

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное образовательное учреждение высшего
образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Политехнический институт
Факультет приборостроения,
информационных технологий и
электроники

Кафедра
«Информационная безопасность
систем и технологий»

Специальность 10.05.02 –

Специализация -

Информационная безопасность
телекоммуникационных систем
Разработка защищенных
телекоммуникационных систем

ДИПЛОМНЫЙ ПРОЕКТ

на тему

ПРОГРАММНЫЙ КОМПЛЕКС МОНИТОРИНГА ПРОЦЕССОВ
ФУНКЦИОНИРОВАНИЯ РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

Студент	_____	<u>Кузнецов Алексей Валерьевич</u>
Руководитель	_____	<u>Козлов А.Ю.</u>
Нормоконтролёр	_____	<u>Иванов А.П.</u>
Рецензент	_____	<u>Будникова М.В.</u>

Работа допущена к защите (протокол заседания кафедры от _____ № _____)

Заведующий кафедрой _____ С.Л. Зефилов

Работа защищена с отметкой _____ (протокол заседания ГЭК
от _____ № _____)

Секретарь ГЭК _____ А.Г. Фатеев

Пенза 2017

T3

T3

T3

РЕФЕРАТ

Отчет содержит 98 с., 18 рис., 18 табл., 12 источников, 11 прил.

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ, МОНИТОРИНГ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ, СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

Объектом исследований в данном дипломном проекте является разработка комплекса мониторинга процессов функционирования распределенной вычислительной системы.

Цель дипломного проекта — разработка и реализация программного комплекса мониторинга процессов функционирования распределенной вычислительной системы.

В процессе дипломного проектирования был проведен анализ предметной области и операционной системы, для которой разрабатывается комплекс, был разработан и реализован программный комплекс мониторинга процессов функционирования распределенной вычислительной системы, была написана документация для разработанного комплекса.

В результате дипломного проектирования был разработан программный комплекс мониторинга процессов функционирования распределенной вычислительной системы.

СОДЕРЖАНИЕ

Введение	10
1 Анализ предметной области	11
1.1 Анализ операционной системы	11
1.2 Выбор инструментальных средств	12
1.3 Принципы построения систем мониторинга	14
2 Модель угроз	18
2.1 Описание информационной системы и особенностей ее функционирования	18
2.2 Модель нарушителя	21
2.3 Актуальные угрозы безопасности информации	26
3 Формирование требований к комплексу и проектирование	31
4 Программная реализация комплекса мониторинга	33
4.1 Разработка базы данных комплекса мониторинга	34
4.2 Модули комплекса мониторинга	38
4.3 Режим работы «Инициализация»	40
4.4 Режим работы «Администрирование»	45
4.5 Режим работы «Работа»	55
5 Тестирование комплекса мониторинга	57
6 Разработка программной документации	61
Заключение	66
Список использованных источников	67
Приложение А. Текст файла mainwindow.h	68
Приложение Б. Текст файла smcore.h	71
Приложение В. Текст файла workwithdatabase.h	72
Приложение Г. Текст файла abstractdaemon.h	74
Приложение Д. Текст файла serversocketfordaemon.h	77
Приложение Е. Текст файла clientsocketfordaemon.h	78

Приложение Ж. Текст файла workwithproc.h	79
Приложение З. Текст файла cryptmodule.h	80
Приложение И. Текст файла connectandworkwithdatabase.h	81
Приложение К. Протокол проверки ВКР на оригинальность	82
Приложение Л. Руководство оператора	84

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В настоящем отчете используются следующие термины с соответствующими определениями.

Система мониторинга – комплекс программно-технических средств, предназначенных для автоматизации процесса сбора и анализа событий с целью выявления инцидентов.

Автоматизированное рабочее место (АРМ) – совокупность электронно-вычислительной и других технических средств АСУ на пункте управления (боевом информационном посту), предназначенном для обеспечения эффективной деятельности должностных лиц (операторов).

Автоматизированная система – система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций.

Источник угрозы безопасности информации – субъект доступа, материальный объект или физическое явление, являющиеся причиной возникновения угрозы безопасности информации.

Контролируемая зона – пространство (территория, здание, часть здания, помещение), в котором исключено неконтролируемое пребывание посторонних лиц, а также транспортных, технических и иных материальных средств.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

КЗ – контролируемая зона;

ИБ – информационная безопасность;

КШМ – командно-штабная машина;

ИС – информационная система;

АРМ – автоматизированное рабочее место;

АС – автоматизированная система;

АСУ – автоматизированная система управления;

ОС – операционная система.

ВВЕДЕНИЕ

Цель дипломного проекта — разработка и реализация программного комплекса мониторинга запускаемых процессов.

В результате дипломного проектирования были проведены:

- анализ предметной области и операционной системы;
- разработка модели угроз;
- программная реализация комплекса мониторинга;
- тестирование комплекса мониторинга;
- разработка программной документации.

В результате дипломного проектирования был разработан программный комплекс мониторинга процессов функционирования распределенной вычислительной системы.

1 Анализ предметной области

1.1 Анализ операционной системы

Комплекс мониторинга предназначен для работы в составе АРМ под управлением операционной системы Astra Linux.

Astra Linux – это операционная система специального назначения. Данная система принадлежит к семейству Unix-подобных и основывается на Linux подобной системе Debian. Используется в Российских силовых ведомствах и государственных органах для организации защищенных автоматизированных систем.

Данная операционная система разрабатывается в двух версиях: Common Edition (общего назначения) и Special Edition (специального назначения).

Так как АРМ входит в состав защищенной автоматизированной системы аппаратной связи модуля сопряжения в качестве операционной системы используется Astra Linux Special Edition версии 1.2 для которой имеется сертификат соответствия № СФ/014-2578 удостоверяющий, что операционная система специального назначения соответствует требованиям ФСБ России к программному обеспечению, используемому в информационных и телекоммуникационных системах специального назначения, и требованиям по защите информации от несанкционированного доступа с использованием средств криптографической защиты информации в автоматизированных информационных системах, расположенных на территории Российской Федерации, 1 класса, и может использоваться для обработки информации, содержащей сведения, составляющие государственную тайну [1].

В данной операционной системы для обеспечения безопасности используется комплекс конструктивных решений и доработок.

Например, в операционной системе используются защищенные версии программного обеспечения для выполнения задач оператора, такие как защищенная версия базы данных PostgreSQL, браузер FireFox и многие другие.

Основными компонентами операционной системы являются:

- программный комплекс организации домена безопасности (единого пространства пользователей и ресурсов локальной вычислительной сети);
- комплексная защита информации разграничения доступа;
- средства разработки;
- терминальный сервер;
- контроль целостности операционной системы;
- аудит и журналирование событий;
- система разграничения доступа к внешним устройствам;
- средства резервного копирования и восстановления;
- FLY - графическая система и полнофункциональный рабочий стол.

1.2 Выбор инструментальных средств

В соответствии с техническим заданием программное обеспечение написано на языке C++ с использованием библиотеки Qt версии 4.8 и среды разработки Qt Creator 4.

Qt – кроссплатформенный инструментальный набор для разработки программного обеспечения на языке программирования C++. Qt позволяет запускать написанное с его помощью программное обеспечение в большинстве современных операционных систем путём простой компиляции программы для каждой операционной системы (ОС) без изменения исходного кода [2]. Также в Qt реализованы собственные варианты классов из стандартной библиотеки языка C++.

Библиотека использует собственный формат проекта, именуемый «.pro файлом», в котором собрана информация о том, какие файлы будут скомпилированы, по каким путям искать заголовочные файлы и много другой информации. Впоследствии при помощи утилиты qmake из них получаются makefile, для make-утилиты компилятора. Это очень удобно, так как позволяет гибко настраивать проекты в любой операционной системе.

Библиотека Qt включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения. Это классы для разработки элементов графического интерфейса, классы для работы с сетью, базами данных и XML, а также многие другие классы для различных областей.

