

# **Instructions of Installing Linux - ROOT on Windows Systems**

Created by Xenophon Karyophyllis, Physics Department of Kapodistrian University of Athens Greece (karyophyllisxen@gmail.com)

For those using Windows, the simplest process is to install Linux through a Virtual Machine.

What we need:

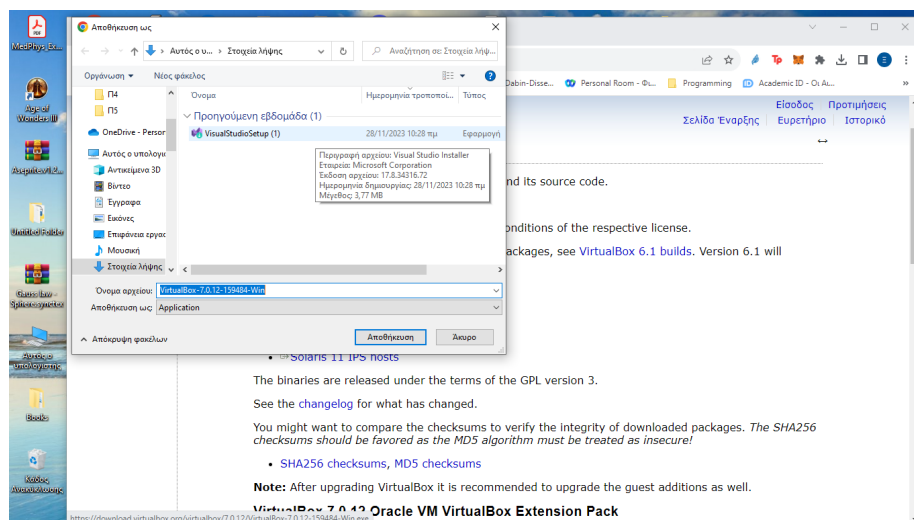
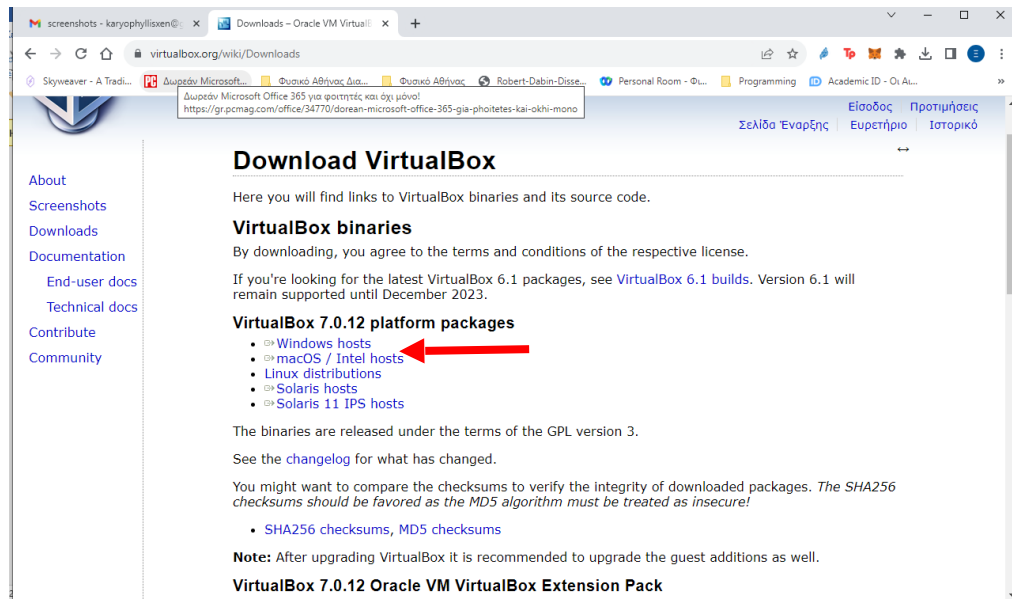
1. VirtualBox (This is what creates the Virtual Machine)
2. A Linux operating system image that we will install on our Virtual Machine.

