Welcome to Linux Foundation Virtual Training

by The Linux Foundation

05/10/2016

Version 1.8

Note:

You will receive an email containing the specific login information for your screenshare and phone bridge, as well as exact dates and times, no later than the week before the class is scheduled. We use Google Hangouts for screensharing because it is the only tool that we have found that works well with a variety of operating systems, including **Linux**, **iOS** and **Windows**.

After instructions applicable to all classes you will be pointed to specific instructions for your class in an **Appendix**.

Please examine `http://training.linuxfoundation.org/linux-courses/general-information-and-faq` and see if any remaining questions are answered by the assemblage of FAQ's in that location.

Contents

1 Hardware and Installation Requirements

Students are expected to **provide their own systems** for **Linux Foundation** classes, whether they are virtual, online classes, or in physical classrooms, whether they are open-enrollment or at an arranged on-site location. In such a case either the local IT staff will provide machines or expect students to provide their own; the Linux Foundation logistical staff should be consulted as required for assistance and clarification.

1.1 Minimum Hardware Specifications:

- x86 64-bit processor, 2 GHz Core Duo or better
- 2GB RAM minimum (if using a **Virtual Machine** at least 4 GB RAM minimum)

2 Networking, Screenshare and Audio or Telephone Requirements

2.1 Internet

Reliable broadband connectivity is required; a minimum of 100 kb/s is required. This is used for the screensharing software.

2.2 Screenshare and Audio

We use **BlueJeans** for for both the screen share and the audio.

Complete instructions for use with **Linux Foundation** classes can be found at `https://docs.google.com/document/d/1mKOuKpLIZG4c5w64nQyMwVji3DZTKdGuQAS18T40nFU`

3 Installation

You can use either a native **Linux** installation of any recent major distribution, or you can use a **virtual machine image** running under a **hypervisor**; either you can build your own or you can use one provided by the **Linux Foundation**.

Please examine http://training.linuxfoundation.org/linux-courses/general-information-and-faq/on-site-linux-training-facility-requirements?id=780. which has a more detailed explanation of all the possible methods of installation.

3.1 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.

These VM's can be found at: `http://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 and other considerations.

3.2 Developer Courses

A normal installation of any major recent distribution (such as **Red Hat Enterprise Linux, Ubuntu, SUSE Enterprise Linux, openSUSE, CentOS, Fedora, Debian,** or **Mint**) will give you almost all necessary tools, and the instructor and course manual can provide guidance on missing ingredients if necessary. All classes require **root** access (administrator or superuser) either through a **root account** or **sudo** privilege.

Please beware that we can't be responsible if your system winds up getting damaged. This warning is particularly important for kernel-level classes such as **LFD320: Kernel Internals and Debugging** and **LFD331: Developing Linux Device Drivers**, where you will be compiling and installing kernels and kernel modules.

Operating system damage, while rare, is still possible. You may wish to do a fresh installation of a 64-bit **Linux** Distribution, perhaps on a fresh partition.

Or our pre-built virtual machine images can be used for most classes, but not for all hardware intensive classes in the **LFD4xx** series. In this case memory and processor requirements tend to be more robust.

3.3 System Administration Courses

System Administration (Enterprise) courses are written for **CentOS, Ubuntu** and **OpenSUSE**. A native or virtual installation of any of the two most recent releases of these **Linux** distributions is recommended. All classes require **root** (administrator or superuser) access either through a **root account** or **sudo** privilege.

Use of virtual machines for **LFS426: Linux Performance Tuning Course** is strongly discouraged, as lab exercises will have much more meaning with a native installation of one of the three previously mentioned **Linux** distributions,

Please note there are course-specific requirements that may supersede these general requirements; please see the course-specific section in the Appendix.

3.4 Checking Your Hardware and Software Setup

Note: If you are using a Linux Foundation virtual machine, the following steps are not necessary, as they have already been run.

The **Linux Foundation** has provided a **bash** script which can be downloaded from http://training.linuxfoundation.org/cm/prep which can be run on an installed system to see if it is up to standards and has the necessary hardware for the class.

You can also run an online version of the script to check what is required for your course.

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 --install LFD411
$ ./ready-for.sh LFD331
```

(If you run the script as root, you will get warnings.)

Please note that because **Linux** distributions are constantly being updated, the script is also always being updated and may not have all details filled in for all classes.

More setup details can be found at: http://training.linuxfoundation.org/linux-courses/general-information-and-faq/on-site-linux-training-facility-requirements?id=780.

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 classes, 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) classes this is not a disqualifying aspect.

Upon enrollment in a class we can make these virtual machine images available to you. (We don't make them available to the general public as they are quite large (2+ GB even in compressed form) and we don't 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.

B Course-Specific Hardware and/or Software Requirements

Note:

Generic Developer Requirements mean the hardware requirements specified in Section 1.1, and the software requirements specified in Section 3, especially 3.2.

Generic System Administration Requirements mean the hardware requirements specified in Section 1.1, and the software requirements specified in Section 3, especially 3.3.

B.1 LFD301: Introduction to Linux, Open Source Development, and GIT

Use generic developer requirements.

B.2 LFD312: Developing Applications For Linux

Use generic developer requirements.

B.3 LFD415: Inside Android: An Introduction to Android Internals

Students must provide their own computers for this class with **Linux** running natively.

If you do not have **Linux** installed (or are unwilling/unable to install on your hardware), we recommend running **Linux** from an external SSD hard drive or large USB flash drive. Virtual machines are not supported due to slowness of cpu-intensive compiling steps, and difficulties in making external devices available through the host.

Must use 64-bit native version of Ubuntu 12.04 LTS. Later Ubuntu versions may work but are not official supported by the **Android Open Source Project**

at least 100 GB free disk space

The embedded board and associated hardware and cabling will be supplied for open enrollment classes; for on-sites procurement will usually be done by customer as agreed upon unless there are other arrangements.

B.4 LFD420: Linux Kernel Internals and Development

(also **LFD320**)

Use generic developer requirements.

B.5 LFD430: Developing Linux Device Drivers

(also **LFD331**)

Use generic developer requirements.

B.6 LFD432: Optimizing Device Drivers for Power Efficiency

Use generic developer requirements, but no virtualized environments

B.7 LFD440: Linux Kernel Debugging and Security

Use generic developer requirements.

B.8 LFD450: Embedded Linux Development

(also **LFD411**)

Students must provide their own computers for this class with **Linux** running natively.

If you do not have **Linux** installed (or are unwilling/unable to install on your hardware), we recommend running **Linux** from an external SSD hard drive or large USB flash drive. Virtual machines are not supported due to slowness of cpu-intensive compiling steps, and difficulties in making external devices available through the host.

i5 or better CPU recommended, especially for virtualized machines.

at least 40 GB free disk space

The embedded board and associated hardware and cabling will be supplied for open enrollment classes; for on-sites procurement will usually be done by customer as agreed upon unless there are other arrangements.

Ubuntu 14.04 or newer recommended. It is possible to use recent **Fedora** or **CentOS** but it may require extra time to get all tools working properly.

B.9 LFD460: Embedded Linux Development with Yocto Project

Students must provide their own computers for this class with **Linux** running natively.

If you do not have **Linux** installed (or are unwilling/unable to install on your hardware), we recommend running **Linux** from an external SSD hard drive or large USB flash drive. Virtual machines are not supported due to slowness of cpu-intensive compiling steps, and difficulties in making external devices available through the host.

It is best to consult the **Yocto Project** documentation at `http://www.yoctoproject.org` for current information. The generic statement is:

The Yocto Project team is continually verifying more and more Linux distributions with each release. In general, if you have the current release minus one of the following distributions you should have no problems.

- **Ubuntu**
- **Fedora**
- **openSUSE**
- **CentOS**
- **Debian**

i5 or better CPU recommended

at least 40 GB free disk space

The embedded board and associated hardware and cabling will be supplied for open enrollment classes; for on-sites procurement will usually be done by customer as agreed upon unless there are other arrangements.

B.10 LFS300: Fundamentals of Linux

Use generic system administration requirements.

B.11 LFS301: Linux System Administration

Use generic system administration requirements.

B.12 LFS311: Advanced Linux System Adminstration and Networking

Use generic system administration requirements.

B.13 LFS416: Linux Security

Use generic system administration requirements.

B.14 LFS422: High Availability Linux Architecture

64 bit dual-core CPU with hardware virtualization capabilities, including nested virtualization

4 GB RAM

40 GB disk space

Operating system: Any current **Linux** distribution that features **KVM**-enabled **Qemu** virtualization and support for the **libvirt** virtualization framework should be suitable for this course.

Be sure any firewalls or mandatory access control mechanisms like **AppArmor** and **SELinux** are disabled or in permissive mode.

The distributions packages for **KVM** and **Qemu** as well as **libvirt** and **virt-manager** must be installed

Hardware virtualization capabilities support must be present and enabled; this is usually a **BIOS** setting that must be switched to On or Enabled. At the kernel level, the kvm module in use needs to support nested virtualization ((`modinfo kvm_intel | grep nested`) and have it enabled on module load. Typically the `kvm_intel` module has nesting support disabled by default.

B.15 LFS426: Linux Performance Tuning

Use generic system administration requirements, plus at least 20 GB of free disk space.

Virtual machines are not recommended as many performance measurements will lack meaning.

B.16 LFS430: Linux Enterprise Automation

Use generic system administration requirements.

B.17 LFS452: OpenStack Administration Fundamentals

Students must provide their own computers for this class capable of connecting to the online lab environment. Lab systems will be assigned during class. You will need a web browser and a terminal emulation program to access them.

If using **Linux** or **Mac**, the native terminal program is fine. If using **Windows** you will need to install **PuTTY** and **PuTTYgen** from `putty.org`. The whole **PuTTY** suite is handy. Make sure any necessary firewall ports are opened for web and **SSH** traffic prior to class.

B.18 LFS462: Linux KVM Virtualization

64 bit dual-core CPU with hardware virtualization capabilities.

4 GB RAM

40 GB disk space

Operating system: any current **Linux** distribution that features **KVM**-enabled **Qemu** virtualization and support for the **libvirt** virtualization framework should work.

The distributions packages for **KVM** and **Qemu** as well as **libvirt** and **virt-manager** must be installed

Hardware virtualization capabilities support must be present and enabled; this is usually a **BIOS** setting that must be switched to On or Enabled.