UNIVERSIDAD PANAMERICANA

OPERATING SYSTEMS FINAL PROJECT

TEACHER
Alfredo Márquez Martínez

TEAM
Luis Fernando Caro Reyna
Sara Carolina Gómez Delgado
Luis Eduardo Robles Jiménez
David Gamaliel Arcos Bravo

Abstract

Our own modified linux distro is focused on one of the most interesting ways to learn programming: Competitive Programming.
We keep this distro simple, with a light desktop environment (running with i3), and obviously with the best tools focused on competitive programming.
It's a Debian based distro, more specifically based on Ubuntu, having all the benefits that it provides but prioritizing the aesthetic and rendiment of our distro.
The best tools were chosen for this distro, and we ventured to develop our own programmed tool focused on our required tasks.

Desktop environment

When developing this distro, we wanted to make it simple, so it contains only the necessary tools and configurations. We wanted to use the mouse as minimally as possible, preferably without using it, so we decided to use a tiling windows manager to keep that simplicity when moving between windows.

I3 is the chosen one. Simple, minimalist, with a customizable environment, this makes i3 a perfect windows manager for our distro, keeping in mind that in competitive programming you always need to keep changing between windows and looking at many windows at the same time.
I3 helps to manage the windows efficiently, with multiple shortcuts that streamline its management.

We use 4 additional programs to complement our i3 customization, those are:

- **Compton -** Helps us to manage multiple desktops in a single session, without losing the easy management on the windows.
- **i3 gaps -** Makes a simple configuration that allows us to keep each single window individually even in the same set of windows.
- **Rofi -** A simple menu that allows us to open external applications and manage it as if those were terminals.
- **I3 blocks -** Show a customized performance status bar on the bottom of our current session, allowing us to get the state of our machine at every moment we require.

As we know not all people are used to working just with windows and we want to make this distro for everyone, we keep the option to run Gnome as the normal Ubuntu if you want the beautiful and simple environment it provides, but keeping in mind it will affect the overall system performance.

Tools

Competitive Programming does not require a lot of sophisticated tools, just something as simple as a web browser, a reliable code editor and the respective compiler to run your programs in your favorite language.

When you are coding on competitive programming, sometimes you have your previous favorite configurations in a template, so we have designed our own template generator to create new files in a most efficient way. The way it works is simple:

First, you need to configure a template for every language you want to work with, and that's it!
Whenever you want to create a new file with that specific template, you just call the command "*template extension nameOfTheFile*".
For example "*template cpp competitive*" will create a cpp file named competitive.cpp with the template specified for c++. If there's not an available template, a new file with the name and extension specified will be opened with the help of vim.
The following code is in charge of the tasks previously described.

Template generator simple code

```
1  #!/bin/bash
2  ext=$1
3  name=$2
4  f=$2"."$1
5  path="~/Desktop/prueba/Template."$ext
6  test -f $path && cp $path $f
7  nvim $f
8
```

The list of the installed tools are:

- Vim - Text Editor
- Brave - Web Browser
- VSCode - Code Editor
- Template generator
- Git
- g++/gcc
- python3
- java
- and so on...

Terminal

No great linux distro is complete without a good terminal. That's why we chose Alacritty as the main terminal in our system.

Alacritty is an open-source terminal emulator, multi-platform with some interesting features. As known as one of the fastest terminal emulators from Linux, its secret comes from the ability of using the GPU to complete some harder tasks in our system.

Alacritty is focused on the simplicity and the performance of our system. As it is focused on the performance, its main characteristics are always focused on keeping this emulator as fastest as possible.

Its predetermined values don't require a user based configuration, but it allows users to configure a lot of aspects in the terminal by modifying the configuration file.

