

Xorg - informacje

Położenie: (nie dotyczy)

© 3bird Projects 2023, <http://edukacja.3bird.pl>

Xorg korzysta w chwili obecnej z serwisu *udev* (automatyczne wykrywanie sprzętu i ładowanie modułów graficznych). W takiej sytuacji, plik konfiguracyjny „*xorg.conf*” zawiera minimum treści lub nie jest potrzebny w ogóle.

Konfiguracja Xorg

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

nvidia-xconfig (*automatyczna*)

Xorg :1 -configure

Definicja domyślnego WM → *~/.dwmrc*

Definicja domyślnego menedżera logowania → */etc/conf.d/display-manager*

Instalacja pakietów

Należy zainstalować:

- *xf86-video-intel* (dla kart Intel, w tym iris)
- *nvidia-drivers* (z flagami: *X*, *acpi*, *gtk3*, *tools* [zawiera *nvidia-settings*], *uvm* [*Unified Memory* → udostępnianie pamięci między CPU a GPU w programach CUDA]),
- *xrandr*,
- *xf86-input-evdev*,
- *xf86-input-synaptics*,
- *ibus*,
- *xsm*,
- *xorg-server*,
- *display-manager*.

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

Można także uruchomić dwa X-serwery: **startx -- :8** (uruchamia go na terminalu 8).

Procedura uruchamiania X

Polecenie „*startx*” jest tylko *front-endem* do *xinit*, które uruchamia *Xserver* i podłącza do niego klientów X. Po wydaniu komendy *startx* wyszukiwany jest plik:

- *~/.xinitrc* (zawiera definicję „*desktop enviroment*”); jeśli go nie ma, wyszukiwany jest:
- */etc/X11/xinit/xinitrc*; on z kolei wyszukuje:
 - *~/.Xresources* (zawiera wartości specyficzne dla aplikacji, np. DPI, antialiasing, fonty, kursory, screensavery, itp.)
 - *~/.Xmodmap* (konfiguracja klawiatury)

- `~/.Xkbmap` (konfiguracja klawiatury)

a jeśli tych plików nie ma, to korzysta z:

- `/etc/X11/Xresources`
- `/etc/X11/Xmodmap`
- `/etc/X11/Xkbmap`

i następnie uruchamia wszystko, co jest w

- `/etc/X11/xinit/xinitrc.d/*`
- `/etc/X11/xinit/xinput.d/*`

a potem uruchamia

- `~/.Xclients` (zdefiniowane środowisko graficzne)
- `/etc/X11/xinit/Xclients` (uruchamia zastępcze środowisko graficzne: *twm*, *gnome*, *kde*, itp.)

Uwaga: Aby ponownie wczytać zmieniony wygląd środowiska graficznego (bez wylogowywania), należy wydać polecenie:

```
# xrdp -load ~/.Xresources
```

Autoryzacja w X

Istnieją 3 metody autoryzacji klientów X:

- oparta na **liście hostów** mających prawo do logowania się na X serwerze;
- oparta o **magiczne cookies** zapisywane przez klientów w `~/.Xauthority`; tu stosuje się dwa sposoby tworzenia cookies:
 - z użyciem MIT-MAGIC-COOKIES (wysyłane jako zwykły tekst);
 - z użyciem XDM-AUTHORIZATION (wysyłane jako dane szyfrowane za pomocą DES);
- oparte o **listę użytkowników** mających prawo logowania się na X serwerze; program `xhost` może utworzyć `localuser / localgroup` z takimi użytkownikami.

nVidia

Xorg a nVidia

Domyślny sterownik jądra o nazwie `"nv"` (*nouveau*), ma bardzo słabe osiągi (nie obsługuje np. GL, bo nVidia nie udostępnia, 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*) 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
```

Niestety, od jądra wersji 5.15*, stare sterowniki nie będą już współpracować (np. wersja 340.108 do karty GeForce G218/G210). W tym wypadku, nie pozostaje nic innego niż korzystanie z modułu

„nv” (sterownik nouveau znajdujący się w jądrze). Dodatkowo należy wymusić ładowanie modułu „nv” w czasie startu środowiska X:

```
# nano /etc/X11/xorg.conf.d/50-device.conf
```

```
Section "Device"
```

```
    Identifier "Stara Nvidia"
```

```
    Driver "nouveau"
```

```
EndSection
```

Należy także upewnić się, że „nv” / „nouveau” nie jest na czarnej liście modułów (*/etc/modprobe.d/blacklist.conf*) oraz że nie jest blokowane przy starcie w grubie (*/etc/default/grub.conf*), gdzie nie powinno być opcji *modetest*. Należy także w tym wypadku odinstalować wszystkie pakiety mające w nazwie „nvidia”, a jednocześnie doinstalować pakiety mające w nazwie „Mesa”, „libdrm”, „dri”. Należy także w GRUB uaktywnić opcję *acpi=on*. W niektórych przypadkach dobrze jest odinstalować „drm-kmp-default”.

Jeśli sterownik „nouveau” nie działa poprawnie, należy go odinstalować i wypróbować „modesetting” jako alternatywę.

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

Jeśli mamy stare złącze AGP, system - przy starcie - musi ładować moduły:

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

Uwaga: Istnieje graficzny konfigurator *nvidia* (instalowany jako osobny pakiet):

```
# nvidia-settings
```

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 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 akcelatorem 3D (procesorem) karty graficznej, ale za pomocą protokołu *X* (routowalny). *GLX* umożliwia więc np. komunikację sieciową terminala (*X-Klient*) ze zdalnym akcelatorem 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
```

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*.

