

Fall 2023







Linux Essentials

Dr. Hatem Yousry



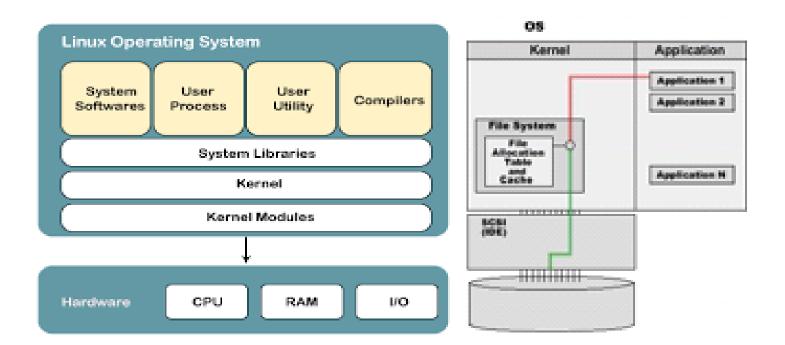
Agenda

- Introduction.
- Computers, Software and Operating Systems.
- The Most Important Operating Systems.
- Linux Distributions.

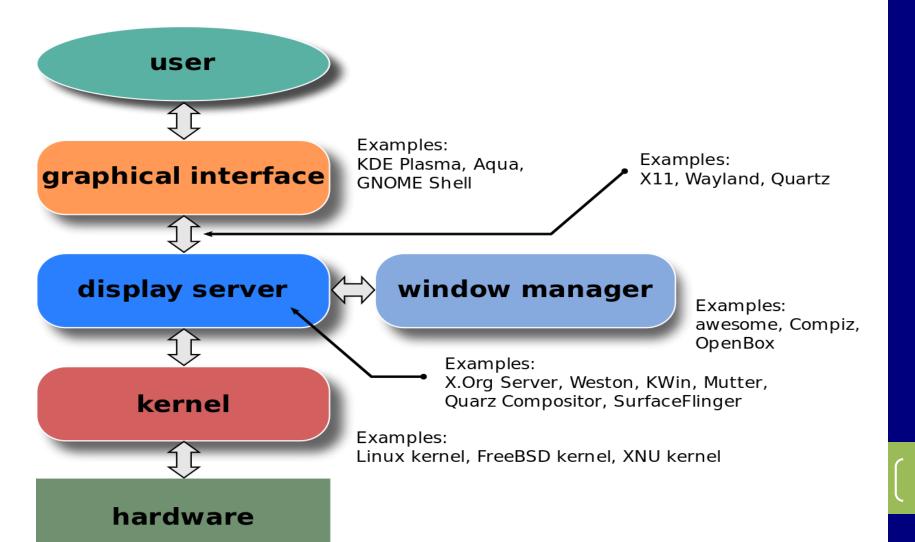


Kernel

• Kernel acts as a **bridge between applications and data processing** performed at hardware level using inter-process communication and system calls. Kernel loads first into memory when an operating system is loaded and remains into memory until operating system is shut down again.



Linux Operating System





Kernel

- There are different kernel architecture designs.
- Monolithic kernels run entirely in a single address space with the CPU executing in supervisor mode, mainly for speed.
- **Microkernels** run most but not all of their services in user space, like user processes do, mainly for resilience and modularity.
- MINIX 3 is a notable example of microkernel design. Instead, the **Linux kernel is monolithic**, although it is also modular, for it can insert and remove loadable kernel modules at runtime.

Applications

Kernel



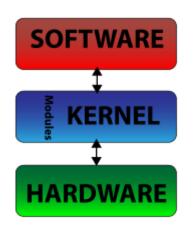
Kernel-wide design approaches

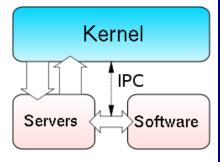
Monolithic kernels

• In a monolithic kernel, all OS services run along with the main kernel thread, thus also residing in the same memory area. This approach provides rich and powerful hardware access.



