Open Source Graphic Drivers—They Don't Kill Kittens

David M. Airlie
Open Source Contractor
airlied@linux.ie

Abstract

This paper is a light-hearted look at the state of current support for Linux / X.org graphics drivers and explains why closed source drivers are in fact responsible for the death of a lot of small cute animals.

The paper discusses how the current trend of using closed-source graphics drivers is affecting the Open Source community, e.g. users are claiming that they are running an open source operating system which then contains a 1 MB binary kernel module and 10MB user space module...

The paper finally look at the current state of open source graphics drivers and vendors and how they are interacting with the kernel and X.org communities at this time (this changes a lot). It discusses methods for producing open source graphics drivers such as the r300 project and the recently started NVIDIA reverse engineering project.

1 Current Official Status

This section examines the current status (as of March 2006) of the support from various manufactures for Linux, X.org[4], and DRI[1] projects. The three primary manufacturers, Intel, ATI, NVIDIA, are looked at in depth along with a brief overview of other drivers.

1.1 Intel

Currently Intel contract Tungsten Graphics to implement drivers for their integrated graphics chipsets. TG directly contribute code to the Linux kernel, X.org, and Mesa projects to support Intel cards from the i810 to the i945G. Intel have previously released complete register and programmer's reference guides for the i810 chipset; however, from the i830 chipset onwards, no information was given to external entities with the exception of Tungsten.

As of March 2006, all known Intel chipsets are supported by the i810 X.org driver.

1.1.1 2D

The Intel integrated chipsets vary in the number and type of connectable devices. The desktop chipsets commonly only have native support for CRTs, and external devices are required to drive DVI or tv-out displays. These external devices are connected to the chipset using either the DVO (digital video output) on i8xx, or sDVO (serial digital video output) on i9xx. These external devices are controlled over an i2c bus. The mobile chipsets allow for an in-built LVDS controller and sometimes in-built tv-out controller.

Due to the number of external devices available from a number of manufacturers (e.g. Sil-

icon Image, Chrontel), writing a driver is a lot of hard work without a lot of manufacturer datasheets and hardware. The Intel BIOS supports a number of these devices.

For this reason the driver uses the Video BIOS (VBE) to do mode setting on these chipsets. This means that unless the BIOS has a mode pre-configured in its tables, or a mode is hacked in (using tools like i915resolution), the driver cannot set it. This stops the driver being used properly on systems where the BIOS isn't configured properly (widescreen laptops) or the BIOS doesn't like the mode from the monitor (Dell 2005FPW via DVI).

1.1.2 3D

The Intel chipsets have varying hardware support for accelerating 3D operations. Intel "distinguish" themselves for other manufacturers by not putting support for TNL in hardware, preferring to have optimized drivers do that stuff in software. The i9xx added support for HW vertex shaders; however, fragment shading is still software-based. The 3D driver for the Intel chipsets supports all the features of the chipset with the exception of *Zone Rendering*, a tile-based rendering approach. Implementing zone rendering is a difficult task which changes a lot of how Mesa works, and the returns are not considered great enough yet.

However, in terms of open source support for 3D graphics, Intel provide by far the best support via Tungsten Graphics.

1.2 ATI

ATI, once a fine upstanding citizen of then open source world (well they got paid for it), no longer have any interest in our little adventures and have joined the kitten killers. The ATI cards can be broken up into 3 broad categories, pre-r300, r3xx-r4xx, and r5xx.

1.2.1 pre-r300

Thanks to the Weather Channel wanting open-source drivers for the Radeon R100 and R200 family of cards, and paying the money, ATI made available cut-down register specifications for these chipsets to the open-source developer community via their Developer Relations website. These specifications allowed Tungsten Graphics to implement basically complete 3D drivers for the r100 and r200 series of cards. However, ATI didn't provide any information on programming any company-proprietary features such as Hyper-Z or TruForm. ATI's engineering also provided some support over a number of years for 2D on these cards, such as initial render acceleration, and errata for many chips.

1.2.2 r300–r4xx

The R300 marked the first chipset that ATI weren't willing to provide any 3D support for their cards. A 2D register spec and development kit was provided to a number of developers, and errata support was provided by ATI engineering. However, no information on the 3D sections of this chipset were ever revealed to the open source community. A number of OEMs have been provided information on these cards, but no rights to use it for open-source work.

