|  |
| --- |
| УТВЕРЖДАЮ |
| Управляющий |
| ООО «НетЛайн» |
| \_\_\_\_\_\_\_\_\_\_\_\_ ИП Песковацкий Д.И. |
| «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**

**«THOR-X LITE»**

Руководство системного программиста

Лист утверждения

RU.ЦРСМ.58.29.32-01 32 01-ЛУ

|  |
| --- |
| Руководитель разработки |
| Руководитель отдела развития |
| \_\_\_\_\_\_\_\_\_\_\_\_ Е.В.Баранов |
| «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |
|  |
|  |
| Ответственный исполнитель |
| Ведущий разработчик |
| \_\_\_\_\_\_\_\_\_\_\_\_ И.С.Зарецкий |
| «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |

 УТВЕРЖДЕН

RU.ЦРСМ.58.29.32-01 32 01-ЛУ

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**

**«THOR-X LITE»**

Руководство пользователя

RU.ЦРСМ.58.29.32-01 32 01

Листов ???

АННОТАЦИЯ

Настоящий программный документ представляет собой руководство системного программиста программного обеспечения «THOR-X LITE» (далее – «программа») и содержит сведения о структуре программы, правила установки и настройки программы.

В данном программном документе в разделе «Общие сведения о программе» указаны назначение и функции программы и сведения о технических и программных средствах, обеспечивающих выполнение данной программы, а также требования к персоналу.

В разделе «Структура программы» приведены сведения о структуре программы, ее составных частях, о связях между составными частями и о связях с другими программами.

В разделе «Настройка программы» приведено описание действий по настройке программы на условия конкретного применения.

В разделе «Проверка программы» приведено описание способов проверки, позволяющих дать общее заключение о работоспособности программы.

В разделе «Сообщения системному программисту» указаны тексты сообщений, выдаваемых в ходе выполнения настройки, проверки программы, а также в ходе выполнения программы, описание их содержания и действий, которые необходимо предпринять по этим сообщениям.

Документ подготовлен в соответствии с ГОСТ 19.503-79 - в части структуры и содержания документов, и в соответствии с ГОСТ 19.104-78, ГОСТ 19.103-77, ГОСТ 19.106-78 - в части наименования, оформления и обозначения документов.

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ
	1. Функциональное назначение программы

Программное обеспечение «THOR-X LITE» используется в составе программно-аппаратных комплексов для контроля за дорожным движением и автоматического сбора, обработки и передачи статистической информации о транспортных средствах. Программа обеспечивает распознавание государственных регистрационных знаков (ГРЗ), марок, моделей и типов транспортных средств (ТС) и определение основных параметров интенсивности дорожного движения.

* 1. Эксплуатационное назначение программы

Область применения: программное обеспечение программно-аппаратного комплекса (видеодетектора транспорта) в составе автоматизированных систем мониторинга дорожного движения.

* 1. Состав функций

Программа обеспечивает:

– настройку видеокамеры программно-аппаратного комплекса;

– настройка обособленных зон контроля для раздельной регистрации транспортных средств по этим зонам;

– контроль работы системы распознавания транспортных средств;

– просмотр видеопотока с камер в реальном режиме времени (онлайн);

– просмотр статистических данных по проездам транспортных средств;

– просмотр архивных видеозаписей;

– сохранение метаданных и фотоматериалов результатов фиксации ТС в базе данных;

– передача информации о проездах ТС в ЦОД по запросам REST API.

* 1. Минимальный состав аппаратных средств

Характеристики аппаратных средств для функционирования серверной части программы:

– процессор Intel Core i7 8650U, 4.2 GHz;

– оперативная память 8 Гб, 2400 МГц;

– жесткий диск 512 Гб;

* 1. Минимальный состав программных средств

Системные программные средства, используемые программой, должны быть представлены локализованной версией операционной системы Linux Debian 10 или Ubuntu 18.04.

Дополнительное программное обеспечение, которое требуется для функционирования программы: Postgres, RabbitMQ, Redis, Linux, FFMPEG, opencv, actix-web, diesel, nginx, Keycloak.

