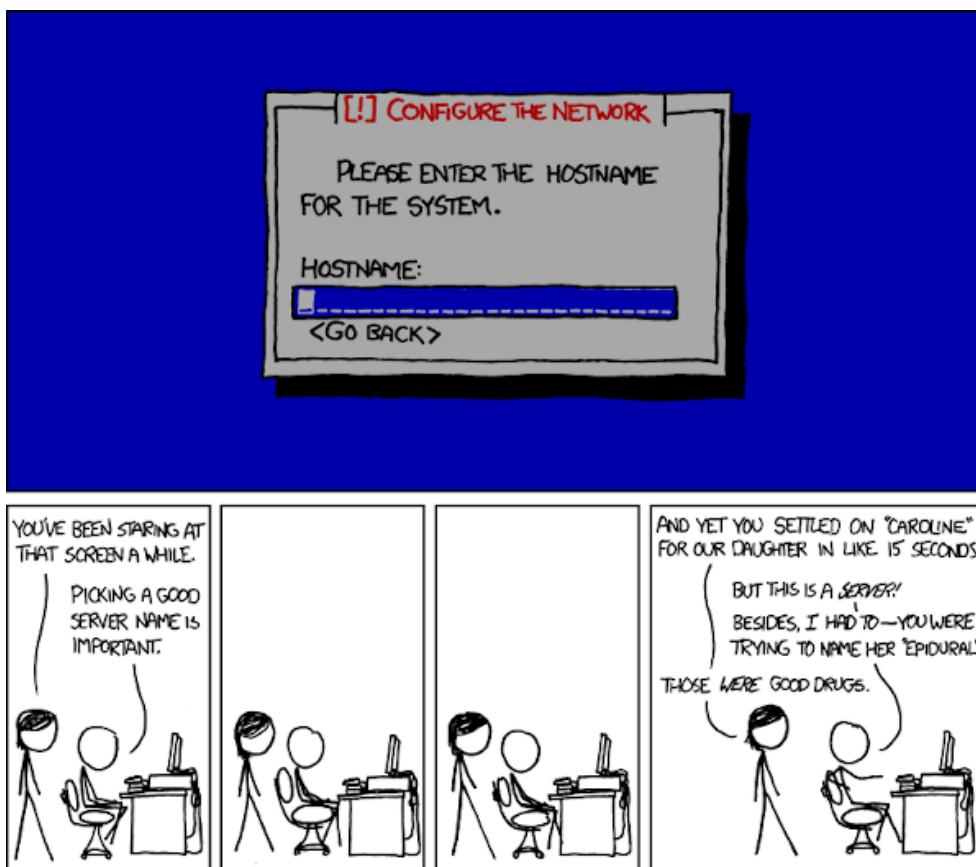# KU LEUVEN

GROUP T LEUVEN CAMPUS

Operating systems: homework 1
Setting up the development environment

Bachelor Electronics/ICT – Operating Systems

Coordinator: Bert Lagaisse

Lab teachers:

- Ludo Bruynseels
- Jonas Verbruggen
- Toon Dehaene

```
--------------------------------------------------------------------
| 1.   Introduction: Setting up your development environment       |
--------------------------------------------------------------------
```

For the pc sessions we want to make sure that all students work in a uniform environment.

The goal of this homework is to set up such an environment on your laptop. This will help you and the teaching staff to optimize the efficiency of these sessions.

We want to avoid that students and staff loose time to diagnose and fix any problem that might arise with your specific laptop or operating system configuration.

