

Окна должны жить: идеальный тайлинг на Perl

Сергей Жмылёв
Организатор Perl-Conf.Ru

Perl-Conf.Ru/26

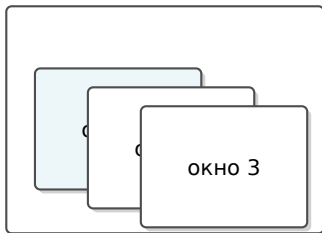
Сергей Жмылёв

Организатор Perl-Conf.Ru

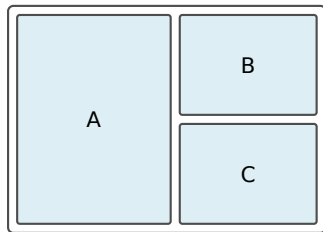


- ▶ Никогда не работал программистом
- ▶ Автор нескольких CPAN модулей, люблю Open Source
- ▶ Написал тайлинговый WM на Perl

Виды оконных менеджеров



Стековый



Тайловый

История

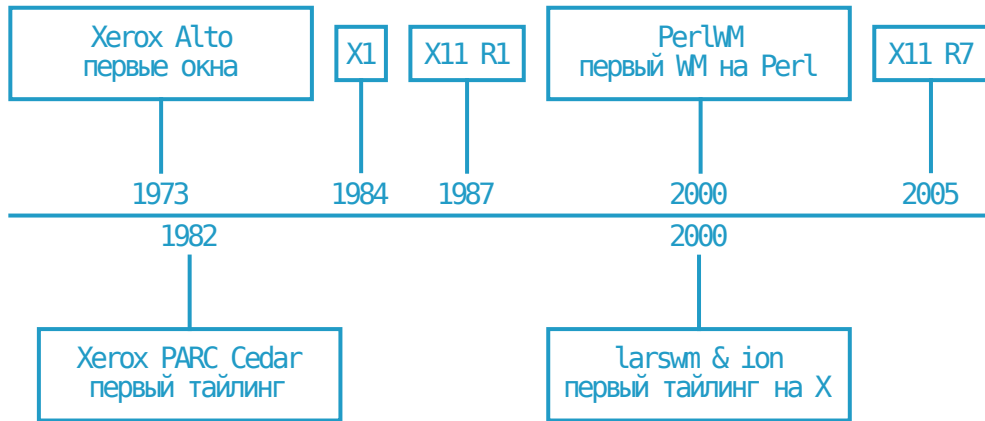
Окна должны жить!

Perl-Conf.Ru/26

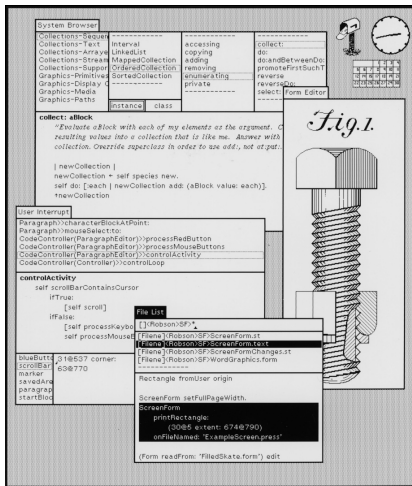
In a world with no Walls and no Fences,
we need no Windows and no Gates!



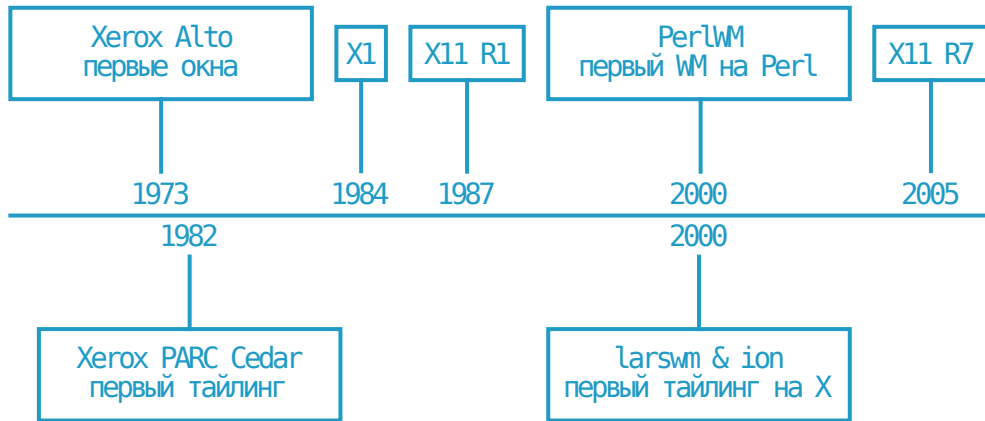
История (1)



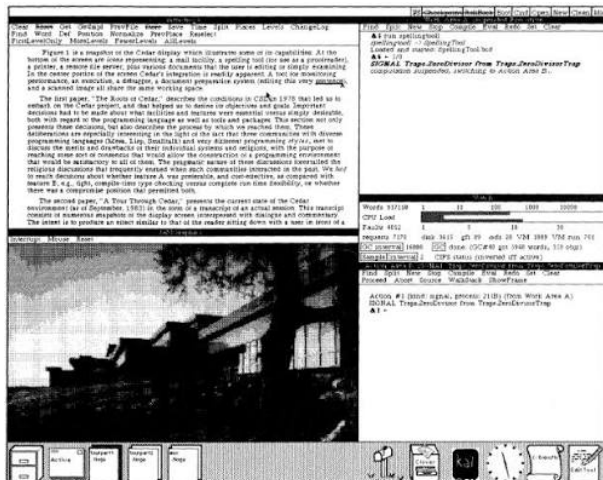
Xerox Alto (1973)