Также стоит отметить, что плюсом Qt является наличие качественной и хорошо оформленной документации, оснащенной большим количеством примеров кода.

Для хранения эталонной и контрольной информации, а также журналов безопасности комплекса была выбрана свободная объектно-реляционная система управления базами данных PostgreSQL 9.

Данная СУБД обладает следующими возможностями:

- поддержка функций – блоков кода исполняемыми на сервере, а не на клиенте БД;
- поддержка триггеров – функции, иницируемые DML-операциями;
- механизм правил и представлений;
- имеется поддержка индексов следующих типов: В-дерево, хэш, R-дерево, GiST, GIN. При необходимости можно создавать новые типы индексов;
- многоверсионность - поддержка одновременной модификации БД несколькими пользователями, без блокировки чтения.

Также данная СУБД поддерживает множество типов:

- численные типы;
- двоичные типы;
- булев тип;
- типы «дата/время»;
- перечисления;
- массивы;
- символьные типы;
- другие специфичные типы данных.

При необходимости PostgreSQL может быть расширен пользовательскими объектами [3].

Программное обеспечение написано для ОС Astra Linux – операционная система специального назначения на базе ядра Linux, созданная для нужд российских силовых ведомств и спецслужб. Сертифицирована в системах сертификации средств защиты информации Минобороны, ФСТЭК и ФСБ России. Включена в Единый реестр российских программ Минкомсвязи России.

1.3 Системы мониторинга

Система мониторинга событий информационной безопасности предназначена для автоматизации процесса сбора и анализа информации о событиях безопасности, поступающих из различных источников. В качестве таких источников могут выступать средства защиты информации, общесистемное и прикладное ПО, телекоммуникационное обеспечение и другие средства. Система мониторинга событий информационной безопасности включает в себя следующие компоненты:

- программно-техническая часть – реализуется на основе продуктов по мониторингу событий безопасности;
- документационная часть – включает в себя набор документов, описывающих основные процессы, связанные с выявлением и реагированием на инциденты безопасности;
- персонал – подразумевает выделение сотрудников, ответственных за работу с системой мониторинга событий информационной безопасности.

Программно-техническая часть системы мониторинга событий информационной безопасности включает следующие компоненты:

- агенты мониторинга, предназначенные для сбора информации, поступающей от различных источников событий и передачи собранной информации на обработку;
- сервер событий, обеспечивающий централизованный анализ и обработку информации о событиях безопасности, которая поступает от агентов;

- хранилище данных, содержащее информацию о работе системы, данные, полученные от агентов, а также результаты обработки и анализа этих данных;
- подсистема управления системой мониторинга, позволяющая в реальном масштабе времени просматривать результаты работы системы, а также управлять ее настройками и параметрами.

Программно-техническая часть может быть реализована с использованием таких средств защиты как:

- система активного мониторинга рабочих станций;
- система обнаружения сетевых атак.

Системы обнаружения атак предназначены для выявления и противодействия сетевым атакам злоумышленников.

Системы активного мониторинга рабочих станций ИС предназначены для выявления и блокирования информационных атак на уровне рабочих станций ИС.

Системы активного мониторинга представляют собой специализированное программно-аппаратное обеспечение с типовой архитектурой и могут включать в себя следующие компоненты:

- модули-датчики для сбора необходимой информации;
- модуль выявления атак, выполняющий обработку данных, собранных датчиками, с целью обнаружения информационных атак;
- модуль реагирования на обнаруженные атаки;
- модуль хранения конфигурационной информации, а также информации об обнаруженных атаках. Таким модулем, как правило, выступает стандартная СУБД;
- модуль управления компонентами системы обнаружения атак.

Датчики системы активного мониторинга устанавливаются на рабочие станции пользователей ИС и позволяют собирать информацию обо всех событиях, происходящих на них.

Примером такой информации может служить:

- информация о приложениях, запускаемых на рабочих станциях;
- информация о пользователях, работающих на станции в текущий момент времени;
- информация о файловом доступе приложений.

Собранная информация поступает в модуль анализа данных системы активного мониторинга, где осуществляется её анализ и обработка.

Предварительно администратор безопасности должен выполнить настройку модуля анализа системы активного мониторинга. Он должен определить и установить требования, которые разрешают или запрещают пользователям ИС выполнение определённых операций на рабочих станциях.

Совокупность таких требований представляет собой политику безопасности системы мониторинга, которая может являться частью более общей политики безопасности организации. Так, например, в соответствии с заданной политикой безопасности некоторым пользователям может быть запрещена работа с принтерами или доступ к определённым файлам, программам.

Любое событие, зафиксированное датчиками системы активного мониторинга и нарушающее ранее заданную политику, считается информационной атакой или инцидентом.

В случае выявления нарушений политики безопасности системы активного мониторинга может реализовываться пассивные и активные методы реагирования.

К пассивным методам относится оповещение администратора безопасности или заданного настройками должностного лица, об обнаруженных несанкционированных действиях пользователей.

Активные методы подразумевают блокирование тех действий пользователей, которые нарушают заданную политику безопасности.

Системы активного мониторинга могут сочетать активные и пассивные методы реагирования.

Системы активного мониторинга должны быть также оснащены подсистемой собственной безопасности, позволяющей защитить компоненты системы активного мониторинга от несанкционированных воздействий нарушителей [4].

2 Модель угроз

2.1 Описание информационной системы и особенностей ее функционирования

В рамках данного дипломного проекта рассматривается ИС, представляющая собой АРМ оператора аппаратной связи наземного модуля сопряжения на базе КШМ.

Цель функционирования АРМ оператора обеспечение передачи и приема информации между различными узлами связи.

Задачи АРМ:

- установление связи между различными узлами связи системы связи;
- доставка информации подразделениям или должностным лицам являющимся получателем информации.

Для установления связи между различными узлами связи системы связи, АРМ выполняет следующие функции:

- запуск и обеспечение работы средств связи;
- прием и передача запросов на установление связи;
- запуск и поддержание функционирования программного обеспечения и процессов, обеспечивающих сеанс связи.

Для передачи информации подразделениям или должностным лицам являющимся получателем информации, АРМ выполняет следующие функции:

- передача принятой информации с узла связи в электронном виде или голосовым сообщением с помощью штатных средств узла связи;
- посылка принятой информации с узла связи курьером на бумажном носителе.

Таким образом, на АРМ оператора выделяются следующие информационные процессы:

- установка и обеспечение сеанса связи;
- прием и передача принятой информации получателям.

Сеанс связи устанавливается и обеспечивается с помощью специального программного обеспечения установленного на АРМ оператора. Оператор АРМ вводит в программное обеспечение информацию необходимую для установления связи и информацию об абонентах. После программного обеспечения в автоматическом режиме устанавливает и обеспечивает связь между абонентами. При организации связи оператор АРМ руководствуется своими должностными инструкциями, документом «Руководство оператора» для специального программного обеспечения установленного на АРМ, а также планом организации связи.

Схема информационного процесса «Установка и обеспечение сеанса связи» продемонстрирована на рисунке 1.



Рисунок 1 – Схема информационного процесса «Установка и обеспечение сеанса связи»

Прием и передача принятой информации получателям производится оператором АРМ. После получения информации оператор принимает решение о способе ее передачи получателю. Он может передать информацию с помощью

штатных средств узла связи либо послать с помощью курьера в бумажном виде. Распечатка документа обеспечивается с помощью принтера, входящего в состав АРМ. Информация поступает на АРМ в виде сигналов пригодных для передачи по среде распространения. В АРМ поступает различная информация для обеспечения управления войсками.

Схема информационного процесса «Прием и передача принятой информации получателям» продемонстрирована на рисунке 2.



Рисунок 2 – Схема инфомрационного процесса «Прием и передача принятой информации получателям»

Для обеспечения выполнения функций ИС в ее состав входят следующие технические средства:

- специальные аппаратные средства связи;
- специализированное средство вычислительной техники оборудованное клавиатурой, мышью, монитором, колонками и источником бесперебойного питания;
- стационарный телефон;
- принтер.

На средстве вычислительной техники установлено следующее программное обеспечение:

- ОС Astra Linux Special Edition версии 1.2, для которой имеется сертификат соответствия № СФ/014-2578;
- защищенная система управления базами данных PostgreSQL;
- пакет офисных программ OpenOffice, LibreOffice;
- Web-сервер Apache2 версии 2.2.9;
- браузер FireFox версии 3.6.12;
- сервер электронной почты Exim4 версии 4.70;
- почтовый клиент Thunderbird 3.1.6;
- редактор растровой графики GIMP версии 2.6.11;
- защищенный сервер печати CUPS 1.4;
- комплекс специализированного программного обеспечения для

аппаратной связи модуля сопряжения.

Данная ИС не имеет подключения к сети Internet и является закрытой системой.

Система подключена к локальной сети узла связи, для организации передачи информации между элементами узла.

2.2 Модель нарушителя

АРМ оператора входит в состав узла связи, для которого принимаются меры по организации и охраны контролируемой зоны.

Поэтому для АРМ оператора аппаратной связи наземного модуля сопряжения будут рассмотрены следующие виды нарушителей:

- внешние нарушители, не имеющие доступа в КЗ;
- внутренние нарушители.

К внешним нарушителям, не имеющим доступа в КЗ, относятся специальные службы иностранных государств (блоков государств) и террористические, экстремистские группировки.

К внутренним нарушителям можно отнести обслуживающий персонал аппаратуры связи, обслуживающий персонал командно-штабной машины, а также оператора автоматизированного рабочего места.

Данный тип нарушителя имеет доступ в контролируемую зону.

Для определения прав доступа, была составлена матрицы доступа к компонентам информационной системы.

Для краткости специальные службы иностранных государств (блоков государств) и террористические, экстремистские группировки были объединены в один блок внешних нарушителей, так как они не обладают правом доступа к компонентам информационной системы и относятся к одному типу нарушителей.

Матрица доступа к компонентам информационной системы продемонстрирована в таблице 1.

Таблица 1 – Матрица доступа к компонентам информационной системы

Нарушитель	Права доступа
внешние нарушители	не имеют доступа к элементам ИС
оператор АРМ	имеет доступ ко всем компонентам ИС
сотрудники, обслуживающие средства связи ИС	имеют доступ к активному и пассивному оборудованию каналов связи (специальные аппаратные средства связи)
сотрудники обслуживающие техническое средство (КШМ), в котором расположена ИС	не имеют доступа к элементам ИС

Для характеристики нарушителей были выявлены предполагаемые сведения об опыте, знаниях и доступных ресурсах.

Сведения об опыте, знаниях и доступных ресурсах продемонстрированы в таблице 2.

Таблицы 2 – Сведения о нарушителях

Нарушитель	Опыт	Знания	Ресурсы
специальные службы иностранных государств	высокий	аппаратных средств и ПО для разведки	спец. программы и технические средства
террористические и экстремистские группировки	высокий	аппаратных средств и ПО для разведки	спец. аппаратура
оператор АРМ	низкий	аппаратуры и ПО ИС	штатные средства
сотрудники, обслуживающие средства связи ИС	низкий	аппаратных средств связи	штатные средства
сотрудники, обслуживающие КШМ	низкий	ТО и эксплуатации КШМ	штатные средства

Характеристика внешних нарушителей приведена в таблице 3.

Таблица 3 – Характеристика внешних нарушителей.

Нарушитель	Возможные цели (мотивация) реализации угроз безопасности информации	Потенциал нарушителя	Возможности по реализации угроз безопасности информации
специальные службы иностранных государств (блоков государств)	- нанесение ущерба государству, отдельным его сферам деятельности или секторам экономики; - дискредитация или дестабилизация деятельности органов государственной власти, организаций.	высокий	обладают всеми возможностями нарушителей с базовым и базовым повышенным потенциалами; имеют возможность создания и применения специальных технических средств для добывания информации (воздействия на информацию или технические средства), распространяющейся в виде физических полей или явлений

Продолжение таблицы 3.

Нарушитель	Возможные цели (мотивация) реализации угроз безопасности информации	Потенциал нарушителя	Возможности по реализации угроз безопасности информации
Террористические и экстремистские группировки	<ul style="list-style-type: none"> - нанесение ущерба государству, отдельным его сферам деятельности или секторам экономики; - совершение террористических актов; - идеологические или политические мотивы; - дестабилизация деятельности органов государственной власти, организаций. 	средний	<p>обладают всеми возможностями нарушителей с базовым потенциалом;</p> <p>имеют осведомленность о мерах защиты информации, применяемых в информационной системе данного типа.</p>

Характеристика внутренних нарушителей приведена в таблице 4.

Таблица 4 – Характеристика внутренних нарушителей

Нарушитель	Возможные цели (мотивация) реализации угроз безопасности информации	Потенциал нарушителя	Возможности по реализации угроз безопасности информации
оператор АРМ	<ul style="list-style-type: none"> - непреднамеренные, неосторожные или неквалифицированные действия; - выявление уязвимостей с целью их дальнейшей продажи и получения финансовой выгоды. 	низкий	<p>имеют возможность получить информацию об уязвимостях отдельных компонентов информационной системы, опубликованных в общедоступных источниках</p>

Продолжение таблицы 4.

Нарушитель	Возможные цели (мотивация) реализации угроз безопасности информации	Потенциал нарушителя	Возможности по реализации угроз безопасности информации
сотрудники, обслуживающие средства связи ИС	непреднамеренные, неосторожные или неквалифицированные действия; - любопытство.	низкий	имеют возможность получить информацию об уязвимостях отдельных компонентов информационной системы, опубликованных в общедоступных источниках
сотрудники обслуживающие техническое средство, в котором расположена ИС	- непреднамеренные, неосторожные или неквалифицированные действия; - любопытство.	низкий	имеют возможность получить информацию об уязвимостях отдельных компонентов информационной системы, опубликованных в общедоступных источниках

2.3 Актуальные угрозы безопасности информации

Объектами защиты для АРМ оператора аппаратной связи наземного модуля сопряжения являются:

- комплекс специализированного программного обеспечения для аппаратной связи модуля сопряжения;
- информация предназначенная для передачи абонентам;
- операционная система и набор системного программного обеспечения;
- локальная база данных АРМ;
- информация о системе защиты системы.

При обработке ПДн на автоматизированном рабочем месте возможна реализация следующих угроз:

- угроз утечки информации по техническим каналам;
- угроз НСД к информации, обрабатываемым в автоматизированном рабочем месте.

Угрозы информационной безопасности, рассматриваемые ИС, представлены в таблице 5.

Таблица 5 – Угрозы ИБ

№	Угроза
1	угроза утечки информации по каналам ПЭМИН
2	угроза утечки информации по акустическим каналам
3	угроза сбора информации о системе защиты
4	угроза доступа в среду функционирования прикладных программ
5	угроза доступа в операционную среду, то есть в среду функционирования локальной операционной системы
6	угроза доступа непосредственно к информации абонентов

Уязвимости, которые могут быть использованы при реализации угроз ИБ, рассматриваемой ИС представлены в таблице 6.

Данные типы уязвимостей характерны для ИС, состоящих из АРМ без доступа к внешним сетям и сетям общего доступа.

Таблица 6 – Уязвимости

№	Уязвимости
1	наличие функции, процедуры, изменение параметров которых определенным образом позволяет использовать их для несанкционированного доступа без обнаружения таких изменений операционной системой
2	наличие функции и процедуры, относящиеся к разным прикладным программам и несовместимые между собой (не функционирующие в одной операционной среде) из-за конфликтов, связанных с распределением ресурсов системы
3	отсутствие необходимых средств защиты
4	недостаточность организационных мер доступа к ИС
5	несовершенство СЗИ от НСД используемых в ИС

Были определены и описаны возможные сценарии инцидентов для идентифицированных угроз. Данные сценарии описаны ниже.