ATI's fglrx closed-source driver appeared with support for many of the 3D features on the cards; it, however, has had certain stability problems and later versions do not always run on older cards.

This range of cards also saw the introduction of PCI Express cards. Support for these cards came quite late, and a number of buggy fglrx releases were required before it was stabilised.

1.2.3 r5xx

The R5xx is the latest ATI chipset. This chipset has a completely redesigned mode setting, and memory controller compared to the r4xx, the 3D engine is mostly similiar. Again no information has been provided to the open-source community. As of this writing no support beyond vesa is available for these chipsets. ATI have not released an open-source 2D driver or a version of fglrx that supports these chips, making them totally useless for any Linux users.

1.2.4 fglrx

The ATI fglrx driver supports r200, r300, and r400 cards, and is built using the DRI framework. It installs its own libGL (the DRI one used to be insufficent for their needs) and a quite large kernel module. FGLRX OpenGL support can sometimes be a bit useless, Doom3 for example crashes horribly on fglrx when it came out first.

1.3 NVIDIA

Most people have thought that NVIDIA were always evil and never provided any specs, which isn't true. Back in the days of the riva chipsets, the Utah-GLX project implemented 3D support for the NVIDIA chipsets using NVIDIA-provided documentation.

1.3.1 2D

NVIDIA have some belief in having a driver for their chipsets shipped with X.org even if it only supports basic 2D acceleration. This at least allows users to get X up and running so they can download the latest binary driver for their cards, or at least use X on PPC and other non-x86 architectures. The nv driver in X.org is supported by NVIDIA employees and despite it being written in obfuscated C}ˆHhex code, the source is there to be tweaked. BeOS happens to have a better open source NVIDIA driver with dual-head support, which may be ported to X.org at some point.

1.3.2 3D

When it comes to 3D, the NVIDIA 3D drivers are considered the best "closed-source" drivers. From an engineering point of view, the drivers are well supported, and NVIDIA interact well with the X.org community when it comes to adding new features. The NVIDIA driver provides support for most modern NVIDIA cards; however, they recently dumped support for a lot of older cards into a legacy driver and are discontinuing support in the primary driver. NVIDIA drivers commonly support all features of OpenGL quite well.

1.4 Others

1.4.1 VIA and SiS

Other manufactures of note are Matrox, VIA, and SiS. VIA and SiS both suffer from a serious lack of interaction with the open-source community, most likely due to some cultural differences between Taiwanese manufacturers and open-source developers. Both companies

occasionally code-drop drivers for hardware with bad license files, no response to feedback, but with a nice shiny press release or get in touch with open-source driver writers with promises of support and NDA'd documentation, but nothing ever comes of it. Neither company has a consistent approach to open source drivers. VIA chipsets have good support for features thanks to people taking their code drops and making proper X.org drivers from them (unichrome and openchrome projects), and SiS chipsets (via Thomas Winischofer) have probably by far the best 2D driver available in terms of features, but their 3D drivers are a bit hit-and-miss, and only certain chipsets are supported at all.

1.4.2 Matrox

Matrox provide open source drivers for their chipsets below G550; however, newer chipsets use a closed-source driver.

2 Closed Source Drivers—Reasons

So a question the author is asked a lot is why he believes closed source drivers are a bad thing. I don't consider them bad so much as pure evil, in the kitten-killing, seal-clubbing sense. Someone has to hold an extreme view on these things, and in the graphics driver case that is the author's position. This sections explores some of the reasons why open-source drivers for graphics card seems to be going the opposite direction to open-source drivers for every other type of hardware.

This is all the author's opinion and doesn't try to reflect truth in any way.

2.1 Reason—Microsoft

The conspiracy theorists among us (I'm not a huge fan), find a way to blame Microsoft for every problem in Linux. So to keep them happy, I've noticed two things.

- Microsoft decided to use a vendor's chip in the XBOX series → no specs anymore.
- Chipset vendors puts DirectX 8.0 support into a chip → no specs anymore.

Hope this keeps that section happy.

2.2 Reason—???

Patents and fear of competitors or patent scumsucking companies bringing infringement against the latest chipset release and delaying it, is probably a valid fear amongst chip manufacturers. They claim releasing chipset docs to the public may make it easier for these things to be found; however, most X.org developers have no problem signing suitable NDAs with manufacturers to access specs. Open source drivers may show a company's hand to a greater degree. This is probably the most valid fear and it is getting more valid due to the great U.S. patent system.

