

Федеральное бюджетное образовательное учреждение  
высшего образования  
«Уфимский университет науки и технологий»  
Кафедра технической кибернетики

10	7	8	9	10	11	12	13	14	15	16	17
90											
80											
70											
60											
50											
40											
30											
20											
10											
0											

**Тайловый менеджер Nырrland в пользовательской  
сборке Arch Linux**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
к курсовому проекту по дисциплине **«Операционные системы»**

**3271.202097.000 ПЗ**

(обозначение документа)

Группа	ИВТ–ПО–201Б	Фамилия И. О.	Подпись	Дата	Оценка
Студент		Нигматуллин Д.Р			
Консультант		Рыжов Г. И.			
Принял		Рыжов Г. И.			

**Уфа 2025**

## ЗАДАНИЕ

на курсовой проект по дисциплине «Операционные системы»

Студент Кумушбаева Г. А. Группа ИВТ-ПО-201Б Консультант Рыжов Г. И.

### 1. Тема курсового проекта

Проектирование гипотетической операционной системы с заданными характеристиками

**2. Основное содержание:** на основе заданного индивидуального задания спроектировать гипотетическую компьютерную операционную систему.

**Индивидуальное задание:**

**Режим работы компьютера:** однопрограммный

**Архитектура операционной системы:** экзоядро

**Подсистема управления процессами:**

- алгоритм планирования процессов: вытесняющий – циклический *LIFO (Last In, First Out)*;
- алгоритм организации взаимодействия процессов: алгоритм пекарни Лэмпорта;
- задача синхронизации процессов: задача «Обедающие философы»;
- способ борьбы с тупиками: предотвращение тупиков – алгоритм банкира.

**Подсистема управления памятью:** сегментное распределение памяти

**Подсистема управления файлами:**

- иерархия каталогов: дерево;
- логическая организация файловой системы: индексная логическая организация;
- физическая организация файловой системы: связанный список индексов.

**Загрузка операционной системы:** загрузка с жесткого диска компьютера.

**Запуск прикладных программ:** запуск с жесткого диска компьютера.

**Структура управляющих блоков базы данных операционной системы:**

- формирование базы данных операционной системы;
- схема взаимодействия управляющих блоков базы данных операционной системы.

**3. Требования к оформлению.** Пояснительная записка должна быть оформлена в редакторе *Microsoft Word* в соответствии с требованиями ГОСТ.

В пояснительной записке должны содержаться следующие разделы:

Титульный лист.

Задание на курсовой проект.

Реферат.

Содержание.

Введение.

Выбор архитектуры вычислительной системы.

Описание режима работы компьютера.

Описание архитектуры операционной системы.

Описание подсистемы управления процессами.

Описание подсистемы управления памятью.

Описание подсистемы управления файлами.

Проектирование графического интерфейса.

Проектирование командного языка.

Загрузка операционной системы.

Запуск прикладных программ.

Структура управляющих блоков базы данных операционной системы.

Описание функционирования операционной системы.

Заключение.

Список использованной литературы.

Дата выдачи 14.10.2024

Дата окончания 21.12.2024

Руководитель \_\_\_\_\_ (Рыжов Г. И.)

## РЕФЕРАТ

Пояснительная записка – 49 с, 10 рис., 9 табл., 14 источн.

ОПЕРАЦИОННАЯ СИСТЕМА, АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ, БАЗА ДАННЫХ, ПРОЦЕССЫ, ПАМЯТЬ, ФАЙЛОВАЯ СИСТЕМА, *LINUX*, ЯДРО, ГРАФИЧЕСКИЙ ИНТЕРФЕЙС, КОМАНДНЫЙ ЯЗЫК

Объект исследования – гипотетическая компьютерная операционная система.

Предмет исследования – процессы в гипотетической компьютерной операционной системе.

Цель курсового проекта:

– закрепление и углубление теоретических знаний в области современных операционных систем, активное применение знаний, полученных в лекционном курсе, на практических и лабораторных занятиях;

– формирование систематизированного представления о основополагающих принципах работы операционных систем, о структурах и механизмах, лежащих внутри операционных системы.

Методами описания операционной системы:

– выделены ключевые составляющие: абстракции (процессы, адресное пространство, файлы, потоки); функции (создание, запись, управление, открытие, распределение); конкретные реализации (архитектуры, алгоритмы);

– определены основные функции: управление устройствами компьютера (его ресурсами), непосредственное управление процессами, ведение файловой структуры, пользовательский интерфейс;

– рассмотрены основные компоненты операционной системы: управление файловой системой, командный язык, графический интерфейс.

Результатом работы является гипотетическая компьютерная операционная система с заданными характеристиками.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	7
1 Теоретические основы тайловых менеджеров и Wayland-экосистемы.....	9
1.1 Эволюция оконных менеджеров: от stacking к tiling.....	9
1.2 Wayland vs X11: архитектурный сдвиг в графической подсистеме Linux ...	10
1.3 Hyprland: динамический тайлер нового поколения.....	12
2 Архитектура пользовательской сборки на Arch Linux.....	14
2.1 Почему Arch и почему Wayland?.....	14
2.2 Режим работы компьютера .....	14
2.3 Принципы проектирования конфигурации.....	15
3 Конфигурация Hyprland: анализ и особенности.....	17
3.1 Мониторы и рабочие пространства — логика разделения .....	17
3.2. Управление окнами: dwindle + pseudotile .....	18
3.3. Внешний вид: не просто «красиво», а функционально.....	19
3.4. Автозапуск: только то, что нужно.....	19
3.5. Правила для окон — автоматизация повседневных задач .....	20
4. Настройка Waybar: функциональность и дизайн .....	22
4.1. Почему Waybar, а не что-то другое? .....	22
4.2. Структура панели: лево — центр — право .....	22
4.3. Интерактивность: панель как пульт .....	23
4.4. Раскладка клавиатуры — кастомный скрипт.....	24
4.5. Стилизация: прозрачность, цвет и читаемость .....	24
4.6. Трей: поддержка legacy и совместимость .....	25

5. Скрипты и автоматизация.....	26
5.1. Зачем нужны скрипты в тайловом менеджере?.....	26
5.2. Переключение раскладки: <code>change_language.sh</code> .....	26
5.3. Индикатор раскладки для Waybar: <code>kb_layout.sh</code> .....	27
5.4. Уведомления о батарее: <code>battery-alert</code> .....	27
5.5. Настройка графического планшета: <code>map-tablet.sh</code> .....	28
5.6. Что можно автоматизировать ещё? .....	28
ЗАКЛЮЧЕНИЕ .....	30
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	32
ПРИЛОЖЕНИЕ.....	34
Приложение А. Конфигурация Hyprland .....	34
Приложение Б. Конфигурация Waybar .....	42
Приложение В. Стилизация Waybar .....	44
Приложение Г. Скриншот пользовательской среды.....	46

## ВВЕДЕНИЕ

Современные операционные системы стремятся к балансу между функциональностью, производительностью и эстетикой пользовательского интерфейса. В условиях роста требований к ресурсоэффективности и персонализации рабочего пространства всё большую популярность приобретают тайловые оконные менеджеры (tiling window managers, TWM) — программные компоненты, автоматически организующие окна приложений на экране без перекрытий и с минимальным участием пользователя. В отличие от традиционных сред рабочего стола (desktop environments, DE), таких как GNOME или KDE Plasma, тайловые менеджеры предоставляют полный контроль над компоновкой окон, управлением ресурсами и внешним видом системы, что особенно ценно для разработчиков, системных администраторов и пользователей, стремящихся к минимализму и высокой производительности.