Nvidia Optimus

Zintegrowana karta *Intel* ma bezpośrednie połączenie z monitorem i to ona wysyła do niego dane. W razie potrzeby uruchamia *nVidię*, a ona zwraca wyrenderowane dane *Intelowi*.

Istnieje kilka rozwiązań w *Linuksie* naśladowujących działanie *Optimus*:

- sterowniki **noveau** (wady: niska wydajność; nie testowany przez mnie);
- aplikacja **bumblebee** („trzmieł”), która potrafi uruchamiać wybrane aplikacje na *nVidia* za pomocą polecenia „*optirun nazwaAplikacji*” (użytkownik powinien być w grupie „*bumblebee*”); wady: niższa wydajność, własna biblioteka *openGL*, zagnieżdżone środowisko X; nie testowane);
- moduł jądra „**VGA Switcheroo**” (działa tylko z laptopem mającym fizyczny sprzętowy przełącznik [*Hybrid Graphics*] i jest zbędny w przypadku technologii *Optimus*);
- **dwie konfiguracje X** (jedna aktywująca tylko *Intel+Mesa*, oraz druga aktywująca *Intel+nVidia*), przełączamy się między nimi za pomocą skryptu (wymaga wylogowania się i ponownego zalogowania).

W każdym przypadku, aby polecenie *startx* działało w technologii *Optimus*, należy najpierw uruchomić *xrandr*. Najlepiej polecenie umieścić w skrypcie */etc/X11/xinit/xinitrc.d/94-xrandr*:

```
#!/bin/sh
```

```
xrandr --setprovideroutputsource modesetting NVIDIA-0
```

```
xrandr -auto
```

Aby zobaczyć jakich mamy „providerów”:

```
# xrandr --listproviders -d :0
```

Aby działało także uruchamianie menedżerów logowania, należy to polecenie umieścić w:

```
GDM → /etc/X11/gdm/Init/Default
```

```
KDM → /usr/share/config/kdm/Xsetup
```

```
LightDM → /etc/lightdm/lightdm.conf (display-setup-script=/etc/X11/xinit/xinitrc.d/94-xrandr)
```

```
SLIM → (nie udało się tego dokonać).
```

Jeśli *xrandr* nie zostanie uruchomiony przed startem środowiska graficznego, pojawi się komunikat błędu:

```
Fatal server error:
```

```
(EE) no screens found
```

W kernerze uaktywnić:

```
Device Support / Graphics Support →
```

```
<*> Intel 8xx/9xx
```

```
[*] Enable modesetting on Intel by default
```

Jak sprawdzić, która karta pracuje?

```
# nvidia-settings
```

```
GPU 0 - GeForce 820M / GPU Utilization →
```

```
0% - gdy tylko Intel pracuje
```

```
93% - gdy nVidia renderuje
```

```
# glxinfo
```

```
(gdy renderuje Intel)
```

```
server glx vendor string: SGI
```

```
client glx vendor string: Mesa Project and SGI
```

```
(gdy renderuje nVidia)
```

```
server glx vendor string: NVIDIA Corporation
```

```
client glx vendor string: NVIDIA Corporation
```

(jeśli rendering w ogóle nie działa, należy ponownie zainstalować nvidia-drivers lub mesa i zrestartować X)

glxgears

Intel: ~5664 frames

nVidia: ~32522 frames

Pliki konfiguracyjne asus-i3 (Optimus)

Opcja „modesetting+nVidia”

Zawartość `/etc/X11/optimus/94-xrandr-nvidia`:

```
#!/bin/sh
xrandr --setprovideroutputsource modesetting NVIDIA-0
xrandr -auto
```

Skrypt `./go-nvidia`:

```
#!/bin/sh
eselect opencl set 2
eselect opengl set 1
cp /etc/X11/optimus/xorg-optimus.conf /etc/X11/xorg.conf
cp /etc/X11/optimus/94-xrandr-nvidia /etc/X11/xinit/xinitrc.d/94-xrandr
/etc/init.d/udev restart
```

