THE LINUX FOUNDATION TRAINING

Coursera: Open Source Software Development, Linux and Git

Welcome to Linux Foundation E-Learning Training

2020/09/03 Version 8.8

If you have any further questions after reading this document, the answer may be found in the FAQ on the website: http://bit.ly/LF-FAQ or http://training.linuxfoundation.org/linux-courses/general-information-and-faq

Contents

1 Hardware Requirements

Students are expected to **provide their own computers** for **Linux Foundation** courses.

Table 1: **Hardware requirements for Coursera: Open Source Software Development, Linux and Git**

Required CPU Architecture	x86_64
Preferred Number of CPUs	2 (minimum 1)
Minimum CPU Performance	2000 bogomips
Minimum Amount of RAM	1 GiB
Free Disk Space in $HOME	5 GiB
Free Disk Space in /boot	128 MiB
Internet Access	Required
Virtual Machine	Acceptable
OS required for class	Linux

The ready-for.sh script in a later section will automate verifying all these values for you.

You need to use a x86_64 processor with at least 2 (minimum 1) CPUS; the number of CPUS can be counted as cores x hyperthreads. The following two commands will tell you your processor architecture and the number of CPUS.

```
uname --processor
nproc
```

The cumulative BogoMIPS of the CPUs needs to exceed 2000 bogoMIPS. The following shell code will tell

you your cumulative BogoMIPS.

```
lscpu | awk '/^CPU.s.:/ {C=$2}; /^BogoMIPS:/ {B=$2} END {print C*B}'
```

Your computer needs at least 1 GiB available. The following shell code will tell you how much RAM you have.

```
free -h | awk '/^Mem/ {print $2}'
```

Your computer needs at least 5 GiB of free disk space. The following shell code will tell you how much free disk space you have.

```
df -h --output=avail $HOME
```

The Linux Foundation logistical staff may be consulted as required for further clarification.

1.1 Using a Virtual Machine Instead

Virtual Machines

If you elect to use a Virtual Machine (instead of native Linux) bear in mind that the hardware requirements double, since you now need enough CPU/RAM for the host operating system as well as the guest OS.

Using a VM for this course can make things faster/easier; if you make a fatal mistake, a simple reboot of the VM will restore things to normal.

More on what distro and software needs to be installed on the VM can be found in the Software Requirements chapter below.

If you want to build your own VM image

You can make sure your own Virtual Machine image is properly setup for the class using the ready-for.sh script which can be found as follows:
https://training.linuxfoundation.org/cm/prep?course=Coursera

1.2 Pre-Built Virtual Machine Images

We provide pre-built **virtual machine images** that work with **VMware** products (e.g. **Workstation, VMplayer, VMFusion**) or **Oracle Virtual Box**. They can also be converted to work on **Linux** hosts using **KVM** as described in accompanying documentation.

Where are the prebuilt VMs?

These VMs can be found at: http://bit.ly/LF-vm or
https://training.linuxfoundation.org/cm/VIRTUAL_MACHINE_IMAGES/
where you should log in with these credentials:

Username: `LFtraining`
Password: `Penguin2014`

The `000README` file in that directory contains deployment instructions.

All the prebuilt Virtual Machine images have been setup for common classes using the aformentioned ready-for.sh script. **However, you may still want to run ready-for.sh again on the VM for your specific course to make sure your VM guest configuration is correct**.

1.3 Using AWS

Amazon Web Services (AWS) offers a wide range of virtual machine products (instances) that can be accessed by remote users in the cloud.

In particular, you can use the **AWS Free Tier** account level for up to a year and the simulated hardware and software choices available may be all you need to perform the exercises for **Linux Foundation training courses** and gain experience with open source software. Or, they may furnish a very educational supplement to working on local hardware, and offer opportunities to easily study more than one Linux distribution.

How can I get a AWS free tier account?

You can download a guide we have prepared to help you experiment with the AWS free tier: https://training.linuxfoundation.org/cm/prep/docs/aws.pdf

2 Software Requirements

Table 2: **Software requirements for Coursera: Open Source Software Development, Linux and Git**

Internet Access	Required
Virtual Machine	Acceptable
OS required for class	Linux
Distro Architecture	x86_64
Free Disk Space in $HOME	5 GiB
Free Disk Space in /boot	128 MiB

The ready-for.sh script in a later section will automate verifying all these values for you. This script will also install any packages required for the course.

2.1 Checking Your Hardware and Software Setup with ready-for.sh

Before you continue...