To anticipate that fact that many students have limited experience with Linux, we designed this tutorial to make your life as easy as possible. (at least in this first installment)

```
--------------------------------------------------------------------
| 2.   Overview                                                    |
--------------------------------------------------------------------
```

Our uniform development environment will be set up in a virtual machine on your laptop. We will set up the following tools:

- Oracle Virtual Box (https://www.virtualbox.org/)
- Virtual box guest additions for Ubuntu such that you can use copy-paste between your native environment and the virtual environment.
- Ubuntu 22.04LTS Desktop (https://ubuntu.com/download/desktop)
- Git
- Clion
- net-tools
- gcc
- make

```
--------------------------------------------------------------------
| 3.   Installing Virtual Box                                      |
--------------------------------------------------------------------
```

Virtual Box 7 is a virtualization product that runs on Windows, Linux, macOS and Solaris hosts, including the new Apple MacBook. Hence, this is a good starting point for a uniform environment. You can download Virtual Box for free from here: https://www.virtualbox.org/wiki/Downloads.

If your workstation already boots Ubuntu, you don't have to install Virtual Box. If your pc boots another Linux distribution, you will probably be able to make all the exercises in this course, but you might encounter small problems during compilation and linking. If this happens you are responsible to solve these problems. (Your friends, teachers, family, Google, Stackoverflow might help you out but without any SLA).

Download both the **platform package** for your operating system as well as the **extension pack**. First install the platform, and then under tools, you can install the extension pack. The platform is installed in the well known Windows style : next – ok - …

During the installation of VirtualBox, you should eliminate some of the default installed features. If you would want to interact with Virtual Box using Python, you would need to install the VirtualBox Python Support feature. Note however that you also would need to install "python core / win32api" prior to install Virtual Box. This feature is not needed in this course, so you don't have to install it .
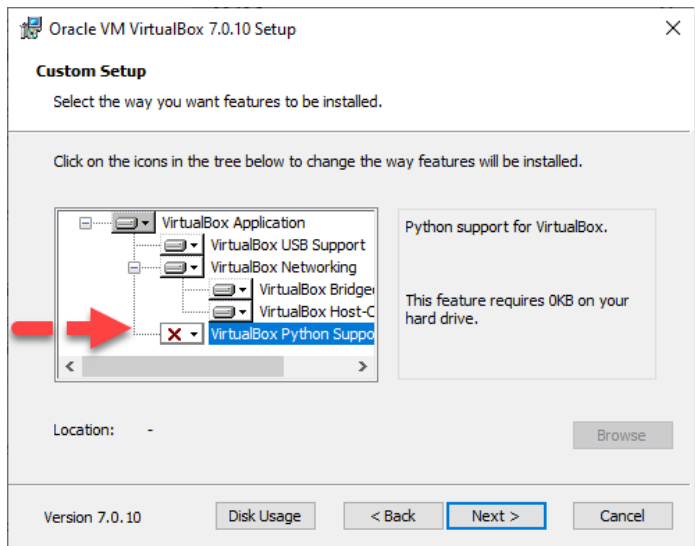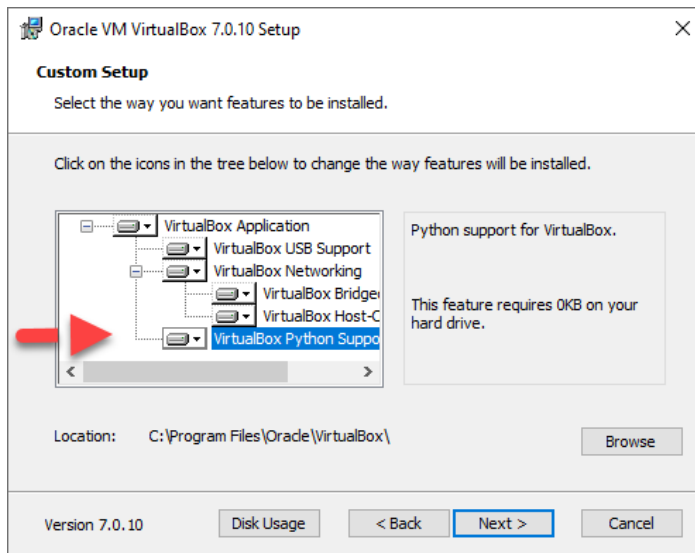
After successful installation you can lauch Virtual Box.
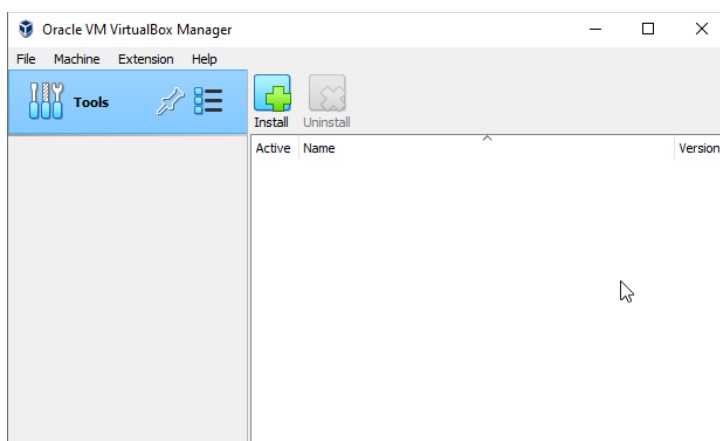
To install the extension pack, click on the install icon in the ribbon just under the main menu bar. Then browse to the location where you downloaded the extension pack. From there on: Install … next … OK.

```
-----------------------------------------------------------------
| 4.   Creating a VM                                            |
-----------------------------------------------------------------
```