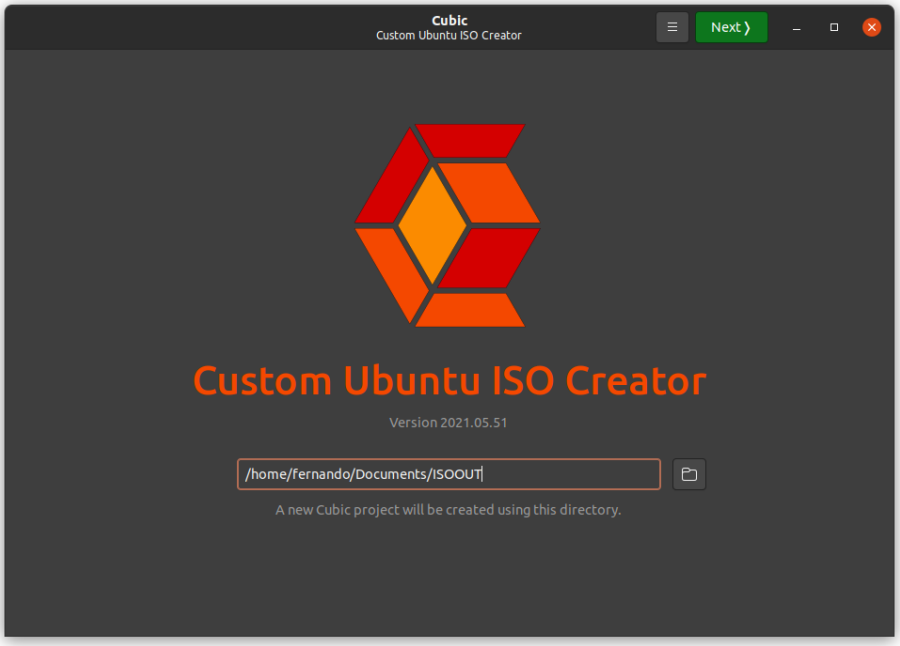
We have made our own configuration file with some characteristics and shortcuts to make this marvelous terminal more dynamic than it was before, but it stills the possibility to keep configuring as the user wants.

How did we create it?

To create our ISO image, we use a tool named Cubic. Cubic allows us to not only create an equal ISO image as the actual distro where we are working, but also to configure some installation configurations and files that we want to add in our distro.

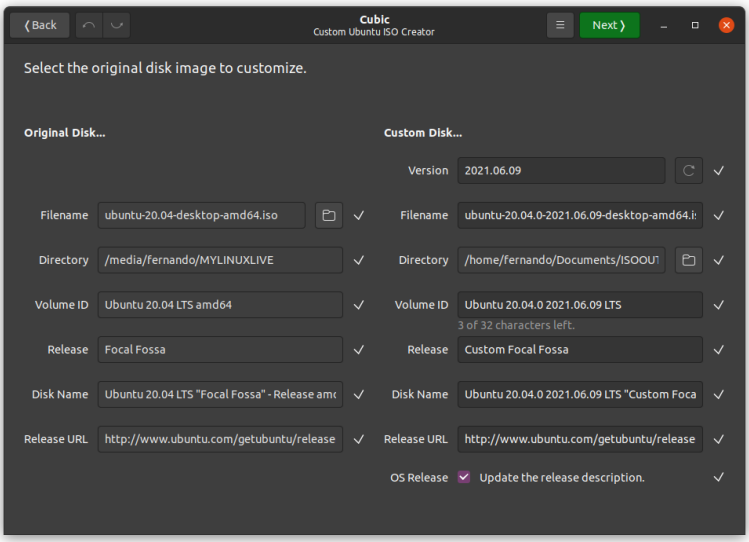
To install it we need to run in a terminal:

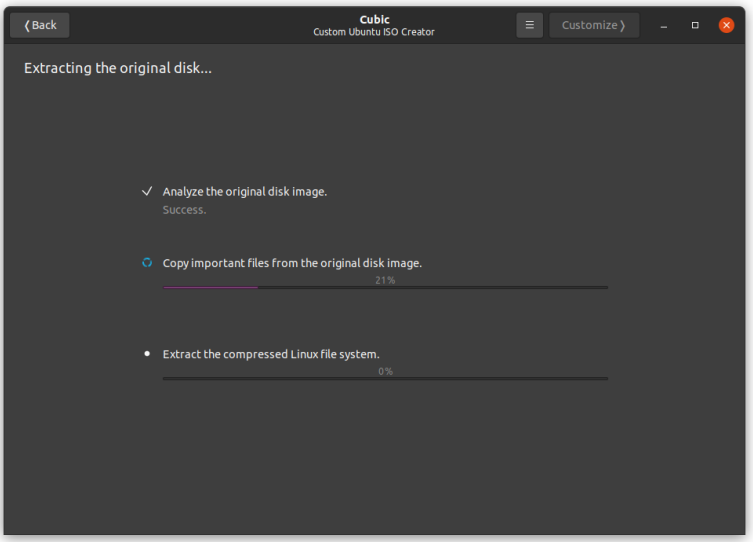
- sudo apt-add-repository ppa:cubic-wizard/release
- sudo apt install cubic



