

Introduction to Linux



KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

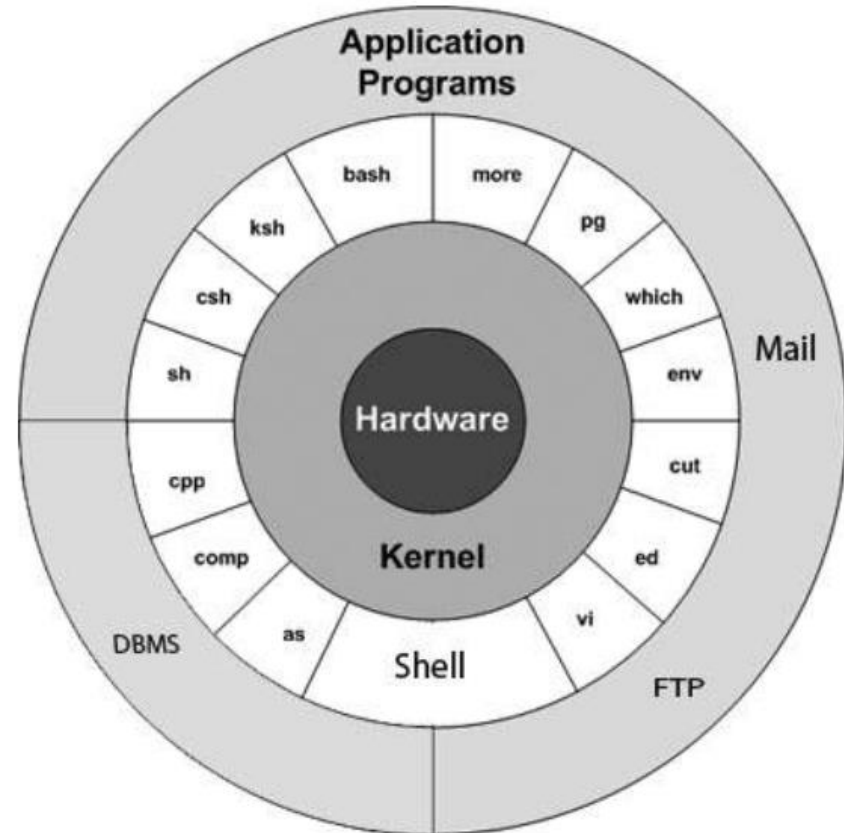
fit@hcmus

Unix

- ☐ Unix was originally developed in 1969 by **Ken Thompson, Dennis Ritchie**, ... at Bell Labs.
- ☐ **Mainly used for scientific-technical applications** on mainframes and workstations, but has become, because of Linux, also popular for classical PC-applications throughout the last years.
- ☐ Is perfectly suited for **application in networks**. Larger systems and networks require an administrator.
- ☐ Offers various alternatives for the solution of most tasks. The multitude of commands (more than in any other OS) are brief and flexible.
- ☐ Is originally **command-line oriented**, but can be used via a graphical user interface

