Reviving Vintage (Underserved) Graphics Hardware on X Server

Kevin Brace
Brace Computer Laboratory
https://bracecomputerlab.com
XDC 2019

Is It Vintage or Really Just *Underserved*?

- My origin of first hearing the word *underserved*

    Remembered a TV news story about underserved neighborhoods of taxi service in one large U.S. city from years ago. (this is before Smartphone and ride share companies existed)

- The word *underserved* is not really a recognized dictionary term

    Shows up as a misspelled word on LibreOffice Writer 5.1.

- The use of the word *underserved* is becoming more common

    Just saw an article a few days ago that used the same word multiple times regarding the lack of medical doctors in rural areas of one large U.S. state.

Having worked on graphics device driver code development for a few years, the word *underserved* appears to perfectly describe underdeveloped, neglected graphics device driver code.

The Issues When Dealing with Older Graphics Hardware

- Ugly, unmaintainable device drivers

    Different authors with varying coding styles. Mainly from XFree86 era. (mid '90s to early 2000s)

- Device driver breakage

    Most breakages are due to XAA (X Acceleration Architecture) support elimination.

- Some of them have too little memory

    1 MB frame buffer with 32-bit bus width . . . (no, that's not a typo)

- Some of them have a RAMDAC speed below 135 MHz

    Some may ask, what is a RAMDAC?

The Issues When Dealing with Older Graphics Hardware

- Some of them do not even support VESA DDC

    No way to acquire monitor screen resolution support information.

- Some devices do not even support 32-bit bpp (24-bit depth) color

    Old SiS graphics (i.e., SiS 6326), S3 pre-ViRGE devices (i.e., Vision964)

- Physical deterioration

    They are 20+ years old. Some of them have electrolytic capacitor "bulging" issue.

- Rare ones are hard to obtain (< 30 USD secondhand on eBay)

    Imagination Technologies PowerVR Series2, STMicro KYRO (i.e., PowerVR Series3), Trident Microsystems Blade XP (or its derivatives), S3 Graphics devices after Savage 2000, SiS Xabre, XGI Volari V5 / V8 (Duo), and XGI XP10

The Approach I Will Take Maintaining Old Code

- At least fix breakage bugs

    If the code worked at least nominally when XAA was still available, the cause of the bug needs to be tracked down and fixed.

- Compilation warnings elimination

    Have done this for several DDXs.

- Clean up white space / tab mixture and indentation issues

    White space conversion can be done automatically, but 72 line manual indentation adjustment is very time consuming. (feels meaningless)

- Release archival versions

    Before I mess up the code.

The Approach I Will Take Developing New Code

- Move legacy UMS (User Mode Setting) code into newly carved out UMS section

- Implement dual UMS / KMS support

  UMS will still be available if the user wishes.

- Leverage the experience I gained from developing OpenChrome DRM

  While OpenChrome DRM development is not really meant for this purpose, but it effectively clears out development obstacles / learning curve issues for other DRM / KMS implementations.

- Prove new code functionality with OpenChrome DRM little by little rather than doing "all from scratch" DRM code at once

- Implement atomic mode setting using atomic modeset helper as a baseline implementation

Currently Working on a New DRM from Scratch

- I will pick Trident Microsystems devices for my first DRM implementations

- Picked Trident because there is no Trident Microsystems DRM in the Linux kernel mainline tree

    ATI Technologies RAGE 128, S3 Savage, Matrox G series, SiS, and 3dfx already have old DRI1 based DRM, and I feel this complicates the code development

- Trident was famous (notorious?) for producing chips that went into many low performance 30 USD graphics cards sold at mom-and-pop computer dealers before integrated graphics became the norm for the low end of the market

- Development pace is fairly slow because OpenChrome DRM development takes precedence

- The code exists only inside my computers for now

    No public repository is set up yet. I need to figure out how to do this.

Accomplishments So Far (X Server)

- "Managed" to release X.Org X Server 1.19.7

    Believe it or not, it took a month and half to figure out how to compile, test, and package the code mainly due to **xorg/util/modular/release.sh** regressions.

    Generally, use the **xorg/util/modular/release.sh** build script used around the time the particular X Server version's original code was released.

    Keyboard mapping data needs to be imported properly or the X Server will not boot.

    Essentially, X Server 1.19.7 is a SiS 6326 24-bit bpp EXA fix (one line of code) edition, and may also issue this same fix for X Server 1.18.

Accomplishments So Far (DDX)

- xf86-video-apm 1.3.0 (Alliance Semiconductor ProMotion DDX)

- xf86-video-chips 1.3.0 and 1.4.0 (Chips & Technologies DDX)

- xf86-video-i128 1.4.0 (Number Nine Imagine 128 DDX)

- xf86-video-i740 1.4.0 (Intel740 DDX)

- xf86-video-mga 2.0.0 (Matrox DDX)

- xf86-video-neomagic 1.3.0 (NeoMagic DDX)

- xf86-video-s3 0.7.0 (S3 DDX)

- xf86-video-s3virge 1.11.0 (S3 ViRGE DDX)

- xf86-video-sis 0.11.0 (SiS DDX)

- xf86-video-tdfx 1.5.0 (3dfx DDX)

Conclusion

- Calling old graphics hardware *vintage* is just a positive spin; *underserved* is a much more accurate term

- New breakage fix, compilation warnings elimination, indentation fix, and archiving before adding KMS support

- Leverage OpenChrome DRM experience and infrastructure in order to help develop other DRMs, and find synergy (i.e., reduce development effort) between them

- Writing DRM with KMS support from scratch without someone else more experienced providing help is pretty unrealistic (my personal opinion)