Introduction

Marek Kozłowski

Faculty of Mathematics and Information Sciences
Warsaw University of Technology

Outline of the Lecture

Outline of the lecture

BIOS MBR vs UEFI GPT

- Modern GNU/Linux distros and BSD systems can use either BIOS or UEFI.
- Enabled UEFI Secure Boot however can result in strange booting problems for non-Windows systems and therefore disabling it should be considered whenever possible.
- For security, stability, simplicity and compatibility reasons numerous *nix administrators prefer using UEFI in legacy (BIOS mode).
- The next slide describes legacy BIOS MBR solution for better clarity but it's worth mentioning that key concepts from our point of view are the same as for UEFI GPT systems.

Boot Manager

- First HDD sector (512B) is called a *Master Boot Record* (MBR).
- MBR contains a *boot loader* and a *partition table*.
- Boot loader is a small program that invokes an OS loader (*stage2* loader) from a respective partition.
- Multiple OSes may be installed into disk partitions.
- Boot manager is a boot loader that allows choosing an OS to be initialized.
- Windows used to assume that no other OSes are installed.
- BSD boot managers (*boot0*) offer selecting a boot partition and have no further configuration.
- Modern Linux distributions typically come with the *GNU GRUB* boot manager.

GRUB Boot Manager

- GRUB1 (0.97, legacy):
  - follows the KISS (*Keep It Simple, Stupid!*) principle,
  - requires manual installation,
  - single configuration file with simple syntax,
  - limited to drives below 2TB,
  - may not support most recent versions of the *ext4* filesystem,
  - doesn't support UEFI.
- GRUB2:
  - supports UEFI, drives over 2TB and all *ext4* versions,
  - configuration file generated automagically based on numerous input files and *os-prober* software,
  - configuration file should NOT be edited by hand.

Booting Unix

Unix boot-up sequence:

1. loading, uncompressing and initializing a *kernel*,
2. detecting devices and loading drivers,
3. initializing a process tree,
4. connecting mass storage (*mounting file systems*),
5. starting *daemons* (services); typically confirmed by some kind of [OK] messages.

After the boot-up sequence has finished users may log in via a tty or a display manager.

Splash Screen

- Optional software that displays animations instead of boot-up messages.
- Some Linux distributions, especially those intended for inexperienced users may start it by default.
- Can be uninstalled or disabled.
- Usually there are key combinations (try pressing `<F2>` or `<Ctrl><F2>`) that allow switching to other views.

Outline of the lecture

Display Manager

- Optional, OS-independent piece of software.
- Typically started as a daemon/service.
- Initializes graphical mode.
- Displays login dialog (and possibly some options).
- Starts a desktop environment according to user's choices.
- Popular DM software:
  - xdm,
  - gdm,
  - kdm,
  - lightdm (used in our labs),
  - slim.

Teletyping Terminal (tty)

- Unix has been designed as a *multiuser OS*.
- Separation of processing and user interface is a key concept.
- A tty is an input/output hardware typically consisting of:
  - keyboard/mouse for text stream input,
  - monitor for text stream output.
- A tty can be:
  - physically connected (directly or via a KVM switch),
  - emulated by *terminal emulator* applications running locally in a GUI mode,
  - connected remotely over a TCP/IP network (*ssh*, *telnet*).

Linux and *BSD TTYs

- Physically connected keyboard/mouse and monitor can be used for first 12 ttys.
- Users can switch between them by pressing `<Alt><Fx>` (function keys).
- Due to possible GUI key re-mappings in graphical mode use `<Ctrl><Alt><Fx>` sequence.
- Usually first 4-6 ttys run *getty* software that allows login.
- Switching to the first one unassociated (try tty7 or tty8) typically returns to a GUI mode.
- Some other ones (often: tty12) may be used for displaying most recent log entries (*syslog* daemon).

Outline of the lecture

Desktop Environment

- OS-independent bundle of graphical applications.
- Core components started together offer a shell-independent GUI interface to the system.
- Applications that form a DE share common architecture and design principles.
- Typical components include:
  - desktop,
  - window manager,
  - panels and applets,
  - file manager,
  - terminal emulator,
  - session manager,
  - control center.

Desktop Environment (2)

- Applications that form a DE work as any other applications (for example: web browsers), in particular:
  - several DEs may be installed into a single system,
  - several instances of the same or different DEs may be started simultaneously,
  - some flexible DEs allow replacing some components with parts of other DEs.
- DEs may be started from a tty (*startx*, etc.) although using display managers is highly recommended.
- Local DE users may be granted additional privileges to periphery devices, multimedia and removable storage as well as to a system power control via dedicated applications.