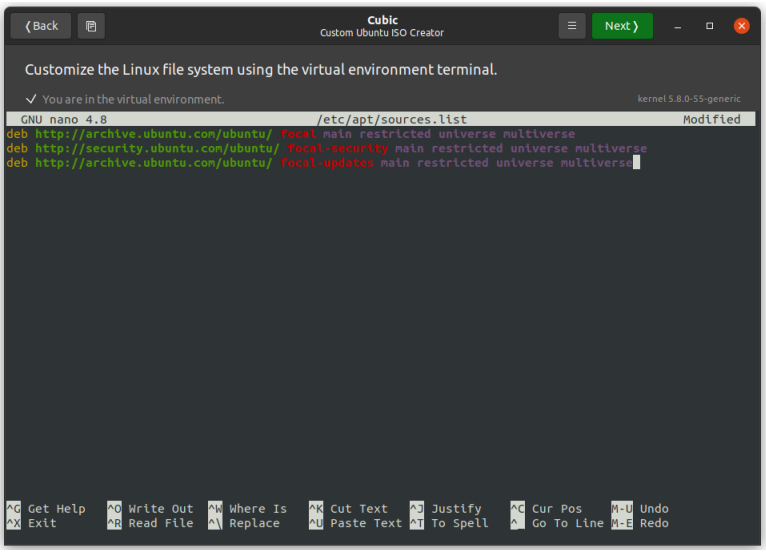
First we need to specify the path where the ISO image is going to be created.

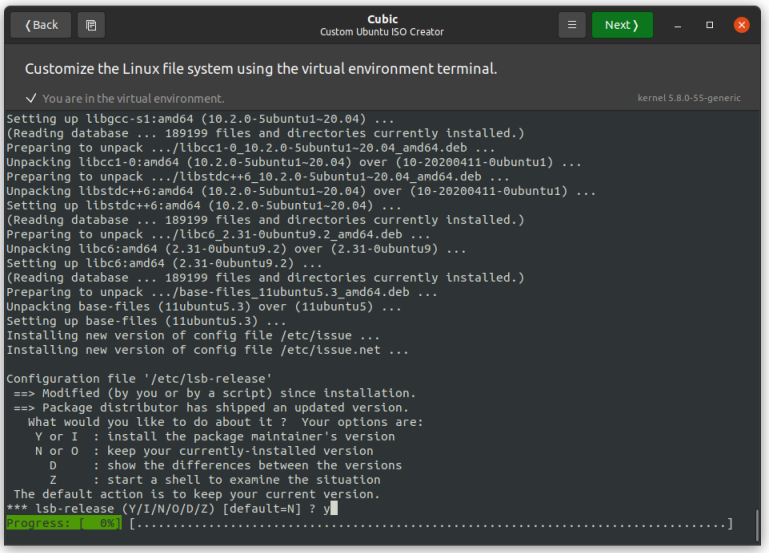
Some extra aspects such as the file name, version, volume, etc are easily managed in the next step, and after filling the required aspects Cubic starts importing files to create the ISO image.



