

# Operating Systems Linux Installation

Florido Paganelli  
Lund University  
florido.paganelli@hep.lu.se

# Outline

- Part A: Theory and practice of computers and operating systems
  - What is a computer?
    - A computer as a finite state machine
    - Brief history and architecture overview
  - Introduction to virtualization
    - Installing the course virtual machine
    - Basic Ubuntu use
- Part B: Installing GNU/Linux in a Virtual Machine
  - Creating a virtual machine with VirtualBox
  - Operating Systems
    - Why do we need it?
    - Examples of operating systems
    - Linux and distributions

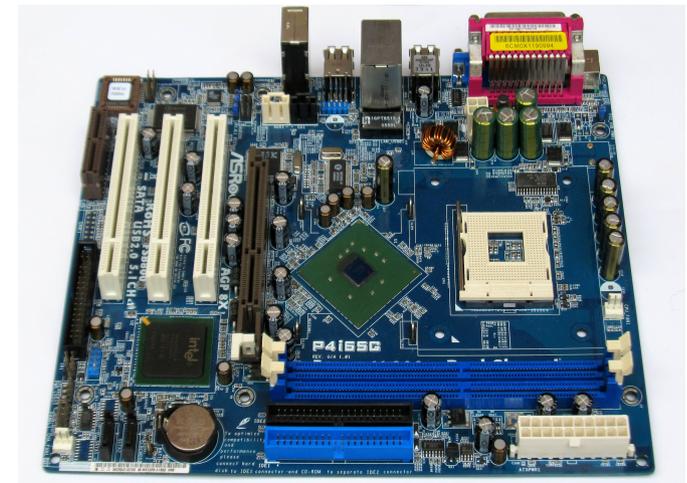
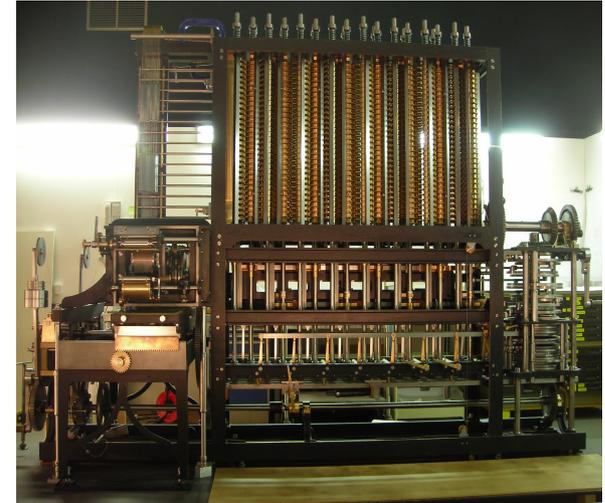
# Part A: Theory and practice of computer systems

- Goals:
  - Understanding Hardware and Software
  - Understanding Virtualization
  - Learn how to use the Course Virtual Machine

This part is important for each and every tutorial session.

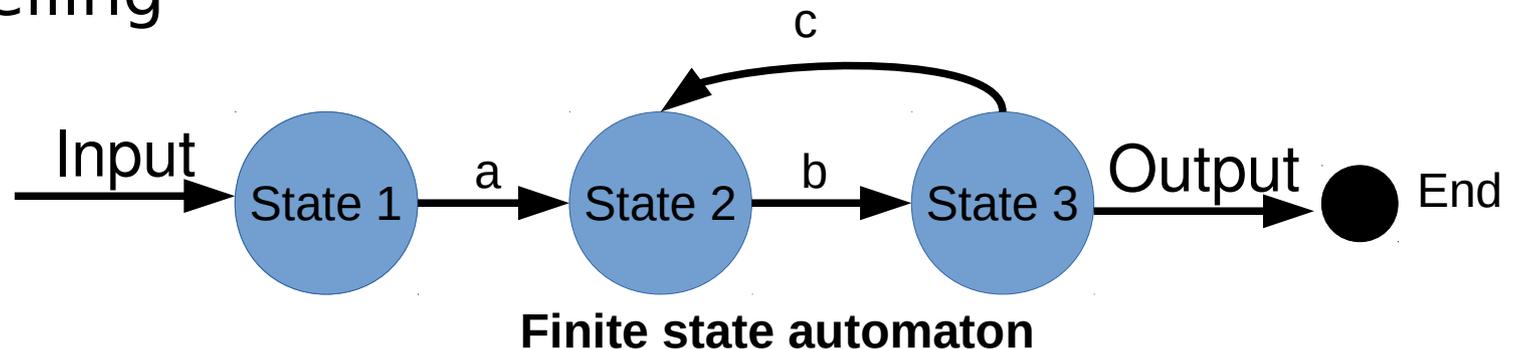
# What is a computer?

- A **programmable machine** that can store, retrieve and process **information**.
- **Information** can be, for example
  - Data
  - Simple and complex operations
- Most of modern computers are based on electronic circuits. Whatever we **program** these circuits to do for our needs is usually called **information processing**.



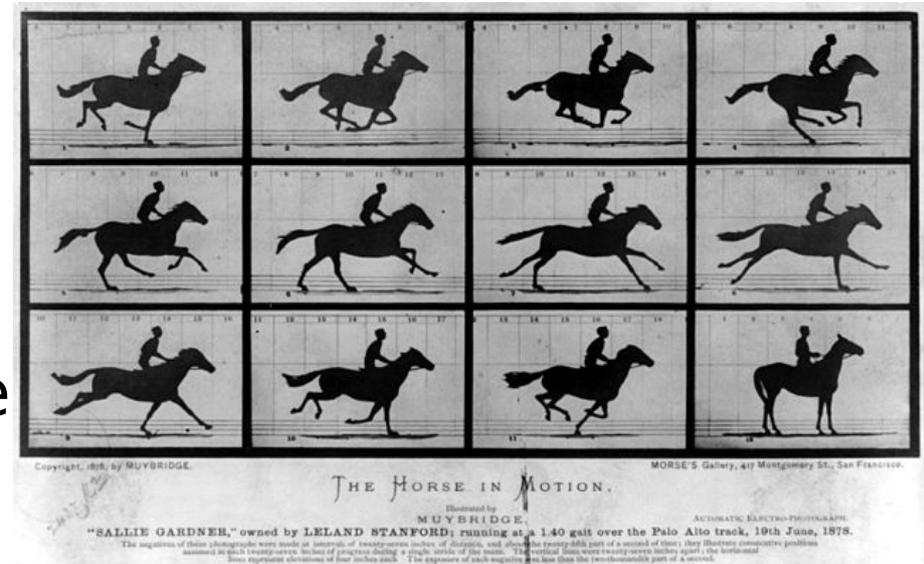
# Finite State Machine

- A mathematical object that represents a sequence of events and their possible outcomes
- You can use this model for:
  - Evolution of a closed system
  - Card games
  - Movies cutting
  - Storytelling
  - Music

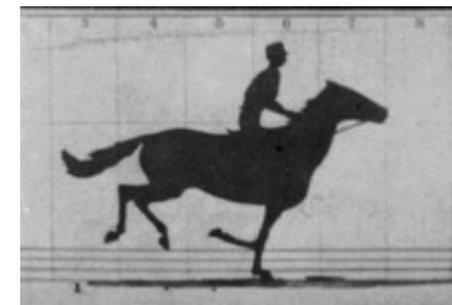


# Finite state machines

- A modern computer is modeled by a finite state machine.
  - A “state” is the contents of “memories” of the machine
  - If we could stop time, the computer would stay in a defined state
- ▶ A state can be restored by restoring the machine's “memories”
  - Examples: hibernation, **virtualization (some about it later)**



States



States during execution  
(only visible in libreoffice)

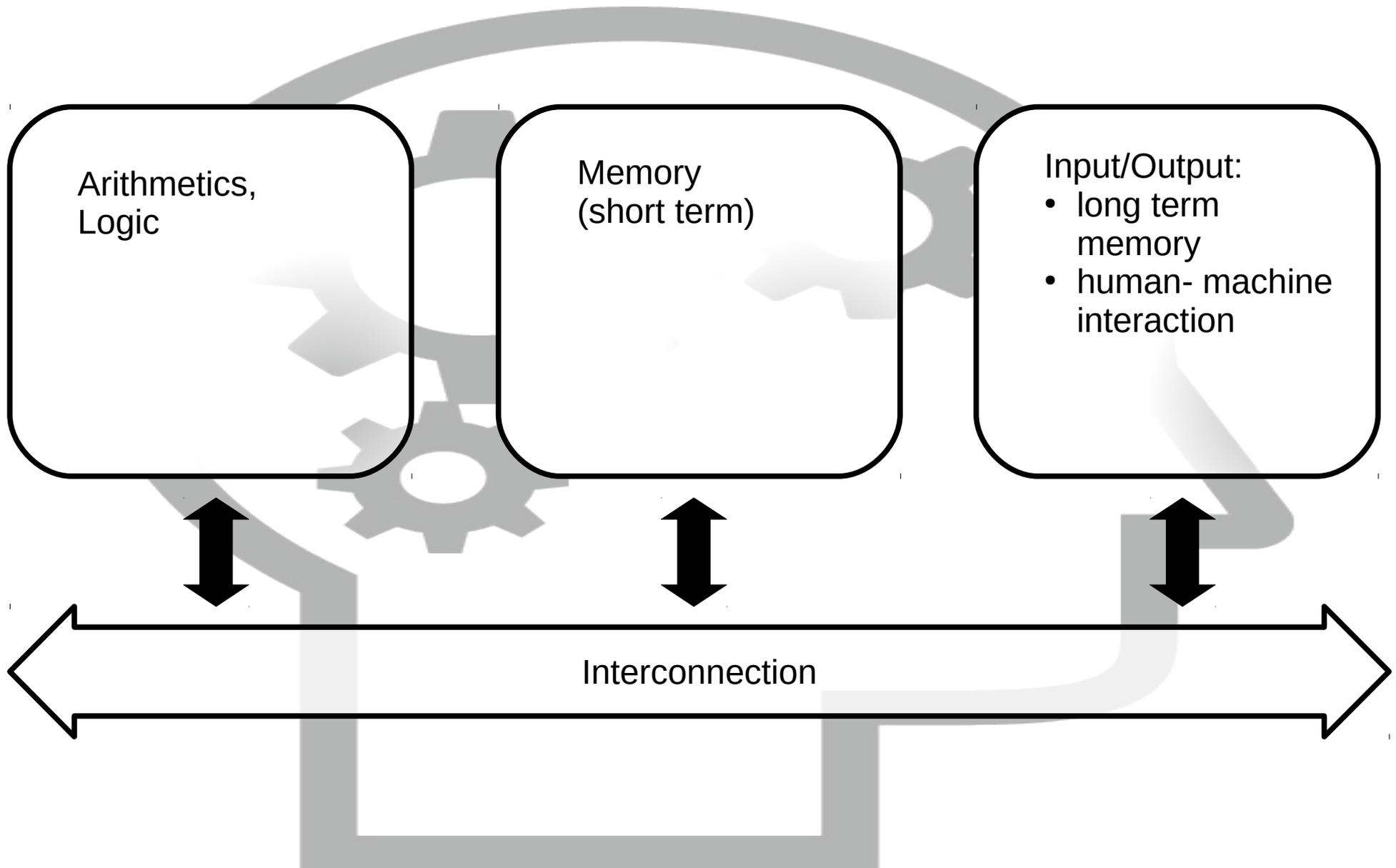


# VERY Brief history of computing

- 1945 Von Neumann's paper[1] defines the modern computer architecture
- 1960-70 Various researchers start improving the communication between components
- 2019: We still use the same basic approach, with lots of improvements, and increased complexity.

[1] First Draft of a Report on the EDVAC, John von Neumann, Contract No. W-670-ORD-4926, June 30, 1945

# Von Neumann-based modern architecture



# Von Neumann-based modern architecture

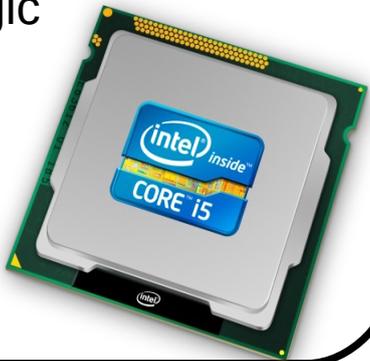
- Arithmetics and Logic: Brain ability to process numbers and operations
- Memory:
  - Short term: used in quick operations
  - Long term: memories
- Interconnection: neurons, the spine or the nervous system
- Interaction with external world: the senses, like sight, smell, taste, touch, ...

# Hardware:

## electronic components of a computer

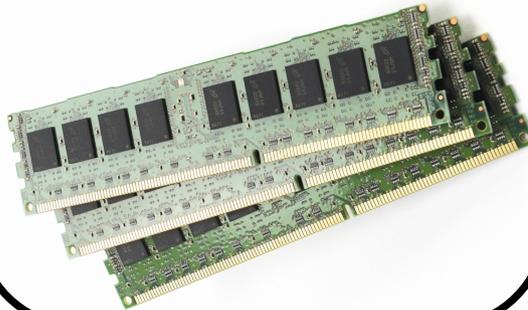
### CPU, Processor

Arithmetics,  
Logic



### RAM

Memory used only  
when machine  
**powered on**



### DEVICES

Input/Output



Long term memory  
persists also when  
machine powered off



Interconnection

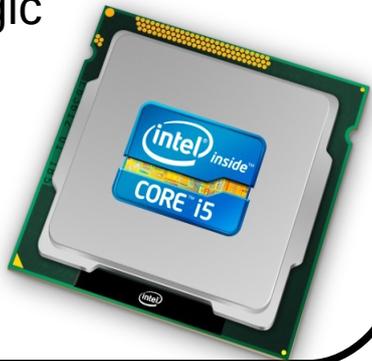
Motherboard's BUS

# Hardware:

electronic components of a computer

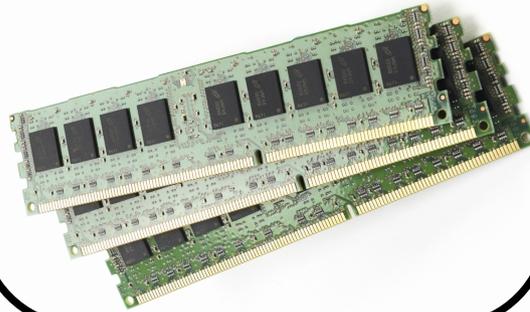
## CPU, Processor

Arithmetics,  
Logic



## RAM

Memory used only  
when machine  
**powered on**



## DEVICES

Input/Output



Motherboard's BUS

# Hardware:

## electronic components of a computer

- Arithmetics and Logic: CPU (Central Processing Unit)
- Memory:
  - Short term storage: RAM (Random Access Memory) only works when powered
  - Long term storage: magnetic discs / USB dongles / cloud storage . Works also when not directly powered.
- Interconnection: BUS(PCIE,SATA,USB)
- Interaction with external world: devices like Network cards, Screen, Keyboard, Touch screen...

# Software

- Anything that is designed to **run** or **execute** in a computer , that is, the information that is processed by the hardware.
- Can be of different kinds:
  - **System software:** used to interact directly with the hardware, usually as an *interface* between the hardware and other kind of software.  
Examples: device drivers, operating systems, firmware...



- **User software:** something with which a user interacts directly to perform a task. Also called **Programs** or **Applications** (shortened: **Apps**). It is usually run inside an operating system.



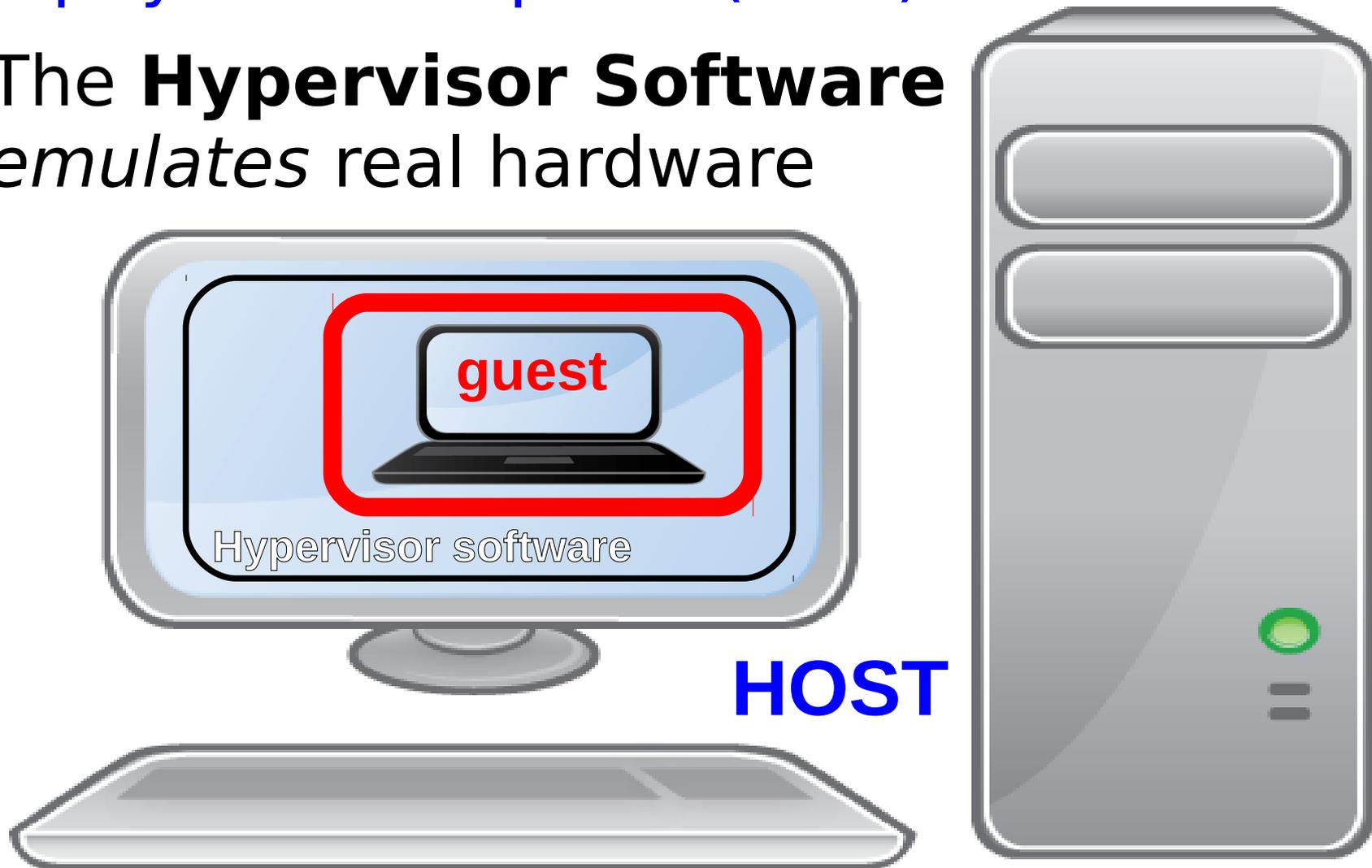
- **Development software:** software that is used to develop and create other software.  
Examples: SDK (Software Development Kit), libraries, compilers...

# Hardware-Software equivalence

- Everything that can be modeled via software can be created in hardware and vice-versa
- This poses the foundation for machine **simulation** and **emulation**
  - **Simulation**: software that behaves exactly like some piece of hardware, internally and externally. For prototypes and testing
  - **Emulation**: write software that whose external behaviour is like a piece of hardware. The internals can differ. It "*pretends*" to be some hardware.

# Virtualization

- Running a **virtual computer (guest)** inside a **physical computer (host)**
- The **Hypervisor Software** *emulates* real hardware



# Virtualization

- Running a **computer (guest)** inside a computer (**host**)
- The **guest** machine is usually called **Virtual Machine**.
- The **Host** machine manages the guest machine using something called **Hypervisor**
- The **host** offers software simulated or emulated hardware, plus it can offer **real** hardware to the guest machine
- The **guest** machine sees all the software simulated/emulated/virtualized hardware as it was real hardware, but it can also be aware that it is virtualized to boost performance

# Using the course VM

- Please follow the slides  
*Course VM Manual*  
as far as you can go.
- We will continue with the next topic in the next hour.

# Part B: Installing GNU/Linux in a Virtual Machine

- Goals:
  - Create a *virtual machine*
  - Understand what an *operating system* is
  - Install *GNU/Linux*

This part is important for the homework.

With what you learn here you can test any operating system on your laptop by running it in a virtual machine.

# Prerequisites

- Your machine needs to have at least 4GB of RAM. 8GB RAM is recommended.
- You need at least 40GB of free harddisk space.
- On Apple computers you need to enable the mouse right click, see [https://support.apple.com/kb/PH18768?locale=en\\_US](https://support.apple.com/kb/PH18768?locale=en_US)
- **This step is not required in the lab. VirtualBox is already installed:**

Install Oracle VirtualBox for your operating system.

Download it from

<http://www.virtualbox.org>

- If you already have GNU/Linux installed, the open source version of VirtualBox can be found in the distribution's repositories.

# Ex. B1: Create a virtual machine

Goal: create a virtual machine to run Ubuntu Linux.

