Linux OS

Tuur Vanhoutte

February 17, 2021

Contents

1 Introductie

1.1 Verschil Server & Workstation

1.1.1 Server

- Deliver services to (multiple) users
- Focussed: only this and nothing else
- Secure
- No GUI, everything happens through the commandline
- $\Rightarrow$ as small a footprint as possible

1.1.2 Workstation

- Use services
- Create documents
- Look for information
- Consume multimedia
- GUI
- $\Rightarrow$ Large footprint

1.2 Extra information/resources

- The Linux Documentation Project: `http://tldp.org`
- Pluralsight LPIC-1: Linux Professional Institute Certification: `https://www.pluralsight.com/paths/lpic-1`
- The Arch Linux Wiki is one of the most extensive sources of info about Linux: `https://wiki.archlinux.org`
  - In this module we will use Debian, not Arch, but many things are very similar
- Google

1.3 What is Linux?

1.3.1 What is an operating system (OS)?

Definitie 1.1 (Operating System) *An operating system, or OS, is software that communicates with the hardware and alows other programs to run.*

It is comprised of system software = the fundamental files your computer needs to function.

Linux is NOT an operating system: Linux = the **kernel**

1.3.2 What is a Kernel?

Definitie 1.2 (Kernel) *The kernel is software that is the core of a computer's operating system, with complete control over the system.*

It is the first program loaded on start-up.

It handles. . . :

- *. . . the rest of the startup*
- *. . . input/output requests from software, translating them into instructions for the CPU*
- *. . . memory*
- *. . . peripherals*

1.4 GNU Operating System

Definitie 1.3 (GNU) *GNU = GNU's Not Unix (recursive algorithm)*

Founded by Richard Stallman (ex-MIT, founder of the Free Software Foundation), 1984

Goal: completely free Operating System

1.5 Linux, the kernel

By Linus Torvalds (Finland), 1991

- Own personal development, not initially intended to distribute
- Interest from other developers, mainly to use with GNU OS
- Meanwhile contributions of over 12000+ developers
- 492 of top-500 supercomputers in the world run Linux
- Basis for Android, Chrome OS

Linux = the kernel

GNU = OS-tools around the kernel

$\Rightarrow$ **GNU/Linux**

1.5.1 Distributions

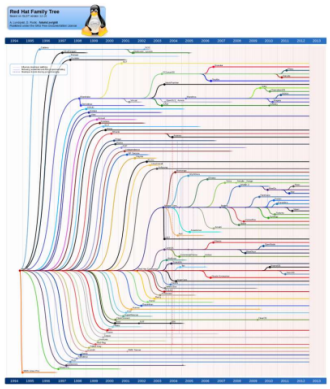
Definitie 1.4 (Distribution) *A Linux distribution (or distro for short) is GNU/Linux + extra tools and applications to create a full-fledged OS.*

That distribution can be easily copied and installed to other computers.

- RedHat (CentOS)
- Debian (Ubuntu)
- Arch Linux
- Void Linux
- Gentoo
- Pop! OS

Red Hat family tree

Debian family tree



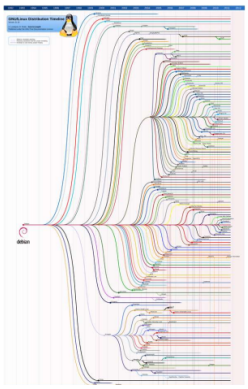


Figure 1: https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg

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

1.6 Open Source

Definitie 1.5 (Open Source) *Open source software is software of which the code is licensed to be open to everyone.*

Anyone can use, change, distribute the software. This allows code to be developed in a public manner.

OPEN SOURCE DOES NOT MEAN FREE

1.6.1 Commercial distributions

= Open source, non-free distributions

- SUSE Linux Enterprise Server (SLES)
- SUSE Linux Enterprise Desktop (SLED)
- Red Hat Enterprise Linux (RHEL)
- Oracle Enterprise Linux

Commercial distributions have official support channels.

⇒ You're not paying for the operating system, you're paying for the support.

1.6.2 In this course: Debian

- Current version: 10.7
- Forms the basis of many others: Ubuntu, Raspbian, Knoppix, Linux Mint
- Available on many platforms: Intel x86, AMD64, Intel64, ARM, MIPS, Power Systems, ...

2 Debian Installation

See Labs for detailed Installation tutorial

2.1 Networking in Linux (with VMWare)

- VMWare presents ethernet adapter
- During creation of virtual machine: MAC-address is created
- During installation: network configuration through DHCP
  - IPv4-address
  - Default gateway
  - DNS-server
  - Optional: proxy-server

2.2 Users in Linux

- Linux is multi-user from the ground up
  - Multiple users can be active at the same time
- 'Administrator'-user is called root
- Each user has a user-ID (uid)
  - root has uid=0
  - uid=0 has all rights
- Each user has a home-directory

2.3 Disks, partition, filesystems

- Our VM has 1 disk
  - Presented on the SCSI-bus
  - First disk on SCSI-bus: **sda**
  - Then sdb, sdc, ...
- Disk = concatenation of blocks
- Divide blocks in collections (=partitions)
  - 1st partition: sda1
  - 2nd partition: sda2
  - ...
- 2 types of partitions
  - Primary
  - Extended

2.3.1 Partitions

Primary partition

- A filesystem can be created inside this
- Up to 4 primary partitions

Extended Partition

- 'Logical' partitions can be created inside this

Our setup:

- sda1: primary partition
- sda2: extended partition
- sda5: 'logical' partition inside extended partition sda2

Figure 2: Our setup

2.4 MBR <> GPT

2.4.1 MBR

We use the MBR Partitioning scheme

Definitie 2.1 (MBR) *MBR, or Master Boot Record, is a special type of boot sector at the start of a disk.*

It contains:

- *a set of instructions necessary to boot operating systems.*
- *info about how partitions are placed on disk*

Limitations:

- Maximum disks of 2TB
- 32-bit for number of logical sectors
- Common sector size: 512 bytes
- $2^{32} \cdot 512$ bytes $= 4294967296 \cdot 512$ bytes $\approx$ 2TB

BIOS can boot from a disk with MBR partitioning

2.4.2 GPT

Definitie 2.2 (GPT) *GPT, or GUID Partition Table, is a standard for the layout of partition tables on a disk. It's an alternative to MBR.*

It uses unique identifiers (GUIDs)

- BIOS cannot boot from a disk with GPT-partitioning: UEFI required when using GPT
- GPT allows disks larger than 2TB

Definitie 2.3 (UEFI) *UEFI, or Unified Extensible Firmware Interface, is a newer firmware interface by Intel (90's) that replaces the BIOS interface by IBM (70's).*

How does it work?

- Disk = collection of blocks
- Group of blocks together = sector
- Common sector size: 512 bytes
- Sectors indicated with Logical Block Addresses (LBA)
- MBR in LBA 0
- GPT headers in LBA 1
- Partition tabel right after that

2.4.3 Bootstrap procedure

1. Motherboard gets electricity
2. Mini-loader hardcoded in memory
    - BIOS gets loaded
3. Boot media are consulted
4. First boot medium, first sectors are being read $\Rightarrow$
5. MBR contains a bit-more-advanced loader: GRUB
    - GRand Unified Bootloader
6. This loader loads a more advanced loader (GRUB second stage bootloader)
7. The OS is loaded

2.4.4 Linux boot process

6 high level steps

- BIOS (Basic Input/Output System) - loads MBR
- MBR (Master Boot Record) - loads GRUB
- GRUB (Grand Unified Bootloader) - loads kernel
- Kernel - executes /sbin/init
- Init - executes runlevel programs
- Runlevel - programs from /etc/rc.d/rcXX.d are started

2.4.5 BIOS <> UEFI

- Recent systems use UEFI, not BIOS
- UEFI is required to boot from GPT-disk
- Linux has no trouble working with UEFI

So why will we use MBR?

