

Linux Graphics Driver Development

Eric Anholt
Open Source Technology Center
Intel Corporation

Where code lives

- kernel modules:
 - [git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux-2.6.git](https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux-2.6.git)
- libdrm:
 - [git://anongit.freedesktop.org/git/mesa/drm](https://anongit.freedesktop.org/git/mesa/drm)
- X Server:
 - [git://anongit.freedesktop.org/git/xorg/xserver](https://anongit.freedesktop.org/git/xorg/xserver)
- 2D driver:
 - [git://anongit.freedesktop.org/git/xorg/driver/xf86-video-intel](https://anongit.freedesktop.org/git/xorg/driver/xf86-video-intel)
- OpenGL driver:
 - [git://anongit.freedesktop.org/git/mesa/mesa](https://anongit.freedesktop.org/git/mesa/mesa)

What's changed

- Kernel modules now developed in the kernel tree
 - Kernel developers are now aware of graphics needs
 - Rapid integration of new development into kernel releases
 - Reduced integration effort for distributions

What's changed

- Kernel memory management reduces userland complexity and common failures
 - No more having 4 memory managers in the driver
 - No more static splits between 2D and 3D and video memory
 - No more static framebuffer size

What's changed

- Kernel modesetting is moving the bulk of the driver to the kernel
 - Faster boot times
 - Less flicker
 - Printing of oopses and panics
 - No VT-switch path
 - Native suspend/resume (no more vbetool)
 - GPGPU support
-
-

Building it from scratch



Kernel modesetting

- Built in `linux-2.6/drivers/gpu/drm/chipset_*.c`
- Ignore command submission for now
- Create a simple memory manager for your chipset's local or UMA memory (`drm_mm.c`)
- Add support for setting video modes using framebuffer memory in a GEM buffer object
- Reference code:
 - `drm-intel-next` branch
`git://git.kernel.org/pub/scm/linux/kernel/git/anholt/drm-intel`
 - `linux-2.6/graphics/drivers/gpu/drm/i915/i915_display.c`
 - `linux-2.6/graphics/drivers/gpu/drm/i915/i915.crt.c`

2D driver for X Server

- No reason to do anything but X.Org X Server
 - Use kernel modesetting to collect and set video modes
 - Add `DRM_IOCTL_CHIPSET_GEM_CREATE`
 - Add `DRM_IOCTL_CHIPSET_GEM_MMAP`
 - Use `fbScreenInit` to set up software rendering
 - (Use `shadowInit` to speed up software rendering)
 - Reference code:
 - `xf86-video-intel` master branch
 - `linux-2.6/drivers/gpu/drm/i915/i915_gem.c`
-
-

First acceleration support

- Add `DRM_IOCTL_CHIPSET_GEM_EXECBUF`
 - Receives list of buffers
 - Receives list of relocations from one buffer to another
 - Uploads each buffer to the card
 - Update all relocations
 - Dispatch the batchbuffer
 - Sync the hardware
 - Download dirty buffers from the card
 - Reference:
 - `linux-2.6/drivers/gpu/drm/i915/i915_gem.c`
-
-

Batchbuffer security

- DRM guarantees that a user can't use it to escalate privilege
 - (DRM doesn't guarantee that a user can't use it to hang the system)
 - If your batchbuffers could DMA to/from system memory, they should be passed to `execbuf` as user memory pointers and checked before submission
 - Reference code:
 - `drivers/gpu/drm/radeon/radeon_cp.c`
-
-

Accelerated 2D driver for X Server

- Create libdrm_chipset in libdrm
 - Add buffer manager to libdrm
 - Add EXA support with pixmaps in buffer objects
 - Solid
 - Copy
 - Save Composite for later
 - Reference code:
 - `drm/libdrm/intel/intel_bufmgr_gem.c`
 - `xf86-video-intel/src/i830_exa.c`
-
-

Better cache handling

- GPU write caches need to be flushed before writing to buffers with the CPU
 - GPU read caches only need to be invalidated when changed in a non-coherent cache domain
 - GPU write caches may need to be flushed for the results to reach scanout.
 - Solution: track which cache domain the buffer has dirty cache lines in, and which read cache domains the buffer has had invalidated since the last write.
 - Intel ended up with CPU, GTT, RENDER, SAMPLER, INSTRUCTION, and COMMAND.
-
-

OpenGL driver

- DRI2 is simplifying OpenGL drivers
 - No more hardware lock
 - No more cliprects
 - No more sarea
 - Should be released by end of year
 - Gallium is still experimental
 - Rapid code change
 - Individual drivers are regularly broken
 - Reference code
 - `mesa/src/mesa/drivers/dri/i965` (shader model 3.0)
 - `mesa/src/mesa/drivers/dri/i915` (DX9-ish and DX7-ish)
-
-

Further work

- Composite hook in EXA gives accelerated text rendering and screen rotation
- EGL support
- GLES 1.1 support
- Reference code:
 - `xf86-video-intel/src/i915_render.c`
 - `git://people.freedesktop.org/~krh/eagle`
 - `http://dgles.sourceforge.net/`