

VRAM Depo Aygıtı NASIL

Yazan:
Michal Schulz

<michal.schulz (at) tu-clausthal.de>

Çeviren:
Arman Aksoy

<armish (at) linux-sevenler.de>

Çeviren:
Erkan Kaplan

<Selamsana (at) uni.de>

20 Eylül 2002

Özet

GFX kartınızdaki belleği farklı bir yoldan nasıl kullanabilirsiniz?

Konu Başlıkları

1. Giriş	3
2. Gerekenler	3
3. Hazırlanma	3
4. MTD Ayarları	3
5. Birkaç Hesaplama	4
6. XF86Config Dosyasında Yapılacaklar	4
7. MTD Aygıtının Oluşumu	4
8. Peki Bununla Ne Yapabilirim?	5

Geçmiş

1.0 20 Eylül 2002 Çeviri: Arman Aksoy, Özgün Belge: Michal Schulz
İlk sürüm – Belgenin özgün sürümü http://hedera.linuxnews.pl/_news/2002/09/03/_long/1445.html adresinde bulunabilir.

Yasal Uyarı

Bu belge çevirisinin, *VRAM Depo Aygıtı NASIL*, 1.0 sürümünün **telif hakkı © 2002 Arman Aksoy ve Erkan Kaplan**'a ve özgün belgenin **telif hakkı © 2002 Michal Schulz**'a aittir. Bu belgeyi, Free Software Foundation tarafından yayınlanmış bulunan GNU Genel Kamu Lisansının 2. ya da daha sonraki sürümünün koşullarına bağlı olarak kopyalayabilir, dağıtabilir ve/veya değiştirebilirsiniz. Bu Lisansın özgün kopyasını <http://www.gnu.org/copyleft/gpl.html> adresinde bulabilirsiniz.

BU BELGE “ÜCRETSİZ” OLARAK RUHSATLANDIĞI İÇİN, İÇERDİĞİ BİLGİLER İÇİN İLGİLİ KANUNLARIN İZİN VERDİĞİ ÖLÇÜDE HERHANGİ BİR GARANTİ VERİLMEMEKTEDİR. AKSİ YAZILI OLARAK BELİRTİLMEDİĞİ MÜDDETCİ TELİF HAKKI SAHİPLERİ VE/VEYA BAŞKA ŞAHISLAR BELGEYİ “OLDUĞU GİBİ”, AŞIKAR VEYA ZIMNEN, SATILABİLİRLİĞİ VEYA HERHANGİ BİR AMACA UYGUNLUĞU DA DAHİL OLMAK ÜZERE HİÇBİR GARANTİ VERMEKSİZİN DAĞITMAKTADIRLAR. BİLGİNİN KALİTESİ İLE İLGİLİ TÜM SORUNLAR SİZE AİTTİR. HERHANGİ BİR HATALI BİLGİDEN DOLAYI DOĞABİLECEK OLAN BÜTÜN SERVİS, TAMİR VEYA DÜZELTME MASRAFLARI SİZE AİTTİR.

İLGİLİ KANUNUN İCBAR ETTİĞİ DURUMLAR VEYA YAZILI ANLAŞMA HARİCİNDE HERHANGİ BİR ŞEKİLDE TELİF HAKKI SAHİBİ VEYA YUKARIDA İZİN VERİLDİĞİ ŞEKİLDE BELGEYİ DEĞİŞTİREN VEYA YENİDEN DAĞITAN HERHANGİ BİR KİŞİ, BİLGİNİN KULLANIMI VEYA KULLANILAMAMASI (VEYA VERİ KAYBI OLUŞMASI, VERİNİN YANLIŞ HALE GELMESİ, SİZİN VEYA ÜÇÜNCÜ ŞAHISLARIN ZARARA UĞRAMASI VEYA BİLGİLERİN BAŞKA BİLGİLERLE UYUMSUZ OLMASI) YÜZÜNDEN OLUŞAN GENEL, ÖZEL, DOĞRUDAN YA DA DOLAYLI HERHANGİ BİR ZARARDAN, BÖYLE BİR TAZMİNAT TALEBİ TELİF HAKKI SAHİBİ VEYA İLGİLİ KİŞİYE BİLDİRİLMİŞ OLSA DAHİ, SORUMLU DEĞİLDİR.