* 1. Требования к персоналу (системному программисту)

Выполнение работ по установке и сопровождению программы предполагает соответствие профессионального уровня системного программиста следующим требованиям:

– высокий уровень квалификации и опыта выполнения работ по модернизации, настройке и мониторингу работоспособности комплекса технических средств;

– опыт установки, настройки и сопровождения серверных ОС семейства UNIX;

– опыт установки, модернизации, настройки параметров и сопровождения программного обеспечения СУБД;

– навыки по диагностике типовых неисправностей, настройке локальной компьютерной сети, контроля доступа к сетевым ресурсам.

1. СТРУКТУРА ПРОГРАММЫ
	1. Сведения о структуре программы

Структура программы представляет собой комплекс информационно и технологически взаимосвязанных модулей, позволяющий осуществлять эксплуатацию программы в любом функциональном наборе в зависимости от потребностей.

Программа структурно состоит из следующих модулей:

1) Модуль распознавания – сервис, обеспечивающий распознавание параметров ТС: ГРЗ, тип, марка и модель.

2) Сервис синхронизации – обеспечивает передачу информации о проездах ТС во внешнюю систему по запросу через http, а также для автоматической выгрузки фотоматериалов.

3) Web-приложение – сервис, обеспечивающий доступ к web-интерфейсу программы.

4) Видео прокси-сервер – сервис, обеспечивающий передачу видеоизображения в web-браузер (конвертация видеопотока H.264, получаемого с камеры по протоколу rtsp, в формат MJPEG).

5) Модуль видеозаписи – обеспечивает запись видеоизображения и сохранение в архиве.

1. УСТАНОВКА И НАСТРОЙКА ПРОГРАММЫ
	1. Установка операционной системы

Перед установкой программы предварительно необходимо установить на сервере, на котором планируется установка программы, локализованную версию операционной системы Linux Debian 10 или Ubuntu 18.04 в минимальной конфигурации.

Дальнейшая настройка проводится удаленно с отдельного компьютера с использованием SSH доступа. Необходимо свободное место в разделе /var на сервере не менее 60 Гб.

* 1. Установка специального программного обеспечения

На хосте, с которого будет выполняться запуск скриптов развёртывания, должны быть установлены система управления конфигурациями Ansible и система управления версиями Git.

Установка и настройка указанных систем выполняется в соответствии с инструкциями, изложенными в поставляемой с ним документацией.

Установка необходимых ролей ansible:

1) Создать каталог для ролей ansible и перейти в него:

mkdir -p $HOME/.ansible/roles

cd $HOME/.ansible/roles

2) Склонировать репозитории с ролями ansible:

– роль для первичной инициализации ОС:

git clone git@gitlab.local:devops/ansible/roles/baseinitialization.git

– роль для установки и настройки `Midnight commander`:

git clone git@gitlab.local:devops/ansible/roles/mc.git

– роль для настройки `Docker`:

git clone git@gitlab.local:devops/ansible/roles/docker-hostconfigure.git

– роль для развёртывания сервисов:

git clone git@gitlab.local:devops/ansible/roles/docker-compose-run.git

Провести клонирование тех же ролей web (каждая команда потребует ввода логина и пароля):

git clone http://gitlab.local/devops/ansible/roles/baseinitialization.git

git clone http://gitlab.local/devops/ansible/roles/mc.git

git clone http://gitlab.local/devops/ansible/roles/docker-hostconfigure.git

git clone http://gitlab.local/devops/ansible/roles/docker-composerun.git

3. Склонировать публичный opensource-репозиторий с ролью ansible:

# Роль для установки `Docker`:

git clone https://github.com/geerlingguy/ansible-role-docker.git

3.2.3 Установка системы «Гарпун»

Задание конфигурации в переменных:

1. Перейти в каталог этого репозитория. Все дальнейшие команды должны выполняться из этого каталога.

