

Content-Type: text/x-zim-wiki Wiki-Format: zim 0.4 Creation-Date: 2018-04-21T19:56:17+02:00

Welcome On Linux

Created samedi 21 avril 2018

@catch-phrase The linux startup guide

@brief Mathieu CAROFF's general guide to learning IT and linux.

@author Loxaan OXYDE — loxaxs Mathieu CAROFF — mathieu4f

@first-publication @last-update 2018-04-21

Titre Sommaire

@note

- Four pass:
 - First pass: Introduction (theory)
 - * Nature
 - * Principle(s)
 - * Problem solved
 - * Elements
 - * Possible actions
 - Second pass: Complementray (theory)
 - * Related concepts
 - * Related bugs and problems
 - * How to identify these bugs
 - * How to solve these bugs
 - Third pass: Exercises
 - * Manipulating the elements
 - * Running the possible actions
 - * Doing usefull things with the concept
 - Threeth pass: Links content
 - * (To get more precise informations)

0) Concepts and history

0) History and Genealogy of operating systems

Multics (1964, MIT)

- MS-DOS (Disk Operating System, **MS**)
 - MS-DOS (1981)
 - MSX-DOS (1984)

- Windows (82%)
 - * 1.0 (x16-x32) (1985)
 - 3.0 - 3.2 (1990-1994)
 - 95, 98
 - ME (2000)
 - * Windows NT (x32-x64)
 - XP (2001)
 - Vista (2006)
 - 7 (2009)
 - 8, 8.1, **10** (2012-)
- Unix (kernel) (portage — .tar.gz) (1973, Nokia **Bell Labs**)
 - Mac OS (12%) [Steve Jobs]
 - * OS X
 - BSD (1977, University of California **Berkeley**)
 - * FreeBSD
 - Linux (kernel) (1.5%) (1991, Linux Torvald, Community)
- Linux (kernel) (1.5%) (1991, Linux Torvald, Community)
 - Debian (1993, Ian Murdock, Community) (apt — deb)
 - * Debian — Server @@@
 - * Ubuntu
 - **Ubuntu** (2005) — Standard, Community @@
 - Linux Mint — Welcoming @@
 - Xubuntu — Light & flexibility @@
 - Lubuntu — Light @@
 - Kubuntu — Heavy @@
 - * Knoppix — Toolbox @@@@
 - RedHat (yum — rpm)
 - * RedHatEnterprise — Customer service, Guarantees
 - * CentOS — Security
 - * Fedora — Security @@@
 - Gentoo (2002) (emerge — .tar.gz) — Customisability, Understandability @@@@
 - Arch Linux (2002) (pacman) — Pedagogic @@@@
 - * Manjaro — Fast @@@
 - Slackware (txz)
 - * Slackware (rpm)
 - * SUSE Linux
 - OpenSUSE
 - —
 - Android (play store ?)
 - Chrome OS (1.0%)
- Android (75%)
- iOS (20%)
- Windows CE (0.58%)
- Firefox OS (0.45%)
- Unknown (2.9%)

Notation: @@ — Estimation of the usage difficulty

Element of glossary:

Distribution: Any operating system whose kernel is Linux.

External resources:

https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg

Learn more:

<https://distrowatch.com/>

Sources: (2018-03-27) — Informations:

<https://en.wikipedia.org/wiki/Multics>

https://en.wikipedia.org/wiki/Microsoft_Windows

<https://en.wikipedia.org/wiki/Linux>

https://en.wikipedia.org/wiki/List_of_operating_systems

https://en.wikipedia.org/wiki/*

— Figures:

<http://gs.statcounter.com/os-market-share/desktop/worldwide/>

0) Opensource Licenses

Note:

- Licenses apply only if you start **distributing** the code. As long as it remains private, there is no restriction

Weak copyleft (non-viral, corporation friendly)

- The Unlicense (~CC0 / Public domain)
- MIT
- Apache
- GNU LGPL (libraries)

Strong copyleft (viral)

- GNU LGPL (v2, v3)
- GNU GPL (v2, v3)
- GNU AGPLv3

For non-software creations:

- CC0
- CC BY(-NC)?(-SA|-ND|)? (v[1234].0)

Learn more:

<https://choosealicense.com/>

<https://creativecommons.org/licenses/>

0) History of Editors

ed vi & emacs vim & emacs

0) History of shells

Windows:

- cmd
 - powershell

Unix:

- csh: **C shell** (1978)
 - tcsh: Tenex C shell (1983)
- sh: **Born shell** (Unix 7) (1979)
 - ash: Almquist shell (1989, on BSD distributions)

Unix, POSIX-compliant:

- dash: Debian Almquist shell (1997, 2002)
 - bash: Bourn Again shell (4x slower than dash) (1989)
 - * zsh: Z shell (1990) Unix:
- fish: Friendly Interactive Shell (2005)

https://en.wikipedia.org/wiki/C_shell

https://en.wikipedia.org/wiki/Bourne_shell

https://en.wikipedia.org/wiki/Almquist_shell

<https://wiki.archlinux.org/index.php/Dash>

https://fr.wikipedia.org/wiki/Debian_Almquist_shell

[https://en.wikipedia.org/wiki/Bash_\(Unix_shell\)](https://en.wikipedia.org/wiki/Bash_(Unix_shell))

0) The second part

0) Computer components

Case

- Alimentation unit

- Motherboard
 - //Chipset//
 - CPU(s)
 - * Radiator
 - //Fan//
 - Graphical Card(s)
 - * //Graphical RAM//
 - * //GPU//
 - * //Radiator / Fan//
 - RAM
 - Hard Drive (HDD or SSD)

Notation: //Italic - //Usually already included in the host component

Element of glossary: Chip: A long rectangular piece of deep-fried potato. Chip (microchip): Small embeddable integrated circuit. Wafer: A thin, light, crisp biscuit, especially one of a kind eaten with ice cream. Wafer: A thin piece of something.

Sources: <https://en.wikipedia.org/wiki/Chipset> <https://www.dictionary.com>

@comment - I let you read this page. [6s] - Do you have questions ?

0) Booting

From you pressing the power button to your desktop appearing before your eyes. (Linux-specific ~ Ubuntu-specific)

- You press the power button
- A component of your motherboard send the RESET signal to the CPU
- The CPU loads the BIOS (Basic Input/Output System) from the ROM
- **BIOS**
 - The BIOS search for a disk with a program to handle the rest.
 - It finds and **loads the boot loader**.
- **Boot loader**
 - Some boot loader names:
 - * Grub (GRUB2, Ubuntu's default)
 - * Syslinux (ISOLINUX, 2005-2010 on CD)
 - * Windows Boot Manager (BOOTMGR)
 - BCD — Boot Configuration Data)
 - The boot loader allows you to select which OS / Linux distribution you want to boot.
 - The boot loader **loads the Kernel**.
 - The boot loader exits, starting the kernel.
- Kernel
 - The kernel is in charge of managing
 - * CPU

- * Memory
- * Devices
- It automatically starts the first process
- First process
 - Some names:
 - * Upstart (Ubuntu until 2012)
 - * Systemd (Ubuntu since 2012, Archlinux since 2002)

Learn more:

http://www.linuxdevcenter.com/pub/a/linux/excerpts/linux_kernel/how_computer_boots.html

https://en.wikipedia.org/wiki/Comparison_of_boot_loaders

Sources:

<https://en.wikipedia.org/wiki/BIOS>

0) The very general approach to using new piece of software

0) Learn how to quit it, how to disable it and how to avoid it getting in the way
1) Learn how to get help about it and learn how to get its version
 2) Learn to use the basic functions corresponding to a tool of this kind
 * You may use a polyglotte / rosetta / correspondance table page about this ; or some other tutorial.
 2) Learn how you can make using the tool equivalent to using some other tool you already know.
 * You may need to guess and search (keywords)
 3) Learn about gotchas & other traps of the tool (if any)
 * For instance, search some blog entry about it's gotcha and how to avoid them. Read tutorial about how to get started with the tool.
 4) Learn to use the specific features of the tool, what makes them interesting
 5) Learn to use more specific, detailed, advanced features of the tool.