Tüm telif hakları aksi özellikle belirtilmediği sürece sahibine aittir. Belge içinde geçen herhangi bir terim, bir ticari isim ya da kuruma itibar kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılmış olması ona onay verildiği anlamında görülmemelidir.

1. Giriş

Neden grafik kartınızın bu kadar çok belleğe sahip olduğunu düşündüğünüz olur mu?

Yeterince RAM'a veya çok yavaş takas alanına sahip olduğunuzu mu düşünüyorsunuz?

Hızlı ram-disklere veya disksiz bir makinaya mı ihtiyacınız var?

Bunu yapabilirsiniz. Gidin ve şu ucuz 32MB grafik kartlarından alın, işin zevkini çıkarın.

2. Gerekenler

VRAM'ı bir depo aygıtı olarak kullanmanız için gerekenler:

- XFree86
- Modül olarak derlenmiş MTD (özellikle slram)
- Kullanmayacağımız kadar fazla belleğe sahip bir GfX kartı

3. Hazırlanma

Bu belleği kullanmak için öncelikle GfX kartımızın PCI adres boşluğu ile nerede olduğunu bulmamız gerekir. Bunu birçok yoldan yapabilirsiniz. Benim ilk yaptığım ise Gfx kartının ve kaynaklarının genel bir kontrolü oldu:

1. `lspci` komutunu vererek (veya daha ayrıntılı olan `lspci -vvv`) "VGA compatible controller" a benzer bir yazı bulunan bölümü bulun. Ekrana gelecek olan çıktıda en azından Gfx kartınız ve kaynaklarıyla ilgili bir girdi bulunacaktır. Benimkisi şöyleydi:

```
01:00.0 VGA compatible controller: nVidia Corporation Vanta [NV6] (rev15)
  (prog-if 00 [VGA])
    Flags: bus master, 66Mhz, medium devsel, latency 248, IRQ 10
    Memory at d6000000 (32-bit, non-prefetchable) [size=16M]
    Memory at d4000000 (32-bit, prefetchable) [size=32M]
    Expansion ROM at <unassigned> [disabled] [size=64K]
    Capabilities: [60] Power Management version 1
    Capabilities: [44] AGP version 2.0
```

Şu an iki adres olasılığı var. Belirtmeliyim ki bu değerler sadece benim `nvidia` kartım için geçerli ve sizinkiler bundan farklı olabilir. `Memory at`'le başlayan ilk satırda `MMIO` adres boşluğu ile ilgili bilgiler var. Diğerisi ise `gfx ram`. Bu bilgi yeterli değil. O zaman bazı ayrıntılar için XFree86'a başvurmamız gerekecek.

2. XFree86 bazı bilgilerin kayıtlarını tutar. Bu kayıt dosyalarını `/var/log` dizininde `XFree86.log` (veya benzeri bir isme sahip) dosyada bulabilirsiniz. Bu dosya içinde yapacağımız küçük bir arama ile bize gereken GFX ram ile ilgili (buradaki ismiyle `linear framebuffer`) bilgilerini edinebiliriz:

```
$ cat /var/log/XFree86.0.log | grep framebuffer
(**) NV(0): Depth 24, (--) framebuffer bpp 32
(--) NV(0): Linear framebuffer at 0xd4000000
```

Gördüğünüz gibi, benim GFX RAM'im `0xd4000000` fiziksel adresinde başlıyor. Artık kartımızın ne kadar belleğe sahip olduğunu ve nerede başladığını biliyoruz, hadi başlayalım artık!

