

Программа управления "Лес"

Инструкция по установке и эксплуатации

2025

Обзор возможностей программы

"Лес" - это программное обеспечение для автоматического выполнения задач по расписанию с интеграцией MQTT. Программа предназначена для промышленных и автоматизированных систем.

ПО "Лес" предназначено для:

- Промышленной автоматизации
- IoT систем управления
- Системы мониторинга и оповещений

Типовые задачи, решаемые с помощью ПО:

- Выполнение задач по стоп-расписанию
- Отправка MQTT сообщений по расписанию
- Интеграция с базами данных PostgreSQL
- Автоматизация производственных процессов

Основными выгодами от использования являются:

- Высокая надежность и стабильность работы
 - Простота конфигурации через файл переменных окружения
 - Автоматическое переподключение к внешним сервисам
 - Поддержка graceful shutdown для безопасной остановки
-

Системные требования

Для стабильной и эффективной работы ПО рекомендуется использовать следующую конфигурацию:

Частота процессора (CPU): от 1 GHz

Количество ядер процессора (CPU): от 1

Объем оперативной памяти (RAM): от 512 MB

Объем свободного места на диске (HDD): от 100 MB

Операционная система (OS): Linux, Windows, FreeBSD

Внешние зависимости:

- PostgreSQL 12 или выше
 - MQTT брокер (Mosquitto, RabbitMQ или аналогичный)
-

Начало работы

Данный раздел поможет вам быстро установить, настроить и начать работать с ПО.

Перед началом работы, пожалуйста, ознакомьтесь с системными требованиями и убедитесь, что:

- PostgreSQL сервер установлен и запущен
 - MQTT брокер доступен
 - У вас есть скомпилированный исполняемый файл программы
-

Установка

Для установки ПО "Лес" потребуется выполнить следующие шаги:

Linux/FreeBSD

1. Подготовить рабочую директорию

```
mkdir /opt/kemsu-scheduler  
cd /opt/kemsu-scheduler
```

2. Скопировать исполняемый файл

```
cp kemsu-scheduler /opt/kemsu-scheduler/  
chmod +x kemsu-scheduler
```

3. Создать пользователя для службы

```
useradd -r -s /bin/false -d /opt/kemsu-scheduler scheduler  
chown -R scheduler:scheduler /opt/kemsu-scheduler
```

Windows

1. Создать директорию для программы

```
C:\Program Files\KEMSU Scheduler\
```

2. Скопировать исполняемый файл

Поместить `kemsu-scheduler.exe` в созданную директорию

Перед установкой ознакомьтесь с системными требованиями.

Запуск

Ручной запуск

Для запуска ПО "Лес" необходимо:

Linux/FreeBSD:

```
cd /opt/kemsu-scheduler
./kemsu-scheduler
```

Windows:

```
cd "C:\Program Files\KEMSU Scheduler"
kemsu-scheduler.exe
```

Проверка работы

После запуска программа выведет сообщения о:

- Загрузке конфигурации
- Подключении к базе данных
- Подключении к MQTT брокеру
- Загрузке активных расписаний
- Старте планировщика

Для остановки программы используйте `Ctrl+C` или отправьте сигнал `SIGTERM`.

Конфигурация

ПО "Лес" использует файл переменных окружения `.env` для настройки всех параметров работы. При первом запуске программа автоматически создаст файл-шаблонс значениями по умолчанию.

Важно: После создания файла `.env` отредактируйте его в соответствии с вашей инфраструктурой перед повторным запуском программы.

Файл конфигурации

Файл `.env` содержит следующие секции конфигурации:

Конфигурация базы данных

```
# Database Configuration
DB_HOST=localhost           # Адрес сервера PostgreSQL
DB_PORT=5432                # Порт PostgreSQL
DB_USER=postgres            # Имя пользователя БД
DB_PASSWORD=your_password_here # Пароль пользователя БД
DB_NAME=scheduler           # Имя базы данных
DB_SSLMODE=disable          # Режим SSL (disable/require/verify-ca/verify-full)
```

Конфигурация MQTT

```
# MQTT Configuration
MQTT_BROKER=localhost       # Адрес MQTT брокера
MQTT_PORT=1883              # Порт MQTT брокера
MQTT_CLIENT_ID=go-scheduler # Уникальный ID клиента
MQTT_USERNAME=              # Имя пользователя (оставить пустым если не требуется)
MQTT_PASSWORD=              # Пароль (оставить пустым если не требуется)
MQTT_QOS=0                  # Уровень качества сервиса (0, 1, 2)
```

Конфигурация приложения

```
# Application Configuration
SHUTDOWN_TIMEOUT=30         # Таймаут graceful shutdown в секундах
```

Настройка базы данных

Программа автоматически создает необходимые таблицы при первом запуске. Убедитесь, что:

1. PostgreSQL сервер запущен и доступен
2. База данных создана:

```
CREATE DATABASE scheduler;
```

3. Пользователь имеет необходимые права:

```
GRANT ALL PRIVILEGES ON DATABASE scheduler TO postgres;
```

Программа создаст следующие таблицы:

- `schedules` - расписания выполнения задач
- `performers` - исполнители задач
- `performer_types` - типы исполнителей

Настройка MQTT брокера

Для работы с MQTT необходимо:

1. Установить и запустить MQTT брокер (например, Mosquitto)
2. Настроить доступ (если требуется аутентификация)
3. Проверить доступность по указанному адресу и порту

Пример настройки Mosquitto:

```
# Установка Mosquitto (Ubuntu/Debian)
sudo apt-get install mosquitto mosquitto-clients

# Запуск брокера
sudo systemctl start mosquitto
sudo systemctl enable mosquitto
```

Тестирование подключения:

```
# Подписка на тестовый топик
mosquitto_sub -h localhost -t "test/topic"

# Отправка тестового сообщения
mosquitto_pub -h localhost -t "test/topic" -m "Hello World"
```

Автозапуск

Linux (systemd)

Создайте файл службы `/etc/systemd/system/kemsu-scheduler.service` :

```
[Unit]
Description=KEMSU Scheduler Service
After=network.target postgresql.service

[Service]
Type=simple
User=scheduler
Group=scheduler
WorkingDirectory=/opt/kemsu-scheduler
ExecStart=/opt/kemsu-scheduler/kemsu-scheduler
Restart=always
RestartSec=10
StandardOutput=journal
StandardError=journal

# Переменные окружения (если необходимо)
Environment=DB_HOST=localhost
Environment=DB_PORT=5432

[Install]
WantedBy=multi-user.target
```

Активация службы:

```
sudo systemctl daemon-reload
sudo systemctl enable kemsu-scheduler
sudo systemctl start kemsu-scheduler
```

Управление службой:

```
# Статус
sudo systemctl status kemsu-scheduler

# Остановка
sudo systemctl stop kemsu-scheduler

# Перезапуск
sudo systemctl restart kemsu-scheduler
```

```
# Просмотр логов
sudo journalctl -u kemsu-scheduler -f
```

FreeBSD (rc.d)

Создайте файл `/usr/local/etc/rc.d/kemsu_scheduler`:

```
#!/bin/sh

# PROVIDE: kemsu_scheduler
# REQUIRE: NETWORKING postgresql
# KEYWORD: shutdown

. /etc/rc.subr

name="kemsu_scheduler"
rcvar=kemsu_scheduler_enable
command="/opt/kemsu-scheduler/kemsu-scheduler"
command_user="scheduler"
pidfile="/var/run/kemsu_scheduler.pid"

load_rc_config $name
run_rc_command "$1"
```

Активация:

```
chmod +x /usr/local/etc/rc.d/kemsu_scheduler
echo 'kemsu_scheduler_enable="YES"' >> /etc/rc.conf
service kemsu_scheduler start
```

Windows (службы)

Для создания службы Windows используйте `sc` команду или сторонние утилиты:

```
sc create "KEMSU Scheduler" binPath= "C:\Program Files\KEMSU Scheduler\kemsu-
scheduler.exe" start= auto
sc description "KEMSU Scheduler" "KEMSU Scheduler Service for task automation"
sc start "KEMSU Scheduler"
```

Мониторинг и логи

Просмотр логов

Программа выводит структурированные логи с цветовой кодировкой:

- **INFO** (зеленый) - информационные сообщения
- **WARNING** (желтый) - предупреждения
- **ERROR** (красный) - ошибки

Типичные сообщения логов

Успешный запуск:

```
INFO: configuration loaded
INFO: database connection established
INFO: database migrations completed
INFO: MQTT client created
INFO: connected to MQTT broker
INFO: Loaded 5 active schedules
INFO: Scheduler service started
```

Graceful shutdown:

```
INFO: Shutdown signal received
INFO: Starting graceful shutdown...
INFO: Scheduler stopped gracefully
INFO: MQTT client disconnected gracefully
INFO: database connection closed
INFO: Application shut down gracefully
```

Мониторинг состояния

Программа автоматически:

- Переподключается к MQTT брокеру при потере соединения
- Обновляет расписания каждые 5 минут
- Ведет учет времени последнего выполнения задач в базе данных

Решение проблем

Ошибка подключения к базе данных:

```
ERROR: failed to connect to database: connection refused
```

Решение: Проверьте параметры подключения в `.env` и доступность PostgreSQL

Ошибка подключения к MQTT:

```
ERROR: failed to connect to MQTT broker: connection refused
```

Решение: Проверьте адрес и порт MQTT брокера в `.env`

Ошибка cron выражения:

```
ERROR: Failed to add job for schedule 1: expected 5 to 6 fields, found 4
```

Решение: Проверьте корректность cron выражений в базе данных

Контактная информация

ПО "Лес" разрабатывается и поддерживается для КемГУ.

Техническая поддержка

По вопросам функциональности программы обращайтесь к разработчику или системному администратору.

Разработка

Программа создана с использованием языка Go и следующих библиотек:

- GORM для работы с PostgreSQL
- RaHo MQTT для MQTT клиента
- Cron для планирования задач
- Logrus для структурированного логирования