- Virtualisation is the norm
- Virtual machines typically have small disks
- Small disks are MBR partitioned

2.5 Filesystems

2.5.1 Windows

- FAT (1977)
- FAT32 (1996)
- NTFS (1993)
- ReFS (2012)

2.5.2 Linux

- Ext (1992)
- Ext2 (1993)
- Ext3 (2001)
- Ext4 (2008)
- ZFS (2005)
- BtrFS (2007)

2.5.3 Swap

= Paging

- Free up physical memory (RAM) by moving pages to slower storage (storage disks instead of RAM)
- Page out = memory page moves to swap
- "Swapiness"
  - = parameter between 0 and 100
  - = how quickly linux will swap
    - 0 = very conservative
    - 100 = very agressive
- Windows uses a swap file (pagefile.sys)
- Linux uses a swap partition

2.6 File structure

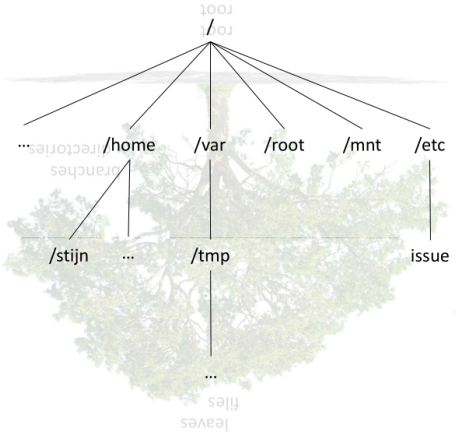


Figure 3: Linux uses a tree structure

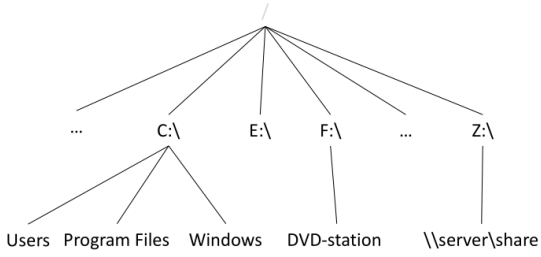


Figure 4: Windows uses a similar structure, but every volume uses a letter.

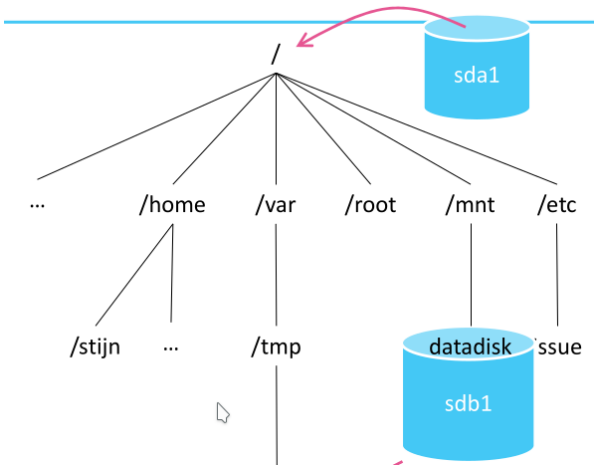


Figure 5: With linux, volumes are 'mounted' to folders somewhere under root /

2.7 Configuration

2.7.1 Packages

- Tools and applications are build up by files
- All files belonging to 1 application are bundled in a package
- Packages in debian have the .deb extension

Repositories

- Packages are collected in repositories
- Are made available through the internet
- Packages have dependencies

2.7.2 Package management

Debian: dpkg & apt (Advanced Package Tool)

- dpkg: Install, remove, give info about .deb packages
  - dpkg -l = lists packages
- apt: Get packages from a repository and install, remove, give info, ...
  - apt update
    - Contact the repositories
    - Get most recent list of packages and versions
  - apt upgrade
    - Of the packages which are more recent in the repositories compared to what is installed: install newest version
  - apt install <xyz>
    - Download package <xyz> from the repository

    * Check the dependencies and download depending packages
    * Install package <xyz> and all corresponding dependencies

Which repositories? See /etc/apt/sources.list for the list of repositories. You can add/remove/change repositories in this file.

2.7.3 Useful packages

- open-vm-tools
- vim
- sudo
- tcpdump

Install multiple pacakges in one command: apt install vim sudo tcpdump ntp

2.8 Shutdown of VM

- Power button (=ACPI shutdown)
- Shut down operating system only
  - = halt
- Shut down operating system and VM, multiple ways:
  - shutdown -P now
  - init 0
  - poweroff
- Reboot
  - reboot
  - init 6
  - shutdown -r now

2.9 Basic network

- No GUI $\Rightarrow$
- Layer 1: Physical (VMWare virtual network)
- Layer 2: Datalink (Ethernet & MAC address)
- Layer 3: Network (IPv4)
- Layer 4: Transport (Transport Control Protocol (TCP), User Datagram Protocol (UDP))
- Layer 5: Application (SSH, HTTP, . . .)