Use the following information:

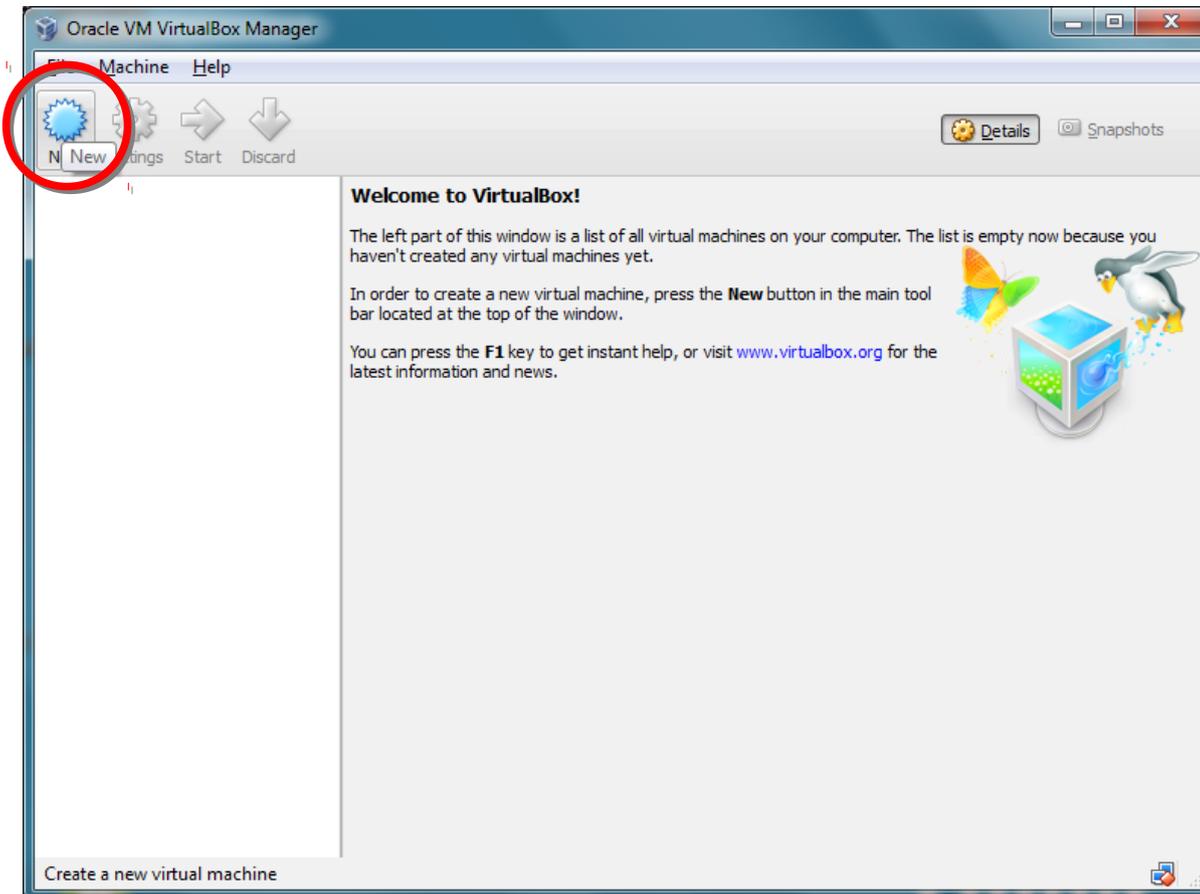
<b>VM Name</b>	MyLubuntuVM
<b>Operating System Type</b>	Linux
<b>Version</b>	Ubuntu (32 bit)
<b>memory (RAM)</b>	1536 MB (that is, 1,5 Gigabytes)

The slides that follow will guide you through the process. A more general and detailed tutorial can be found on VirtualBox official documentation:

<https://www.virtualbox.org/manual/ch01.html#gui-createvm>

# Ex. B1: Create a virtual machine

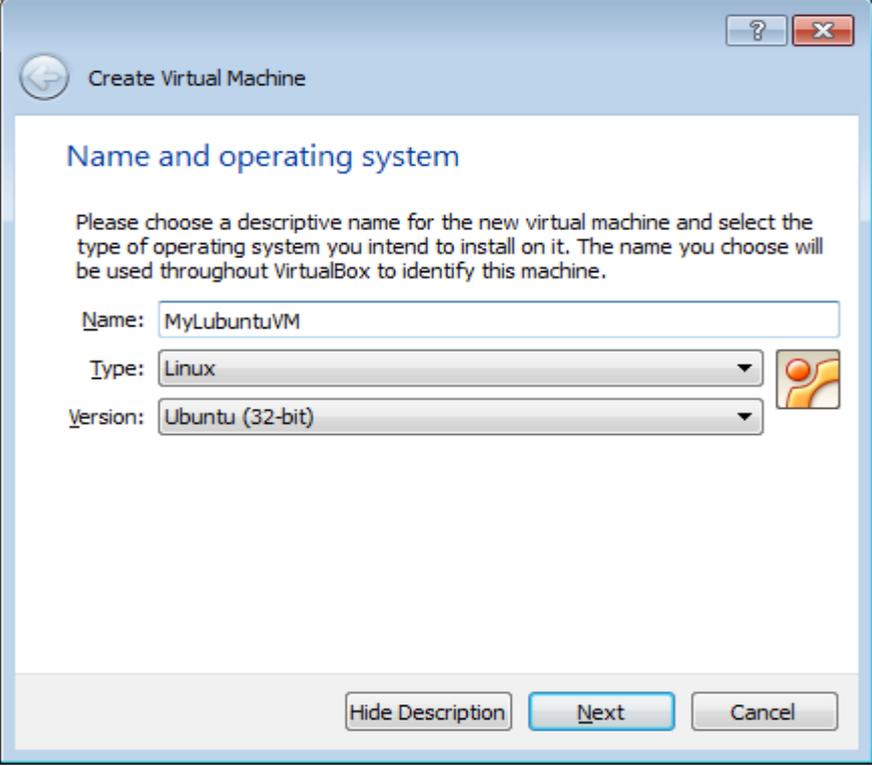
B1.1. Open virtualbox . Click on "New"



Note: following these slides will give you a good introduction. A more general and detailed tutorial can be found on VirtualBox official documentation:  
<https://www.virtualbox.org/manual/ch01.html#gui-createvm>

# Ex. B1: Create a virtual machine

B1.2 Insert the following information when asked:



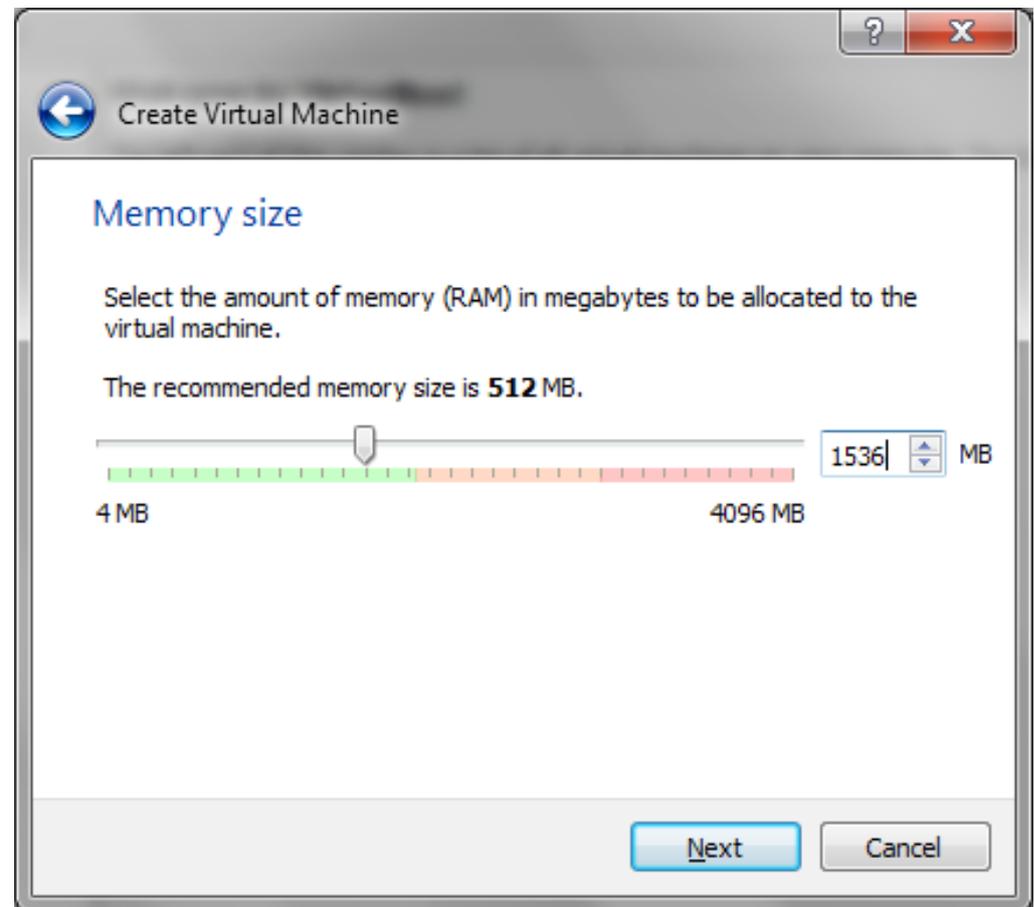
The screenshot shows the 'Create Virtual Machine' dialog box. The title bar reads 'Create Virtual Machine'. Below the title bar, there is a section titled 'Name and operating system'. The text below this section reads: 'Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.' The 'Name' field contains 'MyLubuntuVM'. The 'Type' dropdown menu is set to 'Linux'. The 'Version' dropdown menu is set to 'Ubuntu (32-bit)'. At the bottom of the dialog, there are three buttons: 'Hide Description', 'Next', and 'Cancel'.

# Ex. B1: Create a virtual machine

## B1.3 Set memory size. Suggested: 1536 MB (1.5GB as a multiple of 8)

*Q: Why this weird number?*

A: I calculated that this is the optimal amount of memory we need for the tasks in the course. You may change it to something else, but the suggestion is **never** to take more than half of the available RAM.



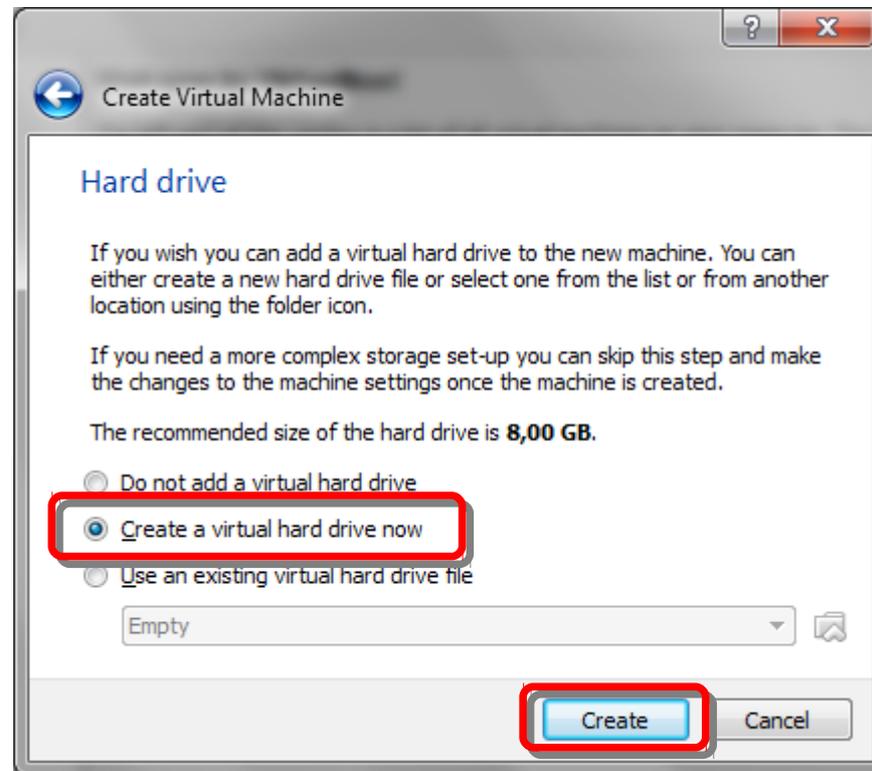
# Ex. B1: Create a virtual machine creating a virtual disk

- Create a **new virtual harddisk** for the machine (step 4 in <https://www.virtualbox.org/manual/ch01.html#gui-createvm>)
- When asked, choose "Create a virtual hard drive now" and click "Create"
- Use the following parameters:

<b>Disk Type</b>	VDI (VirtualBox Disk Image)
<b>Storage on physical hard drive</b>	Dynamically allocated
<b>Name</b>	MyLubuntuLinux
<b>Size</b>	30,00 GB

# Ex. B1: Create a virtual machine creating a virtual disk

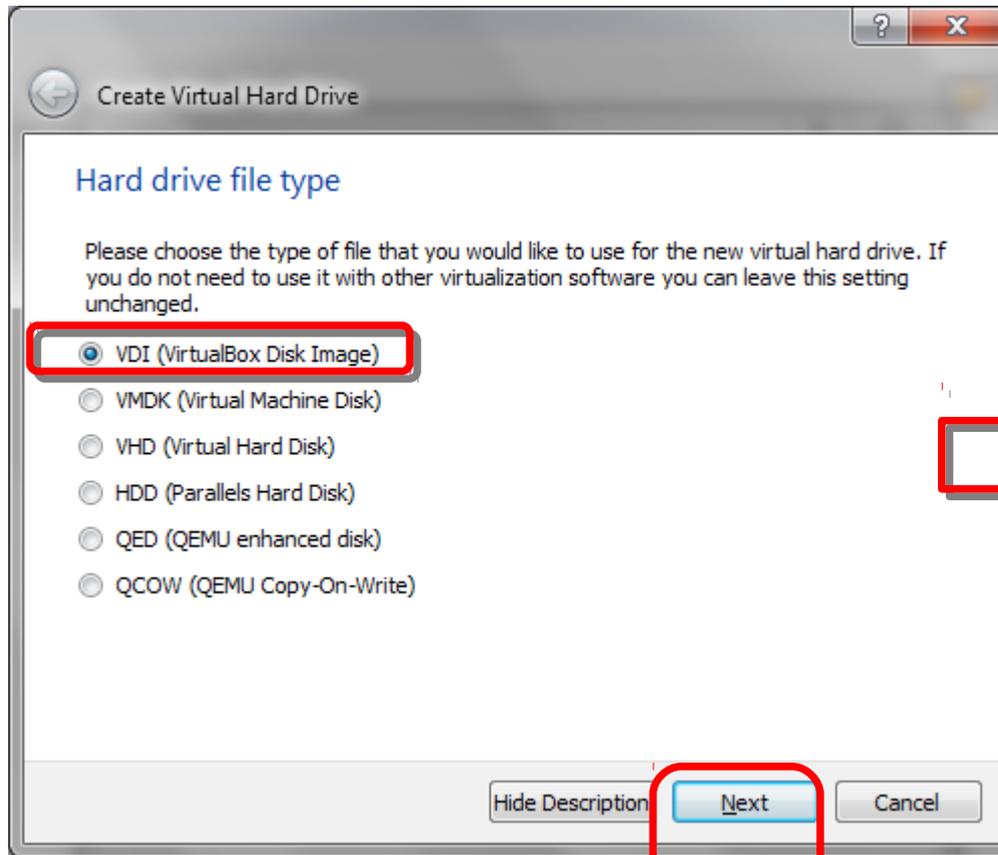
B1.4 Create a **new virtual harddisk** for the machine, click create as shown in the picture.



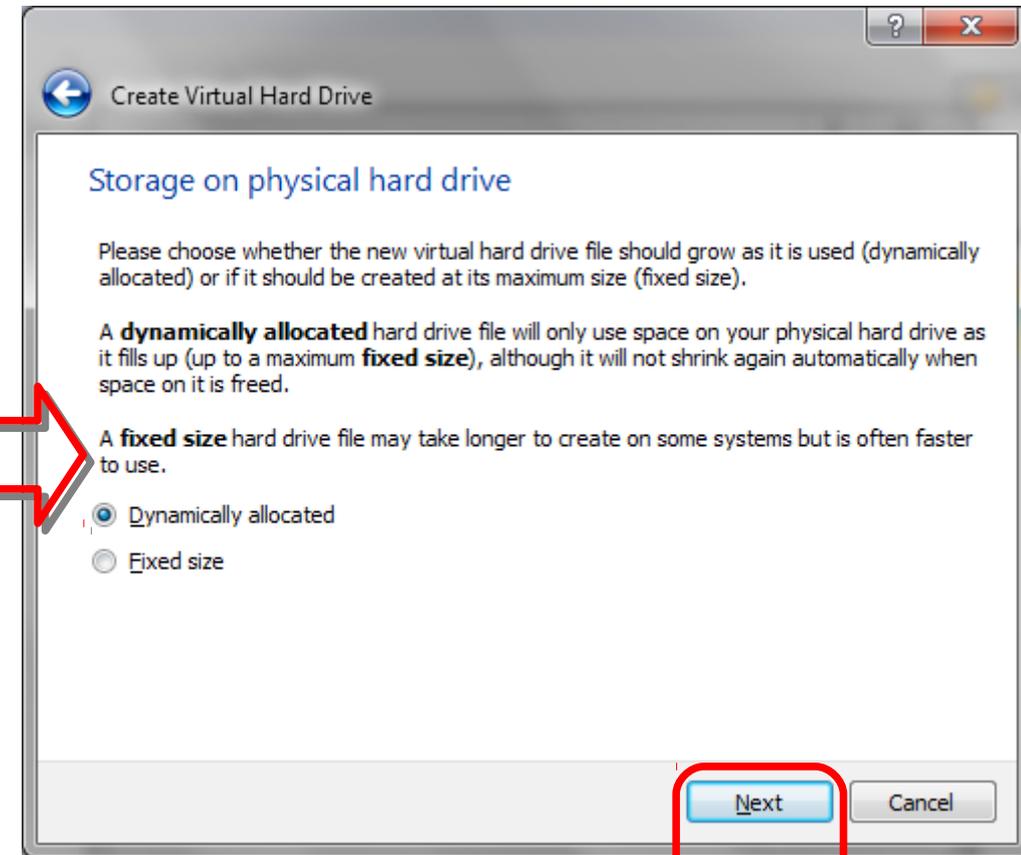
# Ex. B1: Create a virtual machine creating a virtual disk

B1.5. Select VDI as disk type

B1.6. Choose "Dynamically allocated" (saves disk space)



B1.5



B1.6

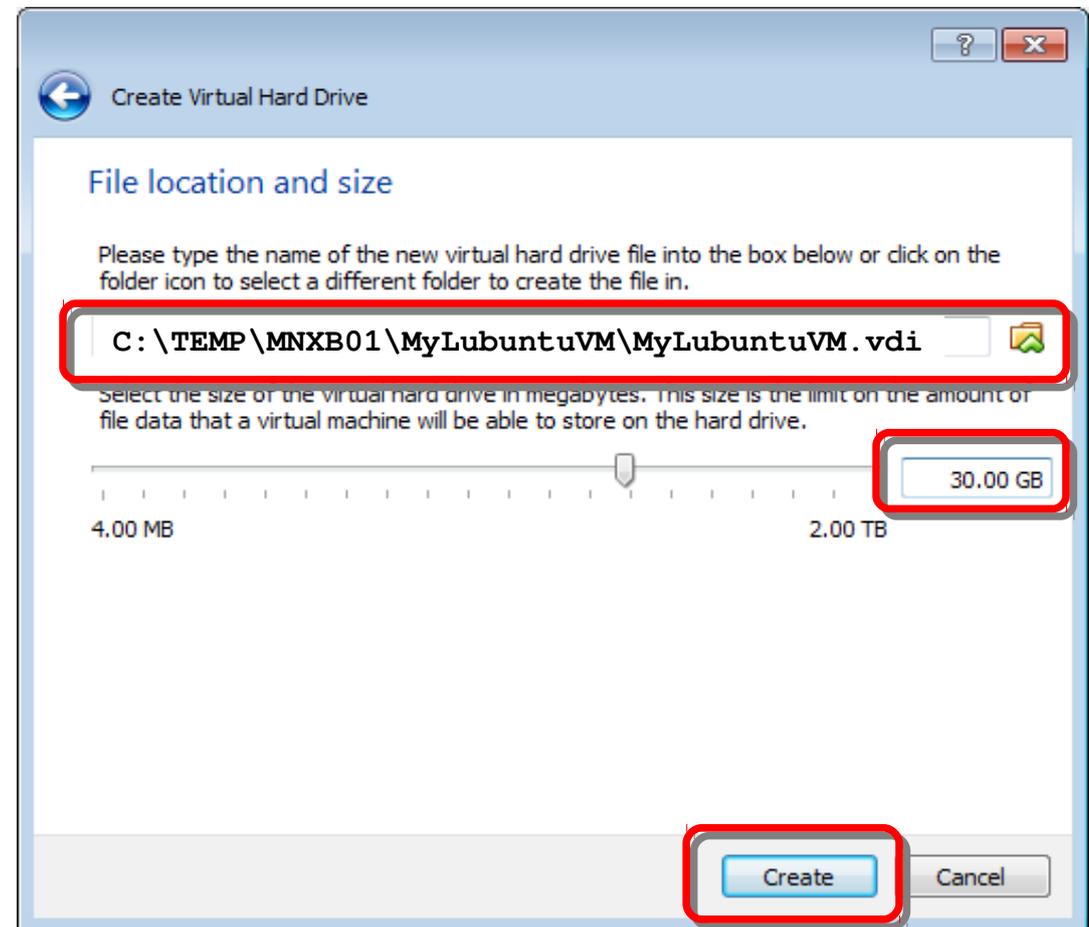
# Ex. B1: Create a virtual machine creating a virtual disk

B1.7. If on a Lab machine, change the haddisk path to be:

`C:\TEMP\MNXB01\MyLubuntuVM\MyLubuntuVM.vdi`

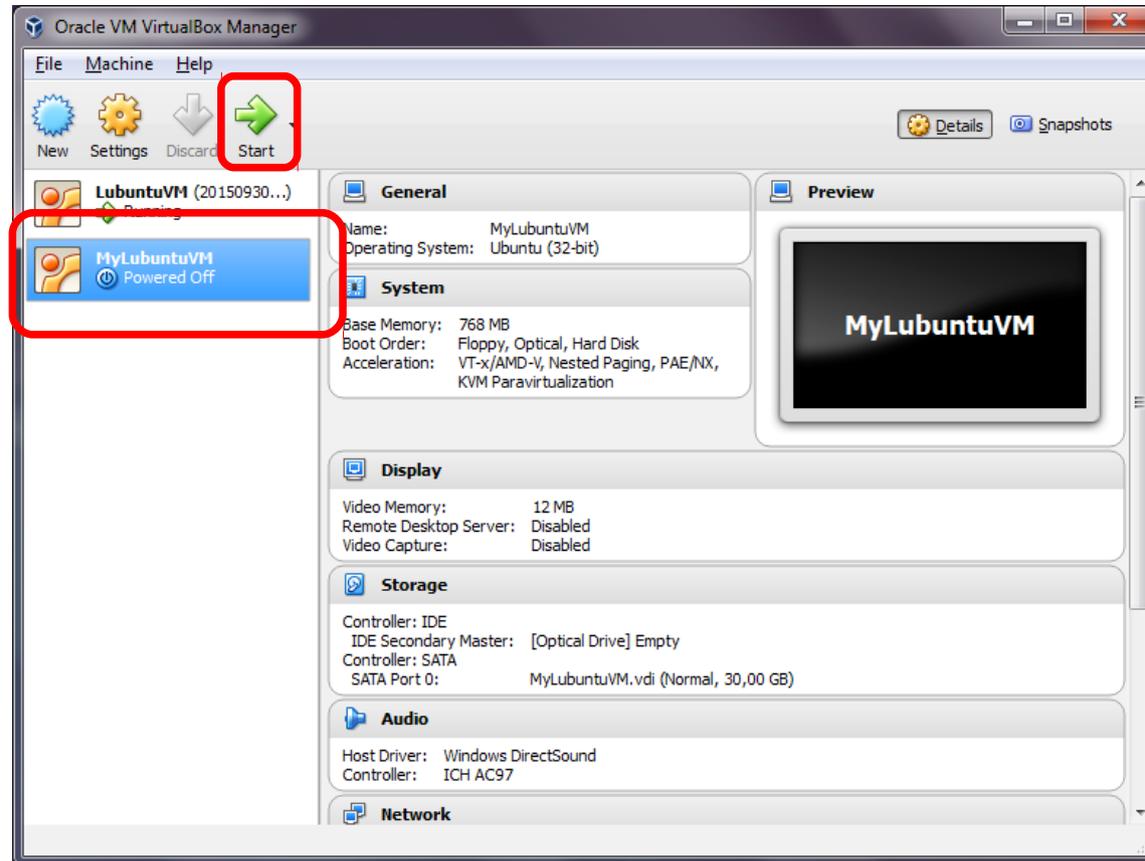
- *You may need to create the MyLubuntuVM folder.*
- *On your laptop you can leave this path to the suggested one.*

6.8. Set the disk size as shown in the picture and create the disk:

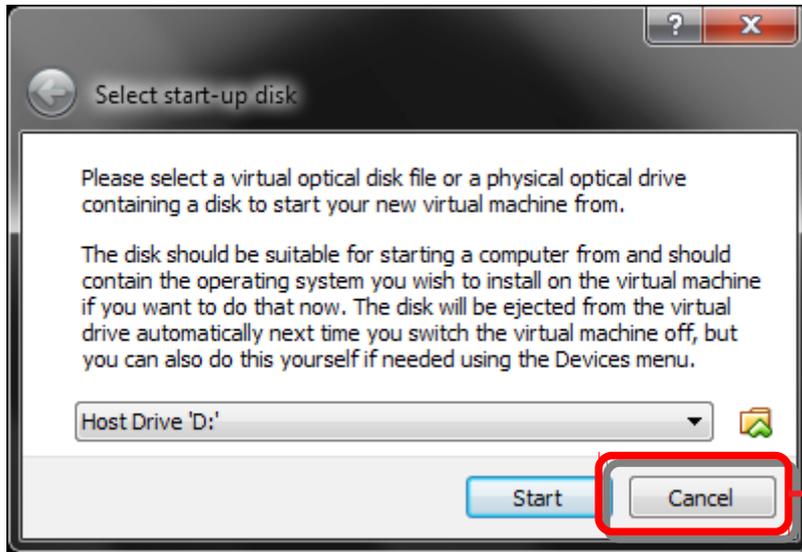


# Ex. B1: Create a virtual machine - start it!

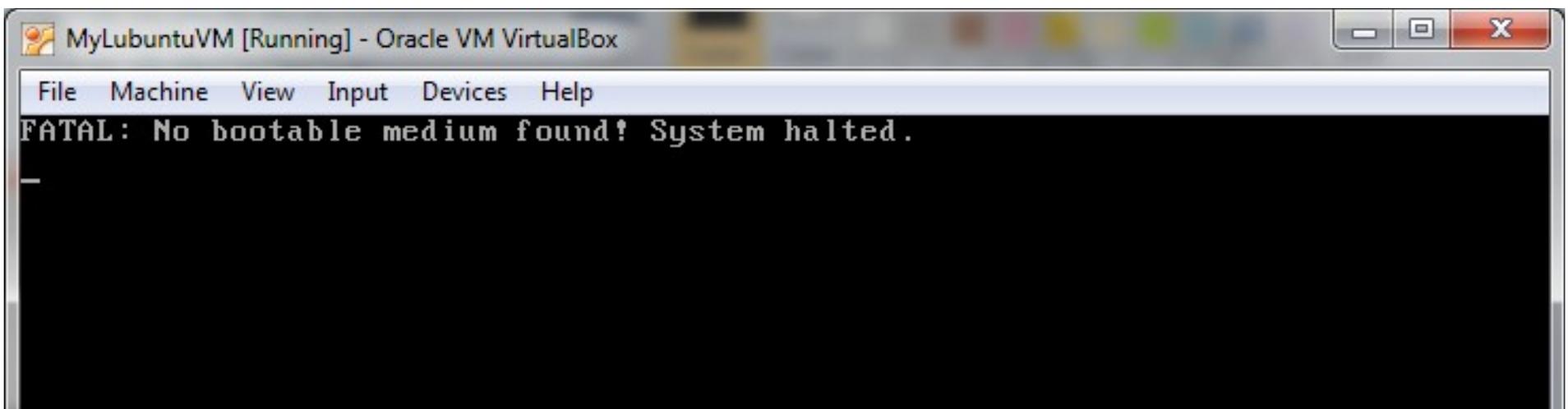
B1.9. Select and Start the virtual machine.  
What happens? Discuss with the teacher.  
(Hint: check slide 15)



