How to Set up a Virtual Machine on Your Computer

Khushika Shah

ENGL 393

Dr. Chris Justice

November 3, 2024

How to Set up a Virtual Machine on Your Computer

A virtual machine (VM) allows users to run an operating system within another operating system, providing a contained environment that functions as a different physical computer. This ability to create systems on a single machine opens up many opportunities for development, and testing what is important for students and professionals in fields like Computer Science and Information Technology. For Computer Science students, especially those interested in systems programming, operating system development, understanding how to set up a virtual machine is a fundamental skill. In addition, many projects can be performed inside of a Virtual Machine. These instructions will familiarize you with the procedures of installing and updating the kernel inside a VM and using the VM for other purposes that cannot be done on the local machine.

With the first set of instructions, you will learn how to set up a VirtualBox on a windows laptop. This provides a hands-on experience that deepens one's understanding of operating systems and their inner workings. These instructions will also improve one's ability to use Git for version control, learn commands, and navigate the terminal. These abilities come in handy while working on open-source projects, creating software, or resolving challenging technical problems. Gaining proficiency in these systems increases your flexibility with different types of operating systems, especially Linux which is open-source and more secure than other systems, increases your ability to excel in the IT and software development industries.

First part of the process is to install the debian environment where the machine can be created. This part involves downloading the Virtual Machine (VM), which creates a safe and isolated environment for kernel development. Later, after setting up the virtual machine you will learn how to access a kernel in the system which is an integral part of the computer and allows communication between applications and the hardware of the computer. You will also learn how to connect a github account to the virtual machine so any projects created on github can be cloned and worked on the virtual machine.

This beginner-friendly approach to setting up a virtual machine, ensures that each essential configuration detail is covered. While there are other methods and tools for creating VMs, like VMware or cloud-based solutions, using VirtualBox and Debian 11 offers several advantages. VirtualBox is open-source and widely supported, making it accessible and cost-effective, especially for students. Debian 11, with its LTS kernel, is also ideal for secure testing, building, and modifying code. By following this structured setup, users gain hands-on experience with elements like downloading, root access, and package installation which are practical skills that translate well to real-world IT and development tasks.

The instructions are preceded by the requirements for setting up a virtual machine. The steps are also followed by a glossary and works cited page. All terms denoted with an asterisk (*) are defined in the glossary page.

Hardware Requirements:

- Computer**:** A Windows laptop
- An x86 CPU
  - Your CPU must support virtualization.
- A 64-bit host operating system.
- 8+ GB of RAM.
- 32+ GB disk space or external storage.

Software Requirements:

- Virtualization Platform: We will be using VirtualBox VM.
- Guest Operating System: Debian 11 (amd64/64-bit version) will be the OS of choice.
- *Linux Kernel Version: Based on the Linux Kernel version 5.15.67, a Long-term support (LTS) release, providing stability and extended support.

Downloading VirtualBox and Debian:

● VirtualBox 6.1:

https://www.virtualbox.org/wiki/Download_Old_Builds_6_1

● Debian 11.0.0 ISO:
https://cdimage.debian.org/mirror/cdimage/archive/11.0.0/amd64/iso-cd/debian-11.0.0-amd64-netinst.iso

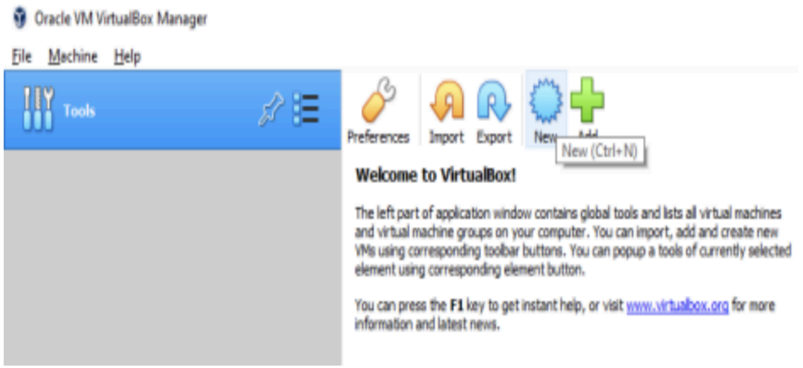
Why Debian 11 and Linux Kernel 5.15.67?