4. MTD Ayarları

MTD, Memory Technology Devices'in kısaltmasıdır. (Ayrıntılı bilgi için <http://www.linux-mtd.infradead.org/> adresine göz atın). MTD bizim Flash bellekleri, DoC aygıtlarını vb. kullanmamızı sağlar. Şimdi ise, bunu biraz farklı bir yoldan kullanacağız (basit bir mantıkla).

MTD'nin sürücülerinden biri PCI BUS'ın doğrudan erişim sağlaması için Ram aygıtlarının adreslerini tutar. Okuyacağımız gibi:

Many PC chipsets are incapable of correctly caching system memory above 64M or 512M. A driver exists which allows you to use this memory with the linux-mtd system.

Yani;

Bazı PC cipsetleri, 64M veya 512M üzerindeki sistem belleklerini doğru bir şekilde önbellek olarak kullanamazlar. Bu belleklerin `linux-mtd` sistemi ile kullanılmasını sağlayan bir sürücü bulunur.

Biraz farklı bir açıklama fakat şu an için en uygunu bu sanırım. Kullanmak istediğimiz ek bir belleğimiz var. Bu belleğin çekirdek tarafından kullanılmasını sağlayamayız (Sağlansa bile bunun için daha farklı bir yöntem gerekir ve büyük ihtimalla iyi bir sonuç elde edilmez). Biz ise, bu belleği bir blok veya karakter aygıtı olarak kullanacağız. Bunu sağlayan sürücü `slram` olarak adlandırılır. Bunun yanında bize `mtddcore`, `mtddblock`(eğer blok aygıtı erişimi gerekiyorsa) ve/veya `mtddchar` (eğer karakter aygıtı erişimi gerekiyorsa) lazım.

5. Birkaç Hesaplama

XFree86, nerede görüntülenebilir bir grafik yaratacağını bilmek için, PCI bilgilerini kullanır. Bu da demek ki, eğer video belleğini azaltırsak ve XFree86'ya video kartın daha az bir belleğe sahip olduğunu söylersek, geriye kalan belleğe X11 tarafından dokunulmayacak veya kullanılmayacaktır. Bu kalan belleği bizim kullanmamız daha sonra mümkün olacaktır.

X11'in ihtiyaç duyduğu bellek miktarı, kullanılan video çözünürlüklerine bağlıdır. En kolay hesaplama yolu ise genişliği, yükseklik ve de ardından makinanın XFree86 tarafından en yüksek modda her pixel için kullandığı byte miktarı ile çarpmaktır. Daha sonra bunun üzerine kullanılmak üzere (mesela XAA uzantısı tarafından) bir miktar eklenebilir. Bu arada, çoğu GFX kartının 24bpp modlarında her pixel için 4 byte kullandığını belirteyim. Eğer konsolda framebuffer kullanıyorsanız, çözünürlüğünü de hesaba katmalısınız.

XFree86'ya 4MB bellek vermeye karar verdiğimiz varsayalım. (onaltılık olarak 4MB `0x00400000`'dir). Eğer video framebuffer `0xd4000000`'dan başlıyorsa, bizim kullanabileceğimiz bellek de (`0xd4000000 + 0x00400000`) `0xd4400000` adresinden başlıyor demektir. Şu an bildiklerimiz ise, aygıtımızın konumu ve bellek miktarı.

6. XFree86Config Dosyasında Yapılacaklar

XFree86'ya ne kadar bellek kullanacağını belirtebilmek için, XFree86 ile biraz oynamamız gerekir. Yoksa X11 bellek miktarını kendi saptayacak ve hepsini kendisi için kullanacaktır. Böyle birşey olmasını istemiyoruz elbette. Device (Aygıt) bölümünde şu satırı ekleyin:

```
VideoRam 4096
```

Bu satır, XFree86'nin kullanacağı bellek sınırını belirtir ki burada bu değer 4MB'dır. Bu ayarın kullanılması için XFree86'yı yeniden başlatmalısınız.



Uyarı

Bu yöntem nVidia sürücülerini çalışmaz, çünkü bu sürücü `VideoRam` seçeneğini önemsemez. Fakat XFree86'nın `nv` sürücüsü ile rahatlıkla çalışabilir.