2.3 Reason—Profit

Graphics card manufacturing is a very competitive industry, especially in the high-end gaming, 3–6 month development cycle, grind-out-as-many-different-cards-as-you-can world that ATI and NVIDIA inhabit. I can't see how open sourcing drivers would slow down these cycles or get in the way—apart from the fact that the dirty tricks to detect and speed up quake 3

might be spotted easier (everyone spots them in the closed source drivers anyways). It doesn't quite explain Matrox and those guys who don't really engage in the gamer market to any great degree. It also doesn't really explain fglrx which are some of the most unsuitable drivers for gaming on Linux.

Also things like SLI and Crossfire bring to question some of the profit motivation; the number of SLI and Crossfire users are certainly less than the number of Linux users.

3 Closed Source Drivers—Killing Kittens

3.1 Fluffy—Open Source OS

Linux is an open source OS. Linux has become a highly stable OS due to its open source nature. The ability for anyone to be able to fix a bug in any place in the OS, within reason, is very useful for implementing Linux in a lot of server and embedded environments. Things like Windows CE work for embedded systems as long as you do what MS wanted you to do; Linux works in these systems because you don't have to follow the plan of another company: you are free to do your own things. Closed source drivers take this freedom away.

If you load a 1MB binary into your Linux kernel or X.org, you are **NO LONGER RUNNING AN OPEN SOURCE OS**. Lots of users don't realise this, they tell their friends all about open source, but use NVIDIA drivers.

3.2 Mopsy—Leeching

So on to why the drivers are a bad thing. Linux developers have developed a highly stable OS and provide the source to it, X.org is finally getting together a window system with some modern features in it and are providing the source to it. These developers are also providing the ideas and infrastructure for these things openly. Closed source vendors are just not contributing to the pool of knowledge and features in any decent ways. Open source developers are currently implementing acceleration architectures and memory management systems that the closed source drivers have had for a few years. These areas aren't exactly the family jewels, surely some code might have been contributed or some ideas on how things might be done.

3.3 Kitty—niche systems

There are a lot of niche systems out there, installations in the thousands that normally don't interest the likes of NVIDIA or ATI. The author implements embedded graphics systems, and originally use ATI M7s but now uses Intel chipsets where possible. These sales, while not significant to ATI or NVIDIA on an individual basis, add up to a lot more than the SLI or CrossFire sales ever will. However these niche systems usually require open source drivers in order to do something different. For example, the author's systems require a single 3D application but not an X server. Implementing this is possible using open source drivers; however, doing so with closed source driver is not possible. Also, for non-x86 systems such as PPC or Sparc, where these chips are also used, getting a functional driver under Linux just isn't possible.

3.4 Spot—out-dated systems

Once a closed vendor has sold enough of a card, it's time to move on and force people somehow

to buy later cards. Supporting older cards is no longer a priority or profitable. This allows them to stop support for these cards at a certain level and not provide any new features on those cards even if it possible. Looking at the features added to the open-source radeon driver since its inception shows that continuing development on these cards is possible in the open source community. NVIDIA recently relegated all cards before a certain date to their legacy drivers. Eventually these drivers will probably stop being updated, meaning running a newer version of Linux on those systems will become impossible.

4 Open Source Drivers—Future

This section discusses the future plans of open source graphic driver development. This paper was written in March 2006, and a lot may have happened between now and the publishing date in July.[1] The presentation at the conference will hopefully have all-new information.

4.1 Intel

Going forwards, Intel appear to be in the best positions. Recent hirings show their support for open source graphics, and Tungsten Graphics have added a number of features to their drivers and are currently implementing an open source video memory manager initially on the Intel chipsets. Once the mode-setting issues are cleared up, the drivers will be the best example out there.

The author has done some work to implement BIOS-less mode setting on these cards for certain embedded systems, and hopes that work can be taken forward to cover all cards and integrated into the open source X.org driver and become supported by Intel/TG.

[1] [Well, *besides* an awful lot of formatting, copyediting, and such—*OLS Formatting Team.*]

4.2 ATI

4.2.1 R3xx + R4xx 3D support

The R300 project is an effort to provide an open-source 3D driver for the r300 and greater by reverse engineering methods. The project has used the fglrx and Windows drivers to reverse engineer the register writes used by the r3xx cards. The method used involved running a simple OpenGL application and changing one thing at a time to see what registers were written by the driver. There are still a few problems with this approach in terms of stability, as certain card setup sequences for certain cards are not yet known (radeon 9800s fall over a lot). These sequences are not that easy to discover; however, tracing the fglrx startup using valgrind might help a lot.

