

Xorg - nVidia i ATI

Położenie: (nie dotyczy)

© 3bird Projects 2011, <http://3bird.net>

Xorg korzysta w chwili obecnej z serwisu HAL (automatyczne wykrywanie sprzętu i ładowanie modułów graficznych). W takiej sytuacji, plik konfiguracyjny „xorg.conf” zawiera minimum treści (w niektórych sytuacjach może być nawet pusty). Należy po prostu mieć włączony serwis HAL: `rc-update add hald default`.

Xorg zapowiada, że w najbliższej przyszłości HAL zostanie zastąpiony przez *udev* i nie będzie już rozwijany.

nVidia

Xorg a nVidia

Domyślny sterownik jądra o nazwie „*nv*”, ma bardzo słabe osiągi (nie obsługuje np. GL, bo nVidia nie udostępniła, jako jedyna, swoich specyfikacji). Dlatego należy z niego zrezygnować (podczas kompilacji jądra), a w zamian zainstalować **binarne** sterowniki „*nvidia*”, które również funkcjonują w postaci modułu (do poprawnej instalacji oryginalnego sterownika *nVidia*, potrzebne są zainstalowane źródła kernela w `/usr/src/linux`). Obie wersje modułu **nie mogą** istnieć jednocześnie (!), gdyż powodują konflikt. Wyłączamy więc moduł „*nv*” w jądrze:

Opcje kernela / Device Drivers / Graphics Support:

```
< > nVidia Framebuffer Support
```

```
< > nVidia Riva Support
```

Należy także pamiętać, że posiadacze starszych kart graficznych (np. *nVidia GeForce2 400*) lub kernela o numerze mniejszym niż 2.6.10-r6, muszą koniecznie instalować sterowniki *nVidia* w wersji nie większej niż 96.43.xx. Nowsze sterowniki (np. w wersji 169.09-r1) nie będą działały poprawnie. Posiadacze takich kart (lub jąder) powinni zamaskować sobie nowsze pakiety *nVidia*:

```
# echo ">=media-video/nvidia-drivers-169.09-r1" >> /usr/portage/profiles/package.mask
```

Zamiast maskowania, można po prostu zainstalować żądany pakiet w taki sposób:

```
# emerge nvidia-drivers-96.43.05
```

Instalację sterowników binarnych należy powtórzyć po każdej zmianie kernela (ewentualnie po jego ponownej kompilacji).

System, przy starcie, musi ładować moduły:

- *intel-agp*
- *agpgart* (used by *nvidia*)

Nvidia a DRI i GL

Nvidia stosuje DRI i GL w jednym rozwiązaniu o nazwie GLX (*OpenGL Extension to the X Windows System*) w pakiecie binarnym (**nie należy** instalować pakietu *x11-drm* ani zaznaczać tej opcji w ustawieniach kernela). Nie należy także włączać modułu „*dri*” w konfiguracji xorg (jedynie moduł „*glx*”). Polega na wprowadzeniu funkcji *OpenGL* do listy poleceń przetwarzanych przez *X-Serwer*. Biblioteka Mesa tłumaczy komendy *OpenGL* na zrozumiałe dla *X-Serwera*. Zewnętrzne moduły *nVidia* komunikują się bezpośrednio z akceleratorem 3D (procesorem) karty graficznej, ale za pomocą protokołu X (routowalny). GLX umożliwia więc np. komunikację siecią terminala (*X-Klient*) ze zdalnym akceleratorem 3D (*X-Serwer*). Wystarczy więc mieć jeden drogi serwer z dobrą kartą graficzną i podpięte do niego terminale bez kart graficznych (dzisiaj ze względu na niskie ceny kart graficznych, jest to rzadko spotykane rozwiązanie).

```
#eselect opengl set nvidia (X-Serwer będzie używał GLX)
```

```
# env-update
```

```
# source /etc/profile
```

```
# etc-update
```

```
# modules-update
```

Należy także sprawdzić czy w `/etc/modules.devfs` znajduje się wpis:

```
Alias /dev/nvidia* nvidia (zamiast Alias /dev/nvidia* Nvdriver)
```

A także czy w `/etc/modules.conf` jest wpis:

```
Alias char-major-195 nvidia
```

Aby DRI było dostępne dla wszystkich użytkowników, należy dodać danego użytkownika do grupy „*video*”

(praktykowane w Gentoo). Jeśli to nie pomoże, można w Xorg włączyć następującą opcję:

```
Section "dri"
    Mode 0666
EndSection
```

Aby działał compiz, należy dać opcję *AddARGBGLXVisuals*. Jeśli zaś są problemy z rozdzielczością, należy wyłączyć opcję *DynamicTwinView*.