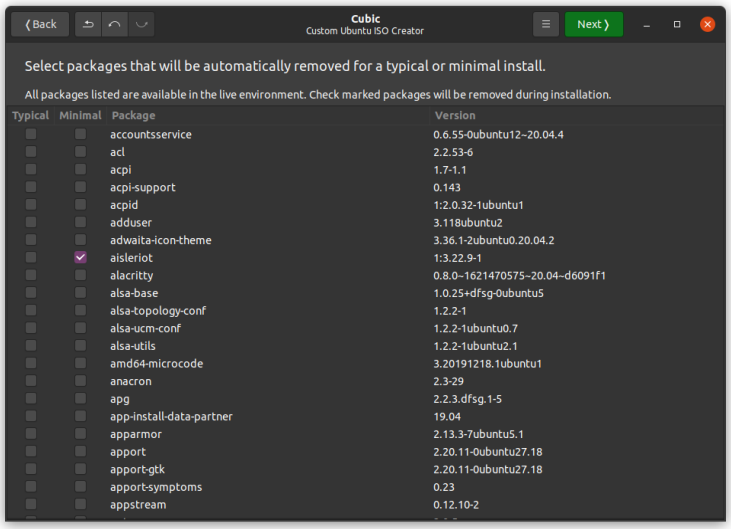


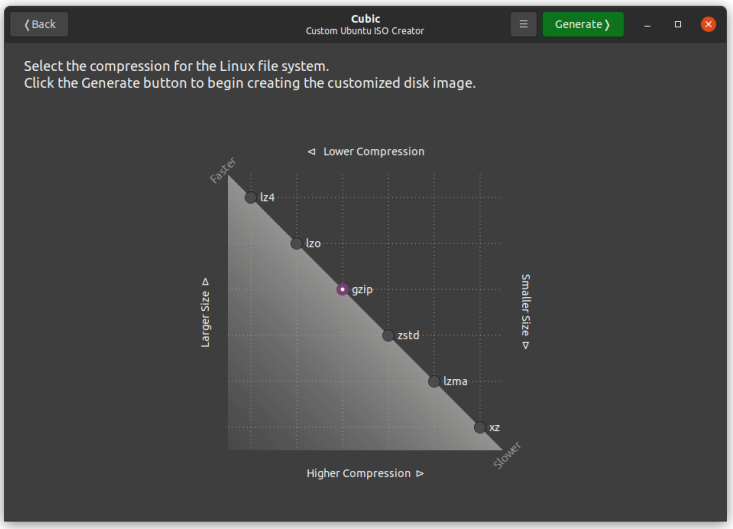
Then you need to customize the Linux file system using the virtual environment terminal. Here is where you can add the repositories, or some packages you want to add in your own distro.