What are the steps we will follow:

1. VirtualBox - Download - Install
2. Ubuntu iso (Image) - Download - Install
3. Create Ubuntu VM with VirtualBox
4. Install Linux
5. Update Linux
6. Install ROOT prerequisites
7. Install ROOT
8. Use ROOT

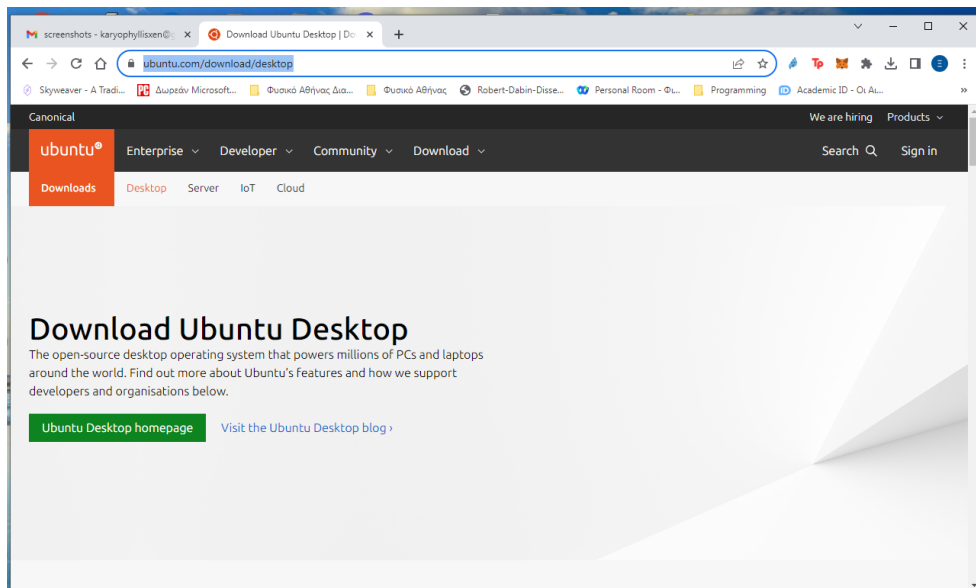
## **1. VirtualBox - Download - Install**

1. Go to the VirtualBox page <https://www.virtualbox.org/wiki/Downloads>, click on the Windows hosts link, and download the installer. Install it on your computer.

