Universiteit Antwerpen
Faculteit Toegepaste Ingenieurswetenschappen

Lab of 3-Network Architecture

Ruben Nietvelt, Nabeel Nisar Bhat

2024-2025

Highlights of the course information 1507FTINWA for the labs

	Lab
Weight	40% of the entire course
Study load (hours)	67 out of 168
Contact hours	24
6.2 Assessment criteria	• Written reports, peer-assessment and continuous evaluation • Minimum of 40% for each part to pass. • Final score calculated according to: BB > Organisations > Examens en onderwijsbeleid > Faculteit Toegepaste Ingenieurswetenschappen > Richtlijnen examens > Tekorten op deelevaluaties

Learning outcomes (more in course configuration)

- The student can manage the Linux operating system, both standalone and networked

- The student is able to analyse an existing network on level 3 and 4 (IP-TCP), solve problems and synthesise new networks.

- The student has the knowledge and skill to work with command line tools and configure a local network with access to the internet.

Scheduled labs for PR01

Session	Date	Subject	Evaluation	Deadline (23:59)
1	01/10/2024	Introduction to the Linux Operating System: Using the shell & exploring the filesystem	Report	07/10/2024
2	08/10/2024	Working with text files and managing running processes	Report	14/10/2024
3	15/10/2024	Writing shell scripts	Report	22/10/2024
4	23/10/2024	Learning system administration, getting & managing software	Report	28/10/2024
5	29/10/2024	Wireshark introduction	Report	05/11/2024
6	06/11/2024	Protocols in action: TCP and UDP	Report	11/11/2024
7	12/11/2024	Ethernet and ARP	Report	19/11/2024
8	20/11/2024	Setting up a DHCP server	Report	25/11/2024
9	26/11/2024	Setting up a DNS server	Report	03/12/2024
10	04/12/2024	Network Address Translation	Report	09/12/2024
11	10/12/2024	Remote Access & Firewalls (1)		**N/A**
12	18/12/2024	Remote Access & Firewalls (2)	Blackboard test	

Scheduled labs for PR02

Session	Date	Subject	Evaluation	Deadline (23:59)
1	02/10/2024	Introduction to the Linux Operating System: Using the shell & exploring the filesystem	Report	08/10/2024
2	09/10/2024	Working with text files and managing running processes	Report	15/10/2024
3	16/10/2024	Writing shell scripts	Report	22/10/2024
4	23/10/2024	Learning system administration, getting & managing software	Report	29/10/2024
5	30/10/2024	Wireshark introduction	Report	05/11/2024
6	06/11/2024	Protocols in action: TCP and UDP	Report	12/11/2024
7	13/11/2024	Ethernet and ARP	Report	19/11/2024
8	20/11/2024	Setting up a DHCP server	Report	26/11/2024
9	27/11/2024	Setting up a DNS server	Report	03/12/2024
10	04/12/2024	Network Address Translation	Report	10/12/2024
11	11/12/2024	Remote Access & Firewalls (1)		**N/A**
12	18/12/2024	Remote Access & Firewalls (2)	Blackboard test	

Agreements

Session 1

Introduction to the Linux operating system

What is GNU/Linux?

- **A traditional operating system, performing tasks such as:**
  - Providing user interfaces
  - Providing user access and authentication
  - Starting up services
  - General programming tools

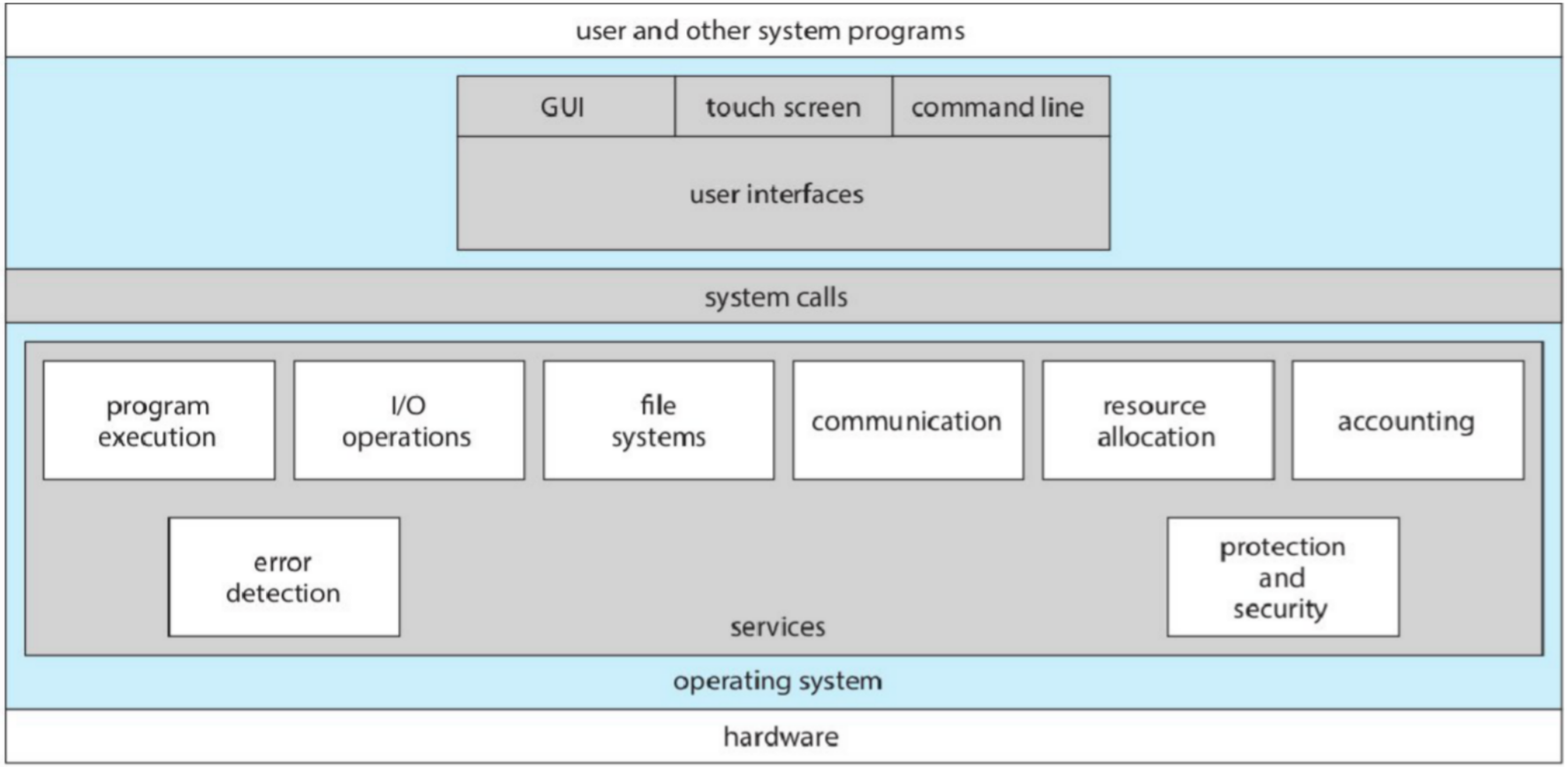
  - Detecting and preparing hardware
  - Managing processes
  - Managing memory
  - Controlling filesystems

GNU:
User space

Linux: Kernel
space

What is GNU/Linux?

- **Operating system structure**



user and other system programs

GUI | touch screen | command line

user interfaces

system calls

program execution | I/O operations | file systems | communication | resource allocation | accounting

error detection | protection and security

services

operating system

hardware

GNU:
User space

Linux: Kernel
space

How does Linux differ from other OS's?