2. В файле inventory в секции [hosts] задать имя хоста, например, harpoon-01 и сразу после него в одну строку уникальные для этого хоста переменные (ниже в примере переменные записаны на разных строках для наглядности).

Значения всех переменных указываются в одинарных кавычках вместо треугольных скобок.

ansible\_host='<ip-адрес хоста>'

ansible\_ssh\_user='<имя пользователя с правом доступа через ssh>'

ansible\_ssh\_pass='<пароль пользователя из переменной ansible\_ssh\_user>'

ansible\_become\_pass='<пароль суперпользователя>'

Такая строка описывает способ подключения к хосту. В этой секции можно указать несколько хостов одновременно и для каждого из них задать свои адреса, логины и пароли. Ansible будет подключаться параллельно к каждому из них.

При необходимости в секции [global:vars] можно задать переменные, которые будут применяться ко всех хостам, указанным в секции [hosts] :

Метод получения привилегий суперпользователя на целевом хосте. su - для Debian, sudo - для Ubuntu

ansible\_become\_method='<метод>

Порт для доступа через ssh (по умолчанию 22):

ansible\_port='<порт>'

3. Для проверки доступа к хосту выполнить:

ansible -i inventory all -m ping

В случае успешного подключения вывод будет таким:

harpoon-01 | SUCCESS => {"changed": false, "ping": "pong"}

4. Так же в файле inventory в секции [hosts:vars] необходимо задать переменные, значения которых будут использовать компоненты системы:

Временная зона. Применяется во всех контейнерах:

inventory\_timezone='<временная зона>'

Имя и пароль суперпользователя в базе PostgreSQL:

inventory\_postgres\_user='<имя>'

inventory\_postgres\_password='<пароль>'

Имя и пароль для суперпользователя брокера сообщений RabbitMQ:

inventory\_rabbitmq\_default\_user='<имя>'

inventory\_rabbitmq\_default\_pass='<пароль>'

Имя и пароль для суперпользователя Keycloak:

inventory\_keycloak\_user='<имя>'

inventory\_keycloak\_password='<пароль>'

Имя и пароль суперпользователя, а также название базы данных в PostgreSQL, которая используется системой Keycloak:

inventory\_postgres\_auth\_user='<имя>' inventory\_postgres\_auth\_password='<пароль>' inventory\_postgres\_auth\_db='<имя базы данных>'

3.2.4 Первоначальная инициализации хоста

Запустить выполнение плейбука:

ansible-playbook -i inventory playbook-10-initialization-system.yml

3.2.5 Установка инфраструктурных компонентов

Запустить выполнение плейбука:

ansible-playbook -i inventory playbook-20-infrastructure.yml

3.2.6 Настройка Keycloak

Примечание:

– система Keycloak позволяет централизованно управлять доступом пользользователей, выполнять аутентификацию и авторизацию.

После выполнения пункта 3.2.5 «Установка инфраструктурных компонентов» на целевом хосте должна работать система Keycloak. Далее требуется её настройка через web-интерфейс.

1. Перейти по адресу http://<ansible\_host>:<inventory\_keycloak\_port> эти переменные заданы в файле inventory. Адрес целевого хоста задаётся в переменной «ansible\_host» в файле inventory.

2. Перейти в раздел «Administration Console» и авторизоваться в системе с помощью логина и пароля, указанных в переменных «inventory\_keycloak\_user» и «inventory\_keycloak\_password» соответственно в файле «inventory» в секции «Настройки Keycloak»:

3. Создание и настройка «Realm».

3.1 Навести мышь на надпись «Master» в верхнем левом углу над основным меню, появится кнопка «Add realm». Нажать на неё.

3.2 В поле «Name» указать имя «harpoon». Это же имя должно быть указано в переменной «inventory\_keycloak\_realm» в файле «inventory» в секции «Настройки Keycloak». Нажать кнопку «Create».