Plik `/etc/X11/optimus/xorg-optimus.conf`:

```
Section "ServerLayout"
    Identifier "Main Layout"
    Screen 0 "nVidiaScreen" # Moze byc tylko jeden "Screen" w tym rozwiazaniu
    Inactive "Intel" # Dotyczy karty graficznej, a nie "Screen"
EndSection

Section "Files"
    ModulePath "/usr/lib64/xorg/modules,/usr/lib64/opengl/xorg-x11/extensions" # Musi
    byc dla dri i dri2
    FontPath "/usr/share/fonts/100dpi"
    FontPath "/usr/share/fonts/75dpi"
    FontPath "/usr/share/fonts/Type1"
    FontPath "/usr/share/fonts/TTF"
    FontPath "/usr/share/fonts/dejavu"
    FontPath "/usr/share/fonts/freefonts"
    FontPath "/usr/share/fonts/liberation-fonts"
    FontPath "/usr/share/fonts/libertine"
    FontPath "/usr/share/fonts/misc"
    FontPath "/usr/share/fonts/urw-fonts"
EndSection

Section "Module"
    Load "synaptics"
    Load "type1"
```

```

Load      "freetype"
# Double Buffer Extension, useful for animation and video operations; modul juz wbudowany w
# sterownik modesetting i nVidia:
# Load      "dbe"
# Obsluga GL. Czasami nalezy wpisac cala sciezke dostepu, np.:
# /usr/X11R6/lib/modules/extensions/libglx.so.1.0.61111:
# Load      "glx"
# Umozliwia rozszerzenia, np. shaped windows, shared memory, video mode switching, DGA, and
# Xv; modul juz wbudowany w sterownik modesetting:
# Load      "extmod"
# Load      "record"
# Direct Rendering w kartach ATI (ponoc nie uruchamiamy tego, gdy mamy nVidia),
# /usr/lib64/opengl/xorg-x11/extensions/libdri.so (modul juz wbudowany w sterownik
# modesetting):
# Load      "dri"
# /usr/lib64/opengl/xorg-x11/extensions/libdri2.so (modul juz wbudowany w sterownik
# modesetting oraz nVidia):
# Load      "dri2"
# Load      "speedo"
# Load      "drm"
# Load      "xtt"
# Load      "pcidata"
#SubSection "extmod"                # laduje moduly, m. in. SHAPE
# Option    "omit xfree86-dga"      # Omija uruchamianie DGA extension (Direct
# Graphics Access)
#EndSubSection
EndSection

```

Section "ServerFlags"

Aby opcja "Xinerama" dzialala musi byc globalnie ustawiona flaga "xinerama". Aplikacje instalowane sa wtedy z rozszerzeniem "xinerama" (tworzone wtedy sa zmienne DISPLAY=:0.0 DISPLAY=:0.1) oraz protokolem "xineramaproto" (z protokolu korzysta xrandr oraz twinview, ale sam protokol tworzy jedynie DISPLAY=:0).

Xinerama umozliwia stworzenie rozszerzonego monitora (ale wylacza DRI):

```

Option     "Xinerama"                "0"
Option     "AIGLX"                   "off"
Option     "AllowEmptyInput"         "off"      # Czy pozwolic na start X bez
# deklaracji myszki i klawiatury?
Option     "AutoAddDevices"          "true"     # Wlacza / Wylacza wykrywanie
# urzadzen (za pomoca udev), tzw.
# hotplugging.
Option     "AutoEnableDevices"       "true"     # Uaktywnia dodane urzadzenia
Option     "AutoAddGPU"              "true"     # Serwer wykrywa dodatkowe (inne
# niz podstawowa) karty graficzne.
# Gdy chcemy separacji kart (do
# roznych profili), dajemy "off".
Option     "BlankTime"               "45"      # Wlacza wygaszacz po 45
# minutach
# Option   "standby time"             "30"
# Option   "suspend time"            "20"

```

```

# Option "off time" "60" # Wylacza komputer po 60
# # minutach bezczynnosc
# Option "AllowMouseOpenFail" # Umozliwia otwarcie X bez obslugi
# # myszki; opcja nie dotyczy
# # touchpada.
# Option "DisableModInDev" # Blokuje automatyczna zmiane
# # myszy i klawiatury
# Option "DontZap" # Blokuje zamykanie X przez
# # <Ctrl><Alt><BackSpace>
# Option "DontVTSwitch" # Blokuje przelaczenie na inne
# # terminale.

```

EndSection

Section "Extensions"

Efekty OpenGL (np. cienie, przezroczystosc); musi byc wylaczone, aby dzialalo DRI w fglrx;
zawarte w nVidia.

```
Option "Composite" "Enable"
```

EndSection

Typy monitorow: VGA (podlaczony do portu VGA, inaczej zwany CRT), DVI (podlaczany do portu DVI), DFP lub LVDS (ekran laptopa), HDMI (podlaczony do portu HDMI). W wielu przypadkach nazwy tych portow moga byc inne. Nalezy zawsze upewnic sie wydajac polecenie "xrandr -q". Modeline ustala sie przez polecenie "xrandr -q" wydane w aktywnym srodowisku graficznym. Format: "<aktywna-rozdzielczosc>" <pclk> <mozliwa-rozdzielczosc x> <mozliwa-rozdzielczosc y>". Wartosc pclk mozna uzyskac w "Kalkulator Modeline" (Internet) lub za pomoca polecenia "cvt 1366 768 60".

