

Программный комплекс
Оптическая система анализа ситуационной обстановки

РУКОВОДСТВО АДМИНИСТРАТОРА

Листов 23

2023

АННОТАЦИЯ

В данном документе приведено руководство администратора программного комплекса «Оптическая система анализа ситуационной обстановки» (далее - ОСА).

Документ содержит сведения, касающиеся назначения, основных характеристик, структуре программы, а также условий, необходимых для установки, настройки и функционирования ОСА. Приведено описание действий по установке и настройке на условия конкретного применения (настройка на состав технических и программных средств, выбор функций и др.), описаны способы проверки, позволяющие дать общее заключение о работоспособности программы.

СОДЕРЖАНИЕ

1	ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ	4
1.1	Назначение программы	4
1.2	Условия выполнения программы.....	4
1.2.1	Минимальный состав аппаратных средств	4
1.2.2	Минимальный состав программных средств	4
1.2.3	Требования к персоналу (пользователю)	5
2	СТРУКТУРА ПРОГРАММЫ	7
2.1	Состав системы.....	7
3	НАСТРОЙКА ПРОГРАММЫ	9
3.1	Установка сервера ОСА	9
3.1.1	Установка дополнительного ПО	9
3.1.2	Установка сервисов ОСА Python ShipControl	12
3.1.3	Установка .Net SDK 7.0	12
3.1.4	Установка .Net сервисов ОСА	13
3.2	Удаление сервисов ОСА	14
3.3	Обновление	14
3.4	Установка АРМ ОСА	14
3.5	Обновление	15
3.6	Удаление	15
4	ПРОВЕРКА ПРОГРАММЫ	16
4.1	Проверка программы.....	16
4.2	Обработка отказов	19
5	Сообщения оператору	21
	Список сокращений.....	22

1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

1.1 Назначение программы

Программный комплекс «Оптическая система анализа ситуационной обстановки» предназначен для автоматизации процесса анализа навигационной ситуации вокруг собственного судна, а также обеспечения оператора поста дистанционного управления визуальной информацией об обстановке.

Комплекс устанавливается на суда автономного и дистанционного управления, а также в береговые центры управления и обеспечивает решение следующих задач:

- непрерывный сбор информации с камер, входящих в состав ОСА;
- получение навигационных данных от системы АНС;
- детекция надводных объектов;
- передача данных на АРМ ОСА;
- формирование панорамного изображения надводной обстановки вокруг собственного судна 225/360 градусов;
- отображение на АРМ ОСА изображения в выбранной зоне интереса или с выбранной оператором камеры;
- отображение обнаруженных целей на круговой диаграмме на мониторе АРМ ОСА;
- диагностику собственного состояния;
- настройку собственных параметров;
- логирование событий и сообщений в системе;
- ведение архива и проигрывание архивных данных по выбранной камере и в виде панорамы.

1.2 Условия выполнения программы

1.2.1 Минимальный состав аппаратных средств

Минимальный состав используемых технических (аппаратных) средств:

Видеокамера:

- разрешение 4k;
- видеовыход Ethernet 1Gb;

Сервер, с характеристиками не хуже:

- процессор Intel Core i7, частота – 4к
- оперативная память DDR4 RAM, объем – 64 Гб
- накопитель SSD – 1 Тб
- видеокарта – Nvidia Geforce RTX 3090
- сетевое хранилище

Компьютер с характеристиками не хуже:

- процессор Intel Core i5, частота 2,5 ГГц;
- оперативная память DDR4 RAM, объем 8 Гб;
- накопитель 500 Гб;

Монитор:

- диагональ монитора -- 24 дюйма;
- разрешение экрана -- 1920×1080 пикселей.

1.2.2 Минимальный состав программных средств

Требования к программному обеспечению сервера ОСА.

Операционная система:

- Microsoft Windows Server начиная с версии 2016;
- Microsoft Windows начиная с версии 10;
- Ubuntu начиная с версии 20.04;
- Debian начиная с версии 9;
- Astra Linux SE начиная с версии 1.7;
- Astra Linux CE начиная с версии 2.12;

Платформа Microsoft .Net Core версии 7.0 или новее

- NET Runtime
- ASP.NET Core Runtime

Python 3.9 или новее, следующие пакеты:

- matplotlib
- pydantic
- libopencv
- numpy
- uvicorn
- fastapi
- opencv
- python-multipart
- python-logging-loki
- scipy
- scikit-learn

Postgresql версия 11 или новее

Prometheus версия 2.43 или новее

Grafana-Loki версия 2.8 или новее

Node-Exporter версия 1.5 или новее

Nvidia GPU exporter версия 1.2 или новее

Nvidia Drive версия 525 или новее

2.3 Требования к программной части АРМ ОСА.

Операционная система:

- Microsoft Windows Server начиная с версии 2016;
- Microsoft Windows начиная с версии 10;
- Ubuntu начиная с версии 20.04;
- Debian начиная с версии 9;
- Astra Linux SE начиная с версии 1.7
- Astra Linux CE начиная с версии 2.12

Платформа Microsoft .Net Core версии 7.0 или новее

- NET Runtime
- ASP.NET Core Runtime

1.2.3 Требования к персоналу (пользователю)

Пользователь программы (оператор) должен иметь высшее или среднее профессиональное техническое образование и пройти специальную подготовку в соответствующем объеме на этапе опытной эксплуатации ОСА. Подготовка должна включать в себя получение навыков пользовательской работы со специальным ПО в объеме выработки навыков поддержания их работоспособности.

Пользователь должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

2 СТРУКТУРА ПРОГРАММЫ

2.1 Состав системы

ОСА состоит из:

1. Сервера ОСА, устанавливаемого на судне, предназначенного для сбора, обработки и передачи данных.
2. Видеоархива, предназначенного для хранения журналов логирования, а также архива исходной видеоинформации ОСА.
3. Бортового клиента ОСА, предназначенного для отображения обработанной информации и взаимодействия с оператором на борту судна.
4. Берегового клиента ОСА, предназначенного для отображения обработанной информации и взаимодействия с оператором в ПДУ.

Функциональная схема ОСА представлена на рисунке 1.

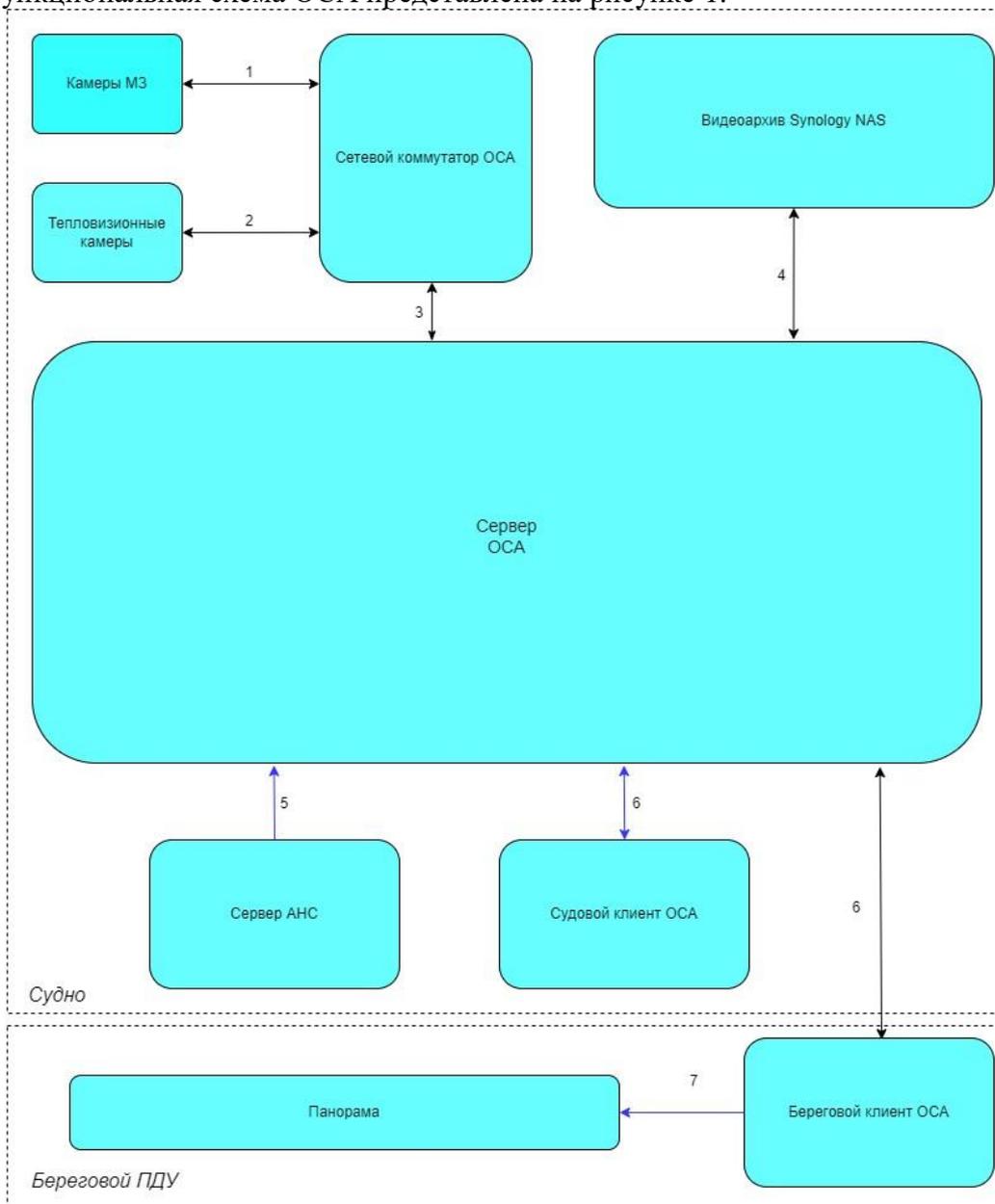


Рисунок 1 – Функциональная схема ОСА

На рисунке 1 приняты следующие обозначения:

1. Видеопоток от камер МЗ: RAW-поток по протоколу GigE, передача необработанного видео, управляющие команды работы дворников, управляющие команды;
2. Видеопоток от тепловизионных камер: RTSP-видеопоток; передача потокового видео; управляющие команды.
3. Видеопотоки от всех камер: передача потоков со всех камер по оптическое линии связи, управляющие команды;
4. Кодированные видеопотоки с камер, панорамы, NMEA-сообщения с навигационными данными, детекции для записи в видеоархив.
5. Навигационные данные о местоположении собственного судна, данные о крене и дифференте, данные по целям РЛС, АИС, отождествлённым целям АНС (UDP-протокол).
6. Видеоданные и метаданные панорамы, детекции, ретранслированные потоки камер, навигационных данных, архивных данных; запросы клиента ОСА на подключение к сервисам (TCP-протокол).
7. Данные панорамы для отображения на мониторе кругового обзора (Цифровой видеосигнал).