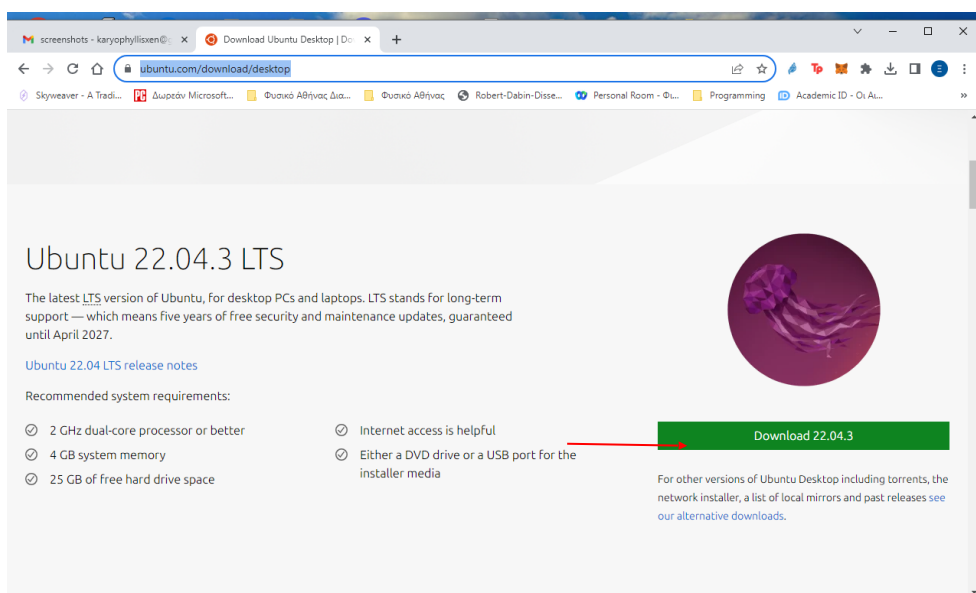


## 1. Ubuntu iso ( Image) – Download – Install

Go to the Virtual Box page <https://ubuntu.com/download/desktop>

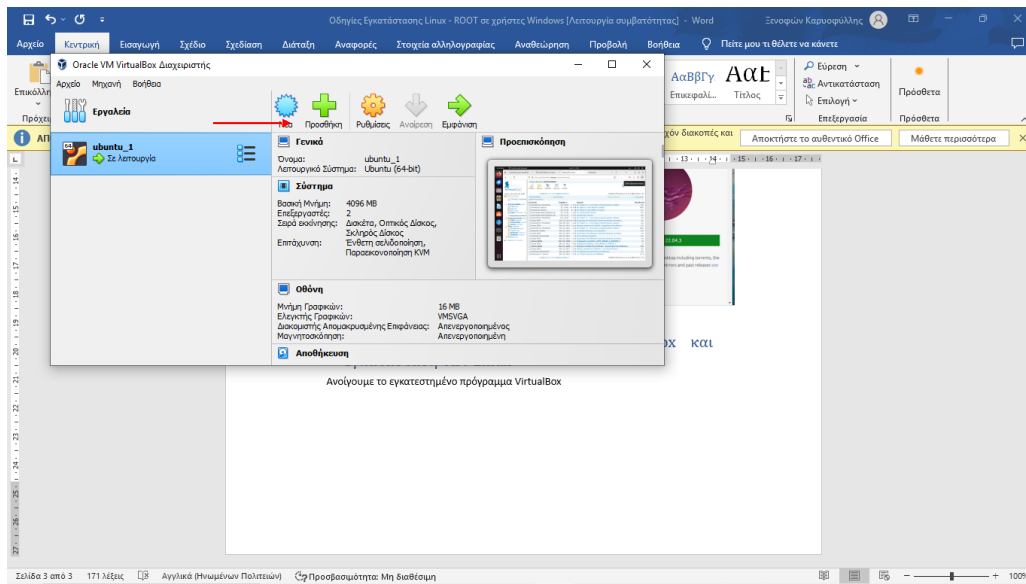


We click on the download link of the specific version that we want to install (in this example I am installing the ubuntu 22.04.3 LTS)

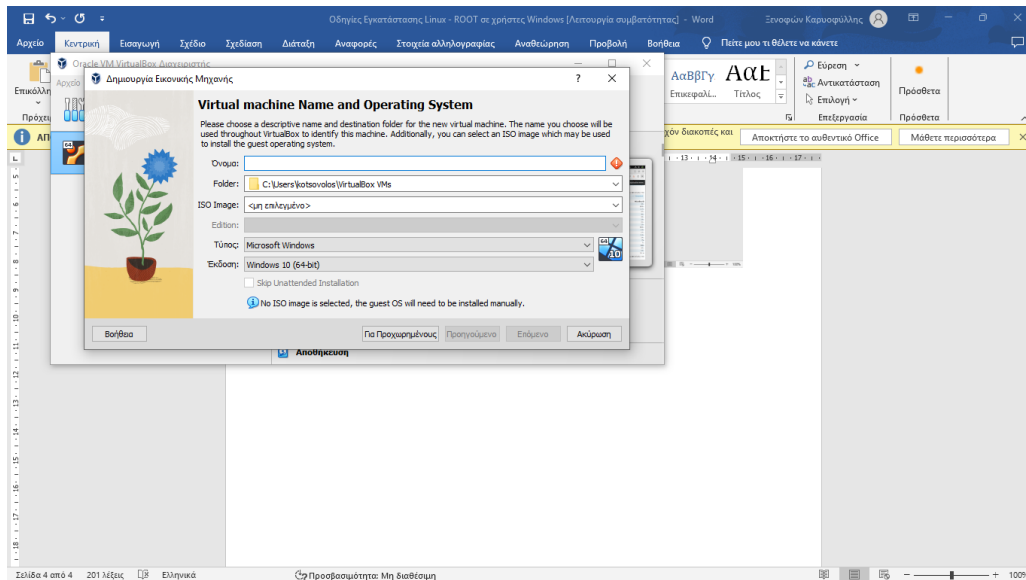


## 2. Create Ubuntu VM with VirtualBox

Open the installed VirtualBox program (if you haven't started any VM, there won't be any active VMs, just like mine doesn't have ubuntu\_1). Click on "New" option.