С переходом от устаревшего X11-стека к современному протоколу Wayland экосистема Linux получила новые возможности в области безопасности, производительности и интеграции графических компонентов. Однако этот переход потребовал создания новых композиторов, сочетающих функции дисплейного сервера и оконного менеджера. Одним из наиболее перспективных решений в этой области стал Hyprland — динамический тайловый композитор, построенный на библиотеке wlroots, поддерживающий как строгий тайлинг, так и плавающие окна, а также современные визуальные эффекты: размытие (blur), тени, анимации и прозрачность.

Данная работа посвящена анализу пользовательской конфигурации Hyprland в среде Arch Linux — дистрибутива, известного своей гибкостью, минимализмом и ориентацией на опытного пользователя. В отчёте рассматриваются архитектурные особенности сборки, включающей:

- композитор Hyprland с настроенным мультимониторным режимом и привязкой рабочих пространств,
- информационную панель Waybar с кастомными модулями (раскладка клавиатуры, управление питанием, Bluetooth),
- систему автоматизации через скрипты и автозапуск ключевых сервисов.

Цель отчёта — не только продемонстрировать работоспособность и эстетическую привлекательность данной конфигурации, но и проанализировать её с точки зрения архитектурной целостности, масштабируемости и соответствия современным практикам настройки Wayland-сред.

# 1 Теоретические основы тайловых менеджеров и Wayland-экосистемы

## 1.1 Эволюция оконных менеджеров: от **stacking** к **tiling**

Оконный менеджер (window manager, WM) — это компонент графической подсистемы операционной системы, отвечающий за размещение, отображение и взаимодействие с окнами приложений. Исторически первые оконные менеджеры, такие как **twm** (Tab Window Manager) или **FVWM**, использовали **stacking-модель** (наложение), при которой окна могут свободно перекрывать друг друга, как листы бумаги на столе. Пользователь вручную перемещает, изменяет размер и упорядочивает окна — подход, ставший основой для современных сред рабочего стола (GNOME, KDE Plasma, XFCE).

Однако с ростом сложности задач и количества одновременно запущенных приложений **stacking-модель** стала демонстрировать недостатки:

- неэффективное использование экранного пространства,
- необходимость постоянного ручного управления окнами,
- снижение продуктивности из-за визуального «шума».

В ответ на эти вызовы в 1990–2000-х годах возникли **тайловые оконные менеджеры** (tiling window managers), такие как **dwm**, **i3**, **xmonad**. В отличие от **stacking-модели**, тайлеры автоматически разбивают экран на неперекрывающиеся области (тайлы), в которые помещаются окна. Пользователь управляет компоновкой с помощью клавиатуры, а не мыши, что повышает скорость работы и снижает когнитивную нагрузку.

Ключевые принципы тайловых менеджеров:

- **Автоматическая компоновка**: окна не перекрываются, всегда видны.
- **Клавиатурно-ориентированное управление**: минимизация зависимости от мыши.
- **Минимализм**: отсутствие панелей, кнопок, анимаций «по умолчанию».
- **Конфигурируемость через код**: большинство TWM требуют редактирования исходного кода или конфигурационных файлов.



Таким образом, тайловые менеджеры представляют собой философский и технический ответ на избыточность традиционных DE, ориентируясь на эффективность, предсказуемость и контроль.

## 1.2 Wayland vs X11: архитектурный сдвиг в графической подсистеме Linux

Долгое время графическая подсистема Linux строилась вокруг **X Window System (X11)** — протокола, разработанного в 1980-х годах. Несмотря на свою гибкость и кроссплатформенность, X11 страдает от архитектурных устареваний:

- отсутствие встроенной поддержки современных графических API (OpenGL/Vulkan),
- сложная модель безопасности (любое приложение может «подслушивать» ввод),
- избыточная сетевая ориентация (редко используется на практике),
- необходимость внешнего композитора для эффектов (тени, анимации).

В 2008 году началась разработка **Wayland** — нового протокола дисплейного сервера, спроектированного с нуля для современных требований. В отличие от X11, где дисплейный сервер (X Server) и оконный менеджер — разные сущности, в Wayland **композитор объединяет обе роли**. Это означает, что:

- композитор сам управляет буферами, вводом, разрешениями и отрисовкой,
- каждое приложение рисует напрямую в собственный буфер (через EGL/OpenGL/Vulkan),
- безопасность повышена: приложения не могут читать ввод или содержимое других окон без разрешения,
- производительность выше за счёт устранения промежуточных копий буферов.

Wayland не является «заменой X11» в прямом смысле — это **новая архитектура**, требующая переписывания клиентских приложений и инструментов. Для совместимости с унаследованным ПО используется **XWayland** — прослойка, запускающая X11-приложения внутри Wayland-сессии.

### 1.3 Hyprland: динамический тайлер нового поколения

На фоне роста популярности Wayland возникла потребность в оконных менеджерах, сочетающих **функциональность тайлеров с современной графикой**. Большинство ранних Wayland-композиторов (например, **sway**) сознательно отказались от визуальных эффектов ради стабильности и минимализма. **Hyprland**, появившийся в 2022 году, стал ответом на запрос сообщества о «красивом, но функциональном» WM. Hyprland — это **динамический тайловый композитор**, построенный на библиотеке **wlroots** (той же, что используется в sway, river, labwc). Его ключевые особенности:

- Поддержка нескольких layout-режимов: **dwindle** (бинарное деление), **master** (главное окно + стек), **спираль** и др.
- Возможность **псевдо-тайлинга** (pseudotile): окно временно становится флотинговым без потери позиции в тайле.
- Встроенные **визуальные эффекты**: размытие (blur), тени, анимации переходов, градиентные рамки.
- Полная поддержка **мультимониторности, масштабирования (HiDPI) и входных устройств** (мышь, сенсорная панель, планшеты).
- Гибкая система **правил для окон** (windowrule), позволяющая задавать поведение по классу, заголовку, монитору и другим параметрам.

Важно отметить, что Hyprland **не является средой рабочего стола** — он предоставляет только базовую функциональность WM. Всё остальное (панель, обои, уведомления, скриншоты) реализуется через сторонние утилиты (Waybar, hyprpaper, grim, mako и др.), что соответствует философии Unix: «делай одну вещь, но делай её хорошо».

Таким образом, Hyprland представляет собой синтез **практичности тайловых менеджеров, безопасности Wayland и эстетики современных интерфейсов**, что делает его привлекательным выбором для опытных пользователей Linux.

## 2 Архитектура пользовательской сборки на Arch Linux

### 2.1 Почему Arch и почему Wayland?

Выбор Arch Linux в качестве основы не случаен. Этот дистрибутив изначально строится «с нуля»: пользователь сам решает, какие компоненты ему нужны, а от каких можно отказаться. В отличие от «готовых» систем вроде Ubuntu или Fedora, здесь нет скрытых демонов, дублирующих сервисов или предустановленных приложений, которые никто не просил. Это даёт полный контроль над системой — особенно важно при работе с тонкими настройками Wayland.

Wayland же был выбран как логичное продолжение этой философии. X11, несмотря на свою надёжность, давно устарел архитектурно: он не умеет работать с современными GPU напрямую, плохо защищает от перехвата ввода и требует внешних композиторов для даже самых простых эффектов. Wayland устраняет эти проблемы, но взамен требует пересмотра привычного стека инструментов. Именно поэтому в сборке используются только нативные Wayland-приложения или их адаптированные аналоги.

### 2.2 Режим работы компьютера

Основа системы — **Hyprland** как композитор и оконный менеджер. Он заменяет собой и X-сервер, и WM, и частично — даже DE. Всё остальное подключается как отдельные модули:

- **Waybar** — лёгкая, но гибкая панель, поддерживающая кастомные скрипты и интерактивность.
- **hyprpaper** — простой менеджер обоев, совместимый с размытием в Hyprland.
- **swaykbd** — индикатор раскладки клавиатуры, необходимый, так как Wayland не передаёт состояние раскладки в панель напрямую.
- **wlogout** — красивое меню выключения/перезагрузки, запускаемое по Super+X.
- **nekoray** — прокси-менеджер, автоматически привязанный к воркспейсу 7 через правило окна.

Все компоненты запускаются через `exec-once` в конфигурации Hyprland, что исключает необходимость в `.xprofile`, `systemd --user` или других менеджерах сессий. Это упрощает отладку и делает запуск предсказуемым.

## 2.3 Принципы проектирования конфигурации

Конфигурация строилась по трём ключевым принципам:

### 1. Разделение зон ответственности.

Рабочие пространства жёстко привязаны к мониторам:

- HDMI-A-1 (внешний монитор) — воркспейсы 1–6 (основная работа: браузер, код, документы),
- eDP-1 (встроенный экран ноутбука) — воркспейсы 7–10 (фоновые задачи: мессенджер, музыка, прокси).

Такой подход позволяет не «терять» окна при переключении между мониторами и чётко разделять контексты.

### 2. Автоматизация без избыточности.

Приложения запускаются не «просто так», а по смыслу: Telegram — на 8-й, терминал Ghostty — на 9-й, Yandex Music — на 10-й. Диалоговые окна, pavucontrol, blueman — автоматически становятся плавающими и прикреплёнными (pin), чтобы не мешать основному workflow.

### 3. Эстетика как функция.

Размытие, тени, градиентная рамка активного окна и плавные анимации — не просто «красиво». Они дают визуальную обратную связь: пользователь сразу видит, какое окно активно, как происходит переключение воркспейсов, где заканчивается одно приложение и начинается другое. Это снижает когнитивную нагрузку при интенсивной работе.

## 2.2 Интеграция с системой и внешними устройствами

Сборка учитывает реальные сценарии использования:

- **Графический планшет** настраивается через кастомный скрипт `map-tablet.sh` (запускается при старте), что особенно важно для художников или тех, кто использует стилус.
- **Яркость и громкость** управляются через `brightnessctl` и `wpcctl` — это нативные Wayland-совместимые утилиты, в отличие от устаревших `xbacklight` или `amixer`.

- **Мультимедийные клавиши** работают «из коробки» благодаря привязке к XF86...-кодам.
- **Bluetooth и сеть** отображаются в панели с иконками и именами устройств, а клик по ним открывает соответствующие настройки (blueman-manager, nmtui).

Особое внимание уделено **безопасности и стабильности**:

- Используется XWayland только для тех приложений, которые пока не имеют нативной поддержки Wayland (например, Maple — запускается по Super+Shift+M).
- Все скрипты хранятся в ~/.config/scripts/, что соответствует XDG Base Directory Specification.
- Нет дублирования функций: например, управление звуком — только через PipeWire/WirePlumber, без PulseAudio в чистом виде.

Таким образом, сборка — это не просто «красивый рабочий стол», а **продуманная среда**, в которой каждый компонент выполняет свою роль, не мешая другим, и адаптированная под конкретные задачи пользователя.

### 3 Конфигурация Hyprland: анализ и особенности

#### 3.1 Мониторы и рабочие пространства — логика разделения

В конфиге явно прописаны два монитора:

- eDP-1 — встроенный экран ноутбука,
- HDMI-A-1 — внешний дисплей, подключённый справа (auto-right).

Это не просто «два экрана», а **разделение контекстов**. Внешний монитор (HDMI) — основное рабочее поле: воркспейсы 1–6. Здесь живут браузер, редактор кода, терминалы, документы. Встроенный экран (eDP) — вспомогательный: воркспейсы 7–10. Туда отправляются фоновые задачи: Telegram, музыка, прокси-менеджер.

Такой подход позволяет не переключаться между «рабочим» и «личным» на одном экране. Если внешний монитор отключён (например, в дороге), всё автоматически переносится на eDP, и ничего не теряется — просто меняется фокус.

Привязка сделана жёстко через `workspace = N, monitor:...`, что гарантирует стабильность. Да, при подключении нового монитора вручную придётся править конфиг, но в моём случае сценарий почти всегда один и тот же — ноутбук + один внешний дисплей, так что это оправдано.

### 3.2. Управление окнами: `dwindle` + `pseudotile`

В качестве основного layout выбран `dwindle` — бинарное деление экрана. Каждое новое окно делит текущую область пополам, что даёт предсказуемую и сбалансированную компоновку даже при большом количестве приложений.

Но главное — включён `pseudotile` (`pseudotile = true`). Это гениальная фишка: нажимаешь `Super+P` — окно временно становится плавающим, но не выходит из тайла. Как только ты переключаешься на другое окно — оно возвращается на своё место. Это идеально для диалогов, временных окон или просмотра чего-то поверх кода без полного перехода в `floating`-режим.

Горячие клавиши тоже продуманы:

- `Super+J` — переключает ориентацию сплита (вертикаль/горизонталь),
- `Super+V` — делает окно полностью плавающим (редко используется, но бывает нужно для игр или старых X11-приложений).

### 3.3. Внешний вид: не просто «красиво», а функционально

Многие считают, что в тайловых менеджерах не должно быть эффектов. Но здесь они работают на **юзабилити**:

- **Размытие (`blur`)** включено с `vibrancy = 0.1696` — достаточно, чтобы фон не отвлекал, но не настолько, чтобы терять контекст.
- **Тени** (`shadow.enabled = true`) чётко отделяют окна друг от друга, особенно когда их много.
- **Градиентная рамка активного окна** — `rgba(33ccffee) → rgba(00ff99ee)` — не просто «цветасто». Это мгновенная визуальная подсказка: *здесь сейчас фокус*. На тёмном фоне она особенно заметна.