# Ex. B1: Create a virtual machine - Missing boot device

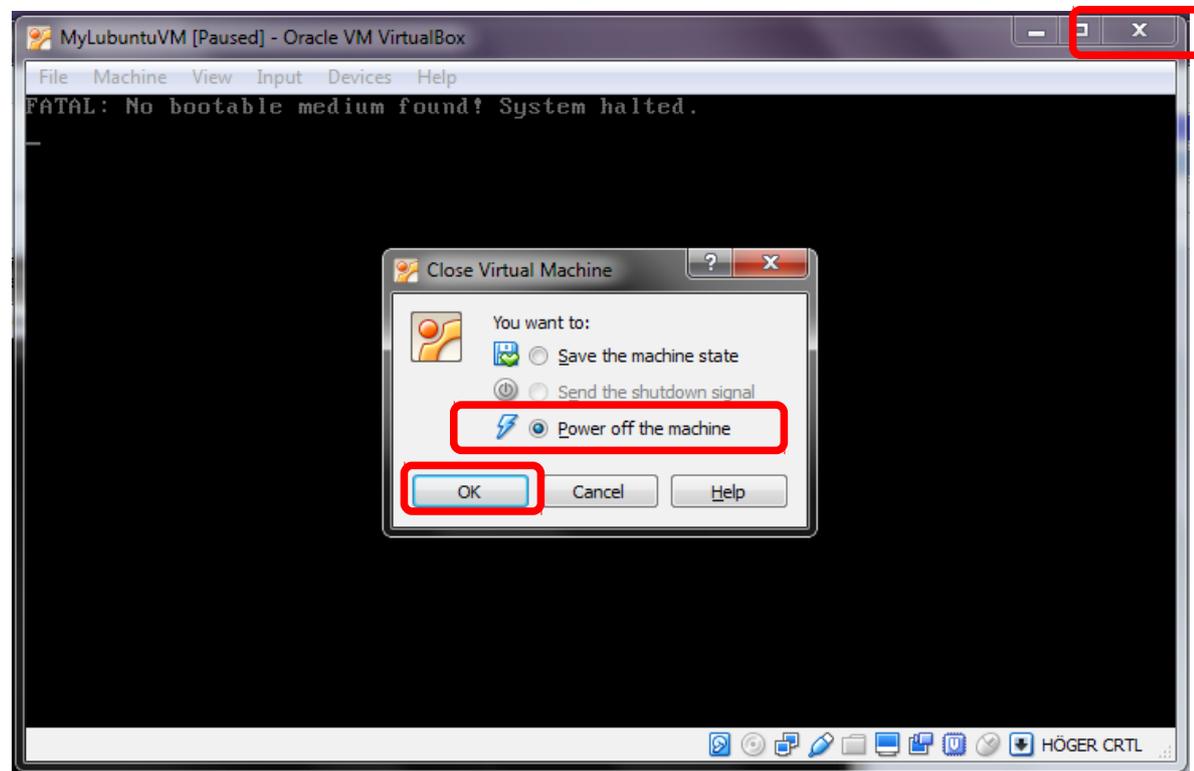


Click on the cancel button.



# Ex. B1: Create a virtual machine - stop it!

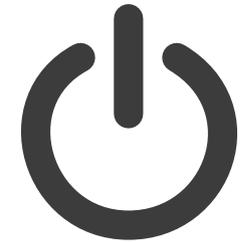
B1.10. Close the virtual machine by clicking the close window (X) button. Choose *power off the machine* and then OK when asked.



**OBS!** : Only use this method if the machine has no operating system installed. This method may cause errors on the virtual machine disk.

# "Booting" a computer (simplified)

- Booting: **starting** a computer. It usually happens when one turns the machine power button on.
- During the *boot process* the computer executes these steps, in order:

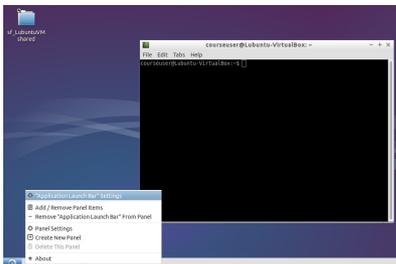


 1. Gives power to all the hardware and runs hardware checks

2. Chooses a *device* where the *Operating System* is *installed* and *boots* (starts) the operating system from that device



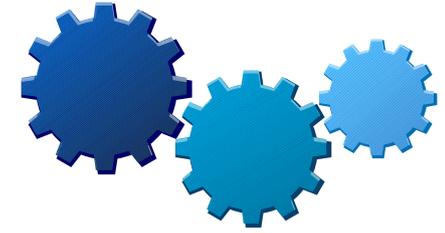
3. Loads the operating system *user interface software*, for example a graphical environment or a terminal.



4. Presents a *login prompt* to the user



# The operating system



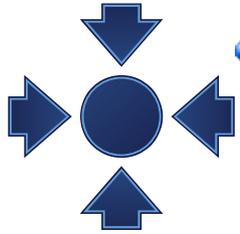
- Is a **collection of programs** running in your computer all the time it is turned on.
- **Orchestrates** the interaction between all components of a computer.
- Usually allows the machine to run **multiple programs** at the same time (multitasking).
- It is usually **installed** on a **long-term storage memory** , typically an HardDisk or a ROM (for example in mobile phones), but can be on a CD, a USB pen...
- The kind of operating system we will see in this course is generally meant to bring the machine “closer” to the **user** – not only allows the machine to **operate**, but also the user **to operate** the machine.

# Examples of operating systems

- Microsoft Windows
- Mac OS X
- UNIX
- GNU/Linux
- Android
- Symbian
- ...



# Basics of operating systems



- **Kernel:** a program more important than the others. It's a **software** that is the **core** of an operating system.

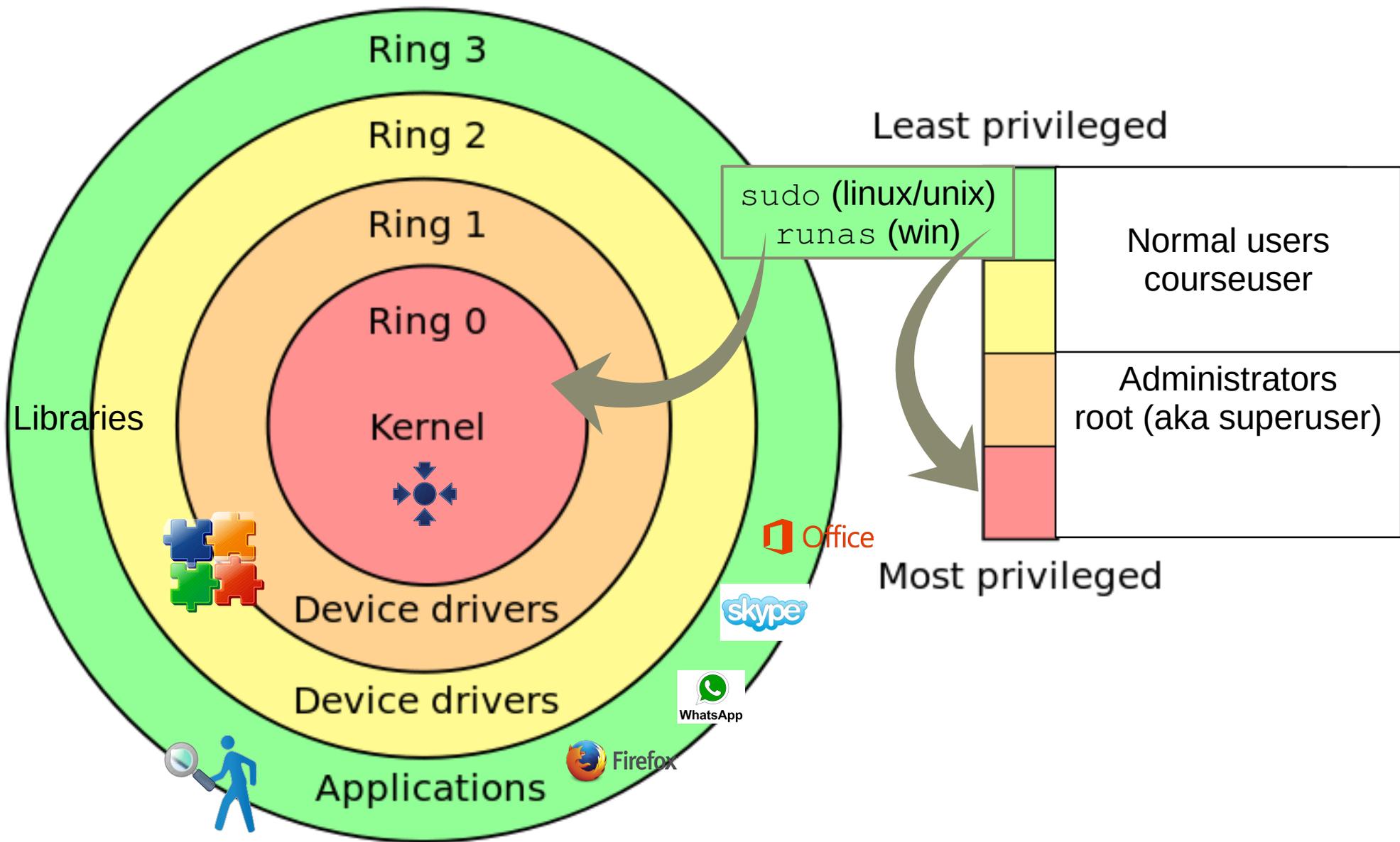


- **Drivers or Modules:** set of programs that allow the kernel to interact directly with the **hardware**



- **User interfaces:** set of programs allowing the users to communicate with the computer and use **software**.

# Security: Protection Rings



# Security: Protection Rings

- An operating system is organized such that an application cannot write on the other application's memory space.
- Additionally, a **three-layered architecture** organizes memory requests such that memory access is controlled according to protection rings:
  - the core **Ring 0** belongs to the kernel, who orchestrates the system. Nobody but the kernel can access its memory. Only the *superuser* (root or administrator) can interact with its state (example: power on/off).
  - **Ring 1 and 2** are for programs that access the hardware and interact with the kernel directly for performance reasons. Some may write the kernel memory directly, some not.
    - **Ring 1:** *Kernel modules* that usually talk/write directly to the hardware (example: turn on webcam)
    - **Ring 2:** *Device drivers* that allow normal users to interact with the Kernel modules (example: send my webcast using the webcam)
  - **Ring 3**, The external layer which is the one where we normal users (like courseuser) run our programs.
- **Privilege escalation:** A normal user can *switch* to an higher privilege by becoming administrator (*root* in Linux/unix) using the applications **sudo** [SuperUserDO] in Linux/Unix or **runas** in windows, so to change the kernel and system configuration files.
  - In some cases privilege escalation is transparent to the user: when we power on or off our computer, a request is sent from our user to the system to become superuser just to shut down the system.

# GNU/Linux or Linux

- One of the **full community based** accessible operating system, based on UNIX (proprietary)
- GNU (GNU's not Unix!) project: community that wrote many of the **drivers, basic tools and user interfaces.**  
Founder Richard Stallman (father of Free Software)
- Linux: the operating system whose **kernel** was developed by Linus Torvalds



GNU/Linux  
P o w e r e d



<<-- Tux the penguin,  
the GNU/Linux mascotte

# Linux, why we're using it

- Popular **free (as in free beer)** alternative to many proprietary operating systems
- **Free access (as free software)** to source code: anybody has **right to**
  - **Run the programs**
  - **Read and modify the programs**
  - **Redistribute modifications to friends**
  - **Propose changes to the community for everybody to benefit**
- Components developed by many universities (e.g. MIT) on a free-for-all knowledge basis
- Accessible libraries to build software upon
  - most of scientific software is written on it
- After 20 years, it still "scales" better than others on cheap hardware – used for intensive computation
  - Many universities install it on workstations, clusters, servers...
- Huge community based effort to keep it up to date and to keep it accessible for everybody

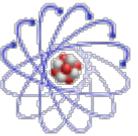
# What is a Linux Distribution

## What are Software repositories

- *Distribution*: A **selection of software** that runs on top of a GNU/Linux operating system. This includes:
  - Installation tools (to install the operating system in a computer)
  - Software installation programs (to add new software to a computer)
  - **Configuration** and **management** software
  - Graphical interfaces
  - Office programs alternatives
  - Development tools
  - Communication software
  - Scientific software
  - Videogames
  - .....
- Bound to a place (server) on the internet where such software is available, often called **software packages repository** or in short **repository**. A repository can be hosted on multiple servers around the world, called **mirrors**.
- Like *Google Play* or *Apple Store*, but existed already **more than 10 years earlier**.



# Popular Linux distributions

- **Debian**, community based   **ubuntu**
  - **Ubuntu, Kubuntu, Lubuntu** based on Debian, commercially maintained by Canonical
    - Derivatives: Linux Mint ...
- **Arch Linux**: community based 
- **RedHat**: first commercial open source Linux
  - Derivatives: Mandrake Linux, ...
  - **CentOS**, community based 
  - **Scientific Linux**, developed at CERN 
- **Novell SuSE**: commercial Linux
  - **OpenSuSE**: community based SuSE 
- Lots more: <http://distrowatch.com/>

# Lubuntu: the one we will use

- Best user community effort backed up by the Canonical company, although with lots of criticism by other communities
- Probably the easiest for you to try at home
- L stands for *lightweight*, that means that we doesn't require a computer with very powerful hardware.



# Linux installation: basic concepts



- The operating system usually comes in a **DVD** image, that one can put on a DVD or on a USB pen. The image contains also an **installer software**.

- The operating system is usually **installed on a hard disk**.



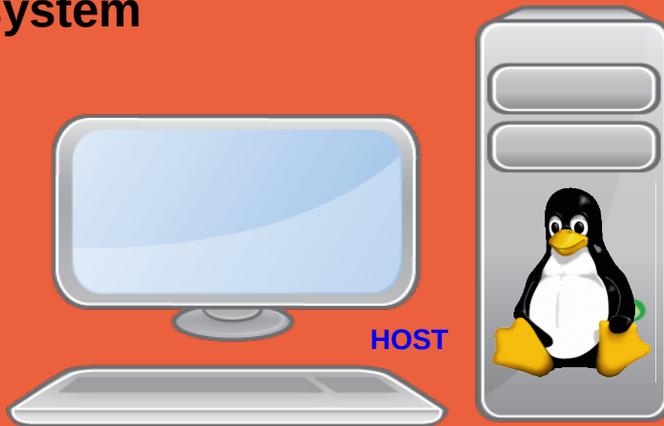
- Popular operating systems like Windows and OSX **do not like to share the hard disk** with other systems: this poses installation problems.
- The Linux community has found many solutions to this limitation, in order for Linux to be installed together with other systems.



# Linux installation: three strategies

Requires to be administrator

1. Linux as the *only* operating system



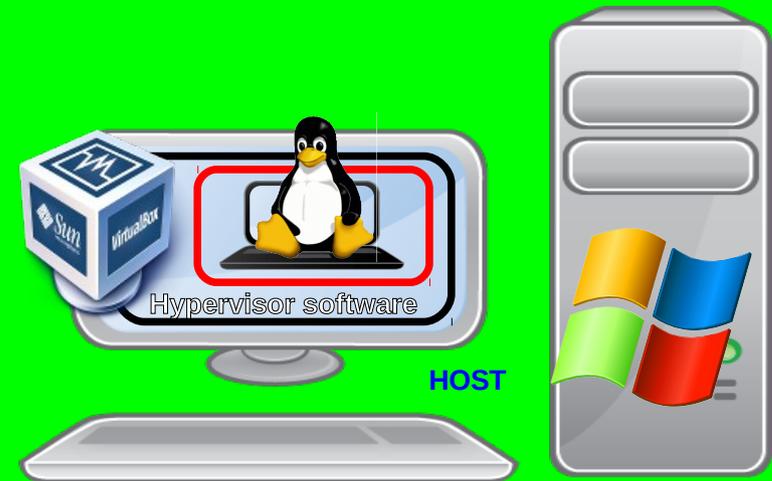
3. Dual boot with GRUB2: two systems installed, but only *one* runs at every restart



Admininstrator must install  
Virtualization software



2. Virtual Machine: operating systems running *at the same time*



# Linux installation: three strategies

**1 Single operating system:** Linux deletes everything on the disk and it becomes the one and only operating system for that computer

- Constraint: user must own the computer (administrator)

**2 Virtual Machine:** Run Linux in a virtual machine where it is the one and only system. The virtual machine is run on an hypervisor that is running in the existing OS.

- Constraint: User must be able to **install software**, i.e. VirtualBox hypervisor

**3 Dual boot:** Linux shares the disk with other operating systems. Requires an alternative boot loader (e.g. GRUB2), a small program that is loaded **BEFORE** any other installed system. This program must be installed at the beginning of the disk.

- Constraint: user must own the computer (administrator)

# Today's tutorial continued

- Download of Ubuntu Linux
- Installation of Ubuntu Linux
- Installing software from repositories
- Reboot
- Software update
- Customizing the work environment
- Repositories and other installation methods

# Download Ubuntu LTS\*



OBS: If the network is too slow in the lab:  
- ask the teacher for a USB stick containing the installation CD-ROM ISO image.  
- there is a copy of the ISO image in C:\Courses\fysik\MNXB01\

a) Go to

<http://ubuntu.me/downloads>

b) Look for "**Version 18.04.3 LTS**"

c) Right-click on the link "**Desktop 32bit**"

Desktop 32-bit



d) choose "save link as..."

e) Select a folder where there is enough space, like

C:\temp\MNXB01\

f) Remember the above folder! We will use it in the next step.

Direct link:

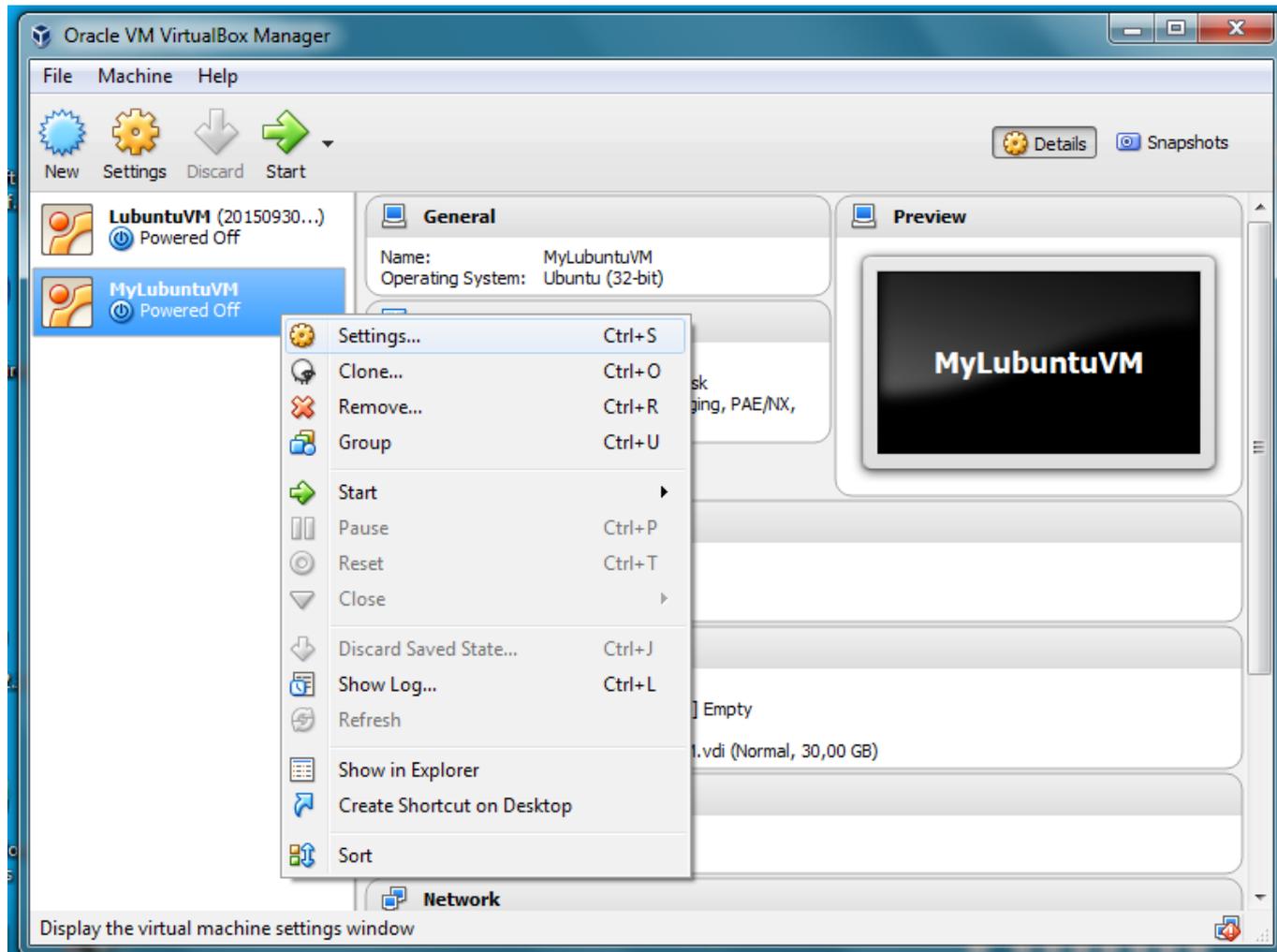
<http://cdimage.ubuntu.com/lubuntu/releases/18.04/release/lubuntu-18.04.3-desktop-i386.iso>

\* LTS stands for Long Term Support.

Means the system is ensured to be stable (no crash/malfunction) while using it, and software updates are provided for some Long Time.

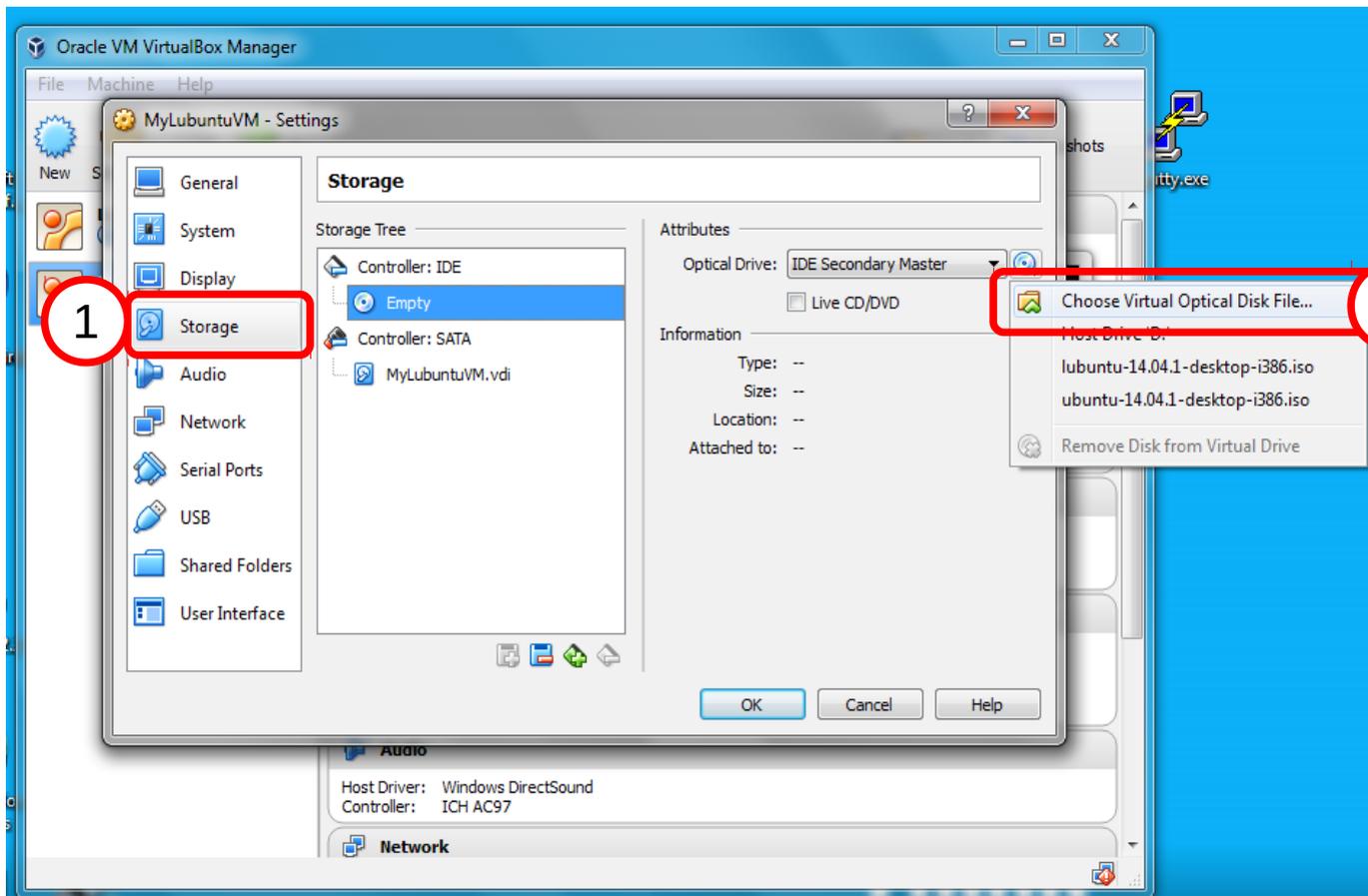
# Ex. B2: Install Lubuntu in MyLubuntuVM

Select the virtual machine and open settings.