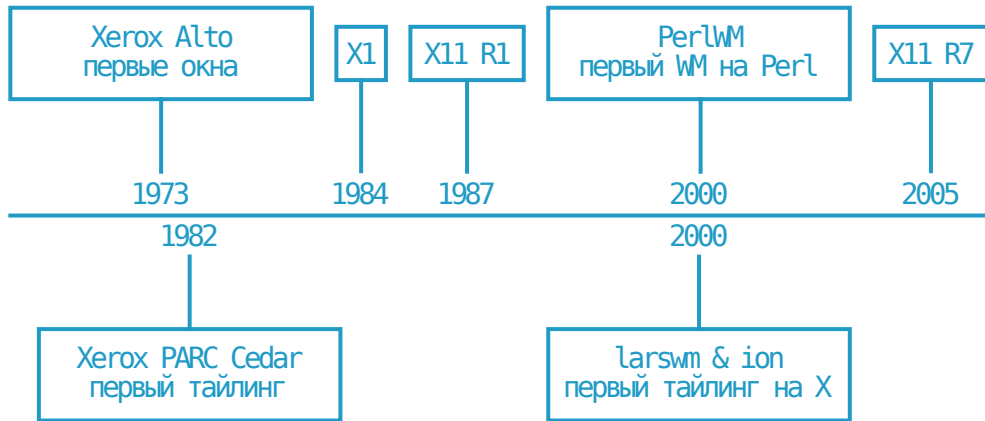
История (1)

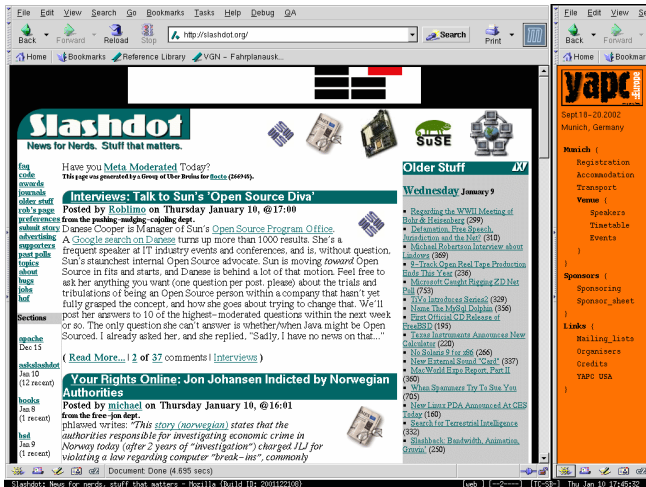


Xerox PARC Cedar (1982)

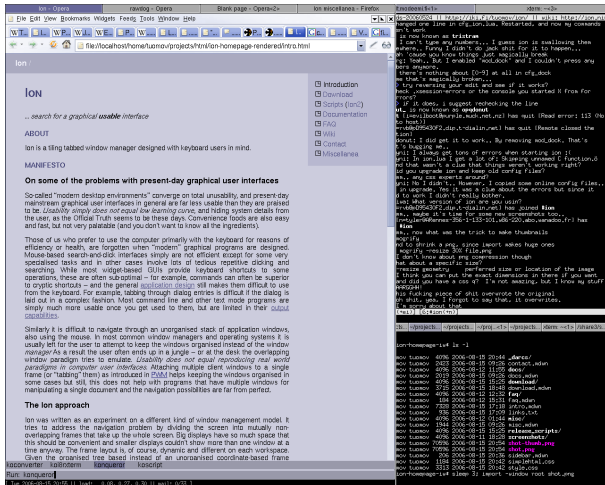


История (1)



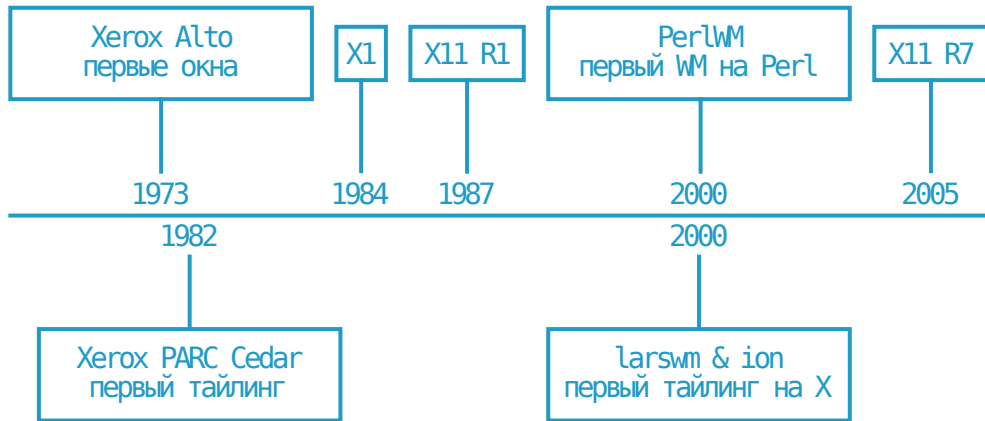


ion (2000)



