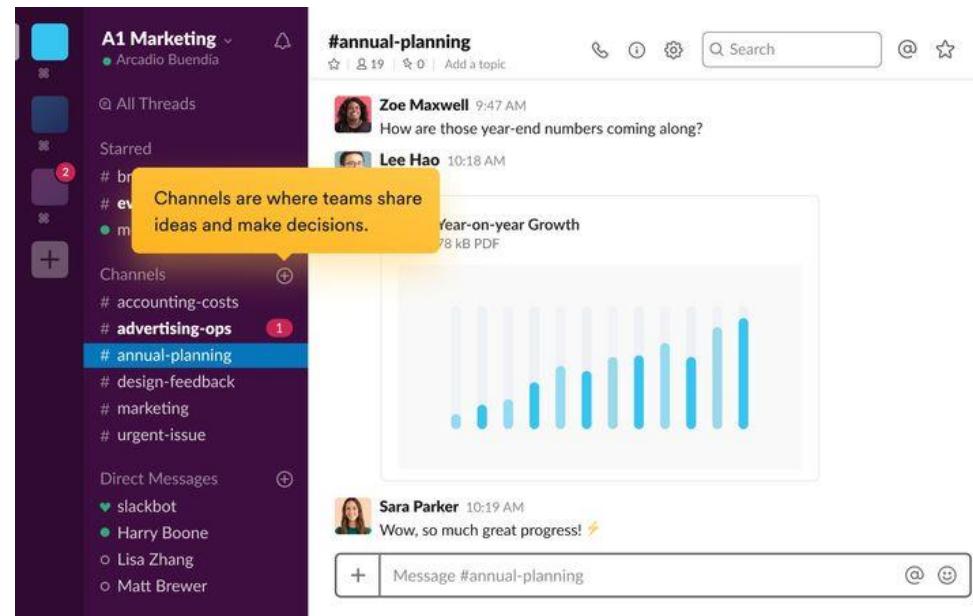
COURSE SUPPORT

SLACK (iords2021.slack.com)

- Course main channel: #general
- Topic channels: #linux, #git, #python, #full-example, #machine-learning

→ Check regularly for course info (esp. pinned items)

→ Do not hesitate to ask questions
(please reply “in thread”)



1-to-1 OFFICE HOURS for course questions:

- 20-min slots every Friday morning between 9AM and 11AM

→ Book a time slot here: <https://tinyurl.com/IORDS-office-hours>

→ Do not hesitate to ask any kind of question, this is a beginner course !

EMAIL: methods@fcbg.ch

Please
whitelist!



Thank You!

Michael Dayan: methods@fcbg.ch