3.3 Автоматически откроется страница «Realm Settings» с настройками только что созданного «Realm» с названием «Harpoon» в верхнем левом углу. Далее перейти на вкладку «Keys». В строке со значением «RSA256» поля «Algorithm» нажать кноку «Public key». Скопировать полученный публичный ключ и присвоить его переменной «inventory\_keycloak\_public\_key» в файле «inventory» в секции «Настройки Keycloak».

3.4 Перейти на вкладку «Tokens» и в поле «Default Signature Algorithm» выбрать «RSA256». Нажать кнопку «Save» внизу страницы.

4 Создание клиента для работы стороннего ПО с Keycloak.

Клиент в данном случае – это программное обеспечение, которое будет использовать сервер Keycloak для авторизации пользователей.

4.1 При выбранном «Realm» «Harpoon» перейти в раздел «Clients» в левом меню и нажать кнопку «Create» справа вверху.

4.2 В поле «Client ID» указать ID «harpoon-client». Этот же ID должен быть указан в переменной i»nventory\_keycloak\_client\_id» в файле inventory в секции «Настройки Keycloak». Нажать кнопку «Save».

4.3 Автоматически откроется страница с настройками для клиента harpoon-client. Иначе можно перейти в раздел «Clients» в левом меню и в таблице выбрать строку со значением «harpoon-client» в поле «Client ID». Можно нажать на имя самого клиента, либо на кнопку «Edit».

4.4 На вкладке «Settings» в поле «Access Typ»e выбрать «confidential».

4.5 Ниже в поле «Valid Redirect URIs» узакать «/» (т.е. просто косую черту). Нажать кнопку «Save».

4.6 После обновления страницы выбрать вкладку «Credentials», скопировать значение поля «Secret» и присвоить его переменно «inventory\_keycloak\_client\_secret» в файле inventory в секции «Настройки Keycloak».

5 Создание пользователя

Для входа в web-интерфейс Системы контроля движения необходимо создать пользователя на сервере Keycloak. Именно через Keycloak выполняется проверка подлинности пользователей, а также определяются права доступа пользователей к различному функционалу Системы контроля движения.

Для создания пользователя необходимо выполнить:

5.1 При выбранном «Realm» «Harpoon» перейти в раздел «Users» в левом меню и нажать кнопку «Add user» справа вверху.

5.2 На странице «Add use»r в поле «Username» задать логин нового пользователя, для примера укажем «admin». В логине можно использовать кириллицу, регистр букв не важен. Нажать кнопку «Save».

5.3 На странице пользователя вновь добавленного пользователя «admin» перейти на вкладку «Credentials». В полях «Password» и «Password Confirmation» задать пароль для пользователя. В пароле можно использовать кириллицу, регистр букв важен. Переключатель «Temporary» выставить в положение «Off». Нажать кнопку «Set password».

3.2.7 Установка компонента Thor-X3

Запустить выполнение плейбука:

ansible-playbook -i inventory playbook-30-thor-x.yml

3.2.8 Проверка доступа к web-интерфейсу

В случае успешного завершения плейбука система готова к работе.

Перейти по адресу http://<ansible\_host>:<inventory\_thorx3\_port> эти переменные заданы в файле inventory.

3.3 Подключение к Системе

1. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

В случае невозможности программы продолжить выполнение команд оператора, появляются сообщения в текущем окне web-браузера с описанием ошибки, после чего программа возвращается в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных.

Если в процессе работы программа перестает реагировать на действия оператора, то следует обновить страницу web-браузера с URL-адресом программы c помощью нажатия клавиши CTRL+F5 или пиктограммы . Если ошибка не устраняется, то следует обратиться к системному администратору.

ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ

URL – Uniform Resource Locator (единый указатель ресурса)

ГРЗ – государственный регистрационный знак

ОЗУ – оперативная память устройства

ПО – программное обеспечение

ТС – транспортное средство

ЦОД – центр обработки данных

|  |
| --- |
| **ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ** |
| Изм. | Номера листов (страниц) | Всего листов (страниц) в документе | Номер документа | Входящий номер сопроводительного документа и дата | Подпись | Дата |
| измененных | замененных | новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |