Linux Introduction

Chih-Hsuan Yang(SCC)

zxc25077667@pm.me

October 20, 2021

Before this talk

- ▶ Browse this slide for 15 minutes first.
- ▶ Write down the section(s) you don't know yet.
- ▶ Be attention on those sections.
- ▶ Ask questions on sli.do/fcmeqjza.
- ▶ Download this slide at Here,
  this full source code (tar archive) at Here.
- ▶ These source files are (CC BY-SA 4.0)

Outline

What is OS?

What is OS?

Time: 00:02:29 → 00:03:09 in "Revolution OS"
https://youtu.be/vWwvh3036Fw?t=149
(The next page is the text version of this segment.)

What is OS?

Linus Torvalds:

The thing about an operating system is that you are never ever supposed to serve. Because nobody really uses an operating system. People use programs on their computers and the only mission in the life of an operating system is to help those programs run. So an operating system never does anything on its' own. It's only waiting for the programs to ask for certain resources, ask for certain files on the disk or ask for the programs to connect them to the outside world. And then the operating system comes steps in and tries to make it easy for people to write programs.
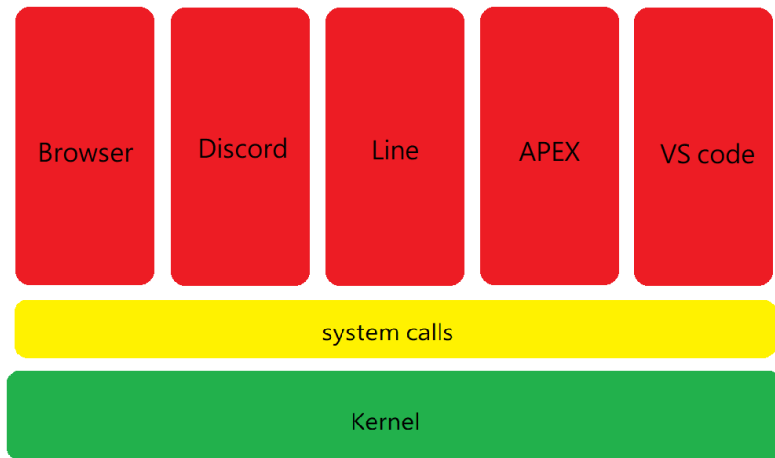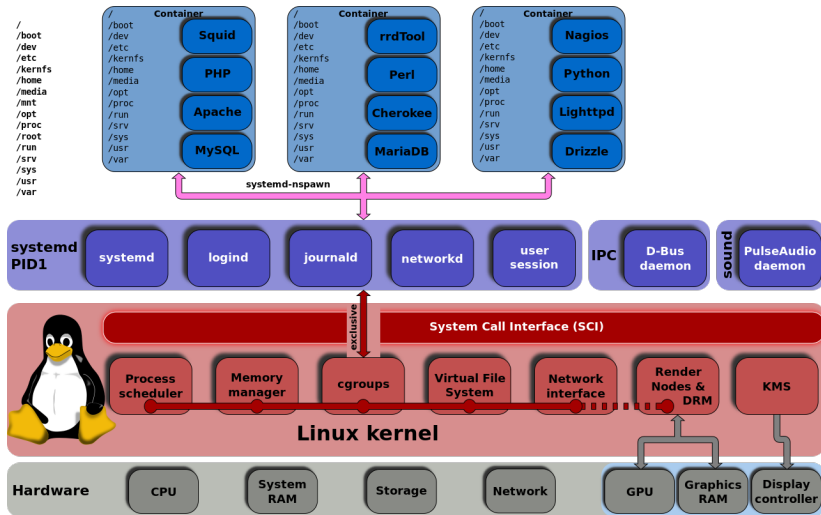
User-space programs in execution

Photo credits: Koul

Components

https://zh.wikipedia.org/wiki/Cgroups
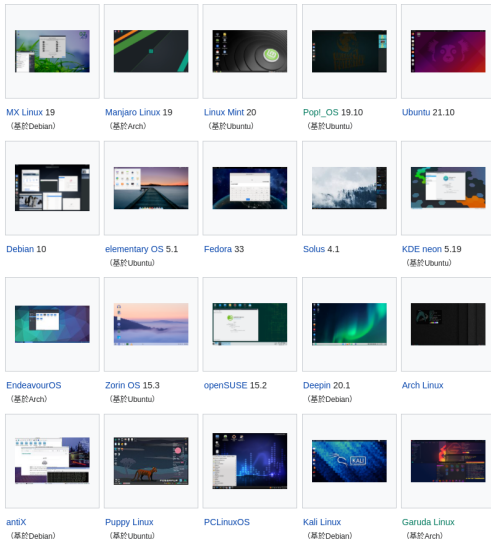
Why you should learn Linux?