Then, the following window appears:



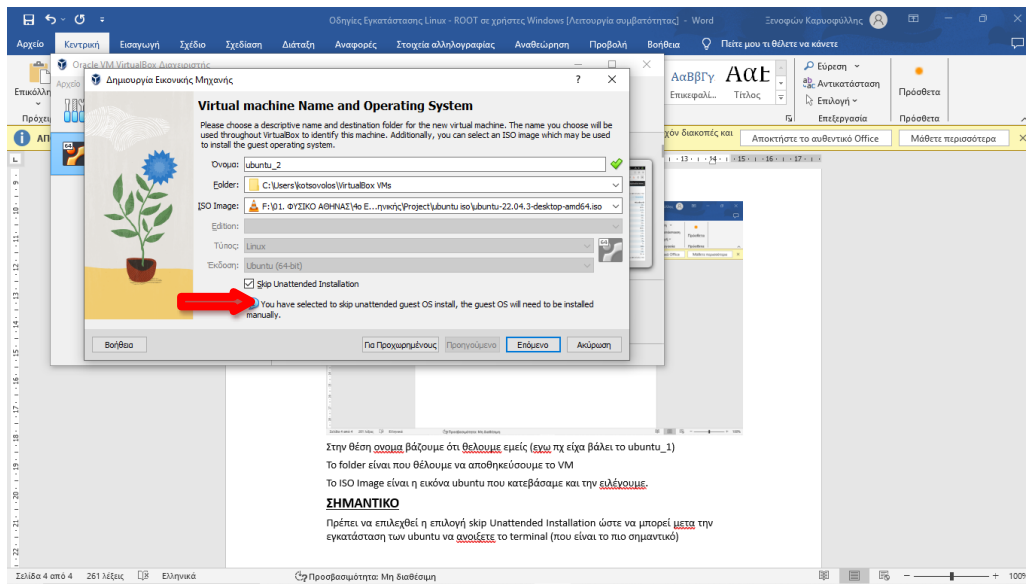
In the "Name" field, input whatever name you prefer (for example, I named it ubuntu\_1).

Choose the folder where you want to store the VM.

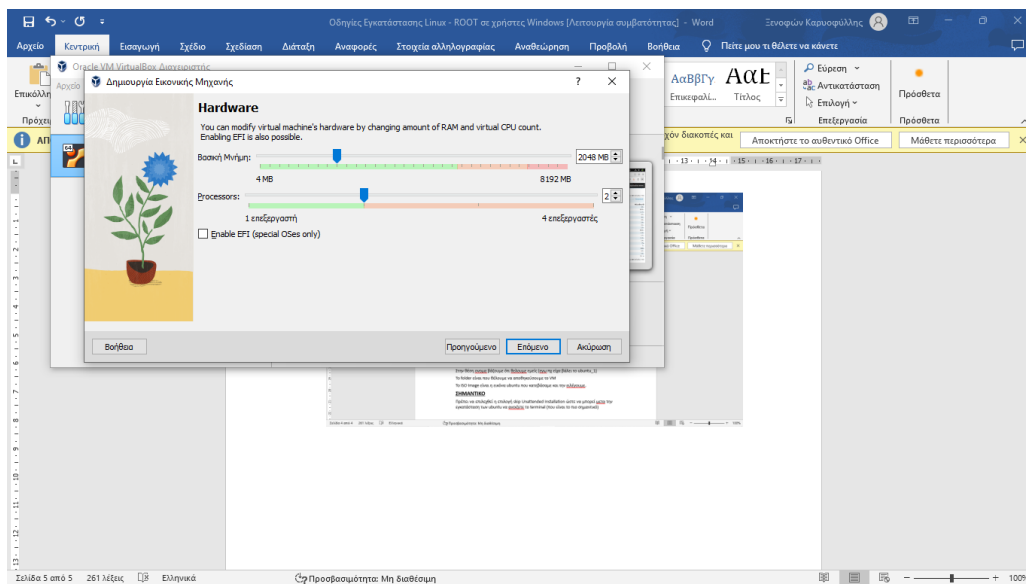
For "ISO Image", select the Ubuntu image you downloaded.