2.9.1 Basic networking commands

- arp
- ping
- route
- bmon

2.10 Services

- Processes that ‘listen’ on the network
  - TCP or UDP port
- Overview of currently running / listening services: ss command
  - ss -tulpn
  - t: show TCP
  - u: show UDP
  - l: show listening
  - p: show process ID
  - n: no name-resolving

2.11 Wooclap Questions

- Why do we talk about GNU/Linux?
- What is a kernel?
- What is the difference between Open Source and free?
- How is the Administrator user called? What is its uid?
- What is MBR?
- What are the limitations of MBR? (Solution?)
- What is swap? What is swappiness?
- What is a package?
- What is a repository?
- What is a dependency?
- What is a package manager?
- What is the difference between ’apt update’ and ’apt upgrade’
- Which protocol makes the link between MAC address & IP address?
- Which command gives you the current ARP-table?
- What are the 5 layers of the TCP/IP network model?
- How do you find the MAC-address of a network interface?
- Put Linux boot process in correct order (6 levels)

- What is a linux distribution?

3 File structure

- Tree structure
  - Leaves = files
  - Branches = directories
  - The tree is inverted, root = /
- Everything is a file (even devices, random numbers, and RAM) under 1 root
- This is in contrast to Windows, where every volume is a root.

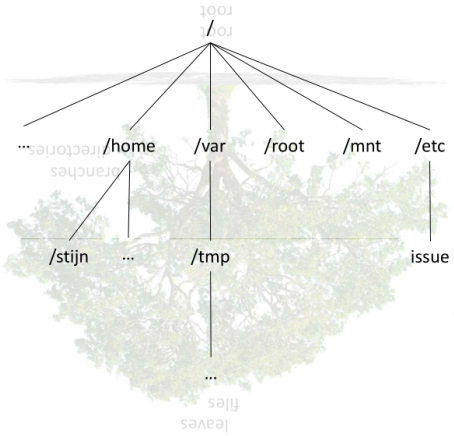


Figure 6

3.1 Intermezzo: single user mode

- Linux (the kernel) is built up as a multi-user system from the beginning
- Standard behaviour = multi-user
- But: also possible to boot in single-user mode
  - No daemons, no multiple logins
  - Sometimes called **Maintenance mode**
- Examples of usage
  - Filesystem repairs
  - Upgrade of distribution
  - Password recovery
  - Adjustments to the root filesystem
  - Forensics after security incident

3.1.1 Runlevels

= predefined operating system status

- Is presented with a number
- Linux has 7 runlevels:
  - 0 = system halt (= VM shutdown)
  - 1 = single user
  - 2 = multi-user, no NFS (no network services, not often used)
  - 3 = multi-user, CLI (Command Line Interface)
  - 4 = self-definable
  - 5 = multi-user, GUI (Graphical User Interface, if installed)
  - 6 = reboot

3.2 Intermezzo: Add disk

Add a new disk without shutting down the system

1. Adjust VM: add disk
2. Detect added disk
3. Partition disk
   - fdisk (for MBR)
   - parted (for GPT)
4. Create filesystem
   - Partition = collection of blocks (sectors)
   - Not usable for the OS ⇒ create filesystem
   - mkfs.ext4 /dev/sdb1

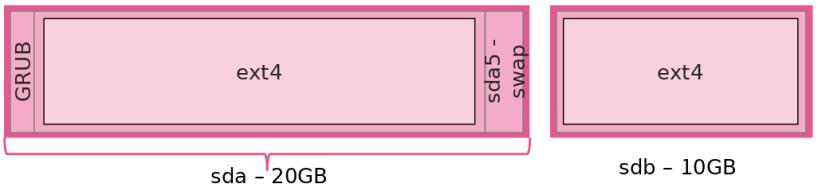


Figure 7

5. Mount filesystem
   - mkdir /mnt/datadisk
   - mount /dev/sdb1 /mnt/datadisk
   - see if it worked: df -h

For detailed steps: see labs!