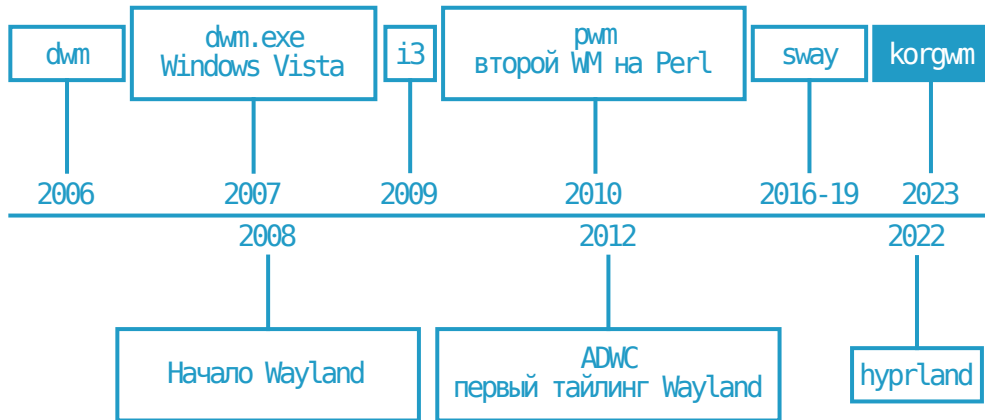
История (1)

Perl-Conf.Ru/26

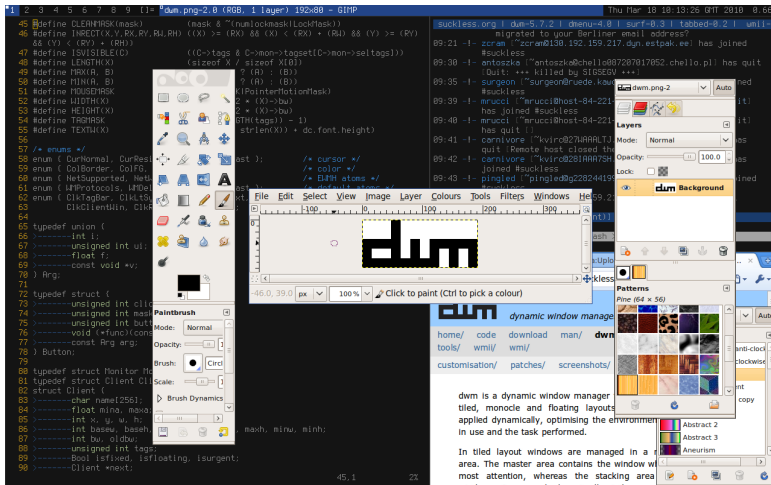


История (2)

Perl-Conf.Ru/26

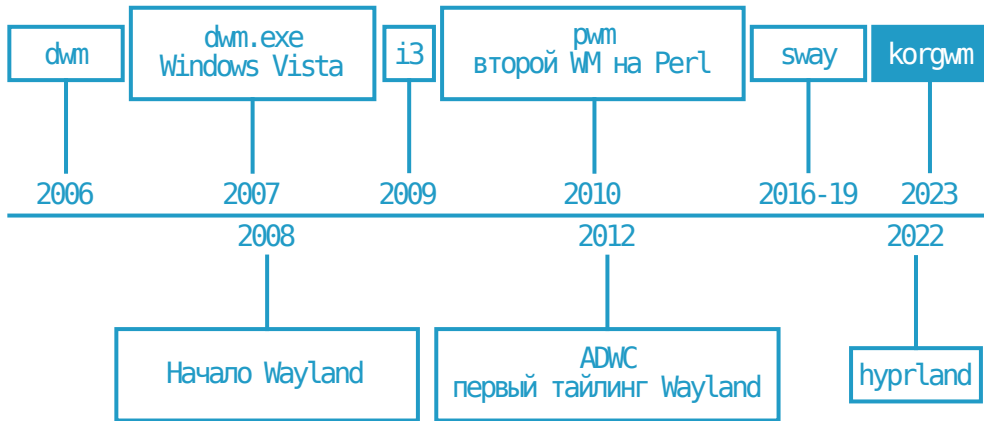


dwm (2006)



История (2)