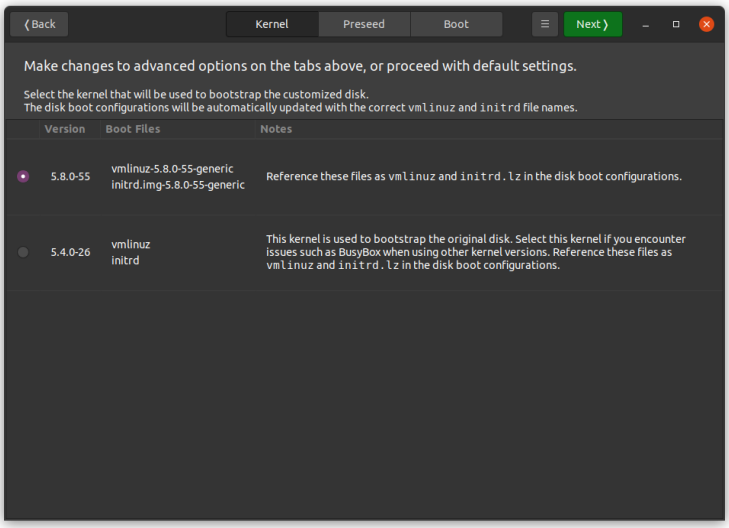


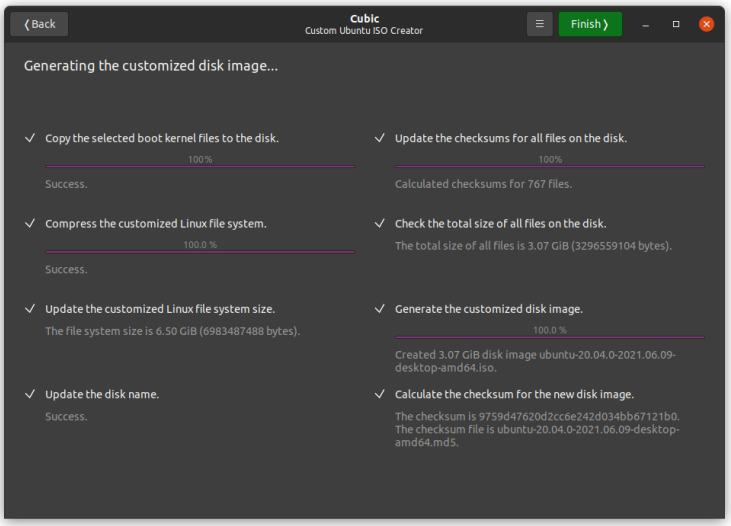
We can also set which packages will be added or removed during the typical or the minimal installation of our system. After this we can also set the compression that our system is going to use.





The last step is choosing the kernel version our system is going to use. After that, the process of creating the ISO image is being completed, and the file is going to be saved in the directory we specify at the beginning.





To include our repositories and tools, we use the following commands by running it at the virtual terminal environment:

Tools

- apt install git
- apt install i3
- apt install g++
- apt install python3
- apt install apt-transport-https curl
- curl -fsSLo /usr/share/keyrings/brave-browser-archive-keyring.gpg https://brave-browser-apt-release.s3.brave.com/brave-browser-archive-keyring.gpg
- echo "deb [signed-by=/usr/share/keyrings/brave-browser-archive-keyring.gpg arch=amd64] https://brave-browser-apt-release.s3.brave.com/ stable main"|tee /etc/apt/sources.list.d/brave-browser-release.list
- apt update
- sudo apt install brave-browser

Alacritty terminal

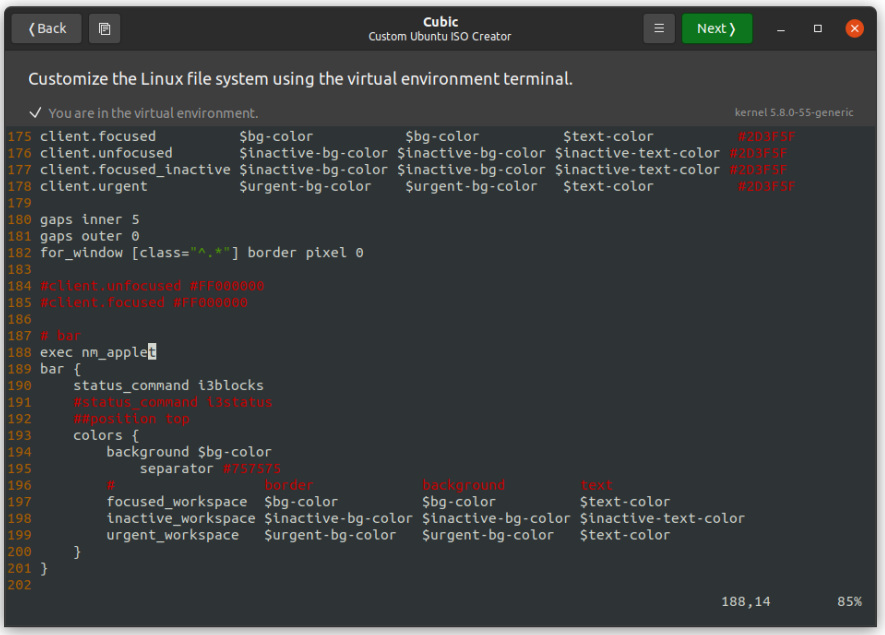
- add-apt-repository ppa:mmstick76/alacritty
- apt install alacritty

I3 configuration *

- apt-get install feh
- apt-get install rofi
- apt-get install compton
- apt-get install i3-blocks

