

Утверждаю
Директор
ООО «Сигма-Софт
Автоматизация»

_____ М.И. Мальцев
“ _____ ” _____ 2023

Программный комплекс «С-Платформа» (S-Platform)

Инсталляция системы

Версия 1.0

Руководство системного программиста

RU.82469608.0001-01 32

Руководитель разработки

Начальник департамента

_____ И.О. Урухин

“ _____ ” _____ 2023

Ответственный исполнитель

Ведущий инженер-программист

_____ В.А. Елизаренко

“ _____ ” _____ 2023

ООО «Сигма Софт»
2023 г.



С-Платформа

Утвержден

RU.82469608.0001-01 32

Программный комплекс «С-Платформа» (S-Platform).

Инсталляция системы

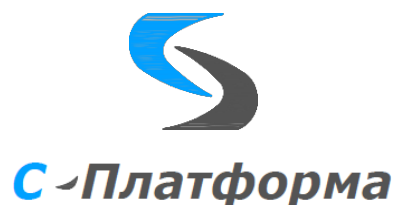
Версия 1.0

Руководство системного программиста

RU.82469608.0001-01 32

Листов 27

ООО «Сигма Софт»
2023 г.



ОГЛАВЛЕНИЕ

1. ПЕРЕЧЕНЬ СОКРАЩЕНИЙ.....	4
2. АННОТАЦИЯ.....	5
3. ОБЩИЕ СВЕДЕНИЯ.....	6
4. Установка ПК «С-Платформа» под ОС Linux.....	7
4.1. Ручная установка сервиса в ОС Linux.....	7
4.2. Ручная установка сервера Приложений в ОС Linux.....	16
4.3. Ручная установка сервера ввода-вывода в ОС Linux.....	21
4.4. Ручная установка модуля «С-Платформа.СНТИ.КЛИЕНТ».....	24
4.5. Ручная установка модуля «АМГО ОПРЧ».....	25
4.6. Автоматизированная установка ПК «С-Платформа».....	26
4.7. Настройка PostgreSQL.....	26

1. ПЕРЕЧЕНЬ СОКРАЩЕНИЙ.

БД	-	база данных
ПО	-	программное обеспечение
СУБД	-	система управления базами данных
ССНТИ	-	система сбора неоперативной технологической информации
АМГО ОПРЧ	-	автоматизированного мониторинга участия. генерирующего оборудования в общем первичном регулировании частоты
СО ЕЭС	-	АО «СО ЕЭС»

2. АННОТАЦИЯ.

Настоящий документ содержит описание процесса инсталляции программного комплекса С-Платформа под управлением операционной системы Astra Linux SE.

В документе описаны два способа инсталляции:

- Ручной
- Автоматизированный

Кроме настоящего документа необходимо также руководствоваться технической и эксплуатационной документацией на ПК С-Платформа:

- ЯКШГ.00067-01 91 01 31 Описание применения.
- ЯКШГ.00067-01 91 01 32 Руководство системного программиста.
- ЯКШГ.00067-01 91 01 33 Руководство программиста.
- ЯКШГ.00067-01 91 01 34 Руководство оператора.
- RU.82469608.0002-01 13. Описание применения (ОПРЧ)
- RU.82469608.0001-01 13. Описание применения (ССНТИ)

3. ОБЩИЕ СВЕДЕНИЯ.

- 3.1. Программный комплекс «С-Платформа» служит для создания многоуровневых диспетчерских и технологических информационно-управляющих систем реального времени.
- 3.2. ПК «С-Платформа» предназначено для функционирования совместно с программной платформой КОТМИ-14 (далее – ПК «КОТМИ-14»).
- 3.3. ПК «КОТМИ-14» используется в качестве базового ядра системы, на основании которого разработан целый ряд коммуникационных протоколов обмена, специализированных сервисов, функциональных технологических программных модулей, что в совокупности с базовым ядром и составляет ПК «С-Платформа».
- 3.4. Серверная часть ПК «С-Платформа» допускает работу без ограничений в среде виртуализации VMWare / Брест.VDI / ECP Veil.
- 3.5. Минимальная аппаратная и программная конфигурация вычислительной системы, необходимая для корректной работы ПК «С-Платформа», включает:
 - центральный процессор, не хуже - Intel i3 / Эльбрус-8С;
 - оперативная память, не менее 4 Гб;
 - объем свободного пространства на жестком диске, не менее 500 Мб;
 - графическая подсистема: разрешение 1280x1024, частота развертки 60 Гц;
 - сетевая подсистема – канал связи Ethernet от 10 Мбит/с;
 - операционная система Microsoft Windows 10 / Astra Linux SE 1.6 / Альт 8 СП / AlterOS;
 - средства ввода – клавиатура, мышь / трекбол.

4. Установка ПК «С-Платформа» под ОС Linux.

4.1. Ручная установка сервиса в ОС Linux

4.1.1. Создание каталогов и копирование файлов.

Скопируйте необходимые для работы сервиса файлы в заранее выбранный каталог. Выбор каталога и его имя зависит от конкретной задачи и не ограничен какими-либо условиями/правилами. Однако, в соответствии с принятыми соглашениями и исходя из опыта эксплуатации комплекса, можно выделить два основных подхода к размещению каталогов и файлов сервиса:

- Размещение папок и файлов в домашнем каталоге пользователя (/home).

Для этого создайте нового пользователя в ОС Linux (здесь и ниже в документации будем считать, что пользователь в системе создан и имеет имя «user») или используйте существующего и добавьте его в группу пользователей (например, «scada») с заранее заданными правами. В домашней папке пользователя создайте каталог с именем «SPlatform» (или любой другой, на ваш выбор), в нём подкаталоги: «Service» и «lib»:

```
$ mkdir /home/user/SPlatform && mkdir /home/user/SPlatform/Service && mkdir /home/user/SPlatform/lib
```

Скопируйте в «lib» файлы инструментальной библиотеки MDX: «libScdMdx_x64.so.X.0.0», «libScdMdx_x64.so.X.0», «libScdMdx_x64.so.X» и «libScdMdx_x64.so» или создайте самостоятельно символические ссылки на библиотеку «libScdMdx_x64.so.X.0.0». Здесь «X» - номер текущей версии библиотеки.

Создайте подкаталог «Bin» в папке «Service» и скопируйте в него следующие файлы: «ScdService_x64», «ScdMonitor_x64» и «ScdMonitor.png». Установите, если это требуется, права на запуск файла (например, такой командой:

```
sudo chmod +x имя_файла).
```

Скопируйте текстовый файл скрипта «scd.service», который используется для регистрации сервиса в ОС Linux, в папку: «/home/user/SPlatform».

- Размещение папок и файлов в каталоге дополнительного программного обеспечения (/opt).

Для этого с правами суперпользователя создайте в папке «/opt» каталог, например, с именем «SPlatform» (или любой другой, на ваш выбор), в нём подкаталоги: «Service» и «lib». Скопируйте в «lib» файлы инструментальной библиотеки MDX: «libScdMdx_x64.so.X.0.0», «libScdMdx_x64.so.X.0», «libScdMdx_x64.so.X» и «libScdMdx_x64.so» или создайте самостоятельно символические ссылки на библиотеку «libScdMdx_x64.so.X.0.0». Здесь «X» - номер текущей версии библиотеки.

Создайте подкаталог «Bin» в папке «Service» и скопируйте в него следующие файлы: «ScdService_x64», «ScdMonitor_x64» и «ScdMonitor.png». Выдайте пользователям необходимые права на доступ к созданным каталогам или добавьте их в соответствующие группы. Установите, если это требуется, права на запуск файлов (например, такой командой:

```
Sudo chmod +x имя_файла).
```

Скопируйте файл скрипта «scd.service», который используется для регистрации сервиса в ОС Linux, в папку: «/opt/SPlatform».

ПРИМЕЧАНИЕ: Не рекомендуется использовать в именах папок кириллицу, а также специальные символы даже если их использование допускается самой ОС.

4.1.2. Регистрация разделяемых библиотек.

После того, как все необходимые папки и файлы успешно созданы, нужно зарегистрировать в кэше разделяемых библиотек ОС модуль инструментальной библиотеки MDX: «libScdMdx_x64.so». Для этого перейдите в каталог «/etc/ld.so.conf» и отредактируйте содержимое одного из файлов с расширением «.conf». Какой именно – принципиального значения не имеет. В случае отсутствия таких файлов, создайте новый текстовый файл с произвольным именем и расширением «.conf». Добавьте в его конец строку содержащую абсолютный путь к разделяемым библиотекам ПК КОТМИ-14: «/home/user/SPlatform/lib». Сохраните файл.