Linux is Free and Open-Source Software (FOSS)

Windows and macOS are both proprietary software:

- The code to create the OS is invisible to you and thus cannot be adapted to fit your needs.
- You cannot check the code for vulnerabilities or simply learn from it.
- Plugging in your own software may be difficult, as the creators decide which programming interfaces are available.
- Come with licensing fees.

Why free and open-source software?

FOSS has driven:

- The internet (think Google, market share of Linux web servers)
- Mobile phones (Android)
- A reduction in computing costs

The arguments for Linux above are among the reasons why companies such as Google and Facebook their computer infrastructures rely on Linux. And thus, there is quite the need for people with the skills to run those systems.

"Google and Facebook owe their success largely to Linux -- not the technology per se, but to the cheap innovation and mass collaboration it enables, Red Hat's CEO says."

Did Linux originate from free software?

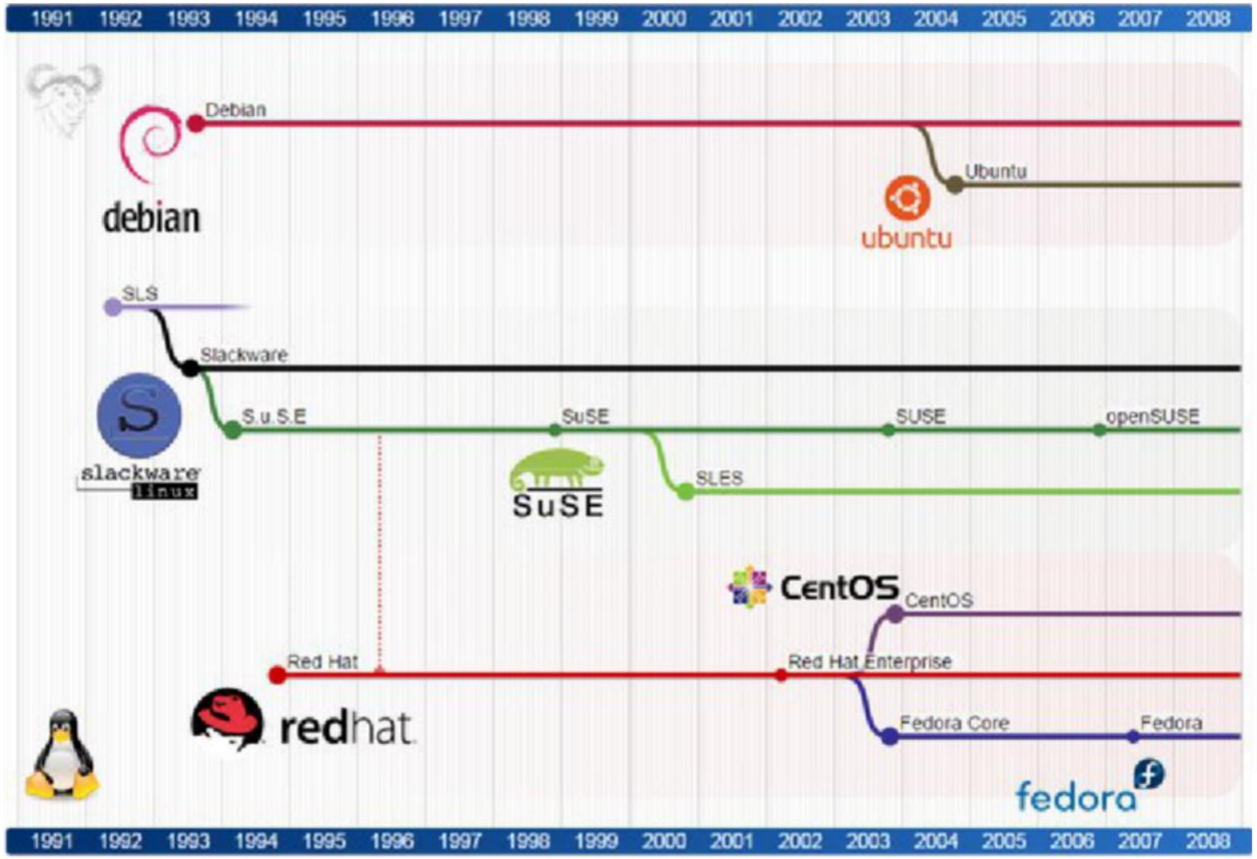
What else is there to Linux?

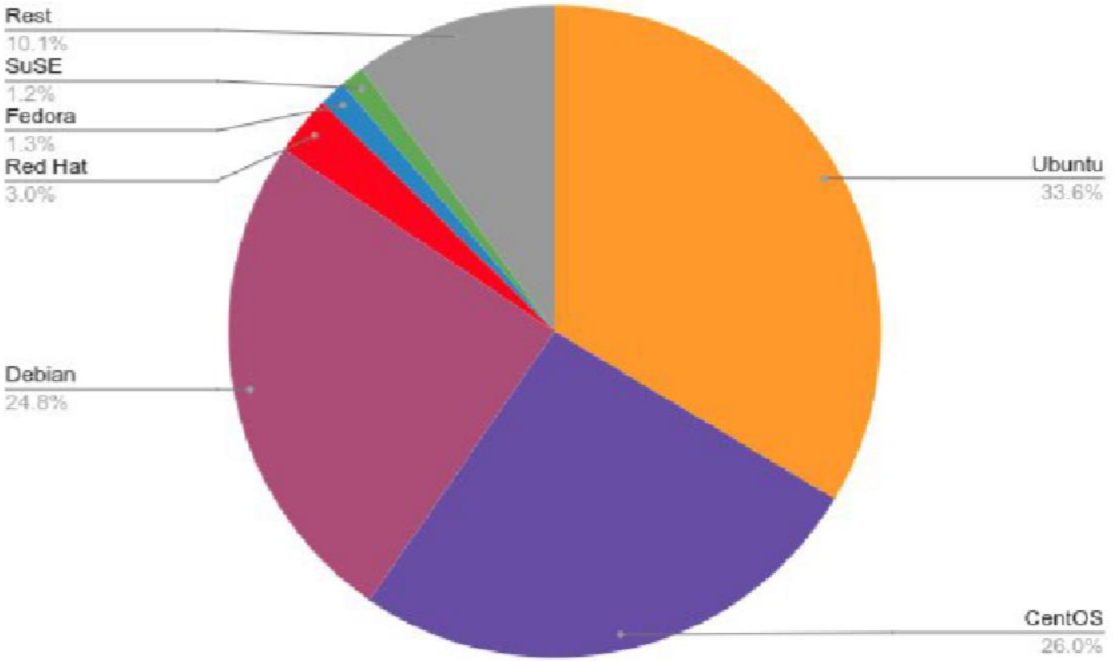
Modern Linux systems offer advanced enterprise features:

- Cloud Clustering
  - In order to improve performance & availability over a single system
  - Multiple systems appear as one to the outside world
- Virtualisation
  - In order to more efficiently manage computing resources
  - A single host appears as multiple to the outside world
- computing is the large-scale management of the previous two points
- Real-time computing

While not all features are covered in this lab, it will serve as a foundation for working with those advanced features.