Vim configuration *

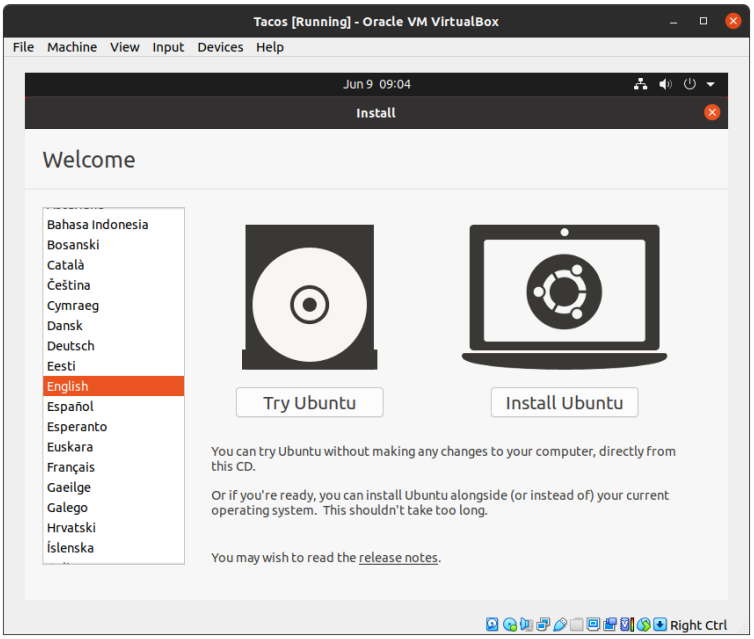
- vim ~/.config/i3/config



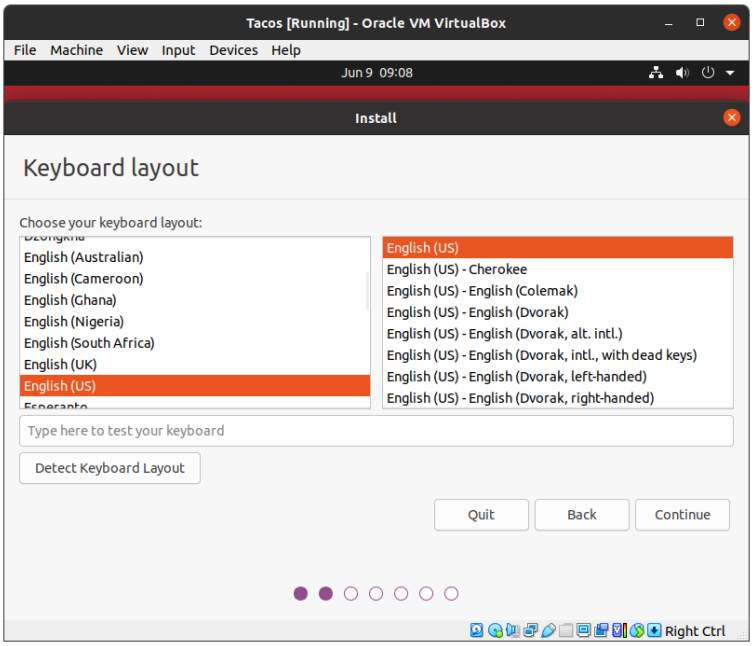
*** Both vim and i3 configurations needs a configuration file that are being installed later**

Installation

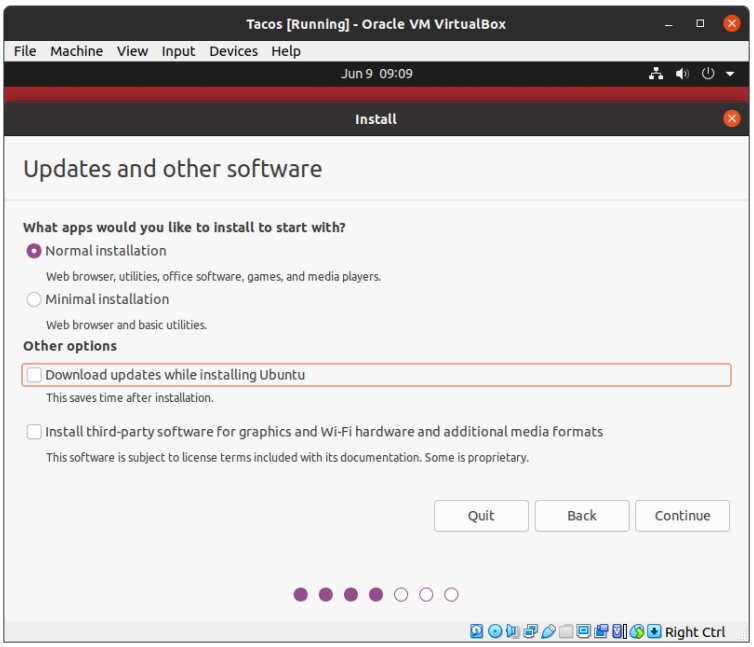
The steps to install this awesome distro are way easier. As it is almost all based on Ubuntu, the installation steps are almost the same as in Ubuntu.