Выполните команду ldconfig с правами суперпользователя:

```
$ sudo ldconfig
```

После чего разделяемые библиотеки будут добавлены в кэш, проверить успешность внесения данных можно командой:

```
$ sudo ldconfig -p
```

Если всё сделано правильно, то в списке вывода можно найти файлы инструментальной библиотеки MDX:

```
$ sudo ldconfig -p | grep libScdMdx_x64
```

```
libScdMdx_x64.so.1.0 (libc6,x86-64) => /home/user/SPlatform/lib/libScdMdx_x64.so.1.0
```

```
libScdMdx_x64.so.1 (libc6,x86-64) => /home/user/SPlatform/lib/libScdMdx_x64.so.1
```

```
libScdMdx_x64.so (libc6,x86-64) => /home/user/SPlatform/lib/libScdMdx_x64.so
```

Теперь при попытке запуска одного из исполняемых файлов, например: «Системный монитор» откроется окно приведенное на рисунке №1.

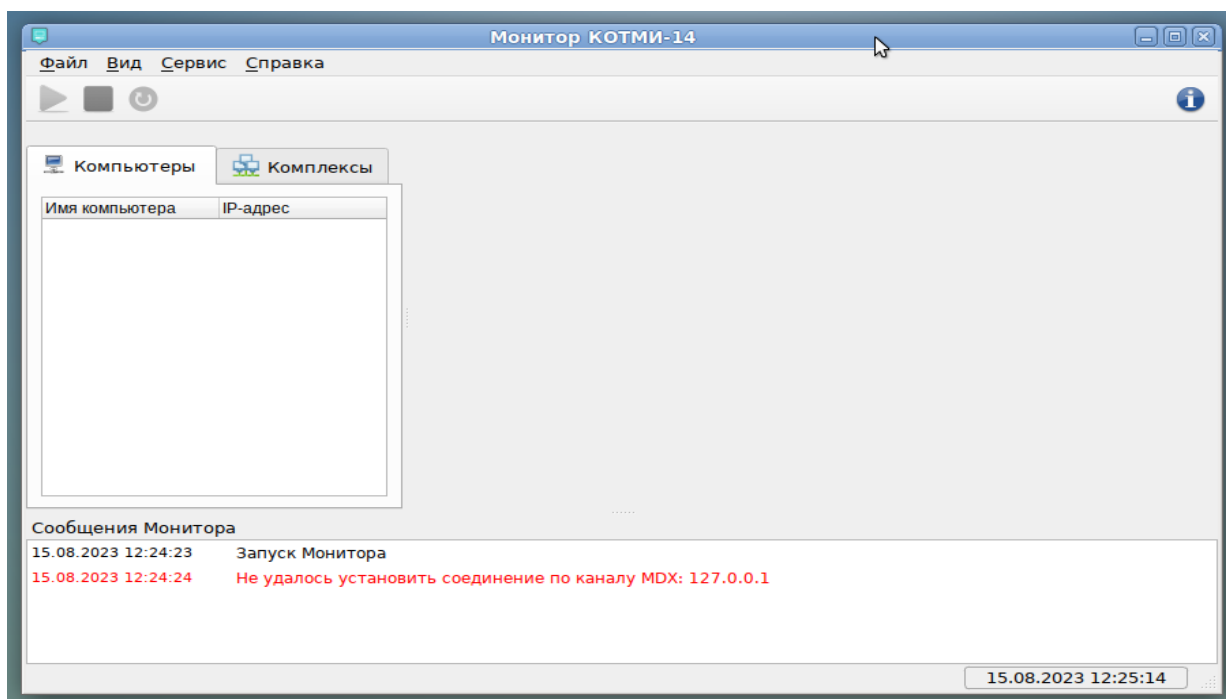


Рис 1. Монитор комплексов.

4.1.3. Установка необходимых пакетов.

Для работы сервиса и монитора КОТМИ-14 требуется наличие в ОС следующих дополнительных библиотек из состава фреймворка Qt5:

libQt5Widgets.so.5, libQt5Gui.so.5, libQt5Network.so.5 u libQt5Core.so.5.

Их можно установить из следующих пакетов:

- Для Alt Linux:

qt5-base-common, libqt5-core, libqt5-widgets, libqt5-gui u libqt5-network.

- Для Astra Linux:

libqt5core5a, libqt5widgets5, libqt5gui5, libqt5core5a u libqt5network5.

Установить пакеты и все необходимые для них зависимости можно через команду «apt-get install имя_пакета», например, так:

- Для Alt Linux:

\$ sudo apt-get install qt5-base-common libqt5-core libqt5-widgets libqt5-gui libqt5-network

- Для Astra Linux:

\$ sudo apt-get install libqt5core5a libqt5widgets5 libqt5gui5 libqt5network5.

4.1.4. Создание хранилища конфигураций.

Хранилище конфигураций — это место на диске (каталог в файловой системе), где находится специальный текстовый файл с именем «*SrvCfg.ini*», он используется сервисом КОТМИ-14 для хранения всех доступных, на локальной машине, конфигураций ПК КОТМИ-14.

Чтобы создать новое хранилище конфигураций, выберите существующий каталог (к примеру: «*/home/user/KOTMI*») или создайте новый:

\$ mkdir /home/user/SPlatform

Создайте в нём новый текстовый файл с именем «*SrvCfg.ini*»:

\$ touch /home/user/SPlatform/SrvCfg.ini

и следующим содержимым:

[Common]

NumConf=1

Path_1=/home/user/SPlatform/Config.cfg

[Password]

1="284371932681020674093584 "

2=B266BF8EAF017031782F

Расшифровка параметров:

- Параметр «*NumConf*» из секции «*Common*»: кол-во конфигураций в хранилище конфигураций;
- Параметр «*Path_1*» из секции «*Common*»: полный путь и имя с расширением к файлу конфигурации «*Config.cfg*» с порядковым номером 1;
- Параметр «1» из секции «*Password*»: хэш №1 для пароля (приведено значение по умолчанию к паролю: «*kotmi14*»);
- Параметр «2» из секции «*Password*»: хэш №2 для пароля (приведено значение по умолчанию к паролю: «*kotmi14*»).

После создания хранилища конфигураций в него можно добавлять новые конфигурации и использовать для дальнейшей регистрации сервиса в системе.

ПРИМЕЧАНИЕ: Несмотря на то, что в файле хранилища конфигураций указывается полный путь к файлу конфигурации (параметр «*Path_No*») сами файлы конфигураций **ОБЯЗАТЕЛЬНО** должны находиться в том же каталоге, что и хранилище конфигураций.

4.1.5. Установка (регистрация) сервиса в системе.

Сервис КОТМИ-14 (имя исполняемого модуля: ScdService_x64) – это специальная программа, оформленная как сервис в ОС Linux. Начиная выполняться в момент старта ОС, сервис позволяет запустить сервер приложений и/или серверные задачи без вмешательства оператора, непосредственно после включения компьютера и загрузки ОС.

Его основными функциями является:

- контроль работоспособности, перезапуск сервера приложений или серверных задач в случае сбоя или аварийной остановки;
- информационный обмен с программой «Монитор серверов КОТМИ-14»: передача данных о серверной конфигурации, активности серверов и серверных программ, содержания log-файлов, списков пользователей, характеристик компьютеров и пр.
- информационный обмен с программой «Конфигуратор серверных комплексов КОТМИ-14» для передачи и приема данных о серверной конфигурации и замене ее, при необходимости, на новую.