# Ex. B2: Install Ubuntu in MyLubuntuVM

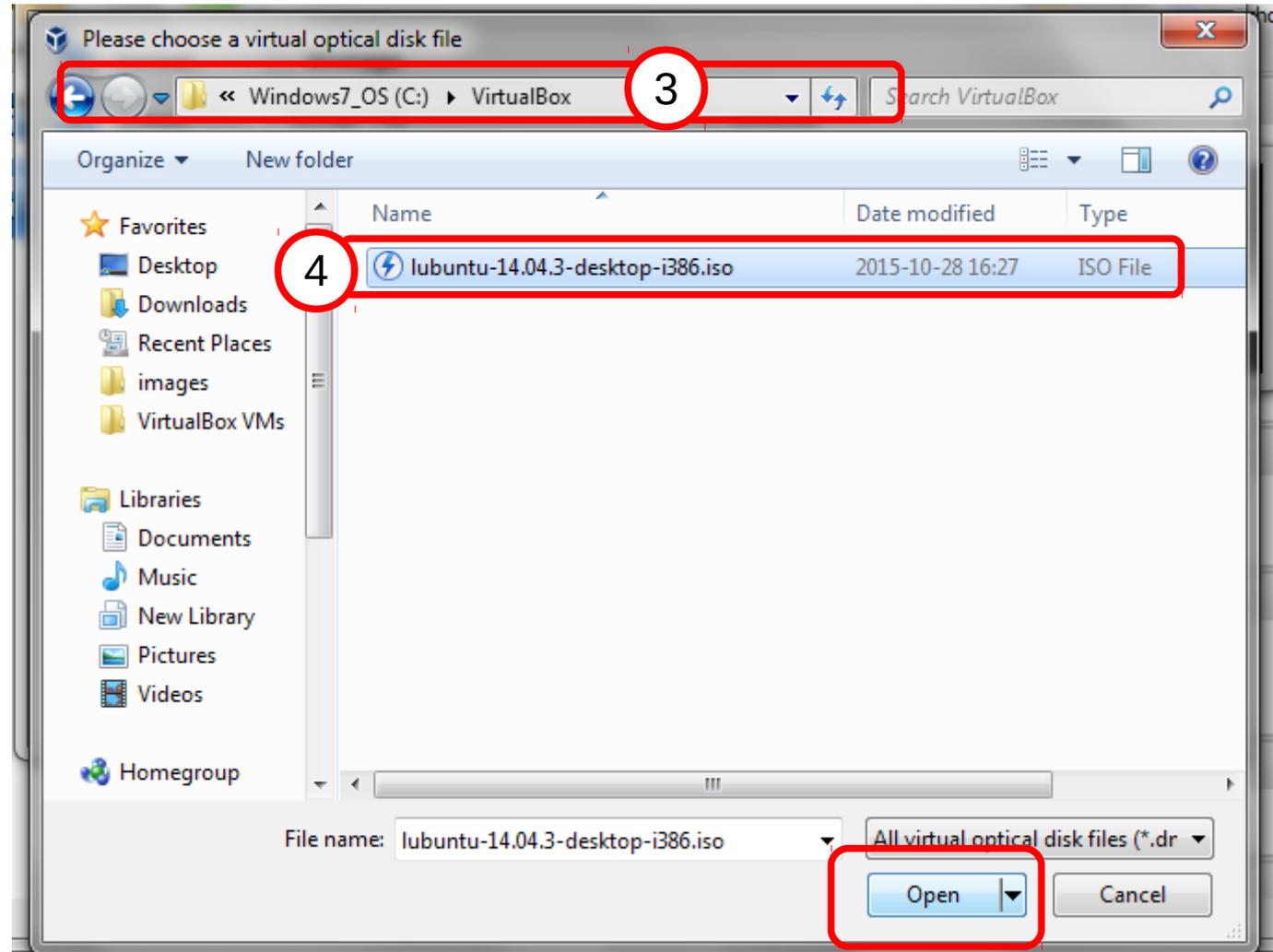
## B2.1 Add the ISO file just downloaded to MyLubuntuVM



# Ex. B2: Install Ubuntu in MyUbuntuVM

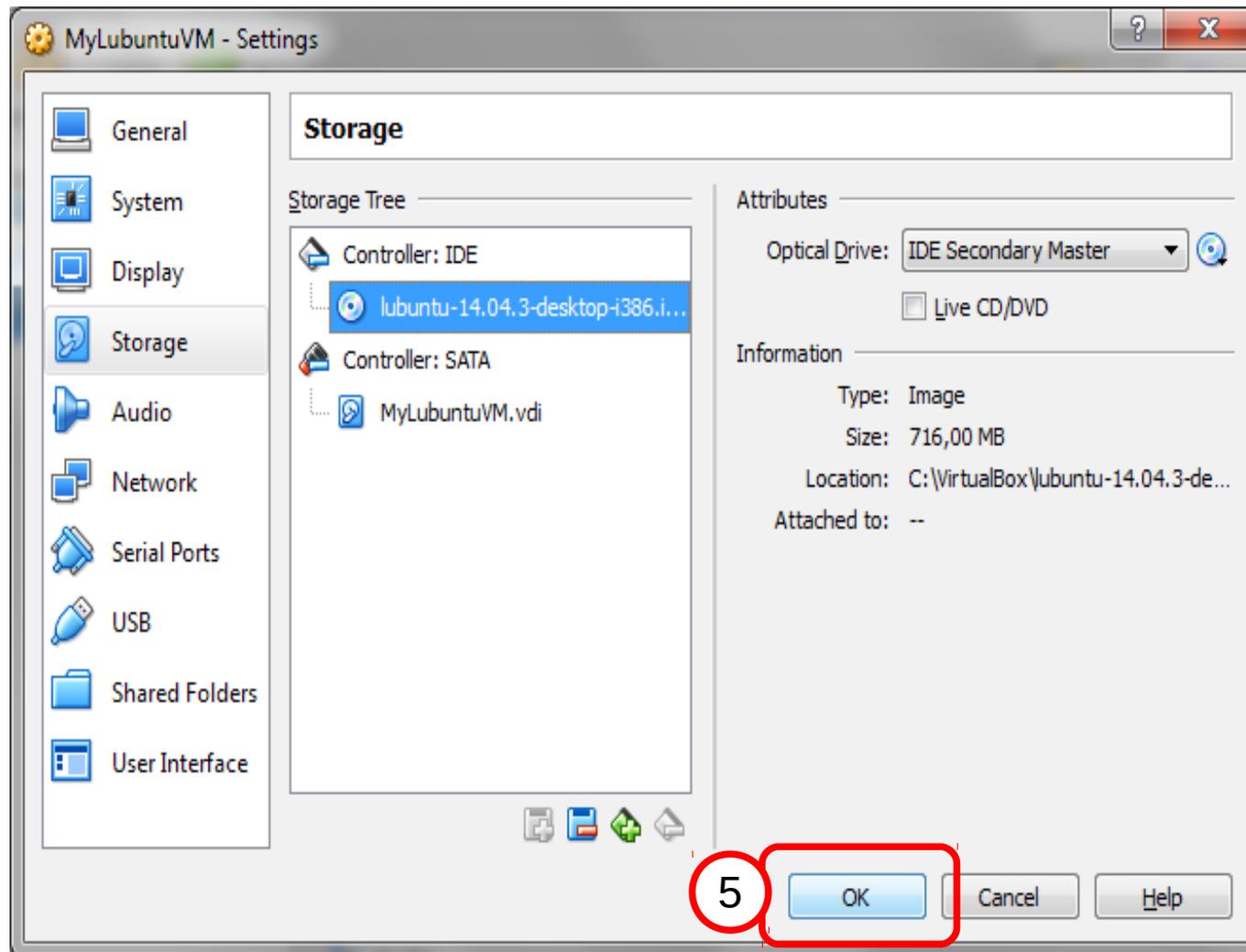
B2.2 Go to the folder where you saved the file (3) at slide 47 and select the ISO file

downloaded (4) (folder name and filename might be different from the picture)



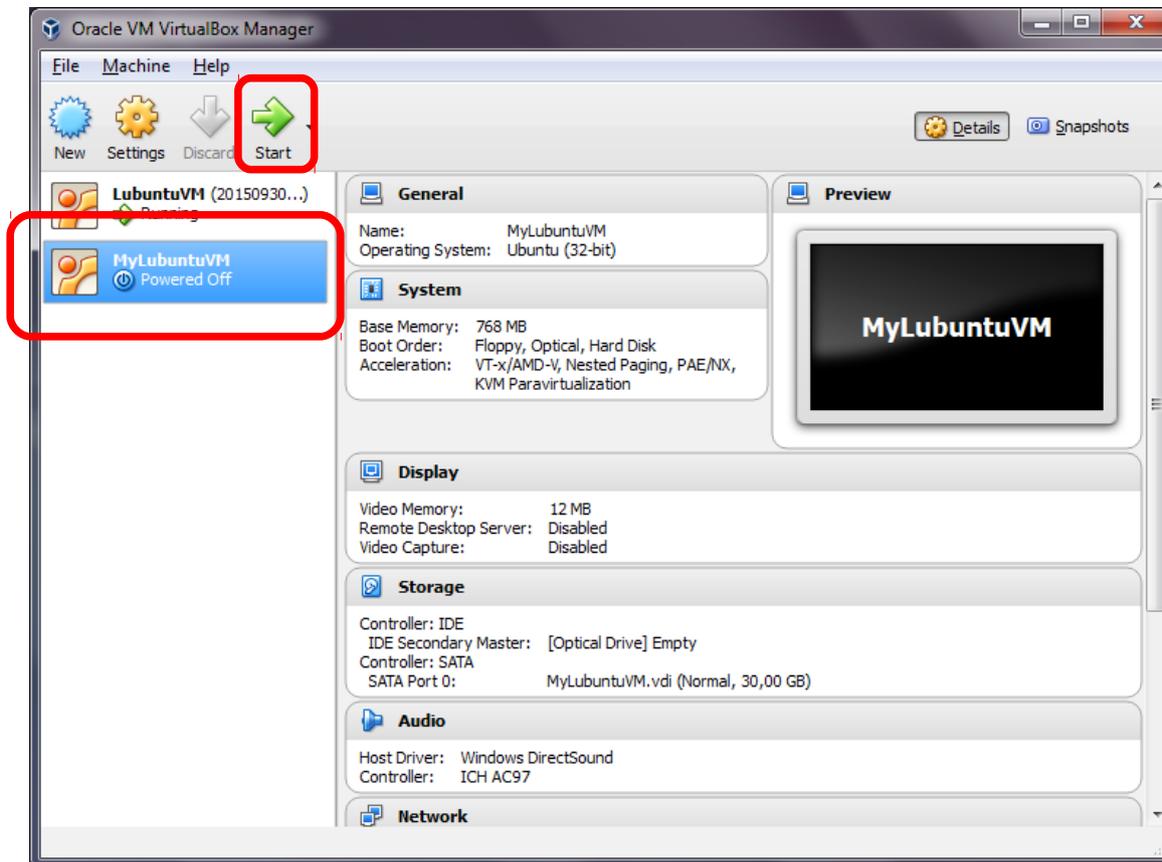
# Ex. B2: Install Ubuntu in MyLubuntuVM

## B2.3 Accept the configuration changes (5)



# Ex. B2: Install Lubuntu in MyLubuntuVM - start it!

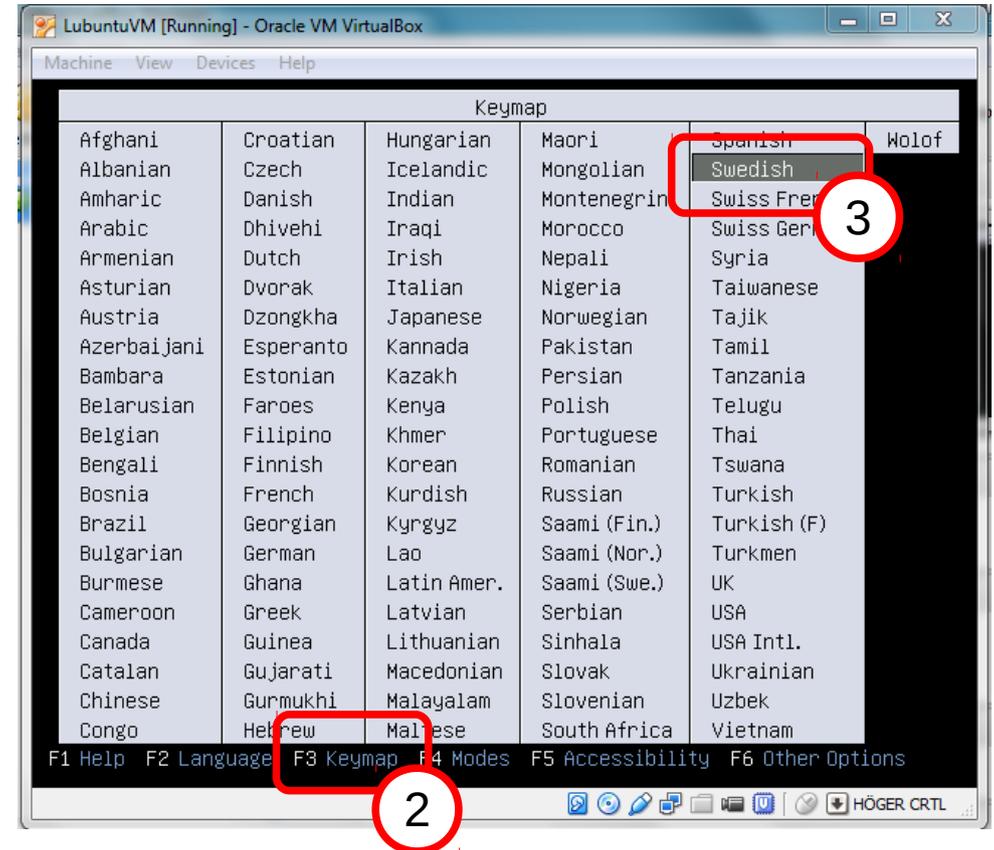
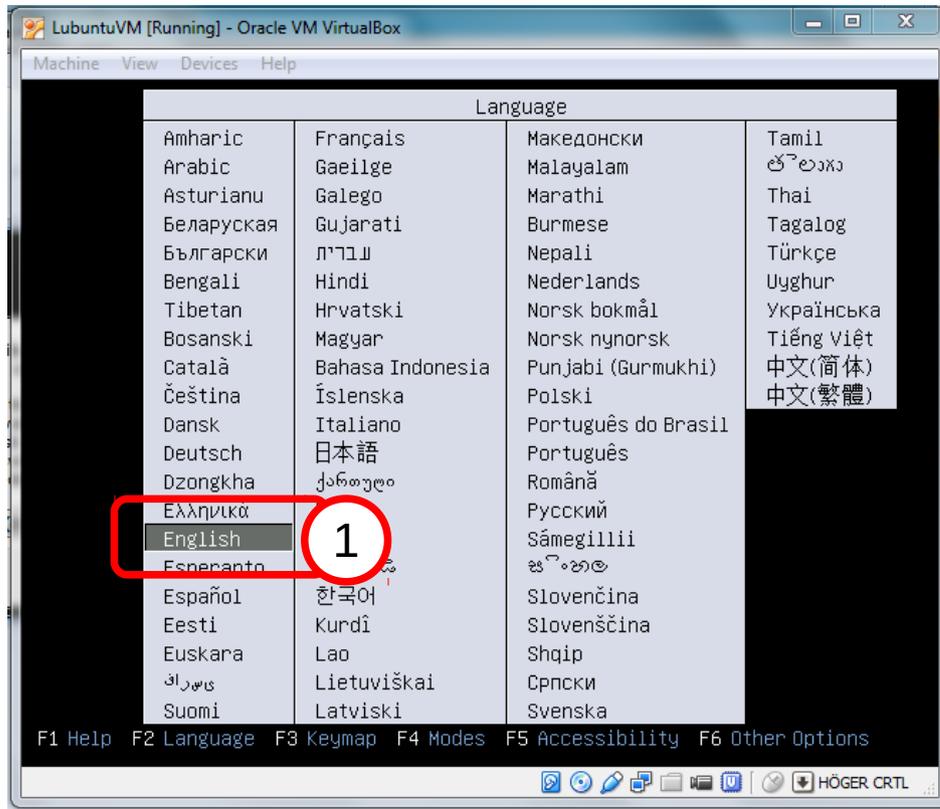
## B2.3 Select and Start the virtual machine.



# Ex. B2: Install Lubuntu in MyLubuntuVM

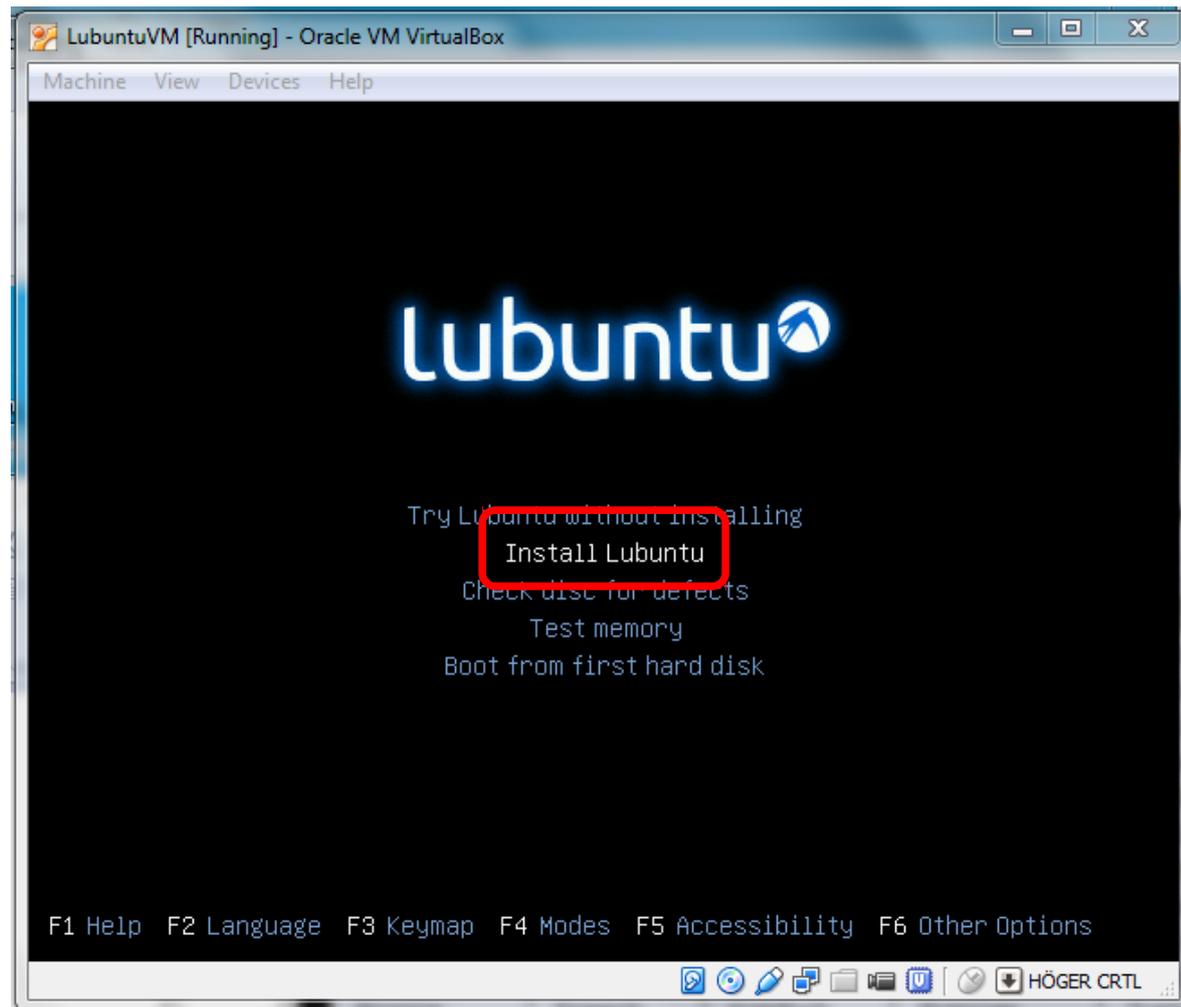
B2.4 Select English as installation language, press F3 and change keyboard to Swedish\*.

\* if you're using your own laptop and has a keyboard in some other language, choose that language!



# Ex. B2: Install Lubuntu in MyLubuntuVM

B2.5 Select *Install Lubuntu* and press enter. Wait until the installer starts.



# Problems? Trashed screen

If you see a screen like this, is because the virtual machine video drivers are not optimized for the VirtualBox virtual video card.

Solution: in Linux you can switch from a video terminal to a text terminal by pressing

**Ctrl + Alt + F2**

To switch back to the Video terminal,  
Press

**Ctrl + Alt + F1**

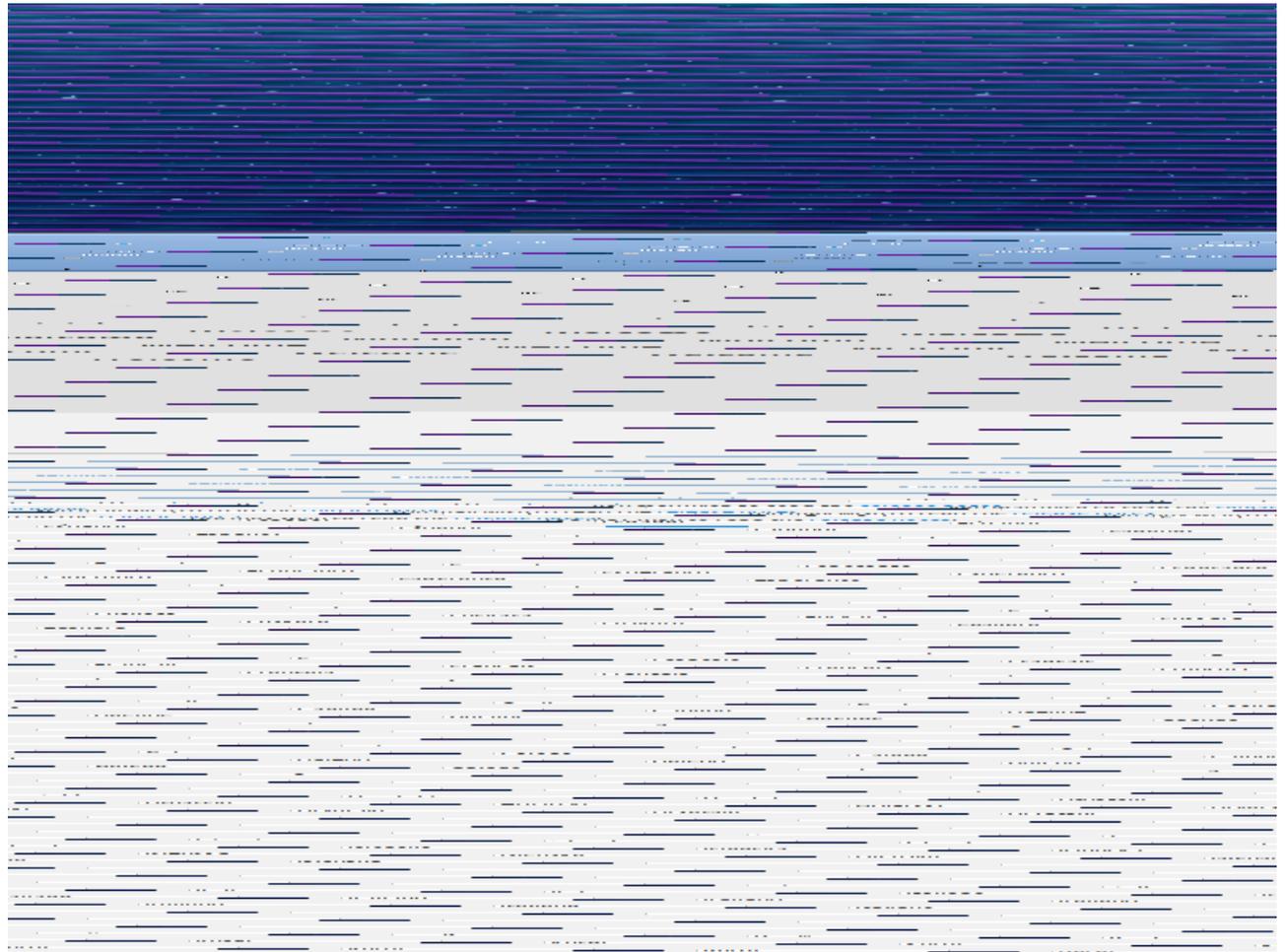
Unfortunately the combination "Ctrl Alt" is intercepted by the host system (Windows) so we cannot use it.