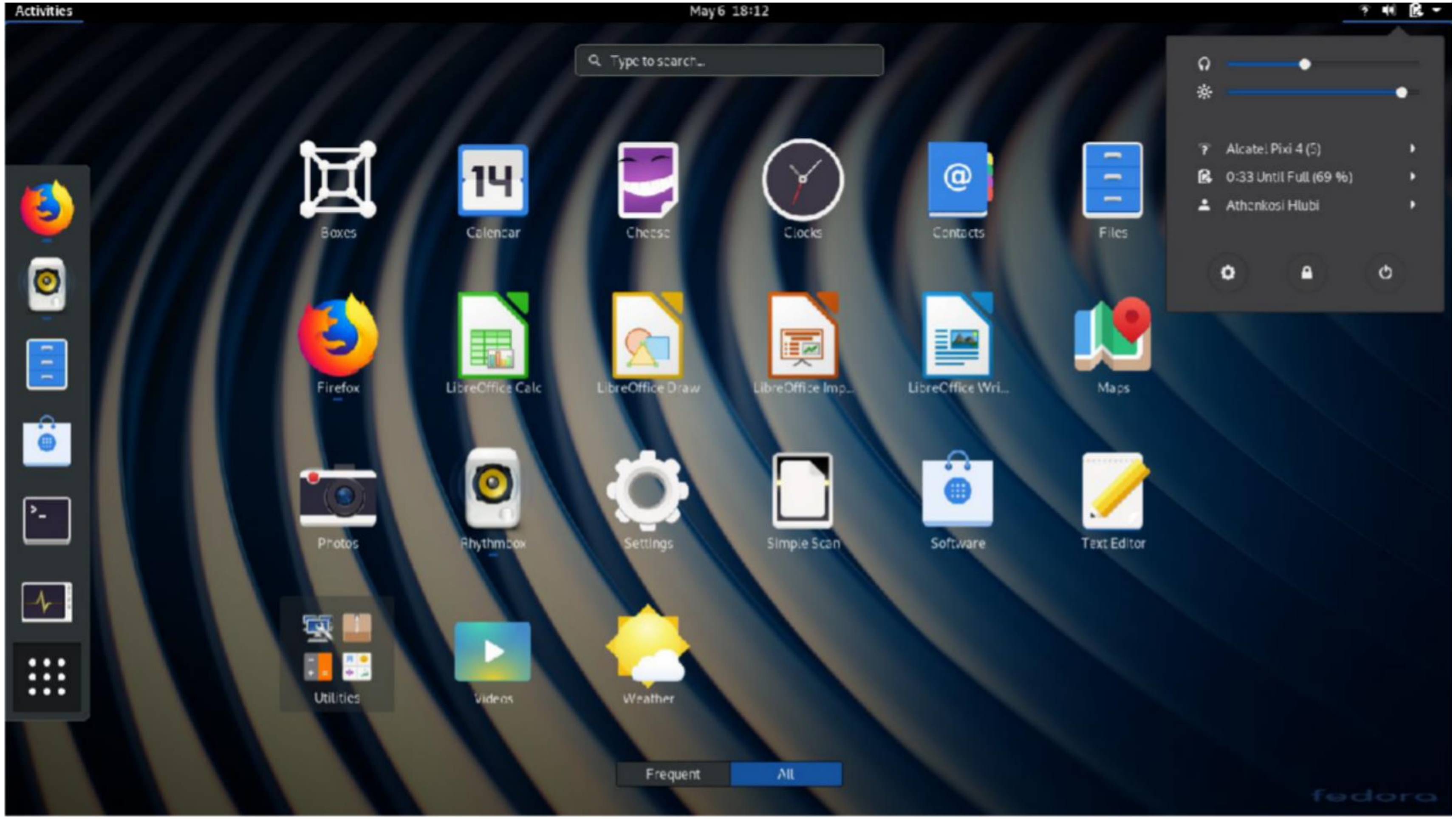
Linux distributions



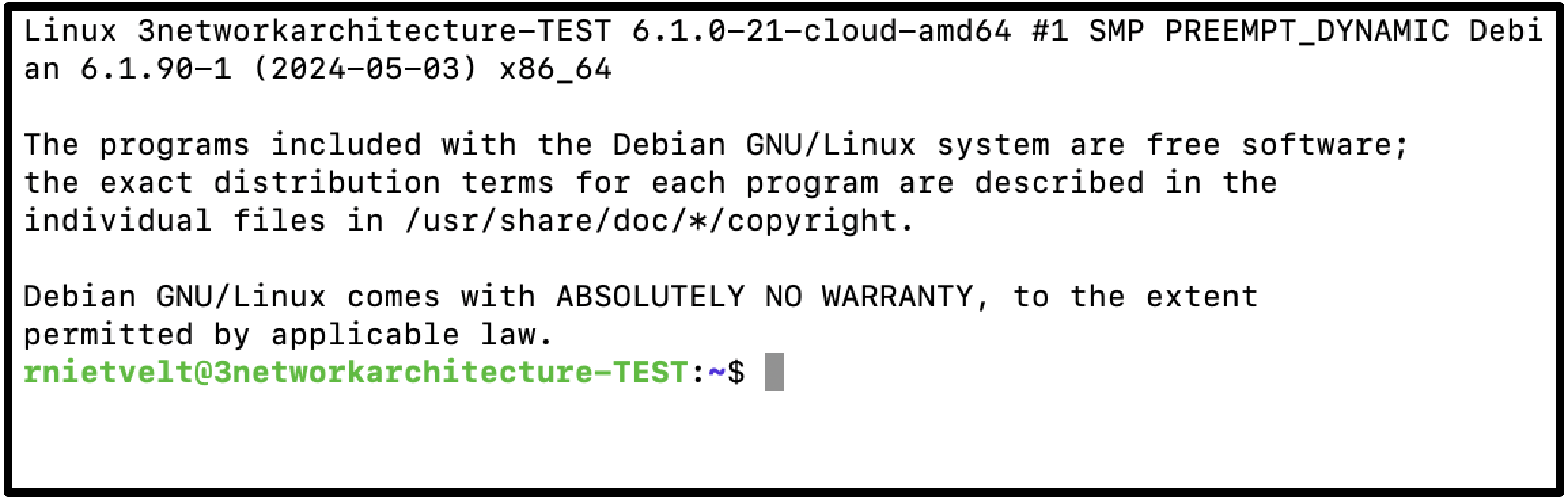


Linux on the desktop

- **GNOME desktop environment**



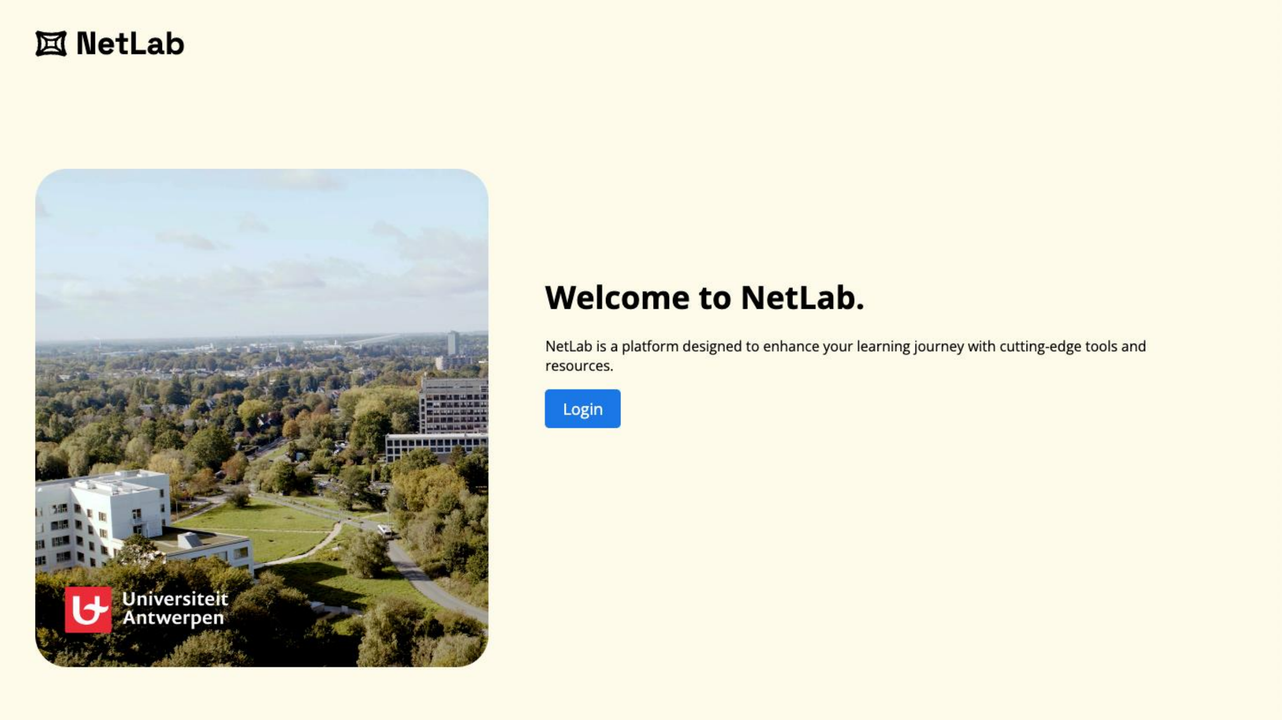
Linux in server environments



```
Linux 3networkarchitecture-TEST 6.1.0-21-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debi
an 6.1.90-1 (2024-05-03) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
rnietvelt@3networkarchitecture-TEST:~$ 
```

