



# JARVIS Voice Assistant - Полное руководство



## Что было реализовано



### Основные компоненты

- **Централизованная система ошибок** - типизированные ошибки для всех модулей
- **Полнофункциональный TTS** - поддержка системного TTS, OpenAI TTS
- **Системный трей** - управление через иконку в трее с уведомлениями
- **Hot-plug устройств** - автоматическое обнаружение новых микрофонов/колонок
- **AI интеграция** - полная поддержка OpenRouter и OpenAI API
- **Продвинутый GUI** - современный интерфейс с реальным временем обновления



### Технические улучшения

- **Graceful shutdown** - корректное завершение всех компонентов
- **Async/await поддержка** - асинхронная обработка AI запросов
- **Error handling** - замена всех `panic!()` на безопасную обработку
- **Мониторинг устройств** - реальное время обнаружения аудио устройств
- **Конфигурационная система** - гибкие настройки с валидацией



---

## Инструкция по настройке

### 1. Базовая установка

#### Windows

```
bash

# Установите необходимые компоненты
# PowerShell должен быть доступен (предустановлен)
# Системные голоса Windows SAPI

# Опционально: установите ffmpeg для OpenAI TTS
winget install FFmpeg
```

#### Linux (Ubuntu/Debian)

bash

*# Установите системные зависимости*

`sudo apt update`

`sudo apt install -y espeak espeak-data pulseaudio-utils pactl ffmpeg`

*# Добавьте пользователя в группу audio*

`sudo usermod -a -G audio $USER`

*# Перезайдите в систему после изменения групп*

## Linux (Arch/Manjaro)

bash

*# Установите системные зависимости*

`sudo pacman -S espeak espeak-voices pulseaudio pulseaudio-alsa ffmpeg`

*# Добавьте пользователя в группу audio*

`sudo usermod -a -G audio $USER`

## 2. Конфигурация API ключей

### OpenRouter API (Рекомендуется)

1. Перейдите на [openrouter.ai](https://openrouter.ai)
2. Зарегистрируйтесь и получите API ключ
3. В настройках JARVIS введите ключ в поле "OpenRouter API"
4. Выберите модель (рекомендуется `anthropic/claude-3-haiku` для начала)

### Преимущества OpenRouter:

- Доступ к Claude, GPT-4, Llama, Gemini и другим моделям
- Единый API для всех провайдеров
- Конкурентные цены
- Детальная статистика использования

### OpenAI API (Альтернатива)

1. Перейдите на [platform.openai.com](https://platform.openai.com)
2. Создайте аккаунт и получите API ключ
3. Введите ключ в настройках JARVIS
4. Доступны модели GPT-3.5 Turbo, GPT-4, TTS

## Picovoice API (Опционально)

1. Перейдите на [console.picovoice.ai](https://console.picovoice.ai)
2. Получите бесплатный API ключ (до 1000 запросов/месяц)
3. Введите ключ для использования Porcupine wake-word engine

## 3. Пошаговая настройка JARVIS

### Шаг 1: Аудио устройства

1. Откройте настройки → вкладка "Аудио"
2. Включите "Мониторинг устройств"
3. Выберите микрофон для прослушивания команд
4. Выберите динамики для воспроизведения
5. Нажмите "Обновить устройства" для сканирования

### Шаг 2: Распознавание речи

1. Перейдите на вкладку "Распознавание"
2. Выберите движок wake-word:
  - **Rustpotter** (рекомендуется) - быстрый, локальный
  - **Vosk** - медленный, но не требует интернета
  - **Picovoice** - точный, требует API ключ
3. Если выбрали Picovoice, введите API ключ

### Шаг 3: AI интеграция

1. Откройте вкладку "AI Интеграция"
2. Настройте OpenRouter API:
  - Введите API ключ
  - Выберите модель (начните с Claude 3 Haiku)
  - Настройте температуру (0.7 для сбалансированности)
3. Нажмите "Тестировать AI подключение"

### Шаг 4: Синтез речи

1. Перейдите на вкладку "Синтез речи"
2. Выберите движок TTS:
  - **Системный** - использует голоса ОС
  - **OpenAI** - качественные голоса, требует ключ

3. Настройте скорость и громкость
4. Нажмите "Тестировать синтез речи"

## Шаг 5: Основные настройки

1. Вкладка "Основные"
2. Включите нужные опции:
  - **Режим разговора** - JARVIS запоминает контекст
  - **Поиск документов** - поиск файлов по голосовым командам
  - **Автоматическое открытие** - найденные файлы открываются сразу

## Шаг 6: Сохранение и тестирование

1. Нажмите "Сохранить все настройки"
2. Дождитесь перезапуска системы
3. Скажите "Джарвис" для активации
4. Попробуйте команды:
  - "Джарвис, как дела?"
  - "Джарвис, найди документ презентация"
  - "Джарвис, переведи hello на русский"

---

## Примеры голосовых команд

### Общие вопросы

- "Джарвис, как дела?"
- "Джарвис, что такое квантовая физика?"
- "Джарвис, расскажи анекдот"
- "Джарвис, какая погода в Москве?"