We will create a virtual Ubuntu 22.04LTS desktop system with

- dual core cpu `// 1 core may not boot`
- 8 GB ram   `// can easily be adjusted afterwards`
- 40 GB hard drive. `// is very difficult to modify afterwards.`

In VirtualBox: main menu bar -> Machine -> New opens a dialog.

Do not mount the iso image of Ubuntu yet, as this will boot a live operating system from the ISO rather than install ubuntu onto your VM. When booting from the ISO, the OS can also be installed but skips the config of the keyboard. This is not funny for dutch/azerty people. However, the keyboard can be reconfigured later.
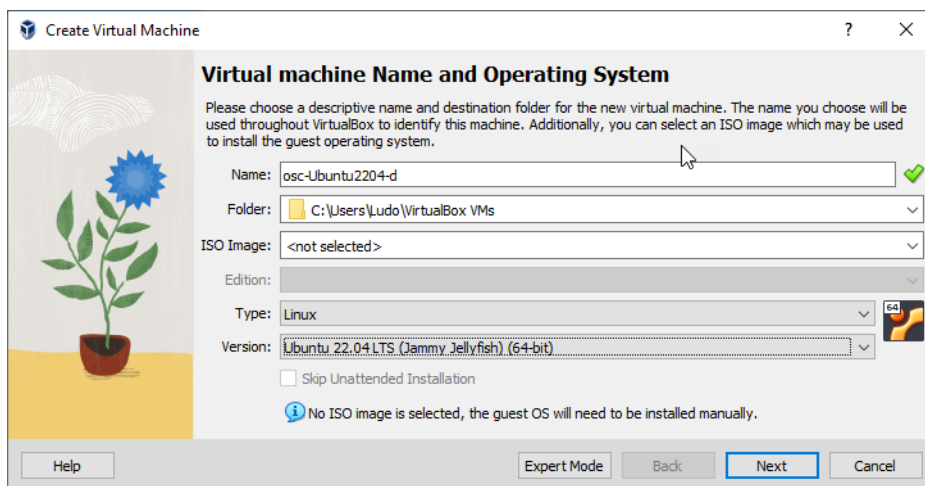