NetLab

- Portal to manage cloud VM’s
- Only accessible via UAntwerp VPN or on campus.
- https://netlab.ei.fti.uantwerpen.be
- After first login => upload SSH key for remote access to your VM(’s)

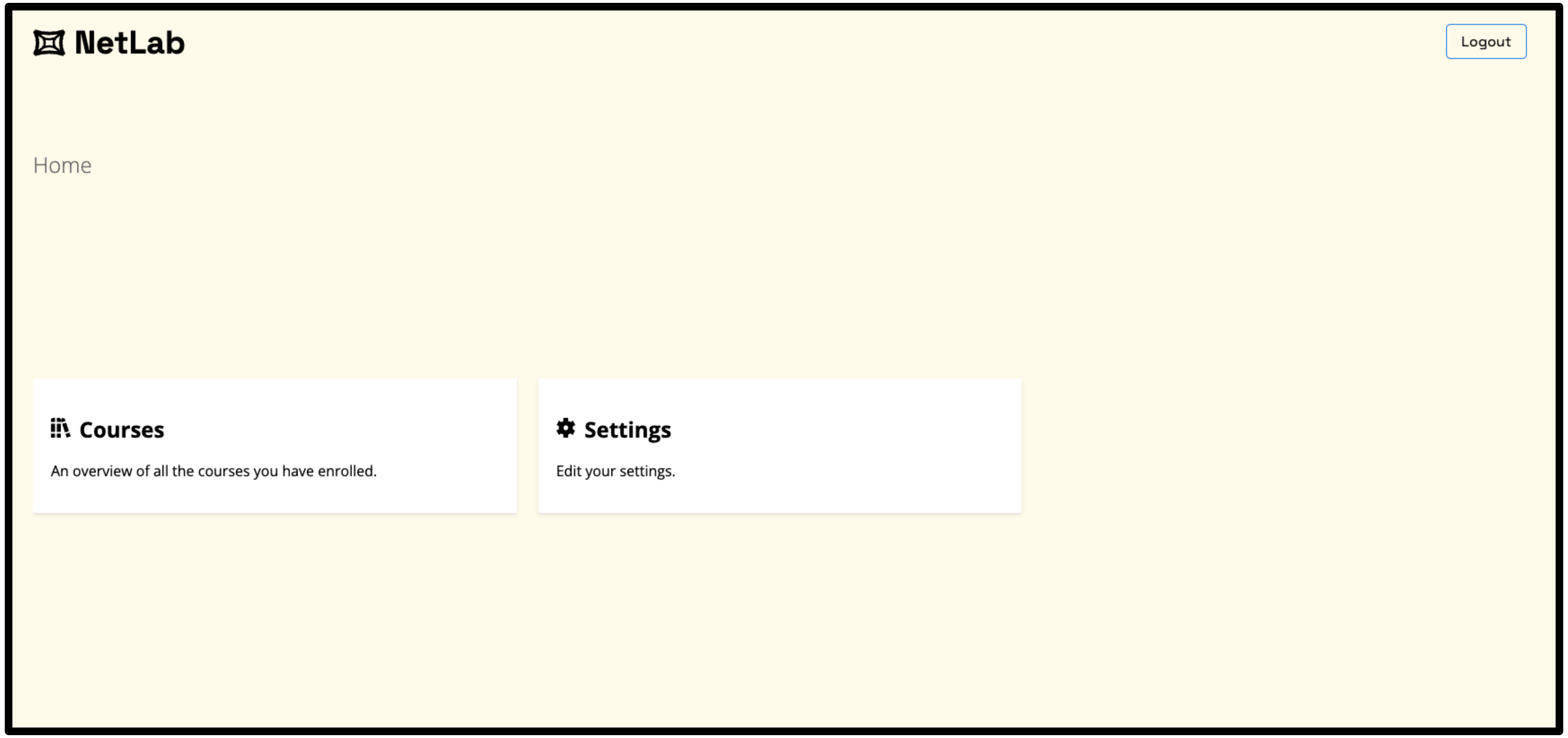


NetLab

Welcome to NetLab.

NetLab is a platform designed to enhance your learning journey with cutting-edge tools and resources.

NetLab – Upload SSH key in "Settings"



Home

Courses

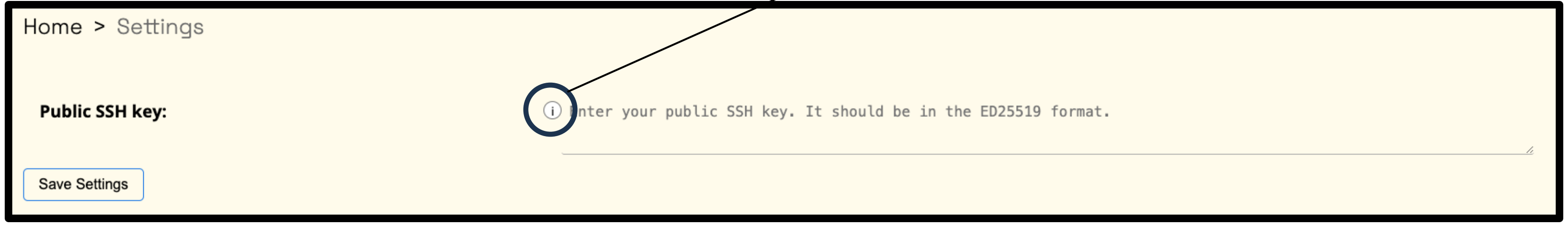
An overview of all the courses you have enrolled.

Settings

Edit your settings.

- **If no SSH key uploaded before:**

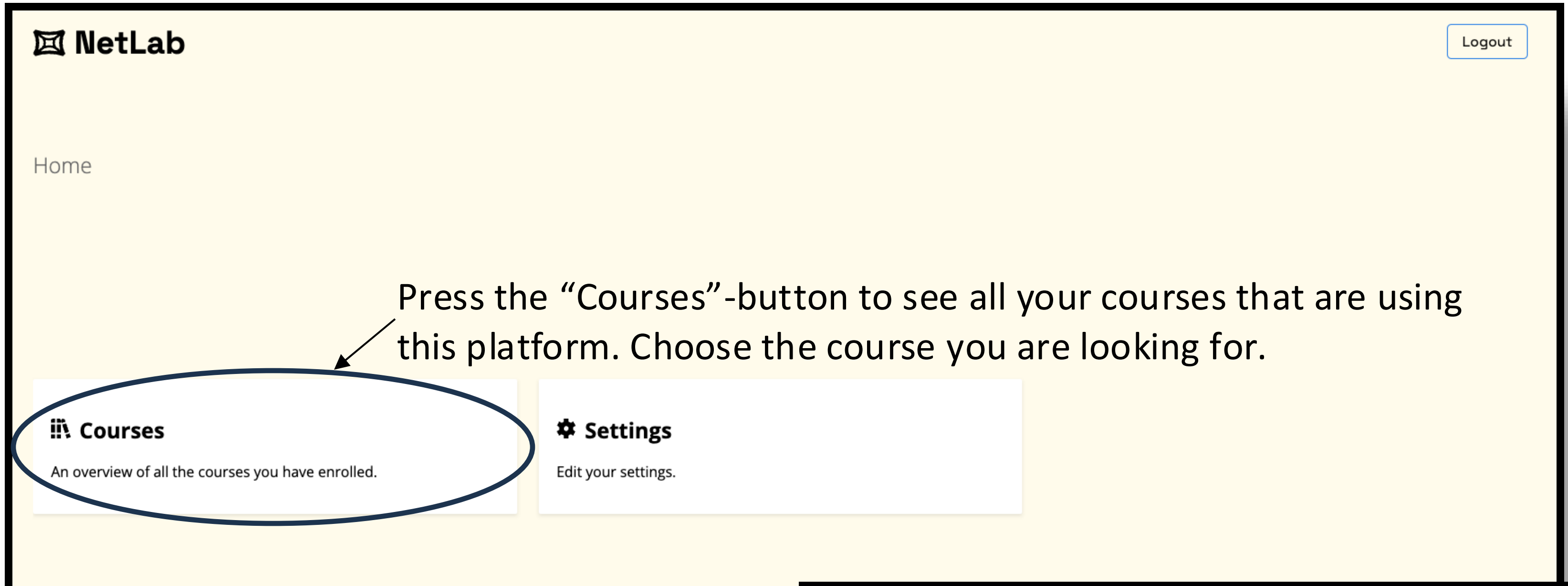
Info button helps you find your SSH key



Home > Settings

Public SSH key: Enter your public SSH key. It should be in the ED25519 format.

NetLab – Create a VM

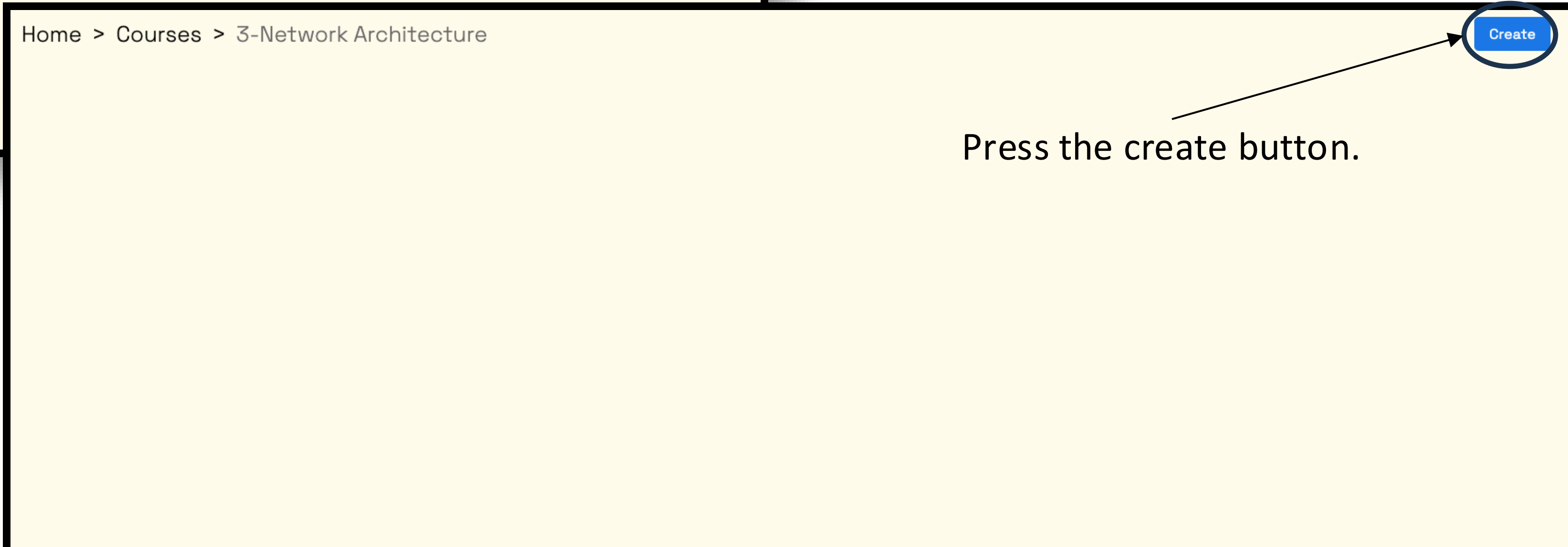


An overview of all the courses you have enrolled.

Edit your settings.

Press the “Courses”-button to see all your courses that are using this platform. Choose the course you are looking for.

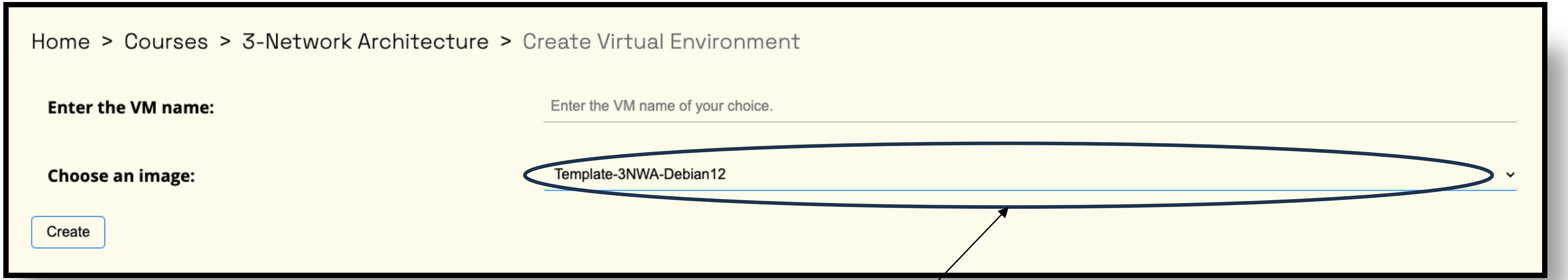
- **Empty screen if the teacher did not create a VM for you.**



Press the create button.

University of Antwerp
Faculty of Applied Engineering

NetLab – Create a VM



Home > Courses > 3-Network Architecture > Create Virtual Environment

Enter the VM name: Enter the VM name of your choice.

Choose an image: Template-3NWA-Debian12

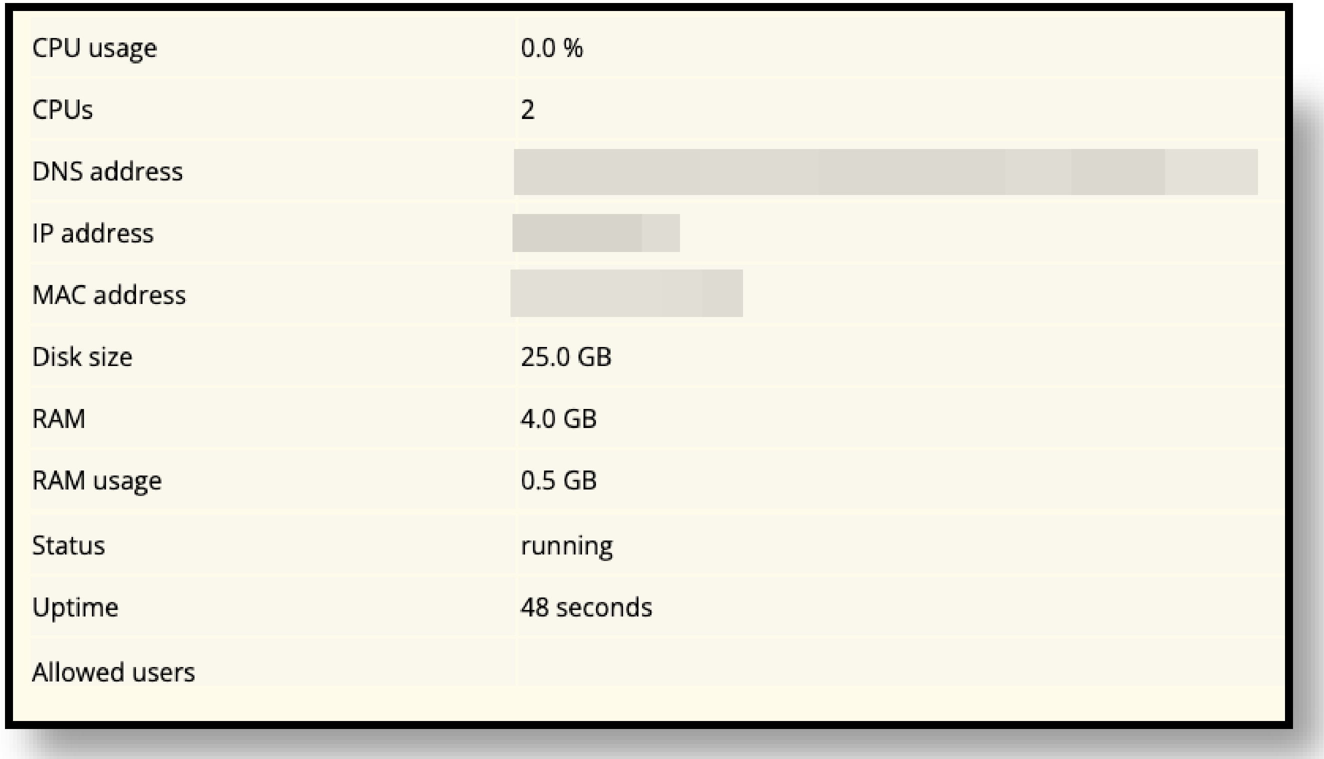
If the teacher applied the settings correctly, a non-empty list of possible images should show up. Follow the guidelines of the teacher on which image to use.

NetLab - Remotely connect to given machine

- Use: ssh shortened_ua_name@address

- If you have multiple SSH keys, the best thing is to specify which private key the command needs to use. This can be done by adding “-i ./.ssh/id_ed25519” with “./.ssh/id_ed25519” being the location of the private key file.

- The DNS/IP address can be found on NetLab, on the overview page of the machine, e.g.:



CPU usage	0.0 %
CPUs	2
DNS address	[redacted]
IP address	[redacted]
MAC address	[redacted]
Disk size	25.0 GB
RAM	4.0 GB
RAM usage	0.5 GB
Status	running
Uptime	48 seconds
Allowed users	

Linux Shell

What is a *shell*?

On UNIX, the program used to interpret and manage commands was referred to as the Bourne shell and its cmd is sh.

Named after its creator Stephen Bourne

- Provides a way to:
  - create executable scripts
  - run programs
  - work with filesystems
  - compile computer code
  - manage the computer
- Less intuitive yet more powerful than GUIs

What is a *shell*?

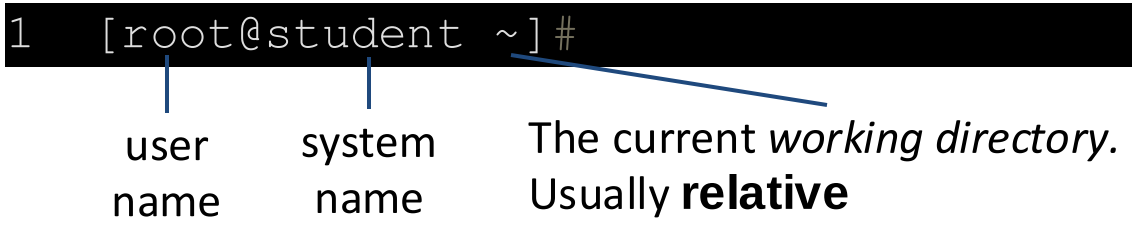
On Linux, the default shell is often the bash shell and its command is bash. Given its roots in the Bourne shell, bash stands for Bourne again shell.
When you open a terminal **or have no GUI**, the interface provided is this bash shell.

Using the shell prompt

A default prompt for **regular users** is the dollar sign $

```
[student@student ~]$
```

A default prompt for **root user** is the pound sign #



```
[root@student ~]#
```

user name

system name

The current *working directory.*
Usually **relative**

To print your current working directory, use the command **pwd**, which stands for 'print working directory’

```
[student@localhost ~]$ pwd
/home/student
[student@localhost ~]$
```

Running commands

```
[student@student ~]$ date
Thu Jun 29 08:14:53 EDT 2019
```

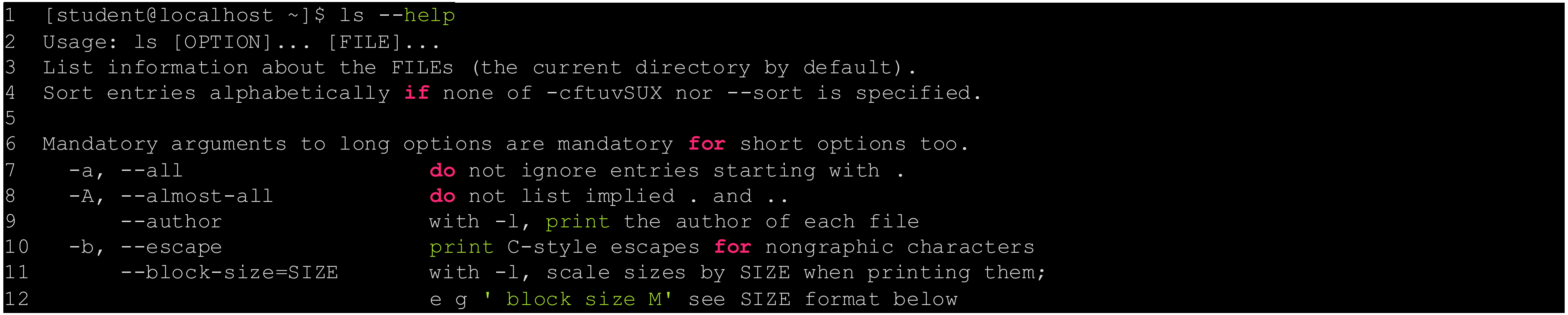
Command syntax. Unsure? **cmd -h** or **cmd --help** or **man cmd**

```
$ ls -l -a -t
$ ls -lat
```

```
[student@localhost ~]$ ls -a
. .bash_logout .cache Documents .local Pictures Videos
.. .bash_profile .config Downloads .mozilla Public .viminfo
.bash_history .bashrc Desktop .esd_auth Music Templates
```