Unix Architecture

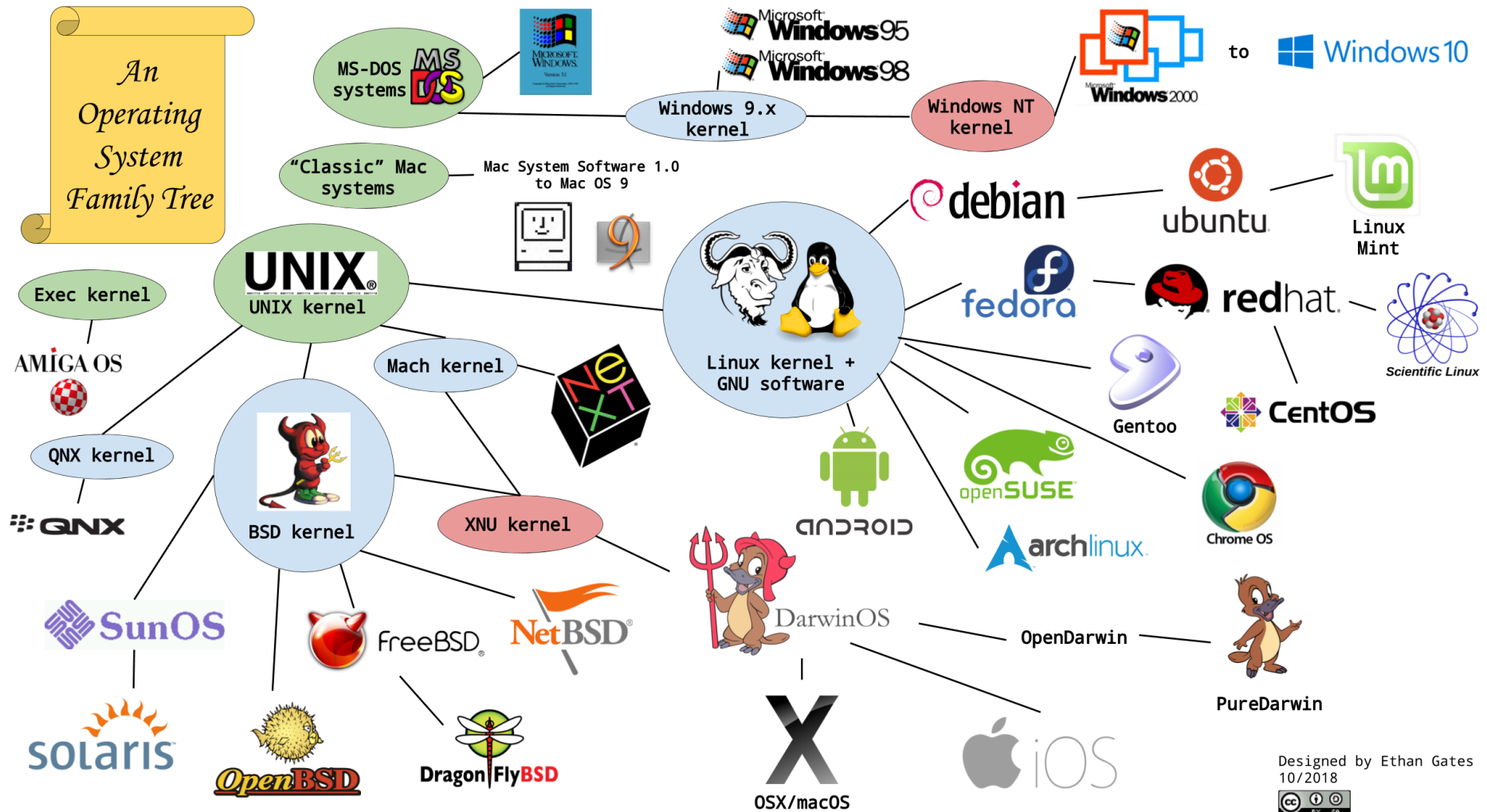
- ☐ **Kernel:** The kernel is the heart of the OS. It interacts with the hardware and most of the important tasks.
- ☐ **Shell:** The shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want.
- ☐ **Commands and Utilities:** There are various commands and utilities which you can make use
- ☐ **Files and Directories:** all the data of Unix is organized into files. All files are then organized into directories.



Linux

- ❑ **Linus** devised a ***kernel named Linux*** in 1991. (Lacking programs like File Manager, Document Editors, Audio-Video programs)
- ❑ He collaborated with other programmers in places like MIT and **applications** for Linux started to appear.
- ❑ The earlier versions of Linux OS were **not so user-friendly**
- ❑ This definitely curbed the Linux's popularity as other commercially oriented **Operating System Windows** got famous.
- ❑ Nonetheless, the **open-source aspect** of the Linux operating system made it more robust.