Brak /dev/nvidia

1. **# Nvmakedevices.sh**
2. W pliku */etc/conf.d/rc* wpisujemy: *RC_DEVICE_TARBALL="yes"*

ATI Radeon

Xorg a ATI (ATI Radeon Xpress RS482 5975, karta zintegrowana)

Najpierw określamy nazwę naszej karty graficznej w pliku */etc/make.conf*: *VIDEO_CARDS="radeon"* (gdy zamierzamy używać otwartego sterownika na zasadach GPL) lub *VIDEO_CARDS="fglrx"* (gdy zamierzamy używać binarnego sterownika ATI).

W pierwszym przypadku, sterownik „*radeon*” zostanie utworzony przy kompilacji pakietu *xorg-server* (nie należy instalować żadnych dodatkowych sterowników oprócz niektórych standardowych pakietów):

```
# emerge xorg-server xorg-x11 libdrm x11-drm mesa
```

```
# modules-update
```

```
#eselect opengl set xorg-x11 (klient xorg ma używać mesy jako sterownika opengl)
```

W drugim przypadku (sterownik *fglrx*) robimy:

```
# emerge ati-drivers ati-drivers-extra
```

```
# modules-update
```

```
#eselect opengl set ati (jako opengl ma być używane fglrx; w zasadzie tę komendę wykonuje już sterownik) lub pobieramy nowszy sterownik ze strony www.ati.com. Nie musimy przy tym instalować x11-drm ani mesa, gdyż sterownik binarny posiada już te funkcje w sobie.
```

Ilekoć kompilujemy kernel, należy powtórzyć czynność instalacji sterowników.

DRI w ATI

DRI (*Direct Rendering Infrastructure*) to bezpośredni rendering (akceleracja 3D), to korzystanie z procesora karty graficznej zamiast procesora głównego, odpowiednik *DirectX* z systemu Windows. Taka metoda akceleracji stosowana jest przez producentów wszystkich kart graficznych za wyjątkiem kart *nVidia* (która stosuje metodę GLX i która jako jedyna nie udostępnia specyfikacji swoich kart). Polega na tym, że przy przetwarzaniu grafiki korzystamy bezpośrednio z procesora karty graficznej (akceleracja sprzętowa), a nie procesora komputera (akceleracja programowa oparta o biblioteki Mesa), co znacznie przyspiesza operacje graficzne. Pakiet odpowiedzialny za DRI to *x11-drm* (w przypadku otwartego sterownika „*radeon*”) lub *ati-drivers* (w przypadku binarnego sterownika „*fglrx*”). W obu przypadkach **nie należy** zaznaczać w jądrze opcji DRM. Jeśli chcemy jednak użyć modułów z jądra (wraz ze sterownikiem „*radeon*”, nie instalujemy pakietu *x11-drm* (ani tymbardziej *ati-drivers*), lecz zaznaczamy w jądrze następujące opcje:

```
Device Drivers --->
```

```
Character devices --->
```

```
<M> Direct Rendering Manager (XFree86 4.1.0 and higher DRI support)
```

```
<M> ATI Radeon
```

Następnie emergujemy:

```
# emerge xorg-x11 (bez flagi „radeon”, gdyż moduł do ATI Radeon jest już zaznaczony w jądrze)
```

Ponad to, należy w konfiguracji *X Window* włączyć opcję:

```
Section "Module"
```

```
Option dri
```

```
EndSection
```

Aby DRI było dostępne dla wszystkich użytkowników, należy w Xorg włączyć następującą opcję:

```
Section "dri"
```

```
Mode 0666
```

EndSection

Należy pamiętać, iż w chwili obecnej, DRI zostanie automatycznie wyłączone, gdy w pliku *xorg.conf* uaktywnimy rozszerzenie „Composite” lub moduł „Xinerama”. Gdy korzystamy ze sterownika „radeon”, moduł „drm” musi zostać załadowany przed startem *X-Serwera*.

Należy zwrócić uwagę, że w mechanizmie DRI, zrezygnowano z pośrednictwa protokołu komunikacyjnego *X-Window*, a polecenia *OpenGL* kierowane są bezpośrednio do akceleratora 3D. Z tego powodu DRI można wykorzystać tylko do obsługi lokalnej karty graficznej (w przeciwieństwie do GLX). Przy próbie korzystania ze zdalnego *X-Serwera*, włączona zostanie lokalna akceleracja programowa (czyli wykorzystanie głównego procesora).

Biblioteki OpenGL

Mesa - biblioteka graficzna zgodna z *OpenGL* (klon *OpenGL*). Na jej podstawie powstał sterownik do kart *Voodoo* zwany *Glide*.

GLCore - biblioteka *OpenGL*, która umożliwia akcelerację programową (gdy sprzętowa jest niemożliwa).