- *Debian 11 provides compatibility with a wide range of applications.
- The LTS kernel ensures that we have a stable base for the kernel.

Installing the VM

Perform the following tasks to create the environment that will be used to build a linux kernel on your computer:

1. After downloading the virtual box and debian, you should be presented with this screen. Select “New” to create a new virtual machine.



Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

Figure 1. *The display screen when VirtualBox is opened*

2. Provide a name for your *VM. The type must be Linux, and the version must be Debian (64-bit).
3. Allocate at least 4 GB of *RAM (8 GB+ is ideal).
4. Select “Create a virtual hard disk now” and “VDI (Virtual Disk Image)” for hard disk type.
5. Select “Dynamically allocated” for storage on the physical hard disk.
6. Assign your virtual hard disk at least 32 GB in disk space and click ‘Create’. Next, we instruct VirtualBox to boot our Debian 11.0.0 ISO.
7. Select Settings for your virtual machine from the previous steps. The machine you created appears in a list of VMs under the name you specified.
8. Navigate to the Storage option and right click the Empty removable storage slot beneath the *IDE Controller.
9. Right clicking produces a context menu -- select "Choose a disk file" and find the Debian 11.0.0 ISO. A link exists to download this ISO under this section requirements.

Setting and Opening up VM

Perform these tasks after installing the kernel environment and opening the virtual box.

1. After selecting ‘start’ in the virtual machine you will be prompted to enter a username and password for your account. Enter a secure and memorable password since you will need this to log into the environment every time and to access root privileges (mentioned later).
2. Select the 64-bit version of Debian Linux in the new VM. Be sure to give your VM enough RAM to work with while not disturbing the rest of your PC.
3. Increase the number of cores that are devoted to your VM in the virtual machine configuration.
4. After you create the VM, select it in the main screen of VirtualBox, hit the Settings button, then in that window select System, go to the Processor tab, and increase the slider labeled Processor.

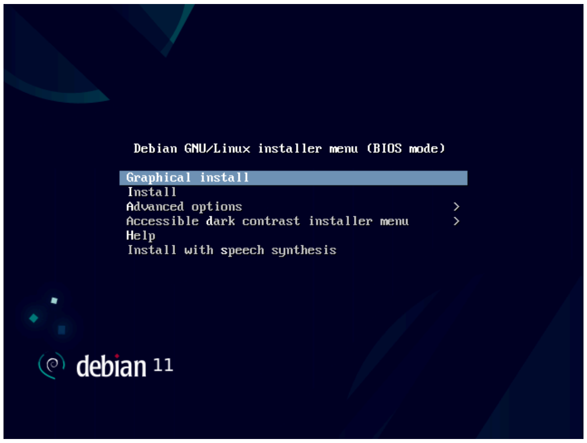

Debian GNU/Linux installer menu (BIOS mode)
Graphical install
Install
Advanced options >
Accessible dark contrast installer menu >
Help
Install with speech synthesis
debian 11

Figure 2: Installing GUI in debian

5. When installing Debian within the VM, use the graphical install option. Be sure to use the 64-bit installation.

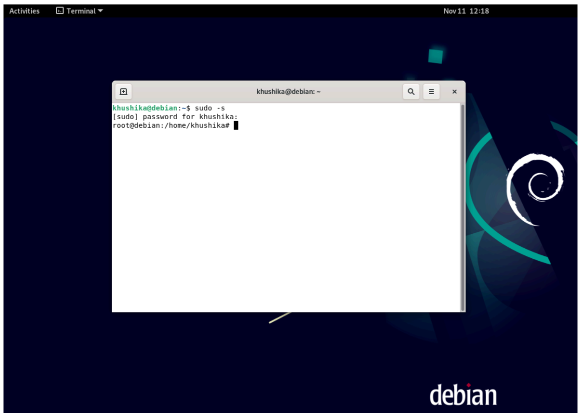
   Note: I personally suggest using GNOME, Cinnamon or MATE (*graphical user interface (GUIs)), as they are relatively lightweight and will ensure you have space to work with later your virtual hard drive.

6. When you get to the part about partitioning, you should select the guided partition option and the option to create one large partition on the disk (having a small swap partition is fine as well).
7. Finally, when installing packages later in the installation, be sure to select at least one graphical environment from the list if you want a GUI. In that same step, you should also select the *SSH server option.
8. Late in the installation, the installer may ask you where to install the boot loader. Be sure to select to install it on the VM's hard drive (/dev/sda).
9. Go to Activities in the top left corner and select ‘terminal’. If the terminal is not in the bar option, search for it in the search bar.

Obtain the Linux Kernel Sources

The next part of this project is to obtain the Linux kernel sources. The following set of steps will guide you through the process.

1. Obtain **root** privileges. These elevated privileges allow you to perform actions as the "root" user, which is similar to an administrator in other operating systems like Windows. When you switch to the root account, you gain full control over the system, so it's essential to use this power responsibly.
    a. Use su - to switch to root user (You will be prompted to enter your password).
    b. Using sudo command to make sure that the command entered is in sudo mode.
    c. Using sudo to open a root shell.
    d. Use exit to exit out of root mode.



```
khushika@debian:~$ sudo -s
[sudo] password for khushika:
root@debian:/home/khushika#
```

Figure 3: *Example of going into root with sudo -s command*

Caution: Pay careful attention to what steps should be run as root. In these instructions, we specifically mention which steps should be run as root. Anything else should not be run in a root shell and should be run as your normal user account. If you fail to do so, you will have problems with your build.

```
apt-get update
apt-get upgrade
apt-get dist-upgrade
apt-get install sudo
usermod --append --groups sudo YourVMUsername
reboot
```

Figure 4: commands for debian installation

2. Update all packages on your Debian installation by **as root**, run the following commands in the terminal, substituting the username you log into the VM with in the second-to-last command as appropriate.
3. Wait for the VM to *reboot itself. After the VM reboots, you should be able to use sudo for running commands as root. You will need to type sudo in front of each command if you are running multiple commands at a time as root.
4. In a terminal, **as root**, install the required packages with the following command:

   ```
   apt-get install build-essential valgrind nano patch diffutils curl fakeroot git pkg-config gedit libssl-dev libncurses5-dev libelf-dev bison flex bc dkms rsync clang-format dwarves qtbase5-dev qtchooser qt5-qmake qtbase5-dev-tools
   ```

5. In a terminal, **as root**, run the following commands:

   ```
   cd /usr/src
   chmod 777 .
   ```

Link Github and VM

Linking your github and your VM will let you clone your projects from github to your VM and after testing working on it you can push those modifications back to github. This is a useful skill in computer science.

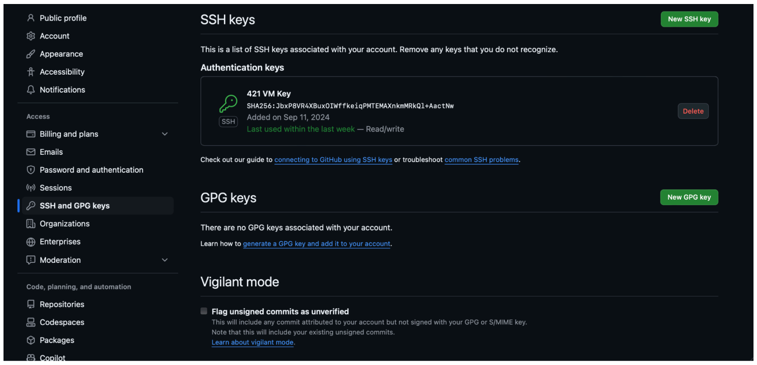
1. Make a github account at github.com if you don't already have one.
2. Go to menu > your repositories and click 'new' and name your repository based on the project you are creating.
3. To work on repositories hosted on GitHub, you will need to have a ssh key set up on your VM and registered on your GitHub account. To do so, run the following command in a terminal as root in your VM. (replacing *yourgithubgmail.com* as the gmail linked to your Github account)

   ```
   ssh-keygen -t rsa -b 4096 -C "yourgithubgmail.com" -f ~/.ssh/id_rsa
   ```

4. You can accept the default settings by pressing Enter each time. When asked, it's recommended to set a passphrase for added security. Make sure you remember this passphrase.
5. Add the SSH Key to Your GitHub Account:
   a. Open your VM's web browser and go to github.com.
   b. Log in with your GitHub credentials.
   c. Go to Settings > SSH and GPG keys.
   d. Click the New SSH key, give it a title, and then return to your terminal.
   e. Copy the SSH Key from terminal:

      ```
      cat ~/.ssh/id_rsa.pub
      ```

6. Paste the copied SSH key into the "Key" box on the GitHub page and click “Add SSH key”.



SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

421 VM Key
SHA256:JbxP8VR4XBuxOIWffkeiqPMTEMAXnkmMRkQl+AactNw
Added on Sep 11, 2024
Last used within the last week — Read/write

Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.

GPG keys

There are no GPG keys associated with your account.

Learn how to generate a GPG key and add it to your account.

Vigilant mode

□ Flag unsigned commits as unverified
This will include any commit attributed to your account but not signed with your GPG or S/MIME key.
Note that this will include your existing unsigned commits.
Learn about vigilant mode.

Figure 5: *how ssh key should look like settings > SSH and GPG keys*

7. In a terminal, run the following commands as your **normal user** (not as root). Replace yourusername with your github username, yourgithubgmail.com with your UMBC email account, and make sure to put your full name where suggested.

```
cd /usr/src

git config --global user.email yourgithubgmail.com

git config --global user.name "Your Full Name"

git clone git@github.com:yourRepositoryName projectName

cd projectName

git remote remove origin

git remote add origin git@github.com:yourRepositoryName/projectName-yourusername.git

git push -u origin main
```

8. The sources in the directory /usr/src/projectName will be referred to as the **working copy** of the kernel. This working copy of the kernel is where you will make all your changes going forward.

Troubleshoot**:** If the git push -u origin main command returns an error message about the repository that it does not exist or you do not have permission to access it, then make sure that the repository has been created and that you have correctly created and added your SSH key to

GitHub, and that you are not trying to run the command as root. Also ensure that you properly run the two git remote commands inside the /usr/src/projectName directory.

Glossary:

Kernel: The core part of an operating system that acts as a messenger between the software and hardware of the computer.

Linux Kernel Version: A specific version of the Linux kernel

Virtual Machine (VM): A computer environment that runs an operating system within another operating system

RAM: Short for Random Access Memory, which is a type of computer memory that temporarily stores data.

Debian: A stable linux based operating system used for development.

IDE Controller: Controller that allows hard drives and storage devices to communicate with the computer.

GUI: Short for Graphical User Interface, it is a visual way to interact with a computer and what the user is presented with on a computer.

SSH: Secure shell is a protocol allowing remote access to a computer or a network

Reboot: restarting a computer for refreshing.

Works Cited

“4.2. Installing, Step by Step.” *The Debian Administrator's Handbook*, https://debian-handbook.info/browse/stable/sect.installation-steps.html. Accessed 11 November 2024.

“How to Create a Virtual Machine : 10 Steps.” *Instructables*, https://www.instructables.com/How-to-Create-a-Virtual-Machine/. Accessed 11 November 2024.