1. Understand how computers work (customize)
2. Open source[1] and ubiquitous
3. Programmer, security researcher, *big/LITTLE*[2] architectures
4. You can Google it!

---

[1] https://www.gnu.org/home.zh-tw.html
[2] https://zh.wikipedia.org/wiki/Big.LITTLE

Distros.

在DistroWatch網站可以看到很多發行版的點擊率和信息，其中關注度位居前列的發行版展示如下：

MX Linux 19
（基於Debian）

Manjaro Linux 19
（基於Arch）

Linux Mint 20
（基於Ubuntu）

Popl_OS 19.10
（基於Ubuntu）

Ubuntu 21.10

Debian 10

elementary OS 5.1
（基於Ubuntu）

Fedora 33

Solus 4.1

KDE neon 5.19
（基於Ubuntu）

EndeavourOS
（基於Arch）

Zorin OS 15.3
（基於Ubuntu）

openSUSE 15.2

Deepin 20.1
（基於Debian）

Arch Linux

antiX
（基於Debian）

Puppy Linux
（基於Ubuntu）

PCLinuxOS

Kali Linux
（基於Debian）

Garuda Linux
（基於Arch）

https://zh.wikipedia.org/wiki/Linux%E5%8F%91%E8%A1%8C%E7%89%88

Linux distribution

Traverse it: Wiki
and https://distrowatch.com/

Booting and history

Traverse it: IBM boot
Netscape history: Wiki
Revolution OS: YouTube
Traverse it: Hurd
Read parts of initrd.

Permissions

File Permissions

$ ls -l

stat: https://linux.die.net/man/1/stat

File Permissions

Traverse it:
https://linuxjourney.com/lesson/file-permissions

Don't 777

ALL[3]
Think about what it means!

Process permissions (capabilities)

Process is a program in execution.

File permission extending: access control and capabilities
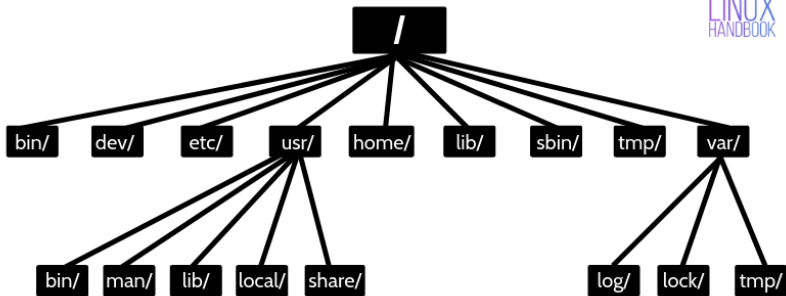
Linux Capabilities intro, (zh_cn)

capabilities since Linux 2.2 (1999/1/25)[4]:

Man capabilities

Search the 2.2.0

---

[4]https://zh.wikipedia.org/wiki/Linux%E5%86%85%E6%A0%B8

File systems

src: https://linuxhandbook.com/linux-directory-structure/

Tree

$ tree -L 1 /; # You can man it!
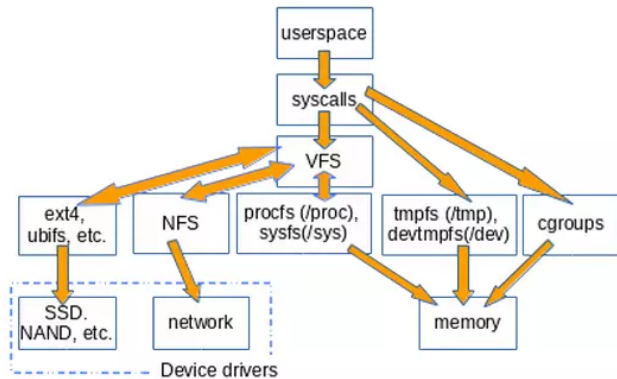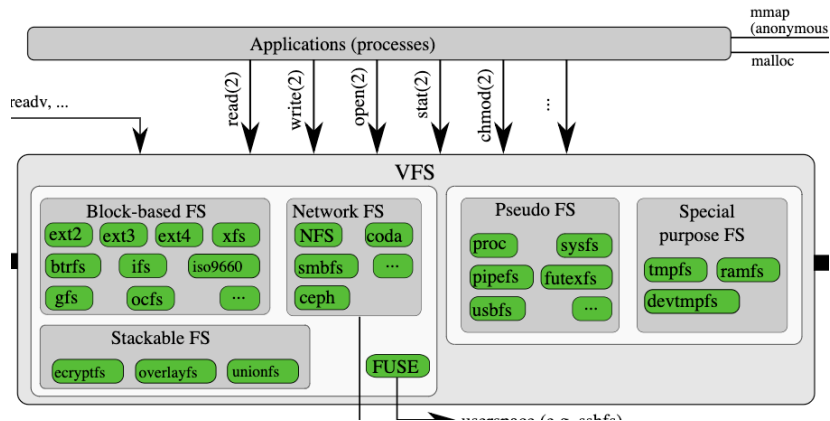
Mount devices

The mount must mount 'device' on the directory.
Traverse it: Mount tutorial

VFS in kernel

VFS in kernel

Everything is a file descriptor

Read parts of Universal-IO

> *Rather than have a variety of device emulation mechanisms (for network, block, and other drivers) , virtio provides a common front end for these device emulations to standardize the interface and increase the reuse of code across the platforms.*

Small is beautiful

The Art of UNIX Programming

Rule of Modularity: Write simple parts connected by clean interfaces.
Rule of Clarity: Clarity is better than cleverness.
Rule of Composition: Design programs to be connected with other programs.
Rule of Separation: Separate policy from mechanism; separate interfaces from engines.
Rule of Simplicity: Design for simplicity; add complexity only where you must.
Rule of Parsimony: Write a big program only when it is clear by demonstration that nothing else will do.
Rule of Transparency: Design for visibility to make inspection and debugging easier.
Rule of Robustness: Robustness is the child of transparency and simplicity.
Rule of Representation: Fold knowledge into data, so program logic can be stupid and robust.
Rule of Least Surprise: In interface design, always do the least surprising thing.
Rule of Silence: When a program has nothing surprising to say, it should say nothing.
Rule of Repair: Repair what you can — but when you must fail, fail noisily and as soon as possible.
Rule of Economy: Programmer time is expensive; conserve it in preference to machine time.
Rule of Generation: Avoid hand-hacking; write programs to write programs when you can.
Rule of Optimization: Prototype before polishing. Get it working before you optimize it.
Rule of Diversity: Distrust all claims for one true way.
Rule of Extensibility: Design for the future, because it will be here sooner than you think.

http://www.catb.org/~esr/
writings/taoup/html/
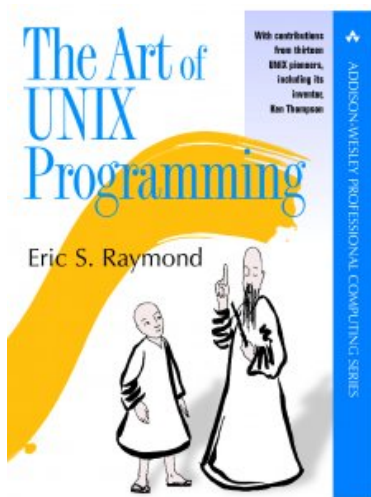
The 17 Rules of Eric Raymond

- Build modular programs
- Write readable programs
- Use composition
- Separate mechanisms from policy
- Write simple programs
- Write small programs
- Write transparent programs
- Write robust programs
- Make data complicated when required, not the program
- Build on potential users' expected knowledge
- Avoid unnecessary output
- Write programs which fail in a way that is easy to diagnose
- Value developer time over machine time
- Write abstract programs that generate code instead of writing code by hand
- Prototype software before polishing it
- Write flexible and open programs
- Make the program and protocols extensible.

Less is more than more

man page of less
Master Foo and the Ten Thousand Lines

Go!