Описание сценария «Угроза утечки информации по каналам ПЭМИН» представлено в таблице 7.

Таблица 7 – Описание сценария «Угроза утечки информации по каналам ПЭМИН»

Характеристика	Возможные варианты
угроза	угроза утечки информации по каналам ПЭМИН
источник угрозы	специальные службы иностранных государств
объект нападения	информация предназначенная для передачи абонентам
метод нападения	использование программно аппаратных средств съема ПЭМИН
уязвимость	недостаточность организационных мер доступа к ИС
последствия инцидента	перехват информации предназначенной абонентам

Таблица 8 – Описание сценария «Угроза утечки информации по акустическим каналам»

Характеристика	Возможные варианты
угроза	угроза утечки информации по акустическим каналам
источник угрозы	специальные службы иностранных государств и террористические, экстремистские группировки
объект нападения	информация предназначенная для передачи абонентам
метод нападения	прослушивание разговоров с помощью направленных микрофонов
уязвимость	- недостаточность организационных мер доступа к ИС; - отсутствие необходимых средств защиты.
последствия инцидента	перехват информации предназначенной абонентам

Таблица 9 – Описание сценария «Угроза доступа в среду функционирования прикладных программ»

Характеристика	Возможные варианты
угроза	угроза доступа в среду функционирования прикладных программ
источник угрозы	- сотрудники, обслуживающие средства связи ИС; - сотрудники обслуживающие техническое средство, в котором расположена ИС.
объект нападения	- комплекс специализированного программного обеспечения для аппаратной связи модуля сопряжения; - информация предназначенная для передачи абонентам.
метод нападения	использование прикладного программного обеспечения для просмотра, модификации или удаления информации
уязвимость	- недостаточность организационных мер доступа к ИС; - несовершенство СЗИ от НСД используемых в ИС.
последствия инцидента	- нарушение свойств целостности, доступности и конфиденциальности информации предназначенной абонентам; - нарушение функционирования прикладного программного обеспечения, в том числе специального программного обеспечения модуля сопряжения.

Таблица 10 – Описание сценария «Угроза сбора информации о системе защиты»

Характеристика	Возможные варианты
угроза	угроза сбора информации о системе защиты
источник угрозы	- сотрудники, обслуживающие средства связи ИС; - сотрудники обслуживающие техническое средство, в котором расположена ИС; - оператор АРМ.
объект нападения	информация о системе защиты
метод нападения	получения сведений об ИС, способствующих созданию благоприятных условий для применения других методов
уязвимость	недостаточность организационных мер на территории контролируемой зоны ИС
последствия инцидента	последствия реализации не причиняют какой-либо ущерб обрабатываемой информации, но дают возможность реализовать непосредственные угрозы, перечисленные выше.

Таблица 11 – Описание сценария «Угроза доступа в операционную среду»

Характеристика	Возможные варианты
угроза	угроза доступа в операционную среду, то есть в среду функционирования локальной операционной системы
источник угрозы	- сотрудники, обслуживающие средства связи ИС; -сотрудники обслуживающие техническое средство, в котором расположена ИС.
объект нападения	-операционная система и набор системного программного обеспечения; - локальная база данных АРМ.
метод нападения	-использование системного программного обеспечения для просмотра, модификации или удаления информации; - запуск вредоносных программ.
уязвимость	- отсутствие необходимых средств защиты; - недостаточность организационных мер доступа к ИС; - несовершенство СЗИ от НСД используемых в ИС.
последствия инцидента	возможностью выполнения несанкционированного доступа путем вызова штатных программ операционной системы или запуска специально разработанных программ, реализующих такие действия

Таблица 12 – Описание сценария «Угроза доступа непосредственно к информации абонентов»

Характеристика	Возможные варианты
угроза	угроза доступа непосредственно к информации абонентов
источник угрозы	- сотрудники обслуживающие средства связи ИС; сотрудники обслуживающие техническое средство, в котором расположена ИС; - оператор АРМ.
объект нападения	- информация предназначенная для передачи абонентам; - локальная база данных АРМ.
метод нападения	доступ у информации с использование штатного программного обеспечения
уязвимость	наличие функции, процедуры, изменение параметров которых определенным образом позволяет использовать их для несанкционированного доступа
последствия инцидента	НСД к информации абонентов

3 Формирование требований к комплексу и проектирование

Для формирования требований к комплексу мониторинга было проведено обследование объекта мониторинга. Объектом мониторинга является АРМ оператора наземного модуля сопряжения. Так, как в данном проекте разрабатывается программный комплекс мониторинга, аспекты, связанные с технической, документационной и кадровой составляющей системы мониторинга информационной безопасности.

Основным источником событий информационной безопасности является операционная система установленная на АРМ оператора.

Для данного источника был определен следующий список действий и событий, мониторинг которых будет проводиться:

- запуск сеанса работы учетной записи;
- запуск процессов в системе;
- изменение целостности файлов;
- попытки доступа и авторизации к системе мониторинга.

События запуска процессов в операционной системе и изменения целостности файлов, необходимо обрабатывать в соответствии с правами доступа активного пользователя. События запуска сеанса работы учетной записи пользователя, необходимо для определения текущего активного пользователя и его прав доступа.

Для комплекса мониторинга были выявлены следующие типы инцидентов:

- несанкционированный запуск программы (запуск нового процесса или процессов в системе);
- несанкционированная модификация или удаление защищаемых файлов.

Архитектура комплекса мониторинга должна соответствовать требованиям отказоустойчивости и надежности.

Для разработки архитектуры комплекса был определен порядок запуска и настройки комплекса мониторинга.

При первом запуске необходимо провести инициализацию комплекса и создать учетную запись администратора. Это обеспечит доступ к комплексу мониторинга только администратору. После инициализации необходимо настроить комплекс для работы. При этом необходимо учитывать, что настройки комплекса в соответствии с политикой безопасности могут изменяться и это не должно затрагивать и приостанавливать работу комплекса мониторинга.

Для обеспечения безопасности комплекса при его первом запуске в системе и возможности его администрирования без прерывания процесса работы, в комплексе должны быть реализованы следующие режимы:

- режим инициализации комплекса;
- режим администрирования комплекса;
- режим работы комплекса.

Исходя из определенных для комплекса требований была, спроектирована структура комплекса мониторинга.

Для обеспечения работы в различных режимах комплекс мониторинга разбит на два модуля:

- модуль инициализации и администрирования;
- модуль мониторинга.

Модуль мониторинга для обеспечения надежности решено разбить на три агента мониторинга. Каждый из агентов должен обладать способностью, автоматически запускаться при аварийном или принудительном завершении. Два агента отвечают за аудит событий, а третий функционирует в качестве сервера обработки и анализа событий, выявленных в ходе аудита.

Для хранения журнала аудита, решено использовать СУБД PostgreSQL.

4 Программная реализация комплекса мониторинга

Комплекс мониторинга состоит из двух программных модулей:

- модуль инициализации и администрирования;
- модуль мониторинга.

Модуль инициализации и администрирования предназначен для первоначальной инициализации комплекса перед запуском, а также для его последующего администрирования и представления отчета о результатах мониторинга.

Модуль мониторинга предназначен для отслеживания запуска процессов в операционной системе, проверки целостности файлов отслеживаемого программного обеспечения, обработки и журналирования результатов мониторинга.

Комплекс может функционировать в режимах «Инициализация», «Администрирование» и «Работа».

Режим «Инициализация» предназначен для создания учетной записи администратора комплекса мониторинга, создания таблиц базы данных, а также проверки выполнения условий запуска комплекса мониторинга.

Режим «Администрирование» предназначен для настройки комплекса мониторинга, управления файлами программного обеспечения и учетными записями пользователей, а также просмотра журнала мониторинга.

Режим «Работа» предназначен, для мониторинга запускаемых процессов в операционной системе, а также проверки контроля целостности файлов программного обеспечения, внесенных в список отслеживаемых.

В состав комплекса входят база данных PostgreSQL, файл «adminGUI.exe», файл «monitor.exe» и скрипт «initsystemmonitoring.sh».

4.1 Разработка базы данных для комплекса мониторинга

База данных для комплекса мониторинга хранит записи журнала мониторинга, списки пользователей и программ, информацию об администраторе и настройки комплекса.

База данных создается перед инициализацией комплекса администратором операционной системы. Необходимо создать, только базу данных, все необходимые таблицы комплекс мониторинга создаст автоматически при инициализации комплекса.

Для работы с базой данных необходимо создать трех пользователей базы данных. Это администратор, «monitoringagent» и «initagent».

Пользователи «monitoringagent» и «initagent» предназначены для работы комплекса. Через учетную запись «monitoringagent» комплекс мониторинга получает доступ на запись в таблицу базы данных «journal» отчетов, а также получает доступ на чтение из таблицы «settings» настроек для работы комплекса мониторинга. Через учетную запись «initagent» комплекс мониторинга получает доступ на чтение из таблицы базы данных «admin» для проверки создания учетной записи администратора.

Пользователи «monitoringagent» и «initagent» имеют прав на доступ к другим таблицам в базе данных, а также прав на внесение изменений в таблицы доступ, к которым им предоставлен. Изменять имена данных пользователей запрещено.

Администратор базы данных системы мониторинга имеет полные доступ к базе данных. Данный пользователь используется при подключении комплекса мониторинга к базе данных в режиме «Администрирования». Имя пользователя и настройки учетной записи задаются администратором системы и вводятся в программу при инициализации.

Базы данных для системы мониторинга состоит из следующих таблиц:

- journal;
- users;
- accessforuser;

- program;
- admin;
- setting.

Таблица «journal» служит для хранения записей журнала мониторинга.

Структура таблицы «journal» базы данных представлена в таблице 13.

Таблица 13 – Структура таблицы базы данных «journal»

Название поля таблицы в базе данных	data	time	username	typeevent	logentry
Пояснение	дата	время	пользователь	тип события	подробный отчет

Таблица «users» служить для хранения учетных записей пользователей. Учетная запись пользователя подразумевает информацию о пользователе и права доступа к программам в системе. По умолчанию в таблице создаются две записи «root» и «defaultUser». Пользователь «root» предназначен для администратора операционной системы и имеет права на запуск любых процессов. Пользователь «defaultUser» применяется для всех учетных записей системы, которые не определены в комплексе мониторинга, то есть, нет соответствующего пользователя в таблице «users». Данный пользователь имеет права на запуск только минимального набора системных процессов.

Структура таблицы «users» базы данных представлена в таблице 14.

Таблица 14 – Структура таблицы «users» базы данных

Название поля таблицы в базе данных	userlogin	username	password	iduser
Пояснение	логин	имя	хэш-функция пароля	идентификатор пользователя

Таблица «accessforuser» предназначена для хранения информации о доступных пользователю процессах (файлов, запускающих данный процесс).

Права пользователя на запуск программ хранятся в отдельной таблице от информации о пользователе, для того, чтобы при смене информации о пользователе не пришлось менять ее во всех записях таблицы, а можно было сменить только в одном месте.

Структура таблицы «accessforuser» базы данных представлена в таблице 15.

Таблица 15 – Структура таблицы базы данных «accessforuser»

Название поля таблицы в базе данных	iguser	idfiles
Пояснение	идентификатор пользователя	идентификатор файла, запускающего процесс

Таблица «program» предназначена для хранения информации о файлах, которые запускают в операционной системе процессы или являются частью программного обеспечения информационно системы. Помимо общей информации о файле, хранится также контрольная сумма для файла, для осуществления проверки целостности.

Структура таблицы «files» базы данных представлена в таблице 16.

Таблица 16 – Структура таблицы «files» базы данных

Название поля	filename	filepath	flagincludein monitoring	controlsum	idfile
Пояснение	имя программы	путь к файлу программы	флаг контроля целостности	контрольная сумма	идентификатор программы

Таблица «admin» используется в служебных целях, для дополнительной проверки аутентификационных данных и упрощения работы в режиме администрирования.

Структура таблицы «admin» базы данных представлена в таблице 17.

Таблица 17 – Структура таблицы «admin» базы данных

Название поля таблицы в базе данных	login	name	flagcreate	databaseinfo
Пояснение	логин в зашифрованном виде	имя в зашифрованном виде	флаг, подтверждающий создание	зашифрованные данные о администраторе

Таблица «setting» предназначена для хранения настроек для системы мониторинга.

Структура таблицы «setting» базы данных представлена в таблице 18.

Таблица 18 – Структура таблицы «setting» базы данных

Название поля таблицы в базе данных	typecryptalg	setting	settingflags
Пояснение	логин	флаги для настройки	флаги для настройки

Общая структура базы данных продемонстрирована на рисунке 3.

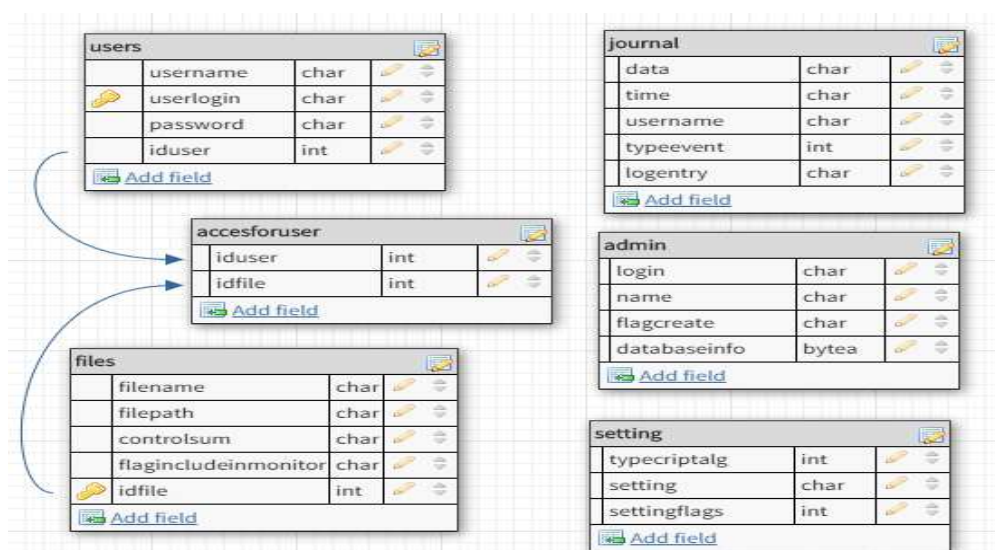


Рисунок 3 – Структура базы данных для комплекса мониторинга

4.2 Модули комплекса мониторинга

Комплекс мониторинга состоит из двух программных модулей:

- модуль инициализации и администрирования;
- модуль мониторинга.

Каждому модулю соответствует свой исполняемый файл. Для модуля инициализации и администрирования – файл «adminGUI.exe», для модуля мониторинга – файл «monitor.exe».

Модуль инициализации и администрирования отвечает за работу комплекса в режимах «Инициализация» и «Администрирование».

Модуль состоит из следующих программных классов:

- класс MainWindow;
- класс WorkWithDataBase;
- класс SMCore.

Класс MainWindow реализует графический интерфейс программы и отвечает за обмен данными между администратором и ядром системы. Реализация объявления класса представлена в приложении А.

Помимо отображения графических окон и ведения диалогового режима с администратором комплекса мониторинга, данный класс при запуске комплекса данный класс иницирует процессы проверки выполнения условий запуска и наличия созданной учетной записи администратора комплекса мониторинга.

Класс SMCore предназначен для предоставления необходимых функций для графического интерфейса пользователя. В инкапсулированы классы и функций необходимые для выполнения задач инициализации и администрирования. Реализация объявления класса представлена в приложении Б.

