

Инструкция по установке, настройке и эксплуатации ПО “СМГМ”

1. Общие сведения

Настоящий документ содержит указания по установке программного обеспечения “СМГМ”, по первоначальной настройке и эксплуатации.

1.1 Техническая поддержка

Техническая поддержка, а также дополнительное консультирование по вопросам, возникающим в процессе эксплуатации приложения, осуществляются производителем и службой технической поддержки.

1.2 Производитель

ООО «МСС»
121096,
Г.МОСКВА,
УЛ. ВАСИЛИСЫ КОЖИНОЙ,
Д. 13, ПОМЕЩ. 1032
Тел.: 8-495-640-3978
E-mail: info@mssglonass.ru

1.3 Служба технической поддержки

ООО «МСС»
121096,
Г.МОСКВА,
УЛ. ВАСИЛИСЫ КОЖИНОЙ,
Д. 13, ПОМЕЩ. 1032
Тел.: 8-495-640-3978
E-mail: info@mssglonass.ru

1.4 Термины

ПО “СМГМ” (Сервер МСС ГЛОНАСС Мониторинг) - серверная программа, предназначенная для получения, обработки, хранения и

предоставления обработанных и необработанных данных телеметрии с ГЛОНАСС/GPS-трекеров.

СУБД (Система Управления Базами Данных) - программа, позволяющая создать базу данных и манипулировать данными.

Sudo — программа для системного администрирования UNIX-систем, позволяющая делегировать те или иные привилегированные ресурсы пользователям с ведением протокола работы.

/etc/sysctl.conf - файл предварительной загрузки/конфигурации sysctl.

sysctl - утилита, предназначенная для управления параметрами ядра на лету. Позволяет читать и изменять параметры ядра.

/etc/security/limits.conf - позволяет устанавливать ограничения на ресурсы для пользователей, вошедших в систему через подключаемые модули аутентификации Linux.

SSH - сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений.

JSON - текстовый формат обмена данными, основанный на JavaScript.

2. Требования к техническим средствам

Для установки и корректной работы программного обеспечения “СМГМ” необходим сервер / виртуальная машина обладающая следующими минимальными характеристиками:

свободное дисковое пространство — не менее 2 Тб;

процессор: 4 ядра, 2 Гц ;

объем оперативной памяти — не менее 32 Гб;

операционная система — Debian Linux;
СУБД - MySQL Community Edition.

3. Описание процедуры установки ПО

Установка программного обеспечения “СМГМ” является задачей администратора системы. Развертывание программы осуществляется в следующей последовательности:

Шаг 1: Скачайте скрипт `setup.sh` с нашего сервера (ссылка будет предоставлена после покупки ПО “СМГМ”):

```
wget http://your-server.com/path/to/setup.sh -O ~/setup.sh
```

Шаг 2: Предоставление прав пользователя на sudo

1. Войдите в систему от имени суперпользователя (root):

```
su -
```

2. Добавьте пользователя `user` в группу `sudo`:

```
usermod -aG sudo user
```

Шаг 3: Запуск скрипта `setup.sh`

1. Переключитесь на пользователя `user` (если вы еще не под ним):

```
su - user
```

2. Запустите скрипт:

```
sudo ~/setup.sh
```

3. Дождитесь завершения выполнения скрипта.

Шаг 4: Внесение изменений в конфигурационные файлы

1. Добавить в /etc/sysctl.conf

```
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.tcp_max_orphans = 65536
net.ipv4.tcp_fin_timeout = 10
net.ipv4.tcp_keepalive_time = 1800
net.ipv4.tcp_keepalive_intvl = 15
net.ipv4.tcp_keepalive_probes = 5
net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.tcp_synack_retries = 1
net.ipv4.tcp_mem = 50576 64768 98152
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
net.ipv4.tcp_orphan_retries = 0
net.ipv4.tcp_syncookies = 0
net.ipv4.netfilter.ip_conntrack_max = 16777216
net.netfilter.nf_conntrack_max = 16777216
net.ipv4.tcp_timestamps = 1
net.ipv4.tcp_sack = 1
net.ipv4.tcp_congestion_control = htcp
net.ipv4.tcp_no_metrics_save = 1
net.ipv4.route.flush = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.lo.rp_filter = 1
net.ipv4.conf.eth0.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.lo.accept_source_route = 0
net.ipv4.conf.eth0.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.ip_local_port_range = 1024 65535
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_window_scaling = 1
```

```
net.ipv4.tcp_rfc1337 = 1
net.ipv4.ip_forward = 0
net.ipv4.icmp_echo_ignore_broadcasts = 1
net.ipv4.icmp_echo_ignore_all = 1
net.ipv4.icmp_ignore_bogus_error_responses = 1
net.core.somaxconn = 65535
net.core.netdev_max_backlog = 1000
net.core.rmem_default = 65536
net.core.wmem_default = 65536
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
fs.inotify.max_user_watches = 16777216
fs.file-max = 131070
```

Добавить в /etc/security/limits.conf

```
* soft nfile 500000
* hard nfile 500000
root soft nproc 500000
root hard nproc 500000
```

Шаг 5: Перезагрузка сервера

Перезагрузите сервер для применения изменений:

```
systemctl reboot
```

Готово.