Section "Monitor"

```
Identifier "Monitor0"
```

```
ModelName "DFP"
```

Czyli monitor laptopa

```
Option "DPMS" "true"
```

Display Power Managment

System

```
Modeline "1366x768_60.00" 85.25 1366 1440 1576 1784 768 771 781 798 -hsync
```

+vsync

EndSection

Ponizsze dwa monitory przetestowac, gdy tylko rozwiaze problem z Optimusem:

#Section "Monitor"

```
# Identifier "Monitor1"
```

```
# ModelName "VGA-0"
```

```
# Option "RightOf" "Monitor0"
```

Wyszwietlanie obrazu na

wielu monitorach (inne

opcje: LeftOf, Above,

Below)

Ponizsze to tylko przyklad. Nalezy sprawdzic za kazdym razem poleceniem "xrandr -q".

```
# Modeline "1024x768_60.00"
```

```
# Modeline "1366x768_60.00"
```

#EndSection

#Section "Monitor"

```
# Identifier "Monitor2"
```

```

# ModelName "HDMI-0"
# Option "RightOf" "Monitor1" # Wswietlanie obrazu na
# wielu monitorach (inne
# opcje: LeftOf, Above,
# Below)

# Poniższe to tylko przykład. Należy sprawdzić poleceniem "xrandr -q".
# Modeline "1366x768_60.00" 85.25 1366 1440 1576 1784 768 771 781 798 -hsync
+vsync
#EndSection

Section "Device"
    Identifier "nVidia"
    VendorName "nVidia Corporation"
    Driver "nvidia" # Inne: radeon, fglrx (radeon
# umozliwia GL, ale nie DRI;
# wiec zamiast niego uzywac
# fglrx)
    BusID "PCI:4:0:0" # Identyfikacja zlacza karty
# graficznej (gdy sie ma np.
# dwie karty). Ustalane za
# pomoca: lspci.
    Option "DPI" "96 x 96"
EndSection

Section "DRI"
    Mode 0666 # Wszyscy uzytkownicy maja
# dostep do DRI
EndSection

Section "Device"
    Identifier "Intel"
    Driver "modesetting" # Sterownik do kart
# Intel/nVidia Optimus.
    BusID "PCI:0:2:0" # Identyfikacja zlacza karty
# graficznej. Ustalane za
# pomoca: lspci.

# Poniższe opcje nie maja sensu w tej konfiguracji.
# Option "Monitor-eDP-0" "Monitor0" # Uaktywnia wswietlanie obrazu
# na wielu monitorach. Format:
# "Monitor-<port>".
# Option "Monitor-DisplayPort-0" "Monitor1" # Uaktywnia wswietlanie obrazu
# na wielu monitorach. Format:
# "Monitor-<port>".
# Option "Monitor-HDMI-0" "Monitor2" # Uaktywnia wswietlanie obrazu
# na wielu monitorach. Format:
# "Monitor-<port>".

# Port jest okreslany poprzez wydanie polecenia w aktywnym srodowisku graficznym: xrandr -q.
EndSection

```



```

Section "Screen"
    Identifier "nVidiaScreen"
    Device "nVidia"
    Monitor "Monitor0"
    SubSection "Display"
        Depth 24
        Modes "1366x768" "1024x768" "800x600"
        Virtual 1366 768
    EndSubSection
EndSection

```

```

Section "Screen"
# Nie trzeba definiowac opcji "Modes" ani subsekcji "Display", gdy uzywamy sterownika
# "modesetting".
    Identifier "IntelScreen"
    Device "Intel"
    Monitor "Monitor0"
EndSection

```

Opcja „Intel Only”

Plik [/etc/X11/optimus/94-xrandr-intel](#):

```

#!/bin/sh
xrandr -auto

```

Skrypt [./go-intel](#):

```

#!/bin/sh
eselect opencl set 1
eselect opengl set 2
cp /etc/X11/optimus/xorg-intel-only.conf /etc/X11/xorg.conf
cp /etc/X11/optimus/94-xrandr-intel /etc/X11/xinit/xinitrc.d/94-xrandr
/etc/init.d/udev restart

```

Plik [/etc/X11/optimus/xorg-intel-only.conf](#):

```

Section "ServerLayout"
    Identifier "Main Layout"
    Screen 0 "IntelScreen"
EndSection

```

Section "Files"

```

ModulePath "/usr/lib64/xorg/modules,/usr/lib64/opengl/xorg-x11/extensions"
# Musi być dla dri i dri2
FontPath "/usr/share/fonts/100dpi"
FontPath "/usr/share/fonts/75dpi"
FontPath "/usr/share/fonts/Type1"
FontPath "/usr/share/fonts/TTF"
FontPath "/usr/share/fonts/dejavu"
FontPath "/usr/share/fonts/freefonts"
FontPath "/usr/share/fonts/liberation-fonts"
FontPath "/usr/share/fonts/libertine"
FontPath "/usr/share/fonts/misc"

