Training - Displaying and rendering graphics with Linux

2-day session

Title	**Training - Graphics display and rendering with Linux**
Overview	Image and color representation Pixel Drawing Pixel Operations Pipeline Components Overview and Generalities Display Hardware Specifics Rendering Hardware Specifics System Integration, Memory and Performance Display Stack Overview TTY Kernel Aspects Framebuffer Device Kernel Aspects DRM Kernel Aspects DRM Userspace Aspects X Window Userspace Aspects Wayland Userspace Aspects Mesa 3D Userspace Aspects
Materials	Materials for this course are still under development. They will be released under a free documentation license after the first session is delivered.
Duration	**Two** days - 16 hours (8 hours per day). 75% of lectures, 25% of demos.
Trainer	One of the engineers listed on: `https://bootlin.com/training/trainers/`
Language	Oral lectures: English or French. Materials: English.
Audience	People developing multimedia devices using the Linux kernel
Prerequisites	**C programmig language, basic knowledge of concepts related to low-level hardware interaction (e.g. registers, interrupts), kernel-level system management (e.g. virtual memory mappings) and userspace syscalls (e.g. ioctl, mmap). Basic knowledge of concepts related to hardware interfaces (e.g. clocks, busses).**

Required equipment	**For on-site sessions only** Everything is supplied by Bootlin in public sessions. • Video projector • Large monitor • Drawing board
Materials	Electronic copies of presentations slides

Day 1 - Morning

Lecture - Image and Color Representation

- Light, pixels and pictures
- Sampling, frequency domain, aliasing
- Color quantization and representation
- Colorspaces and channels, alpha
- YUV and chroma sub-sampling
- Pixel data planes, scan order
- Pixel formats, FourCC codes, modifiers

Introducing the basic notions used for representing color images in graphics.

Lecture - Pixel Drawing

- Accessing and itegrating over pixel data
- Concepts about rasterization
- Rectangle drawing
- Linear gradient drawing
- Disk drawing
- Circular gradient drawing
- Line drawing
- Line and shape aliasing, sub-pixel drawing
- Circles and polar coordinates
- Parametric curves

Presenting how to access pixel data in memory and draw basic shapes.

Lecture - Pixel Operations

- Region copy
- Alpha blending
- Color-keying
- Scaling and interpolation
- Linear filtering and convolution
- Blur filters
- Dithering

Providing basic notions about filtering, with very common examples of how it's used.

Demo - Drawing and operations

- Examples of various shapes and region drawing
- Examples of basic pixel operations

Illustrating the concepts presented along the way.

Day 1 - Afternoon

Lecture - Pipeline Components Overview and Generalities	**Lecture - Display hardware**
• Types of graphics hardware implementations • Graphics memory and buffers • Graphics pipelines • Display, render and video hardware overview *Presenting the hardware involved in graphics pipelines.*	• Visual display technologies: CRT, plasma, LCD, OLED, EPD • Display timings, modes and EDID • DIsplay interfaces: VGA, DVI, HDMI, DP, LVDS, DSI, DP • Bridges and transcoders *Presenting the inner workings of display hardware.*

Lecture - Rendering Hardware Specifics	**Lecture - System Integration, Memory and Performance**
• Digital Signal Processors (DSPs) • Dedicated hardware accelerators • Graphics Processing Unit (GPUs) *Describing the architecture of processing and rendering hardware.*	• Graphics integration and memory • Shared graphics memory access • Graphics memory constraints and performance • Offloading graphics to hardware • Graphics performance tips *Topics related to graphics integration, memory management and performance aspects.*

Day 2 - Morning

Lecture - Display Stack Overview

- System-agnostic overview: kernel, display and render userspace
- Linux kernel overview
- Linux-compatible low-level userspace overview
- X Window and Wayland overview
- High-level graphics libraries and desktop environments overview

Presenting what software components are required for modern computer graphics and how they are divided between kernel and userspace.

Lecture - TTY Kernel Aspects, Framebuffer Device Kernel Aspects

- Linux TTY subsystem introduction
- Virtual terminals and graphics
- Virtual terminals switching and graphics

- Fbdev overview
- Fbdev basic operations
- Fbdev limitations

How TTYs interact with graphics in Linux along with a short presentation of fbdev and why it's deprecated.

Lecture - DRM Kernel Aspects

- DRM devices
- DRM driver identification and capabilities
- DRM master, magic and authentication
- DRM memory management
- DRM KMS dumb buffer API
- DRM FourCCs and modifiers
- DRM KMS resources probing
- DRM KMS modes
- DRM KMS framebuffer management
- DRM KMS legacy configuration and page flipping
- DRM event notification
- DRM KMS object properties
- DRM KMS atomic
- DRM render
- DRM Prime zero-copy memory sharing (dma-buf)
- DRM sync object fencing
- DRM debug and documentation

An exaustive presentation of the DRM interface.

Demo - Kernel Aspects

- Linux TTY and virtual terminals
- DRM KMS mode-setting
- DRM KMS driver walkthrough
- DRM render driver walkthrough

Illustrating how kernel aspects work.

Day 2 - Afternoon

Lecture - X Window Userspace Aspects

- X11 protocol and architecture
- X11 protocol extensions
- Xorg architecture and acceleration
- Xorg drivers overview
- X11 and OpenGL acceleration: GLX and DRI2
- Xorg usage, integration and configuration
- Major issues with X11
- Xorg debug and documentation

Presenting all things related to X11 and Xorg.

Lecture - Wayland Userspace Aspects

- Wayland overview and paradigm
- Wayland protocol and architecture
- Wayland core protocol detail
- Wayland extra protocols
- Wayland asynchronous interface
- Wayland OpenGL integration
- Wayland status and adoption
- Wayland debug and documentation

An in-depth presentation of Wayland.

Lecture - Mesa 3D Userspace Aspects

- Standardized 3D rendering APIs: OpenGL, OpenGL ES, EGL and Vulkan
- Mesa 3D overview
- Mesa 3D implementation highlights
- Mesa 3D internals: Gallium 3D
- Mesa 3D internals: intermediate representations
- Mesa 3D Generic Buffer Management (GBM)
- Mesa 3D hardware support status
- Mesa 3D versus proprietary implementations
- Mesa 3D hardware support: debug and documentation

Presenting 3D APIs and the Mesa 3D implementation.

Demo - Userspace Aspects

- Xorg code walkthrough
- Wayland compositor core walkthrough
- Wayland client examples
- Mesa code walk-through
- OpenGL and EGL examples

Illustrating userspace aspects, client and server implementations.