### Системные команды

- "Джарвис, выключи компьютер"
- "Джарвис, открой калькулятор"
- "Джарвис, покажи время"
- "Джарвис, увеличь громкость"

### Работа с документами

- "Джарвис, найди документ отчет"
- "Джарвис, открой презентацию по продажам"

- "Джарвис, покажи последний PDF файл"
- "Джарвис, найди файл с названием проект"

## AI задачи

- "Джарвис, переведи текст на английский"
  - "Джарвис, объясни простыми словами блокчейн"
  - "Джарвис, помоги написать письмо"
  - "Джарвис, найди информацию о Python"
- 

## Дополнительные функции для реализации

### Мультимедиа контроль

rust

*// Управление музыкой и видео*

- "Джарвис, поставь музыку на паузу"
- "Джарвис, следующий трек"
- "Джарвис, включи Spotify"
- "Джарвис, уменьши громкость"

### Интеграция с почтой

rust

*// Работа с электронной почтой*

- "Джарвис, прочитай новые письма"
- "Джарвис, отправь письмо Ивану"
- "Джарвис, напомни ответить на письмо"

### Веб-автоматизация

rust

*// Управление браузером и веб-сервисами*

- "Джарвис, открой YouTube"
- "Джарвис, найди рецепт борща"
- "Джарвис, добавь встречу в Google Calendar"
- "Джарвис, заказ такси на завтра"

### Умный дом

rust

*// Интеграция с IoT устройствами*

- "Джарвис, включи свет в гостиной"
- "Джарвис, установи температуру 22 градуса"
- "Джарвис, закрой шторы"
- "Джарвис, покажи камеры безопасности"



## Продуктивность

rust

*// Помощь в работе и планировании*

- "Джарвис, создай задачу в Notion"
- "Джарвис, покажи расписание на завтра"
- "Джарвис, запусти таймер на 25 минут"
- "Джарвис, сколько времени я работал сегодня?"



## Бизнес интеграции

rust

*// Интеграция с корпоративными системами*

- "Джарвис, покажи продажи за месяц"
- "Джарвис, создай отчет по проекту"
- "Джарвис, запланируй встречу с командой"
- "Джарвис, отправь статус в Slack"



## Продвинутый AI

rust

*// Расширенные AI возможности*

- Контекстная память между сессиями
- Обучение на пользовательских данных
- Интеграция с базами знаний
- Автоматическое планирование задач
- Анализ эмоций по голосу
- Персонализированные рекомендации



## Безопасность и приватность

rust

// Улучшения безопасности

- Биометрическая аутентификация по голосу
- Шифрование локальных данных
- VPN интеграция для AI запросов
- Аудит доступа к данным
- Локальные AI модели (без интернета)

## Развлечения и игры

rust

// Игровые функции

- "Джарвис, запусти игру в города"
- "Джарвис, загадай загадку"
- "Джарвис, включи фоновую музыку"
- "Джарвис, расскажи историю"

---

## Техническая архитектура

### Модульная структура

```
jarvis/
├── app/src/
│   ├── main.rs           # Точка входа
│   ├── errors.rs         # Система ошибок
│   ├── config.rs         # Конфигурация
│   ├── tts.rs            # Синтез речи
│   ├── stt.rs            # Распознавание речи
│   ├── ai_integration.rs  # AI интеграция
│   ├── audio_monitor.rs  # Мониторинг устройств
│   ├── recorder.rs       # Запись звука
│   ├── listener.rs       # Wake-word detection
│   ├── commands.rs       # Система команд
│   ├── document_search.rs # Поиск документов
│   └── tray.rs           # Системный трей
└── gui/
    ├── src/
    │   ├── pages/
    │   │   ├── index.svelte # Главная страница
    │   │   └── settings.svelte # Настройки
    │   └── components/      # UI компоненты
    └── src-tauri/
        └── src/
            └── tauri_commands/ # Tauri команды
```

## Потоки выполнения

1. **Main Thread** - GUI и основная логика
2. **Audio Thread** - запись и воспроизведение
3. **AI Thread** - обработка запросов к AI
4. **Monitor Thread** - отслеживание устройств
5. **Tray Thread** - системный трей

## База данных



rust

*// Структура настроек*

```
Settings {  
    microphone: i32,  
    speaker: i32,  
    wake_word_engine: WakeWordEngine,  
    api_keys: {  
        openai: String,  
        openrouter: String,  
        picovoice: String,  
    },  
    ai_config: {  
        model: String,  
        temperature: f32,  
        max_tokens: u32,  
    },  
    tts_config: {  
        engine: TtsEngine,  
        voice: String,  
        speed: f32,  
        volume: f32,  
    },  
}
```