### **IMPORTANT:**

Make sure to select "Skip Unattended Installation" option so that you can open the terminal after Ubuntu installation (which is crucial).



Next, the window prompts us to choose the amount of memory and processors the computer will allocate to the virtual machine. Each person allocates resources as they see fit, but I think a mid-range allocation works well for our purposes.



Next, it allows us to select the size of the hard disk it will keep. For our purposes, 20-30GB should be sufficient.



## 5. Install ROOT prerequisites

To download the ROOT prerequisites, use the following command:

```
sudo apt-get update
```

```
sudo apt-get install git dpkg-dev cmake g++ gcc binutils libx11-dev libxpm-dev libxft-dev libxext-dev libafterimage-dev
```

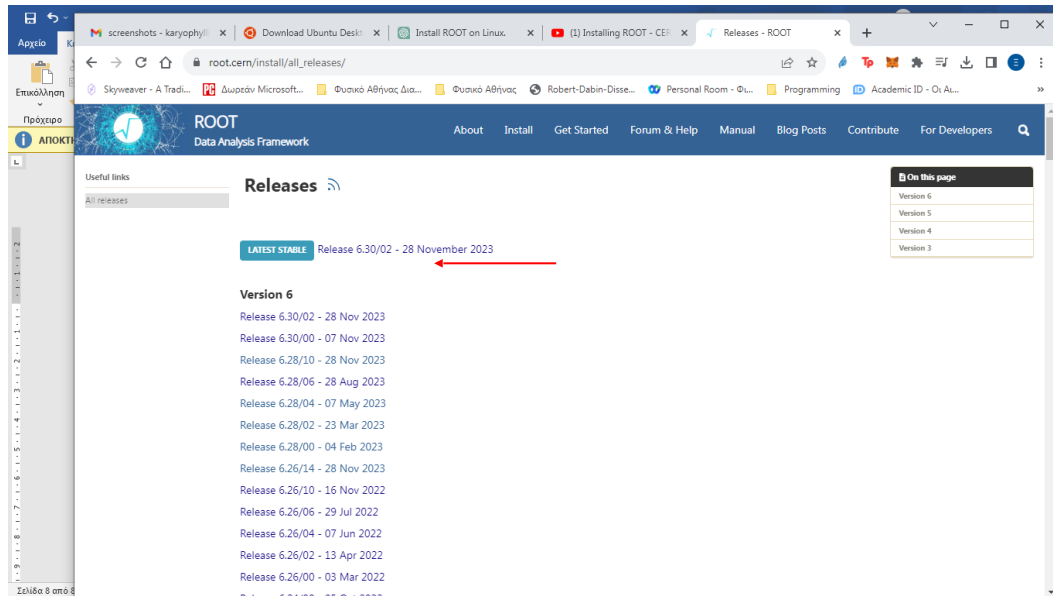
```
empristiss@empristiss-VirtualBox: ~  
gjs libgjs0g  
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.  
empristiss@empristiss-VirtualBox: ~$ sudo apt-get update  
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Hit:2 http://gr.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:3 http://gr.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:4 http://gr.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Reading package lists... Done  
empristiss@empristiss-VirtualBox: ~$ sudo apt-get install git dpkg-dev cmake g++ gcc binutils libx11-dev libxpm-dev libxft-dev libxext-dev  
empristiss@empristiss-VirtualBox: ~$ sudo apt-get install git dpkg-dev cmake g++ gcc binutils libx11-dev libxpm-dev libxft-dev libxext-dev  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  binutils-common binutils-x86-64-linux-gnu build-essential cmake-data dh-elpa-helper fakeroot g++-11 gcc-11 git-man  
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libbinutils libbrotli-dev libc-dev-bin  
  libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl liberror-perl libexpat1-dev libfakeroot  
  libfile-fcntllock-perl libfontconfig-dev libfontconfig1-dev libfreetype-dev libfreetype6-dev libgcc-11-dev libitm1 libjsoncpp25  
  liblsan0 libnsl-dev libpng-dev libpng-tools libpthread-stubs0-dev libquadmath0 librtmp-dev libstdc++-11-dev libtirpc-dev libtsan0  
  libubsan1 libxau-dev libxcb1-dev libxdmcp-dev libxrender-dev linux-libc-dev lto-disabled-list make manpages-dev pkg-config  
  rpcsvc-proto uuid-dev x11proto-dev xorg-sgml-doctools xtrans-dev zlib1g-dev  
Suggested packages:  
  binutils-doc cmake-doc ninja-build cmake-format debian-keyring g++-multilib g++-11-multilib gcc-11-doc gcc-multilib autoconf  
  automake libtool flex bison gcc-doc gcc-11-multilib gcc-11-locales git-daemon-run | git-daemon-sysvinit git-doc git-email  
  git-gui gitk gitweb git-cvs git-mediawiki git-svn libc-dev-bin libc6-dev bzip2 freetype2-doc libstdc++-11-doc libx11-doc libxcb-doc libxext-doc  
  make-doc  
The following NEW packages will be installed:  
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cmake cmake-data dh-elpa-helper dpkg-dev fakeroot g++ g++-11  
  gcc gcc-11 git git-man libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libbinutils  
  libbrotli-dev libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl liberror-perl  
  libexpat1-dev libfakeroot libfile-fcntllock-perl libfontconfig-dev libfontconfig1-dev libfreetype-dev libfreetype6-dev  
  libgcc-11-dev libitm1 libjsoncpp25 liblsan0 libnsl-dev libpng-dev libpng-tools libpthread-stubs0-dev libquadmath0 librtmp-dev  
  libstdc++-11-dev libtirpc-dev libtsan0 libubsan1 libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxext-dev libxft-dev  
  libxpm-dev libxrender-dev linux-libc-dev lto-disabled-list make manpages-dev pkg-config rpcsvc-proto uuid-dev x11proto-dev  
  xorg-sgml-doctools xtrans-dev zlib1g-dev
```

```
Not 29 18:59  
empristiss@empristiss-VirtualBox: ~  
Setting up libalgorithm-merge-perl (0.08-3) ...  
Setting up libtsan0:amd64 (11.4.0-1ubuntu1-22.04) ...  
Setting up libctf0:amd64 (2.38-4ubuntu2.3) ...  
Setting up pkg-config (0.29-2-ubuntu3) ...  
Setting up git (1:2.34.1-1ubuntu1.10) ...  
Setting up libgcc-11-dev:amd64 (11.4.0-1ubuntu1-22.04) ...  
Setting up cmake (3.22.1-1ubuntu1-22.04.1) ...  
Setting up libc6-dev:amd64 (2.35-0ubuntu3.4) ...  
Setting up binutils-x86-64-linux-gnu (2.38-4ubuntu2.3) ...  
Setting up binutils (2.38-4ubuntu2.3) ...  
Setting up dpkg-dev (1.21.1ubuntu2.2) ...  
Setting up libexpat1-dev:amd64 (2.4.7-1ubuntu0.2) ...  
Setting up uuid-dev:amd64 (2.37.2-4ubuntu3) ...  
Setting up libstdc++-11-dev:amd64 (11.4.0-1ubuntu1-22.04) ...  
Setting up zlib1g-dev:amd64 (1:1.2.11.dfsg-2ubuntu9.2) ...  
Setting up gcc-11 (11.4.0-1ubuntu1-22.04) ...  
Setting up g++-11 (11.4.0-1ubuntu1-22.04) ...  
Setting up libpng-dev:amd64 (1.6.37-3build5) ...  
Setting up gcc (4:11.2.0-1ubuntu1) ...  
Setting up libfreetype-dev:amd64 (2.11.1+dfsg-1ubuntu0.2) ...  
Setting up g++ (4:11.2.0-1ubuntu1) ...  
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode  
Setting up build-essential (12.9ubuntu3) ...  
Setting up libfontconfig-dev:amd64 (2.13.1-4.2ubuntu5) ...  
Setting up libfreetype6-dev:amd64 (2.11.1+dfsg-1ubuntu0.2) ...  
Setting up libfontconfig1-dev:amd64 (2.13.1-4.2ubuntu5) ...  
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...  
Processing triggers for man-db (2.10.2-1) ...  
Processing triggers for sgml-base (1.30) ...  
Setting up x11proto-dev (2021.5-1) ...  
Setting up libxau-dev:amd64 (1:1.0.9-1build5) ...  
Setting up libxdmcp-dev:amd64 (1:1.1.3-0ubuntu5) ...  
Setting up libxcb1-dev:amd64 (1.14-3ubuntu3) ...  
Setting up libx11-dev:amd64 (2:1.7.5-1ubuntu0.3) ...  
Setting up libxpm-dev:amd64 (1:3.5.12-1ubuntu0.22.04.2) ...  
Setting up libxext-dev:amd64 (2:1.3.4-1build1) ...  
Setting up libxrender-dev:amd64 (1:0.9.10-1build4) ...  
Setting up libxft-dev:amd64 (2.3.4-1) ...
```