Класс WorkWithDataBase предназначен для работы с базой данных. Он предоставляет классу SMCore интерфейс для подключения и работы с базой данных PostgreSQL. Наследует функции класса ConnectAndWorkWithDataBase. Реализация объявления класса представлена в приложении В.

Модуль мониторинга отвечает за работу комплекса в режиме «Работа».

Модуль состоит из следующих программных классов:

- класс AbstractDaemon;
- класс AbstractNotifyingDaemon;
- класс DaemonProcessMonitor;
- класс DaemonFileMonitor;
- класс DaemonAgentMonitoring;
- класс CommunicateWithDataBase.

Классы AbstractDaemon и AbstractNotifyingDaemon – являются абстрактными классами, не имеющими экземпляров. Класс AbstractDaemon реализует функции необходимые для запуска программ в фоновом режиме. Он содержит виртуальную функцию – mainLoop, которую должны реализовать классы наследники. В функции mainLoop проводится вся работа фоновой программы. Реализация объявления данных классов представлены в приложении Г.

Класс AbstractNotifyingDaemon наследует функции класса AbstractDaemon и ClientSocketForDaemon, и расширяет их функциями для обмена информацией с помощью механизма сокетов.

Классы DaemonProcessMonitor и DaemonFileMonitor наследуют функции класса AbstractNotifyingDaemon. В них реализованна функция mainLoop в соответствии с задачами решаемыми данными классами. Данные классы продемонстрированы в приложении Г.

Класс DaemonProcessMonitor решает задачи мониторинга запускаемых процессов в системе и оповещения обо всех происходящих событиях класса DaemonAgentMonitoring.

Класс DaemonFileMonitor решает задачи мониторинга контроля целостности файлов программного обеспечения информационной системы и оповещения о нарушениях контроля целостности класса DaemonAgentMonitoring.

Класс DaemonAgentMonitoring наследует функции классов AbstractDaemon и ServerSocketForDaemon, и расширяет их функциями анализа, обработки и журналирования информации о событиях, приходящих от классов

DaemonProcessMonitor и DaemonFileMonitor. Реализация объявления класса DaemonAgentMonitoring представлена в приложении 3.

Класс CommunicateWithDataBase реализует функции для подключения к базе данных комплекса мониторинга и записи в таблицу «journal». Наследует функции класса ConnectAndWorkWithDataBase.

Также данные модули содержат набор общих программных классов:

- класс ServerSocketForDaemon;
- класс ClientSocketForDaemon;
- класс ConnectAndWorkWithDataBase;
- класс CriptModule;
- класс WorkWithProc.

Классы ServerSocketForDaemon и ClientSocketForDaemon предназначены для создания сервера и клиентов для обмена информацией с помощью механизма сокетов.

Класс ConnectAndWorkWithDataBase предоставляет общие функции для подключения и работы с базой данных PostgreSQL.

Класс CriptModule предназначен для предоставления различных криптографических функций. Он реализует алгоритмы нахождения Хэш-функций и криптографические алгоритмы.

Класс WorkWithProc предназначен для работы с операционной системой Astra Linux. Он определяет запущенные процессы и считывает информацию о них. Также он определяет информацию о учетной записи активного пользователя операционной системы.

4.3 Режим работы комплекса мониторинга «Инициализация»

Перед запуском комплекса необходимо подготовить операционную систему. Для этого необходимо отключить другие системы мониторинга для избежания конфликтов. Затем необходимо установить СУРБД PostgreSQL и настроить ее для работы с комплексом мониторинга.

Комплекс мониторинга должен запускаться из учетной записи администратора системы (пользователя root).

При запуске комплекс мониторинга проверяет, учетную системную запись в сеанс работы, которой был запущен.

Если комплекс был запущен не под учетной записью администратора системы, то он завершает свою работу.

Если при запуске комплекса в системе не было обнаружено учетной записи администратора комплекса мониторинга, то комплекс переходит в режим «Инициализация».

При этом проверяется наличие в системе базы данных PostgreSQL и если в системе база данных отсутствует, то администратору выводится сообщение об этом и комплекс завершает работу.

Если база данных обнаружена, то администратору предлагается ввести данные для дальнейшего входа в комплекс мониторинга, а также данные для работы с базой данных.

После заполнения данных, комплекс проверяет правильность их заполнения, проводит необходимую настройку и пытается подключиться к базе данных.

Если при заполнении данных была допущена ошибка или не удастся подключиться к базе данных, то инициализация считается неуспешной и пользователю выводится сообщение о неудачной попытке инициализации и указания для дальнейших действий. Если инициализация прошла успешно, то комплекс переходит в режим «Администрирование».

В режиме инициализации администратору, выводится графическое окно информирующее, о том, что учетная запись администратора комплекса мониторинга не создана и предложением начать регистрацию.

Данное графическое окно представлено на рисунке 4.

После нажатия на кнопку «Начать регистрацию» пользователю необходимо пройти два этапа регистрации, на которых ему предложат ввести данные которые,

будут использоваться для входа в комплекс мониторинга и данные для доступа комплекса к базе данных.

Графические окна с формами для заполнения данных, выводимые пользователю программой, представлены на рисунках 5 и 6.

На втором этапе регистрации вводятся данные для доступа к предварительно созданной базе данных для комплекса мониторинга.

После заполнения форм на этапах 1 и 2 комплекс производит попытку создать учетную запись и в случае успеха выводит графическое окно, представленное на рисунке 7.

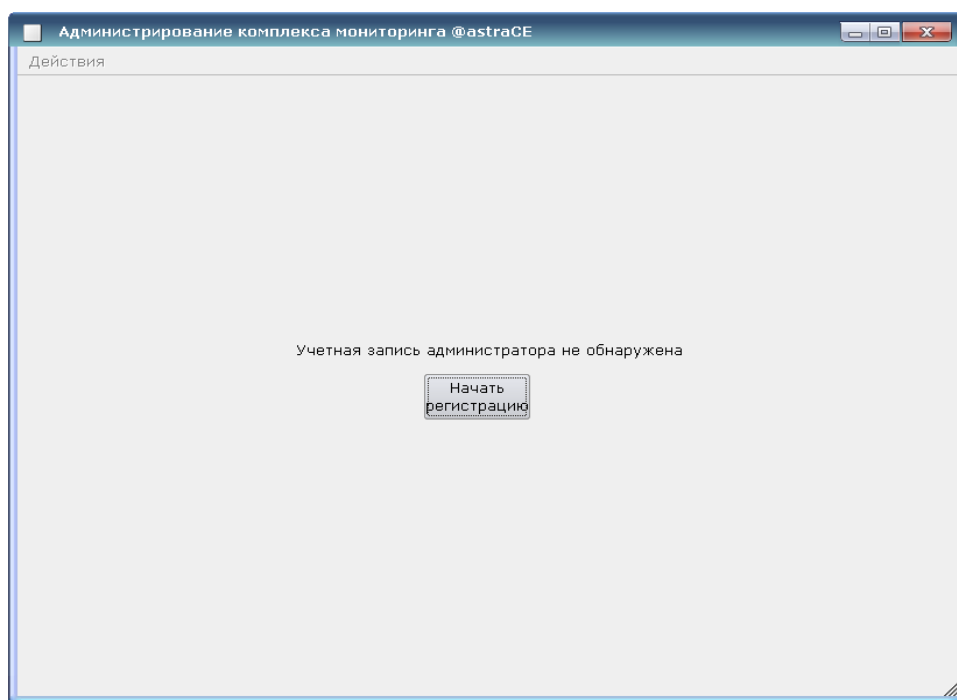


Рисунок 4 – Графическое окно начала регистрации

Администрирование комплекса мониторинга @astraCE

Действия

Этап 1
Создание учетной записи

Заполните поля

Логин*: admin

Пароль*: *****

Повторите пароль*: *****

Имя администратора alex

Далее

Поля помеченные * обязательны для заполнения

Рисунок 5 – Графическое окно с формой для регистрации, этап 1.

Если создать учетную запись не удалось, то выводится сообщение с описанием причин ошибки создания. Пользователю предлагается заново ввести данные или завершить работу с комплексом.

При создании учетной записи, происходит проверка корректности введенных данных. За это отвечает класс MainWindow.

Администрирование комплекса мониторинга @astraCE

Действия

Этап 2
Настройка подключения к базе данных
Введите логин и пароль администратора базы данных

Название базы данных database

Логин adminsystemmonito

Пароль *****

Далее

Рисунок 6 – Графическое окно с формой для регистрации, этап 2.

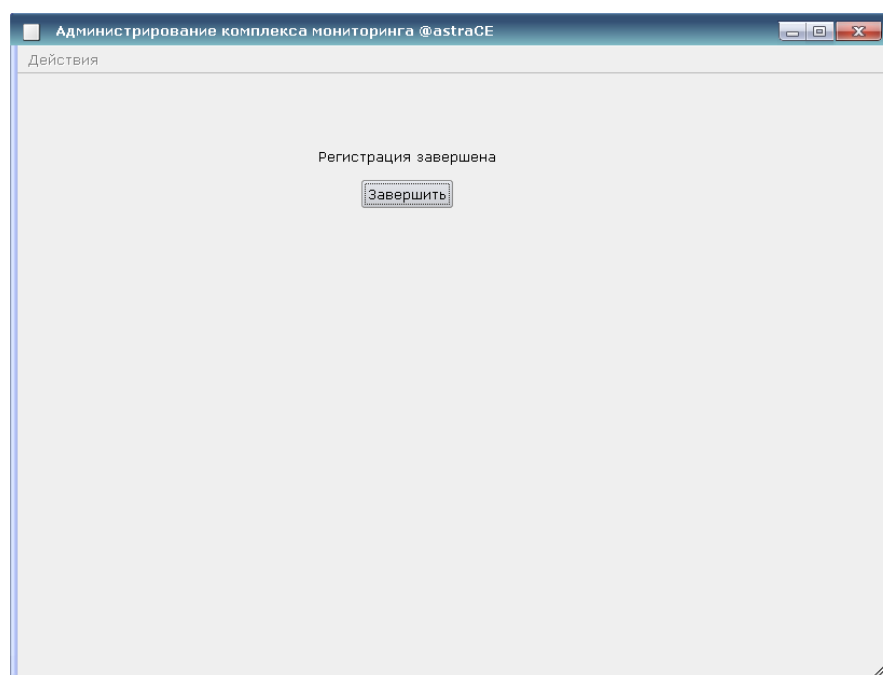


Рисунок 7 – Графическое окно завершения регистрации.

После данные из форм класс `MainWindow` передает в качестве параметров функции `initSM` класса `SMCore`. В данном классе производится попытка подключения к базе данных системы мониторинга с указанными при регистрации учетными данными. После подключения к базе данных, в функции `initSM` запускаются функции для создания всех необходимых для работы системы мониторинга таблиц в базе данных, а затем происходит заполнение таблиц начальными значениями.

Введенные при регистрации значения логина и имени базы данных заносятся в таблицу «admin». Логин заносится в зашифрованном виде. Для шифрования используется алгоритм DES. Для шифрования используется функция `cryptData`, класса `CryptModule`, с параметром DES.

После происходит шифрование данных для входа в базу данных. Логин и пароль для администратора базы данных объединяются в одну строку с добавлением специального разделителя. Данная строка шифруется с помощью алгоритма DES. В качестве ключа используется последовательность байт полученная в результате нахождения Хэш-функции от результата конкатенации логина и пароля учетной записи администратора комплекса мониторинга. Результат

шифрования данных для авторизации администратора базы данных, заносится в таблицу «admin» базы данных.

Впоследствии при авторизации из базы данных с помощью пользователя «initagent» с минимальными правами на доступ, считывается данные для авторизации администратора базы данных из таблицы «admin» и расшифровываются с помощью введенного логина и пароля администратора комплекса мониторинга.

4.4 Режим работы комплекса мониторинга «Администрирование»

В режиме «Администрирование» администратору необходимо авторизоваться для дальнейшей работы. Для этого классом MainWindow предоставляется графическое окно для авторизации, представленное на рисунке 8.

Если логин и пароль введены неверно, то выводится соответствующая надпись, в том же графическом окне.

В случае успешной авторизации, выводится графическое окно с возможными дальнейшими действиями для пользователя. Данное графическое окно представлено на рисунке 9.

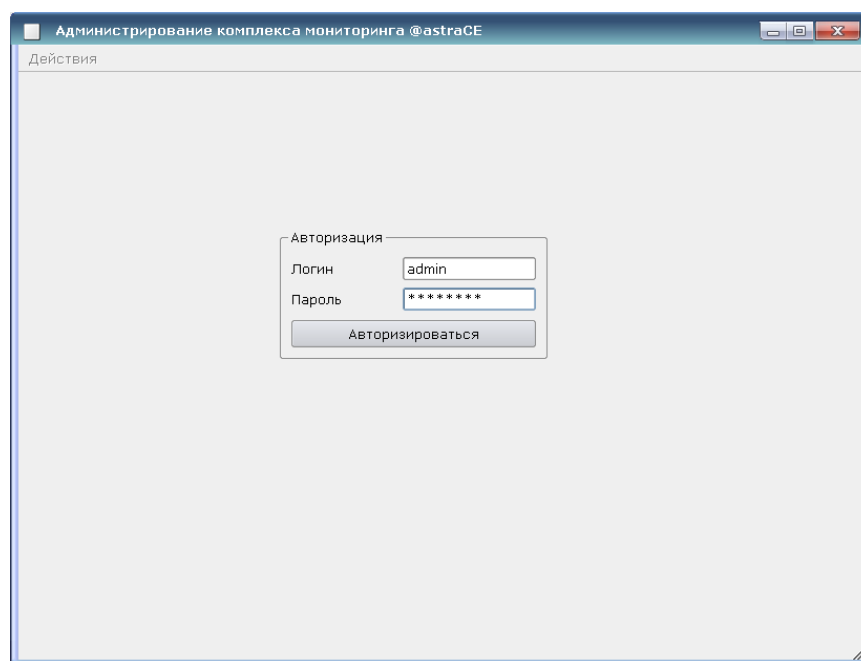


Рисунок 8 – Окно для введения данных для авторизации.

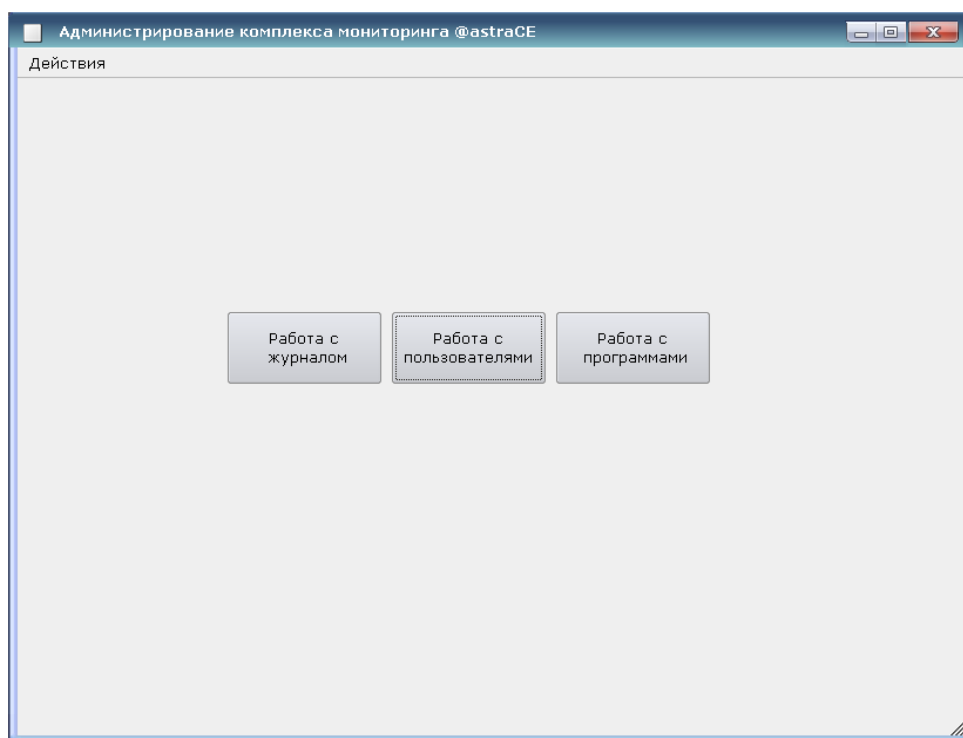


Рисунок 9 – Графическое окно с действиями для пользователя после прохождения авторизации.