Для регистрации сервиса в ОС Linux создайте новый текстовый файл с именем «*scd.service*» (или любым другим):

```
$ touch /home/user/SPlatform/scd.service
```

и содержимым:

```
[Unit]
```

```
Description=SPlatform Control
```

```
Wants=network.target
```

```
After=syslog.target network-online.target
```

```
[Service]
```

```
Type=simple
```

```
Environment="SCDCFG_PATH=/home/user/SPlatform"
```

```
WorkingDirectory=/home/user/SPlatform
```

```
ExecStart=/home/user/SPlatform/Service/Bin/ScdService_x64 $SCDCFG_PATH
```

```
Restart=always
```

```
RestartSec=10
```

```
User=user
```

```
KillMode=process
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Здесь основными параметрами являются:

- Параметр «*User*»: указать пользователя из-под которого происходит запуск сервиса;
- Параметр «*Environment*»: задает путь к файлу хранилища конфигураций;

- Параметр «*WorkingDirectory*»:
указать рабочий каталог сервиса, который задан или как ‘значение по умолчанию’ (*/home/user/.config*) или введен пользователем вручную;
- Параметр «*ExecStart*»:
путь и имя исполняемого файла сервиса вместе с аргументом командной строки – значением параметра «*Environment*»;

Сохраните созданный файл и выполните команду с правами суперпользователя:

```
$ sudo systemctl enable /home/user/SPlatform/scd.service
```

Если всё сделано правильно, сервис успешно регистрируется в системе. Чтобы убедиться в этом и проверить его статус, сначала запустите сервис, как показано ниже:

```
$ sudo systemctl start scd.service
```

Затем запросите статус:

```
$ sudo systemctl status scd.service
```

Результатом должно быть следующее (приведен вывод команды в ОС Astra Linux):

- ***scd.service - SPlatform Control***

Loaded: loaded (/home/user/SPlatform/scd.service; enabled; vendor preset: enabled)

Active: active (running) since Wed 2023-00-00 18:00:00 MSK; 2s ago

Main PID: 13437 (ScdService_x64)

Tasks: 6 (limit: 4915)

CGroup: /system.slice/scd.service

***└─13437 /home/user/SPlatform/Service/Bin/ScdService_x64 /home/user/SPlatform
a62 23 18:00:00 astra systemd[1]: Started SPlatform-14 Control.***

При наличии установленного X-Server основные параметры сервиса и его текущее состояние можно узнать, запустив «Панель управления системой» для ОС Astra Linux или «Центр управления системой» для ОС Alt Linux (Рис №2 и Рис №3).

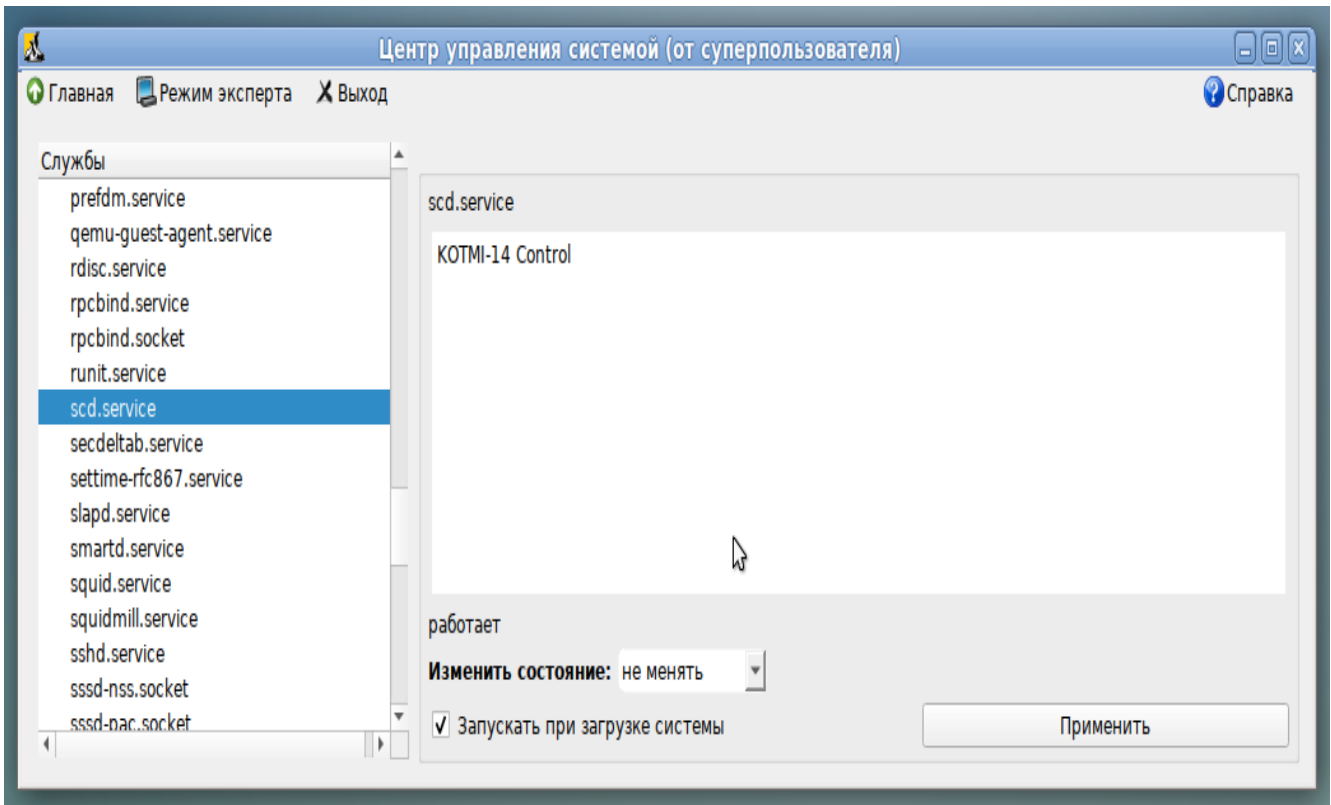


Рис 2. Установленный сервис в ОС Alt Linux.

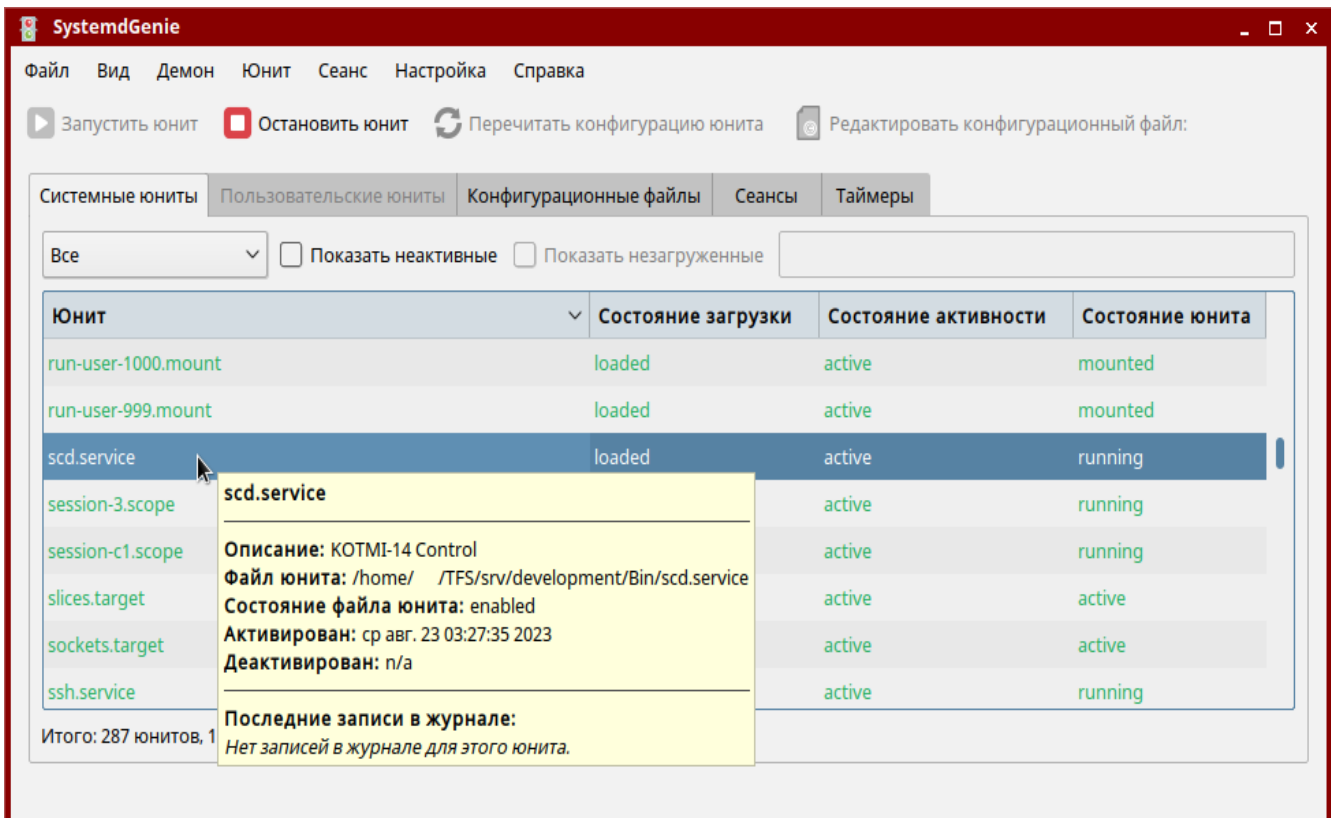


Рис 3. Установленный сервис в ОС Astra Linux.

Кроме того, проверить запущенный сервис можно по имени приложения, выполнив команду:

```
$ ps -aux | grep ScdService_x64
```

Результат вывода:

```
$ user 3721 0.0 0.3 185056 15640 ? Ssl 20:05 0:00  
/home/user/SPlatform/Service/Bin/ScdService_x64
```

После запуска сервиса, откройте программу «Системный монитор» и добавьте IP адрес вашей машины в адреса прямой рассылки, находящихся в окне настройки Монитора (см. документацию). Перезапустите Монитор и убедитесь, что он подключился к Сервису и получил список установленных конфигураций (Рис 4).

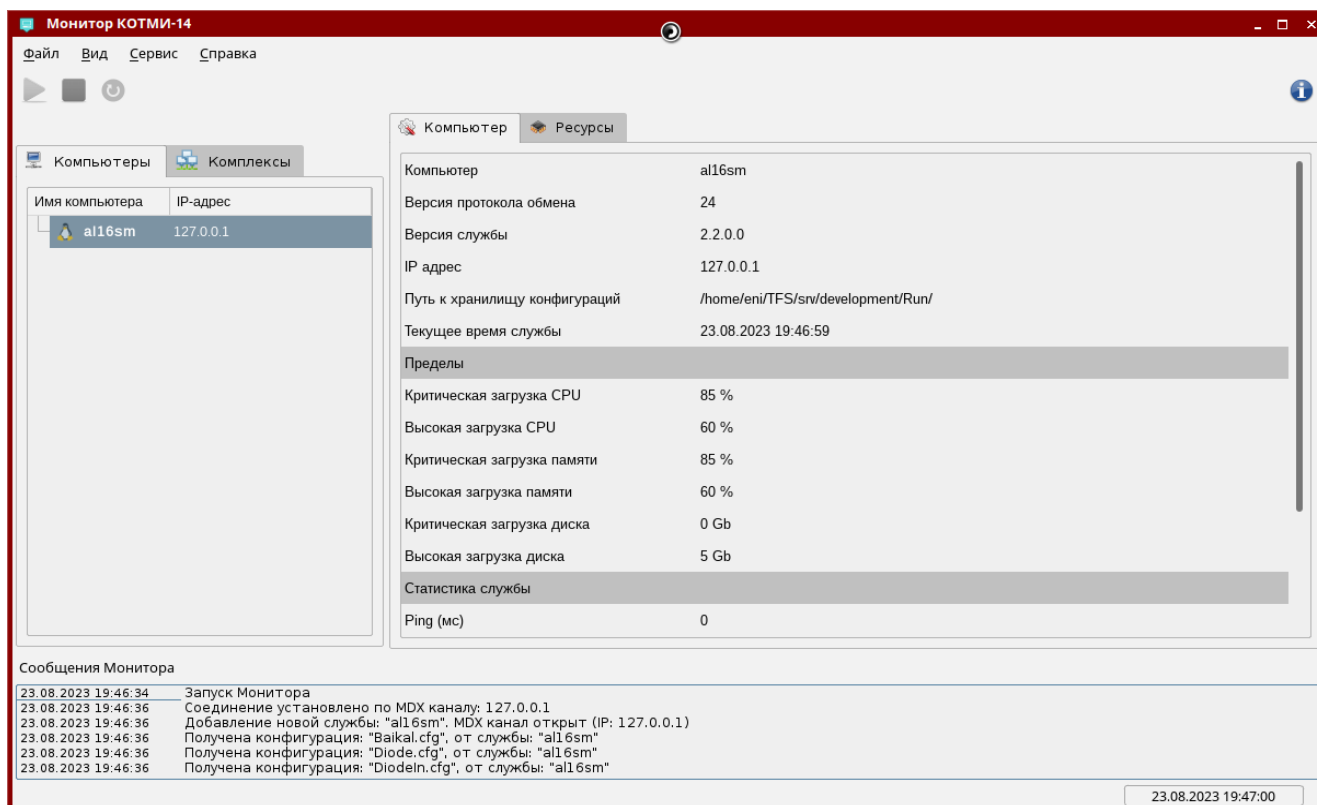


Рис 4. Монитор с запущенным сервисом (имя сервиса: al16sm).

4.1.6. Создание ярлыков для исполняемых модулей.

Чтобы создать ярлыки для исполняемых модулей на рабочем столе пользователя в ОС Linux, нажмите правую кнопку мыши на любом свободном месте рабочего стола.

- Для ОС Astra Linux:

В открывшемся контекстном меню выберите пункт:

«Создать» → «Ярлык». В окне редактора ярлыков заполните необходимые поля: «Имя», «Значок» и «Команда», как показано на рисунке №5.

- Для ОС Alt Linux.

В открывшемся контекстном меню выберите пункт:

«Создать кнопку запуска». В окне создания кнопки выберите тип «Приложение», заполните необходимые поля: «Имя» и «Команда», как показано на рисунке №6.

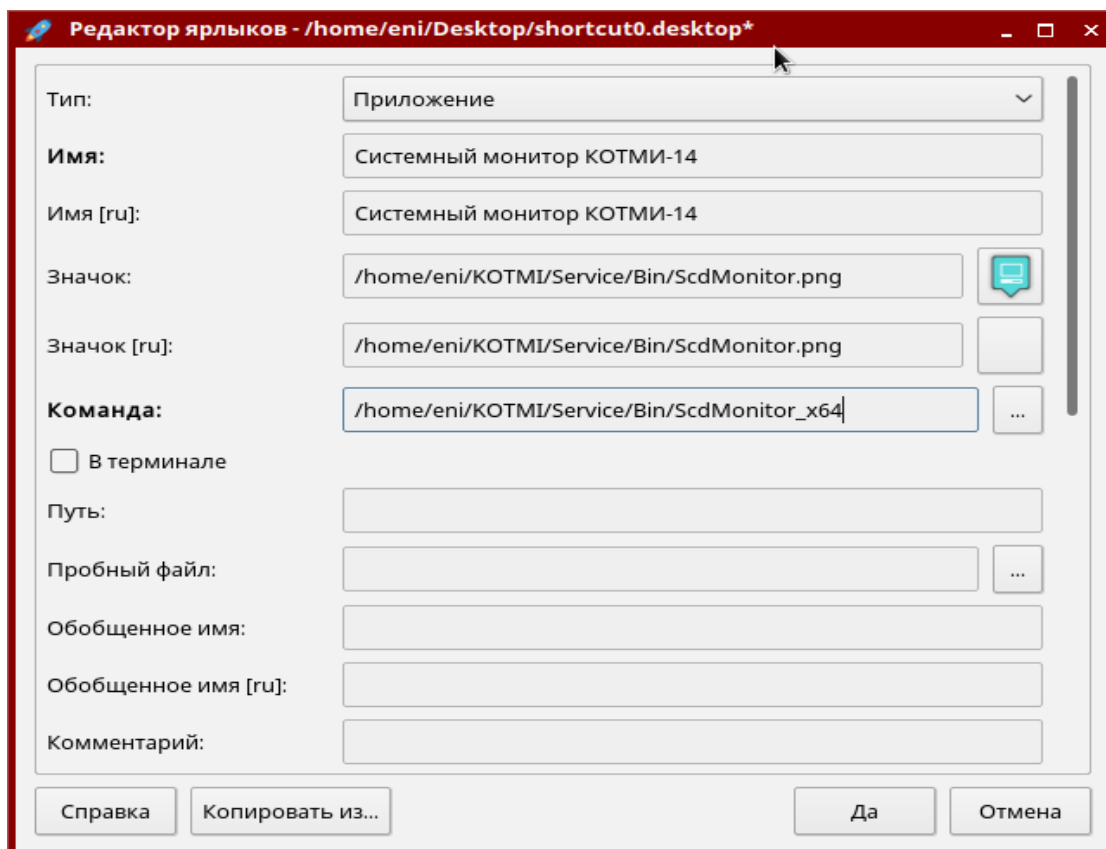


Рис 5. Создание ярлыка в ОС Astra Linux.

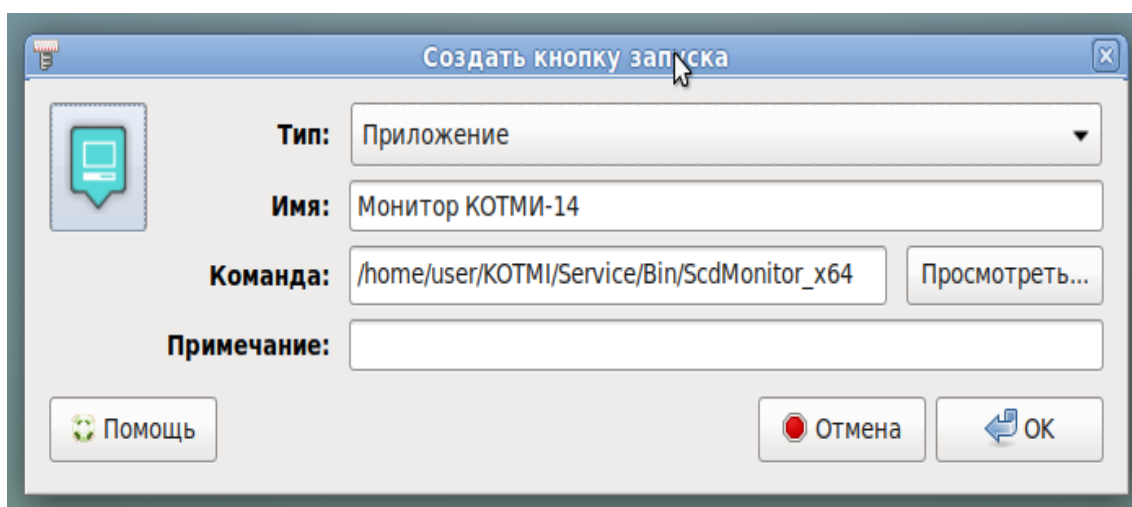


Рис 6. Создание ярлыка в ОС Alt Linux.

В итоге на рабочем столе ОС появятся новые ярлыки, как показано на рисунке №7.

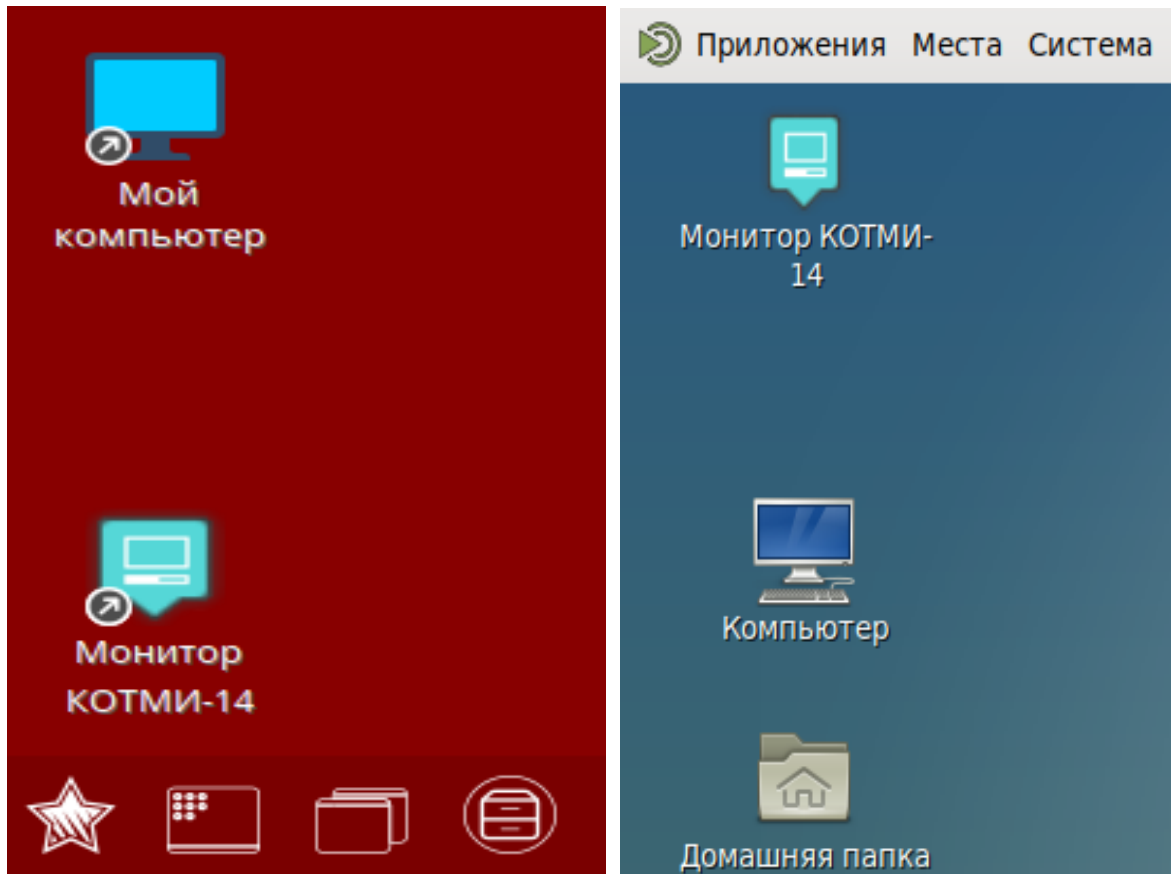


Рис 7. Ярлык в ОС Astra Linux (слева) и ОС Alt Linux (справа).

4.2. Ручная установка сервера Приложений в ОС Linux

4.2.1. Создание каталогов и копирование файлов.

Скопируйте необходимые для работы сервера Приложений файлы в заранее выбранный каталог. Выбор каталога и его имя зависит от конкретной задачи и не ограничен какими-либо условиями\правилами. Однако, в соответствии с принятыми соглашениями и исходя из опыта эксплуатации комплекса, можно выделить два основных подхода к размещению каталогов и файлов сервиса:

□ Размещение папок и файлов в домашнем каталоге пользователя (/home).

Для этого создайте нового пользователя в ОС Linux (здесь и ниже в документации будем считать, что пользователь в системе создан и имеет имя «user») или используйте существующего и добавьте его в группу пользователей (например, «scada») с заранее заданными правами. В домашней папке пользователя создайте каталог, если он еще не создан, с именем «SPlatform» (или любой другой, на ваш выбор), в нём подкаталоги: «Server», «Config» и «lib»:

```
$ mkdir /home/user/SPlatform && mkdir /home/user/SPlatform/Server && mkdir /home/user/SPlatform/lib && mkdir /home/user/SPlatform/Config
```

Скопируйте в «lib» файлы инструментальной библиотеки MDX и библиотеки лицензий: «libScdMdx_x64.so.X.0.0», «libScdMdx_x64.so.X.0», «libScdMdx_x64.so.X», «libScdMdx_x64.so», «libLicense_x64.so.X.0.0», «libLicense_x64.so.X.0», «libLicense_x64.so.X» и «libLicense_x64.so» или создайте самостоятельно символические ссылки на библиотеку «libScdMdx_x64.so.X.0.0» и «libLicense_x64.so.X.0.0». Здесь «X» - номер текущей версии библиотеки.

Создайте подкаталог «Bin» в папке «Server» и скопируйте в него исполняемый файл сервера и библиотеки работы с СУБД: «ScdServer_x64», «libScdDB_x64.so.X.0.0», «libScdDB_x64.so.X.0», «libScdDB_x64.so.X», «libScdDB_x64.so» и «ScdServer.png».

Создайте подкаталог «Bin» в папке «Config» и скопируйте в него исполняемый файл конфигуратора серверных комплексов КОТМИ-14: «ScdConfig_x64» и «ScdConfig.png».

Установите, если это требуется, права на запуск исполняемых файлов (например, такой командой: **sudo chmod +x имя_файла**).

□ Размещение папок и файлов в каталоге дополнительного программного обеспечения (/opt).

Для этого с правами суперпользователя создайте в папке «/opt» каталог, например, с именем «SPlatform» (или любой другой, на ваш выбор), в нём подкаталоги: «Server», «Config» и «lib». Скопируйте в «lib» файлы инструментальной библиотеки MDX и библиотеки лицензий: «libScdMdx_x64.so.X.0.0», «libScdMdx_x64.so.X.0», «libScdMdx_x64.so.X», «libScdMdx_x64.so», «libLicense_x64.so.X.0.0», «libLicense_x64.so.X.0», «libLicense_x64.so.X» и «libLicense_x64.so» или создайте самостоятельно символические ссылки на библиотеку «libScdMdx_x64.so.X.0.0» и «libLicense_x64.so.X.0.0». Здесь «X» - номер текущей версии библиотеки. Создайте

подкаталог «Bin» в папке «Server» и скопируйте в него исполняемый файл сервера и библиотеки работы с СУБД: «ScdServer_x64», «libScdDB_x64.so.X.0.0», «libScdDB_x64.so.X.0», «libScdDB_x64.so.X», «libScdDB_x64.so» и «ScdServer.png».

Создайте подкаталог «Bin» в папке «Config» и скопируйте в него исполняемый файл конфигуратора серверных комплексов КОТМИ-14: «ScdConfig_x64» и «ScdConfig.png». Выдайте пользователям необходимые права на доступ к созданным каталогам или добавьте их в соответствующие группы. Установите, если это требуется, права на запуск исполняемых файлов (например, такой командой: `sudo chmod +x имя_файла`).

ПРИМЕЧАНИЕ: Не рекомендуется использовать в именах папок кириллицу, а также специальные символы даже если их использование допускается самой ОС.

4.2.2. Регистрация разделяемых библиотек.

Зарегистрируйте нужные библиотеки, если это не было сделано ранее, как описано выше.

4.2.3. Установка необходимых пакетов.

Для работы сервера Приложений, библиотеки связи с БД и конфигуратора серверных комплексов КОТМИ-14 требуется наличие установленных в ОС библиотек:

- *libuuid.so.1*
- *libqsqlpsql.so*
- *libQt5Sql.so.5*
- *libQt5Gui.so.5*
- *libQt5Core.so.5*
- *libQt5Widgets.so.5*
- *libQt5Network.so.5*

Их можно установить из следующих пакетов:

Для ОС Astra Linux:

- *libuuid1*
- *libqt5sql5*
- *libqt5sql5-psql*
- *libqt5core5a*
- *libqt5widgets5*
- *libqt5gui5*
- *libqt5network5*

Для Alt Linux:

- *libuuid;*
- *qt5-base-common*
- *libqt5-sql*
- *qt5-sql-postgresql*
- *libqt5-core*
- *libqt5-widgets*
- *libqt5-gui*
- *libqt5-network.*

Установить пакеты и все необходимые для них зависимости можно через консольную команду «apt-get install имя_пакета», например, так:

Для ОС Astra Linux:

```
$ sudo apt-get install libuuid1 libqt5sql5 libqt5sql5-psql libqt5core5a libqt5widgets5 libqt5gui5 libqt5network5.
```

Для ОС Alt Linux:

```
$ sudo apt-get install libuuid libqt5-sql qt5-sql-postgresql qt5-base-common libqt5-core libqt5-widjets libqt5-gui libqt5-network
```

4.2.4. Развертывание Баз Данных.

Для работы сервера Приложений, библиотеки связи с БД и конфигуратора серверных комплексов КОТМИ-14 требуется наличие установленных в ОС библиотек:

Сервер Приложений КОТМИ-14 в своей работе использует несколько SQL БД: для доступа к Нормативно Справочной Информации - БД НСИ, для работы с событиями комплекса - БД Событий и для работы с SQL-архивами - БД Архивов. В случае использования ОС Linux, в качестве СУБД рассматривается использование PostgreSQL (не ниже версии 9.6) и Postgres PRO (не ниже 10 версии). В обоих случаях для их развертывания необходимо конвертировать уже существующие БД ПК КОТМИ-14 (см. описание конвертора *ScdSqlTools_x64*), с дальнейшим снятием бэкапа, или использовать ранее созданные бэкапы (в комплект поставки комплекса входит 2 бэкапа БД НСИ: Empty – пустая БД и SrvDB – демо версия комплекса «МеркурЭнерго»).

Создать новый бэкап можно при помощи программы pgAdmin (программного обеспечения с открытым исходным кодом для разработки и администрирования баз данных PostgreSQL, Postgres PRO и т.д.) или через выполнение консольной команды:

```
$ pg_dump --host 127.0.0.1 --port 5432 --username "postgres" --role "postgres" --no-password --format custom --blobs --no-privileges --no-tablespaces --disable-macs --no-security-labels --no-unlogged-table-data --file "/home/user/SPlatform/DB/SrvDB.backup" "SrvDB"
```

Указав в качестве параметров:

- Хост СУБД (127.0.0.1);
- Порт СУБД (5432);
- Имя пользователя СУБД (postgres);
- Путь и имя файла бэкапа (/home/user/SPlatform/DB/SrvDB.backup);
- Имя БД для которой создаётся бэкап (SrvDB).

Чтобы развернуть из имеющихся бэкапов базы данных, на заранее установленный экземпляр СУБД, необходимо:

- Для доступа к СУБД необходимо заранее выдать права тем пользователям, из-под которых происходит развертывание БД и запуск сервера Приложений. Прописать необходимые разрешения в файле «*pg_hba.conf*» и «*postgresql.conf*», если это требуется;

□ Создать 3 пустые БД (имена можно изменить): НСИ – «*SrvDB*», Событий – «*Events*» и Архивов – «*Archives*»;

□ Восстановить из бэкапа БД НСИ, например, такой командой:

```
#!/usr/lib/postgresql/9.6/bin/pg_restore --dbname "SrvDB" --no-password --no-security-labels --verbose "/home/user/SPlatform/DB/SrvDB.backup"»;
```

□ Указать строки подключения через «*Конфигуратор серверных комплексов*»: «*Database=SrvDB;Usr=postgres;Pwd=123;Host=127.0.0.1;Port=5432*». Где:

- «*Database*» - имя БД для подключения;
- «*Usr*» - имя пользователя;
- «*Pwd*» - опциональный пароль для пользователя;
- «*Host*» - адрес хоста с СУБД Postgres;
- «*Port*» - порт СУБД Postgres (Рис №8).

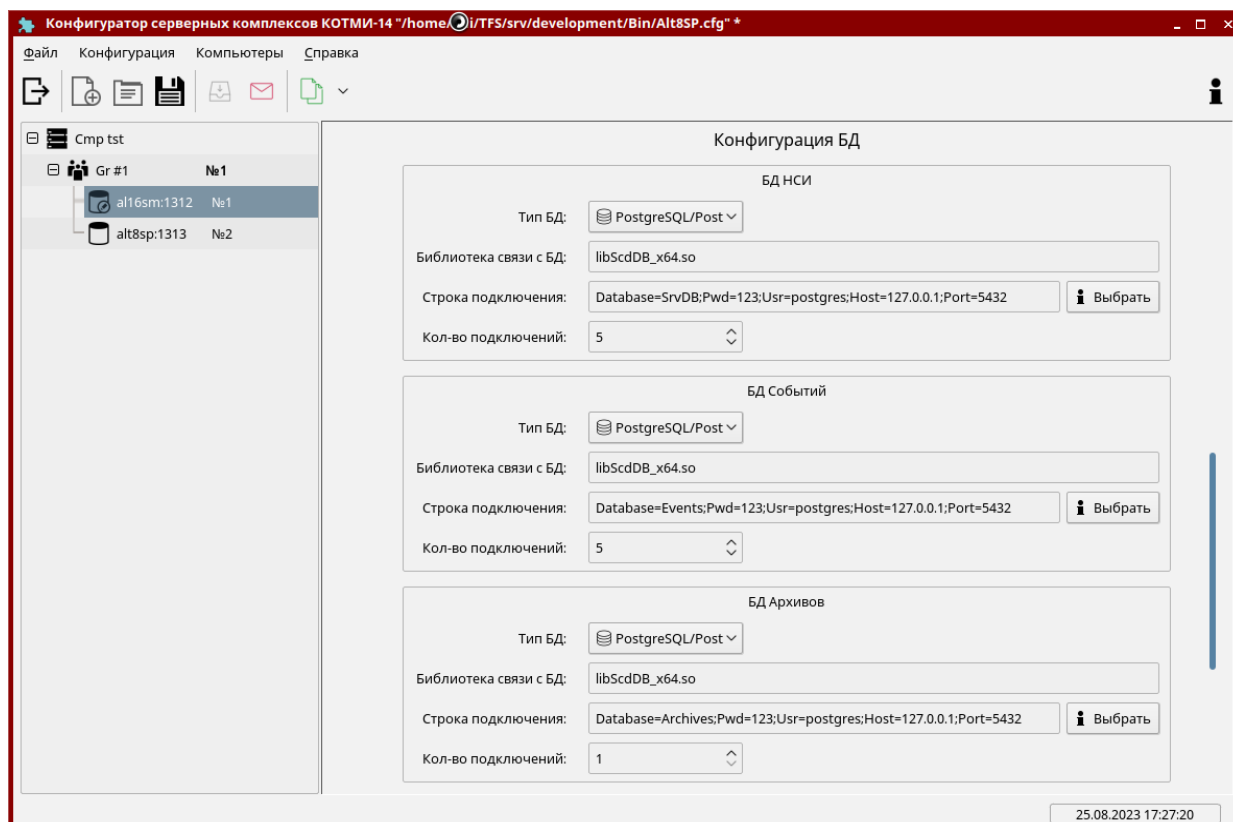


Рис 8. Внешний вид конфигуратора с заполненными строками подключения.

ПРИМЕЧАНИЕ: Восстанавливать из бэкапов БД Событий и БД Архивов отдельно не требуется, т.к. вся их структура, включая таблицы и поля, будет создана автоматически в момент первого старта сервера Приложений.

ПРИМЕЧАНИЕ: Старайтесь не использовать в именах БД кириллицу, спец символы или знак пробела. Это может привести к проблемам в процессе эксплуатации комплекса.

4.2.5. Запуск сервера Приложений.

Перед тем, как запустить сервер Приложений нужно создать его конфигурацию (см. документацию), она хранится в специальном файле с расширением «.cfg». Запуск сервера возможен в двух вариантах:

- Ручной запуск через консоль или заранее созданный ярлык;
- Запуск в автоматическом или полуавтоматическом режиме, через сервис Linux КОТМИ-14.

В случае запуска сервера с консоли, его работа контролируется тем пользователем, который его запустил. Если сервер Приложений запускается из-под сервиса его работа и состояние контролируется самим сервисом, а управление возможно через «Монитор КОТМИ-14» или консоль Telnet- клиента.

Чтобы запустить сервер с консоли, необходимо в качестве его аргументов запуска указать номер сервера из конфигурации и полный путь с расширением, к конфигурационному файлу. Например, для сервера с номером 1 из конфигурации с именем «*Config.cfg*», строка запуска будет выглядеть следующим образом:

```
$ /home/user/SPlatform/Server/Bin/ScdServer_x64 1 /home/user/SPlatform/Config.cfg
```

Для автоматического запуска сервера через сервис Linux необходимо, чтобы его файл конфигурации был прописан в файле хранилища конфигураций, а сам сервер имел установленный признак «*Автоматически*», в поле «*Тип запуска компьютером*» (см. документацию).

4.2.6. Создание ярлыков для исполняемых модулей.

См. пункт: *4.1.6 Создание ярлыков для исполняемых модулей.*

4.3. Ручная установка сервера ввода-вывода в ОС Linux

Скопируйте необходимые для работы сервера ввода-вывода файлы в заранее выбранный каталог. Выбор каталога и его имя зависит от конкретной задачи и не ограничен какими-либо условиями\правилами.

В домашней папке пользователя создайте каталог, если он еще не создан, с именем «SPlatform» (или любой другой, на ваш выбор), в нём подкаталог «ServerIO», в котором, в свою очередь, подкаталоги «Protocols2» и «Bin»:

```
$ mkdir /home/user/SPlatform && mkdir /home/user/SPlatform/ServerIO && mkdir /home/user/SPlatform/ServerIO/Protocols2 && mkdir /home/user/SPlatform/ServerIO/Bin
```

В каталог \...\ServerIO\Protocols2 скопировать файлы шаблона протокола (все шаблоны из соответствующего каталога Protocols2 предоставленного дистрибутива):

- 101.xml
- 103.xml
- 104.xml
- 61850cli.xml
- 61850srv.xml
- Arkmed.xml
- bridge.xml
- C37.118.xml
- can.xml
- cet.xml
- Eni101.xml
- Eni103.xml
- Eni104.xml
- EniBacnet.xml
- EniMB.xml
- EniMqtt.xml
- ICCPcli.xml
- ICCPsrv.xml
- Iec62541.xml
- mde.xml
- mdx.xml
- merc230.xml
- modbus.xml
- opc.xml
- sample.xml
- snmp.xml
- spabus.xml
- Eni101.xml»;
- Eni103.xml»;
- Eni104.xml»;
- EniMB.xml;
- EniMqtt.xml;
- EniBacnet.xml.

В каталог \...\ServerIO\Bin скопировать файлы библиотеки протоколов (файлы *.so.1.0.0). Создать для этих файлов символические ссылки.

Например:

- Для файла «libRdxEni101.so.1.0.0» - символическая ссылка «libRdxEni101.so»;
- Для файла «libRdxEni103.so.1.0.0» - символическая ссылка «libRdxEni103.so»;
- Для файла «libRdxEni104.so.1.0.0» - символическая ссылка «libRdxEni104.so»;
- Для файла «libRdxEniMB.so.1.0.0» - символическая ссылка «libRdxEniMB.so»;
- Для файла «libRdxEniMqtt.so.1.0.0» - символическая ссылка «libRdxEniMqtt.so»;
- Для файла «libRdxEniBacnet.so.1.0.0» - символическая ссылка «libRdxEniBacnet.so»;

Перечень файлов протоколов (копируются все файлы протоколов из соответствующего каталога Bin предоставленного дистрибутива)

- libAsnAcse_x64.so
- libAsnAcse_x64.so.1
- libAsnAcse_x64.so.1.0
- libAsnAcse_x64.so.1.0.0
- libAsnIso8823_x64.so
- libAsnIso8823_x64.so.1
- libAsnIso8823_x64.so.1.0
- libAsnIso8823_x64.so.1.0.0
- libAsnMms_x64.so
- libAsnMms_x64.so.1
- libAsnMms_x64.so.1.0
- libAsnMms_x64.so.1.0.0
- libCfgICCPcli.so
- libCfgICCPcli.so.2
- libCfgICCPcli.so.2.0
- libCfgICCPcli.so.2.0.0
- libCfgICCPsrv.so
- libCfgICCPsrv.so.2
- libCfgICCPsrv.so.2.0
- libCfgICCPsrv.so.2.0.0
- libCfgIec61850cli.so
- libCfgIec61850cli.so.2
- libCfgIec61850cli.so.2.0
- libCfgIec61850cli.so.2.0.0
- libCfgIec62541.so
- libCfgIec62541.so.2
- libCfgIec62541.so.2.0
- libCfgIec62541.so.2.0.0
- libCfgMdx.so
- libCfgMdx.so.2
- libCfgMdx.so.2.0
- libCfgMdx.so.2.0.0
- libLicense_x64.so
- libLicense_x64.so.1
- libLicense_x64.so.1.0
- libLicense_x64.so.1.0.0

- libRdxBridge.so
- libRdxBridge.so.2
- libRdxBridge.so.2.0
- libRdxBridge.so.2.0.0
- libRdxC37.118.so
- libRdxC37.118.so.2
- libRdxC37.118.so.2.0
- libRdxC37.118.so.2.0.0
- libRdxEni101.so.1.0.0
- libRdxEni103.so.1.0.0
- libRdxEni104.so.1.0.0
- libRdxEniBacnet.so.1.0.0
- libRdxEniMB.so.1.0.0
- libRdxEniMqtt.so.1.0.0
- libRdxICCPcli.so
- libRdxICCPcli.so.2
- libRdxICCPcli.so.2.0
- libRdxICCPcli.so.2.0.0
- libRdxICCPsrv.so
- libRdxICCPsrv.so.2
- libRdxICCPsrv.so.2.0
- libRdxICCPsrv.so.2.0.0
- libRdxIec101.so
- libRdxIec101.so.2
- libRdxIec101.so.2.0
- libRdxIec101.so.2.0.0
- libRdxIec103.so
- libRdxIec103.so.2
- libRdxIec103.so.2.0
- libRdxIec103.so.2.0.0
- libRdxIec104.so
- libRdxIec104.so.2
- libRdxIec104.so.2.0
- libRdxIec104.so.2.0.0
- libRdxIec61850cli.so
- libRdxIec61850cli.so.2
- libRdxIec61850cli.so.2.0
- libRdxIec61850cli.so.2.0.0
- libRdxIec62541.so
- libRdxIec62541.so.2
- libRdxIec62541.so.2.0
- libRdxIec62541.so.2.0.0
- libRdxMdx.so
- libRdxMdx.so.2
- libRdxMdx.so.2.0
- libRdxMdx.so.2.0.0
- libRdxModbus.so
- libRdxModbus.so.2

- libRdxModbus.so.2.0
- libRdxModbus.so.2.0.0
- libRdxMonitor.so
- libRdxMonitor.so.2
- libRdxMonitor.so.2.0
- libRdxMonitor.so.2.0.0
- libRdxSnmp.so
- libRdxSnmp.so.2
- libRdxSnmp.so.2.0
- libRdxSnmp.so.2.0.0
- libScdMdx_x64.so
- libScdMdx_x64.so.1
- libScdMdx_x64.so.1.0
- libScdMdx_x64.so.1.0.0
- QtMonitor
- RdxConfig
- RdxServer

В каталог \...\ServerIO скопировать конфигурационные файлы.

- для протокола МЭК 60870-101 файл «RdxEni101.ini»;
- для протокола МЭК 60870-103 файл «RdxEni103.ini»;
- для протокола МЭК 60870-104 файл «RdxEni104.ini»;
- для протокола Modbus RTU файл «RdxEniMB.ini»;
- для протокола MQTT файл «RdxEniMqtt.ini»;
- для сервера ввода-вывода файл «RdxServer.ini»

4.4. Ручная установка модуля «С-Платформа.ССНТИ.КЛИЕНТ».

Скопируйте необходимые для работы клиента ССНТИ файлы в заранее выбранный каталог. Выбор каталога и его имя зависит от конкретной задачи и не ограничен какими-либо условиями\правилами.

В домашней папке пользователя создайте каталог, если он еще не создан, с именем «SPlatform» (или любой другой, на ваш выбор), в нём подкаталог «SSNTI»:

