

Lecture 1 - Introduction

Required texts

Open Sources: Voices from the Open Source Revolution, 17 authors

« <http://www.oreilly.com/openbook/opensources/book/> »

Getting Started with Ubuntu 16.04, The Ubuntu Manual Team, « <https://ubuntu-manual.org/> »

Linux Command Line, William E. Shotts, Jr., « <http://linuxcommand.org/tlcl.php> »

Dive Into Python 3, Mark Pilgrim, « <http://www.diveintopython3.net/> »

Managing Projects with GNU Make, Robert Meeklenburg,
« <http://www.oreilly.com/openbook/make3/book/index.csp> »

SCons Users Guide, Steven Wright, « <http://www.scons.org/doc/HTML/scons-user/> »

Pro Git, Scott Chacon, « <https://git-scm.com/book/en/v2> »

Installing Software, Community Ubuntu Documentation,
« <https://help.ubuntu.com/community/InstallingSoftware> »

Compiling Easy How To, Community Ubuntu Documentation,
« <https://help.ubuntu.com/community/CompilingEasyHowTo> »

FreeBSD Handbook, The FreeBSD Documentation Project,
<https://www.freebsd.org/doc/handbook/> »

What is UNIX?

Unix is considered as the mother of most of the operating systems.

The design of Unix systems is based on “Unix Philosophy” which includes the following characteristics:

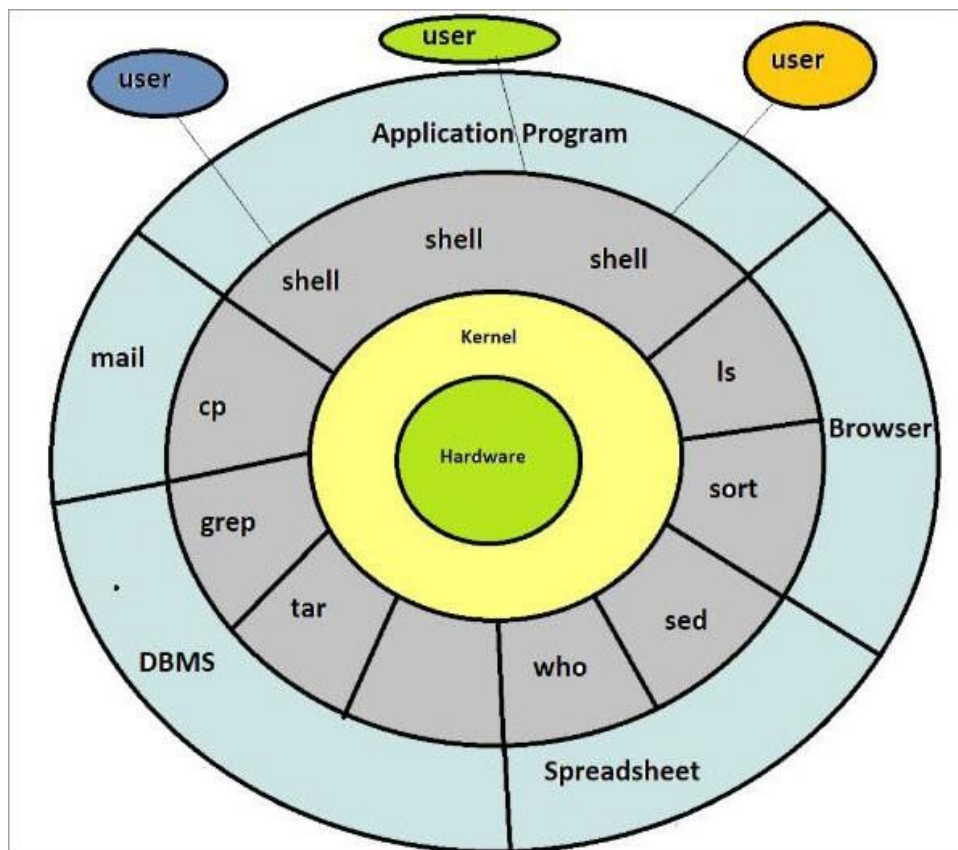
- Usage of plain text for data storage.
- Hierarchical file system.
- Handling devices and some specific kinds of inter-process communication (IPC) as files.
- Employing a huge number of software tools.
- Multiple small, simple and modular programs which can be threaded together via a command-line interpreter using pipes, contrasting to use a single monolithic program which comprises of all the same functionality.

It's worth mentioning here the below quote about Unix Philosophy:

“Although that philosophy can’t be written down in a single sentence, as its heart is the idea that the power of a system comes more from the relationships among programs than from the programs themselves. Many UNIX programs do quite trivial things in isolation, but, combined with other programs, become general and useful tools.” – Brian Kernighan & Rob Pike

Unix Architecture

The below diagram will depict you the Unix architecture.



The master control program of Unix is its Kernel. The kernel has full control over the entire system. It has subsystems that offer services to file system handling, resource handling, memory management, start & stop programs and few other low-level core tasks.

The kernel is the heart of the OS and acts as an interface between the user and hardware. Each kernel subsystem has certain features like concurrency, virtual memory, paging, and a virtual file system.

In the outer layers of the architecture, we have the shell, commands and application programs. Shell is the interface between the user and the kernel. Shell and the user types in the commands, interpret these commands and call the computer programs accordingly.

Example of the Unix operating system is Solaris and HP-UX. The largest distributors of UNIX systems include IBM, HP, and SUN.

What is Linux?

By now you would have got a fair idea about Unix. Let's now explore Linux in detail.

People do confuse a lot between the terms Unix and Linux and they generally ask questions like “**Is Unix Different from Linux?**” / “**Are Linux and Unix the same thing?**” / “**Is Linux like Unix?**” / “**Is Linux built on Unix?**”.

Here is the answer to all such questions. First, let me clear your confusion in a one-liner. Linux and Unix are different but they do have a relationship with each other as Linux is derived from Unix.

Linux is not Unix, but it is a Unix-like operating system. Linux system is derived from Unix and it is a continuation of the basis of Unix design. Linux distributions are the most famous and healthiest example of the direct Unix derivatives. BSD (Berkley Software Distribution) is also an example of a Unix derivative.

At this juncture, it is important for us to make you clear about what is Unix-like?

A Unix-like OS (also called as UN*X or *nix) is the one that works in a way similar to Unix systems, however, it is not necessary that they conform to Single UNIX Specification (SUS) or similar POSIX (Portable Operating System Interface) standard.

SUS is a standard which is required to be met for any OS to qualify for using ‘UNIX’ trademark. This trademark is granted by ‘The Open Group’.

Few Examples of currently registered UNIX systems include macOS, Solaris, and AIX. If we consider the POSIX system, then Linux can be regarded as Unix-like OS.

As per Linux kernel official README file, **Linux is a UNIX clone** that is developed from scratch by Linus Torvalds and team. It targets for POSIX compliance. The Linux kernel code was completely written from scratch. It is designed in such a way so that it acts like Unix but it does not have the original Unix code in it.