```

```
FontPath "/usr/share/fonts/urw-fonts"
EndSection
```

```
Section "Module"
    Load      "synaptics"
    Load      "type1"
    Load      "freetype"
EndSection
```

```
Section "ServerFlags"
    Option     "Xinerama"           "0"
    Option     "AIGLX"              "off"
    Option     "AllowEmptyInput"   "off"      # Czy pozwolac na start X bez
                                                # deklaracji myszki i klawiatury?
    Option     "AutoAddDevices"    "true"     # Wlacza / Wylacza wykrywanie
                                                # urzadzen (za pomoca udev), tzw.
                                                # hotplugging.
    Option     "AutoEnableDevices" "true"     # Uaktywnia dodane urzadzenia
    Option     "AutoAddGPU"        "true"     # Serwer wykrywa dodatkowe (inne
                                                # niz podstawowa) karty graficzne.
                                                # Gdy chcemy separacji kart (do
                                                # roznych profili), dajemy "off".
    Option     "BlankTime"         "45"      # Wlacza wygaszacz po 45
                                                # minutach
EndSection
```

```
Section "Extensions"
# Efekty OpenGL (np. cienie, przezroczystosc); musi byc wylaczone, aby dzialalo DRI w fglrx;
zawarte w nVidia.
```

```
    Option     "Composite"         "Enable"
EndSection
```

```
Section "Monitor"
    Identifier   "Monitor0"
    ModelName    "eDP1"           # Czyli monitor laptopa
    Option       "dpms"          # Display Power Managment
                                # System
    Modeline     "1366x768_60.00" 85.25 1366 1440 1576 1784 768 771 781 798 -hsync
+vsync
EndSection
```

```
# Projektor w STO:
```

```
Section "Monitor"
    Identifier   "Monitor1"
    ModelName    "DP1"
    Option       "LeftOf"         "Monitor0"
    # Wyszwietlanie obrazu na wielu monitorach (inne opcje: LeftOf, Above, Below). Ponizsze
    # nalezy sprawdzic za kazdym razem poleceniem "xrandr -q" oraz na kalkulatorze online:
```

```
    Modeline      "1366x768_60.00" 85.25 1366 1440 1576 1784 768 771 781 798 -hsync
+vsync
EndSection
```

```
# Projektor w TEB:
```

```
Section "Monitor"
```

```
    Identifier     "Monitor2"
```

```
    ModelName      "HDMI1"
```

```
    Option         "RightOf"          "Monitor0"
```

```
# Wyszwietlanie obrazu na wielu monitorach (inne opcje: LeftOf, Above, Below). Ponizsze
```

```
# nalezy sprawdzic poleceniem "xrandr -q" i wyliczyc na kalkulatorze online:
```

```
# Modeline      "1366x768_60.00" 85.25 1366 1440 1576 1784 768 771 781 798 -hsync
```

```
# +vsync
```

```
# Modeline      "1024x768_60.00" 64.11 1024 1080 1184 1344 768 769 772 795 -hsync
```

```
# +vsync
```

```
EndSection
```

```
Section "Device"
```

```
    Identifier     "nVidia"
```

```
    VendorName     "nVidia Corporation"
```

```
    Driver         "nvidia"
```

```
# Inne: radeon, fglrx (radeon umozliwia
```

```
# GL, ale nie DRI; wiec zamiast niego
```

```
# uzywac fglrx)
```

```
    BusID          "PCI:4:0:0"
```

```
# Identyfikacja zlacza karty graficznej
```

```
# (gdy sie ma np. dwie karty). Ustalane
```

```
# za pomoca: lspci.
```

```
    Option         "UseDisplayDevice" "none"
```

```
# Karta nVidii przekazuje zrenderowany
```

```
# obraz nie do monitora, a do karty Intel
```

```
# (ona do monitora).
```

```
    Option         "NoLogo"          "false"
```

```
# Wylacza logo nVidii
```

```
EndSection
```

```
Section "DRI"
```

```
    Mode           0666
```

```
# Wszyscy uzytkownicy maja dostep do
```

```
# DRI
```

```
EndSection
```

```
Section "Device"
```

```
    Identifier     "Intel"
```

```
    Driver         "intel"
```

```
# Sterownik do kart Intel/nVidia Optimus.
```

```
    BusID          "PCI:0:2:0"
```

```
# Identyfikacja zlacza karty graficznej.
```

```
# Ustalane za pomoca: lspci.
```

```
# Aktywuje kontrole jasnosci ekranu:
```

```
    Option         "Backlight"       "intel_backlight"
```

```
# Uaktywnia wyswietlanie na wielu monitorach. Sposob wyswietlania (klonowanie lub
```

```
# rozciaganie) definiuje sie w sekcji Monitor. Nazwy monitorow sprawdzamy komenda
```

```
# "xrandr -q" w aktywnym srodowisku X. Format: "Monitor-<port>".
```

```
    Option         "Monitor-eDP1"    "Monitor0"
```

```
    Option         "Monitor-DP1"     "Monitor1"
```

```

Option          "Monitor-HDMI1"      "Monitor2"
EndSection

Section "Screen"
Identifier      "IntelScreen"
Device         "Intel"
Monitor        "Monitor0"
EndSection

```

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 libdrm xorg-drivers mesa
```

```
#eselect opengl set xorg-server
```

(klient xorg ma używać mesy jako sterownika opengl)

W drugim przypadku (sterownik *fglrx*) robimy:

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

```
# 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.

Ileokroć 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 tymbardziejziej *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-server
```

(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 akcelatora 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).

Intel Iris XE

W wielu kartach Intel do akceleracji 3D (openGL) wykorzystywany jest moduł „i965_dri.so”. W kartach Iris powinien być wykorzystany moduł „iris” z pakietu *xf86-video-intel*. W przypadku, gdy w logach Xorg pojawia się informacja o braku modułu *i965_dri*, należy w pliku */etc/X11/xorg.conf* umieścić:

```
Section "Device"
```

```
Identifier "Intel Graphics"
```

```
Driver "intel"
```

```
Option "AccelMethod" "sna"
```

```
# Option "TearFree" "true" # tylko, gdy powyżej jest SNA
```

```
Option "DRI" "iris"
```

```
# Option "DRI" "3" # wersja 3 DRI
```

```
EndSection
```

Biblioteki OpenGL

Mesa - biblioteka graficzna zgodna z *OpenGL* (klon *OpenGL*). Na jej podstawie powstał sterownik do kart *Voodoo* zwany *Glide*. Istnieją dwa sterowniki Mesa (dwaj prowadzący): **classic** (zintegrowane karty *Intel i915/i965*) oraz **gallium** (*radeon, nvidia*; technologie *vaapi, vdpau, llvm*). W przypadku posiadania karty Intel trzeba wybrać tylko jednego prowadzącego: *classic*.

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

Czcionki

Istnieją dwa standardy czcionek:

- **corefont** - ich nazwy można uzyskać wydając polecenie „*cat /usr/share/fonts/misc/fonts.dir*”;
- **xft**

```
# eselect fontconfig list (pokazuje czcionki i ich opcje)
```

- **anti-aliasing** - wygładzanie;
- **hinting** - uzupełnianie w locie pikseli w czcionkach na ekranach z dużą rozdzielczością;
- **sub-pixel rendering** - w matrycach LCD każdy piksel składa się z trzech kolorów podstawowych w celu zwiększenia rozdzielczości ekranu, co powoduje problemy z kolorem czarnym i kontrastem; rendering próbuje to naprawić, ale litery mogą być bardziej zamazane (każdy musi sam przetestować swoje preferencje w tym temacie);
- **lcdfilter** - predefiniowane ustawienia specjalnie dla monitorów LCD;

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*; jeśli mamy EFI, to tylko *Simple Framebuffer* wystarczy)

fbset -a (status framebuffera)

fbset 1024x768-72 (ustawienie rozdzielczości i odświeżania)

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

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 „**NoAccel**” „**no**”.

Przeźroczystość okien

Section "Extensions"

Option "Composite" "true"

EndSection

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

xdpyinfo | 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 przeźroczystość. 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 przeźroczystości)

transset .2 (stopień przeźroczystości: 0=pełna, 1=brak)

Zmiana rozdzielczości w locie

xrandr -q (wyświetla nazwy monitorów)

xrandr --output eDP-1-0 --mode 1024x768 --rate 60.04

xrandr --output eDP-1-0 --auto (powrót do ustawień domyślnych)

Projektor

Projektor należy podpiąć do monitora w czasie startu systemu (nie wykrywa dynamicznie podłączenia do portu VGA). Następnie należy dostosować do siebie rozdzielczość monitora oraz projektora (większość projektorów nie osiąga rozdzielczości powyżej 1024x768).

Sposób pierwszy: należy więc na zwykłym koncie użytkownika (koniecznie w trybie graficznym) wydać polecenie:

xrandr -q

Polecenie to jest częścią rozszerzenia xorg (pakiet „xrandr”) i wyświetla możliwe rozdzielczości oraz podpięte monitory. Następnie należy podać jako parametr numer linii z żadaną rozdzielczością, (przy czym linie liczone są od 0) np.:

```
# xrandr -s 0 (dla rozdzielczości 1366x768)
```

```
# xrandr -s 1 (dla rozdzielczości 1024x768)
```

```
# xrandr -s 3 (dla rozdzielczości 800x600)
```

Aby włączyć klonowanie obrazu na projektor (ekran laptopa nosi nazwę „eDP-1-0”):

```
# xrandr --output eDP-1-0 --mode 1366x768 --output HDMI-1-0 --mode 1366x768 --same-as eDP-1-0
```

```
# Pulpit rozszerzony:
```

```
# xrandr --output eDP-1-0 --mode 1366x768 --output HDMI-1-0 --mode 1366x768 --right-of eDP-1-0
```

Można spróbować również klonować mając dwie różne rozdzielczości, inną na monitorze laptopa, inną na projektorze:

```
# xrandr --fb 1366x768 --output eDP-1-0 --mode 1366x768 --panning 1366x0 („panning” to „panoramowanie”, tj. można przesuwać pulpit myszką).
```

Ustawienie najlepszej rozdzielczości podpiętego projektora:

```
# xrandr --output eDP-1-0 --auto
```

Aby wyłączyć monitor na porcie VGA:

```
# xrandr --output eDP-1-0 -off
```

Sposób drugi: w pliku *xorg.conf* należy dodać opcje do sekcji „Screen”):

```
Option "nvidiaXineramaInfoOrder" "DFP-0"
```

```
Option "metamodes" "VGA-0: nvidia-auto-select +0+0, LVDS-0: nvidia-auto-select +0+0"
```

W starszej wersji xorg było to:

```
Option "TwinView" "1"
```

```
Option "TwinViewXineramaInfoOrder" "DFP-0"
```

```
Option "metamodes" "DFP:1024x768 +0+0, CRT:1024x768 +0+0"
```

Monitor DFP to wbudowany monitor laptopa, natomiast CRT to projektor.

Trzeci sposób: można także przełączyć na projektor w locie za pomocą polecenia (na zwykłym koncie użytkownika):