Perl-Conf.Ru/26



```

xcb (<"/i3/src/> - VM
state = state_for_frame(con->frame);

if (state->name != NULL) {
    LOG("pushing name %s for con %p\n", state->name, con);
    xcb_change_property(conn, XCB_PROP_MODE_REPLACE, con->frame,
        XCB_ATOM_WM_NAME, XCB_ATOM_STRING, 8, strlen(state->name), sta
te->name);
    FREE(state->name);

    if (con->window == NULL) {
        /* Calculate the height of all window decorations which will be drawn on to
        * this frame. */
        uint32_t max_y = 0, max_height = 0;
        TAILQ_FOREACH(current, &(con->nodes_head), nodes) {
            rect = &current->decor_rect;
            if (dr->y >= max_y && dr->height >= max_height) {
                max_y = dr->y;
                max_height = dr->height;
            }
        }
        rect->height = max_y + max_height;
        if (rect->height == 0)
            con->mapped = false;

        /* Reparent the child window (when the window was moved due to a sticky
        * container) */
        if (state->need_reparent && con->window != NULL) {
            LOG("Reparenting child window");

            /* Temporarily set the event masks to XCB_NONE so that we won't get
            * UnmapNotify events (otherwise the handler would close the container).
            * These events are generated automatically when reparenting. */
            uint32_t values[] = { XCB_NONE };
            xcb_change_window_attributes(conn, state->old_frame, XCB_CHEVENT_MASK, values);
            xcb_change_window_attributes(conn, con->window->id, XCB_CHEVENT_MASK, values);

            xcb_reparent_window(conn, con->window->id, con->frame, 0, 0);

            values[0] = FRAME_EVENT_MASK;
            xcb_change_window_attributes(conn, state->old_frame, XCB_CHEVENT_MASK, values);
            values[0] = CHILD_EVENT_MASK;
            xcb_change_window_attributes(conn, con->window->id, XCB_CHEVENT_MASK, values);

            state->old_frame = XCB_NONE;
            state->need_reparent = false;

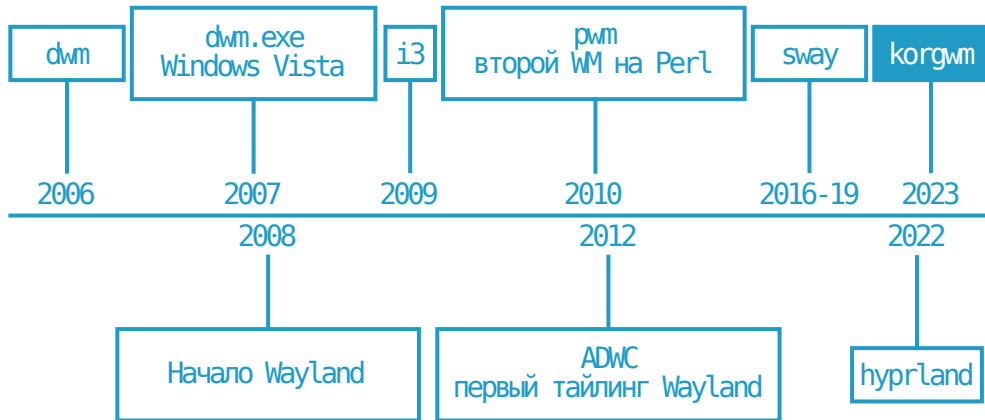
            con->ignore_unmap++;
            LOG("Ignore unmap for reparenting of con %p (win %x) is now %d\n",
                con, con->window->id, con->ignore_unmap);
        }

        bool fake_notify = false;
        /* Set new position if rect changed (and if height > 0) */
    }
}
src/x.c 540,57 542

```

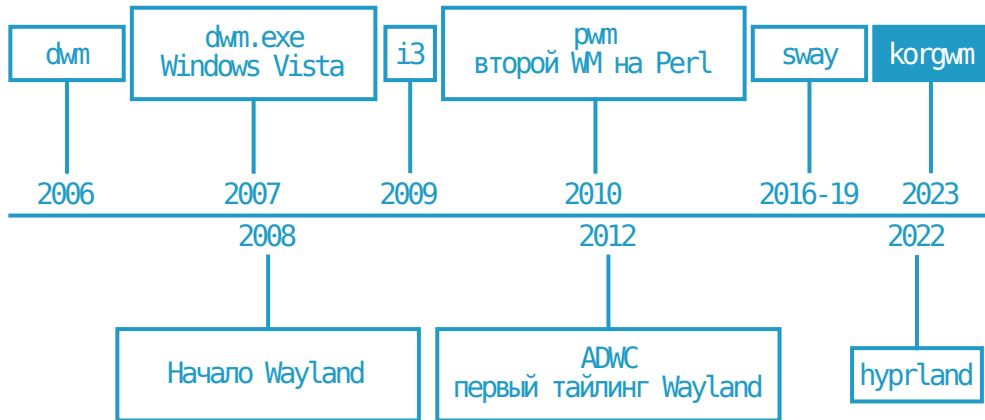
История (2)

Perl-Conf.Ru/26



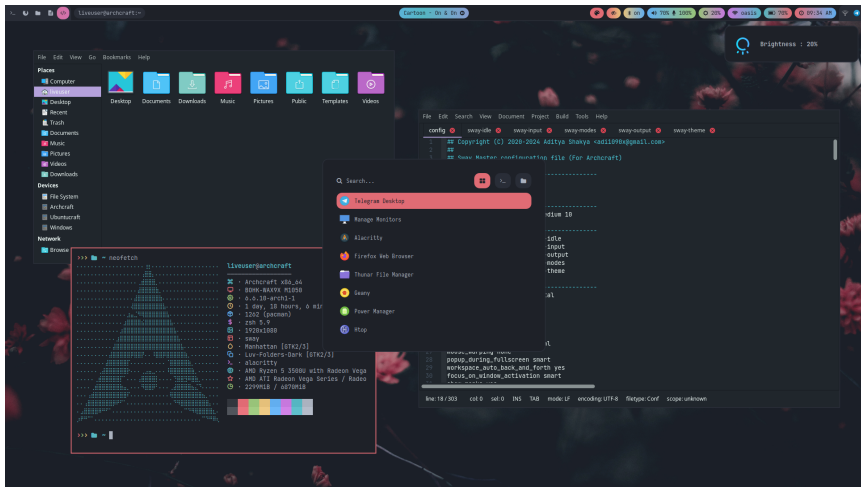
История (2)