Underlying Libraries

- *X Window System* (X11, X) provides basic graphical functionality:
  - input drivers,
  - graphical drivers and accelerators,
  - fonts,
  - libraries that allow using graphical primitives (lines, dots, circles, etc.).
- *Widgets libraries* (over X) offer building blocks for interfaces: windows, menus, buttons, toolbars, scrollbars, dialog windows, etc.
- Most DEs are built on the top of one of the following two widget libraries:
  - GTK+ (GIMP Toolkit),
  - Qt.

GTK+ (GIMP Toolkit)

- Originally written in C (although there are bindings for most languages).
- Programming high level languages with GTK+ is relatively easy.
- Using GTK+ in C programs requires advanced programming skills (casting structure and function pointers).
- Free software (GNU LGPL).
- Well supported on all Unix systems.
- Ported to other platforms (including Windows).
- Three versions co-exist: GTK2 (obsolete), GTK3 (legacy) and GTK4 (current).
- website: `https://www.gtk.org`

Qt

- Originally written in C++ (although there are bindings for other object-oriented languages).
- Easier for beginners in comparison to GTK+.
- Initially proprietary software of Trolltech (free of charge for non-commercial use). Later releases are cross-licensed.
- Best support under Linux. Support under commercial Unices varies.
- Ported to other platforms (including Windows). Well supported on mobile devices.
- Three versions co-exist: Qt4 (obsolete), Qt5 (legacy) and Qt6 (current).
- website: `https://www.qt.io`

Popular Free Desktop Environments

- GTK+:
  - MATE (GNOME2-lite),
  - GNOME3, GNOME4,
  - XFCE,
  - LXDE,
  - Cinammon, etc.
- Qt:
  - KDE,
  - LXQt.

Stable and Unstable Releases

GNOME, MATE and XFCE follow the same version numbering scheme: X.Y.Z, where:

- X (*major number*) – main line,
- Y (*minor number*) – version/release:
  - even number indicates a *stable* (end-user, final) release,
  - odd number indicates an *unstable* (development, under way) release.
- Z (*revision number*) – a bugfix sub-release; bugs fixed, no new features added.

The same convention is commonly accepted by numerous free and open projects.

GNOME2

GTK2-based desktop environment:

- for many years de facto standard DE for Unices and Linuces,
- defines HIG (*Human Interface Guidelines*) – rules for designing user-friendly, uniform, intuitive, clear interfaces,
- the most user-friendly and easy-to-use DE,
- moderately eye-candy but functional,
- complete set of (integrated) applications for misc. purposes included,
- reduced configuration options (criticized by many inc. Linus Torvalds),
- large, heavy, resource-consuming, inflexible, tightly integrated,
- subsequent stable releases every 6 months; abandoned since 2011.

GNOME3 and GNOME4

GTK3-based project started in 2011:

- a controversial, completely new, *modern* interface similar to those designed for mobile devices,
- no usual elements like a desktop, panels, main menu etc,
- limited configuration and customization,
- heavy, inflexible, tightly integrated,
- initially received a very critical reception,
- stable releases every 6 months,
- website: `https://www.gnome.org`

Introducing GTK4 library resulted in changing major number to GNOME4.

XFCE

GTK+ (GTK3) desktop environment:

- GNOME compatible,
- compliant with HIG and other open standards,
- modular, lightweight, minimalistic, simple and flexible,
- fully functional and feature-rich,
- slightly old-fashioned look, especially with default configuration,
- subsequent stable releases *when we have finished*,
- website: `https://www.xfce.org`

MATE

GTK+ desktop environment:

- based on abandoned GNOME2 code (all sub-project names changed),
- rewritten to GTK3,
- limited to only core components, non critical ones removed (aka GNOME2-lite),
- development aims at making it more lightweight, flexible and configurable,
- features and requirements comparable to XFCE,
- new stable releases every 6 months (same as GNOME),
- website: `https://www.mate-desktop.org`

LXDE

GTK+ (GTK2 and GTK3) desktop environment:

- similar to XFCE and MATE,
- aimed at providing functional DE for older and slower computers,
- lowest resource consumption,
- heterogeneous; re-uses components of other DEs,
- lightweight and flexible but some incoherences and design issues do exist,
- rolling release model for components, no releases of a whole DE,
- website: `https://www.lxde.org`

KDE

The most popular Qt desktop environment:

- numerous Windows users find it easy and natural,
- the most *eye-candy* and full of *bells and whistles*,
- highly configurable,
- doesn't follow HIG recommendations,
- huge, heavy, resource consuming, inflexible, de facto monolithic,
- website: `https://www.kde.org`

LXQt

- LXDE clone ported from GTK+ to Qt.
- Initially intended as a replacement for LXDE; finally those two projects co-exist.
- Website: `https://lxqt-project.org`

Freedesktop.org

An initiative for DE interoperability:

- standard libraries, communication routines and configuration file formats,
- common font rendering engines,
- unified description files for adding applications to menu entries,
- standardized icon themes,
- common directories/files for desktop, trashbin, etc.
- applications designed for GTK+ environments should look natively under Qt environments and vice versa.