



wayland を動かす

野島 貴英 nozzy@debian.or.jp

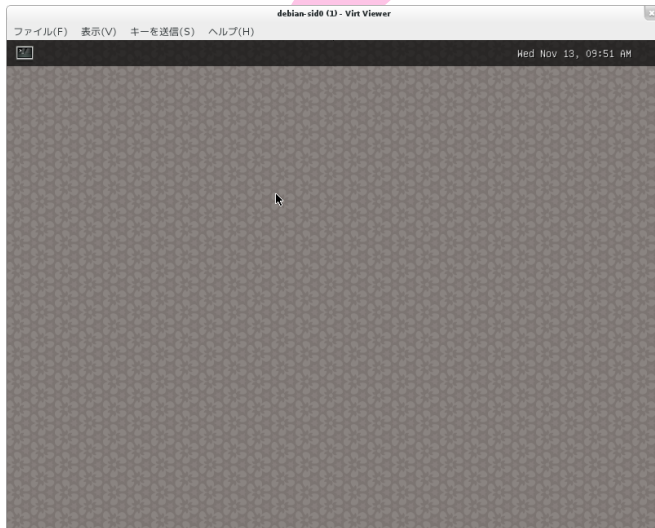
2013 年 11 月 16 日

- wayland
ディスプレイサーバーのプロトコルの一つ。
- weston
wayland プロトコルを扱えるディスプレイサーバーの実装。

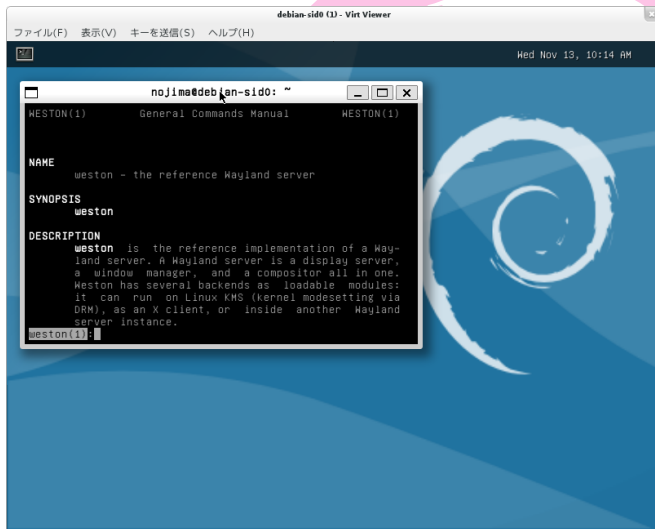
両者共に Kristian Høgsberg さんらが中心で鋭意開発中。

要はウィンドウシステムの1つの実装です。

weston 動いている様子(カスタマイズ無し)



weston 動いている様子 (weston.ini いじる)



The screenshot shows a window titled "debian-sid0 (1) - Virt Viewer". Inside the window, there is a terminal window titled "nojima@debian-sid0: ~". The terminal displays the output of the command `man weston`, showing the manual page for weston. The manual page includes the following sections:

```
WESTON(1)      General Commands Manual      WESTON(1)

NAME
  weston - the reference Wayland server

SYNOPSIS
  weston

DESCRIPTION
  weston is the reference implementation of a Wayland server. A Wayland server is a display server, a window manager, and a compositor all in one. Weston has several backends as loadable modules: it can run on Linux KMS (kernel modesetting via DRM), as an X client, or inside another Wayland server instance.

weston(1):
```

静止画だとわかりにくいのですが、にゅっとウィンドウが開いてます。

当然の疑問: Xは? ねえ? Xは?

UNIX 界で標準的ディスプレイサーバーの実装とプロトコルに X があります。が、1984 年からすっごい頑張って機能拡張と保守が続けられてきたのですが、結果的に

- Daniel Stone, “The real story behind Wayland and X”, linux.conf.au 2013,
<http://people.freedesktop.org/~daniels/lca2013-wayland-x11.pdf>
<http://www.youtube.com/watch?v=RIctzAQ0e44>

で説明があるとおりで、もういろいろ無理出まくってる状況なので、一回やり直した方がいいんじゃない? という状況になっているようです。

当然の疑問: Xは? ねえ? Xは?

補足:

wayland について phoronix の記事¹にて、Høgsberg さんによれば、「チラついたり描画してるところが見えちゃったりする事の無いウィンドウシステムを作る」(“every frame is perfect”) ということですが、実際にはXの諸々の問題をどうにかする(見直す)という事も含んでいるようです。

¹<http://www.phoronix.com>

weston 出力デバイスは何がつかえる?

upstream が用意している出力デバイスと、debian sid 提供の weston の利用できる出力デバイス一覧。

項番	出力デバイス	バックエンド名	パッケージ搭載済
1	DRM/KMS	drm-backend.so	
2	フレームバッファ	fbdev-backend.so	
3	X	x11-backend.so	
4	Wayland	wayland-backend.so	
5	Headless	headless-backend.so	
5	Raspberry Pi	rpi-backend.so	×
6	RDP	rdp-backend.so	×

動かす時の前準備

- ① まずは、`apt-get install weston` してね!
- ② 環境変数 `XDG_RUNTIME_DIR` に読み書き可能なディレクトリを指定。
 - `systemd` が動く場合:`systemd-logind` が勝手に指定してくれるよ!
 - `systemd` が動かない場合: `export XDG_RUNTIME_DIR=/tmp` とかすれば OK
- ③ `weston-launch` コマンドを使うなら、`/etc/group` の `weston-launch` グループにユーザを追加。

```
# usermod -a -G weston-launch <your-login-id>  
... この後ログアウトして、ログインしなおしてね...
```

以上!

X上で動かす

```
Xのターミナルから  
$ weston
```

!一番簡単!

DRM/KMS 上で動かす

linux のカーネルのモジュールのうち、ウィンドウシステムでよくつかわれる物に、DRM/KMS があります。

DRM Direct Rendering Manager の略。グラフィックカードを複数のアプリケーションから扱う時に必要な機能や、そもそもアプリケーションがグラフィックカードを扱うのに必要な機能 (DMA 転送のコントロールとか、リソースロックとか、Memory Mapped I/O を直接いじる時に安全にいじれるようにするとか) を提供するモジュール。

要は、ウィンドウシステム上で3D とかやろうとすると、アプリもウィンドウシステムもみんなよってたかってGPUをいじらなきゃいけないので、誰か調停しないとイケない。これを行う実装がDRM。次から次へと出てくるGPUの仕様に振り回されてる状況。

KMS Kernel-based Mode Setting の略。グラフィックカードを初期化するのに、グラフィックカードを初期化するコードをユーザランドに置いて行くと、テキストモードとグラフィクスモードへの移行の度に画面がバチバチ揺らく現象が起きる。カーネル側でブート時にあらかじめグラフィクスモードへ移行させておき、ユーザランドからはカーネルが設定したグラフィクスモードをそのまま使えるようにすれば、グラフィックを扱うアプリが起動する度にバチバチ画面が揺らくのを抑える事ができる。この処理を実現するモジュール。

DRM/KMS 上で weston 動かす

DRM/KMS 上で weston 動かすなら、

- ① グラフィカルなログイン画面が出ているようであれば、こちらを停止させます。

例: gdm3 が立ち上がっている場合の止め方

```
Ctrl-Alt-F1 等を押下してコンソールに切り替える
$ su
# service gdm3 stop
```

- ② KMS/DRM ドライバが有効であることを確かめます。

```
$ lsmod | egrep '(i915|radeon|nouveau)'
... (i915/radeon/nouveau のどれかの文字列が出れば OK) ...
```

- ③ 最後に weston を動かします。

```
$ weston-launch
```

- デモしてみます。
- こちらはIntel のチップセット入りの SONY VAIO のマシンですが、 DRM/KMS で動かすと外部VGA 端子にも weston の画面が立ち上がってますので、デモに便利です。カーソルは画面左端へ持っていけばそのまま外部VGA 端子に出力されている画面に移動します。

フレームバッファで動かす

Linux には昔からフレームバッファデバイスというのがあります。こちらを使う事により、コンソールに対して2D グラフィクス描画ができます。こちらを利用して weston を動かします。なお、詳しい動かし方は第 106 回東京エリア Debian 勉強会資料を参照ください。

- KVM で、X の力を借りずに weston を動かすにはこちらが便利です。
- debian sid のパッケージは weston-1.3.0 ですが、fbdev-backend.so に仮想端末制御まわりのバグがあり、正常に動作しません。ですので、勉強会資料に記載したとおり、weston-1.3.1 のパッケージを作り導入して稼働させる必要があります。

おまけ: wayland で動かす

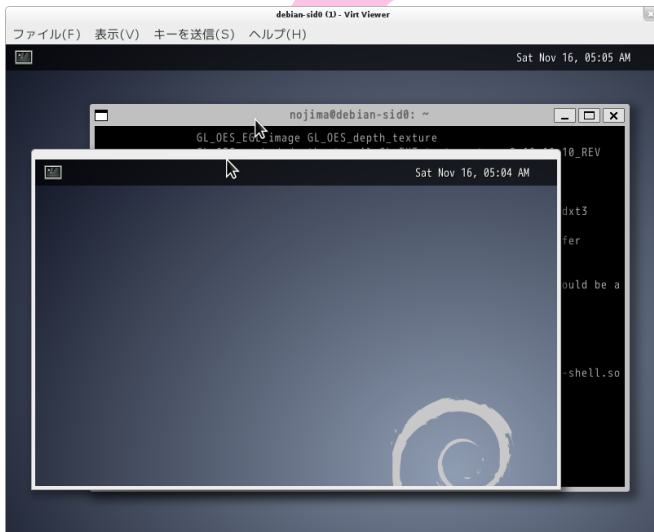
weston は wayland-backend.so を使う事により、weston 上で weston を動かす事が出来ます。

- ① weston を今までの方法で動かす
- ② wayland-terminal を開き、新しいソケットを指定して、weston を動かす。

```
$ weston -Swayland-1 --width=640 --height=400  
(width/height を指定しないと画面からはみ出すので)
```

Xnest みたいなもんです

おまけ: wayland で動かす



debian での wayland まわり対応状況

debian での wayland/weston まわり対応状況は次の通りです。

- weston 上で X アプリケーションを動かす為の機能である、XWayland は、現在の debian sid でも動作しません。なので、未だ X アプリケーションをそのまま weston 上で動かす事ができません。

理由: X サーバーに wayland 用のパッチが当たっていない状況です。ですので、xwayland.so が X を起動する際に指定する -wayland というオプションを現行の debian の X が知らないため、すぐに X が落ちてしまいます。

10月に debian-x @ d.o にパッチが流れていた²ので、きっとそのうち搭載される? かも。

²<http://lists.debian.org/debian-x/2013/10/msg00205.html>

debian での wayland まわり対応状況

- debian sid で提供される gtk などの有名どころのツールキットも、debian 提供のパッケージでは wayland が有効になっていないため、weston の上で動かす事ができません。

本来なら、

```
$ aptitude install gtk-3-examples  
$ env GDK_BACKEND=wayland gtk3-demo
```

とかすれば、gtk のデモが weston/wayland 上で動くはずなんですけどね...

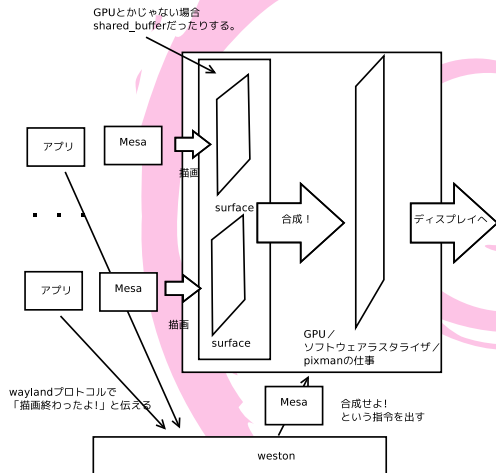
みでの通り、

- Xで実現されているレベルのネットワーク透過性は、もう設計上完全に省略(投げ捨ててるとも言える)
- 代わりに、レイヤ (pixmap)/サーフェース (EGL/mesa) を活用しまくって、エレガントにウィンドウシステムを実現しています。ウィンドウをレイヤ、もしくは、サーフェースの上に描く構造にして、weston の指示により一気に全画面を合成する作りになってます(コンポジット型。)なので、書き換えの際のチラつきがなく、また、ウィンドウの透過も正しい透過の描画が可能です(後ろのウィンドウが正しい重ね合わせで透ける等。)
- 入力デバイス対応は、xkbcommon とかを活用しています。

- weston-launch は suid で root 権限での動作をします。weston-launch は weston 稼働中に root 権限がどうしても必要な操作を担当し、root 権限不要になった資源 (ファイルディスクリプタ等) を、socket 経由で一般ユーザ権限で動く weston へ引き渡しています (sendmsg/recvmmsg の方法ですね)

weston と wayland アプリの構造

どうやって描画してるの？



westonの仕組み!

- 今のところ、
\$(MESA_SRC)/include/pci_ids/pci_id_driver_map.h
中の driver_map 変数に定義されているチップセットは
mesa により 3D アクセラレーション付きで対応している
ようです(自分は i915 のみ評価。) 求む!
radion/nouveau での動作評価。

weston が動く DRM/KMS の状況

mesa に含まれている GPU ドライバの例 (GPU じゃないのもあるけど...)

```
$(MESA_SRC)/include/pci_ids/pci_id_driver_map.h
の
static const struct {
61     int vendor_id;
62     const char *driver;
63     const int *chip_ids;
64     int num_chips_ids;
65 } driver_map[] = {
66     { 0x8086, "i915", i915_chip_ids, ARRAY_SIZE(i915_chip_ids) },
67     { 0x8086, "i965", i965_chip_ids, ARRAY_SIZE(i965_chip_ids) },
68 #ifndef DRIVER_MAP_GALLIUM_ONLY
69     { 0x1002, "radeon", r100_chip_ids, ARRAY_SIZE(r100_chip_ids) },
70     { 0x1002, "r200", r200_chip_ids, ARRAY_SIZE(r200_chip_ids) },
71 #endif
72     { 0x1002, "r300", r300_chip_ids, ARRAY_SIZE(r300_chip_ids) },
73     { 0x1002, "r600", r600_chip_ids, ARRAY_SIZE(r600_chip_ids) },
74     { 0x1002, "radeonsi", radeonsi_chip_ids, ARRAY_SIZE(radeonsi_chip_ids) },
75     { 0x10de, "nouveau", NULL, -1 },
76     { 0x15ad, "vmwgfx", vmwgfx_chip_ids, ARRAY_SIZE(vmwgfx_chip_ids) },
77     { 0x0000, NULL, NULL, 0 },
78 };
```

weston が動く DRM/KMS の状況

- '-use-pixman' を付与して weston-launch を起動して GPU による 3D アクセラレーションを投げ捨てれば、
`/lib/modules/`uname -r`/kernel/drivers/gpu/drm/`
以下にある DRM/KMS ドライバを利用して mesa が対応していない GPU の元でも動かすことができそうなものですが、何故か weston が SEGV で落ちまくるなどの現象が発生するようです。こちらの理由は、自分には良く解っていません。
(これがなければ、グラフィクスチップの 2D アクセラレーションすら無いフレームバッファは使わなくても済む...)

linux 向けのグラフィクス界限は、いろいろ技術がありまして... よい文章は

DRM/KMS,FB,V4L2 の説明でよい文章

http://elinux.org/images/2/22/KMS_FB_and_V4L2_How_to_Select_a_Graphics_and_Video_API.pdf

pixman pixman は、2D の描画をアセンブラで記述する等して、非常に最適化された2D 描画を可能にするライブラリです。画像の合成、ポリゴンを台形単位に分割して描画する機能なども持ちます。アンチエイリアシングなども搭載しているため非常に美しい描画が可能です。しかしながら、使い方のドキュメントが存在してないのが残念です。本家: <http://www.pixman.org/>

`mesa` mesa は、オープンソースでの実装の OpenGL です。最近のトレンドとしては、`gallium3d` と `llvmpipe` があります。`gallium3d` は state tracker と呼ばれる仕組みを用いて、GPU 個別の事情をうまく住み分ける事ができるアーキテクチャであり、`llvmpipe` は CPU のソフトウェアレンダラの実装ですが、LLVM の技術を利用して高速化とシェーダ言語への対応を行っているものです。本家: <http://www.mesa3d.com>

gallium3d の良い文献

Gallium3D Mesa's New Driver Model

<http://www.slideshare.net/olvaffe/gallium3d-mesas-new-driver-model>

オープンソースのグラフィクス事情について詳しい情報が活発に載るニュースサイト

phoronix

<http://www.phoronix.com>

は、

があります。ここは、毎日オープンソースプロダクトのグラフィクス事情についての記事が精力的に書かれ、weston/wayland の動向もこちらに真っ先に乗ります。オープンソースプロダクトのグラフィクス性能は独自のベンチマーク評価を実際に行い公表しているのも面白いです。もし、オープンソースのグラフィクス事情を素早くキャッチしたい時はこちらのサイトがおすすめです。他にも良いサイトがあったら教えて~!

おわりに

weston/wayland は、いろいろ賑わってきていますし、実装もシンプルでやることはいろいろと多いと思ってます。ですので、Hack するにはちょうどよいプロダクトだとおもいます。