7. MTD Aygıtının Oluşumu

Herşey hazırlandı. Artık başlayabiliriz: (root olarak)

```
# modprobe slram map=VRAM,0xd4400000,+0x00c00000
```

Komut Açıklaması: Burada VRAM aygıtını oluşturmak için yeni bir modül yüklüyoruz. `0xd4400000` belleğin fiziksel başlangıç adresi, `+0x00c00000` ise bellek miktarıdır.

Benim grafik kartım 16Mb belleğe sahip. X11 için 4Mb ayırdım. Gerisi (12MB= 0x00c00000) ise MTD aygıtı için. Bakalım neler yapmışız:

```
# cat /proc/mtd
dev: size erasesize name
mtd0: 00b00000 00010000 "VRAM"
```

Hurraa...:-)) `mtd0` aygıtımız yüklendi ve kullanıma hazır. Peki bunu nasıl kullanacağız? Çok kolay.

Öncelikle `/dev/mtd0` ve `/dev/mtdblock0` aygıtlarının sistemde bulunup bulunmadığını kontrol edelim (devfs'li sistemlerde konum farklı olabilir). Eğer yoklarsa oluşturulmaları gerekir. Eğer karakter aygıtı lazım ise yapmanız gereken:

```
# mknod /dev/mtd0 c 90 0
```

Ve `mtdchar` modülünün yüklenmiş olduğundan emin olun (veya çekirdeğin gerektiği zamanda yükleyeceğinden).

Blok aygıtı için yapmanız gereken:

```
# mknod /dev/mtdblock0 b 31 0
```

Ve erişim için `mtdlock` modülüne ihtiyaç vardır.

Aygıt hazırlandı. Bu kadar. Hala sağ kalan var mı :) O zaman devam...

8. Peki Bununla Ne Yapabilirim?

Pek çok şey. Bu konu hakkında düşünürken bunu kullanmak için 2 yol buldum. Birincisi bu aygıtın üzerinde bir dosya sistemi yaratmak:

```
# mkfs.ext2 /dev/mtdblock0
```

Ve bunu bir yerlere bağlamak:

```
# mount -t ext2 /dev/mtdblock0 /mount/edilecek/konum
```

2. yol ise bunun üzerine bir takas alanı oluşturmak:

```
$ mkswap /dev/mtdblock0
Setting up swap space version 1, size = 12582912 bytes
$ swapon /dev/mtdblock0
```

Daha sonra farklı olasılıklar oluştu. Örneğin bunu X11 terminallerinde, ağ band genişliğini sınırlamak için kullanabilirsiniz. Açılış sırasında terminal çekirdeği ve sıkıştırılmış dosya sistemini yükler. Dosya Sistemi `mtdblock` aygıtı üzerine yerleştirebilir ve çekirdek buradan açılış yapabilir. Sadece konsol kullanan sunucu sistemlerinde yeni nesil 32MB GfX kartlardaki bellek çok güzel bir takas alanı olarak kullanılabilir (disklerdeki takastan daha hızlı bir şekilde).

Yeni görüşlere her zaman için açığız :)

Notlar

- a) Belge içinde dipnotlar ve dış bağlantılar varsa, bunlarla ilgili bilgiler bulundukları sayfanın sonunda dipnot olarak verilmeyip, hepsi toplu olarak burada listelenmiş olacaktır.
- b) Konsol görüntüsünü temsil eden sarı zeminli alanlarda metin genişliğine sığmayan satırların sığmayan kısmı `␣` karakteri kullanılarak bir alt satıra indirilmiştir. Sarı zeminli alanlarda `␣` karakteri ile başlayan satırlar bir önceki satırın devamı olarak ele alınmalıdır.

Bu dosya (vram-howto.pdf), belgenin XML biçiminin \TeX Live ve belgeler-xsl paketlerindeki araçlar kullanılarak PDF biçimine dönüştürülmesiyle elde edilmiştir.

14 Şubat 2007