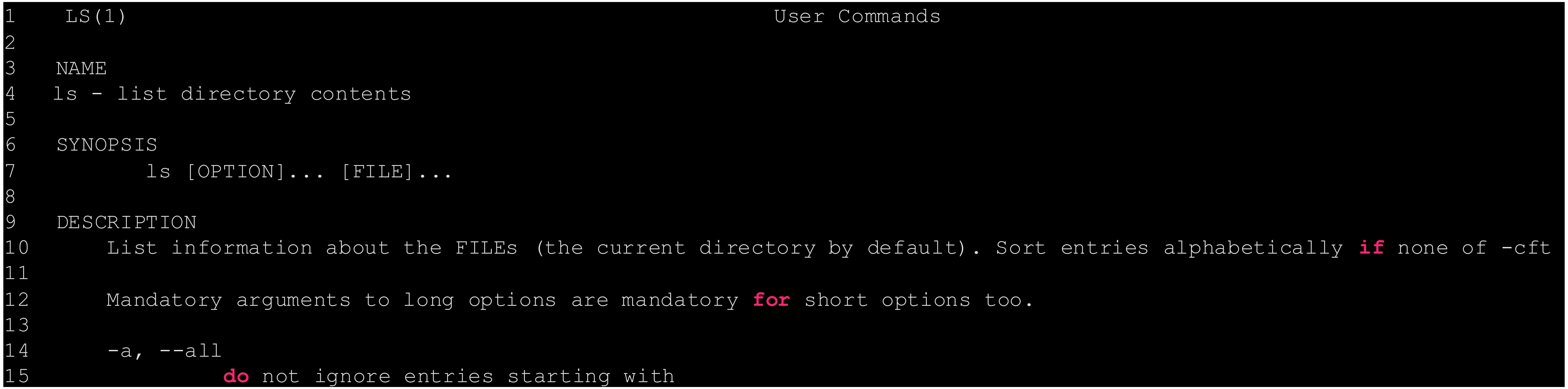
```
[student@localhost ~]$ ls -la
total 80
drwx------. 14 student student 4096 Sep 29 01:23 .
drwxr-xr-x. 3 root root 4096 Sep 28 16:18 ..
-rw-------. 1 student student 470 Oct 6 13:43 .bash_history
-rw-r--r--. 1 student student 18 Jan 28 2020 .bash_logout
-rw-r--r--. 1 student student 141 Jan 28 2020 .bash_profile
-rw-r--r--. 1 student student 376 Jan 28 2020 .bashrc
drwx------. 18 student student 4096 Sep 29 01:35 .cache
drwxr-xr-x. 21 student student 4096 Sep 29 17:45 .config
```

Usually there is a help option **--help**



```
[student@localhost ~]$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                  do not ignore entries starting with .
  -A, --almost-all           do not list implied . and ..
      --author               with -l, print the author of each file
  -b, --escape               print C-style escapes for nongraphic characters
      --block-size=SIZE      with -l, scale sizes by SIZE when printing them;
                             e g ' block size M' see SIZE format below
```

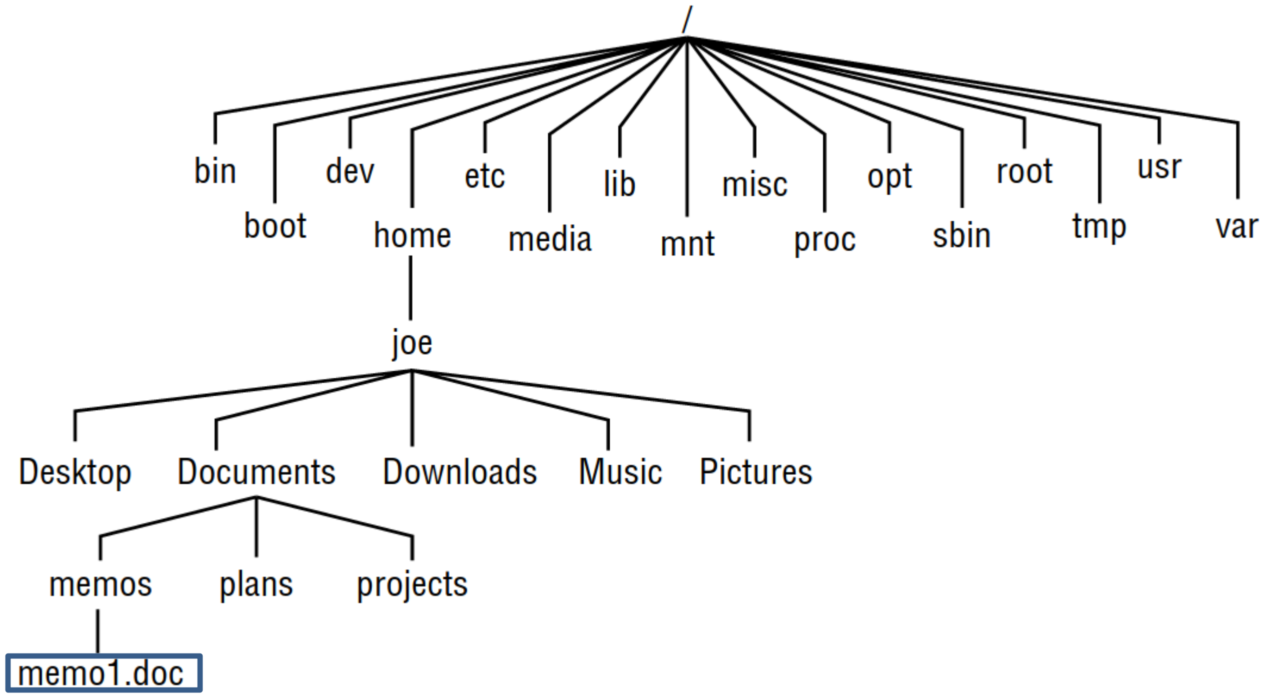
But the most information on how to use a command is the **manual pages**. Invoke it with **man <cmd>**



```
 LS(1)                                                  User Commands

NAME
ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -cft

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
             do not ignore entries starting with
```

Moving around the filesystem

The Linux filesystem



Every piece of information can be referred to using the filesystem hierarchy.

‘/’ is the root directory (not the root user’s directory).

Absolute path:
/home/joe/Documents/memos/memo1.doc

Relative path:
memos/memo1.doc

Relative paths are convenient, but attention to your working directory when using them.

File Permissions

Since all data, commands, symbolic links, devices, and directories are represented by **items in the filesystem**, they share the same manner of permission management.

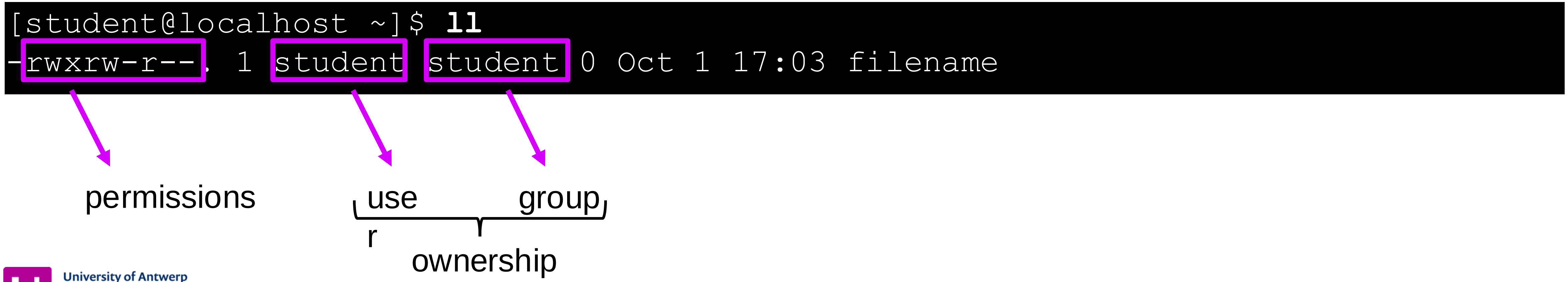
An item (often a file or folder) can be **read**, **written** or **executed** by different users and groups. To find out who can do what to a certain file, we will use the **ls** command with the **–l** argument to get a “long list” output of the items in the current working directory. Usually, there is an alias “**ll**”.

```
[student@localhost ~]$ ll
-rwxrw-r--. 1 student student 0 Oct 1 17:03 filename
```

File Permissions

Since all data, commands, symbolic links, devices, and directories are represented by items in the filesystem, they share the same manner of permission management.