Framebuffer

Framebuffer to pośrednik między *X-Serwerem* a kartą video. Umożliwia ustawienie różnej rozdzielczości w trybie tekstowym i różną ilość kolorów. Do poprawnego działania framebuffera należy na stałe wkompiłować w jądro:

[*] *VESA VGA Graphics Support* (typ *vesafb*; nie działa moduł *rivaafb*!)

fbset -a (status framebuffera)

fbset 1024x768-72 (ustawienie rozdzielczości i odświeżania)

Należy zrezygnować z wyboru innych sterowników do framebuffera (np. *Nvidia* lub *ATI* jako framebuffer).

Konfiguracja Xorg

Główny plik konfiguracyjny to */etc/X11/xorg.conf*. Domyślnie jednak nie istnieje, należy go najpierw utworzyć, na przykład za pomocą narzędzi:

xorgconfig (seria pytań)

lub

xorgcfg (tryb graficzny)

lub

xorgcfg -textmode (zalecane)

Najlepiej jednak utworzyć plik konfiguracyjny ręcznie w edytorze tekstu. Utworzony plik można testować za pomocą polecenia:

startx -config /root/xorg.conf.new

Podczas uruchamiania X-ów odczytywane są konfiguracje w następującej kolejności:

1. *~/xinitrc* (można np. wpisać tam: “*exec icewm*”).
2. Odczytuje zawartość zmiennej *XSESSION* i uruchamia sesję z */etc/X11/Sessions*.
3. Inaczej uruchamia TWM.

Dla kart *nVidia* zalecane jest wyłączyć wczytywanie modułu DRI (gdyż korzysta już z GLX), paradoksalnie przyspieszy to działanie grafiki (u mnie: z 1654fps do 1698fps).

Uruchamianie X-serwera

x (uruchamia sam *X-Serwer*)

startx (uruchamia *X-Serwer* z menadżerem okien)

Zmniejszenie rozdzielczości w locie (klawiatura numeryczna!): *Ctrl+Alt+(+)* lub *Ctrl+Alt+(-)*.

Zakończenie pracy X-serwera: *Ctrl+Alt+Backspace* (jeśli konfiguracja xorg na to pozwala). Inaczej:

init 3

Akceleracja 2D

1. W BIOS zaznaczamy opcję: *MTRR = discrete* (a nie *continuous*)
2. **# cat /proc/mtrr** (powinno być “*write-back*” oraz “*write-combining*”)

3. Akcelerację 2D obsługuje w xorg moduł XAA (*X Acceleration Architecture*) ładowany za pomocą opcji „**No-Accel**” „no”.

Xinerama

Przełączanie monitorów w locie:

```
# xvattr -a XV_SWITCHCRT -v 0 (LCD)
# xvattr -a XV_SWITCHCRT -v 1 (CRT)
```

Przezroczystość okien

Section "Extensions"

Option "Composite" "true"

EndSection

Aby sprawdzić, czy po restarcie X, rozszerzenie zostało załadowane, należy wydać polecenie:

```
# xdpinfo | grep Composite
```

Dla posiadaczy kart *nVidia* należy ustawić tę opcję:

```
Option "AllowGLXWithComposite" "On" (w sekcji karty graficznej)
```

Nie wszystkie jednak *Windows Managers* obsługują natywnie tę przezroczystość. Jeśli nie obsługują, należy zainstalować:

```
# emerge xcompmgr transset
```

```
# xcompmgr -c (obsługa cieni pod oknami)
```

```
# xcompmgr -cCfF -r7 -o.65 -l-10 -t-8 -D7 & (przykład cieni)
```

```
# transset (ustawienie przezroczystości)
```

```
# transset .2 (stopień przezroczystości: 0=pełna, 1=brak)
```

TouchPad a Firefox

Aby przesuwanie palcem po touchpadzie nie skutkowało w przeglądarce zmiany strony, należy włączyć tę opcję w FireFox:

```
about:config
```

```
mousewheel.horizscroll.withnokey.action ---> 0
```

```
mousewheel.horizscroll.withnokey.sysnumlines ---> true
```

Diagnozowanie

1. `# /usr/bin/nvidia-bug-report.sh` (tworzy w katalogu domowym plik z raportem)
2. `# glxinfo` (czy działa GL? ---> "*Direct rendering: Yes*")
3. `# /sbin/modinfo nvidia` (informacja o wybranym module)
4. `# cat /var/log/Xorg.0.log` (wyświetla logi)
5. `# glxgears` (podaje szybkość GL, czyli ilość ramek na sekundę, np. 1698Fps)
6. `# fgl_glxgears` (szybkość GL w kartach ATI; moja prędkość: 214.000 fps)
7. `# xrandr` (podaje możliwe rozdzielczości, aktualną rozdzielczość oraz odświeżanie)
8. `# cat /proc/driver/nvidia/agp/*` (parametry AGP)

Ostatnia aktualizacja: 1 styczeń 2011.