- Microkernel (also abbreviated μK or uK) is the term describing an approach to operating system design by which the functionality of the system is moved out of the traditional "kernel", into a set of "servers" that communicate through a "minimal" kernel, leaving as little as possible in "system space" and as much as possible in "user space".
- Linux is a monolithic kernel with a modular design (e.g., it can insert and remove loadable kernel modules at runtime),



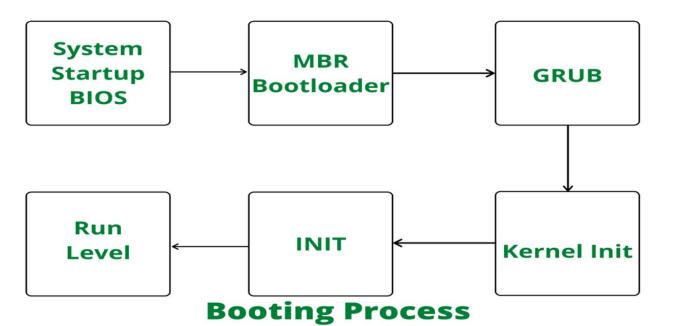




Linux kernel booting

- There are many processes are running in the background when we press the power button of the system.
 - 1. The machine's **BIOS** or boot **microcode** hundreds and runs a boot loader.
 - 2. Boot loader finds the **kernel image** on the disk and loads it into memory, to start the system.
 - 3. The kernel initializes **the devices** and their drivers.
 - 4. The kernel mounts the **basis filesystem.**
 - 5. The kernel starts a program referred to as **init** with a method ID zero
 - 6. init sets the remainder of the **system processes** in motion.
 - 7. For some purpose, init starts a method permitting you to **log in**, typically at the top or close to the top of the boot sequence.

```
1.0052271 serio: i8042 KBD port at 0x50,0x64 irg 1
    1.0060721 serio: 18042 AUX port at 0x50,0x54 irq 12
    1.007317) mousedev: PS/2 mouse device common for all mice
    1.089087] input: AT Translated Set 2 keyboard as /devices/platform/18042/se
ic0/input/input0
    1.091075) rtc_cmos rtc_cmos: registered as rtc0
    1.0919291 rtc_cmos rtc_cmos: alarms up to one day, 114 bytes myram
    1.0942061 device-mapper: uevent: version 1.0.3
    1.096061) device-mapper: ioctl: 4.39.0-loctl (2018-04-03) initialised: dm-d
vel@redhat.com
    1.0992241 Initializing XFRM netlink socket
    1.1002061 NET: Registered protocol family 17
    1.1012361 Key type das resolver registered
    1.1031661 AUX version of gcm_enc/dec engaged.
    1.104095) AES CTR mode by8 optimization enabled
    1.1322081 sched_clock: Marking stable (1132194021, 0)->(1696167804, -563973
VB33
    1.1343491 registered taskstats version 1
    1.1351101 Loading compiled-in X.509 certificates
    1.1438211 Key type encrypted registered
    1.8873771 tsc: Refined TSC clocksource calibration: 2493.720 MHz
    1.8883691 clocksource: tsc: mask: 0xffffffffffffffff max_cycles: 0x23f20436
Me, max_idle_ms: 440795277732 ms
```







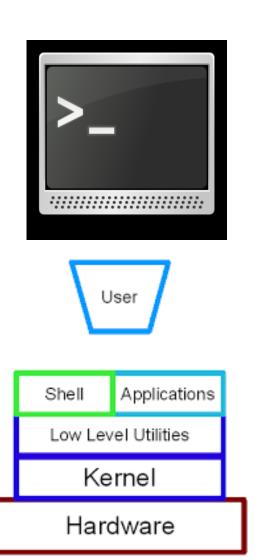
What the kernel does

- The kernel has 4 jobs:
- 1. Memory management: Keep track of how much memory is used to store what, and where
- 2. **Process management:** Determine which processes can use the central processing unit (CPU), when, and for how long
- 3. **Device drivers:** Act as mediator/interpreter between the hardware and processes
- **4. System calls and security:** Receive requests for service from the processes



Linux Operating System

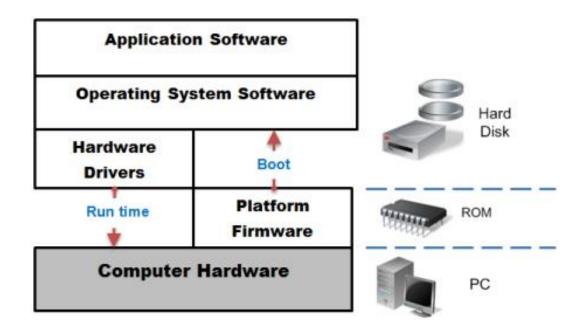
- As the Linux® kernel is the main component of a Linux operating system (OS) and is the core interface between a computer's hardware and its processes. It communicates between the two, managing resources as efficiently as possible.
- A Shell is a user interface for access to an operating system's services. Most often the user interacts with the shell using a command-line interface (CLI).
- The **terminal** is a program that opens a graphical window and lets you interact with the shell.





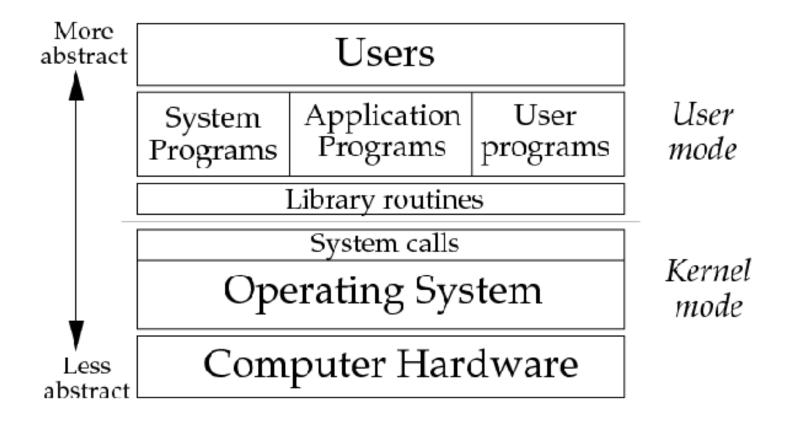
Computer Software

- "software"—the programs it is running. This can very roughly be divided into three categories:
 - The firmware.
 - The operating system.
 - User-level programs.





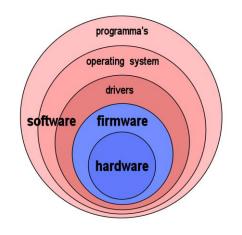
Computer Software





Firmware

- Firmware is stored on the **computer's motherboard** and can only be firmware changed or replaced inconveniently if at all. **It is used to put the computer into a defined state after switching it on.**
- Often there is a way of invoking a setup mode that allows you to set the clock and enable or disable certain properties of the motherboard.
- On PCs, the firmware is called "BIOS" (Basic Input/Output System) or, on newer systems, "EFI" (Extensible Firmware Interface).
- Some motherboards include a small Linux system that purportedly boots more quickly than Linux and which is supposed to be used to surf the Internet or watch a DVD without having to boot into Windows.







Linux kernel vs Firmware

- **Firmware** is software that typically exists on **a microchip device**. Your PC's BIOS is one example. The simple (Basic Input Output System, BIOS) program that runs every time you turn on your PC scans/communicates with your CPU, RAM, video card, hard drives, floppy drives, USB controller, etc. Since the program is stored (and run) on a chip, it's called **"firm"ware**, whereas "soft"ware is loaded from a flexible source (a hard drive) and typically run on a general purpose processor (your computer's CPU).
- A kernel is a (mid-level) piece of software that interfaces between applications and the PC's hardware. It's not as low-level as firmware (which often runs ON the hardware, itself); nor is it a high-level program like a user program (word processor, spreadsheet app, browser, etc).
- Unlike firmware, the kernel is software, as it gets loaded into memory when a PC boots from the hard drive (it's one of the first things to get loaded and executed).



Operating System

- The operating system makes the computer into a usable device: It manages operating system the computer's resources such as the RAM, the hard disks, the processing time on the CPU(s) available to individual programs, and the access to other peripherals.
- It allows **starting and stopping programs and enforces** a separation between several users of the computer. Besides, it enables—on an elementary level—the participation of the computer in a local area network or the Internet.
- The operating system frequently furnishes a **Graphical User Interface (GUI)** and thus determines how the computer "looks and feels" to its
- users.
- When you buy a new computer it is usually delivered with a preinstalled operating system: PCs with Microsoft Windows, Macs with OS X, smartphones often with Android (a Linux derivative).
- The operating system, though, is not tied as closely to a computer as the firmware, but can in many cases be replaced by a different one—for example, you can install Linux on most PCs and Macs.



User-level Programs

- User-level programs allow you to do something useful, such as write documents, draw or manipulate pictures, compose music, play games, surf the applications Internet or develop new software. Such programs are also called **Applications**.
- Additionally, there are often utilities that the operating system provides in order to allow you—or a designated "system administrator"—to make changes to the computer's configuration and so on.
- Servers, in turn, often support software that provides services to other computers, such as web, mail or database servers.

The Most Important Operating Systems

- Windows, Mac OS and Linux.
- When talking about computer operating systems, most people will automatically think of Microsoft Windows. This is due to the fact that nowadays **most PCs are sold with Windows** preinstalled—really not a bad thing in itself, since their owners can get them up and running without having to take the trouble to install an operating system first, but, on the other hand, a problem because it makes **life hard for alternative operating systems such as Linux.**



Microsoft or Apple



- The big difference between Windows and OS X is that OS X is sold exclusively with **Apple computers and will not run on "normal" PCs.** This makes it much more straightforward for Apple to provide a system that is obviously very homogenous.
- Windows, on the other hand, must **run on all sorts of PCs** and support a much wider array of hardware components that can occur in completely unforeseen combinations. Hence, Windows users have to contend with incompatibilities that are sometimes difficult or even impossible to sort out.
- On the other hand, there is a much greater selection of hardware for Windows-based computers, and prices are, on the whole, less exorbitant.
- Windows and **OS X are similar in that they are both "proprietary"** software:
- Users are **forced to accept** what Microsoft or Apple put in front of them, and they cannot examine the actual implementation of the system, let alone make changes to it. They are bound to the upgrade schedule of the system, and if the manufacturer removes something or replaces it by something else, they need to adapt to that.

Linux



- Linux is an operating system that was first started out of curiosity by Linus Torvalds, but then took on a life of its own—in the meantime, hundreds of developers (not just students and hobbyists, but also professionals at companies such as IBM, Red Hat, or Oracle) are developing it further.
- Linux was inspired by Unix, an operating system developed in the 1970s at AT&T Bell Laboratories and geared towards "small" computers.
- Unix soon became the preferred system for research and technology. For the most part, Linux uses the same concepts and basic ideas as Unix, and it is easy to get Unix software to run on Linux, but Linux itself does not contain Unix code, but is an independent project.



Linux

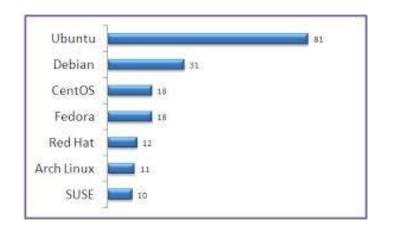


• Unlike Windows and OS X, Linux isn't backed by an **individual** company whose economic success hinges on the success of Linux. Linux is "freely available" and can be used by anyone—even commercially—who subscribes to the rules of the game (as outlined in the next chapter). This together with the fact that by now Linux no longer runs just on PCs, but in substantially identical form on platforms ranging from telephones (the most **popular smartphone operating system, Android, is a Linux offshoot**) to the largest mainframes (the ten fastest computers in the world are all running Linux) makes Linux the most versatile operating system in the history of modern computing.

NCT

Popular Linux

































Linux Distributions

- Strictly speaking "Linux" is just the operating system kernel, i. e., the program that handles the allocation of resources to applications and utilities.
- Since an **operating system without applications isn't all that useful**, one usually installs a Linux distribution, which is to say a package consisting of "Linux" proper and a selection of applications, utilities, documentation and other useful stuff. The nice thing is that, like Linux itself, most Linux distributions are **"freely available"** and hence available free of charge or at very low cost.
- This makes it possible to equip a computer with software whose equivalents for Windows or OS X would run into thousands of dollars, and you do not run the risk of falling foul of licensing restrictions just because you installed your Linux distribution on all your computers as well as Aunt Millie's and those of your buddies Susan and Bob.



More Differences And Similarities

- Actually, the three big operating **systems—Linux**, **Windows**, **and OS X**—differ only in detail in what they present to the users.
- All three offer a **Graphical User Interface (GUI)** which allows even casual users to manage their files through simple gestures like "**drag and drop**".
- Many popular applications are available for all three operating systems, so which one you are using at the end of the day becomes almost immaterial as long as you are spending most of your time inside the web browser, office package, or email program.
- This is an advantage because it enables a "gradual" migration from one system to the other.

More Differences And Similarities

- Besides the graphical interface, all three systems also offer away to use a "command line" to input textual commands which the system then executes.
- With Windows and OS X, this feature is mostly used by **system administrators**, while "normal" users tend to shun it—a question of culture. **With Linux**, on the other hand, the command line is much less ostracized, which may have to do with its descent from the **scientific/technical** Unix philosophy. As a matter of fact, **many tasks are performed more conveniently and efficiently from the command line**, especially with the powerful tools that Linux (and really also OS X) provide.
- As a budding Linux user, you do well to open up to the command line and learn about its strengths and weaknesses, just as you should learn about the strengths and weaknesses of the GUI. A combination of both will give you the greatest versatility.



Important Linux Distributions

- Linux, the operating system kernel, but a complete software environment based on Linux. This normally includes the **shell** (bash) and command-line tools from the GNU project, the X.org graphics server and a graphical desktop environment such as **KDE** or **GNOME**, productivity tools like **LibreOffice**, **Firefox** or The GIMP and lots of other useful software from the previous section.
- Of course it is possible to assemble all these tools from their original sources on the Internet, but most Linux users prefer a pre-made software collection or "Linux distribution".



GNU project



- The name "GNU" was chosen because it met a few requirements; first, it was a recursive acronym for "GNU's Not Unix".
- GNU is a Unix-like operating system. That means it is a collection of many programs: applications, libraries, developer tools, even games.
- The project to develop the GNU system is called the "GNU Project". The GNU Project was conceived in **1983** as a way of bringing back the cooperative spirit that prevailed in the computing community in earlier days—to make cooperation possible once again by removing the obstacles to cooperation imposed by the owners of proprietary software.
- https://www.gnu.org/

Terminals and Shells

- Even within a graphical Linux environment it is often convenient to access a "terminal window" where you can enter textual commands in a "shell".
- Fortunately, on most Linux desktop environments a terminal window is only a few mouse clicks away. In KDE on Debian GNU/Linux, for example, there is an entry called "Konsole (Terminal)" within the start menu under "System", which will open a convenient program running a shell that will accept and execute textual commands.





Portfolio

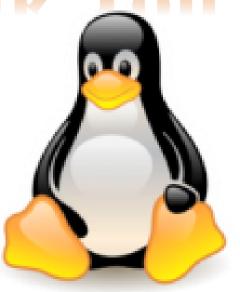


- Selective Lecture Slides.
- Your Private Notes from Lectures, Sections, Trips, Reports, Labs, and Recommendations.
- Complete answers Tasks and Assignments.
- Complete answers to Quizzes, Midterm.
- Final Project Paper.
- In addition to, any related Course martials.





Thank You



Linux

Dr. Hatem Yousry

E-mail: Hyousry@nctu.edu.eg