3.2.1 What after a reboot?

Use /etc/fstab = a file that contains what needs to be mounted at boot

- Device (/dev/sdXY or UUID)
- Mountpoint (/mnt/folder)
- Type of filesystem (ext4, ntfs, . . .)
- Options

3.3 Navigate through the tree

- pwd
  - Print working directory
  - Shows where in the tree you are
- ls
  - Show a list of files in the working directory
  - ls -la : 10 characters at the beginning of each line. The d == directory (see later)
- When you login, you are in your home directory
- / (= the filesystem root) is not the same as /root (the home directory of the root user)
- . = current directory
- .. = the directory one higher

3.3.1 Relative vs absolute path

Relative paths:

- cd .. = go to the directory above the current directory
- cat ../etc/issue = go to the etc directory, one directory above the current directory. Open the issue file

Absolute paths:

- cd / = go to the root directory
- cat /etc/issue = go to the etc/ directory under / (root)

3.4 Filesystem Hierarchy Standard (FHS)

- Describes how the filesystem in Linux is build up
- Maintained by the Linux Foundation
- Most recent version: v3.0 (2015)

3.4.1 Rules in the standard

- / is the root of the tree structure
- /bin
  - essential binaries (executable files), required for single user mode
- /boot
  - the place on the filesystem where the boot files reside
  - configuration files for GRUB
  - kernels
  - initrd
    - initial RamDisk
    - During boot a temporary root-filesystem is being created in RAM
    - This is used so the kernel can load important modules, so it can then switch to the real root filesystem
    - part of step 3 of the linux boot process (BIOS - MBR - GRUB - kernel - init - runlevel)
- /dev
  - Devices get a place in the filesystem
    - sda
    - rtc
    - random
    - cpu
    - urandom
    - null
  - ls -lah /dev/
- /etc
  - Host-specific sytem-wide configuration files
  - Configuration for this host, readable for the whole system
- /home
  - Each (non-system) user has a home directory
  - except for root $\Rightarrow$ /root
- /mnt
  - (temporarily) 'mounted' filesystems
    - Network shares
    - USB-disk
    - DVD-ROM
    - Extra disks

  - Some distributions use /media for this

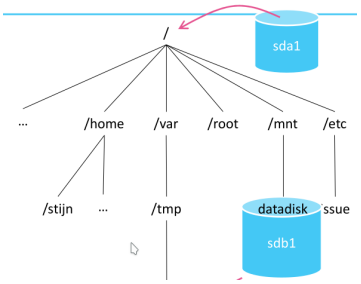


Figure 8

- /opt
  - Optional application software packages
  - Our installation ⇒ no applications installed yet = empty (for now)
- /proc
  - Virtual filesystem
  - Provides information about processes and the linux kernel
  - cat /proc/cpuinfo
  - cat /proc/sys/net/ipv4/ip_forward
  - cat /proc/partitions
- /sbin
  - Essential system binaries
  - Only executable by root user
  - fsck, init, route
- /tmp
  - Directory for temporary files
  - Emptied at reboot (with most distributions)
- /usr
  - Read-only user data
  - Constains most user (non-root) utilities and applications
- /var
  - Variable files
  - Files that are expected to change continuously during normal system use
  - Logs, spool files, temporary e-mail files, . . .

3.5 Some useful tips

3.5.1 History

```
~# history

# shows a list of former commands executed by this user
# spans log-in sessions
# in reality, it shows the contents of the ~/.bash_history file
# if you use another shell like zsh, it's the ~/.zsh_history file
```

CTRL + r:

- Search the command history
- Show commands that match what you're typing
- repeatedly press ctrl+r to scroll through results

3.5.2 Bind mount

Situation

- /mnt/storage is the normal mountpoint for other filesystems (e.g. SAN)
- Filesystem could not be mounted, but a process already started writing data
- ⇒ this data arrives on the / filesystem under the directory /mnt/storage
- Problem fixed and filesystem can be mounted again ⇒ mounted under /mnt/storage
- ⇒ the already written data is now hidden

The solution

- Create /mnt/storage and put some data in it
- Create a 1GB disk, ext4 formatted, mount under /mnt/storage ⇒ data is now hidden
- Use mount -o bind to get data back without unmounting

3.5.3 dd

= Command to read or write bytes

```
# Example: overwrite first 2048 bytes of a disk with zeros
~# dd if=/dev/zero of=/dev/sdb count=4 bs=512
# Example: overwrite disk with random data when taking out of service
~# dd if=/dev/random of=/dev/sdb bs=1M
```

4