Perl-Conf.Ru/26



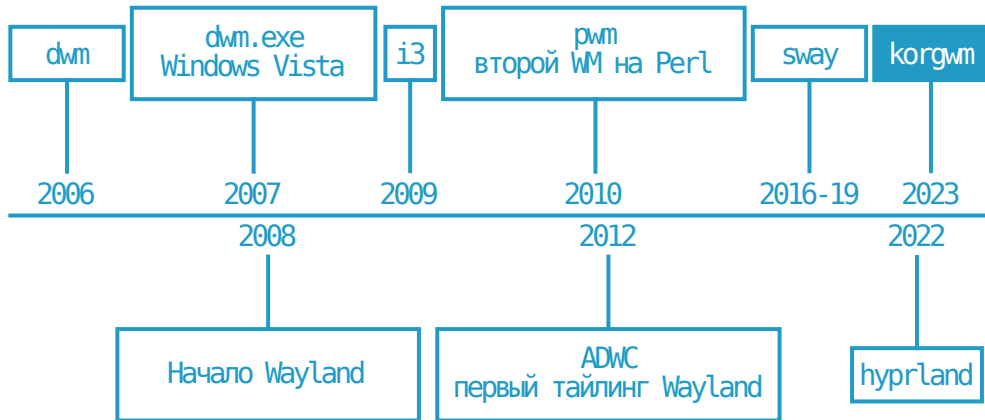
Sway (2016, 2019)

Perl-Conf.Ru/26



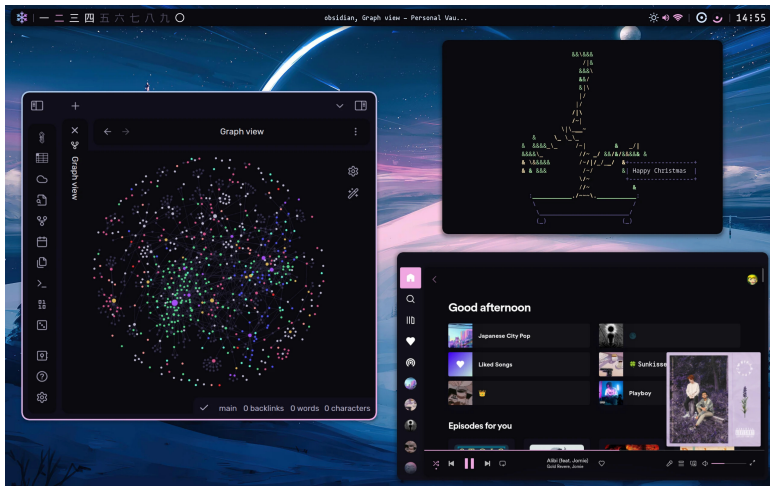
История (2)

Perl-Conf.Ru/26



Hyprrland (2022)

Perl-Conf.Ru/26



Perl

- ▶ Автор: rdw (Rich Williams)
- ▶ Разработка: 2000—2004
- ▶ `X11::Protocol`
- ▶ *So, rather than do the right thing and contribute to an existing window manager, I decided to write my own - in perl. All the mature window managers looked pretty tricky to make any major changes - especially changes in behaviour.*
- ▶ <https://perlwm.sourceforge.net/>

PerlWM — первый WM на Perl

Perl-Conf.Ru/26



```
emacs: perlwm
[1] /usr/bin/perl -w

#####
use X11::Protocol;
use Data::Dumper;

use strict;
use vars qw($x $client %font %color %gc %dim $focus %actions %bindings);

#####

%font = (title => 'lucidasans-14',
         title_focus => 'lucidasans-bold-14');

%color = (frame => 'grey78',
         frame_focus => 'white',
         title => 'black',
         title_focus => 'red');

%gc = (frame => { foreground => 'frame' },
      frame_focus => { foreground => 'frame_focus' },
      title => { font => 'title',
                foreground => 'title' },
      title_focus => { font => 'title_focus',
                      foreground => 'title_focus' });

%dim = (frame_width => 2);

%actions = (move => { drag => \&move_drag },
           size => { press => \&size_press, drag => \&size_drag });

%bindings = (frame => { 'Button1' => 'move' },
            window => { 'Mod1 Button1' => 'move',
                       'Control Mod1 Button1' => 'size',
                       'Button1' => '+click' });

#####
--XEmacs: perlwm (CPerl CVS:1.1.1.1 font)---Top
Scan for "hard" Perl constructions completed.
```

- ▶ Автор: ohsaki (Hiroyuki Ohsaki)
- ▶ Copyright: 2010—2016, v 2.40 2024/10/21
- ▶ X11::Protocol с добавлением X11::Xlib для захвата клавиатуры
- ▶ Комбинированный WM:
 - programmed mode — расположение по правилам
 - tiled mode — тайлинг
- ▶ Только Debian, приоритет Emacs, максимум 30 окон
- ▶ <http://lsnl.jp/~ohsaki/software/pwm/>