4. Внесение изменений в конфигурационный файл

Программа “СМГМ” состоит из нескольких частей, выполняющих различные функции, настройку и управление которыми возможно осуществлять при помощи внесения изменений в файл конфигурации и SSH-запросами.

Части “СМГМ”:

api - часть, отвечающая за http-запросы;

devices - часть, отвечающая за получение данных с трекеров;

translator - часть, отвечающая за трансляцию данных на другие сервера;

cmd - часть, контролирующая работоспособность предыдущих частей;

С помощью SSH возможны также следующие обращения:

all - сразу все части;

main - api и devices.

При запуске “СМГМ” происходит генерация файла **vmsrv.last.json** в директории программы с текущими настройками программы. Для корректной работы ПО на новом сервере может понадобится переопределить пути хранения данных и “логов”. Сделать это можно двумя способами:

- переименовать **vmsrv.last.json** в **vmsrv.json**, удалить из него все что не будет изменяться и поменять на новые значения переопределяемые параметры;
- руками создается **vmsrv.json**, в него вносятся нужные изменения и файл помещается в директорию ПО (пример ниже).

Для редактирования файла Вы можете использовать nano, vim или любой другой редактор на ваш выбор. Например, с помощью nano:

```
nano vmsrv.json
```

На скриншоте ниже Пример настройки общей для всех частей ПО секции “global”:

```
json Копировать код
{
  "global": {
    "dataPath": [
      {
        "dt": "01.01.2010",
        "path": "/backup/VMSrv/points"
      },
      {
        "dt": "01.01.2021",
        "path": "./data/points"
      }
    ]
  }
},
```

Для различных частей общие настройки можно будет переопределить в секции “override”. Как показано в примере Путей, где хранятся данные от трекеров, может быть несколько:

- данные начиная с 01.01.2010
"path": "/backup/VMSrv/points" // хранятся по этому пути
- аналогично с 01.01.2021 по этому:
"path": "./data/points"

В примере ниже настройка секции “override”, которая переопределяет настройки из “global” для различных частей ПО. Для **api** (часть, отвечающая за http-запросы) мы определяем путь хранения “логов”. В примере ниже это:

- "path": "/backup/VMSrv/logs/api"

Также указано время перезапуска **api** в данной части:

- "restart": { "onFixedTimeTimes": ["00:30:00", "01:30:00", "02:30:00", "03:30:00", "04:30:00", "05:30:00", "06:30:00", "18:30:00"] }

Ниже в примере определяется путь хранения “логов” для **devices** и **translator**. Путь для **devices** (часть, отвечающая за получение данных с трекеров) и **translator** (часть, отвечающая за трансляцию данных на другие сервера) соответственно:

- "path": "/backup/VMSrv/logs/devices"
- "path": "/backup/VMSrv/logs/translator"

```
"override": {
  "api": {
    "log": {
      "path": "/backup/VMSrv/logs/api"
    },
    "restart": {
      "onFixedTimeTimes": [
        "00:30:00",
        "01:30:00",
        "02:30:00",
        "03:30:00",
        "04:30:00",
        "05:30:00",
        "06:30:00",
        "18:30:00"
      ]
    }
  },
  "devices": {
    "log": {
      "path": "/backup/VMSrv/logs/devices"
    }
  },
  "translator": {
    "log": {
      "path": "/backup/VMSrv/logs/translator"
    }
  }
}
```

Ниже приводим текст примера `vmsrv.json` целиком с комментариями:

```
{
  "global": {
    // Общая для всех частей секция. Для различных частей общие
    // настройки можно будет переопределить в секции override
    "dataPath": [
      // Пути, где хранятся данные от трекеров. Может быть
      // несколько
      {
        "dt": "01.01.2010",
        // Данные начинаются с
        "path": "/backup/VMSrv/points" // хранятся по этому пути
      },
      {
        "dt": "01.01.2021",
        // Аналогично для
        "path": "./data/points"
      }
    ]
  },
  "override": { // Секция с настройками, которые переопределяют настройки из global для различных
    // частей ПО
    "api": { // Переопределяем настройки api
      "log": { // Настройки лога
        "path": "/backup/VMSrv/logs/api" // Путь, куда будут складываться логи api
      },
      "restart": { // Когда перезапускать api
        "onFixedTimeTimes": [
          "00:30:00",
          "01:30:00",
          "02:30:00",
          "03:30:00",
          "04:30:00",
          "05:30:00",
          "06:30:00",
          "18:30:00"
        ]
      }
    },
    "devices": {
      "log": {
        "path": "/backup/VMSrv/logs/devices"
      }
    },
    "translator": {
      "log": {
        "path": "/backup/VMSrv/logs/translator"
      }
    }
  }
}
```

Сохраните изменения и закройте редактор:

Для nano: нажмите Ctrl+O для сохранения, затем Enter и Ctrl+X для выхода. Перезапустите серверное приложение с помощью скрипта `srv.sh` (подробнее по работе с ним ниже):

```
./srv.sh all start
```

Для получения консультации по другим параметрам пользователь может обратиться с вопросом в службу поддержки:

info@mssglonass.ru.

Просмотр логов

Логи хранятся в папках, указанных в `vmsrv.last.json`. Для их просмотра можно использовать команды `cat`, `grep`.

5. Техническое обслуживание

Для управления серверным приложением используется скрипт `srv.sh`, находящийся в папке с программой. Ниже приведена инструкция по использованию этого скрипта для выполнения различных задач.

Подключитесь к серверу через SSH и перейдите в директорию, где находится скрипт `srv.sh` (находится в папке программы).

Формат вызова скрипта `srv.sh`

Скрипт вызывается следующим образом:

```
./srv.sh <target> <cmd> [<other>]
```

Определение параметров

- `<target>` — определяет, к какой части ПО применяется команда:
 - `api` — часть, отвечающая за http-запросы.
 - `devices` — часть, отвечающая за получение данных с трекеров.

- `translator` — часть, отвечающая за трансляцию данных на другие серверы.
- `cmd` — часть, контролирующая работоспособность предыдущих частей.
- `all` — все части сразу.
- `main` — `api` и `devices`.
- `<cmd>` — команда для выполнения:
 - `start` — запустить или перезапустить.
 - `stop` — остановить.
 - `info` — получить информацию (запущена/не запущена).
 - `build` — пересобрать из исходников.
- `[<other>]` — необязательные параметры для команды `build`:
 - `noclean` — не выполнять полную сборку, пересобрать только измененные части.
 - `nostart` — не запускать после сборки.

6. Описание функционального назначения программы

Программное обеспечение "СМГМ" представляет собой прикладную серверную программу, предназначенную для работы на сервере под управлением операционной системы Linux Debian. В качестве систем управления базами данных (СУБД) применяются open-сурсное решение MySQL Community Edition.

Программа обеспечивает получение телеметрических данных с ГЛОНАСС/GPS-трекеров производителей/протоколов: Навтелеком (протокол NTCB), StandardOEM8, MTA17, Gelix, Egrix, Egrix2, CW701G, ЕГТС, Автограф (в шестнадцатеричном и бинарном представлении), Teltonika, Ruptela, WialonIPS, GoSafe, Escort, Galileo, Naviset, ShtrikhTachoRus, Arnavi, OsmAnd, ScoutOpen, EGTS, cGuard, Wialon Combine. В ходе улучшения работы программы, происходит добавление устройств других производителей и протоколов.

Хранение данных в базе данных: все полученные телеметрические данные должны сохраняться в базе данных для дальнейшей обработки и анализа. "СМГМ" поддерживает авторизацию пользователей по аri-ключу, идентификатору и паролю для выполнения http-запросов к серверу. Помимо необработанных данных, должна быть обеспечена возможность получения обработанных и агрегированных данных:

- статус ТС (движется, работает, стоит, просрочен)
- местоположение
- скорость и направление движения
- список ТС, доступных для отслеживания пользователем
- наименования ТС
- уровень GSM сигнала
- уровень GPS сигнала
- вкл./выкл. блокировки двигателя
- напряжение аккумулятора
- температура аккумулятора
- уровень топлива
- суточный пробег
- суточная время движения
- суточное время стоянки с включенным двигателем
- суточное время стоянки с выключенным двигателем
- средняя и максимальная скорость
- расход топлива
- трек движения
- сливы/заправки.

Данные ответов на http-запросы веб-сервера предоставляются в формате JSON. Информация о GPS-трекерах передается на веб-сервер по http.

Технологический стек. Программа написана в интегрированной среде разработки Qt Creator и скомпилирована с использованием GNU Compiler Collection (GCC). Для реализации используются библиотеки QT в виде динамически подключаемых библиотек, с соблюдением условий лицензии [LGPLv3](#).

Таким образом, ПО "СМГМ" предназначено для сбора, хранения и обработки телеметрических данных с различных ГЛОНАСС/GPS-трекеров, предоставляя пользователю доступ к этим данным через авторизованные http-запросы в формате JSON.

6.1 Архитектура ПО