Click next:
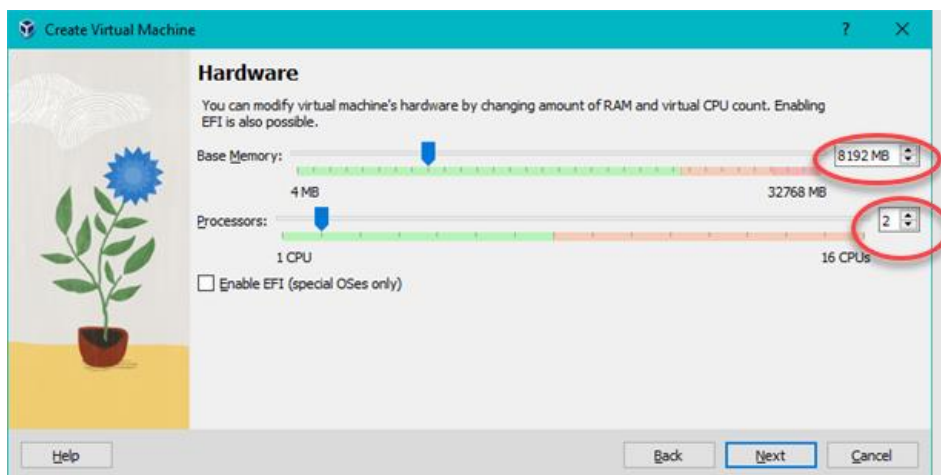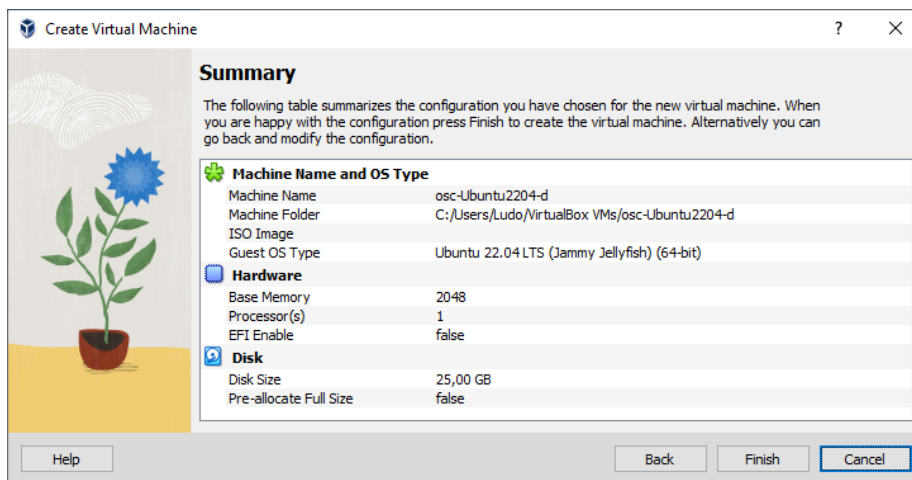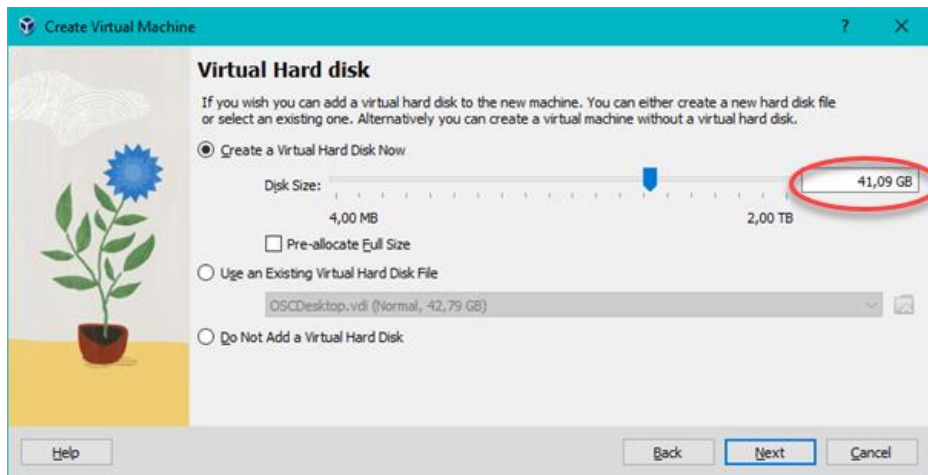
```
 ----------------------------------------------------------------
| 5.   Installing Ubuntu 22.04LTS Desktop.                       |
 ----------------------------------------------------------------
```

- First, download the iso from https://ubuntu.com/download/desktop
- Boot the VM. The VM will find no boot medium and prompts you to mount a bootable medium. Browse to your download folder and select the ISO file. Then click the 'Mount and Retry boot' button.