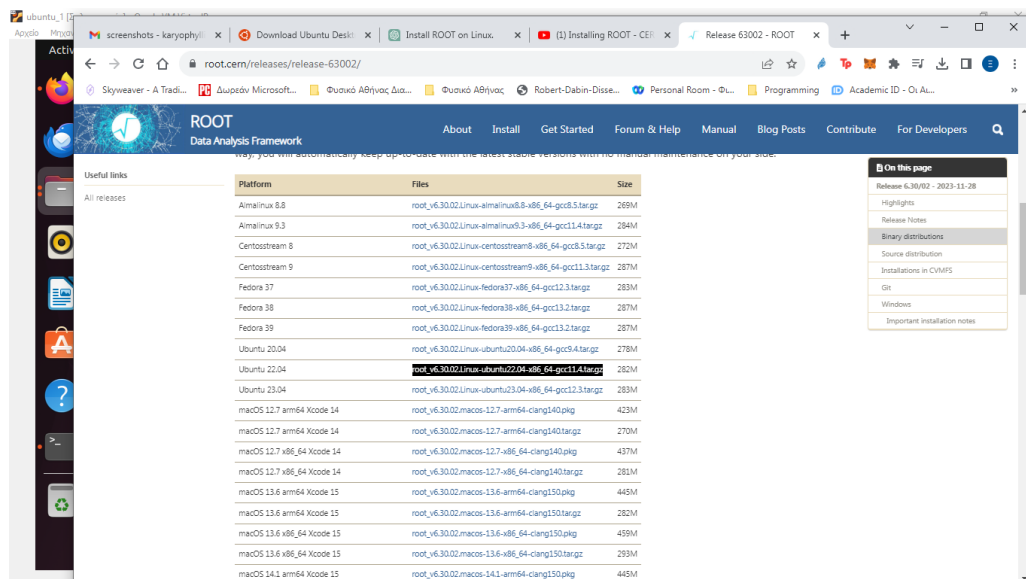
## 6. Install ROOT

Next, we'll install ROOT.

Go to the ROOT page with various releases and choose the release you want. (Currently, I have the latest root v6.03).



Then, go to the corresponding version for our operating system (Ubuntu 22.04 now) and select the tar file and copy it.



Then, go to the terminal and type the corresponding command:

wget <https://root.cern/download/> and we right click in order to paste and have the following command:

wget [https://root.cern/download/root\\_v6.30.02.Linux-ubuntu22.04-x86\\_64-gcc11.4.tar.gz](https://root.cern/download/root_v6.30.02.Linux-ubuntu22.04-x86_64-gcc11.4.tar.gz) and we press enter

Then, it needs to be compiled writing the following command:

Tar -xzf root\_v6.30.02.Linux-ubuntu22.04x-86\_64-gcc11.4.tar.gz



It will install ROOT.

## 7. Using ROOT

Every time we run a terminal, we need to provide the environmental variable path to the terminal.

This can be done with the command:

```
source root/bin/thisroot.sh
```

In Ubuntu 22.04, this can be automated with the following commands:

```
nano ~/.bashrc
```

With this command the file `bashrc` opens with the nano editor

At the end of the file we write the environmental variable command that we wrote above:

```
source root/bin/thisroot.sh
```

After saving the file, in the terminal, type the command:

```
source ~/.bashrc
```

So, whenever we reopen the terminal, simply type 'root', and it will enter the root environment without needing to do it manually.