В таблице ниже описаны компоненты ОСА и взаимодействующие с ней системы, указанные на приведённой функциональной схеме.

Таблица 1– Компоненты

№	Название	Краткое описание
1.	Судовой клиент ОСА, Береговой клиент ОСА	АРМ ОСА - пользовательское приложение. Обеспечивает отображение информации и взаимодействие с оператором (судоводителем). На схеме присутствует несколько раз. На судне (локальный) и на берегу (в ПДУ).
2.	Сервер ОСА	Набор сервисов, устанавливаемых на судне, предназначено для сбора, обработки и передачи данных.
3.	АНС-сервер	Консольное приложение. Система автоматического управления автономным судном.
4.	Камеры МЗ	Камеры машинного зрения, осуществляющие формирование первичной видео информации: RAW-поток по протоколу GIGE.
5.	Камеры тепловизионные	Камеры тепловизионные, осуществляющие формирование декодированной видео информации в формате RTSP-видеопотока.
6.	Коммутатор ОСА	Коммутатор, осуществляющий сбор всех видеопотоков и передачу на сервер ОСА

3 НАСТРОЙКА ПРОГРАММЫ

3.1 Установка сервера ОСА

Сервер ОСА представляет собой набор сервисов, устанавливаемых на судне для сбора обработки и передачи информации. Управление сервисами производится через командную строку. Конфигурация сервера ОСА, а именно подключение технических средств и настройка взаимодействия с сопряженными системами производится путем редактирования конфигурационных файлов сервисов в формате txt и json.

Порядок установки программного обеспечения:

1. Подготовка дистрибутивов и конфигурационных файлов.
2. Установка дополнительного ПО.
3. Установка ПО на целевую машину.
4. Проверка работоспособности.

Подготовьте набор дистрибутивов и конфигурационных файлов соответствующий целевой системе.

Перед установкой сервисов сервера ОСА необходимо выполнить установку дополнительного ПО.

3.1.1 Установка дополнительного ПО

Установка Prometheus

Необходимо подготовить дистрибутив, выполнить установку и конфигурирование сервиса.

Для установки ПО необходимо использовать командную строку (установка через Терминал). Выполните следующие команды:

```
#sudo wget https://github.com/prometheus/prometheus/releases/download/v2.44.0/Prometheus-2.44.0.linux-amd64.tar.gz
# sudo groupadd --system prometheus
# sudo useradd -s /sbin/nologin --system -g prometheus prometheus
# sudo mkdir /var/lib/prometheus
# sudo mkdir -p /etc/prometheus /rules
# sudo mkdir -p /etc/prometheus /rules.d
# sudo mkdir -p /etc/prometheus /files_sd
# cd /opt/
# tar xvfz Prometheus-<версия>.linux-amd64.tar.gz
# cd prometheus--<версия>-amd64/
# cp prometheus /usr/local/bin/
# cp promtool /usr/local/bin/
# cp -r console_libraries/ /etc/prometheus/
# cp -r consoles /etc/prometheus/
```

Далее необходимо заменить /etc/prometheus/prometheus.yml файл конфигурации, созданный по умолчанию при установке на заранее подготовленный либо выполнить конфигурацию файла в соответствии с параметрами системы и составом технических средств.

Далее необходимо заменить /etc/systemd/system/prometheus.service файл конфигурации, созданный по умолчанию при установке на заранее подготовленный либо выполнить конфигурацию файла в соответствии с параметрами системы и составом технических средств.

```
# chown -R prometheus:prometheus /etc/prometheus/
# chmod -R 755 /etc/prometheus/
# chown -R prometheus:prometheus /var/lib/prometheus/
# chmod -R 755 /var/lib/prometheus/
```

Далее выполните перезагрузку сервисов с помощью команды:

```
#systemctl daemon-reload
```

Далее необходимо проверить установку и запустить сервис, а также включить автоматический перезапуск сервиса при каждой перезагрузке целевой машины.

```
#systemctl start prometheus.service
#systemctl status prometheus.service
#systemctl enable prometheus
```

Установка NodeExporter

Необходимо подготовить дистрибутив, выполнить установку и конфигурирование сервиса.

Для установки ПО необходимо использовать командную строку (установка через Терминал). Необходимо выполнить следующие действия:

Разместить NodeExporter в opt:

```
# cd /opt/
# tar xvfz node_exporter--<версия>..linux-amd64.tar.gz
# cd node_exporter--<версия>..linux-amd64
# mv node_exporter /usr/local/bin
# chown prometheus:prometheus /usr/local/bin/node_exporter
```

Заменить `/etc/systemd/system/node_exporter.service` файл конфигурации, созданный по умолчанию при установке на заранее подготовленный либо выполнить конфигурацию файла в соответствии с параметрами системы и составом технических средств.

Выполнить перезагрузку сервисов.
systemctl daemon-reload

Проверить установку и запустить сервис, а так же включить автоматический перезапуск сервиса при каждой перезагрузке целевой машины.

```
systemctl start node_exporter.service
systemctl status node_exporter.service
systemctl enable node_exporter.service
```

Установка GPUExporter

Необходимо подготовить дистрибутив, выполнить установку и конфигурирование сервиса.

Для установки ПО необходимо использовать командную строку (установка через Терминал). Необходимо выполнить следующие действия:

Разместить GPUExporter в opt:

```
https://github.com/utkuozdemir/nvidia_gpu_exporter/releases/download/v1.2.0/nvidia-gpu-exporter_1.2.0_linux_amd64.deb
```

или скачать:

```
/opt#
```

```
wget
```

```
https://github.com/utkuozdemir/nvidia_gpu_exporter/releases/download/v1.2.0/nvidia-gpu-exporter_1.2.0_linux_amd64.deb
```

Установить GPUExporter с помощью команды;

```
# dpkg -x nvidia-gpu-exporter_1.2.0_linux_amd64.deb nvidia-gpu-exporter_1.2.0_linux_amd64
# mv nvidia-gpu-exporter_1.2.0_linux_amd64/usr/bin/nvidia_gpu_exporter /usr/local/bin/
# chown prometheus:prometheus /usr/local/bin/nvidia_gpu_exporter
```

Заменить `etc/systemd/system/nvidia_gpu_exporter.service` файл конфигурации, созданный по умолчанию при установке на заранее подготовленный либо выполнить конфигурацию файла в соответствии с параметрами системы и составом технических средств.

Выполнить перезагрузку сервисов :

```
systemctl daemon-reload
```

Проверить установку и запустить сервис, а также включить автоматический перезапуск сервиса при каждой перезагрузке целевой машины.

```
#systemctl start nvidia_gpu_exporter.service
#systemctl status nvidia_gpu_exporter.service
#systemctl enable nvidia_gpu_exporter.service
```

Установка PostgreSQL

Необходимо подготовить дистрибутив, выполнить установку и конфигурирование сервиса.

Для установки ПО необходимо использовать командную строку (установка через Терминал). Необходимо выполнить следующие команды:

```
# sudo apt update
# sudo apt dist-upgrade
# apt policy postgresql-11
# sudo apt install postgresql-11
# sudo su - postgres
# psql
# create database videoserverdb ;
# create database osa_archive;
# create user osa with encrypted password '<пароль>';
# grant all privileges on database videoserverdb to osa;
# grant all privileges on database osa_archive to osa;
```

Установка Grafana Loki

Необходимо подготовить дистрибутив, выполнить установку и конфигурирование сервиса.

Для установки ПО необходимо использовать командную строку (установка через Терминал). Необходимо выполнить следующие действия:

Установить Docker.

```
# sudo apt install docker.io
# sudo usermod -aG docker $USER
# exec su - $USER
```

Скачать файлы конфигурации на сервере.

```
wget https://raw.githubusercontent.com/grafana/loki/v2.9.1/cmd/loki/loki-local-config.yaml -O
loki-config.yaml
```

```
docker run --name loki -d -v $(pwd):/mnt/config -p 3100:3100 grafana/loki: <версия> -
config.file=/mnt/config/loki-config.yaml
```

```
wget https://raw.githubusercontent.com/grafana/loki/: <версия>/clients/cmd/promtail/promtail-
docker-config.yaml -O promtail-config.yaml
```

Запустить установку.

```
docker run --name promtail -d -v $(pwd):/mnt/config -v /var/log:/var/log --link loki
grafana/promtail: <версия>.-config.file=/mnt/config/promtail-config.yaml
```

Установка Miniconda3

Необходимо подготовить дистрибутив, выполнить установку и конфигурирование сервиса.

Для установки ПО необходимо использовать командную строку (установка через Терминал). Необходимо выполнить следующие действия:

- Скачать в папку /opt/ и установить.
\$ sudo ./Miniconda3-py-Linux-x86_64.sh -b -p /opt/conda
- Инициализировать Miniconda.
\$ /opt/conda/bin/conda init
- Скопировать заранее подготовленный файл настройки окружения в папку /opt/conda/envs/.
– Установить зависимости.
\$ sudo /opt/conda/bin/conda env create -f shipcontrol.yml
\$ conda activate shipcontrol

Подготовить bash файл для запуска сервиса.

Заменить файл конфигурации /etc/systemd/system/имя_сервиса.service, созданный по умолчанию при установке на заранее подготовленный либо выполнить конфигурацию файла в соответствии с параметрами системы и составом технических средств.

3.1.2 Установка сервисов ОСА Python ShipControl

Необходимо подготовить дистрибутивы, выполнить установку и конфигурирование сервисов.

Для установки ПО необходимо использовать командную строку (установка через Терминал). Необходимо выполнить следующие действия:

- Разместить deb-пакеты на сервере ОСА.
- Установить дополнительные библиотеки.
pythonScripts_osa.pythonScripts.version.deb
pythonScripts_osa.pythonScripts.utilslogging.deb
- Установить python сервисы в следующем порядке:
pythonScripts_osa.pythonScripts.aggregation.deb - сервис агрегации;
pythonScripts_osa.pythonScripts.tracker.deb - сервис трекинга;
pythonScripts_osa.pythonScripts.panorama.deb - сервис панорамы.

Установка выполняется командой:

```
# sudo apt install <путь до deb-пакета>/<python сервис>
```

Заменить /usr/share/ShipControl/ShipControl.PythonScripts.<имя сервиса>/Configuration/configuration.json файл конфигурации, созданный по умолчанию при установке на заранее подготовленный либо выполнить конфигурацию файла в соответствии с параметрами системы и составом технических средств.

Выполнить перезагрузку сервисов :

```
systemctl daemon-reload
```

Далее выполняем действия, описанные ниже для всех установленных сервисов.