Luckily VirtualBox allows us to use the Host Key **Right Ctrl**:  
Switch to text console:

**Right Ctrl + F2**

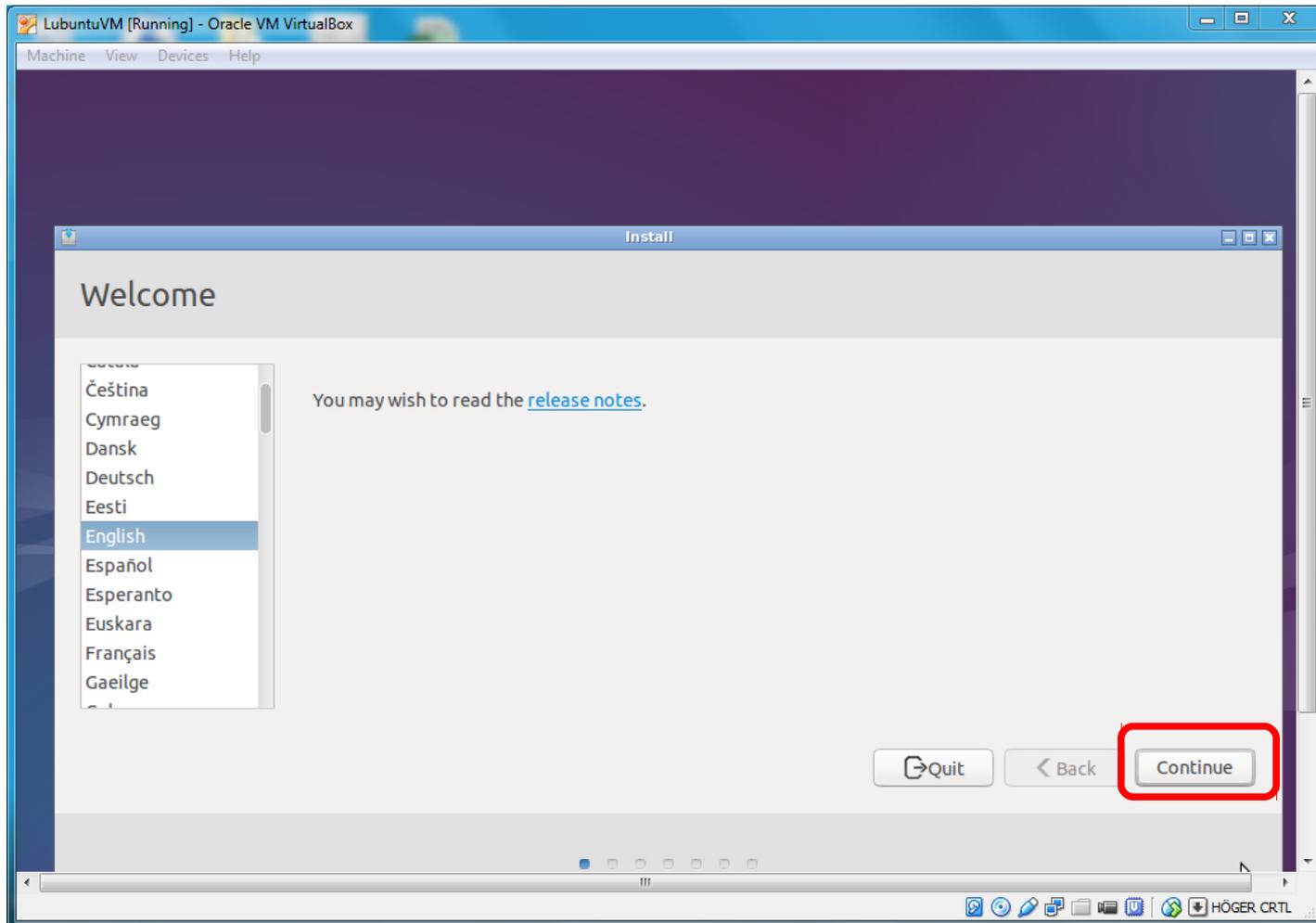
Switch to graphical:

**Right Ctrl + F1**



# Ex. B2: Install Ubuntu in MyLubuntuVM

## B2.6 Click on continue

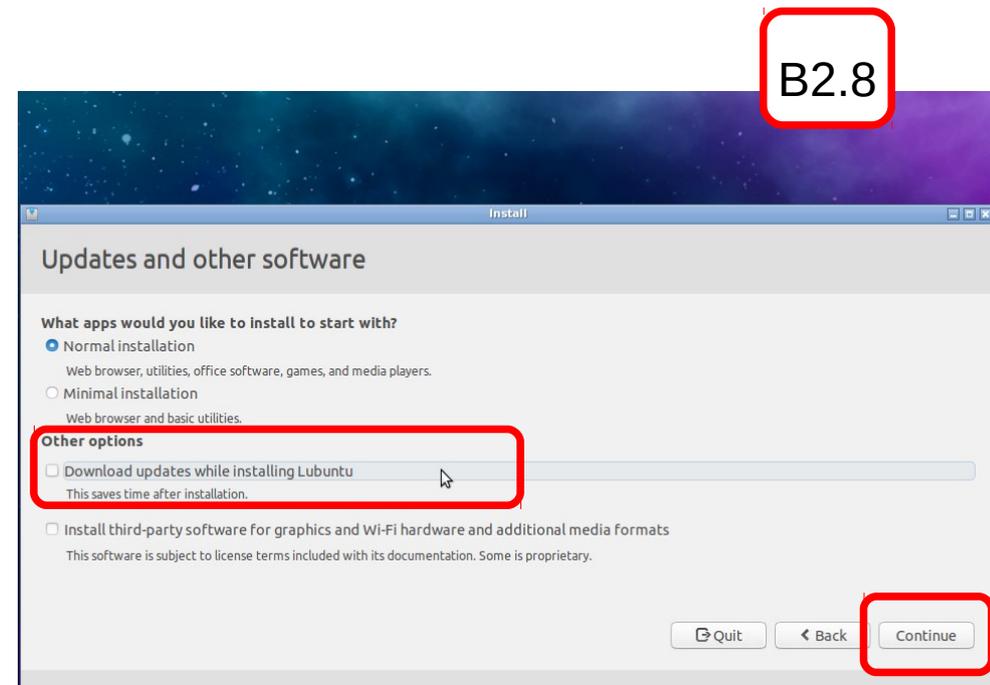
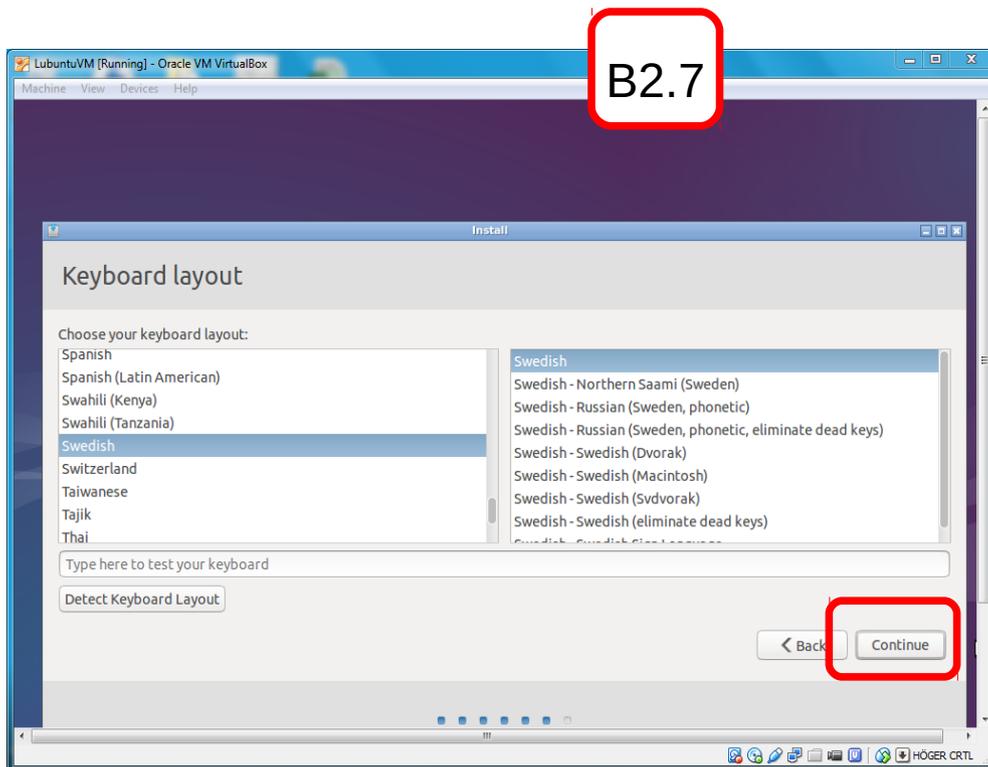


# Ex. B2: Install Ubuntu in MyUbuntuVM

## B2.7 Select the Swedish Keyboard\*

\* or you own keyboard

B2.8 Select "Normal installation" and **deselect** download updates while installing. It will take too much in the lab. You can do it at home when you have plenty of time. It can also be done after the installation.

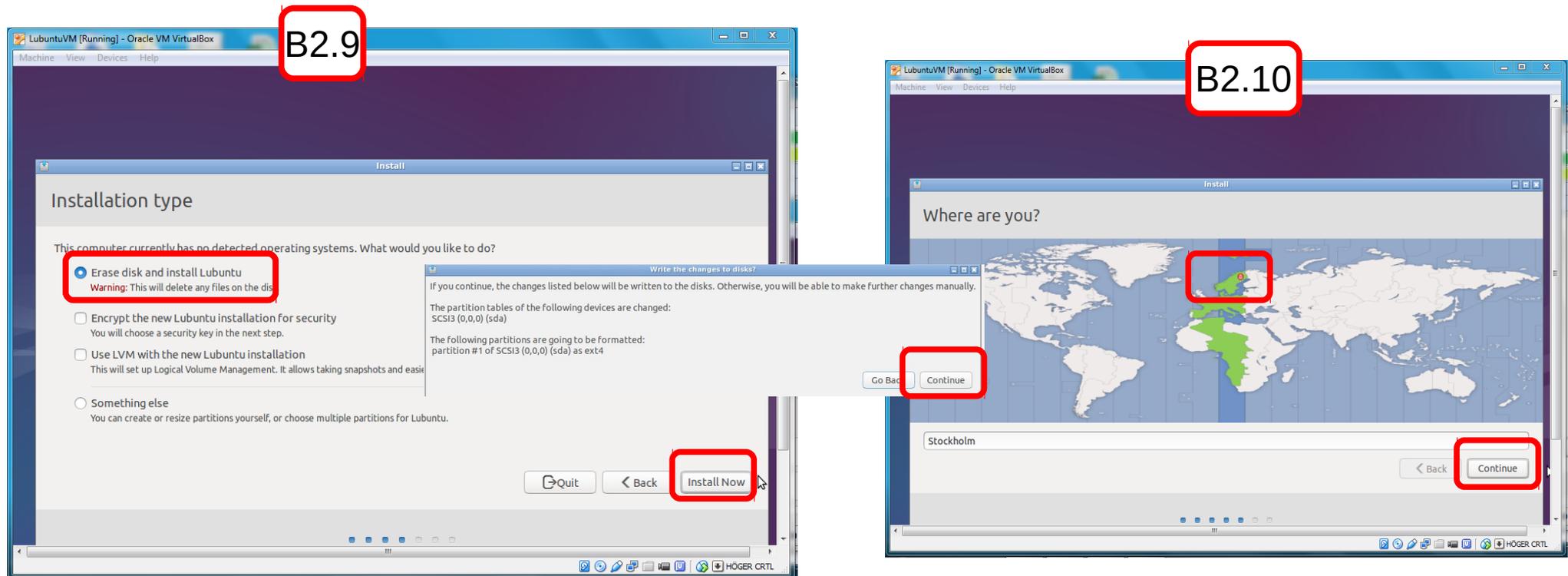


# Ex. B2: Install Ubuntu in MyLubuntuVM

B2.9 Choose installation type "Erase disk and install ubuntu" and click "Install Now".

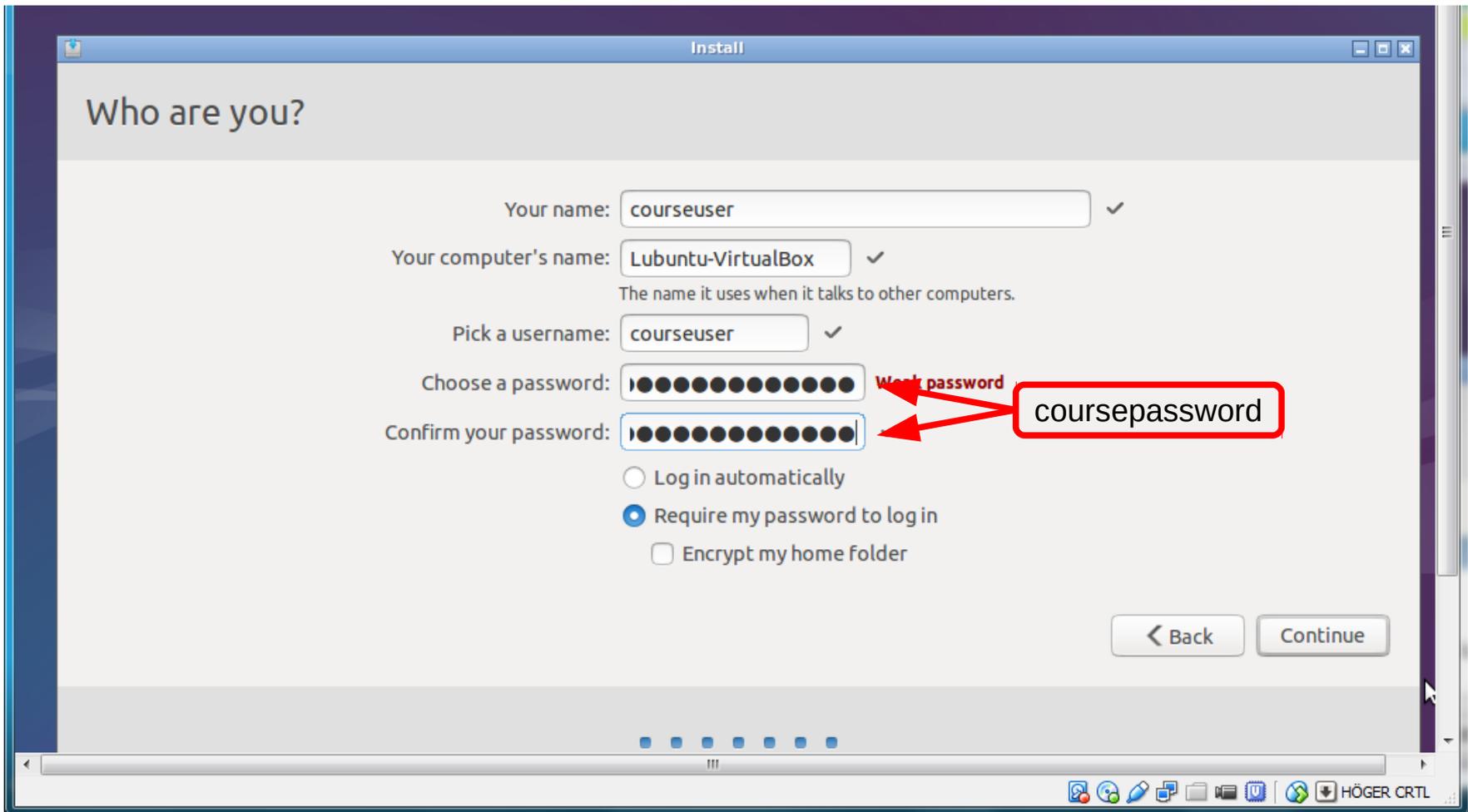
Click "continue" on the "Write changes to disk?" dialog.

B2.10 Select the Stockholm timezone



# Ex. B2: Install Lubuntu in MyLubuntuVM

B2.10 Insert computer name and username/  
password as in the picture:

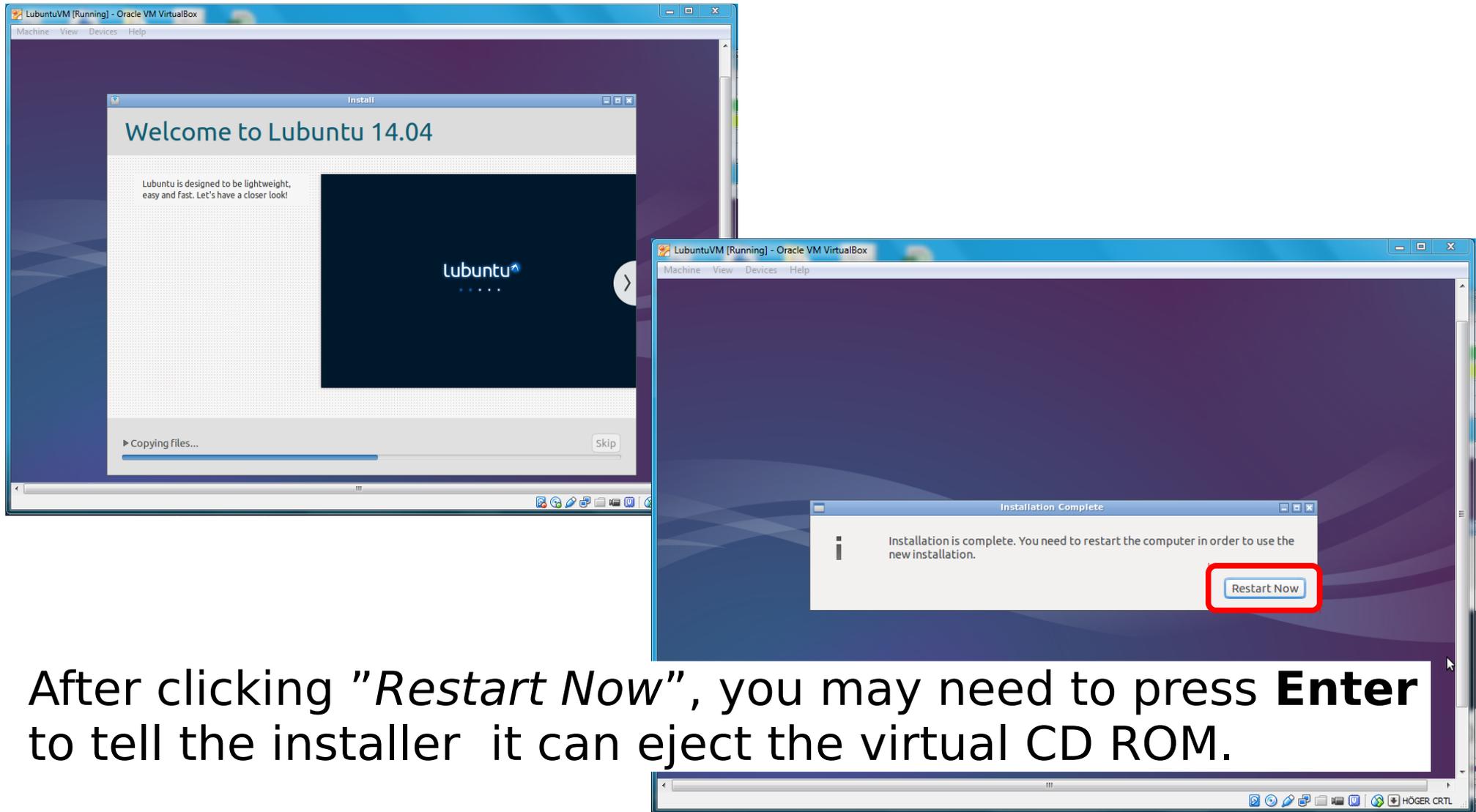


# Disclaimer

- **NEVER use such a simple password such as *coursepassword* in a machine you install.**
- Read this recommendation for good passwords:  
<http://www.hep.lu.se/staff/paganelli/doku.php/variousnotes#passwords>
- Read this about how easy is to crack(discover) weak linux passwords:  
<https://null-byte.wonderhowto.com/how-to/hack-like-pro-crack-user-passwords-linux-system-0147164/>
  - Fun: you can try the above at home and watch it cracking courseuser's password.

# Ex. B2: Install Ubuntu in MyUbuntuVM

## B2.11 Let's wait for installation to complete!



After clicking "*Restart Now*", you may need to press **Enter** to tell the installer it can eject the virtual CD ROM.