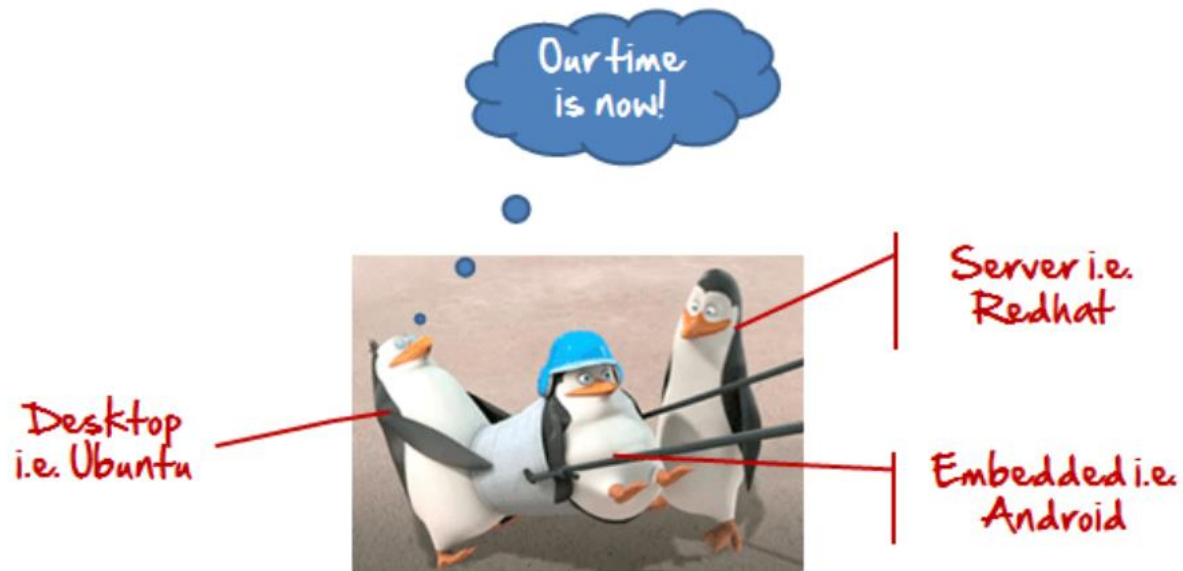
OPERATING SYSTEMS



<https://github.com/EG-tech/digipres-posters>

Linux

- ❑ The main advantage of Linux was that programmers were able to use the Linux Kernel to **design their own custom** operating systems.
- ❑ With time, a new range of **user-friendly OS's** stormed the computer world.
- ❑ Now, Linux is one of the most popular and widely used Kernel



Benefits of Linux

- ☐ Being **open-source**, anyone can modify it.
- ☐ It is **easy to learn** Linux for beginners
- ☐ The Linux operating systems now offer millions of programs/applications and Linux softwares to choose from, most of them are **free**!
- ☐ Once you have Linux installed you no longer need an antivirus! Linux is a **highly secure system**.
- ☐ Linux freeware is the OS of choice for Server environments due to its **stability and reliability** (Mega-companies like Amazon, Facebook, and Google use Linux for their Servers). A Linux based server could run non-stop without a reboot for years on end.

Why using Linux

- ☐ When working for companies in the future, it's likely that you have to interact with Linux computer, and probably have to interact with it through commands (e.g. when you use remote server)
- ☐ Or when you work with Google Colab, the only way to interact with Linux computer behind Google Colab is to use commands

What is Linux Distro

- A Linux distribution (distro) is a ***complete operating system*** that includes the *Linux kernel*, *system libraries*, *utilities*, *application software*, and a *package management system*.
- It is created by assembling various software components from different sources and packaging them together to provide a cohesive and user-friendly computing environment.

What is Linux Distro

- ❑ **Linux Kernel:** core component of the operating system, providing low-level functionality, hardware abstraction, and device drivers.
- ❑ **System Libraries:** a set of system libraries, such as the GNU C Library (glibc), which provide essential functions and interfaces for applications to interact with the underlying operating system.
- ❑ **User Interface:** Linux distributions offer different user interfaces, including graphical user interfaces (GUIs) like GNOME, KDE, or Xfce, as well as command-line interfaces (CLIs).
- ❑ **Software Packages:** Linux distributions come with a collection of software packages
- ❑ **Package Management System:** Linux distributions utilize a package management system to install, update, and remove software packages.
- ❑ **Configuration and Customization:** tools and utilities for configuring various aspects of the system
- ❑ **Support and Community:** Linux distributions are often backed by communities and support forums.