Проверить установку и запустить сервис, а также включить автоматический перезапуск сервиса при каждой перезагрузке целевой машины.

```
#systemctl start <python сервис>.service
#systemctl status <python сервис>.service
#systemctl enable <python сервис>.service
```

3.1.3 Установка .Net SDK 7.0

Необходимо подготовить дистрибутив, выполнить установку.

Для установки ПО необходимо использовать командную строку (установка через Терминал). Необходимо выполнить следующие действия:

```
Выполнить установку .Net SDK 7.0
$ sudo wget https://dot.net/v1/dotnet-install.sh
$ sudo chmod +x dotnet-install.sh
```

```
$sudo bash ./dotnet-install.sh -c STS --install-dir /etc/net7/
```

Создать папку /opt/shipcontrol.networks.weights и разместить в нее заранее подготовленную папку production, содержащую модели нейросети.

3.1.4 Установка .Net сервисов ОСА

Необходимо подготовить дистрибутивы, выполнить установку и конфигурирование сервисов.

Для установки ПО необходимо использовать командную строку (установка через Терминал). Необходимо выполнить следующие действия:

Разместить deb-пакеты .NET сервисов на сервере ОСА (набор пакетов зависит от проекта).

Установить .NET сервисы в следующем порядке:

```
shipcontrol.service_<версия>_amd64
shipcontrol.network.service_<версия>_amd64.deb
shipcontrol.monitoring_<версия>_amd64.deb
shipcontrol.monitoring.network_<версия>_amd64.deb
shipcontrol.encoder.service_<версия>_amd64.deb
shipcontrol.archive.service_<версия>_amd64.deb
shipcontrol.archive.orchestrator_<версия>_amd64.deb
shipcontrol.panorama.archive_<версия>_amd64.deb
shipcontrol.aggregation.service_<версия>_amd64.deb
shipcontrol.tracker.service_<версия>_amd64.deb
Разместить следующие папки, подготовленные заранее в /bin:
video.server
retranslator.server
```

Далее выполняем действия, описанные ниже для установленных сервисов в следующем порядке:

```
shipcontrol.encoder.service - энкодер видеопотоков;
shipcontrol.network.service – сервис нейросетей;
shipcontrol.service – сервис ядра ОСА;
retranslator.server - ретранслятор видеопотоков;
shipcontrol.monitoring.service – сервис мониторинга ОСА;
shipcontrol.archive.service – архив детекций;
shipcontrol.archive.orchestrator.service – оркестратор архива;
shipcontrol.panorama.archive.service – сервис архива панорамы;
video.server – видеоархив;
shipcontrol.monitoring.network.service - сервис мониторинга сети
```

Установка выполняется командой:

```
# sudo apt install <путь до deb-пакета>>/<сервис>
```

Заменить /usr/share/ShipControl/ShipControl.<имя сервиса>/Configuration/configuration.json файл конфигурации на заранее подготовленный в соответствии с параметрами системы и составом технических средств конфигурационный файл сервиса вместо файла по умолчанию созданного при установке.

Выполнить перезагрузку сервисов:

```
systemctl daemon-reload
```

Далее выполняем действия, описанные ниже для всех установленных сервисов.

Проверить установку и запустить сервис, а также включить автоматический перезапуск сервиса при каждой перезагрузке целевой машины.

```
#systemctl start <имя сервиса>.service
```

```
#systemctl status <имя сервиса>.service
```

```
#systemctl enable <имя сервиса>.service
```

Далее переходим к установке ОСА клиент.

3.2 Удаление сервисов ОСА

Для удаления ПО необходимо использовать командную строку (через Терминал).

Для всех установленных сервисов выполнить команду:

```
$ sudo apt purge <наименование сервиса>
```

3.3 Обновление

Перед обновлением сервисов ОСА необходимо сохранить все ранее установленные конфигурационные файлы (из папок /usr/share/ ShipControl).

Для обновления ПО необходимо использовать командную строку (установка через Терминал).

Для каждого обновляемого сервиса выполнить:

```
$ sudo apt purge <имя сервиса>
```

Далее установить обновленные сервисы с помощью команды:

```
$ sudo apt install <путь к файлу>/<название файла с учетом расширения>
```

При необходимости выполнить конфигурирование используя заранее подготовленные или сохраненные перед обновлением файлы (описано в п. Установка сервера ОСА).

3.4 Установка АРМ ОСА

АРМ ОСА представляет собой пользовательское приложение, устанавливаемое как на судне, так и на ПДУ, которое обеспечивает отображение информации и взаимодействие с оператором (судоводителем). Конфигурация АРМ ОСА, а именно подключение технических средств и настройка взаимодействия с сопряженными системами производится путем редактирования конфигурационного файла.

Порядок установки программного обеспечения:

1. Подготовка дистрибутивов и конфигурационных файлов.
2. Установка дополнительного ПО.
3. Установка ПО на целевую машину.
4. Проверка работоспособности.

Подготовьте набор дистрибутивов и конфигурационных файлов соответствующий целевой системе.

Перед установкой АРМ ОСА необходимо выполнить установку дополнительного ПО .NET7 SDK.

Для установки ПО необходимо использовать командную строку (установка через Терминал).

Выполнить установку .Net SDK 7.0

```
wget https://dot.net/v1/dotnet - install .sh
```

```
sudo chmod +x dotnet - install .sh
```

```
sudo bash ./ dotnet - install .sh -c STS --install -dir /etc/net7/
```

Выполнить установку пакета АРМ ОСА:

```
$sudo apt install osacient_<версия>_amd64.deb
```

Далее необходимо заменить конфигурационный файл (/usr/share/ShipControl/ShipControl.UI.Visualization.Crossplatform/Configuration/) созданный при установке по умолчанию на предварительно подготовленный конфигурационный файл.

```
cp <путь к файлу>/configuration.json /usr/share/ShipControl/ShipControl.UI.Visualization.Crossplatform/Configuration/
```

ПО после установки необходимо запустить вручную. Для запуска программы дважды

щелкните левой кнопкой мыши по ярлыку программы на рабочем столе



3.5 Обновление

Перед обновлением необходимо сохранить все ранее установленные конфигурационные файлы.

Для обновления ПО необходимо использовать командную строку (установка через Терминал).

Далее выполнить

```
$ sudo apt purge osaclient
```

Далее установить заранее подготовленный пакет с помощью команды.

```
$ sudo apt install <путь к файлу>/osaclient_<версия>_amd64.deb
```

При необходимости выполнить конфигурирование (описано в п. Установка АРМ ОСА) с помощью заранее подготовленного файла конфигурации, либо используя сохраненный ранее.

3.6 Удаление

Для удаления ПО необходимо использовать командную строку (установка через Терминал).

```
$ sudo apt purge <наименование ПО>
```

4 ПРОВЕРКА ПРОГРАММЫ

4.1 Проверка программы

Перед первым запуском программы проверьте наличие предустановленных программных средств, описанных в п.1.2.2 настоящего Руководства администратора.

Перед каждым запуском программы необходимо проверять наличие и работоспособность всех технических средств, описанных в п.1.2.2 настоящего Руководства администратора.

Для запуска программы АРМ ОСА дважды щелкните левой кнопкой мыши по ярлыку



программы на рабочем столе

Далее необходимо проверить наличие связи с судном и доступность сервисов ОСА.

Для запуска программы АРМ ОСА дважды щелкните левой кнопкой мыши по ярлыку



программы на рабочем столе. Откроется главное окно программы (рисунок 2).

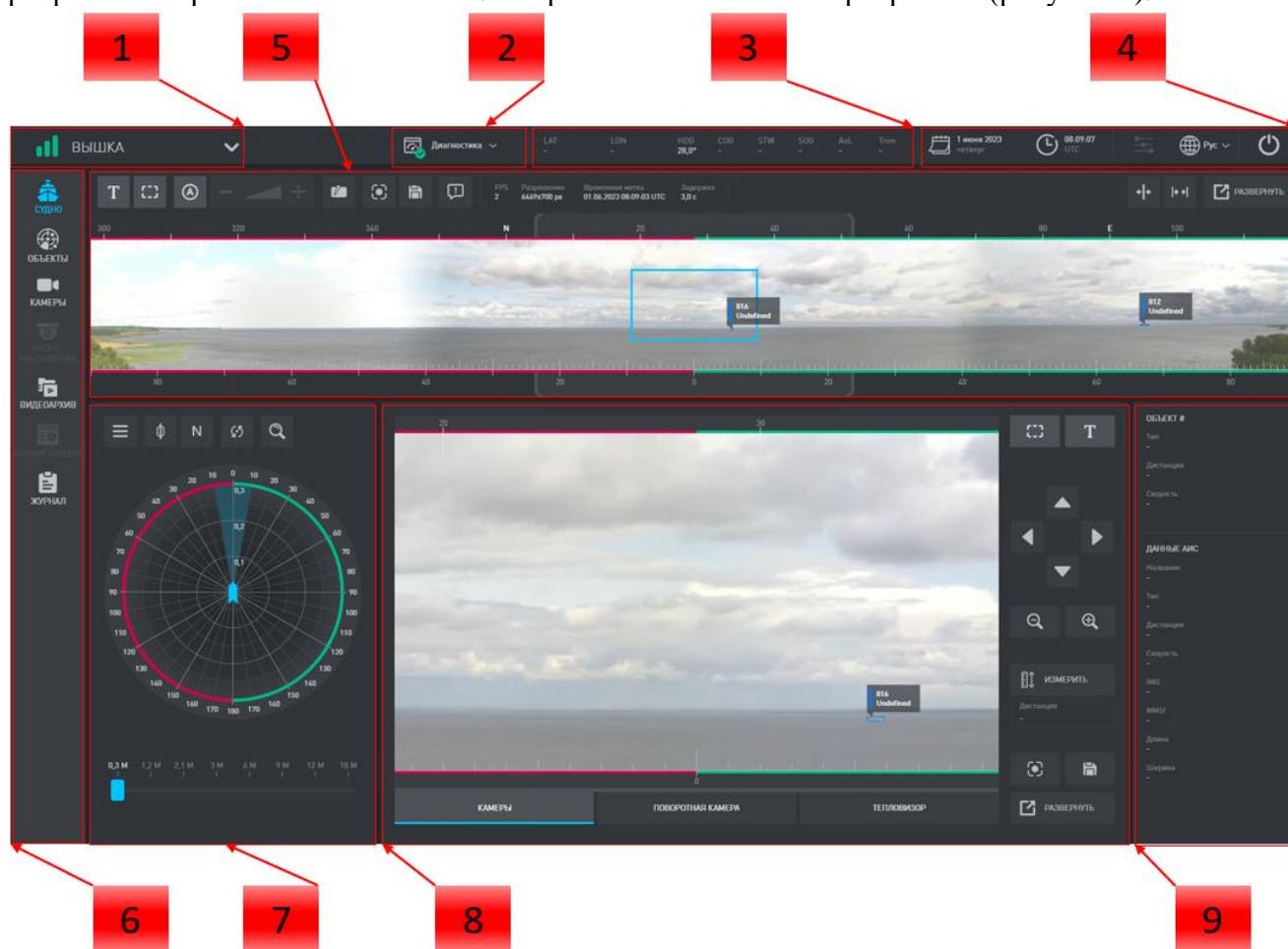


Рисунок 2 – Главное окно программы

Главное окно программы включает в себя следующие элементы:

1. панель выбора судна;
2. панель диагностики;
3. панель навигационных данных судна;
4. панель АРМа;
5. панель панорамы;
6. боковая панель;

7. панель радара;
8. панель окна интереса.
9. панель карточки объекта.

Для проверки связи с судном нажмите на панель в любом месте, появится выпадающий список (рисунок 3), нажатием левой кнопки мыши выбрать нужное судно, выбор будет отмечен .

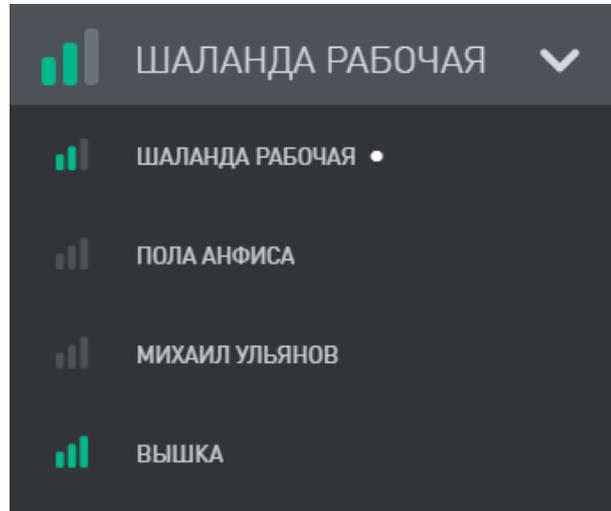
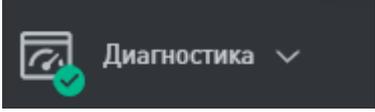


Рисунок 3 – Выбор судна

Иконка  слева от названия судна показывает качество связи с судном. При качестве связи  корабль не доступен (параметры настраиваются в конфигурации).

Для просмотра диагностической информации о работе сервисов сервера ОСА необходимо

нажать на кнопку . Откроется окно диагностической информации Рисунок 4.

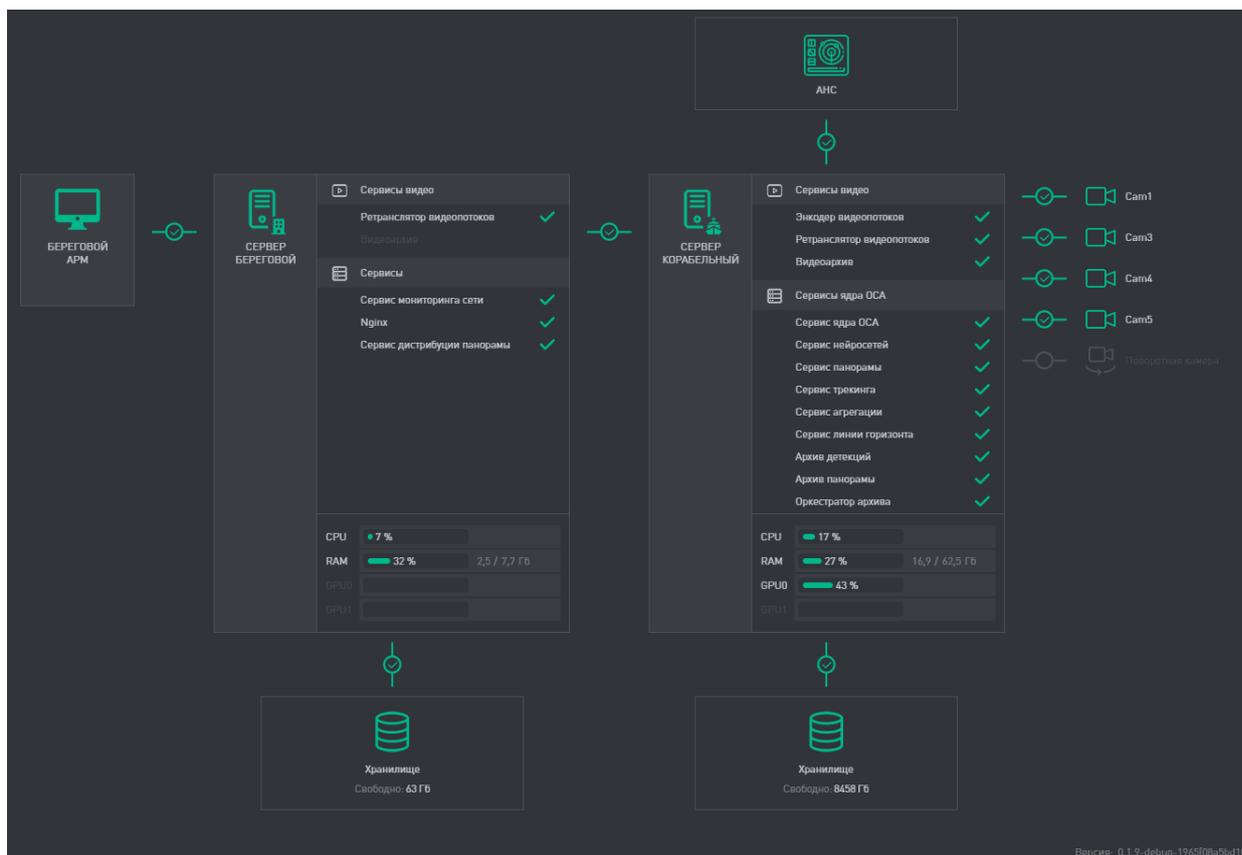


Рисунок 4. Окно диагностики.

В окне диагностики отображается нагрузка на сервер и запущенные сервисы ОСА как на береговом, так и на корабельном сервере, доступность хранилища, остаток места на хранилище и версию клиента.

Сервисы, помеченные белым шрифтом -- сервисы которые участвуют в работе ОСА. Сервисы, помеченные серым шрифтом -- сервисы которые не установлены на корабле или береговом сервере.

Каждый сервис обладают следующими статусами:

- Зеленая галочка  сервис работает без ошибок;
- Желтый восклицательный знак  -- предупреждение, генерируется если за последние n секунд было m Error в сервисе, параметры настраиваются в сервисе мониторинга;
- Красный крест  -- авария, сервис не работает.

При возникновении аварии сервисов, отвечающих за передачу панорамы, панель

диагностики изменит вид на . При отказе сервисов, которые не влияют на передачу панорамы или при множественных ошибках, панель диагностики изменит вид

на .

Таблица 1 – Соответствия индикации приоритета неисправности панели диагностики ПО ОСА

Сервис/Оборудовани	Тип неисправности	Цвет отображения общей неисправности вкладки «Диагностика»
--------------------	-------------------	--

Береговой сервер		
Ретранслятор видеопотоков	Ошибка	Красный
Сервис мониторинга сети	Предупреждение	Желтый
Сервис дистрибьюции панорамы	Ошибка	Красный
Nginx	Ошибка	Красный
Хранилище	Предупреждение	Желтый
Корабельный сервер		
Энкодер видеопотоков	Ошибка	Красный
Ретранслятор видеопотоков	Ошибка	Красный
Видеоархив	Предупреждение	Желтый
Сервис ядра ОСА	Ошибка	Красный
Сервис ядра нейросети	Ошибка	Красный
Сервис панорамы	Ошибка	Красный
Сервис трекинга	Ошибка	Красный
Сервис агрегации	Ошибка	Красный
Сервис линии горизонта	Ошибка	Красный
Архив детекций	Предупреждение	Желтый
Сервис оркестрации	Предупреждение	Желтый
Хранилище	Предупреждение	Желтый
Оборудование		
Камера	Предупреждение	Желтый камера
	Ошибка	Красный – больше 1 камеры
Дальномер	Предупреждение	Желтый
PTZ-камера	Предупреждение	Желтый

При отсутствии проблем в работе необходимых сервисов возможно переходить к эксплуатации ПО в соответствии с Руководством оператора.

4.2 Обработка отказов

Обработка отказов в работе сервисов ОСА.

При наличии предупреждений и аварий в работе сервисов ОСА необходимо на сервере ОСА возможно восстановить его работоспособность одним из следующих способов.

1) Перезапуск сервиса.

Выполнить перезапуск сервиса и проверить статус работы сервиса.

Для этого выполните команду:

```
#systemctl restart <имя сервиса>.service
```

```
#systemctl status <имя сервиса>.service
```

После перезапуска сервис должен перейти в состояние active.

2) Выполнить повторное конфигурирование сервиса, таким же образом, как и при установке (см. п. 3.1.2 и 3.1.3).

3) Выполнить повторную установку сервиса и его конфигурирование (см. п. 3.1.2 и 3.1.3).

Ошибки чтения лицензионного ключа

При возникновении ошибок чтения аппаратного USB-ключа:

“На лицензионном ключе не найден продукт: ОСА и компонент OSA_CLIENT.

Дальнейшая работа приложения не возможна!”

“Лицензионный ключ найден по времени: <Значение времени>”.

“Лицензионный ключ утрачен по времени: <Значение времени>”.

“Время отсутствия лицензионного ключа: <Интервал времени>”.

Необходимо:

1. Убедиться в наличии аппаратного USB-ключа Guardant Sign в USB-порте целевой машины, содержащего необходимые лицензии.
2. Выполните настройку правил для работы аппаратного USB-ключа в соответствии с инструкцией по настройке работы аппаратного USB-ключа Guardant Sign.
3. Сохраните код и текст ошибки, обратитесь к службе поддержки разработчика ПО.

5 СООБЩЕНИЯ ОПЕРАТОРУ

При работе программы на экране могут появляться текстовые сообщения, поясняющие возникшие ситуации, которые требуют дополнительных действий.

Данные сообщения описаны в разделе 3 Руководства оператора ОСА и в разделе 3 данного документа.

Действия, которые необходимо предпринять по этим сообщениям описаны в разделе 4 настоящего документа.

СПИСОК СОКРАЩЕНИЙ

AMD	- Advanced Micro Devices;
АРМ	- автоматизированное рабочее место;
АНС	- автономная навигационная система;
ОСА	- автоматизированная навигационная система.
ПО	- программное обеспечение
ПК	- персональный компьютер
ПДУ	- пункт дистанционного управления