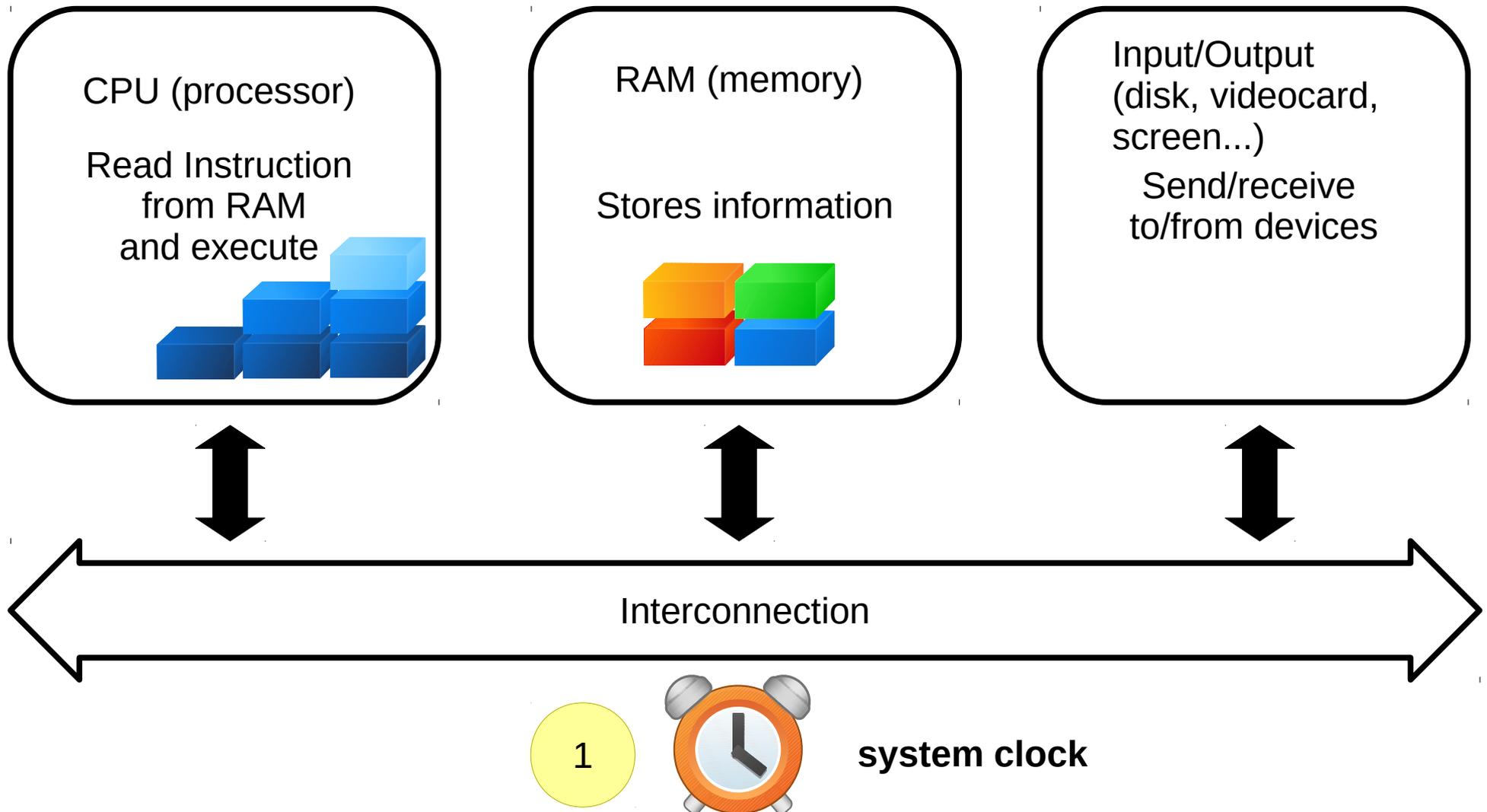
# Ex. B2 summary: Install Lubuntu in MyLubuntuVM

- Start the virtual machine with the Start button with the arrow.
- At the lubuntu prompt:
  - Choose "English"
  - Press F3. Select the Swedish keyboard.
  - Select "Install lubuntu"
- Follow the on-screen instructions. When prompted, insert the informations/choices in the following table:

Installation dialog	Click continue button, select keyboard, deselect download packages, choose timezone
Installation type dialog	Select "Erase disk and install ubuntu" and click continue
User name	courseuser
Computer name	Lubuntu-VirtualBox
User name	courseuser
password	coursepassword (insert twice)

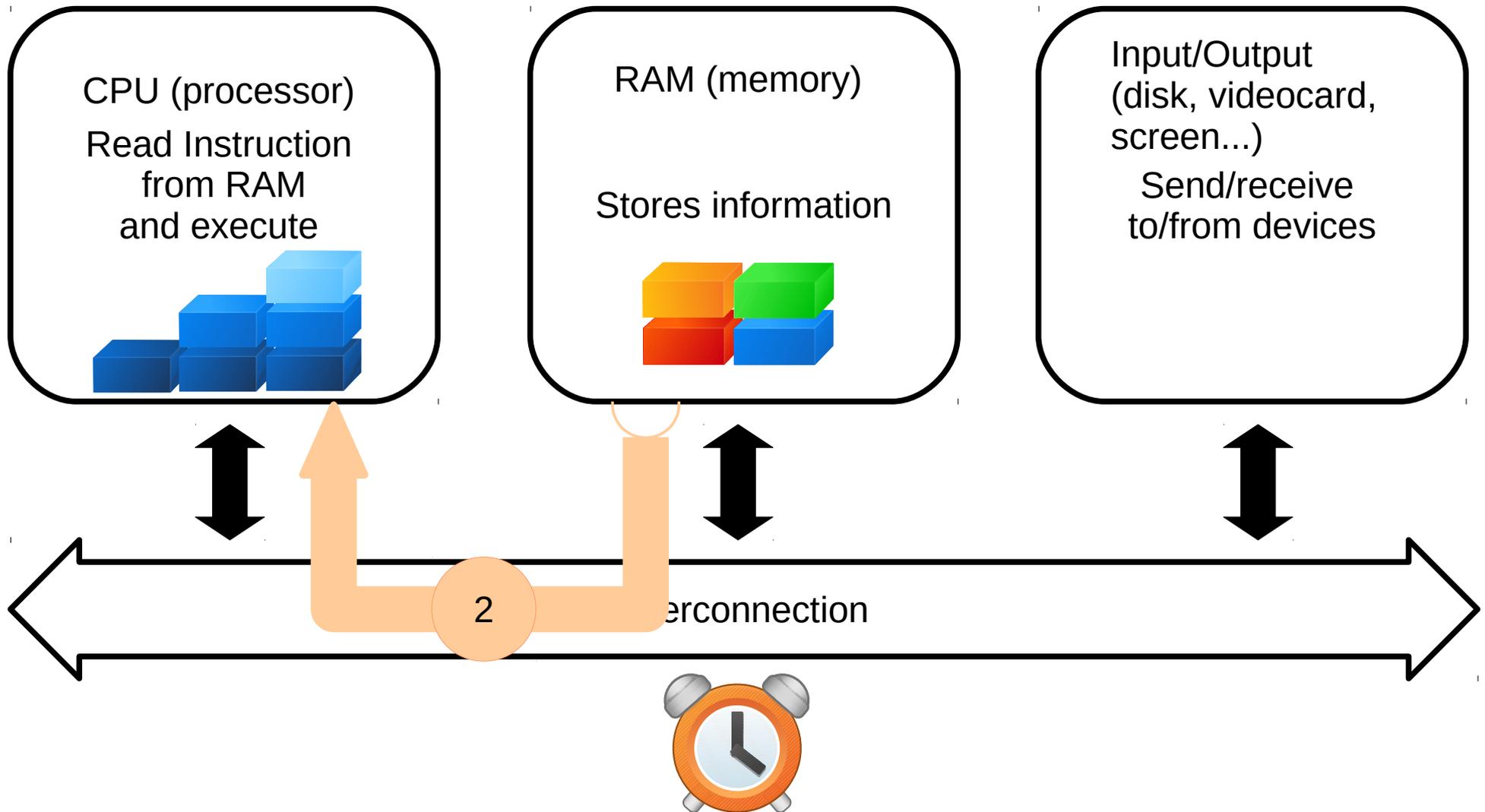
# How does it work?

## The computing cycle



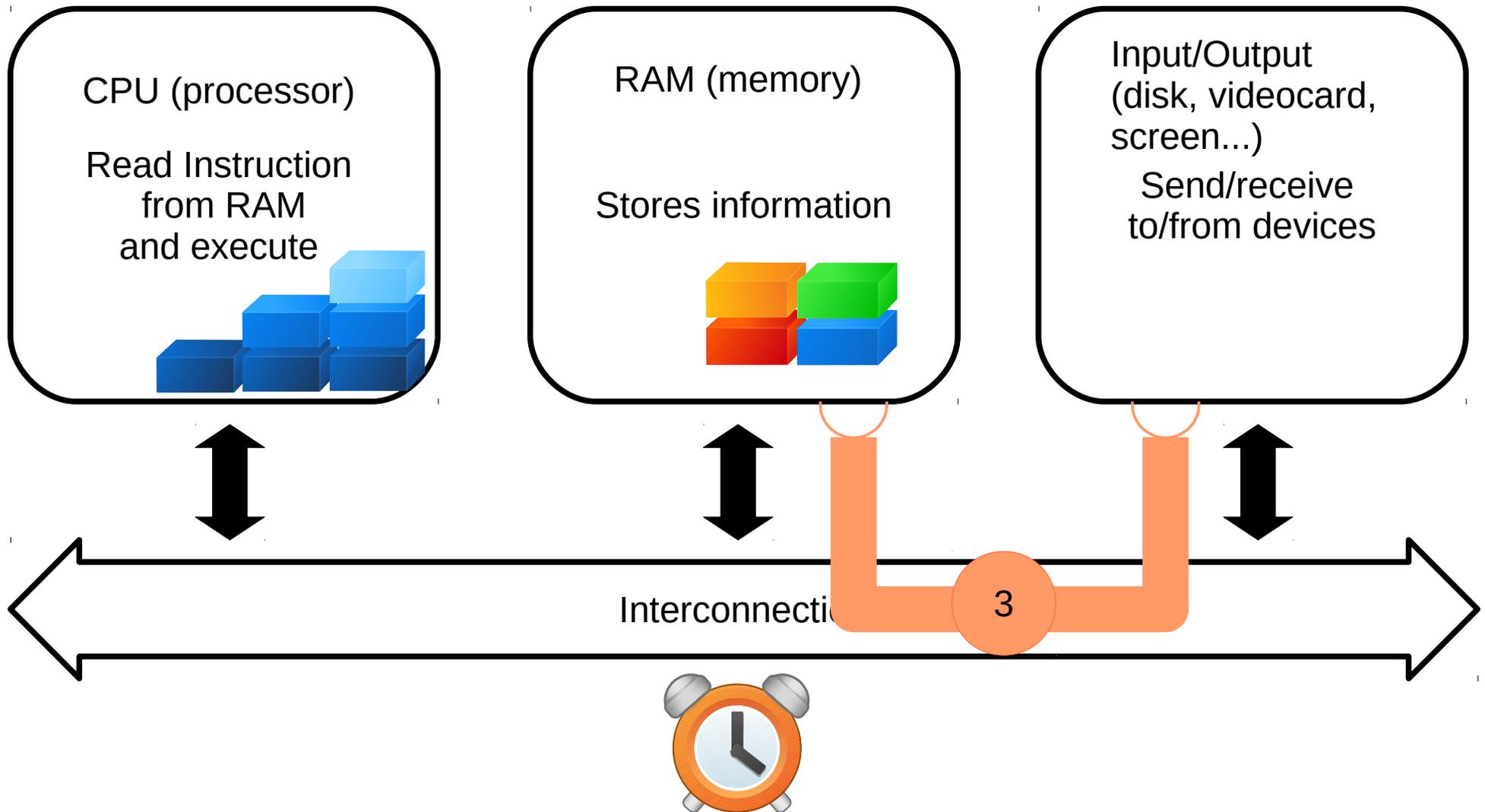
# How does it work?

## The computing cycle



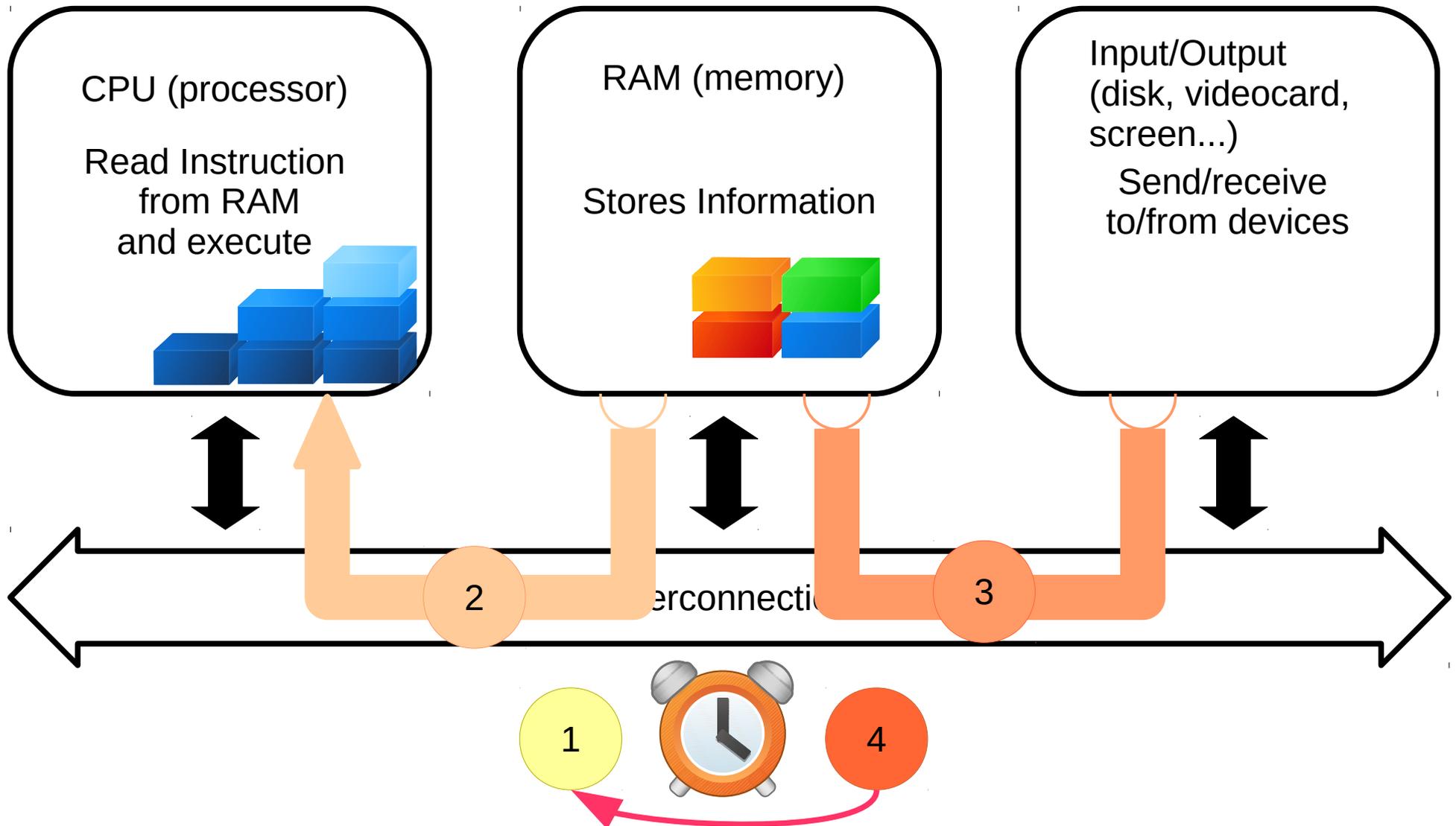
# How does it work?

## The computing cycle



# How does it work?

## The computing cycle



# How does it work?

## The computing cycle (summary)

The execution cycle and the clock

- 1) clock ticks
- 2) CPU reads content of RAM(instructions) into registries and executes
- 3) Execution might dispatch information over the bus
- 4) Wait for next clock cycle

The execution is **always serial**, but gives us a feeling of parallel tasks because of speed.

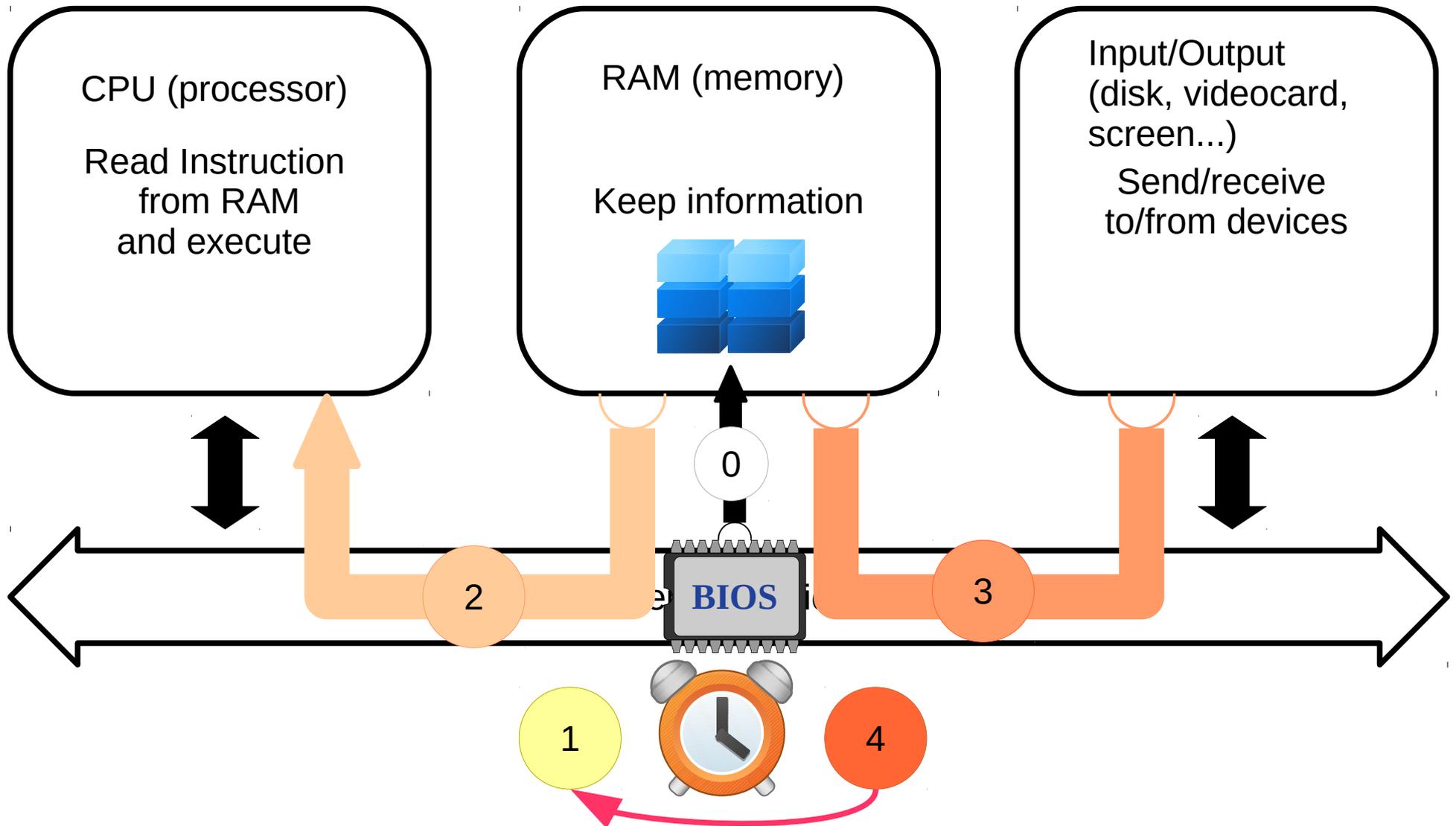
It might require more than one clock cycle to execute an instruction.

# But... How does it start?

- When a computer is turned on, the first thing it does is to **boot**
- **Boot**, or the bootstrap sequence, is a set of operations done in order to start the the computing cycle as described before.
- A small program is copied into the RAM as soon as the computer starts, and this is executed by the machine.
- This program is usually stored in a long term memory chip and is called **BIOS**

# But... how does it start?!?

**BIOS:** Basic Input/Output System  
to bootstrap the computer



# But... how does it start?!?

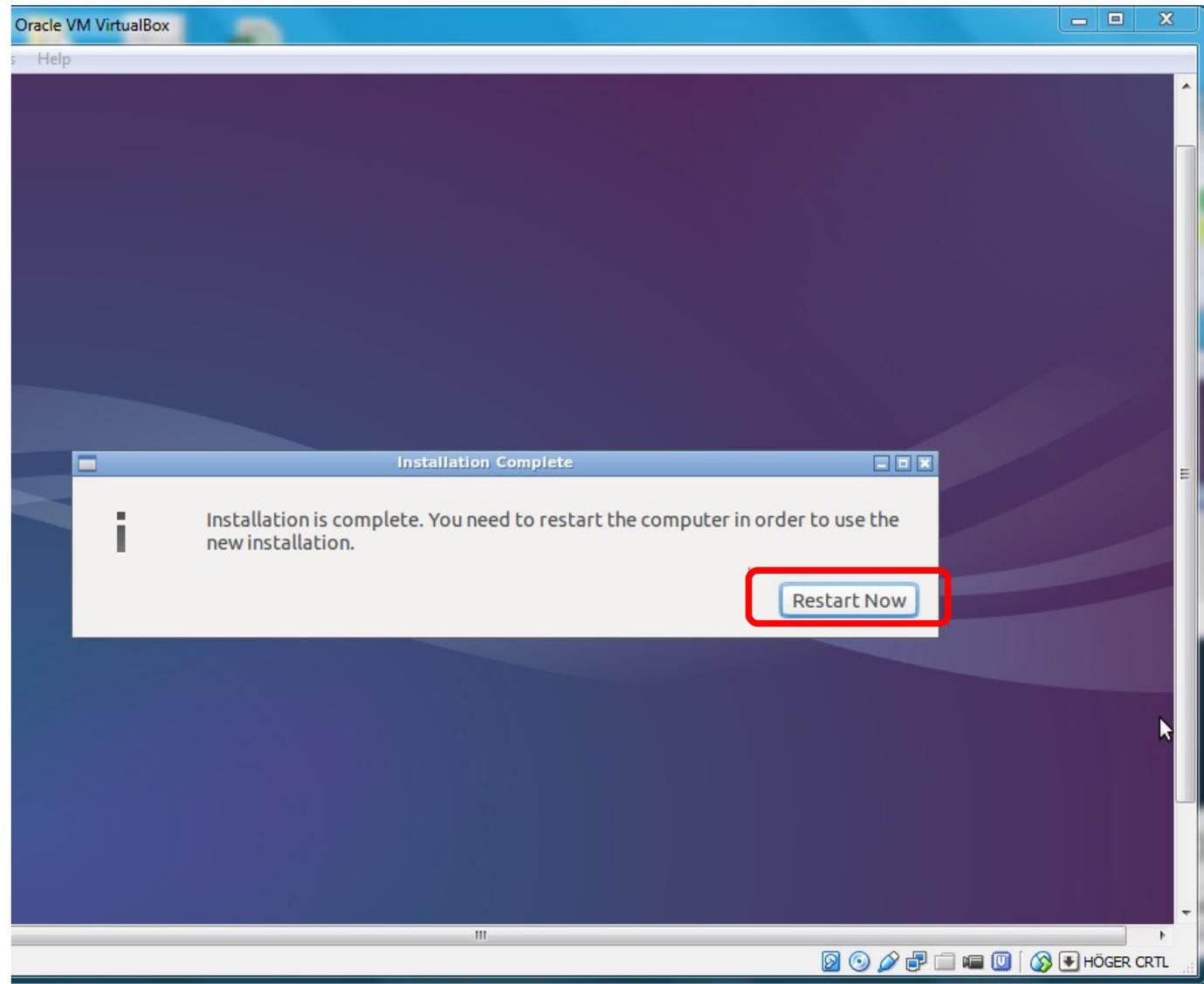
**BIOS:** Basic Input/Output System  
to bootstrap the computer

0. The BIOS loads a small program (a set of instructions and the data needed) into the RAM.  
When the clock starts, the CPU will start executing as explained.

# Ex. B2: Install Ubuntu in MyUbuntuVM

## B2.12 If installation completed, Restart Now!

If the system looks stuck, press *Enter*: the installer is asking if you want to eject the (virtual) CD-ROM, but sometimes it is not possible to see such message! The system cannot detect that this is a virtual machine.



# What happens after "Restart Now"?

- You may see a black screen with text.
- The Ubuntu installer is waiting for you to eject the installation cdrom (!).
- Just press **Enter** and the reboot should continue. The hypervisor will emulate a cdrom eject.

# Typical issues

- Screen is messed up.  
Solution: Enable 3D acceleration in the VM settings.

# Ex. B3: Installing the virtual machine graphical environment

- Did you notice how small is the screen? This is because **video drivers** for the **virtual video card** are missing.
- Lubuntu is aware it is running in a virtual machine, but needs to know **how to access the virtual hardware**.
- We will install the so-called **guest additions**
- There is a problem: these are not compatible with Lubuntu graphical stack. We will have to uninstall the existing graphical stack.
- We will use the command line.