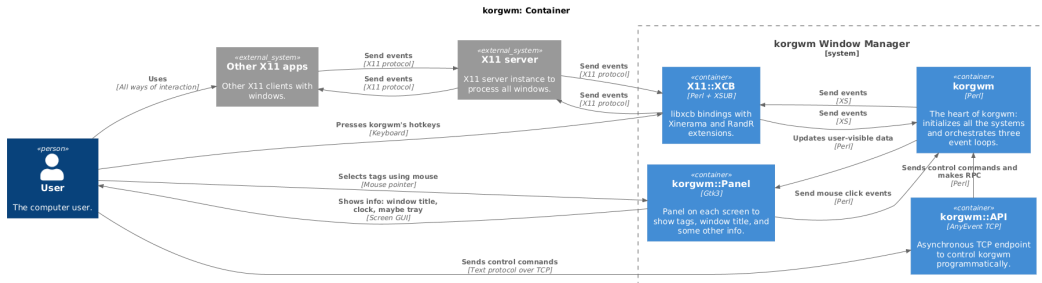
- ▶ 2005—2009: GNOME2 (Compiz Fusion)
- ▶ 2009—2013: Openbox, Awesome, dwm, wmfs
- ▶ 2013—2022: Windows, MacOS
- ▶ 2022—2023: GNOME3
- ▶ 2023—...: [korgwm](#)

All hail WMFS!

- ▶ 2010-й — самый впечатляющий WM: **WMFS** v1 (@xorg62)
- ▶ Автор бросил его ради WMFS2 (не хватало EWMH, Xft, always_on_top)
- ▶ С тех пор — zhmylove/wmfs
- ▶ Технологии ушли вперёд, форк заброшен, но осталась идея написать своё

korgwm = window manager by KorG

Компоненты korgwm



https://raw.githubusercontent.com/zhmylove/korgwm/refs/heads/master/architecture/10_context.svg

Perl || .* = Perl

Выбор очевиден!

- ▶ X11 — сервер, Wayland — протокол
- ▶ Wayland: самописный композитор (50к строк) или wlroots/smithray/etc.
- ▶ Wayland баги на Debian 12: screen sharing в WebRTC, проблемы с курсором в иерархии страниц Atlassian Confluence
- ▶ wlroots: *Pluggable, composable, unopinionated modules for building a Wayland compositor; or about 50,000 lines of code you were going to write anyway.* [Way-cooler book](#)

- ▶ X11: нет HiDPI, тиринг, нет изоляции приложений, *что-то ещё?*
- ▶ Perl:
 - `X11::Protocol`
 - `X11::Xlib`
 - `X11::XCB` (только клиент)
 - `WL` (только клиент)

<https://github.com/zhmylove/korgwm/blob/master/doc/Development.md>

- ▶ Сырой `X11::Protocol`: свой двоичный кодер-декодер для 200+ структур — дорого
- ▶ **Xlib** (1985): берёт всё на себя, но синхронная и переусложнена legacy
- ▶ **XCB** (2001): лёгкие асинхронные биндинги, но документация...

```
$ find /usr/share/man -name 'xcb_*'  
-exec zcat {} + |grep -c TODO
```

10666

REQUEST ARGUMENTS

conn The XCB connection to X11.

deviceSpec
 TODO: NOT YET DOCUMENTED.

full TODO: NOT YET DOCUMENTED.

partial TODO: NOT YET DOCUMENTED.

firstType TODO: NOT YET DOCUMENTED.

nTypes TODO: NOT YET DOCUMENTED.

firstKeySym
 TODO: NOT YET DOCUMENTED.

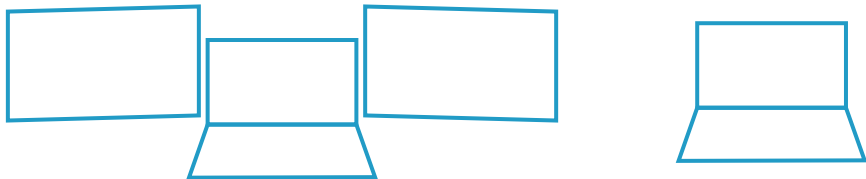
nKeySyms TODO: NOT YET DOCUMENTED.

Manual page xcb_xkb_get_map(3) line 60 (press h for help or q to quit)

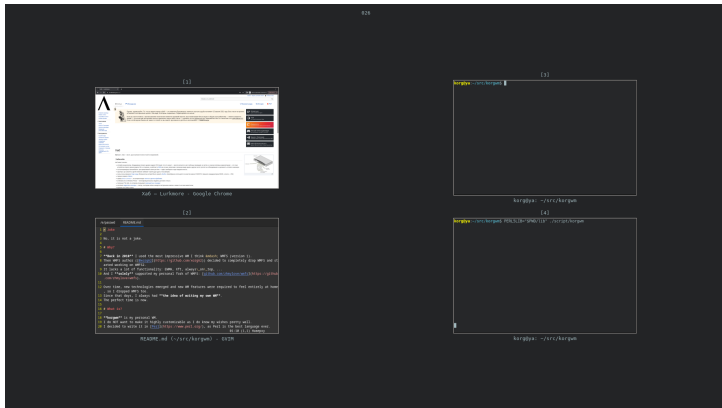
- ▶ Автор Perl-модуля X11::XCB — **Michael Stapelberg**, создатель i3
- ▶ Написал только клиентскую часть для тестирования i3
- ▶ Я предложил крупный патч, но потом мы договорились о мейнтейнерстве
- ▶ Обмен: перед релизом тестирую X11::XCB с последним тегом i3 и его master-веткой
- ▶ Добавил серверную функциональность — теперь X11::XCB годен для написания WM