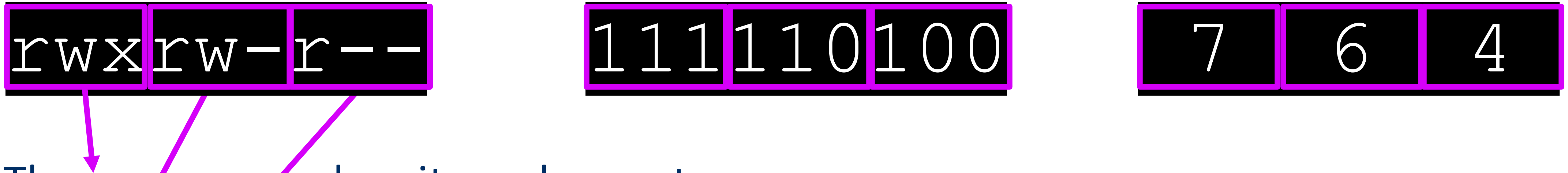
An item (often a file or folder) can be **read**, **written** or **executed** by different users and groups. To find out who can do what to a certain file, we will use the **ls** command with the **–al** argument to get a "long list" output of the items in the current working directory. Usually, there is an alias fo "**ll**". But not in our Debian distribution.



```
[student@localhost ~]$ ll
-rwxrw-r--. 1 student student 0 Oct 1 17:03 filename
```

Permission format

```
[student@localhost ~]$ ll
-rwxrw-r--. 1 student student 0 Oct 1 17:03 filename
```



rwx rw- r--

111 110 100

7 6 4

The **user** can **r**ead, **w**rite and e**x**ecute.

The **group** (thus users belonging to this group) can **r**ead and **w**rite.

All **others** can only **r**ead.

Important commands

Most important commands

Command	Explanation
ls	Lists directory contents of files and directories.
cd	Change directory.
touch	Updates the access and modification times of each file.
mkdir	Make directory.
cat	Copy content of a file (to terminal or other file).
echo	Display lines of text or strings that are passed as arguments.
sudo	Execute command using root privileges. This access is password-protected and only valid for a limited time. Use ‘sudo su’ for $ ➡️ #
apropos	Searching for commands without knowing their exact names.

Exercises

Exercises

1. Run the **date** command in such a way that the output from that command produces the current day, month, date, and year. Have that read into another command line, resulting in text that appears like the following (your date, of course, will be different): *Today is Thursday, December 19, 2019*.

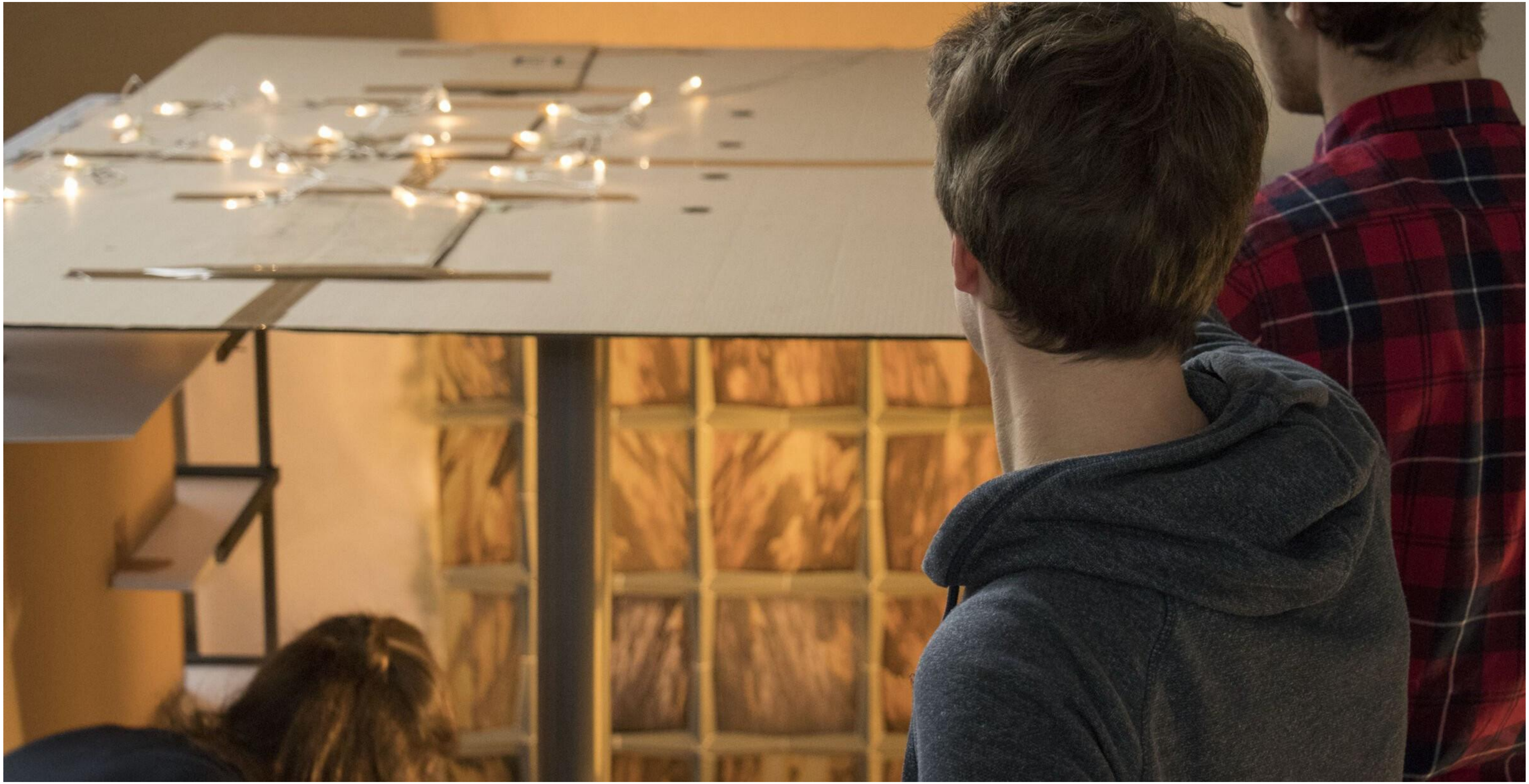
2. Using variables, find out what your hostname, username, shell, and home directories are currently set to.

3. Create a folder **myfolder** and two files **~/myfolder/myfile1** and **~/myfolder/myfile2** in your user's home directory and change the permission of ~/myfolder/myfile1 so that it can be read by and written to by the user who owns the file, read and executed by the group and only read by others.

4. Recursively change permissions of the **~/myfolder** directory so that nobody has write permission to any files or directory beneath that point in the filesystem.

References

- Silberschatz, A. (2017). Operating System Concepts, 10th Edition. New York: John Wiley & Sons, Incorporated.



University of Antwerp
Faculty of Applied Engineering