About how to quit ^C, ^D, q, :q, ^X (Trick) ^Z, then job kill %1
About how to get help: ** -help man **google: documentation** info (exception for the bash builtin: help)

0) The system

(Linux-specific ~ Ubuntu-specific)

- Process
 - Child process
 - Process state
 - Return code
 - Dead parent
- Hostname, distribution version and kernel version
- The systeme controler
 - System V

- Service
- The disks
 - The partition tables (MSDOS, GPT)
 - The partitions themselves
 - * Their format
 - * Their flags
 - The swap partition
 - Mounting a partition
 - * `sudo mount`
 - * `udiskctl mount`
 - * `systemd mount` (unsure)
 - The fstab file
- Time, time zone and locales
- The package manager
- GRUB
- Directory architecture bin
boot cdrom core dev elevs etc home initrd.img initrd.img.old lib lib64
lost+found media mnt opt proc root run sbin snap srv swapfile sys tmp
usr var
- System commands (...)

Sources:

https://wiki.archlinux.org/index.php/installation_guide

0) Graphical Server, Desktop Environnemnt,

0) Window Manager, Terminal emulators,

0) Terminal multiplexors, VTs, TTYs and Shells

VTs

<http://blog.hawkhost.com/2010/06/28/tmux-the-terminal-multiplexer/>

<https://wiki.archlinux.org/index.php/Tmux>

0) Permissions (& Privileges)

0) Processes

0) Environnement

0) Learning Bash & Fish

Quiting ^D Getting help man info ** -help -

Builtins: true false echo cd pwd help which history

Daily commands:

cat more less - grep sed cut awk - find locate whereis - tea - awk bc - python -

Less used:

tac rev -

System

ls-* lsblk lsb_release

Developing

make gcc g++ gdb -

0) Learning to use command line tools

less grep sed cut find - bc awk

0) Learning to use other command line tools

apt dpkg git

0) Learning to use synaptic, apt-* & dpkg

@dep History and Genealogy of operating systems

<https://wiki.archlinux.org/index.php/Pacman/Rosetta>

0) Programming languages characterisations

- Procedural languages
 - Typages
 - * Static vs Dynamic
 - * Manifest (Explicit) vs Inferred (Implicit)
 - * Strong vs Weak (Conversion implicitness)
 - * Strong vs Weak (Type safety)

- * Strong vs Weak (Difficulty to solve type-based bugs)
- Paradigmes
 - * Imperative
 - * Functional
 - * Object-oriented
- Backend / Frontend
- Scripting languages
- Web languages
- 0-indexed / 1-indexed
- External characteristics
 - * Popularity
 - * Community
 - Opensource
 - * Package management tool
- Description languages
- Markdown languages

Source:

https://en.wikipedia.org/wiki/Type_system

0) Existing programming languages

History of languages Relative importance per language today

0) Programming practices

- Code formatting
 - Style
 - * Compactness
 - Code beautifier
- Practices
 - Variable naming conventions
 - Not mutating object (Persistence)
- Principles
 - Correctness before optimisation
 - Using the languages natural, integrated way of doing things.
 - Testable code only
- Project Management
- Philosophy

0) Regular expressions

Regular expressions are useful tool to parse strings, find patterns, ensure strings respect a given format, replace a complex pattern in a string by another, plus

a few more usage I forgot. They are available in all programming languages and used in a lot of tools, including text editors and command line tools. Learn about them on <https://www.regular-expressions.info/>.

0) Simple Text editor VS Poweruser Text Editor VS Integrated Development Environnement

You should switch to VSCode or Atom or SublimeText

0) Version Control System

History of version control systems Sophistication level of version control systems
* Highest: * Git (git) * Mercurial (hg) * Fossil (fossil)

0) Alternativeto

Find replacement for any of your current software using this wonderful tool:
<https://alternativeto.net/>

0) Other programmer knowleges

- Programming jokes:
 - xkcd.com
 - FB page (ask Youen)
- Learning, asking:
 - Quora.com
 - *.stackoverflow.com
 - * Stackoverflow.com
 - * Askubuntu.com
 - [KXL]?Ubuntu | Debian — System, Desktop environnement, Installation, Drivers
 - * Unix & Linux (Unix.stackoverflow.com)
 - Shell, linux kernel, linux configuration, linux tool
 - * Superuser.com
 - Windows admin / Linux root
 - * Serverfault.com
 - Server programming, configuration
 - Reddit.com
 - * r/linux