```
$ mkdir /home/user/SPlatform && mkdir /home/user/SPlatform/ServerIO && mkdir /home/user/SPlatform/SSNTI
```

Скопируйте в него из дистрибутива файлы:

- libkdsoap.so.2;
- settings.in;
- SSNTI_Cli_Settings_la.qm;
- SSNTI_Cli_Settings;
- SSNTI_Client;
- SSNTI_Client_la.qm.

Файлам SSNTI_Cli_Settings, SSNTI_Client назначьте атрибут «Выполнение» (например, командой: **sudo chmod +x имя_файла**);

Создайте ярлык для быстрого доступа оператора к файлу SSNTI_Client на рабочем столе для запуска «С-Платформа.ССНТИ.КЛИЕНТ» , либо пропишите путь к этому файлу в конфигураторе серверных комплексов;

Создайте каталог /.../SSNTI/Lib;

Создайте каталог /.../SSNTI/plugins;

Создайте каталог /.../SSNTI/plugins/platforminputcontexts;

Создайте каталог /.../SSNTI/platforms;

Создайте каталог /.../SSNTI/sqldrivers;

Скопируйте файлы из каталогов дистрибутива в соответствующие каталоги на диске.

4.5. Ручная установка модуля «АМГО ОПРЧ».

Скопируйте необходимые для работы модуля ОПРЧ файлы в заранее выбранный каталог. Выбор каталога и его имя зависит от конкретной задачи и не ограничен какими-либо условиями\правилами.

В домашней папке пользователя создайте каталог, если он еще не создан, с именем «SPlatform» (или любой другой, на ваш выбор), в нём подкаталог «Oprch»:

```
$ mkdir /home/user/SPlatform && mkdir /home/user/SPlatform/ServerIO && mkdir /home/user/SPlatform/Oprch
```

Скопируйте в него из дистрибутива файлы:

- | | |
|-----------------|--|
| - oprch_x64 | - исполняемый модуль клиента ОПРЧ |
| - oprch.ini | - конфигурационный файл клиента ОПРЧ |
| - oprch.lic | - файл лицензии клиента ОПРЧ |
| - Alarm.wav | - файл звука аварийного события |
| - OprchCnsl_x64 | - исполняемый модуль расчетной части ОПРЧ |
| - OprchCnsl.ini | - конфигурационный файл расчетной части ОПРЧ |

Файлам oprch_x64 и OprchCnsl_x64 назначьте атрибут «Выполнение»

Для удобства работы для модулей oprch_x64 и OprchCnsl_x64 создайте ярлыки запуска

4.6. Автоматизированная установка ПК «С-Платформа» .

Создайте нового пользователя в ОС Linux (здесь и ниже в документации будем считать, что пользователь в системе создан и имеет имя «user») или используйте существующего и добавьте его в группу пользователей (например, «scada») с заранее заданными правами. В домашней папке пользователя создайте каталог с именем «SPlatform» (или любой другой, на ваш выбор):

```
$ mkdir /home/user/SPlatform
```

Скопируйте необходимые для работы установочные пакеты в каталог Splatform (перечисленные ниже пакеты собраны для работы под 64-рядной версией ОС Astra Linux 1.7 для процессоров архитектуры x86):

- s-platform-kotmi-14.deb - сервер приложений
- s-platform-RdxServer.deb - сервер ввода-вывода
- s-platform-Workstation.deb - клиент (АРМ)
- s-platform-oprch.deb - подсистема АМГО ОПРЧ
- s-platform-protocols.deb - протоколы информационного обмена
- s-platform-ssnti.deb - клиент ССНТИ

Поочередно для каждого пакета выполните команду:

```
$ sudo dpkg -i (имя файла)
```

Для работы сервера приложений и клиента необходимо наличие СУБД.

В дистрибутиве предоставляется шаблон базы данных для СУБД PostgreSQL (п.4.7).

Для сервера ввода-вывода из рабочего каталога выполнить команду kotmi_setup.sh.

В рабочий каталог будут скопированы (из установочного пакета) каталоги configurations, protocols2, файл RdxServer.ini, ярлыки запуска для RdxServer, QtMonitor, RdxConfig будут добавлены на рабочий стол.

В рабочем каталоге будет создан файл rdxrun, для запуска приложений с консоли (надо ввести команды: ./rdxrun RdxServer, ./rdxrun QtMonitor, ./rdxrun RdxConfig).

Более подробно операции по установке сервера ввода-вывода описаны в документе КОТМИ-RDX.Руководство системного программиста/

4.7. Настройка PostgreSQL.

На машине с ОС Astra Linux установите СУБД PostgreSQL версии не ниже 9.6.

Для удобства конфигурирования необходимо установить пакет pgadmin.

В СУБД PostgreSQL всегда присутствует корневая учётная запись postgres, наделённая полными административными правами. В ОС Astra Linux для данной учётной записи по умолчанию не задан пароль, подключение к СУБД возможно только из терминала соответствующей операционной системы.

Необходимо задать пароль для административной учётной записи postgres.

- выполните команду psql от имени пользователя postgres с помощью sudo:

```
$ sudo -u postgres psql
```

- Средствами командной строки psql выполните:

```
> \password
```

- Введите пароль, нажмите Enter.
- В следующей строке повторите ввод пароля. Пароль задан.
- Выйдите из командной строки psql:

```
> \q
```

Запустите pgadmin. Создайте базы данных:

- «SrvDb» - база данных НСИ.
- «Events» - база данных Событий.
- «Archives» - база данных Архивов.

Для базы данных «SrvDb» выполните восстановление из файла **Empty.backup** (входит в поставляемый дистрибутив как отдельный файл). В результате в базе данных будут созданы все необходимые для работы комплекса таблицы (и поля в них).

Для баз данных «Events» и «Archives» первичная инициализация не требуется. Все необходимые таблицы в базы данных создаются и заполняются сервером приложений при первом обращении к базам.

После завершения инсталляции программных модулей и базы данных непосредственная настройка комплекса «С-Платформа» производится через программы-конфигураторы, описание которых находится в руководствах на соответствующие подсистемы.