## Ex. B3: Installing the virtual machine graphical environment - prerequisites

- Open a an LXTerminal

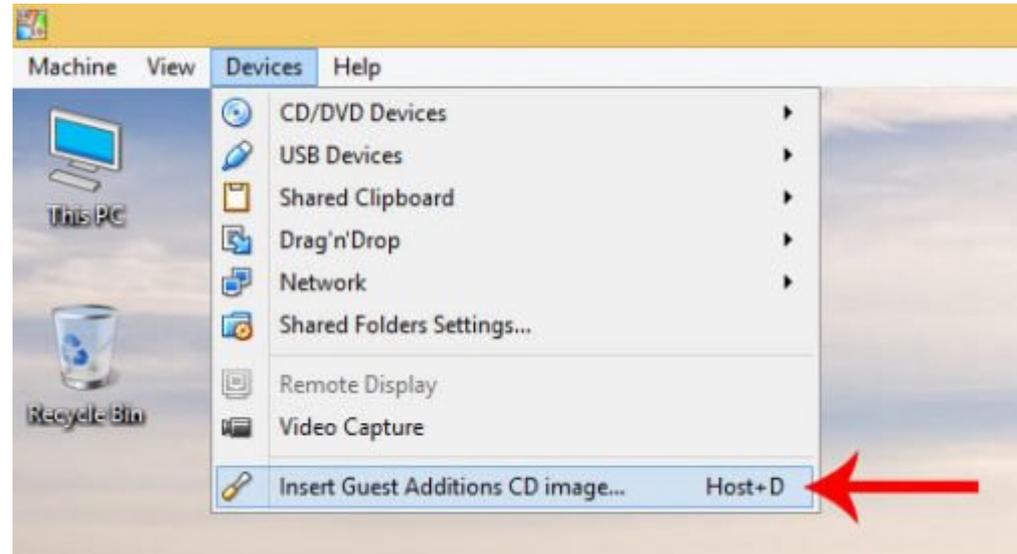
- Write the command:

```
sudo apt-get install build-essential  
and press Enter.
```

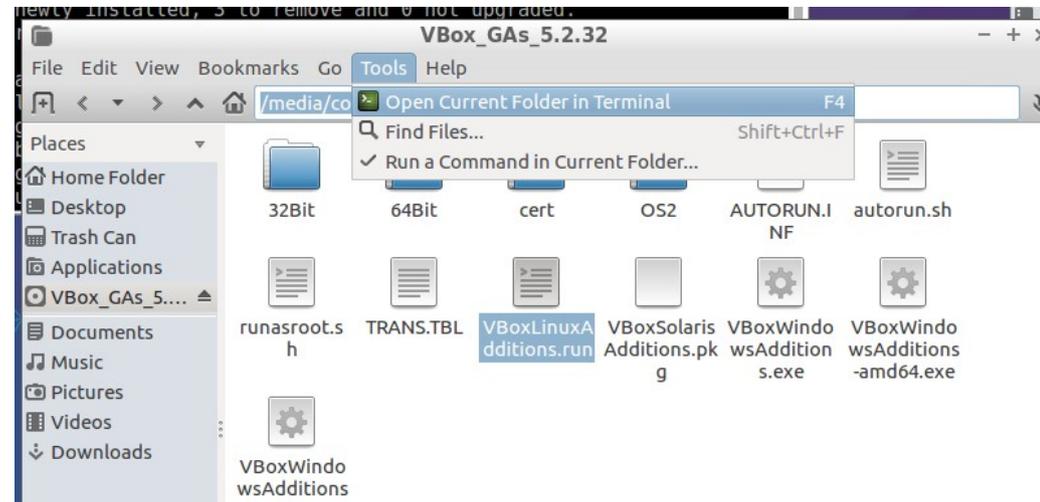
- You may be asked to input the password.
- Type Y end Enter at the prompt.

# Ex. B3: Installing the virtual machine virtualbox "guest additions"

- Find and click the "Install Virtual Box Guest additions" in the machine Devices menu
  - This will download and insert a virtual CD rom in the virtual cdrom drive that contains special drivers.



- A file browser window should appear on the desktop.
- Open the Tools → "Open current folder in terminal" menu and click on it



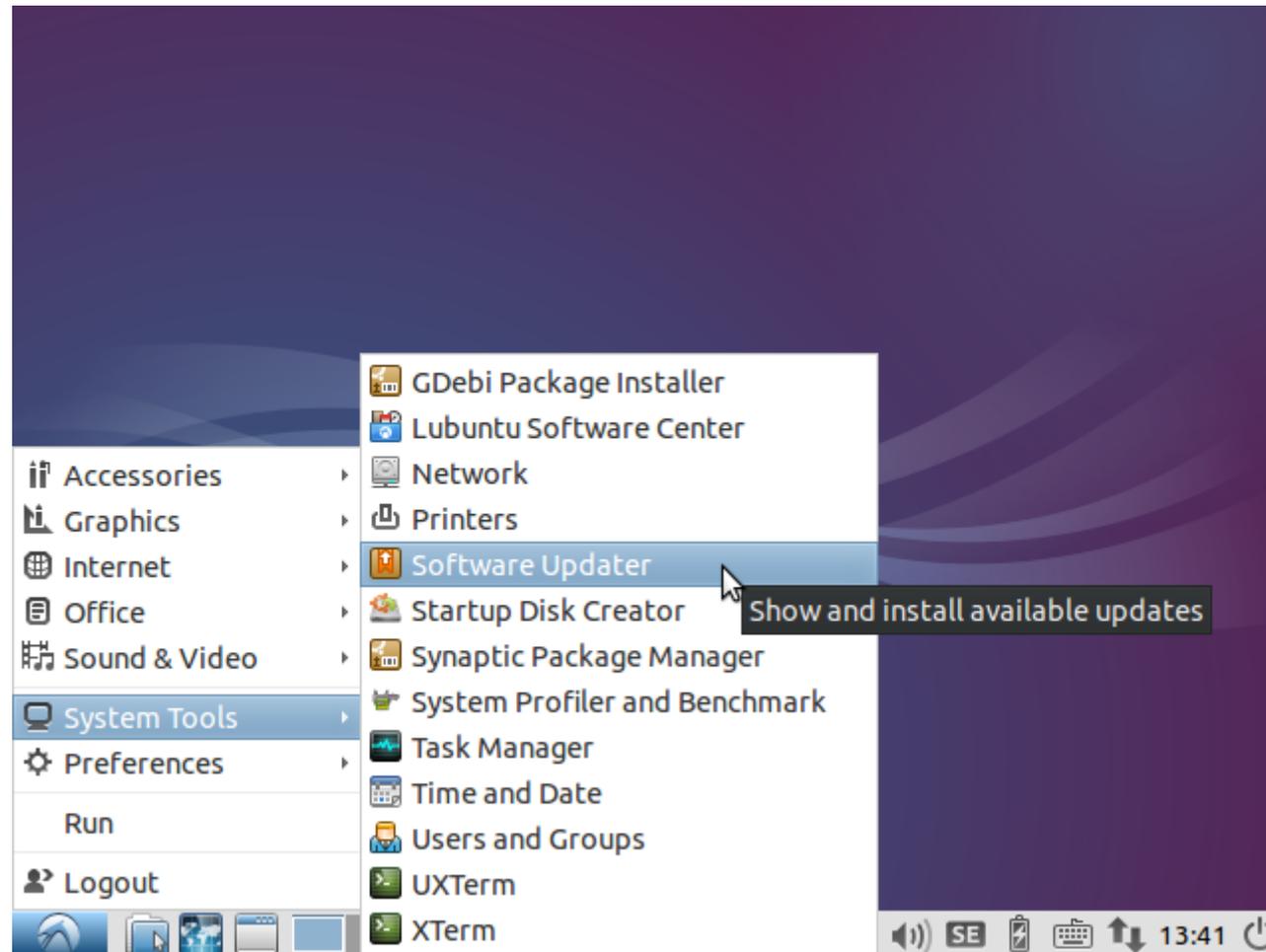
# Ex. B3: Installing the virtual machine virtualbox "guest additions"

- Write the following command on the terminal that appears:

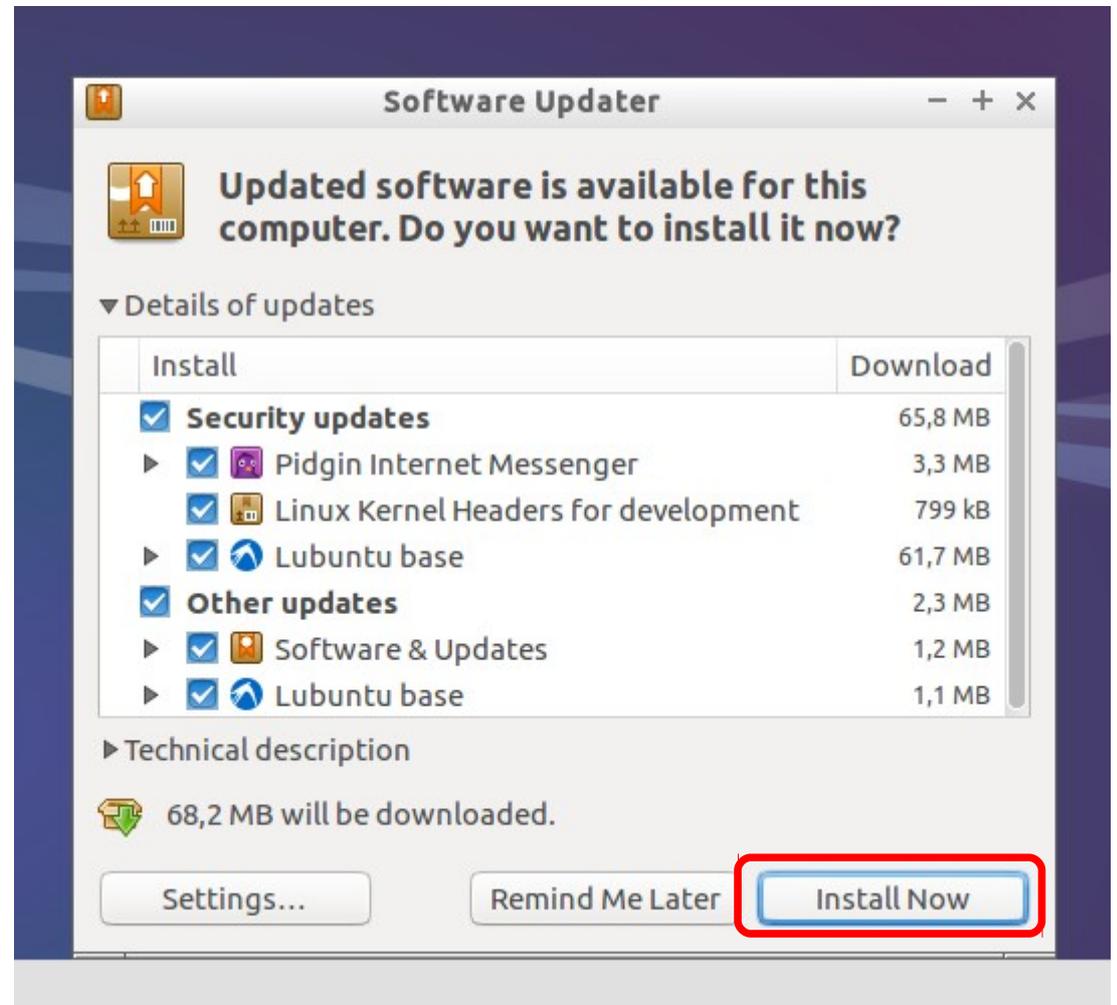
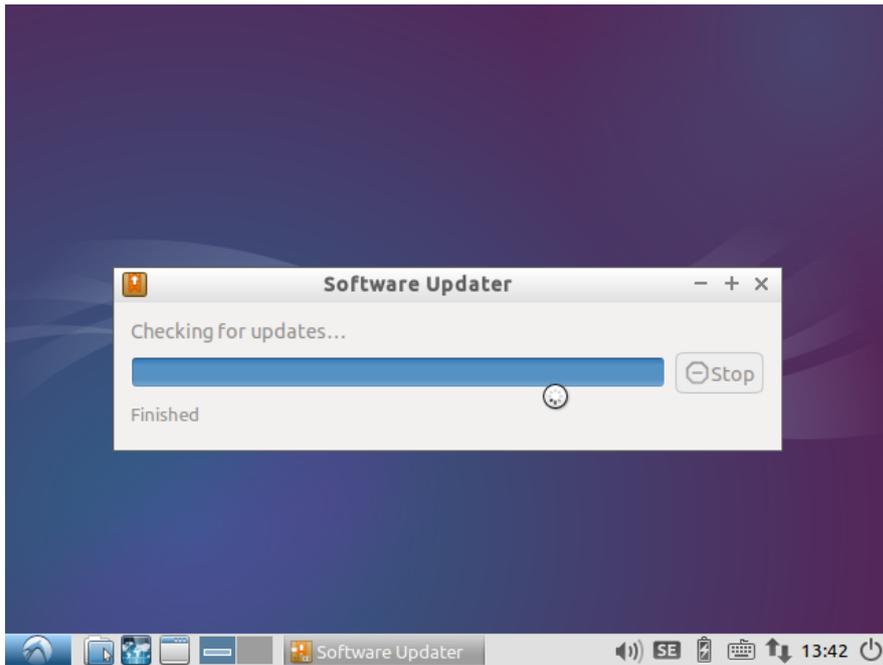
```
sudo ./VBoxLinuxAdditions.run
```

- And press Enter
- Answer yes to the various questions
- Once done, the screen should already resize.

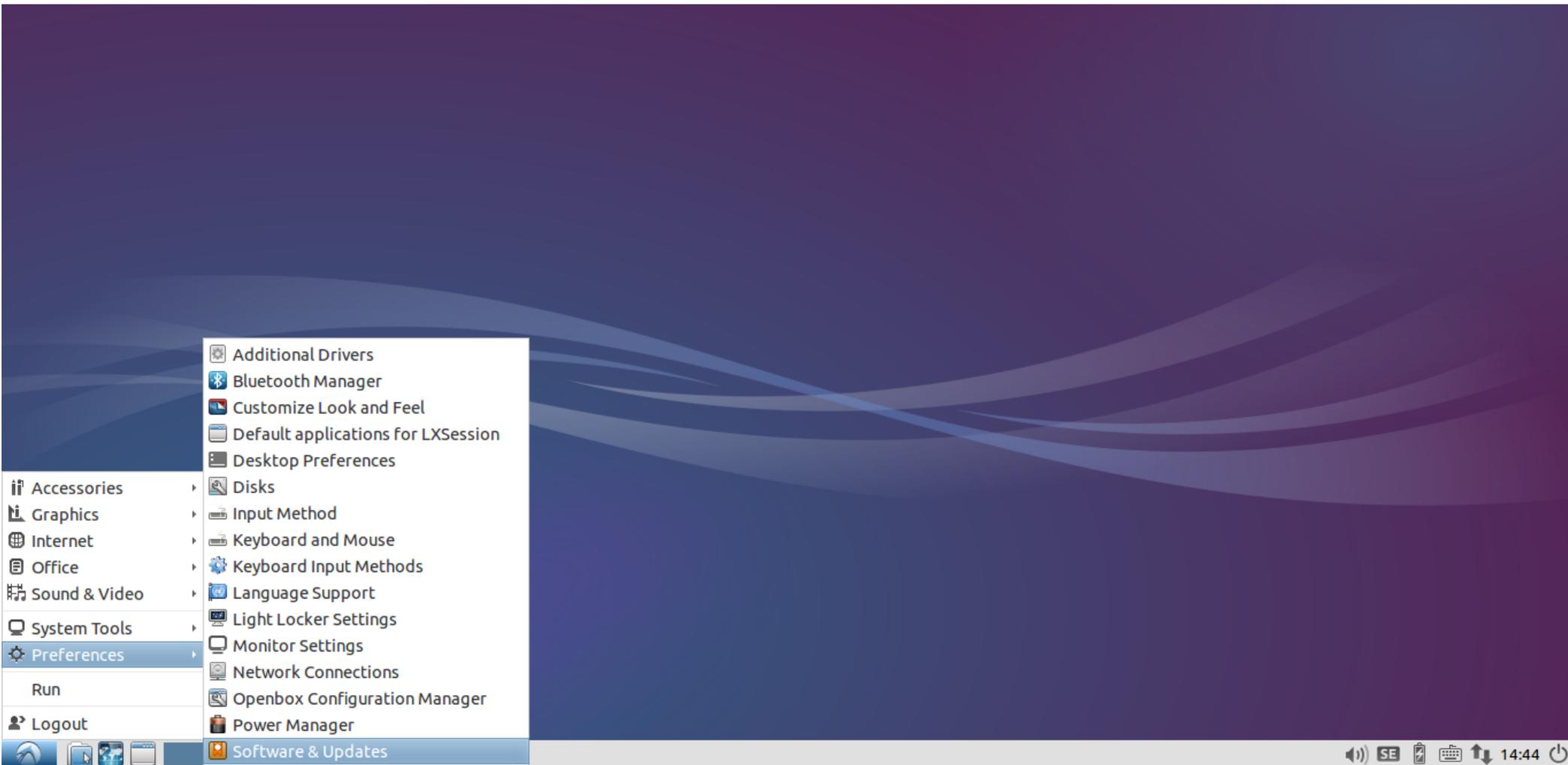
# Ex. B4: Software update



# Ex B4: Software update

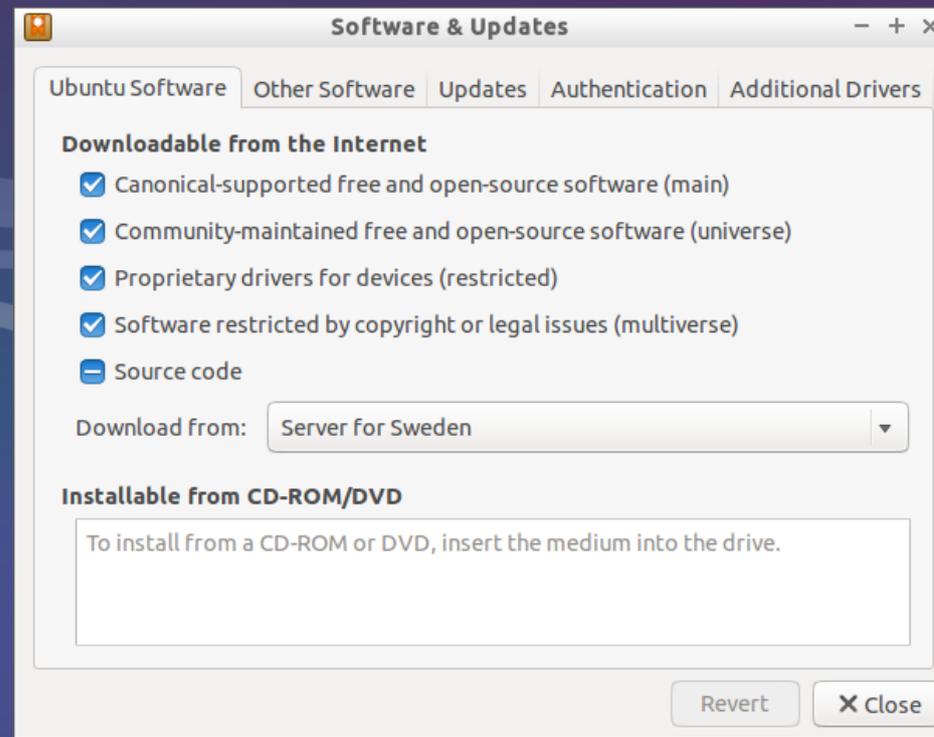


# Ex. B5: Software Repositories



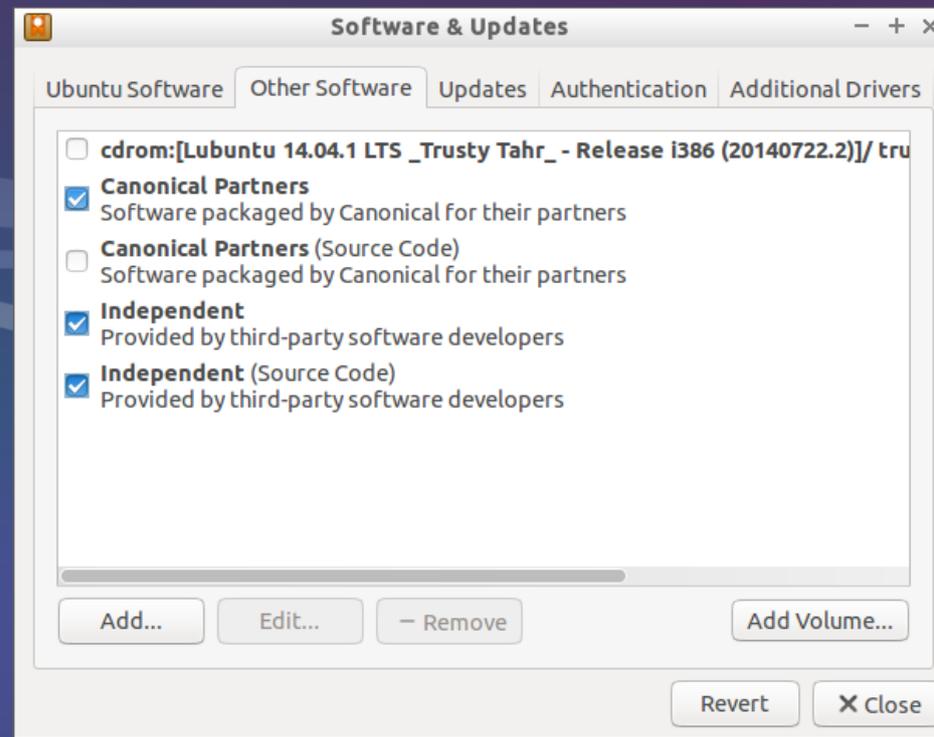
What were repositories? [Click here for the slide about repositories](#)

# Ex. B5: Software Repositories



Ubuntu software shows officially supported channels by the Ubuntu Distribution. You can select additional repositories and which mirror to use.

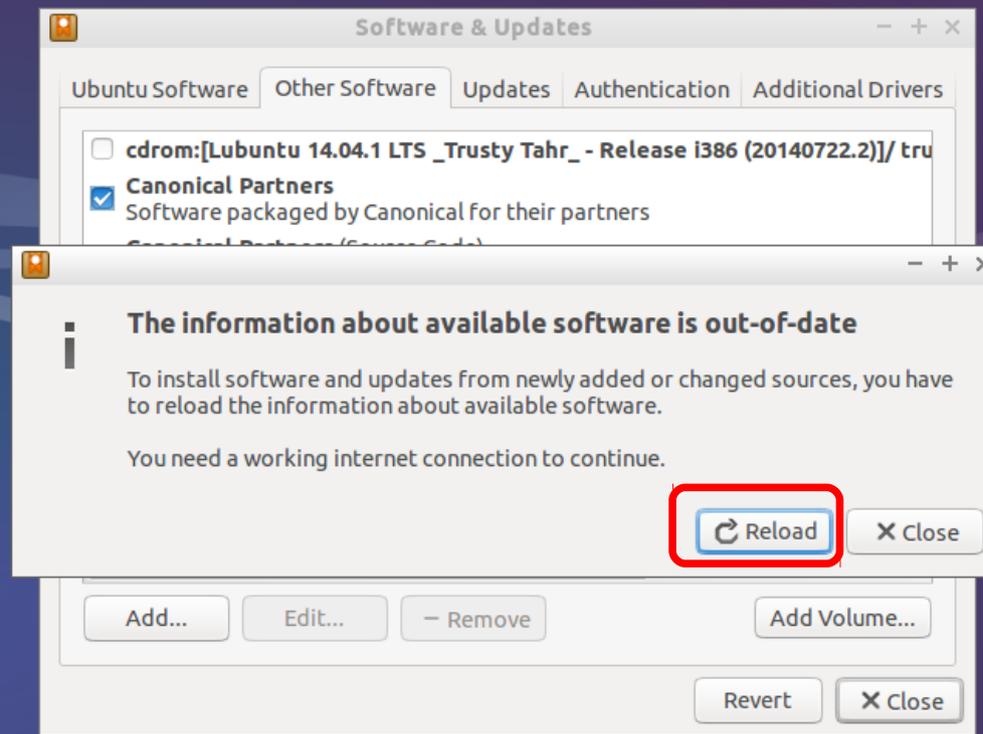
# Ex. B5: Software Repositories



Switch to the "Other Software" tab.

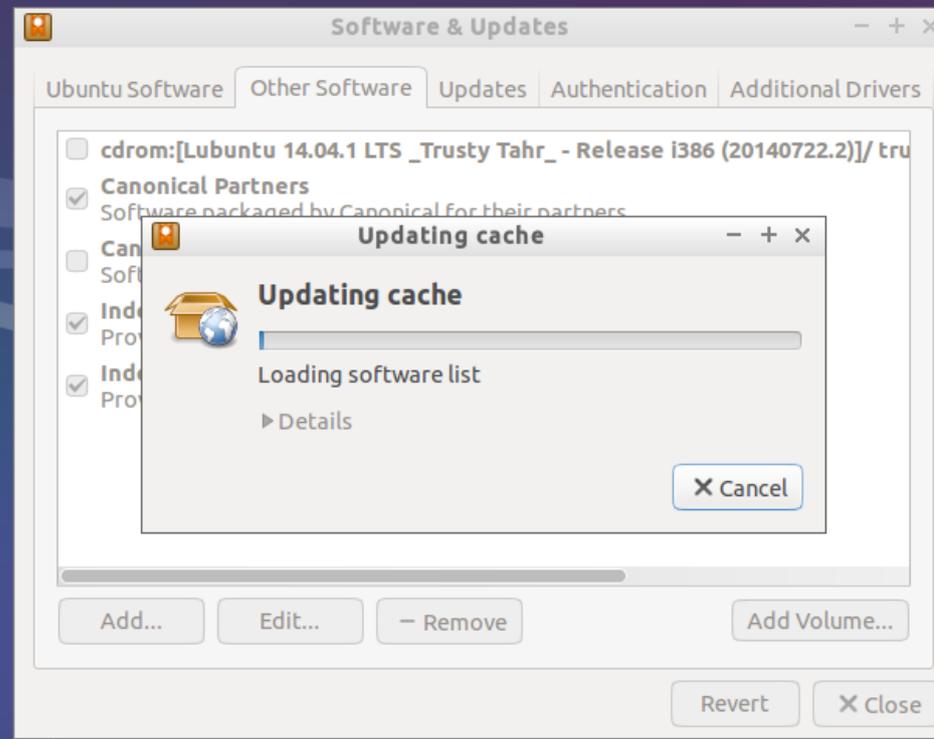
Tick "Canonical Partners". This repository adds external software such as Adobe Flash and other non-free-software stuff. You might need to enter the password.