What is Linux Distro

- ❑ Popular free Linux distributions:
 - ❑ Ubuntu from Canonical
 - ❑ Debian from the Debian Project
 - ❑ Fedora from the Fedora Project
 - ❑ OpenSUSE from SUSE
 - ❑ Rocky and AlmaLinux
- ❑ Commercial distributions:
 - ❑ Oracle Linux
 - ❑ Red Hat Enterprise Linux (RHEL)
 - ❑ SUSE Enterprise Server (SLES).

Key points about linux

- ☐ Linux is an OS which is open-source, free, and widely used (especially for server computer)
- ☐ Precisely, Linux is not a full-fledged OS, but is the OS's core/kernel
- ☐ Because Linux source code is published, many groups out there have used Linux to build OS's with different flavors (a Linux-based OS with a specific flavor is call a Linux distribution or Linux distro): Debian, Ubuntu, Linux Mint, Arch, Manjaro, ...
- ☐ Why are Linux commands and Mac commands similar to each other? Because both share the same root: the ancient Unix OS (Linux gets inspired by Unix, but Linux source code is rewritten from scratch and different from Unix source code; Mac is related more directly to Unix)

Linux vs. Windows

- ❑ Linux is an **open source operating system** so user can change source code as per requirement whereas Windows OS is a commercial operating system so user doesn't have access to source code.
- ❑ Linux is very well **secure** as it is easy to detect bugs and fix whereas Windows has a huge user base, so it becomes a target of hackers to attack windows system.
- ❑ Comparing Windows file system vs Linux file system, Linux **runs faster** even with older hardware whereas Windows are slower compared to Linux.

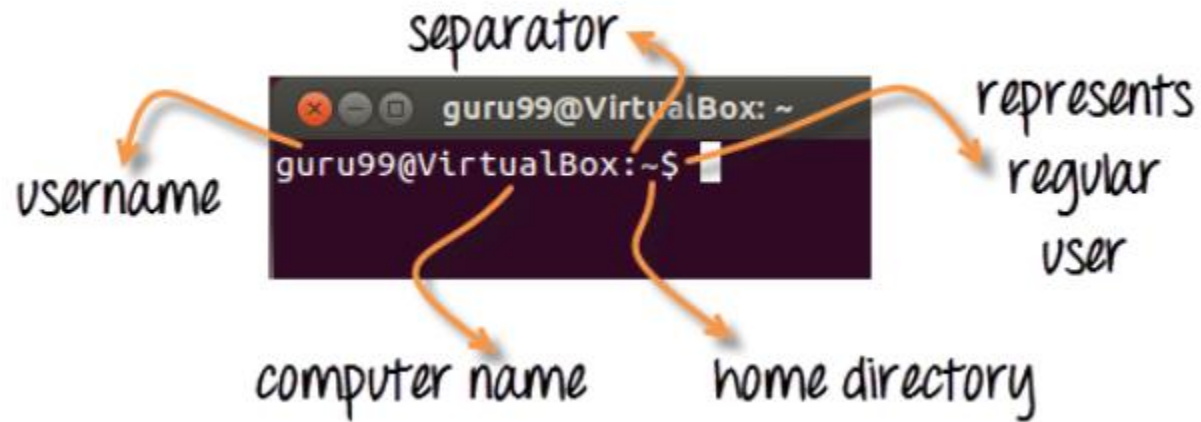
Linux vs. Windows

- ☐ Linux **peripherals** like hard drives, CD-ROMs, printers are considered files whereas Windows, hard drives, CD-ROMs, printers are considered as devices
- ☐ Linux files are ordered in a **tree structure** starting with the root directory whereas in Windows, files are stored in folders on different data drives like C: D: E:
- ☐ In Linux you would find the system and program files in different directories whereas in Windows, system and program files are usually saved in C: drive.
- ☐ Linux filename is **case-sensitive**. Windows is not