It is important that you download and install a version 22.04.x LTS. X was '1' when we prepared this text but is now '3'.

Why? Because the whole teaching team uses a 22.04.x LTS version. So we should be able to assist you when problems. At least we try very hard. LTS means 'Long Term Support' and Canonical will not change features overnight. For more info on Ubuntu releases and lifecycle:
https://ubuntu.com/about/release-cycle
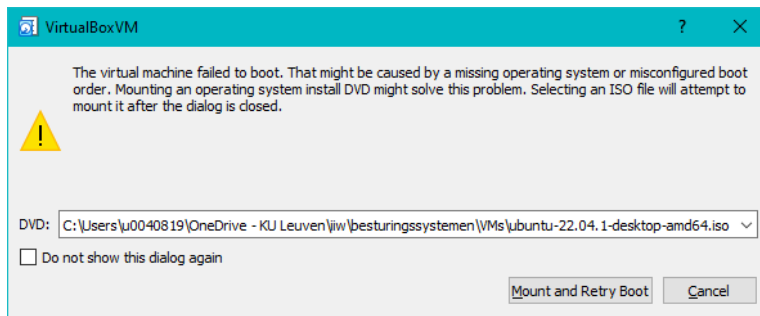
Reboot the VM and choose to *try or install Ubuntu* in the text-based menu interface. After a while you will be presented a GUI to install Ubuntu. Choose to install Ubuntu.
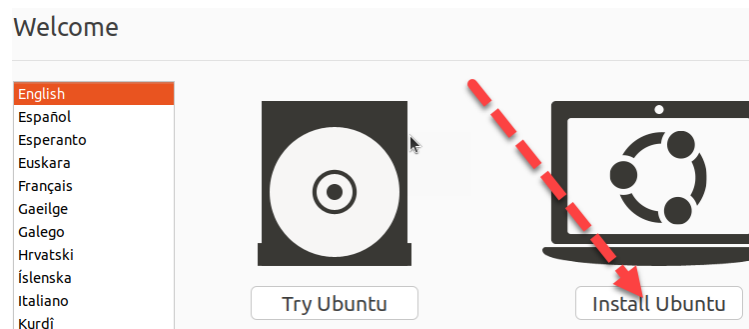
Configure your keyboard layout and language and choose to continue the installation with the default configured options (normal installation, download updates, erase disk and install Ubuntu) and continue with "install now" or "continue". (BE azerty keyboard = Belgian (Wang 724 AZERTY)
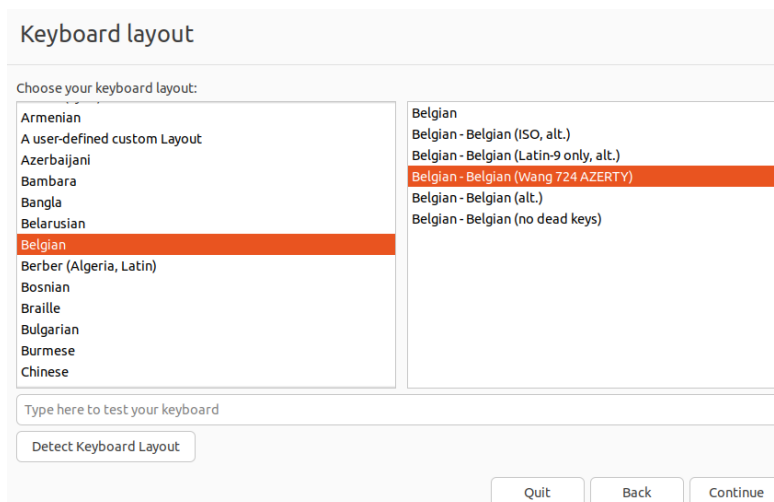
You can now create a computer name and user account with a name and password
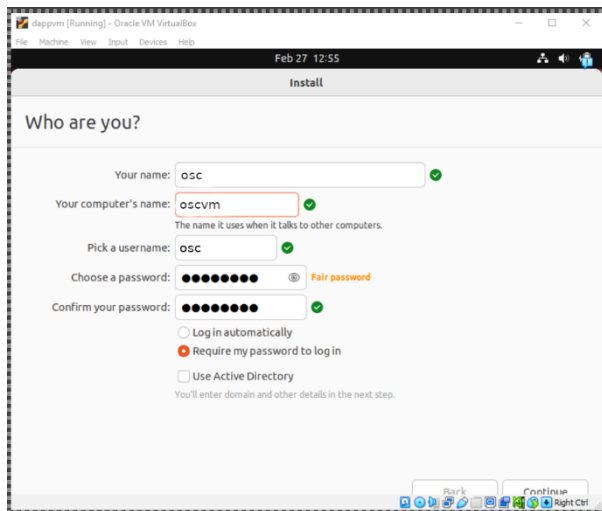
Choose names and passwords that you can remember and type easily. Your VM is protected from the outside world by your laptop and its operating system.

Choose a short username and password: you will have to enter it very often!

Relax & wait until the installation is finished. At the end, reboot to start Ubuntu Desktop.

Unmount the installation DVD (in the menu Devices) and reboot.

You can skip the start-up screen that wants to configure some bloatware and other software.

It is advised to install the updates when Ubuntu asks to update. Reboot at the end.

You can change the resolution of the screen on the desktop to a resolution fitting your laptop and put the VM in full screen.

One last thing to have a functional virtual environment is supporting copy paste between your native laptop OS and the VM. To achieve this, we need to install the guest additions.
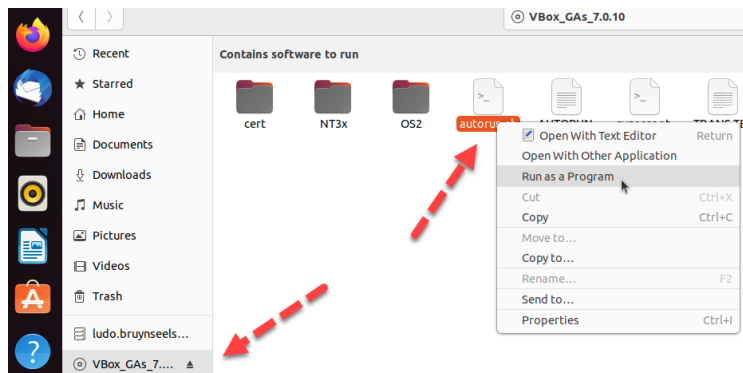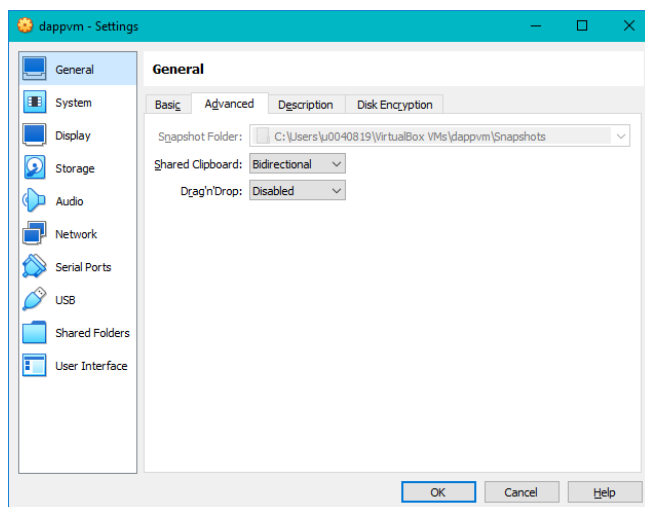
1. Insert Guest additions DVD iso image.
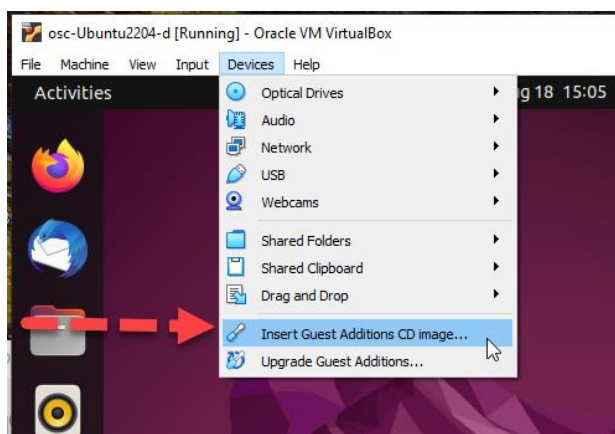

2. Open 'files'

3. Browse the contents of the cdrom and run autorun.sh (right click)

4. In the settings of your virtual machine, adjust following settings. After rebooting the virtual machine, you will be able to copy/paste between your guest and host operating system.

In Ubuntu the installed applications can be found using "show applications" at the left bottom of the screen. Select "terminal" from the applications overview to open a commandline interface (CLI).
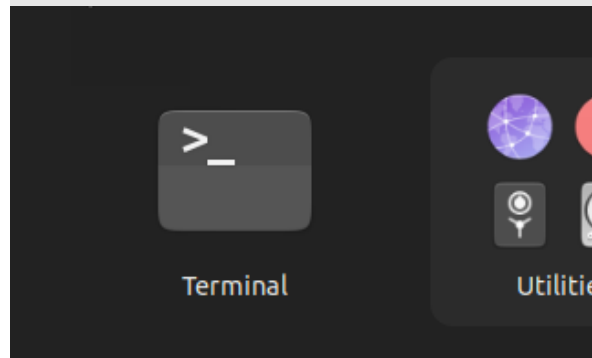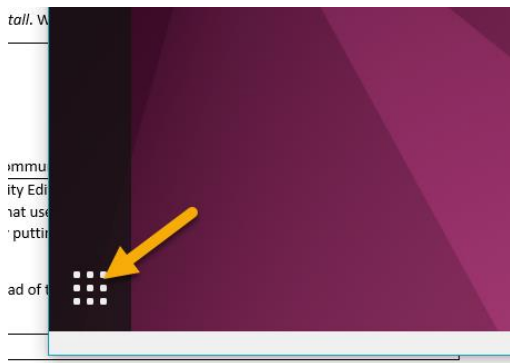
If you right click on the terminal icon, you can add it to your favorites. This may save some point & click later. You can now control your ubuntu both via the GUI and the Terminal.

```
------------------------------------------------------------------
| 6.   Setting up sshd on your ubuntu desktop                    |
------------------------------------------------------------------
```

To login to your VM via ssh or to transfer files to and from your VM, you need to install the ssh server on your ubuntu desktop. As such, your ssh client (e.g putty) or sftp client (e.g. filezilla) on your laptop's native operating system can connect to your VM. SSH uses port 22 by default.

We have also chosen the default port 22 on our native host OS. However, if you already run Linux natively with an sshd on your laptop, you might need to choose another port on the host OS (e.g. 2222). After all, only one process can listen on a reserved port.