1.- First we need to set the language that our system will use.

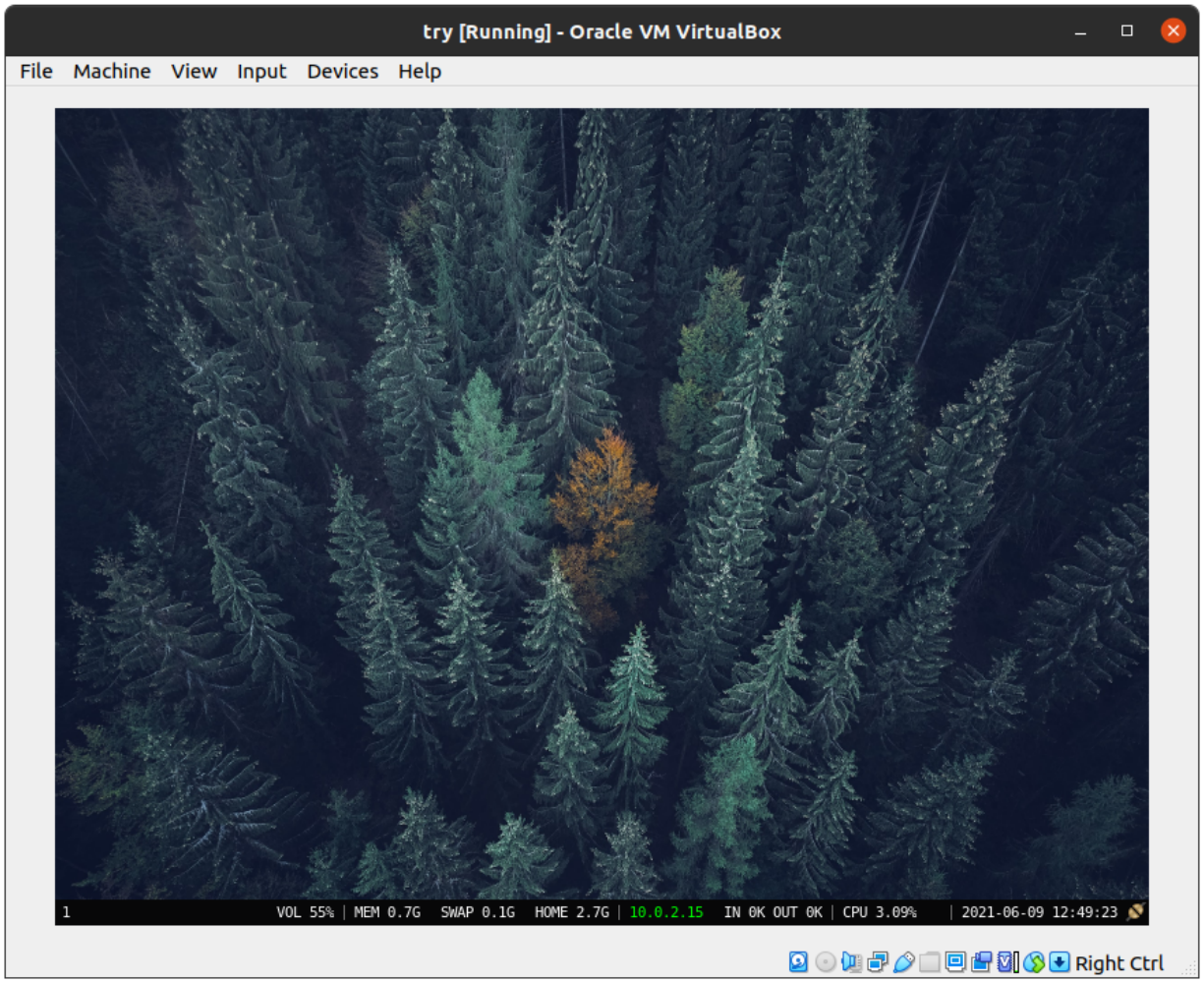


2.- Then we select the keyboard distribution we want in our system.