Command Line Interface

- Even though the world is moving to GUI based systems, CLI has its specific uses and is widely used in **scripting** and **server administration**.
- ▣ Comparatively, Commands offer more options & are flexible. Piping and stdin/stdout are immensely powerful are not available in GUI
- ▣ Some configurations in GUI are up to 5 screens deep while in a CLI it's just a single command
- ▣ Moving, renaming 1000's of the file in GUI will be time-consuming while in CLI, using regular expressions so can do the same task with a single command.
- ▣ CLI load fast and do not consume RAM compared to GUI. In crunch scenarios this matters.

Command Line Interface

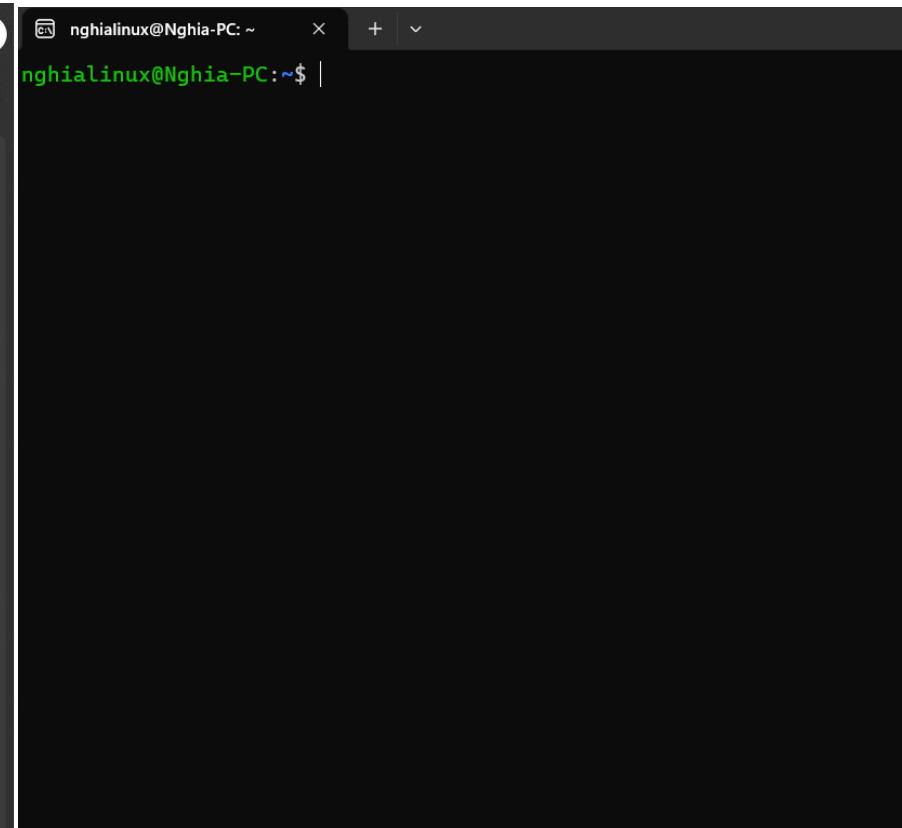
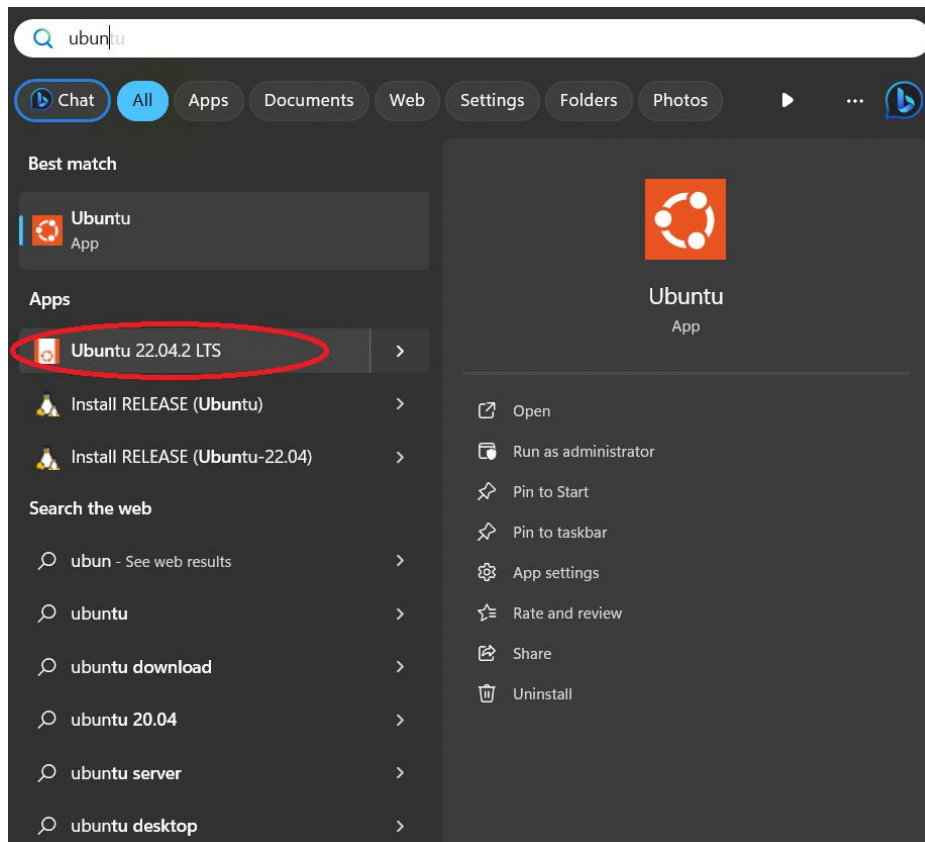


How to install Linux (on Windows)

- <https://ubuntu.com/tutorials/install-ubuntu-on-wsl2-on-windows-11-with-gui-support>
 - Install WSL
 - Download Ubuntu
 - Configure Ubuntu

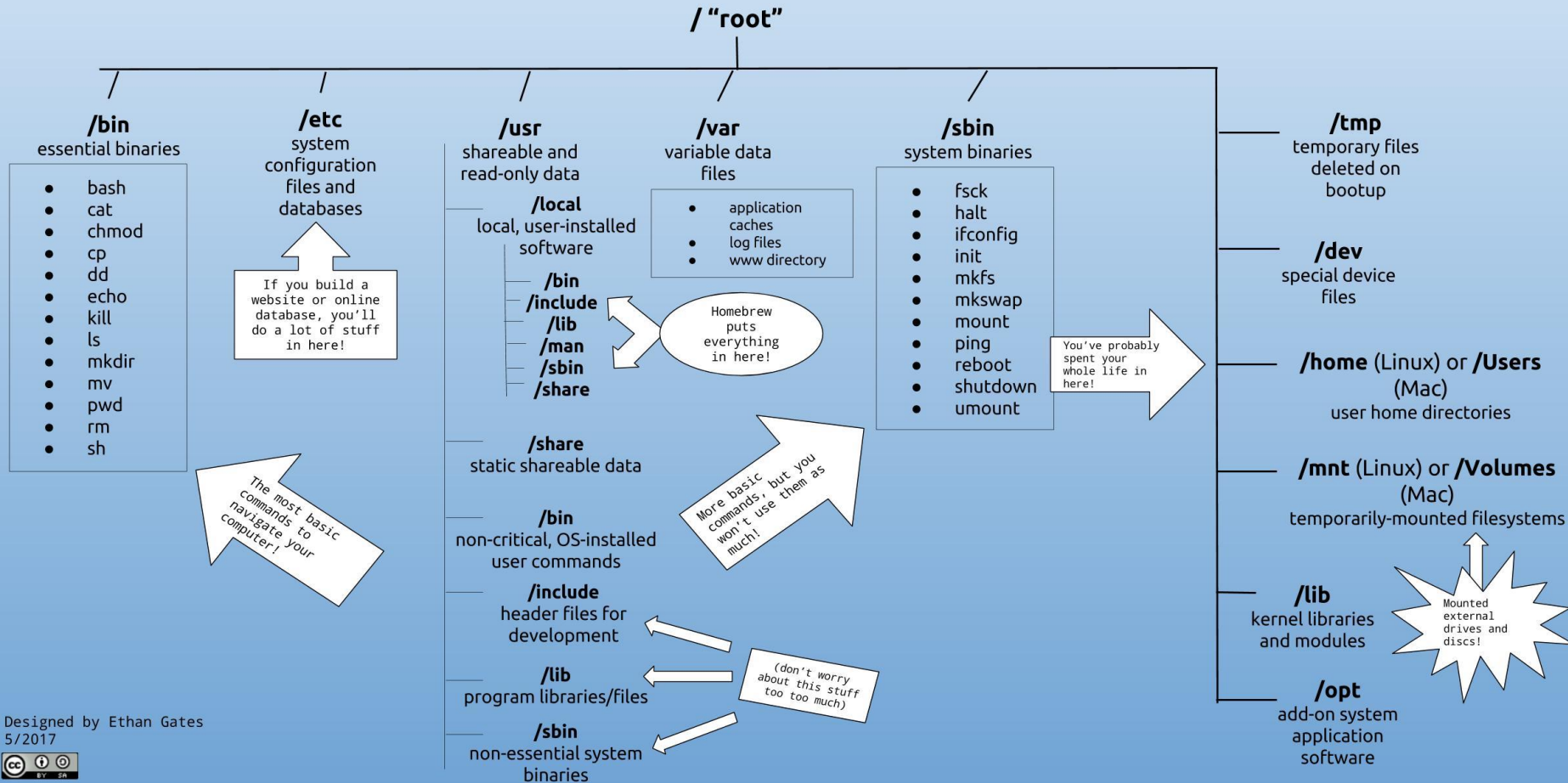
How to install Linux (on Windows)

- ❑ After install:
 - ❑ Type: “Ubuntu”
 - ❑ Choose ubuntu version



Root

What's inside your Unix computer?



File Permission

```
nghialinux@Nghia-PC:~/02-DemoDir$ ls -l
total 1044
drwxr-xr-x 4 nghialinux nghialinux    4096 Oct  1 21:37 dir1
-rw-r--r-- 1 nghialinux nghialinux     32 Sep 24 2021 file1.txt
-rw-r--r-- 1 nghialinux nghialinux     32 Sep 24 2021 file2.txt
-rw-r--r-- 1 nghialinux nghialinux 1053440 Sep 24 2021 long_file
```

- ☐ Owner permissions
- ☐ Group permissions
- ☐ Other (world) permissions
- ☐ Specific permission:
 - ☐ read (r)
 - ☐ write (w)
 - ☐ execute (x)
 - ☐ -

File Permission

```
nghialinux@Nghia-PC:~/02-DemoDir$ ls -l
total 1044
drwxr-xr-x 4 nghialinux nghialinux    4096 Oct  1 21:37 dir1
-rw-r--r-- 1 nghialinux nghialinux     32 Sep 24 2021 file1.txt
-rw-r--r-- 1 nghialinux nghialinux     32 Sep 24 2021 file2.txt
-rw-r--r-- 1 nghialinux nghialinux 1053440 Sep 24 2021 long_file
```

- ☐ First character indicates whether it is a normal file (-) or directory (d)
- ☐ Next 9 characters are permissions for the file or directory
- ☐ The next field is the number of blocks
- ☐ The next field is the owner of the file or directory
- ☐ The next field is the group the file or directory belongs to
- ☐ Following this is the file size.
- ☐ Next up is the file modification time.
- ☐ Finally we have the actual name of the file or directory.