Installing and starting the ssh server:

```
$ sudo apt update
$ sudo apt upgrade
$ sudo apt install openssh-server
$ sudo systemctl start ssh
$ sudo systemctl status ssh
```

You will also need to forward the local port 22 to your vm:
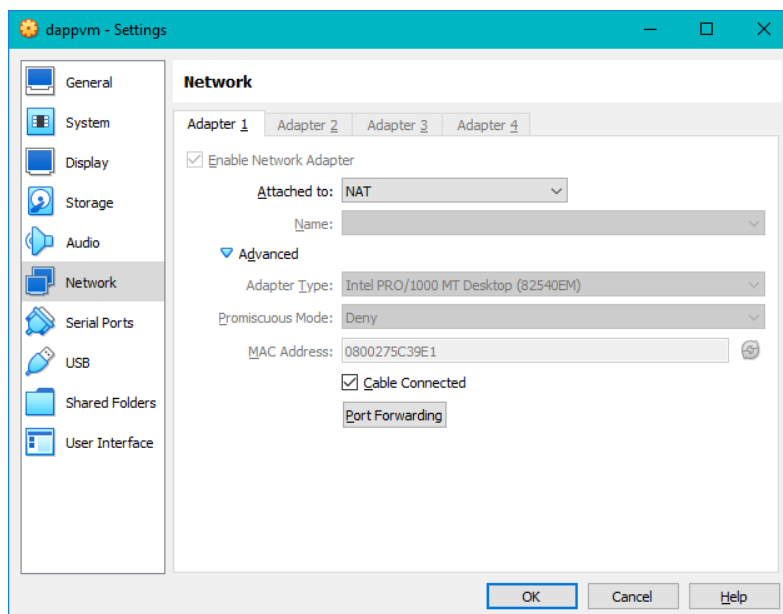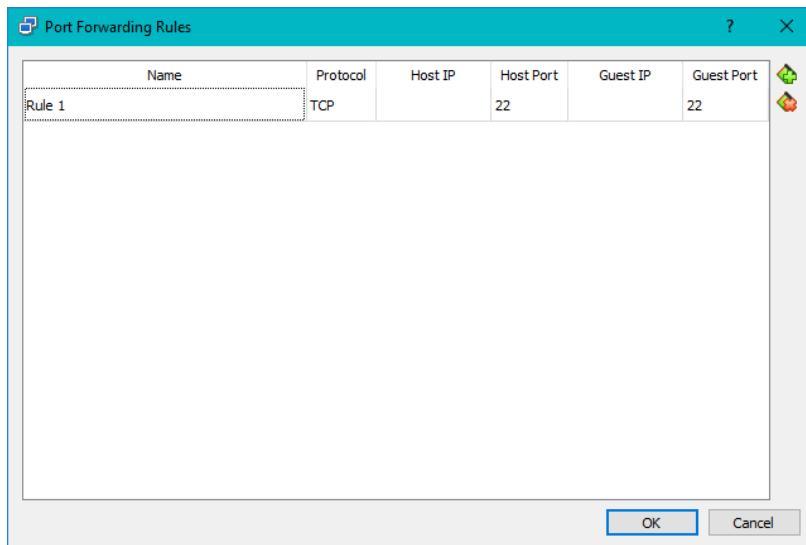
Testing the setup

SSH into your vm from your native host OS, e.g. using the ssh command line client:

```
C:\> ssh osc@localhost
```

You can verify the installation and version of the required tools using.

So far, so good. We now have a working ubuntu desktop system.

```
----------------------------------------------------------------------
| 7.   Configuring sshkeys                                           |
----------------------------------------------------------------------
```

Access to remote git repositories requires authentication of your identity, and to avoid relying on passwords, we will use sshkeys which are based on asymmetric encryption (public and private keys)

1. Open a terminal session
2. Type ssh-keygen -t ed25519
3. Accept the proposed filename (unless you have a good reason to choose another one)
4. You may protect your keys with a passphrase but you may leave this empty. Be aware : if you choose a passphrase, you must remember it later! If you leave this empty, nobody will nag about it.
5. cd ~/.ssh (the ~ sign is short for your home folder) There you find your private key and your public key. We will need them later to access remote git repositories.

```
----------------------------------------------------------------------
| 8.   Conclusion                                                    |
----------------------------------------------------------------------
```

This first practical work was a quick tutorial for students new to Virtual Box and Ubuntu desktop. You should now be able to fluently work in the Ubuntu desktop environment and develop C applications with CLion in the next practical sessions