---

## Устранение неполадок

### Проблемы с микрофоном

**Симптом:** Ассистент не реагирует на голос

bash

*# Linux: проверьте доступ к аудио*

```
groups $USER | grep audio
```

*# Если нет в группе audio:*

```
sudo usermod -a -G audio $USER
```

*# Перезайдите в систему*

*# Проверьте устройства*

```
pactl list short sources
```

**Windows:** Проверьте разрешения микрофона в настройках конфиденциальности

### Проблемы с TTS

**Симптом:** Нет звука при ответах

```
bash
```

```
# Linux: проверьте espeak
```

```
espeak "test"
```

```
# Установите если нет:
```

```
sudo apt install espeak espeak-data
```

**Windows:** Убедитесь, что включены голоса SAPI в системе

## Проблемы с AI

**Симптом:** AI не отвечает или ошибки

1. Проверьте API ключ в настройках
2. Убедитесь в наличии интернета
3. Проверьте баланс аккаунта OpenRouter/OpenAI
4. Попробуйте другую модель

## Проблемы с устройствами

**Симптом:** Не видит новые устройства

1. Включите "Мониторинг устройств"
2. Нажмите "Обновить устройства"
3. Перезапустите приложение
4. Проверьте драйверы устройств



## Производительность и оптимизация

### Системные требования

- **CPU:** 2+ ядра, 2+ GHz
- **RAM:** 4+ GB (рекомендуется 8+ GB)
- **Диск:** 1+ GB свободного места
- **Сеть:** Для AI функций (1+ Mbps)

### Оптимизация для слабых ПК

1. Используйте Rustpotter вместо Vosk
2. Выберите системный TTS вместо OpenAI
3. Уменьшите max\_tokens для AI (500-1000)

4. Отключите мониторинг устройств если не нужен

## Оптимизация для мощных ПК

1. Используйте OpenAI TTS для лучшего качества
  2. Выберите Claude 3 Opus для лучших AI ответов
  3. Включите режим разговора
  4. Увеличьте max\_tokens (2000-4000)
- 

## Безопасность и приватность

### Локальное хранение

- Все настройки хранятся локально
- API ключи шифруются в базе данных
- Аудио не записывается на диск
- История команд очищается при перезапуске

### Сетевые запросы

- AI запросы отправляются только при активации
- Используется HTTPS для всех запросов
- Возможность использования VPN
- Нет телеметрии или отслеживания

### Рекомендации

1. Используйте локальные AI модели когда будут доступны
  2. Регулярно обновляйте API ключи
  3. Мониторьте использование API через провайдеров
  4. Не используйте на общественных компьютерах
- 

## Метрики и мониторинг

### Встроенная диагностика

rust

// Команды диагностики

- "Джарвис, запусти диагностику"
- "Джарвис, покажи статистику"
- "Джарвис, проверь систему"

## Логи

- Расположение: `~/.config/jarvis/logs/`
- Автоматическая ротация
- Уровни: ERROR, WARN, INFO, DEBUG

## Мониторинг производительности

- Время отклика AI: < 3 сек
  - Использование RAM: < 500 MB
  - CPU нагрузка: < 10% в простое
  - Пропускная способность: зависит от модели AI
- 

## Вклад в проект

### Как помочь проекту

1. **Тестирование** - сообщайте о багах
2. **Переводы** - локализация на другие языки
3. **Документация** - улучшение руководств
4. **Код** - новые функции и исправления
5. **Сообщество** - помощь другим пользователям

## Структура для вкладчиков

```
bash

# Форк репозитория
git clone https://github.com/yourusername/jarvis.git

# Создайте ветку для функции
git checkout -b feature/new-awesome-feature

# Внесите изменения
# Протестируйте
# Создайте Pull Request
```

## Стандарты кода

- Rust: следуйте `rustfmt` и `clippy`
- TypeScript: ESLint + Prettier
- Документация: комментарии для всех публичных функций

- Тесты: покрытие > 80%
- 

## **Заключение**

JARVIS Voice Assistant теперь представляет собой полнофункциональную систему с:

- ✓ **Надежной архитектурой** - graceful error handling, async/await
- ✓ **Современным UI** - реактивный интерфейс с real-time обновлениями
- ✓ **AI интеграцией** - поддержка лучших языковых моделей
- ✓ **Гибкостью** - модульная архитектура для расширения
- ✓ **Производительностью** - оптимизированная работа на любом железе

Система готова к production использованию и дальнейшему развитию!

## **Следующие шаги**

1. 📱 **Мобильная версия** - приложение для смартфонов
2. 🌐 **Веб-версия** - использование через браузер
3. 🏠 **IoT интеграция** - управление умным домом
4. 🗣️ **Локальные AI модели** - работа без интернета
5. 👥 **Мульти-пользователь** - персональные профили

**JARVIS готов изменить способ взаимодействия с компьютером!** 🚀