

Wayland - informacje

Położenie: (nie dotyczy)

© 3bird Projects 2024, <http://edukacja.3bird.pl>

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 (aplikacja zawiera w sobie zarówno serwer - czyli kompozytor - jak i klienta; dlatego istnieje izolacja klientów graficznych). Istnieje kilka różnych implementacji kompozytorów (*Kwin (KDE)*, *Weston*, *Mutter (Gnome)*, *Wayfire*, *Sway* - wiele z nich korzysta ze wspólnej biblioteki *wlroots*). Problem w tym, że spora część aplikacji nie potrafi współpracować z protokołem *Wayland*, 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.

Niestety, na chwilę obecną nie jest jeszcze możliwe całkowite zrezygnowanie z *X.org* (wciąż potrzebny jest pakiet *xorg-server* i pokrewne, wciąż nie można ustawić globalnie flag „-X, -xorg”).

Instalacja pakietów

Należy zainstalować:

- *qtwayland*
- *kwin* (tzw. kompozytor)
- *spectacle (printscreen)*
- ~~*xf86-video-intel*~~ (dla kart Intel, w tym *iris*, z flagą *SNA [SandyBridge's New Acceleration]*; w dokumentacji jednak napisane, że jest przestarzały; lepiej *xorg-drivers*, który doinstaluje *xorg-server* [to tutaj jest sterownik *i915 / Intel DDX*]; lub też alternatywnie sterownik *modesetting DDX* z modułem *glamor*)
- *qtwayland* (z flagą „*compositor*”)
- *xwayland* (klient *Wayland*, który umożliwia działanie starych aplikacji pracujących tylko z *xorg*)
- *xlsclients* (pozwala wylistować, które aplikacje korzystają z X)
- *xf86-input-evdev*,
- *xf86-input-synaptics*,
- *ibus*,

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

- *xsm*,
- *display-manager-init*.

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; zużywa dużo procesora
    Option "DRI" "iris"
EndSection
```

Jeśli zamiast sterownika *xf86-video-intel* korzystamy z nowszego modesetting DDX (pakiet *xorg-drivers+xorg-server*):

```
Section "Device"
    Identifier "Intel Graphics"
    Driver "modesetting"
    Option "AccelMethod" "glamor"
    Option "DRI" "3"
EndSection
```

Uruchamianie Wayland

startplasmacompositor

Jeśli wystąpiły problemy, warto to polecenie poprzedzić:

XDG_SESSION_TYPE=wayland

export \$(dbus-launch)

Jeśli nadal nie działa, spróbować:

kwin_wayland --xwayland

Warto także sprawdzić zawartość pliku:

cat /usr/share/wayland-sessions/plasmawayland.desktop

Biblioteki OpenGL

Mesa - *open source*'owa biblioteka graficzna zgodna z *OpenGL* (klon *OpenGL*). Na jej podstawie powstał sterownik do kart *Voodoo* zwany *Glade*. Istnieją dwa sterowniki Mesa (dwaj prowadzący): **classic** (zintegrowane karty *Intel i915/i965*, flaga *Glamor*) 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;

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 Management 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 (w KDE jest to traktowane jako myszka).

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
                                           # możliwe jest przesuwanie okien i obiektów
#   Option      "FingerLow"           "13"         # Poniżej tej granicy dotknięcie uważane jest
                                           # za naciśnięcie przycisku (inni mają 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"
    Option      "SHMConfig"            "on"          # Potrzebne do diagnostyki synclient.
    MatchIsTouchpad      "on"
    MatchDevicePath      "/dev/input/event*"
    MatchOS               "Linux"
EndSection
```

Konfiguracja klawiatury

Pakiet „`libinput`”.

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 pisaniu 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

Ostatnia aktualizacja: 26 maja 2024.