Bash Scripting

- A Bash script: define a series of actions which the computer will then perform without us having to enter the commands ourselves.
 - ▣ If a particular task is done often, or it is repetetive, then a script can be a useful tool.
- A Bash script is interpreted
- Anything you can run on the command line you may place into a script and they will behave exactly the same.
 - ▣ Anything you can put into a script, you may run on the command line and again it will perform exactly the same.

Bash Scripting: Example

- ☐ Create a script file (DemoScript.sh)

```
#!/bin/bash
# Simple demo script
echo Here is the files in your directory:
ls -l
```

- ☐ **Beware if create file on windows**
- ☐ **First line:**
 - ☐ The very first line of a script should always be this line.
 - ☐ This line identifies which interpreter should be used.
 - ☐ The first two characters “#!” are referred to as a shebang. After that (important, no spaces) is the path to the interpreter.
- ☐ Linux is an extensionless system. “.sh” extension is is purely for convenience and is not required

Bash Scripting: Example

- ☐ Run the script

```
(base) nghialinux@Nghia-PC:~/02-DemoDir$ ls -l myscript.sh
-rw-r--r-- 1 nghialinux nghialinux 50 Oct 10 11:17 myscript.sh
(base) nghialinux@Nghia-PC:~/02-DemoDir$ chmod 755 myscript.sh
(base) nghialinux@Nghia-PC:~/02-DemoDir$ ls -l myscript.sh
-rwxr-xr-x 1 nghialinux nghialinux 50 Oct 10 11:17 myscript.sh
(base) nghialinux@Nghia-PC:~/02-DemoDir$ ./myscript.sh
Here are....
dir1  file1.txt  file2.txt  long_file  myscript.sh
(base) nghialinux@Nghia-PC:~/02-DemoDir$ |
```

- ☐ `chmod 755 myscript.sh`: set execute right.
- ☐ `./myscript.sh`: run the script
 - ☒ `./myscript.sh` we are actually telling the system to look in our current directory to find the script

Bash Scripting: Variables

- Create a script file (variableexample.sh)

```
#!/bin/bash
# A simple demonstration of variables

name='Ryan'
echo Hello $name
```

- When we set a variable, we specify it's name, followed directly by an equals sign (=) followed directly by the value.
 - ▣ No spaces on either side of the = sign.
- When we refer to a variable, we must place a dollar sign (\$) before the variable name.

Bash Scripting:

Command line arguments

- ☐ Create a script file (commandlineagvDemo.sh)

```
#!/bin/bash
# A simple demonstration of variables

echo My name is $0 and I have been given $# command line arguments
echo Here they are: $*
echo And the 2nd command line argument is $2
```

- ☐ \$0 - The name of the script.
- ☐ \$1 - \$9 - Any command line arguments given to the script. \$1 is the first argument, \$2 the second and so on.
- ☐ \$# - How many command line arguments were given to the script.
- ☐ \$* - All of the command line arguments.

A Sample Backup Script

- Create a script file (projectbackup)

```
#!/bin/bash
# Backs up a single project directory

date=`date +%F`
mkdir backups/$1_$date
cp -R $1 backups/$1_$date
echo Backup of $1 completed
```

- *For more info about bash scripting (if statement, loop, function...) can check this [Bash Scripting Tutorial](#)*

Basic command line

- <https://cheatography.com/davechild/cheat-sheets/linux-command-line/>

Simple Tutorial

- ☐ <https://missing.csail.mit.edu/2020/course-shell/>
- ☐ <https://ubuntu.com/tutorials/command-line-for-beginners>
- ☐ <https://missing.csail.mit.edu/>
- ☐ <https://ryanstutorials.net/linuxtutorial/>

Reference

- [1] The operating system Linux and programming languages: An introduction, Joachim Puls and Michael Wegner
- [2] <https://www.tutorialspoint.com/unix/>
- [3] <https://www.guru99.com/introduction-linux.html>