Get, and run, the online tool at the following URL which will automate checking the course-specific hardware and software requirements on your computer.
https://training.linuxfoundation.org/cm/prep?course=Coursera

The Linux Foundation has provided a **bash** script which can be downloaded from the aformentioned webpage. This script is meant to be run on an installed computer to see if it is up to standards and has the necessary packaged installed and hardware for the course.

```
$ wget http://bit.ly/LFready -O ready-for.sh
```


Once you have downloaded the `ready-for.sh` script you can make it executable and run it as in:

```
$ chmod 755 ready-for.sh
$ ./ready-for.sh Coursera
$ ./ready-for.sh --install Coursera
```

Because **Linux** distributions are constantly being updated, the script is also always being updated and may not have all details filled in for all courses.

For More Information

For a more detailed explanation of all the possible methods of installation, please examine the Appendix or view it online at http://bit.ly/LFinstall or http://training.linuxfoundation.org/linux-courses/general-information-and-faq/on-site-linux-training-facility-requirements

Appendices

A More Details on Installing Linux

A.1 Installing Virtual Machine Images run under a Hypervisor

We can provide pre-built virtual machine images that work with **VMware** hypervisors, **Oracle Virtual Box**, or **KVM**. The host machine can be running any operating system with an available hypervisor, including all flavors of **Windows**, **Linux** and **Mac OS**.

Once you have the hypervisor installed, the actual installation time for a virtual machine is basically zero since all you have to do is attach our image file to it. These pre-built images already contain all the needed software and for the kernel-level courses, also conveniently contain a copy of the **Linux** kernel source git repository. The virtual machine images are updated with each new kernel release, which occurs every three months or so.

An advantage of using the virtual machine images is that you can't fundamentally destroy your system while running them, and they run as an unprivileged application and will get you into less trouble with IT staff if that is an issue. A further advantage, especially with on-line classes, is that a system failure does not take you off-line from the virtual class.

The disadvantages have mostly to do with performance and requiring somewhat more memory and CPU power. However, in most (but not all) courses this is not a disqualifying aspect.

Upon enrollment in a class we can make these virtual machine images available to you. (We do not make them available to the general public as they are quite large (2+ GB even in compressed form) and we do not have the dedicated bandwidth to support widespread downloading.)

A.2 Performing a Native Linux Installation

Virtually all popular **Linux** distributions have straightforward installation instructions these days, and most provide a **live CD** or **USB** stick which can also be used to do an install. One first boots off the Live media; a successful boot verifies that the **Linux** distribution is out-of-the-box compatible with your hardware, and you can then click on install to place the Linux distribution on your hard disk. (Using **Wubi** to install **Ubuntu** from within **Windows** does not count as a native installation). Performance is worse than using a virtual machine as discussed above and we do not support this option.

In order to proceed with installation, you generally need enough available space on the hard disk. Furthermore, free disk space may not be sufficient, as it has to be in either unallocated free space outside of any existing partition, or partitions must be available for reformatting.

This is non-trivial for most systems that have not already had multi-boot configurations setup before, and this step, which must be taken care of first, can easily be more time-consuming than the actual installation. We have seen systems which can take hours to prepare as far as the partitioning goes, but once done, installation can be performed in 20 minutes or so.

Most LiveCD/USB media contain system software to resize, move, create and delete disk partitions; most use a program called **gparted**. If you are lucky you can simply use **gparted** to shrink an already existing partition and free up 20–30 GB or so, then do your normal installation. Be careful during the procedure to properly answer any questions about your hard disk layout so you do not destroy previously existing in-use partitions.

However, many OEM-installed systems have already used four **primary** disk partitions; if this is the case you cannot create any new partitions. (You can have no more than four primary partitions, or up to three primary partitions plus an **extended** partition in which you can create a number of **logical** partitions.) On these brain-dead systems one usually finds two partitions reserved for **Windows** (a boot partition and the `C:` drive), one partition reserved for the recovery disk and one partition for manufacturer diagnostics. If you are stuck with this situation, you have to delete a partition to get your primaries down to three or do more complicated things such as converting one of the primary partitions to a logical one, and you will still have to do some steps of shrinking and moving partitions.

It is impossible for us at the **Linux Foundation** to give detailed instructions on how to do this. Each system varies as to its pre-existing layout, and the potential for turning your system into a doorstop is quite high. We do not have the technical support bandwidth to take care of things like this. Therefore, we will simply refer you to your favored distribution and its install pages for technical assistance.

Please note that very recent hardware may contain **UEFI Secure Boot** mechanisms on the motherboard. If this is enabled in the **BIOS**, the situation is more complicated and there is not a universally accepted method of making Linux co-exist with it for now. It is beyond our current ability to give technical support in this situation.

The bottom line is that unless you feel comfortable messing with your partitioning setup, have the time to deal with any potential problems, and have an available lifeline if disaster strikes, you will probably be better off doing a virtual machine installation.

As mentioned under **Installing Virtual Machine Images**, once you have the hypervisor installed, the actual installation time for a virtual machine is basically zero since all you have to do is attach our image file to it.