Package manager
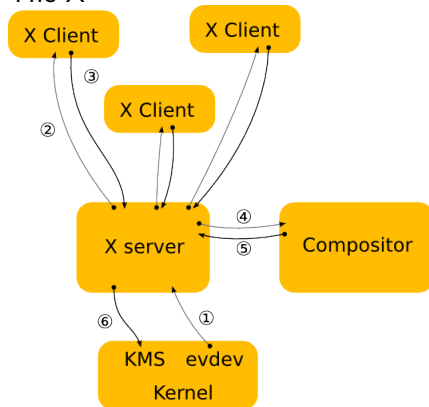
Best Linux Package Managers

- ▶ DPKG
- ▶ RPM
- ▶ Pacman (AUR)
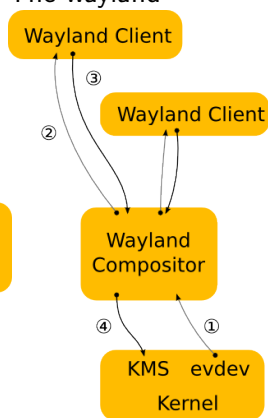
Use Docker image to demo.

GUI Concepts and DE

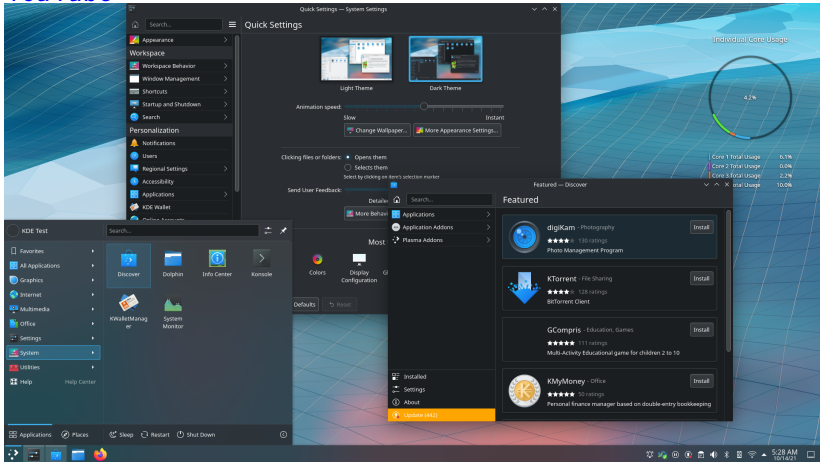## Take WSL as an example

The X
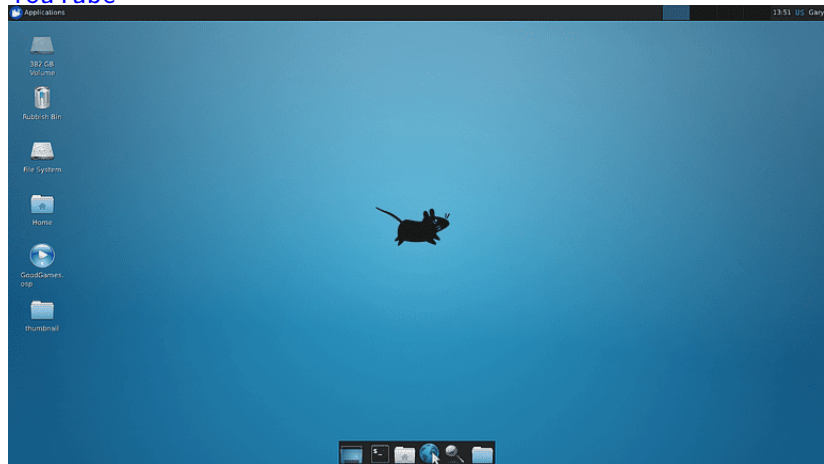

The wayland

Wiki

GNOME

YouTube: Plugins
Youtube: GNOME 40

KDE

XFCE
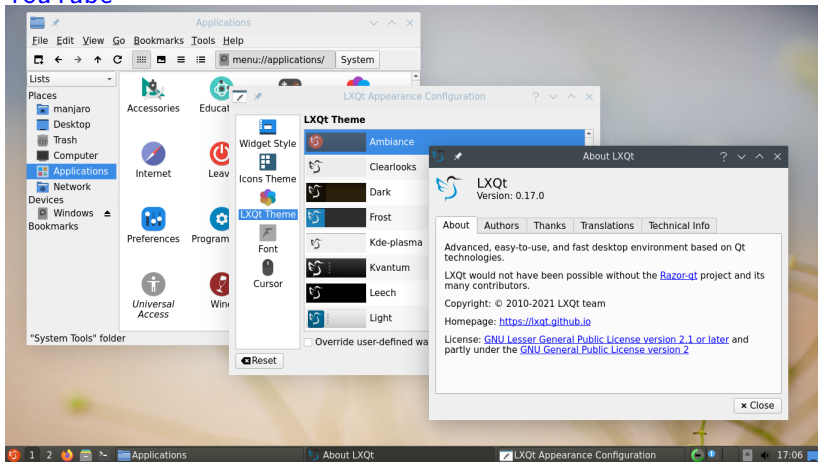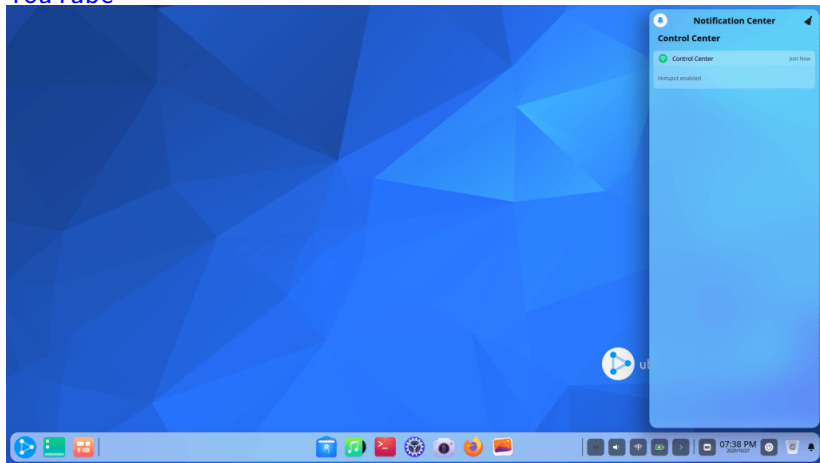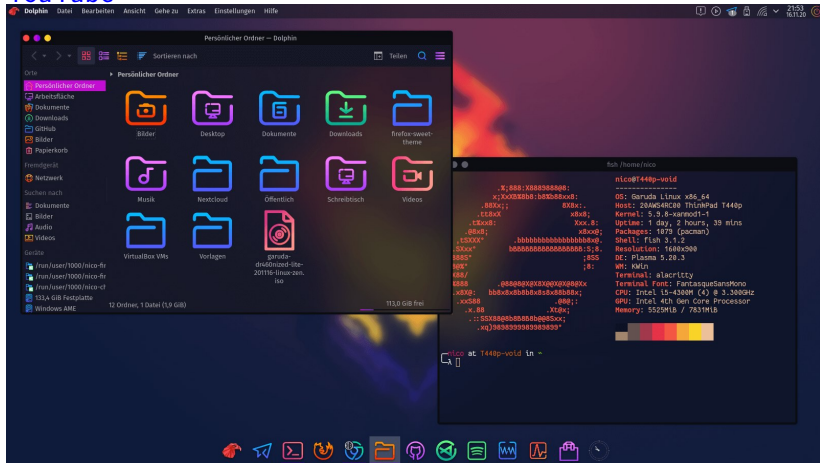
YouTube

LXDM/LXQT

Deepin: DDE

YouTube

Garuda KDE Dr460nized

Command Line Interface

Above features are added strength

In my opinion: there are 3 main utilities of CLI

- ▶ Scripting (faster)
- ▶ Logging
- ▶ GUI is not essential

Scripting

Try:

- ▶ Create a folder named 'a' for 100 times recursively.
- ▶ Find where are this 'word' in this directory.
- ▶ Poke me to make the counter overflow($> 2^{16} - 1$).
  You can win a secret price if you do it! (Show me.)

Logging

We are developers. We need those information to solve the errors.

GUI is not essential

"A wifi access point does not need a desktop environment."
change my mind.

How to learn commands?

Click me

How to learn commands *faster*?

Flash your entire device and install a Linux distro.