# Ex. B5: Software Repositories



Click on Reload to update the package database with your selection.

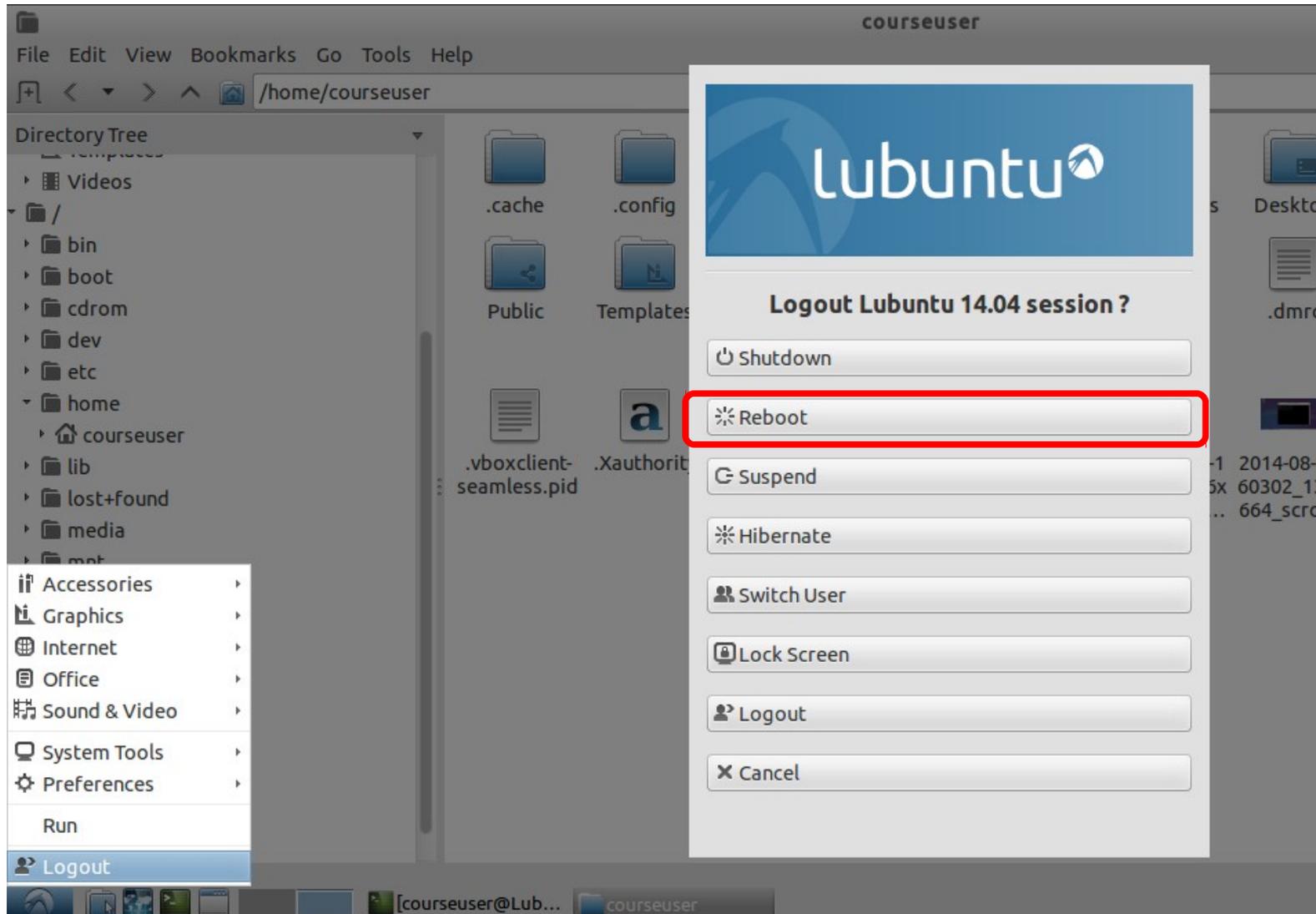
# Ex. B5: Software Repositories



It might take quite some time....

# Ex. B6: Logout dialog: Reboot

## B5.1 Reboot the machine.



# Homework Rules

- All assignments must be sent using **canvas.education.lu.se**
  - Email will **not** be accepted unless there are clear reasons to do that.
- Submit what you have, I will help you progress if I think is not enough for the grade
- My deadlines are not strict. However, if you stick to the deadline I can help you improve your grade. Highly recommended!
- Whatever happens,



# Homework

**New to GNU/Linux? Install your own linux! Repeat the installation steps with a Linux distribution of your own choice.**

Read about linux distributions (slide 39) and pick one you like.

Read about the different installation strategies (slides 42-43, 86-92).

Install your own GNU/Linux. You can either use virtualbox or a physical machine but **make sure you read the dangers of the different installation strategies or you risk of breaking your computer!**

1. If the installer asks for a username, use your first name or your favourite nickname
2. If the installer asks for hostname or computer's name, write: *distributionname-yourname* where
  1. *distributionname* is the Linux distribution you have installed
  2. *yourname* is your first name or favourite nickname
3. Open a terminal, make the terminal window big (NOT the virtual machine window!) and type the following commands:
  1. `uname -a` (and press enter)
  2. `id` (and press enter)
  3. `pwd` (and press enter)
  4. `hostname -f` (and press enter)
  5. `cat /etc/machine-id` (and press enter)
4. Take a screenshot of the whole screen including your windows/linux/macOS desktop that runs virtualbox (you can do with your phone or any other method like the PrtScn key)
5. Run a software update (slide 76-77) and take a screenshot while the process is ongoing
6. Post the **screenshots** as your assignment result and **describe in few words what you did.**

# Homework – Alternative for those who already have linux installed

## Already have GNU/Linux? Make your Linux MNXB01-ready!

If you already know how to install linux or you cannot use a virtual machine (for example because your laptop doesn't meet the Hardware Requirements) make your machine MNXB01-ready.

### 1) Install the following software:

```
build-essential geany geany-plugins git gitg htop kile nedit openssl root-system  
screen subversion texlive texlive-base texlive-full tig wget meld vim
```

`build-essential` is a collection of developer tools, in other distributions the software needed is called in a different way. It includes:

```
libc6-dev libc-dev gcc g++ make dpkg-dev
```

### 2) Install the ARC6 grid software clients:

1) add nordugrid repositories for your distro from

<http://www.nordugrid.org/arc/arc6/common/repos/repository.html>

2) Install:

```
nordugrid-arc-client nordugrid-arc-doc ca-policy-igtf* ca-incommon-igtf-server-ca
```

### 3) When done:

1) send a text file containing the installed packages in your machine. How this is done depends on the distribution you installed, for ubuntu is

```
dpkg -l > packagelist.txt
```

2) Send the output of the commands

```
uname -a  
id  
pwd  
hostname -f  
cat /etc/machine-id
```

# Linux installation: three strategies

**1 Single operating system:** Linux deletes everything on the disk and it becomes the one and only operating system for that computer

- Difficulty: easy
- Constraint: user must own the computer (administrator)
- Pros:
  - Performance: Linux can take all the resources available on the machine
- Cons:
  - The machine can only run Linux programs.

# Method 1 - Single Operating system prerequisites

1.1. Make sure you are using an hard disk that you can **completely wipe. No data but Lubuntu will be left on the disk after installation!! You will lose everything!**

1.2. Download the Lubuntu ISO as seen in the tutorial.

1.3. Burn the ISO on a CD/DVD or on a USB stick. There are several ways to do this, search Ubuntu mainpage or the internet to find how. Start here:

<http://www.ubuntu.com/download/desktop/install-ubuntu-desktop>

1.4. Boot the machine from the ISO. Usually it is enough to insert the burned CD or the USB pen and start the machine.

1.4. Install Linux as we've seen in the course. Remember, **everything on the hard disk will be removed!!**

# Linux installation: three strategies

**2 Virtual Machine:** Run Linux in a virtual machine where it is the one and only system. The virtual machine is run on an hypervisor that is running in the existing OS.

- Difficulty: Medium
- Constraint: User must be able to install software, i.e. VirtualBox hypervisor
- Pros:
  - The machine can run programs for the existing OS **and** linux, **at the same time**
- Cons:
  - No need to touch the existing OS boot: all is taken care by the Virtualization Hypervisor
  - Performance loss due to virtualized hardware.

# Method 2 – Using Virtualbox prerequisite:

<https://www.virtualbox.org/wiki/Downloads>

- 1.1. Download  
*VirtualBox <some version> for Windows hosts*
- 1.2. double click on the Virtualbox file just downloaded and follow the installation procedure on screen.
- 1.3. install linux as seen during the tutorial.
- 1.4. If the machines does not start or you get errors, you may need to enable virtualization instructions in the BIOS. Information here:  
<https://www.wikihow.tech/Enable-Hardware-Virtualization>

# Linux installation: three strategies

**3 Dual boot:** Linux shares the disk with other operating systems.

Requires an alternative boot loader (e.g. GRUB2), a small program that is loaded **BEFORE** any other installed system. This program must be installed at the beginning of the disk.

- Difficulty: Hard
- Constraint: user must own the computer (administrator)
- Pros:
  - Performance: Linux can take all the resources available on the machine
  - The machine can run programs for windows and linux, at the cost of a reboot
- Cons:
  - Rewriting the initial part of the disk usually prevents windows to boot properly. One needs to be very aware of what he is doing.
  - Only one operating system can control the machine at time: requires reboot to switch from one operating system to the other.

# Method 3 – Dual boot prerequisites

**DISCLAIMER: USE THIS METHOD ONLY IF YOU UNDERSTAND WHAT YOU'RE DOING. YOU RISK DESTROYING YOUR EXISTING DATA.**

1.1. Download the Lubuntu ISO as seen in the tutorial.

1.2. Burn the ISO on a CD/DVD or on a USB stick. There are several ways to do this, search Ubuntu mainpage or the internet to find how. Start here:

<http://www.ubuntu.com/download/desktop/install-ubuntu-desktop>

1.3. Make space on the harddisk to fit linux. You usually need at least **100GB**. After the installation that space can no longer be used by windows, so make sure your disk is big enough.

To free space on windows, read about it here:

<http://technet.microsoft.com/en-us/magazine/gg309169.aspx>

1.4. Decide how to install linux: using GRUB, using Windows bootloader... and follow the procedures. More information here:

<https://help.ubuntu.com/community/WindowsDualBoot>

1.4.a **Legacy BIOS**: For Windows up to Windows 7, I suggest to use windows boot loader and a tool called EasyBCD. A guide here:

<http://askubuntu.com/questions/325402/booting-win7-12-04-what-do-i-need-for-easybcd>

1.4.b **UEFI BIOS**: If you have Windows 8 or 10, those support a new booting method called UEFI. This is a very annoying technology and setup is more complicated.

Read this: <https://help.ubuntu.com/community/UEFIBooting>

Follow the instructions here: <https://itsfoss.com/install-ubuntu-1404-dual-boot-mode-windows-8-81-uefi/>

1.5. Install linux as described in the course, but targetting the correct partition/disk you made space for on the disk. **Do not choose "erase entire disk" when prompted.**

# Linux installation: Recommendations

During this course we will use **method 2 (Virtual Machine)**, as it is supposed to work on every platform.

Methods 1 and 3 are only recommended to those who want to learn about the boot process or plan to use Linux for intense computation, or simply to switch to Linux as their main operating system (It's fun! :D).

# Exercise: Install the geany editor

- Use synaptic
- Find the geany text editor and install it.
- Test that it works by finding it in the applications menu.
- Remove the geany package (hint: search for it in synaptic and untick the checkbox!)
- Test that is removed: the icon should not be anymore in the menu.
- What happen if you remove it while you're using it? Discuss with the teacher.

# References

- Lubuntu installation official documentation  
<https://help.ubuntu.com/community/Lubuntu/InstallingLubuntu>
- Lubuntu official documentation / How Tos  
<https://help.ubuntu.com/community/Lubuntu/Documentation>
- Virtualbox  
<https://www.virtualbox.org/>
- Lubuntu Desktop
- GNU, <https://www.gnu.org/home.en.html>
- The Linux kernel organization  
<https://www.kernel.org/category/about.html>
- The Sociotechnical Boundaries of Hardware and Software: A Humpty Dumpty History, Brent K. Jesiek,  
<http://bst.sagepub.com/content/26/6/497>

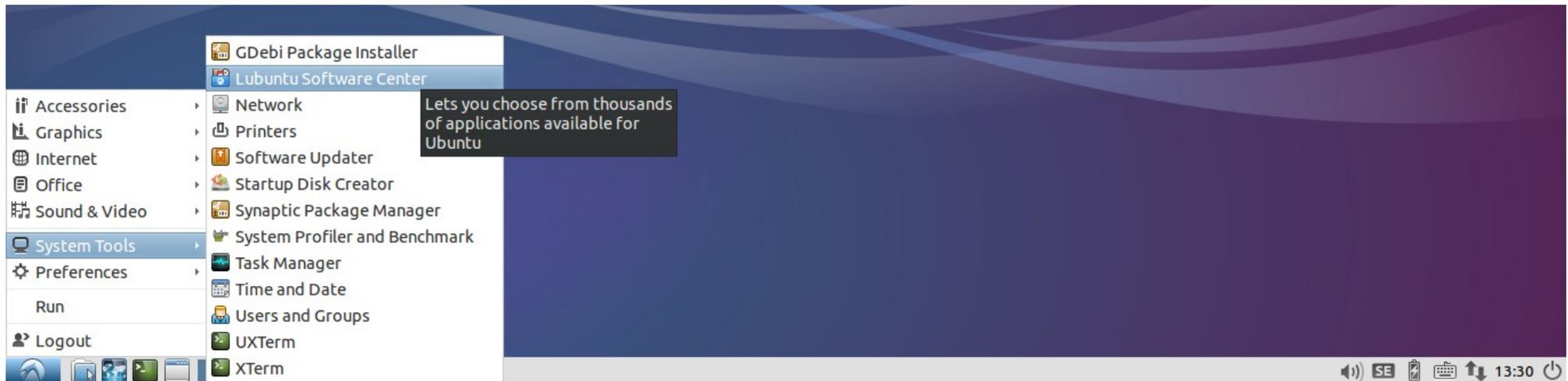
# Pictures references (not complete)

- <https://openclipart.org/>
- <http://members.peak.org/~jeremy/superlative/pix/babbageMachine.jpg>
- [http://en.wikipedia.org/wiki/Eadweard\\_Muybridge](http://en.wikipedia.org/wiki/Eadweard_Muybridge)
- [http://commons.wikimedia.org/wiki/File:ASRock\\_P4i65G\\_motherboard\\_view.jpg](http://commons.wikimedia.org/wiki/File:ASRock_P4i65G_motherboard_view.jpg)
- [http://elaanisvital.com/final\\_png/icon\\_-35.png](http://elaanisvital.com/final_png/icon_-35.png)
- <https://nakamotoinstitute.org/mempool/dont-panic/>
-

# Backup slides

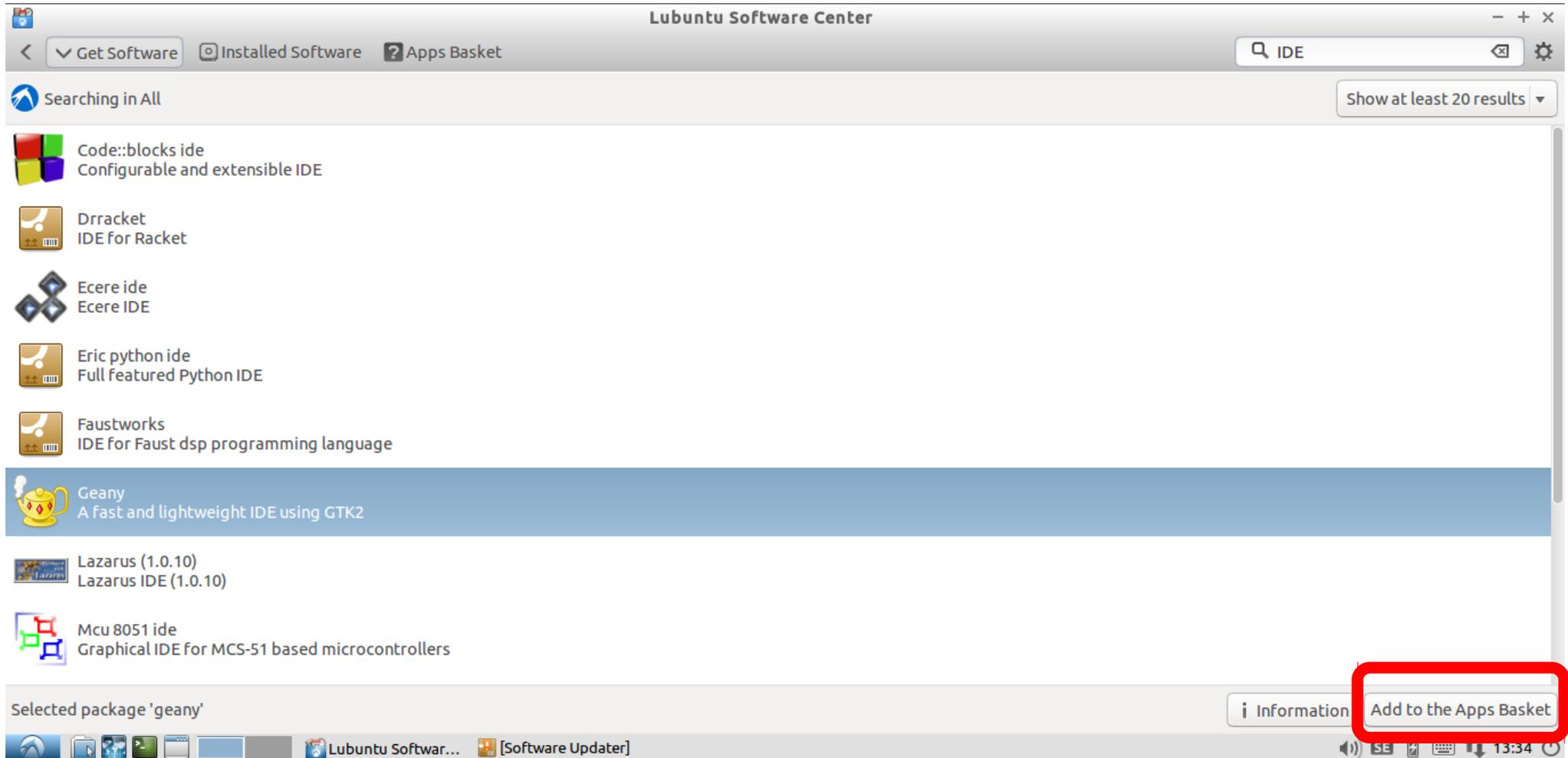
# Ex. 13: Other installation methods

- Normal user: Lubuntu Software Center



# Ex. 13: Other installation methods

- Installing Geany editor



# Ex. 13: Other installation methods

- Installing Geany editor

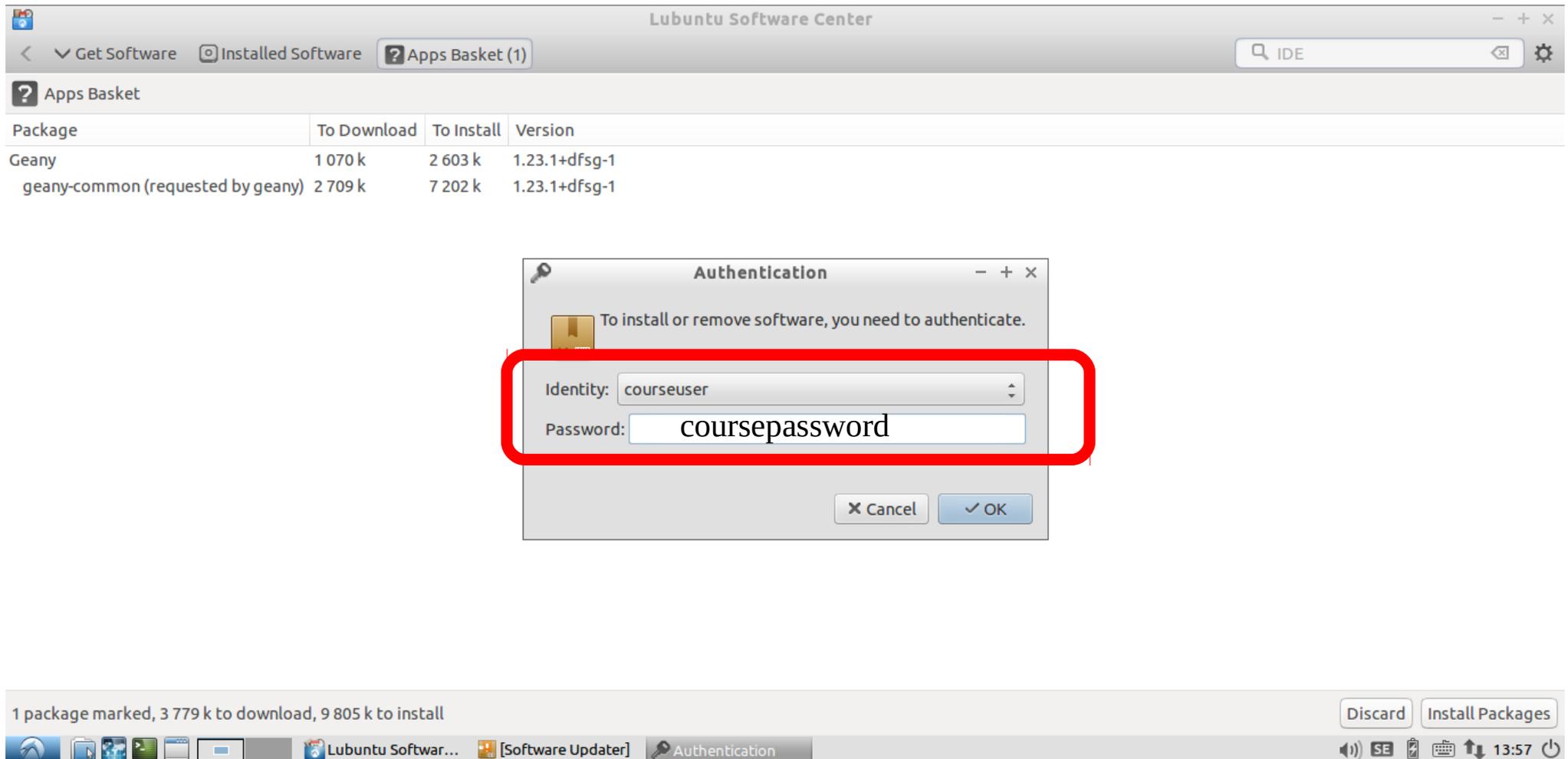
The screenshot shows the Ubuntu Software Center window. The title bar reads "Lubuntu Software Center". The main area is titled "Apps Basket (1)". A search bar contains the text "IDE". Below the search bar, a table lists the packages in the basket:

Package	To Download	To Install	Version
Geany	1 070 k	2 603 k	1.23.1+dfsg-1
geany-common (requested by geany)	2 709 k	7 202 k	1.23.1+dfsg-1

At the bottom of the window, a status bar indicates "1 package marked, 3 779 k to download, 9 805 k to install". On the right side of the status bar, there are two buttons: "Discard" and "Install Packages". The "Install Packages" button is highlighted with a red rectangular box.

# Ex. 13: Other installation methods

- Installing Geany editor



# OPTIONAL: Installing the virtual machine graphical environment using the open source drivers

- Use this method **only if the guest additions didn't work for you (rare)**
- Switch to a text terminal by pressing Right Ctrl + F1 (in a real machine is Ctrl + Alt + F1)
- Login with username *courseuser* and password *coursepassword*
- Run the following commands to achieve the mentioned effect:
  - Replace the existing graphical stack with the generic software stack:  
`sudo apt-get install xserver-xorg`
  - Install Virtualbox open source drivers. This will compile the drivers for you for every version of the kernel that is installed.  
`sudo apt-get install virtualbox-guest-utils virtualbox-guest-x11 virtualbox-guest-dkms`
  - Reboot the machine via command line  
`sudo reboot`
  - After reboot, test that the machine screen resizes every time you resize the machine window.