3.- We select the Minimal installation option, and also download the updates while installing our distro. After this step the installation will start.

After these three simple steps we will have our installation completed, and our distro just ready to be used.



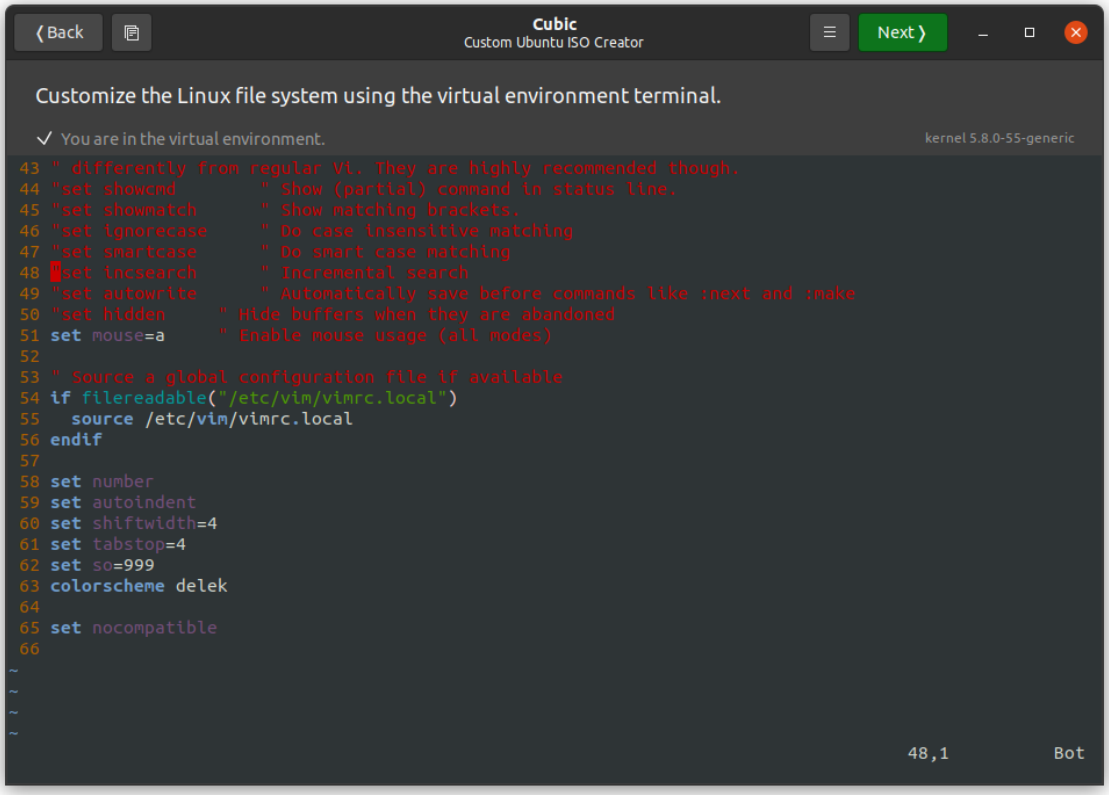
Forward Steps

After the installation, we will need to configure both Vim and I3 to use it more efficiently. This is because the configuration of these two components needs to be configured directly by the user before the installation of the ISO in some specific folders created by the user. That's why it is impossible to upload a previous configuration directory on the ISO image.

We created a specific Github repository to manage the configuration files that we need to add at our system directory. You only need to download the repo and run the pre-build script to add the needed configurations.

Link to the repository -> https://github.com/CaroFernando/TacOSConfigFiles.git

VIM CONFIGURATION FILE



I3 CONFIGURATION FILE