Фича: авто-раскладка на мониторах

- ▶ Окно само едет на правильный экран
- ▶ Раскладка пересобирается автоматически при смене мониторов



Аналог MacOS Mission Control



- ▶ Сначала использовал Gtk3 и GdkPixbuf
It's a bad function, and people should feel bad about using it
- ▶ WebRTC: пришлось добавить поддержку Composite в X11::XCB и реализовать `get_image_data()`
- ▶ Потекла память...

```
uint8_t *  
xcb_get_image_data (const xcb_get_image_reply_t *R)  
{  
    return (uint8_t *) (R + 1);  
}
```

Проблемы при реализации Expose (2)

Perl-Conf.Ru/26

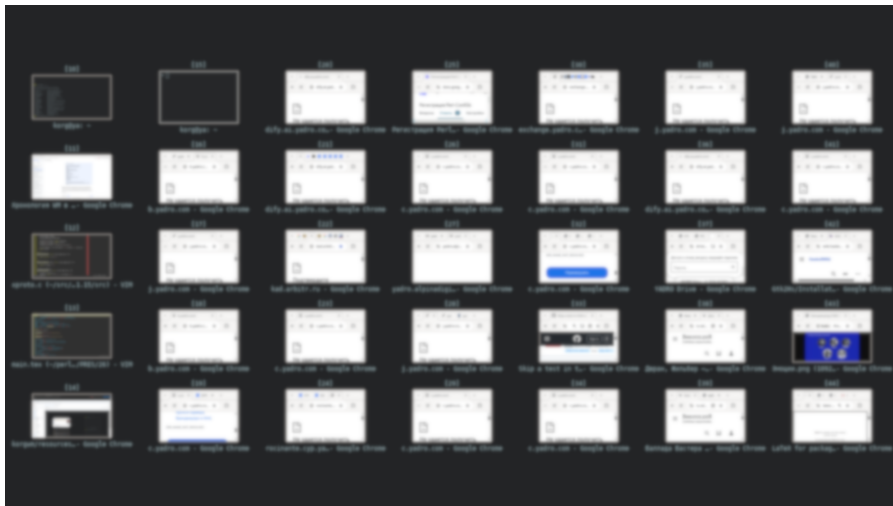
- Оказалось, цвета в X11 отдаются в BGRA, пришлось написать `get_image_data_rgba()`:

```
/* We've written this code with @TheTremblingDoe to optimize BGRA -> RGBA
   translation */
for (i = 0; i < xcb_get_image_data_length(reply) / sizeof(uint64_t); i++) {
    ((uint64_t*)xcb_get_image_data(reply))[i] =
        (((uint64_t*)xcb_get_image_data(reply))[i] & 0x000000FF000000FF) << 16) |
        (((uint64_t*)xcb_get_image_data(reply))[i] & 0x00FF000000FF0000) >> 16) |
        (((uint64_t*)xcb_get_image_data(reply))[i] & 0xFF00FF00FF00FF00));
}

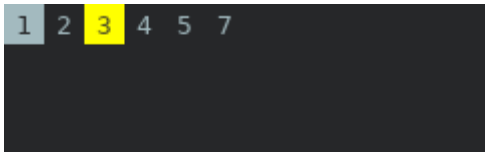
hv_store(hash, "sequence", strlen("sequence"), newSViv(reply->sequence), 0);
hv_store(hash, "length", strlen("length"), newSViv(reply->length), 0);
hv_store(hash, "depth", strlen("depth"), newSViv(reply->depth), 0);
hv_store(hash, "visual", strlen("visual"), newSViv(reply->visual), 0);
hv_store(hash, "data", strlen("data"),
    newSVpv(newSVpv((const char*)xcb_get_image_data(reply),
        xcb_get_image_data_length(reply)), 0);
```

Фича: Expose (2)

Perl-Conf.Ru/26



- ▶ Super+Shift+S <буква> — поставить метку
- ▶ Super+S <буква> — перейти к окну по метке



- ▶ На `AnyEvent::DBus` реализован собственный демон уведомлений, который находит окно и подсвечивает тэг, если оно невидимо
- ▶ Аналогичная подсветка включается, когда окно ставит `urgency hint` в `WM_HINTS(WM_HINTS)`

- ▶ Любой хоткей вызывает функцию
- ▶ Ту же функцию можно дёрнуть через простое TSP-подключение

```
(  
    echo "screen_select($SCREEN_PROJECTOR)"  
    echo "tag_select($line)"  
    echo "screen_select($SCREEN_LAPTOP)"  
) | nc -N 127.0.0.1 27015
```

- ▶ Надстройка над возможностями без правки WM

<https://github.com/zhmylove/korgwm/blob/master/API.md>

Внешний вид korgwm

The screenshot shows a korgwm window with two panes. The top pane contains a terminal window with the following content:

```
korg@ya:~/src/korgwm$  
korg@ya:~/src/korgwm$ scrot
```

The bottom pane contains a web browser window displaying the manual page for `mpd5`. The browser address bar shows `man.freebsd.org/cgi/man.cgi?query=mpd5`. The manual page content is as follows:

NAME
mpd5 -- netgraph multi-link PPP daemon

SYNOPSIS
mpd5 [-bkov] [-d directory] [-f file] [-p pid-file] [-s ident] [-m service] [configuration]