- **Gaps:** `gaps_in = 5`, `gaps_out = 20` — дают «воздух» между окнами и краями экрана.

Это снижает визуальное напряжение при долгой работе.

Всё это работает без падения FPS, потому что Hyprland использует GPU-ускорение через wlroots и не тянет за собой тяжёлый DE.

### 3.4. Автозапуск: только то, что нужно

Все сервисы запускаются через `exec-once`, и ничего лишнего:

- `waybar & hyprpaper` — панель и обои,
- `nekoray` — сразу на 7-м воркспейсе (правило ниже),
- `swaykbd` & — индикатор раскладки для Waybar,
- `battery-alert` — скрипт, который следит за зарядом и уведомляет,
- `map-tablet.sh` — настройка графического планшета (Wacom и подобные).

Закомментирован `swww` для обоев — потому что `hyprpaper` уже справляется, а `swww` нужен только если хочешь анимированные или градиентные фоны. Пока не требуется.

### 3.5. Правила для окон — автоматизация повседневных задач

Самая мощная часть конфига — секция `windowrule`. Она превращает рутину в автоматику:

- Telegram → воркспейс 8,
- Ghostty → воркспейс 9 + прозрачность 0.9 (чтобы не «давил» на глаза),
- Yandex Music → воркспейс 10 + прозрачность 0.8,
- Nekoray → воркспейс 7.

Диалоговые окна (`.*dialog.*`) автоматически становятся плавающими — без этого они ломали бы layout.

`Pavucontrol` и `blueman-manager` не только флоатятся, но и:

- имеют фиксированный размер (35% 30%),
- прикреплены к верхнему левому углу (`move 0% 10%`),
- закреплены (`pin`) — не исчезают при смене воркспейса.

Это критично: когда настраиваешь звук или Bluetooth, ты не хочешь, чтобы окно «потерялось» при переключении на другой рабочий стол.

Также есть правило `suppress event maximize` — оно игнорирует запросы на максимизацию от приложений. В тайловом менеджере «максимизировать» окно — бессмыслица, и без этого правила некоторые приложения (особенно XWayland) вели бы себя странно.

Эта глава показывает, что конфигурация — не просто набор настроек, а продуманная система, где каждая строчка решает конкретную задачу пользователя. Никакой «магии», только логика и практика.



## 4. Настройка Waybar: функциональность и дизайн

### 4.1. Почему Waybar, а не что-то другое?

Когда я искал замену панели для Hyprland, главным требованием была **гибкость без перегруза**. Polybar — мощный, но заточен под X11 и требует костылей под Wayland. swaybar — слишком минималистичный, почти «голый». Waybar же оказался «золотой серединой»: он лёгкий, нативно работает под Wayland, поддерживает CSS-стилизацию и позволяет вставлять кастомные скрипты без танцев с бубном.

Важно и то, что Waybar **не пытается быть всем сразу**. Он не управляет уведомлениями, не рисует обои, не ловит скриншоты — он просто показывает информацию и реагирует на клики. Всё остальное делают другие утилиты (тако для уведомлений, grim для скриншотов и т.д.), и это соответствует философии Unix.

### 4.2. Структура панели: лево — центр — право

Панель разделена на три зоны — как в классических DE, но без лишнего:

- **Слева:** инструменты для работы — файловый менеджер, звук, микрофон, нагрузка на CPU/RAM, Bluetooth, сеть.
- **По центру:** время и раскладка клавиатуры — то, на что чаще всего смотришь, не отрываясь от задачи.
- **Справа:** системные сервисы — трей, батарея, яркость, кнопка выключения.

Такое расположение не случайно. Левая часть — «активные» модули, с которыми взаимодействуешь (клик по звуку → pavucontrol, клик по сети → nmtui). Центр — пассивная информация. Правая — статус и управление питанием.

### 4.3. Интерактивность: панель как пульт

Самое важное в Waybar — он **реагирует на действия**:

- Клик по значку глаза открывает **Nautilus** в текущей директории — удобно, когда работаешь в терминале и нужно быстро перейти в GUI.
- Клик по громкости или микрофону — запускает **pavucontrol** или переключает mute.


- Скролл по иконке микрофона — регулирует уровень записи (`pactl set-source-volume`).
- Клик по Bluetooth — открывает **blueman-manager**, а в панели отображается имя подключённого устройства (не просто «вкл/выкл»).
- Клик по сети — запускает **nmtui** в Alacritty — текстовый, но надёжный способ настроить подключение без GUI-зависимостей.
- Клик по кнопке питания — вызывает **wlogout** с красивым меню выключения, перезагрузки и сна.

Даже время кликабельно: правый клик открывает **Google Calendar** в Firefox. Это мелочь, но она экономит десятки кликов в неделю.

#### 4.4. Раскладка клавиатуры — кастомный скрипт



Wayland не передаёт состояние раскладки напрямую в панель, поэтому используется внешний скрипт:

```
bash ~/.config/waybar/scripts/kb_layout.sh
```

Он опрашивает текущий layout раз в секунду и выводит us или ru. Иконка  рядом даёт визуальную привязку. Без этого приходилось бы смотреть в терминал или надеяться на память — особенно раздражает при частом переключении.

#### 4.5. Стилизация: прозрачность, цвет и читаемость

Файл `style.txt` задаёт **единый визуальный язык**:

- Фон панели — полупрозрачный бирюзовый: `rgba(51,255,255,0.15)`. Это отсылка к цветовой схеме всей системы (рамка активного окна в Hyprland — тоже бирюзово-зелёная).
- Каждый модуль имеет лёгкий подтон: `rgba(55, 146, 179, 0.1)` — тёмно-бирюзовый, почти незаметный, но создающий ощущение «цельности».
- Шрифт — **JetBrainsMono Nerd Font**, который поддерживает иконки из Font Awesome и других наборов. Это позволяет использовать символы вроде  (Bluetooth) или  (яркость) без подключения отдельных шрифтов.

- При отсутствии сети текст становится **красным** (#ff5555) на тёмном фоне — сразу видно проблему.

Прозрачность подобрана так, чтобы панель не «сливалась» с фоном, но и не перекрывала важные детали на обоих. Особенно это заметно при работе с кодом или графикой — панель остаётся информативной, но не навязчивой.

#### 4.6. Трей: поддержка legacy и совместимость

Модуль tray включён с опцией `show-passive = true`, что позволяет отображать даже «тихие» иконки — например, от **nekoray** или **nm-applet**. Это важно, потому что многие приложения всё ещё полагаются на системный трей для индикации состояния (прокси, сеть, облачные клиенты).

Без этого пришлось бы держать отдельные окна или терминалы «на всякий случай». Трей — компромисс между чистотой Wayland и реальностью существующего ПО.