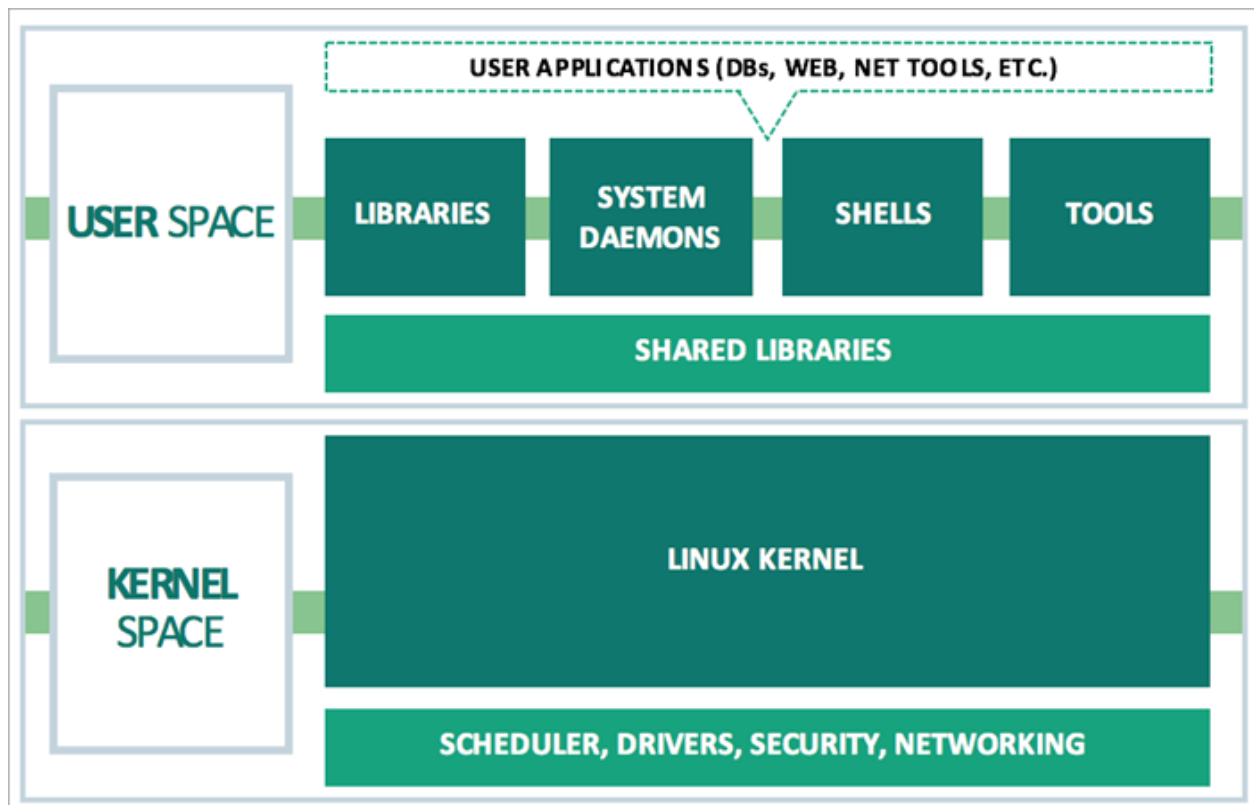
It is also significant to note that **Linux is just the kernel and not the complete OS**. This Linux kernel is generally packaged in Linux distributions which thereby makes it a complete OS. Thus, Linux is only the Kernel, while Linux distributions can be treated as the OS. On the other hand, UNIX in itself is a complete OS as everything (all required application tied together) comes from a single vendor. **For Example,** Solaris.

Linux distribution (also called as a distro in short) is an operating system that is created from a collection of software built upon the Linux Kernel and is a package management system.

A standard Linux distribution consists of a Linux kernel, GNU system, GNU utilities, libraries, compiler, additional software, documentation, a window system, window manager and a desktop environment.

Most of the software included in Linux distribution is free and open source. They may include some proprietary software like binary blobs which is essential for a few device drivers.

Linux based OS Architecture



Thus, Linux distributions actually make the Linux kernel completely usable as an operating system by adding different applications to it. There are various flavors of Linux distributions that serve a wide range of user needs.

History of Open Source

Ken Thompson and Dennis Ritchie presented the first Unix paper at the Symposium on Operating Systems Principles at Purdue University in November 1973

UC Berkeley was one of the very early adopters of Unix, and began distributing Unix tools called Berkeley Software Distribution, BSD, to other universities.

TCP/IP was added to BSD in 1981 then broken out separate from BSD later.

BSD fought lawsuits, and finally was allowed to freely distribute code in 1994. Much of what we now know as Unix comes from BSD.

IETF (Internet Engineering Task Force) – Open meetings, open standards- anyone may comment, or contribute. Standards and RFC's are freely available for downloading from IETF.

GNU GNU's Not Unix – Richard Stallman – Began with a compiler, now known as GCC (GNU C Compiler) part of every Linux OS.

Copyleft – the basic license philosophy - The central idea of copyleft is that we give everyone permission to run the program, copy the program, modify the program, and distribute modified versions--but not permission to add restrictions of their own

Anything added to or combined with a copylefted program must be such that the larger combined version is also free and copylefted.

Linux – A Unix-like kernel developed by Linus Torvalds in 1991. In 1992 it was combined with the GNU system to make a complete OS. In the years since then, several versions of Linux have evolved. Features have been added, and most are competitive with MS Windows.