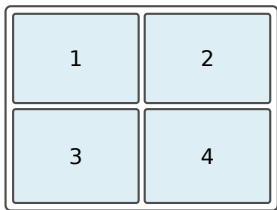
DESCRIPTION
mpd5 is a user mode PPP daemon using the *netgraph* networking system. By using Netgraph, mpd5 combines the robustness and flexibility of a user-mode PPP implementation with the speed and reliability of kernel-mode packet forwarding. All PPP negotiation is handled in user level code, while all data intensive operations such as encryption, compression, and multi-link framing are handled strictly in the kernel. Mpd supports several link layer types, a fully event-driven modes chat scripting language, and other features.

mpd5 creates a *ng_node(4)* netgraph node that is placed between a *ng_iface(4)* netgraph interface one or more link layer devices, performing multi-link PPP negotiation and encapsulation. In multi-link PPP a bundle is a collection of one or more links between two peers. Each link corresponds to some device (e.g., a modem), and each bundle corre-

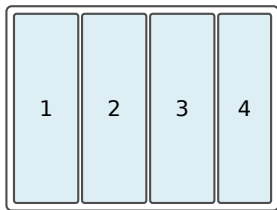
On the right side of the terminal, the output of the `top` command is shown:

```
top - 01:16:31 up 2:19, 14 users, load average: 0.14,  
Tasks: 405 total, 2 running, 403 sleeping, 0 stopped  
%Cpu(s): 0.2 us, 0.1 sy, 0.0 ni, 99.5 id, 0.2 wa, 0  
MiB Mem : 38917.4 total, 29222.5 free, 5252.8 used,  
MiB Swap: 977.0 total, 977.0 free, 0.0 used.
```

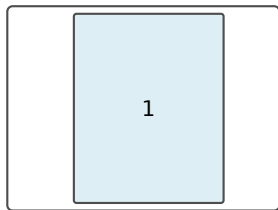
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU
27745	korg	20	0	1130.9g	128720	97652	S	1.3
3830	korg	20	0	32.6g	169924	110184	S	0.7
27411	korg	20	0	11640	5424	3272	R	0.7
27413	korg	20	0	11640	5464	3320	R	0.7
1215	root	20	0	1874144	54432	29856	S	0.3
3344	korg	20	0	1121.2g	179660	120236	S	0.3
3650	korg	20	0	509104	48244	31216	S	0.3
3895	korg	20	0	1129.1g	274700	91500	S	0.3



Динамическая сетка



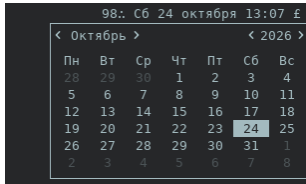
Колонки



Узкая колонка

<https://github.com/zhmylove/korgwm/blob/master/lib/X11/korgwm/Layout.pm>

- ▶ Сначала использовал устаревший механизм указания цветов, не зная, что он устаревший
- ▶ Не смог перекрасить календарь, пришлось весь код переписать на свежий CSS



Урок: надо читать инструкцию

- ▶ Машинально решил подчистить оконные индексы в `Window::DESTROY()`
- ▶ Пришлось переписать:

```
sub DESTROY($self) {  
    # Wanna free some resources?  
    # Do it inside Destroy handler:  
    # korgwm.pm/annihilate_window()  
    1;  
}
```

Урок: `Devel::MAT` незаменим!

Event loop(s)

```
for(;;) {  
    die "Segmentation fault (core dumped)\n" if $exit_trigger;  
  
    my $limit = 1024;  
    while ($limit-->0) {  
        my $evt = $X->poll_for_event();  
        unless ($evt) {  
            Gtk3::main_iteration_do(0);  
            last;  
        }  
        ...;  
    }  
  
    my $pause = AE::cv;  
    my $w = AE::timer $cpu_saver, 0, sub { $pause->send };  
    $pause->recv;  
}
```

```
#!/perl -T
use strict;
use warnings FATAL => 'all';
no warnings 'experimental::signatures';
use feature 'signatures';
use Test::More;

BEGIN { $ENV{KORGWM_DEBUG_CONFIG} = "" }

use X11::korgwm::Config;
use X11::korgwm::Common;
use YAML::Tiny;

# Verify that sample matches the default config
my $sample = YAML::Tiny->read("korgwm.conf.sample");
ref($sample) eq "YAML::Tiny" and $sample = $sample->[0];

is_deeply($sample, $X11::korgwm::Config::debug_config,
    "korgwm.conf.sample matches the default config");

done_testing();
```

- ▶ ~4100 строк, без AI, 2023—2026, март: 6.1
- ▶ стек: `libxcb` через `X11::XCB` + `AnyEvent` + `Gtk3`
- ▶ `Gtk3` — панель, `expose`, календарь
- ▶ Расширения X11:
 - `RandR`
 - `Xinerama`
 - `Composite`
 - `Xkb`
- ▶ Три активных пользователя

1. Написать свой WM легко

1. Написать свой WM ~~легко~~ несложно

1. Написать свой WM легко несложно,
особенно, с ИИ агентом

1. Написать свой WM ~~легко~~ несложно, особенно, с ИИ агентом
2. Если ещё не пробовали тайлинговые оконные менеджеры — попробуйте!

Спасибо за внимание!

Вопросы?

Perl-Conf.Ru/26