```
# nvidia-settings (X Server Display Configuration → Clones)
```

Opcję „Clones” należy ustawić tylko w sekcji głównego monitora AUO, który klonuje obraz na projektor.

Konfiguracja touchpada

Większość obecnych touchpadów obsługiwanych jest przez sterowniki *DesignWare* (biblioteka wysokiego poziomu dla kodu VHDL, współdziała z biblioteką Synopsys i jest tworzona przez firmę Synopsys). Aby touchpad był wykrywany przez system, należy w jądrze aktywować następujące moduły:

- i2c_designware_core
- i2c_designware_platform
- i2c_designware_pci
- intel_pmt_telemetry
- intel_lpss_pci
- mfd_intel_lpss (*Low Power Subsystem*)
- intel_pmc_core (*Power Managment Controller*)
- hid_multitouch
- hid_sensor_hub

- hid_generic

także opcja w jądrze:

Device Drivers / Input Device Support / Mice / <M> Synaptics I2C Touchpad Support

Touchpad składa się z dwóch obszarów:

- **trackpad** – obszar po którym przesuwamy palec;
- **clickpad** – obszar, gdzie znajdują się fizyczne przyciski.

Istnieją też dwa typy bibliotek obsługujących touchpady:

- **xf86-input-libinput** – większość dystrybucji preferuje właśnie tę bibliotekę;
- **xf86-input-synaptics**.

Najpierw musimy dowiedzieć się, jaki mamy model touchpada:

```
# cat /proc/bus/input/devices
```

```
# xinput list
```

```
# xinput list-props 11 (sprawdzamy właściwości; 11 to numer ID naszego touchpada)
```

```
Capabilities: 1,0,0,1,1,0,0
```

```
# libinput list-devices
```

Legenda:

- 1 - czy jest fizyczny lewy przycisk?
- 2 - czy jest fizyczny środkowy przycisk?
- 3 - czy jest fizyczny prawy przycisk?
- 4 - czy możliwa jest obsługa dwóch palców?
- 5 - czy jest możliwa obsługa trzech palców?
- 6 - czy istnieje obsługa pionowej rozdzielczości?
- 7 - czy istnieje możliwość obsługi poziomej rozdzielczości.

Ustalamy, który event reprezentuje touchpad:

```
# cat /dev/input/event0 (i ruszamy palcem po touchpadzie; jeśli nic nie wykrywa, to następny event)
```

```
# xev (ruszamy palcem po touchpadzie)
```

```
# libinput debug-events (ruszamy palcem po touchpadzie)
```

W przypadku laptopa asus-i7 należy wyedytować plik *drivers/input/mouse/elan_i2c_core.c* (lub nawet w pliku binarnym) i dodać odpowiedni model touchpada.

Zawartość pliku */etc/X11/xorg.conf.d/50-synaptics-touchpad.conf*:

Section "InputClass"

```
Identifier "Touchpad"
Driver     "synaptics"           # Wartości: evdev, synaptics
Option    "HorizEdgeScroll"    "0"
Option    "VertEdgeScroll"     "0"
Option    "TapButton1"        "1"           # Uderzenie jednym palcem na touchpadzie
                                           # odpowiada naciśnięciu lewego klawisza
Option    "TapButton2"        "3"           # Uderzenie dwoma palcami na touchpadzie
                                           # odpowiada naciśnięciu prawego klawisza
Option    "TapButton3"        "3"           # Uderzenie trzema palcami na touchpadzie
                                           # także odpowiada naciśnięciu prawego
                                           # klawisza
Option    "AreaBottomEdge"     "1300"       # Aktywna powierzchnia od górnej do dolnej
                                           # krawędzi (to cała powierzchnia to 1575)
Option    "LockedDrags"        "1"           # Czy po dwukrotnym uderzeniu palcem
```



```

# Option "FingerLow" "13" # mozliwe jest przesuwanie okien i obiektow
# Ponizej tej granicy dotkniecie uwazane jest
# za nacisniecie przycisku (inni maja tu
# wpisane 35, a w High 40)
Option "ClickPad" "1" # Uaktywnia fizyczne przyciski
# Poniższa opcja definiuje obszar tylko prawego przycisku, środkowy jest pomijany (bo go
fizycznie nie ma). Inny przykład: "60% 0 0 0 40% 60% 0 0" - zaczynając od prawej strony: area
dla prawego zaczyna się od 60%, a środkowy od 40% do 60%.
Option "SoftButtonAreas" "50% 0 85% 0 0 0 0 0"
Option "SHMConfig" "on" # Potrzebne do diagnostyki synclient.
MatchIsTouchpad "on"
MatchDevicePath "/dev/input/event*"
MatchOS "Linux"
EndSection

```

Diagnozowanie

1. **# glxinfo** (czy działa GL? ---> "Direct rendering: Yes"; zawarty w pakiecie mesa-progs)
2. **# cat /var/log/Xorg.0.log** (wyświetla logi)
3. **# glxgears** (podaje szybkość GL, czyli ilość ramek na sekundę, np. 1698Fps; zawarty w pakiecie mesa-progs)
4. **# fg_lgxgears** (szybkość GL w kartach ATI; moja prędkość: 214.000 fps)
5. **# cat /proc/driver/nvidia/agp/*** (parametry AGP)

Konfiguracja klawiatury

Zawartość pliku `/etc/X11/xorg.conf.d/60-klawiatuara.conf`:

```

Section "InputClass"
Identifier "Keyboard0"
Driver "evdev"
Option "CustomKeycodes" "on"
Option "XkbLayout" "pl,gr"
Option "XkbRules" "xorg"
Option "XkbVariant" ",polytonic"
Option "XkbOptions" "compose:rctrl,grp:lwin_toggle,lv3:ralt_switch_multikey"
Option "XkbModel" "pc105" # Inne: evdev.
MatchIsKeyboard "on"
EndSection

```

Konfiguracja tableta Wacom

W podstawowym zakresie, tablet powinien być wykryty bez żadnej konfiguracji. Ale warto wykonać poniższe kroki:

```
# emerge wacomtablet libwacom xf86-input-wacom
```

W jądrze należy aktywować:

`Device drivers` →

HID support →
Special HID drivers →
<M> Wacom Intuos / Graphire tablet support (USB)

Device drivers →
Input device support →
[*] Tablets →
<M> Wacom protocol 4 serial tablet support

W pliku `/etc/portage/make.conf` należy aktywować:
`INPUT_DEVICES="mouse keyboard evdev libinput wacom"`

Po kompilacji jądra, testujemy:

```
# cat /dev/input/wacom (ruszamy piórkim po tablecie; wyświetla „krzaczkę”)  
$ xsetwacom list devices  
... lista urządzeń Wacom (m.in. Stylus, Eraser)  
$ kde_wacom_tabletfinder (graficzna nakładka, pochodzi z pakietu „wacomtablet”)
```

Info: W wersji dla Windows, rejestracja, sterowniki i ustawienia dostępne są na stronie:
<http://wacom.com/start/one>

W razie konieczności należy także przeprowadzić konfigurację w pliku `/etc/X11/xorg.conf`:

```
Section "InputClass"  
    Identifier "Tablet"  
    Driver "wacom"  
    MatchIsTablet "on"  
EndSection
```

Ustawienia siły nacisku w GIMP:

Edycja / Konfiguracja urządzeń wejściowych /

- Wacom One by Wacom M Pen eraser → Tryb: Ekran
- Wacom One by Wacom M Pen stylus → Tryb: Ekran

W przyborniku GIMP:

Pędzel / Właściwości → Dynamika: Pressure Opacity (mocniejszy docisk to ciemniejszy kolor; lepiej sprawdza się przy rysowaniu)

Pędzel / Właściwości → Dynamika: Pressure Size (mocniejszy docisk to grubsza kreska; lepiej sprawdza się, czy pisanie i objaśnianiu)

Info: Warto wypróbować także inne sposoby dynamiki. Możemy także stworzyć swój własny szablon dynamik (mamy do dyspozycji tablicę / matrix).

Ustawienie siły nacisku w Inkscape

Edycja / Urządzenia zewnętrzne /

- Wacom One by Wacom M Pen eraser → Tryb: Przesiej
- Wacom One by Wacom M Pen stylus → Tryb: Przesiej
- [x] Używaj czułości nacisku tabletu (wymaga ponownego uruchomienia)

Ołówek / Tryb: Użyj nacisku wejścia

Info: W maszynie wirtualnej (Windows) może wystąpić problem braku kursora.

Wayland - następca X.org?

X.org to rozwiązanie pochodzące z lat '80 (X11 to po prostu wersja 11 protokołu X opublikowana w 1987 roku). Losy tego serwera były burzliwe. W 2004 następuje komercjalizacja kodu przez XFree86, więc część deweloperów przenosi się do X.org i rozwija tylko to środowisko (XFree86 upadło 4 lata później). Obecnie, X.org zawiera nadmiar kodu i luki w bezpieczeństwie, na przykład związane z możliwością zapisywania wszystkiego, co wprowadza użytkownik (np. hasła) za pomocą wbudowanego keyloggera (*xinput*). Spowodowane to jest faktem, że X.org nie stosuje izolacji między klientami graficznymi. Aby to przetestować wystarczy ustalić ID klawiatury:

```
$ xinput list (polecenie musi być wydane w konsoli graficznej)
```

a następnie, na innej konsoli, uruchamiamy *de facto* keyloggera:

```
$ xinput test ID (zamiast ID wpisać numer klawiatury)
```

Konsola, na której uruchomiony został ten „keylogger” przechwytyje wszystko, co wpisujemy w jakiegokolwiek aplikacji.

Temu wszystkiemu ma zarządzić serwer grafiki (display server) *Wayland Compositor* (protokół *Wayland*), który - w przeciwieństwie do *X.org* - sporą część zadań związanych z wyświetlaniem graficznym przekazuje klientom (one muszą być na to gotowe i potrafić to obsługiwać) i stosuje izolację klientów graficznych. Problem w tym, że spora część aplikacji nie potrafi współpracować z *Waylandem*, zaś on sam nie obsługuje sterowników z zamkniętym kodem (*nVidia* oraz *AMD*¹), jedynie karty graficzne z otwartym kodem (np. *Intel*). Z tego właśnie powodu, jeśli nasza karta graficzna jest na chipie *nVidia* lub *AMD* (zamknięty kod, binarne sterowniki), możliwe jest zastosowanie jedynie *X.org* (protokół *X11*, biblioteka *libX11* dla klientów), które współpracuje z takimi sterownikami.

Ostatnia aktualizacja: 28 czerwca 2023.

¹ AMD porzuciła niedawno rozwój otwartego kodu *FGLRX (Catalyst)* na rzecz rozwiązań zamkniętych (*AMDGPU-Pro*).