FUNDAMENTALS OF OPEN SOURCE IT AND CLOUD COMPUTING (LFS200)

horizontal line

**First version: FEBRUARY 19, 2023 / 10:37 PM CT**[**https://trainingportal.linuxfoundation.org/learn/course/fundamentals-of-open-source-it-and-cloud-computing-lfs200**](https://trainingportal.linuxfoundation.org/learn/course/fundamentals-of-open-source-it-and-cloud-computing-lfs200) **00 The Operating System**

1. An operating system is "software that controls the operation of a computer and directs the processing of programs".
2. The heart of an operating system is its kernel. This is sometimes a very big program, but still a program. In fact, the word Linux technically only refers to the kernel, and not the whole operating system. The kernel has privileged features such as the ability to directly control devices connected to it. Devices such as a mouse, keyboard, screen, memory, storage, and many, many more are controlled by the kernel.
3. If the outside world would like to access these devices controlled by the Linux kernel, there must be an interface to interact with it; this interface is in fact another program, a program that can communicate with the kernel to cause some action in the hardware to occur (e.g., actions like light a bulb, read a key button being depressed, etc).
4. We need an interface between the kernel and the user program. The system call is the fundamental interface between an application and the Linux kernel. This interface between the user program and the Linux kernel is [the system call (or syscall)](https://man7.org/linux/man-pages/man2/syscalls.2.html), which is privileged code that can make requests to the Linux kernel. The kernel accepts the properly formatted request, performs the requested action and issues a return code indicating access or failure.

**01 Tasks Performed by an Operating System**

1. **Processes** - A process is any actively running instance of a program. On computers with multi-core processors, Linux can manage how to divide the program so that it runs on more than one processor at a time.
2. **Memory Management** - Linux manages the sharing of internal memory among processes. Allocating and deallocating addresses in RAM is an important aspect of memory management and will happen thousands of times per second.
3. **File Management** - Linux ensures there is sufficient available memory space to load files from disk to memory and back, enforcing the permissions stored with the data.
4. **Input/Output** - The Linux kernel handles requests to the hardware like mouse, keyboard, monitor, etc., and ensures the appropriate device driver is loaded for the device.
5. **Interpreting user commands** - When a user presses a keyboard key, touches a touchscreen or uses a mouse, the operating system will make sure that the machine responds as the user expects.
6. **Resource allocation** - The term is used to describe the combination of process and memory management to ensure that applications have enough processor time and can be allocated sufficient space in RAM as necessary.

**02 What is Linux?**

Linux is a free, open source operating system which has now grown to run on a large number of physical computer architectures. While Linux was originally based on the UNIX operating system, it was written from scratch by Linus Torvalds with assistance from a loosely-knit team of developers across the Net. It aims towards POSIX and Single UNIX Specification compliance where possible. Linux is distributed under the [GNU General Public License, version 2 (GPLv2)](https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html).

**03 What are some of the reasons to use Linux?**

1. Free
2. Stable
3. Secure
4. Maintainability
5. Runs on wide range of hardware
6. Easy to use
7. Support
8. Open Source

**04 Devices and Drivers**

Device drivers control many different types of hardware. Standardization has helped define how hardware is assembled into “main boards” or “motherboards”, where additional components can be connected. There are several “bus architectures” used to connect devices in a manner that the CPU can read, write and control the devices. Almost every bus has a selection method, an address mechanism and a data stream.

**05 Simple Bus Architecture**

A typical motherboard will have a collection of connections and buses depending on the motherboard form factor. There are decades of development and change in the bus architecture as improvements for speed and capacity continue to drive innovation. Additional reference information is included in the following short list of some of the common buses.

* 1. **PCI** - Peripheral Component Interconnect (PCI) is a local computer bus for attaching hardware devices to a computer and is part of the PCI Local Bus standard.
  2. **PCI Express** - Peripheral Component Interconnect Express (PCI Express), abbreviated as PCIe or PCI-e, is a high-speed serial computer expansion bus standard, designed to replace the older PCI, PCI-X and AGP bus standards.

**06 The Path from User Application to Hardware Access**

The Kernel is the connection between the computer hardware and end user applications. The levels of abstraction increase, the requirement for absolute control is relinquished to the Kernel and its device drivers to perform the desired actions.

**07 Connecting to Hardware (x86)**

The application program uses the operating system (OS). The OS controls the hardware, so when we use the computer, the application needs to talk to the OS. In order to manage the input and output devices, the OS needs to perform its control functions. As end-users, we interact with input devices such as typing. Typing a letter or moving the mouse and performing other functions is interpreted by the OS, the information is passed on to the application, and the application then performs the specific tasks that we want to conduct.

**08 Hardware Management**

Notice that the hardware is directly connected to the kernel, which provides services to the user. This interface is known as device drivers, rather than hardware control, since there are many devices to control. Some of the device drivers act on software "devices", which programs that react like real hardware, but are made entirely out of software.

**09 Types of Computers**

1. **Servers** - Servers serve content to other computers and client users, and are the backbone of how the modern cloud computing environment works today. Whether you are using email, messaging, streaming videos, or a web browser, your data is being served by a server. Servers are generally "workhorses" low on frills and high on compute power and/or storage.
2. **Desktops** - Desktop computers usually have monitors, keyboards and mice attached to them, as the user is typically using the machine for business applications like email, and custom applications relevant to the business (like "point of sale" or "finance" applications).
3. **Embedded** - Small purpose built computers are everywhere: cars, refrigerators, TVs and almost anything you can imagine. Many of these special purpose built systems run on highly customized versions of Linux. These are known as embedded systems.