Таким образом, Waybar в этой сборке — не просто «полоска сверху», а **инструмент управления**, визуальный индикатор состояния системы и мост между терминалом и графическим интерфейсом. Каждый элемент здесь решает конкретную задачу, и ничего не добавлено «просто потому что можно».

## 5. Скрипты и автоматизация

### 5.1. Зачем нужны скрипты в тайловом менеджере?

Hyprland — мощный композитор, но он не делает всё сам. Он не переключает раскладку, не следит за батареей, не управляет планшетом и не показывает текущий язык в панели. Для этого нужны **внешние скрипты** — небольшие программы, которые заполняют «пробелы» между компонентами Wayland-стека.

В этой сборке скрипты лежат в `~/.config/scripts/` и `~/.config/waybar/scripts/`, что соответствует стандарту XDG и упрощает резервное копирование (всё в одном месте, легко синхронизировать с Git).

### 5.2. Переключение раскладки: `change_language.sh`

Стандартный способ переключения раскладки в Wayland — через `hyprctl switchxkblayout`. Но он не даёт визуальной обратной связи. Поэтому используется кастомный скрипт:

```
#!/bin/sh
# ~/.config/scripts/change_language.sh
current=$(hyprctl devices -j | jq -r '.keyboards[] | select(.active) | .active_keymap')
if [ "$current" = "us" ]; then
    hyprctl switchxkblayout all next
else
    hyprctl switchxkblayout all prev
fi
```

Он определяет текущую раскладку через `hyprctl` и `jq`, а затем переключает её. Важно: **он не дублирует** `grp:win_space_toggle` из конфига **Hyprland**, а дополняет его — потому что `swaykbd` (индикатор в Waybar) не умеет отслеживать переключение по системной горячей клавише. Этот скрипт запускается по `Super+Space` и гарантирует, что Waybar обновит статус.

### 5.3. Индикатор раскладки для Waybar: `kb_layout.sh`

Этот скрипт работает в фоне каждую секунду:

```
#!/bin/sh
# ~/.config/waybar/scripts/kb_layout.sh
hyprctl devices -j | jq -r '.keyboards[] | select(.active) | .active_keymap' | head -n1
```

Он просто выводит `us` или `ru`, и `Waybar` подставляет это в модуль `custom/kbdlayout`. Без него пришлось бы угадывать, на какой раскладке ты сейчас — особенно раздражает при вводе паролей или команд в терминале.

#### 5.4. Уведомления о батарее: `battery-alert`

Скрипт `battery-alert` — простой, но полезный:

```
#!/bin/sh

while true; do
capacity=$(cat /sys/class/power_supply/BAT0/capacity 2>/dev/null)
status=$(cat /sys/class/power_supply/BAT0/status 2>/dev/null)
if [ "$status" != "Charging" ] && [ "$capacity" -lt 15 ]; then
notify-send "☐ Низкий заряд!" "Осталось $capacity%. Подключите зарядку." -u critical
fi
sleep 120
done
```

Он проверяет заряд раз в две минуты и выдаёт **критическое уведомление**, если батарея разряжена ниже 15% и не заряжается. Запускается через `exes-once` и работает в фоне. Никаких демонов, только `notifi-send` + `мако` (или другой нотифайер).

#### 5.5. Настройка графического планшета: `map-tablet.sh`

Для владельцев `Wacom` или `XP-Pen` планшет часто «видит» весь экран, включая второй монитор, что ломает привязку стилуса. Скрипт решает это:

```
#!/bin/sh
# ~/.config/hypr/scripts/map-tablet.sh
xsetwacom set "Wacom Intuos S Pen stylus" MapToOutput HDMI-A-1
xsetwacom set "Wacom Intuos S Pen eraser" MapToOutput HDMI-A-1
```

Он жёстко привязывает стилус к основному монитору (`HDMI-A-1`). Да, это `X11`-утилита (`xsetwacom`), но под `XWayland` она работает корректно — пока нет полноценной замены в чистом `Wayland`.

#### 5.6. Что можно автоматизировать ещё?

В конфиге закомментированы два полезных скрипта:

- `random_wallpaper` — переключает обои по таймеру (требуется `swww` или `hugraper` с поддержкой нескольких изображений).
- `scrcpy` — запуск трансляции `Android`-устройства (полезно для разработки).

Также можно добавить:

- Автоматическое отключение Bluetooth при отключении наушников,
- Смену профиля яркости в зависимости от времени суток,
- Резервное копирование конфигов в Git по расписанию.

Но пока всё ограничено **минимализмом**: если задача решается двумя строками — она решается. Если требует сложного демона — откладывается.

Скрипты в этой сборке — не «навороты», а **практические решения реальных проблем**, которые не покрываются стандартными инструментами Wayland. Они маленькие, прозрачные и легко читаются — как и должно быть в Unix-стиле.

## ЗАКЛЮЧЕНИЕ

Разработка и настройка пользовательской среды на базе Hyprland в Arch Linux — это не просто техническое упражнение, а процесс проектирования рабочего пространства под конкретные задачи, привычки и эстетические предпочтения. В ходе анализа конфигурации стало ясно: каждая строка в `hyprland.conf`, каждый модуль в Waybar и каждый скрипт в `~/.config/scripts/` решает реальную проблему — будь то разделение контекстов между мониторами, мгновенное переключение раскладки или визуальная индикация активного окна.

Сборка сочетает в себе минимализм Unix-подхода (каждый компонент делает одну вещь, но делает её хорошо) и современные требования к удобству (анимации, размытие, интерактивная панель). Она не пытается копировать GNOME или KDE — она создаёт собственный язык взаимодействия с системой, где клавиатура заменяет мышь, а конфигурационные файлы становятся инструментом самовыражения.

Важно и то, что система остаётся открытой для развития: её можно легко модуляризировать, адаптировать под новые устройства, усилить с точки зрения безопасности или расширить за счёт новых скриптов. Это не «готовый продукт», а живая среда, которая растёт вместе с пользователем.

В конечном счёте, цель любой пользовательской сборки — не в том, чтобы быть «самой красивой» или «самой быстрой», а в том, чтобы не мешать работать. И в этом смысле данная конфигурация достигает своей цели: она прозрачна, предсказуема и эффективна — именно то, что и должно быть у хорошего тайлового менеджера.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Hyprland: Official Documentation [Электронный ресурс]. — Режим доступа: <https://wiki.hyprland.org>, свободный. — Загл. с экрана. — Дата обращения: 05.04.2025.
2. Wayland Protocol Specification [Электронный ресурс]. — Режим доступа: <https://wayland.freedesktop.org/docs/html/>, свободный. — Загл. с экрана. — Дата обращения: 05.04.2025.
3. ArchWiki: Hyprland [Электронный ресурс] // Arch Linux Wiki. — Режим доступа: <https://wiki.archlinux.org/title/Hyprland>, свободный. — Загл. с экрана. — Дата обращения: 05.04.2025.
4. Waybar: Modular status bar for Wayland [Электронный ресурс]. — Режим доступа: <https://github.com/Alexays/Waybar>, свободный. — Загл. с экрана. — Дата обращения: 05.04.2025.
5. wlroots — Modular Wayland compositor library [Электронный ресурс]. — Режим доступа: <https://gitlab.freedesktop.org/wlroots/wlroots>, свободный. — Загл. с экрана. — Дата обращения: 05.04.2025.
6. PipeWire Documentation [Электронный ресурс]. — Режим доступа: <https://docs.pipewire.org/>, свободный. — Загл. с экрана. — Дата обращения: 05.04.2025.
7. Nerd Fonts: Developer targeted fonts with icons [Электронный ресурс]. — Режим доступа: <https://www.nerdfonts.com/>, свободный. — Загл. с экрана. — Дата обращения: 05.04.2025.
8. grim — Screenshot utility for Wayland [Электронный ресурс]. — Режим доступа: <https://github.com/emersion/grim>, свободный. — Загл. с экрана. — Дата обращения: 05.04.2025.
9. slurp — Select a region for screenshots in Wayland [Электронный ресурс]. — Режим доступа: <https://github.com/emersion/slurp>, свободный. — Загл. с экрана. — Дата обращения: 05.04.2025.