While this project has been highly successful in terms of implementing the feature set of the cards, the lack of documentation and/or engineering support hamper any attempts to make this a completely stable system.

4.2.2 R5xx 2D support

A 2D driver for the R5xx series of cards from the author may appear; however, the author would like to engage ATI so as to avoid getting sued into the ground, due to a lot of information being available under NDA via an OEM. Most of the driver has, however, been reverse engineered by tracing the outputs from the video bios when asked to set a mode, using a modified x86 emulator.

4.3 NVIDIA

Recently an X.org DRI developer (Stephane Marcheu) announced the *renoveau project* [3], an attempt to build open-source 3D drivers for NVIDIA cards. This project will use the same methods as the r300 project to attempt to get at first a basic 3D driver for the NVIDIA cards.

5 Reverse Engineering Methodologies

This section just looks at some of the commonly used reverse engineering methodologies in developer graphics drivers.

5.1 2D Modesetting

Most cards come with a video BIOS that can set modes. Using a tool like LRMI[2], the Linux Real Mode Interface, the BIOS can be run inside an emulator. When the BIOS uses an `inl` or `outl` instruction to write to the card, LRMI must actually do this for it, so these calls can be trapped and used to figure out the sequence of register writes necessary to set a particular mode on a particular chipset. Multiple runs can be used to track exactly where the mode information is emitted.

This method has been used by the author in writing mode-setting code for Intel i915 chipsets, for intelfb and X.org, and also for looking at the R520 mode setting.

Another method, if a driver exists for a card under Linux already: a number of developers have discussed using an mmap trick, whereby a framework is built which loads the driver, and fakes the mmap for the card registers. The framework then catches all the segmentation faults and logs them while passing them through. This has been used by Ben Herrenschmidt for radeonfb suspend/resume support on Apple drivers. An enhancement to this by the author (who was too lazy to write an mmap framework for x86) uses a valgrind plugin to track the mmap and read/writes to an mmaped area. This solution isn't perfect (it only allows reading back writes after they happen), but it has been sufficent for work on the i9xx reverse engineering.

5.2 3D

Most 3D cards have some form of two-section drivers, a kernel-space manager and a userspace 3D driver linked into the application via libGL. The kernel-space code normally queues up command buffers from the userspace driver. The userspace driver normally mmaps the command queues into its address space. An application linked with libGL can do some simple 3D operations and then watch the command buffers as the app fills them. Tweaking what the highlevel application does allows different command buffers to be compared and a map of card registers vs. features can be built up. The r300 project has been very successful with this approach.

The r300 project also have a tool that runs under Windows, that constantly scans the shared buffers used by the windows drivers, and dumps them whenever they change in order to do similiar work.

6 Conclusion

This paper has looked at the current situation with graphics driver support for Linux and

X.org from card manufacturers. It looks at why closed source drivers are considered evil and looks at what the open source community is doing to try and provide drivers for open source cards. Just remember, save those kittens.

References

[1] DRI Project.
`http://dri.freedesktop.org`.

[2] Linux Real Mode Interface.
`http://lrmi.sf.net`.

[3] The Renoveau Project.
`http://renoveau.sf.net/`.

[4] The X.org Project.
`http://www.x.org/`.

Proceedings of the Linux Symposium

Volume One

July 19th–22nd, 2006
Ottawa, Ontario
Canada

Conference Organizers

Andrew J. Hutton, *Steamballoon, Inc.*
C. Craig Ross, *Linux Symposium*

Review Committee

Jeff Garzik, *Red Hat Software*
Gerrit Huizenga, *IBM*
Dave Jones, *Red Hat Software*
Ben LaHaise, *Intel Corporation*
Matt Mackall, *Selenic Consulting*
Patrick Mochel, *Intel Corporation*
C. Craig Ross, *Linux Symposium*
Andrew Hutton, *Steamballoon, Inc.*

Proceedings Formatting Team

John W. Lockhart, *Red Hat, Inc.*
David M. Fellows, *Fellows and Carr, Inc.*
Kyle McMartin