- 10.XDG Desktop Portal [Электронный ресурс] // freedesktop.org. — Режим доступа: <https://github.com/flatpak/xdg-desktop-portal>, свободный. — Загл. с экрана. — Дата обращения: 05.04.2025.
- 11.swaykbd — Keyboard layout indicator for Sway/Hyprland [Электронный ресурс]. — Режим доступа: <https://github.com/ArtemGordinsky/swaykbd>, свободный. — Загл. с экрана. — Дата обращения: 05.04.2025.
- 12.Руководство по установке Arch Linux [Электронный ресурс] // Arch Linux Wiki. — Режим доступа: [https://wiki.archlinux.org/title/Installation\\_guide](https://wiki.archlinux.org/title/Installation_guide), свободный. — Загл. с экрана. — Дата обращения: 05.04.2025.

# ПРИЛОЖЕНИЕ

## Приложение А. Конфигурация Hyprland

```
#####
# AUTOGENERATED HYPRLAND CONFIG.
# PLEASE USE THE CONFIG PROVIDED IN THE GIT REPO /examples/hyprland.conf AND EDIT IT,
# OR EDIT THIS ONE ACCORDING TO THE WIKI INSTRUCTIONS.
# #####

#autogenerated = 1 # remove this line to remove the warning

# This is an example Hyprland config file.
# Refer to the wiki for more information.
# https://wiki.hyprland.org/Configuring/

# Please note not all available settings / options are set here.
# For a full list, see the wiki

# You can split this configuration into multiple files
# Create your files separately and then link them to this file like this:
# source = ~/.config/hypr/myColors.conf

#####
### MONITORS ###
#####

# See https://wiki.hyprland.org/Configuring/Monitors/
monitor=eDP-1,preferred,auto,1
monitor=HDMI-A-1,preferred,auto-right,1

workspace = 7, monitor:eDP-1, default:true
workspace = 8, monitor:eDP-1, default:true
workspace = 9, monitor:eDP-1, default:true
workspace = 10, monitor:eDP-1, default:true

workspace = 1, monitor:HDMI-A-1, default:true
workspace = 2, monitor:HDMI-A-1, default:true
workspace = 3, monitor:HDMI-A-1, default:true
workspace = 4, monitor:HDMI-A-1, default:true
workspace = 5, monitor:HDMI-A-1, default:true
workspace = 6, monitor:HDMI-A-1, default:true

#####
### MY PROGRAMS ###

#####
```

```
### MY PROGRAMS ###
```

```
#####
```

```
# See https://wiki.hyprland.org/Configuring/Keywords/
```

```
# Set programs that you use
```

```
$terminal = ghostty
```

```
$fileManager = dolphin
```

```
$menu = rofi --show drun
```

```
#####
```

```
### AUTOSTART ###
```

```
#####
```

```
# Autostart necessary processes (like notifications daemons, status bars, etc.)
```

```
# Or execute your favorite apps at launch like this:
```

```
# exec-once = $terminal
```

```
# exec-once = nm-applet &
```

```
exec-once = waybar & hyprpaper # & firefox
```

```
exec-once = nekoray
```

```
exec-once = swaykbd &
```

```
exec-once = /home/kk2gvb/.config/scripts/battery-alert
```

```
#exec-once = swww img /home/kk2gvb/wallpapers/1.jpg
```

```
exec-once = ~/.config/hypr/scripts/map-tablet.sh
```

```
#####
```

```
### ENVIRONMENT VARIABLES ###
```

```
#####
```

```
# See https://wiki.hyprland.org/Configuring/Environment-variables/
```

```
env = XCURSOR_SIZE,24
```

```
env = HYPRCURSOR_SIZE,24
```

```
#####
```

```
### PERMISSIONS ###
```

```
#####
```

```
# See https://wiki.hyprland.org/Configuring/Permissions/
```

```
# Please note permission changes here require a Hyprland restart and are not applied on-the-fly
```

```
# for security reasons
```

```
# ecosystem {
```

```
# enforce_permissions = 1
```

```
# }
```

```
# permission = /usr/(bin/local/bin)/grim, screencopy, allow
```

```
# permission = /usr/(lib/libexec/lib64)/xdg-desktop-portal-hyprland, screencopy, allow
```

```
# permission = /usr/(bin/local/bin)/hyprpm, plugin, allow
```

```
#####  
### LOOK AND FEEL ###  
#####
```

```
# Refer to https://wiki.hyprland.org/Configuring/Variables/
```

```
# https://wiki.hyprland.org/Configuring/Variables/#general  
general {  
    gaps_in = 5  
    gaps_out = 20
```

```
border_size = 2
```

```
# https://wiki.hyprland.org/Configuring/Variables/#variable-types for info about colors  
col.active_border = rgba(33ccffee) rgba(00ff99ee) 45deg  
col.inactive_border = rgba(595959aa)
```

```
# Set to true enable resizing windows by clicking and dragging on borders and gaps  
resize_on_border = false
```

```
# Please see https://wiki.hyprland.org/Configuring/Tearing/ before you turn this on  
allow_tearing = false
```

```
layout = dwindle  
}
```

```
# https://wiki.hyprland.org/Configuring/Variables/#decoration  
decoration {  
    rounding = 10  
    rounding_power = 2
```

```
# Change transparency of focused and unfocused windows
```

```
active_opacity = 1.0  
inactive_opacity = 1.0
```

```
shadow {  
    enabled = true  
    range = 4  
    render_power = 3  
    color = rgba(1a1a1aee)  
}
```

```
# https://wiki.hyprland.org/Configuring/Variables/#blur  
blur {  
    enabled = true  
    size = 3  
    passes = 1
```

```
vibrancy = 0.1696
```

```
}  
}
```

```
# https://wiki.hyprland.org/Configuring/Variables/#animations
```

```
animations {  
    enabled = yes, please :)
```

```
# Default animations, see https://wiki.hyprland.org/Configuring/Animations/ for more
```

```
bezier = easeOutQuint,0.23,1,0.32,1  
bezier = easeInOutCubic,0.65,0.05,0.36,1  
bezier = linear,0,0,1,1  
bezier = almostLinear,0.5,0.5,0.75,1.0  
bezier = quick,0.15,0,0.1,1
```

```
animation = global, 1, 10, default  
animation = border, 1, 5.39, easeOutQuint  
animation = windows, 1, 4.79, easeOutQuint  
animation = windowsIn, 1, 4.1, easeOutQuint, popin 87%  
animation = windowsOut, 1, 1.49, linear, popin 87%  
animation = fadeIn, 1, 1.73, almostLinear  
animation = fadeOut, 1, 1.46, almostLinear  
animation = fade, 1, 3.03, quick  
animation = layers, 1, 3.81, easeOutQuint  
animation = layersIn, 1, 4, easeOutQuint, fade  
animation = layersOut, 1, 1.5, linear, fade  
animation = fadeLayersIn, 1, 1.79, almostLinear  
animation = fadeLayersOut, 1, 1.39, almostLinear  
animation = workspaces, 1, 1.94, almostLinear, fade  
animation = workspacesIn, 1, 1.21, almostLinear, fade  
animation = workspacesOut, 1, 1.94, almostLinear, fade  
}
```

```
# Ref https://wiki.hyprland.org/Configuring/Workspace-Rules/
```

```
# "Smart gaps" / "No gaps when only"
```

```
# uncomment all if you wish to use that.
```

```
# workspace = w[1], gapsout:0, gapsin:0
```

```
# workspace = f[1], gapsout:0, gapsin:0
```

```
# windowrule = bordersize 0, floating:0, onworkspace:w[1]
```

```
# windowrule = rounding 0, floating:0, onworkspace:w[1]
```

```
# windowrule = bordersize 0, floating:0, onworkspace:f[1]
```

```
# windowrule = rounding 0, floating:0, onworkspace:f[1]
```

```
# See https://wiki.hyprland.org/Configuring/Dwindle-Layout/ for more
```

```
dwindle {
```

```
    pseudotile = true # Master switch for pseudotiling. Enabling is bound to mainMod + P in the keybinds section below
```

```
    preserve_split = true # You probably want this
```

```
}
```

```
# See https://wiki.hyprland.org/Configuring/Master-Layout/ for more
```

```
master {
```

```

new_status = master
}

# https://wiki.hyprland.org/Configuring/Variables/#misc
#misc {
# force_default_wallpaper = -1 # Set to 0 or 1 to disable the anime mascot #wallpapers
# disable_hyprland_logo = false # If true disables the random hyprland logo / #anime girl background. :(
#}

#####
### INPUT ###
#####

# https://wiki.hyprland.org/Configuring/Variables/#input
input {
kb_layout = us,ru
kb_variant =
kb_model =
kb_options = grp:win_space_toggle
kb_rules =

follow_mouse = 1

sensitivity = 0 # -1.0 - 1.0, 0 means no modification.

touchpad {
natural_scroll = false
}
}

# https://wiki.hyprland.org/Configuring/Variables/#gestures
gestures {
workspace_swipe = false
}

# Example per-device config
# See https://wiki.hyprland.org/Configuring/Keywords/#per-device-input-configs for more
device {
name = epic-mouse-v1
sensitivity = -0.5
}

#####
### KEYBINDINGS ###
#####

# See https://wiki.hyprland.org/Configuring/Keywords/
$mainMod = SUPER # Sets "Windows" key as main modifier

# Example binds, see https://wiki.hyprland.org/Configuring/Binds/ for more

```

```
bind = $mainMod, Q, exec, $terminal
bind = $mainMod, C, killactive,
bind = $mainMod, M, exit,
bind = $mainMod, E, exec, $fileManager
bind = $mainMod, V, togglefloating,
bind = $mainMod, R, exec, $menu
bind = $mainMod, P, pseudo, # dwindle
bind = $mainMod, J, togglesplit, # dwindle
bind = $mainMod SHIFT, C, exec, codium
bind = $mainMod SHIFT, F, exec, firefox
bind = $mainMod SHIFT, T, exec, Telegram
bind = $mainMod SHIFT, ESCAPE, exec, gnome-system-monitor
bind = , Print, exec, grim -g "$($slurp)" - | wl-copy
bind = $mainMod, Space, exec, ~/.config/scripts/change_language.sh
bind = $mainMod SHIFT, M, exec, ~/maple2022/bin/xmaple
bind = $mainMod, X, exec, wlogout
bind = $mainMod SHIFT, N, exec, nekoray
bind = $mainMod SHIFT, D, exec, discord
bind = $mainMod SHIFT, S, exec, steam
#bind = $mainMod, W, exec, ~/.config/scripts/random_wallpaper
#bind = $mainMod SHIFT, P, exec, scrpy
bind = $mainMod SHIFT, Y, exec, yandex-music
```

*# Move focus with mainMod + arrow keys*

```
bind = $mainMod, left, movefocus, l
bind = $mainMod, right, movefocus, r
bind = $mainMod, up, movefocus, u
bind = $mainMod, down, movefocus, d
```

*# Switch workspaces with mainMod + [0-9]*

```
bind = $mainMod, 1, workspace, 1
bind = $mainMod, 2, workspace, 2
bind = $mainMod, 3, workspace, 3
bind = $mainMod, 4, workspace, 4
bind = $mainMod, 5, workspace, 5
bind = $mainMod, 6, workspace, 6
bind = $mainMod, 7, workspace, 7
bind = $mainMod, 8, workspace, 8
bind = $mainMod, 9, workspace, 9
bind = $mainMod, 0, workspace, 10
```

*# Move active window to a workspace with mainMod + SHIFT + [0-9]*

```
bind = $mainMod SHIFT, 1, movetoworkspace, 1
bind = $mainMod SHIFT, 2, movetoworkspace, 2
bind = $mainMod SHIFT, 3, movetoworkspace, 3
bind = $mainMod SHIFT, 4, movetoworkspace, 4
bind = $mainMod SHIFT, 5, movetoworkspace, 5
bind = $mainMod SHIFT, 6, movetoworkspace, 6
bind = $mainMod SHIFT, 7, movetoworkspace, 7
bind = $mainMod SHIFT, 8, movetoworkspace, 8
bind = $mainMod SHIFT, 9, movetoworkspace, 9
bind = $mainMod SHIFT, 0, movetoworkspace, 10
```

```

# Example special workspace (scratchpad)
bind = $mainMod, TAB, togglespecialworkspace, magic
bind = $mainMod SHIFT, TAB, movetoworkspace, special:magic

# Scroll through existing workspaces with mainMod + scroll
bind = $mainMod, mouse_down, workspace, e+1
bind = $mainMod, mouse_up, workspace, e-1

# Move/resize windows with mainMod + LMB/RMB and dragging
bindm = $mainMod, mouse:272, movewindow
bindm = $mainMod, mouse:273, resizewindow

# Laptop multimedia keys for volume and LCD brightness
bindel = ,XF86AudioRaiseVolume, exec, wpctl set-volume -l 1 @DEFAULT_AUDIO_SINK@ 5%+
bindel = ,XF86AudioLowerVolume, exec, wpctl set-volume @DEFAULT_AUDIO_SINK@ 5%-
bindel = ,XF86AudioMute, exec, wpctl set-mute @DEFAULT_AUDIO_SINK@ toggle
bindel = ,XF86AudioMicMute, exec, wpctl set-mute @DEFAULT_AUDIO_SOURCE@ toggle
bindel = ,XF86MonBrightnessUp, exec, brightnessctl -e4 -n2 set 5%+
bindel = ,XF86MonBrightnessDown, exec, brightnessctl -e4 -n2 set 5%-

# Requires playerctl
bindl = , XF86AudioNext, exec, playerctl next
bindl = , XF86AudioPause, exec, playerctl play-pause
bindl = , XF86AudioPlay, exec, playerctl play-pause
bindl = , XF86AudioPrev, exec, playerctl previous

#####
### WINDOWS AND WORKSPACES ###
#####

# See https://wiki.hyprland.org/Configuring/Window-Rules/ for more
# See https://wiki.hyprland.org/Configuring/Workspace-Rules/ for workspace rules

# Example windowrule
#windowrule = float,class:^(kitty)$,title:^(kitty)$

# Ignore maximize requests from apps. You'll probably like this.
windowrule = suppresseset maximize, class:.*

# Fix some dragging issues with XWayland
windowrule = nofocus,class:^(xwayland:1,floating:1,fullscreen:0,pinned:0

# Ignore maximize requests from apps. You'll probably like this.
windowrule = suppresseset maximize, class:.*

# Fix some dragging issues with XWayland
windowrule = nofocus,class:^(xwayland:1,floating:1,fullscreen:0,pinned:0

# nautilus
windowrule = float, class:org.gnome.Nautilus

```



```

windowrule = size 50% 50%, class:org.gnome.Nautilus

# Pavucontrol
windowrule = float, class:org.pulseaudio.pavucontrol
windowrule = size 35% 30%, class:org.pulseaudio.pavucontrol
windowrule = move 0% 10% , class:org.pulseaudio.pavucontrol
windowrule = pin, class:org.pulseaudio.pavucontrol

# blueman-manager
windowrule = float, class:blueman-manager
windowrule = size 35% 30%, class:blueman-manager
windowrule = move 0% 10% , class:blueman-manager
windowrule = pin, class:blueman-manager

# nmtui

windowrule = float, class:Nmtui
windowrule = move 50% 50% , class:Nmtui

# Picture in picture
windowrule = float, title:Picture-in-Picture
windowrule = pin, title:Picture-in-Picture

# Nekoray
windowrule = workspace 7, class:nekobox

# Telegram
windowrule = workspace 8, class:org.telegram.desktop

# Ghostty
windowrule = workspace 9, class:com.mitchellh.ghostty
windowrule = opacity 0.9, class:com.mitchellh.ghostty

# dialog window
windowrule = float, title:.*dialog.*
windowrule = float, title:.*Dialog.*

# Yandex music
windowrule = workspace 10, class:yandex-music
windowrule = opacity 0.8, class:yandex-music

```

## Приложение Б. Конфигурация Waybar

```

{
  "layer": "top",
  "position": "top",
  "modules-left": ["custom/eye", "pulseaudio", "pulseaudio#input", "cpu", "memory", "bluetooth", "network"],
  "modules-center": ["clock", "custom/kbdlayout"],
  "modules-right": ["tray", "battery", "backlight", "custom/power"],

  "bluetooth": {

```

```
"format": "□ {status}",
"format-connected": "□ {device_alias}", // Иконка и имя подключенного устройства
"format-disabled": "□ Off", // Иконка для выключенного Bluetooth
"tooltip-format": "Bluetooth: {status}\nDevice: {device_alias}",
"on-click": "blueman-manager" // Запуск Bluetooth-менеджера при клике
},
```

```
"custom/eye": {
"format": "□",
"tooltip": false,
"on-click": "nautilus ."
},
```

```
"pulseaudio": {
"format": "{volume}% □",
"format-muted": "□ {volume}%",
"tooltip": false,
"on-click": "pavucontrol"
},
```

```
"pulseaudio#input": {
"format": "{format_source}",
"format-source": "\uf130 {volume}%",
"format-source-muted": "\uf131 off",
"on-click": "pactl set-source-mute @DEFAULT_SOURCE@ toggle",
"on-scroll-up": "pactl set-source-volume @DEFAULT_SOURCE@ +5%",
"on-scroll-down": "pactl set-source-volume @DEFAULT_SOURCE@ -5%",
"scroll-step": 5,
"on-click-right": "pavucontrol"
},
```

```
"network": {
"format-wifi": "{ssid}",
"format-ethernet": "□ LAN",
"format-disconnected": "□ Нет сети",
"tooltip": false,
"on-click": "alacritty -e nmtui"
},
```

```
"cpu": {
"format": "\ue266 {usage}%",
"tooltip": false
},
```

```
"memory": {
"format": "\uf31b {used}%",
"tooltip": false
},
```

```
"backlight": {
```

```

"format": "□ {percent}%",
"tooltip": false
},

"battery": {
"format": "□ {capacity}%",
"format-charging": "⚡ {capacity}%",
"tooltip": false
},

"clock": {
"format": "{:%H:%M %d/%m/%Y}",
"tooltip-format": "<tt><small>{calendar}</small></tt>",
"calendar": {
"mode": "month",
"mode-mon": 1
},
"on-click-right": "firefox calendar.google.com/calendar/u/0/r"
},

"custom/kbdlayout": {
"exec": "bash ~/.config/waybar/scripts/kb_layout.sh",
"interval": 1,
"format": "□ {} ",
"tooltip": false
},

"custom/power": {
"format": "□ ", // Иконка кнопки (можно заменить на другую, например, из Font Awesome)
"on-click": "wlogout", // Запуск wlogout при клике
"tooltip": "Power Menu"
},

"tray": {
"spacing": 10,
"show-passive": true
},

}

```

## Приложение В. Стилизация Waybar

```

* {
border: none;
border-radius: 8px;

```

```
font-family: JetBrainsMono Nerd Font, sans-serif;
font-size: 14px;
min-height: 0;
padding: 0 10px;
margin: 3px;
color: #ffffff;
background-color: transparent;
}
```

```
window#waybar {
background: rgba(51,255,255,0.15);
}
```

```
#custom-eye,
#pulseaudio,
#pulseaudio.input,
#custom-networkicon,
#cpu,
#memory,
#temperature,
#battery,
#backlight,
#clock,
#custom-kblayout,
#custom-power,
#bluetooth,
#network {
background-color: rgba(55, 146, 179, 0.1);
color: #ffffff;
}
```

```
#network.disconnected {
background-color: #2d2d2d;
color: #ff5555;
}
```

```
#tray {
padding: 0 10px;
}
```

## Приложение Г. Скриншот пользовательской среды