Также становится активным меню «Действия» в верхнем левом углу. В данном меню можно выбрать доступные действия над комплексом мониторинга.

В режиме «Администрирование» можно просматривать журналы мониторинга и проводить настройку комплекса.

Во время настройки комплекса необходимо заполнить список файлов программного обеспечения, которые будут запускать разрешенные процессы.

Графическое окно для работы с файлами отображено на рисунке 10.

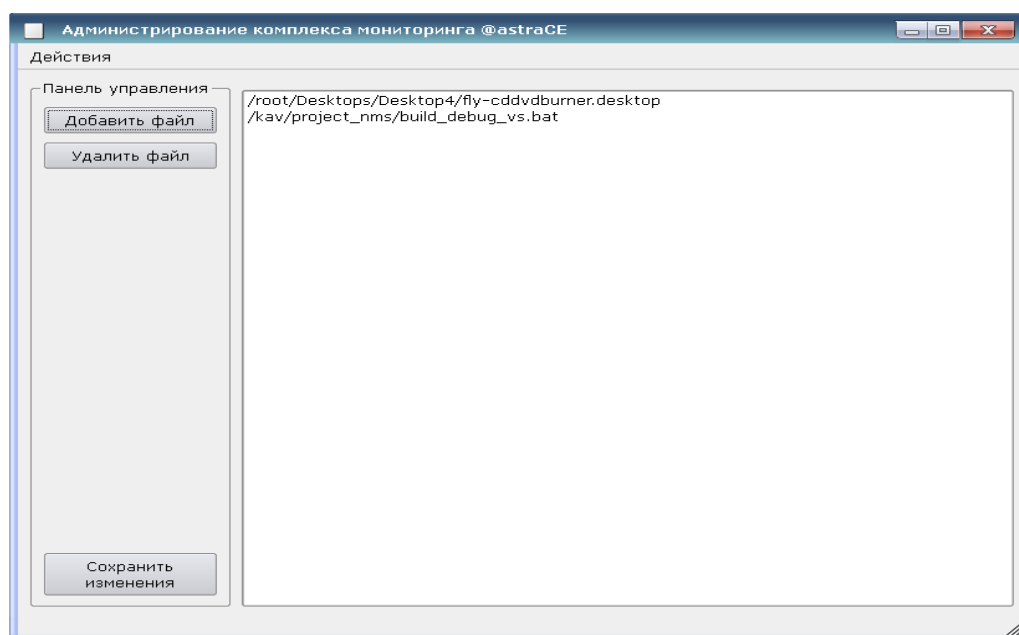


Рисунок 10 – Графическое окно для работы с файлами.

При добавлении файла можно указать, что для файла необходимо проводить проверку контроля целостности.

При нажатии на кнопку «Добавить файл» выводится диалог выбора файла для добавления. После выбора файл появляется в списке включенных для мониторинга файлов. При нажатии на кнопку «Удалить файл» выбранный в списке файл будет из него удален. Добавление и удаление файлов происходит из локального списка, и окончательно список сохраняется в базу данных после нажатия кнопки «Сохранить изменения».

Перед сохранением списка файлов программного обеспечения, комплекс находит все файлы из списка помеченные для контроля целостности. Для этих файлов находится контрольная сумма, которая будет эталонной контрольной суммой. Контрольной суммой является Хэш полученный с помощью алгоритма MD5. Функция хэширования с помощью алгоритма MD5 находится в классе CryptModule.

Эталонная информация вместе со списком файлов сохраняется в базу данных.

Также во время настройки комплекса мониторинга необходимо запослнить список пользователей операционной системы с заданными правами на запуск процессов из списка файлов программного обеспечения.

Графическое окно для работы с пользователями продемонстрировано на рисунке 11.

Всем пользователям, которые не указаны в системе мониторинга, будет присвоен список файлов по умолчанию. В список файлов по умолчанию включены только процессы необходимые для работы операционной системы.

После нажатия на кнопку «Создать нового пользователя» выводится графическое окно, представленное на рисунке 12.

При создании нового пользователя необходимо указать логин, пароль и имя пользователя, а также добавить файлы в список доступных пользователю файлов программного обеспечения.

Список файлов программного обеспечения пользователя формируется из общего списка файлов программного обеспечения, внесенного администратором. При удалении из списка файлов программы, данная программа исчезнет и из списка файлов программного обеспечения пользователя.

Создание нового пользователя происходит после нажатия кнопки «Завершить создание». Программа после нажатия данной кнопки возвращается к графическому окну работы с пользователями, а в списке пользователей появляется новая созданная запись. Созданную учетную запись можно удалить или отредактировать.

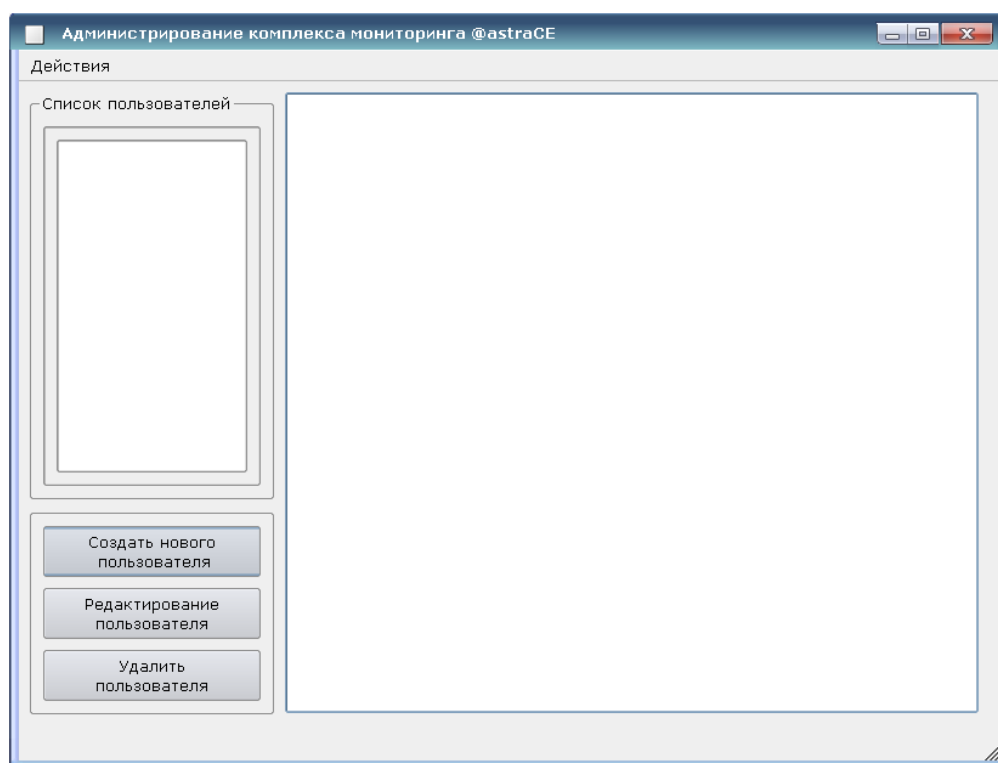


Рисунок 11 – Графическое окно для работы с пользователями

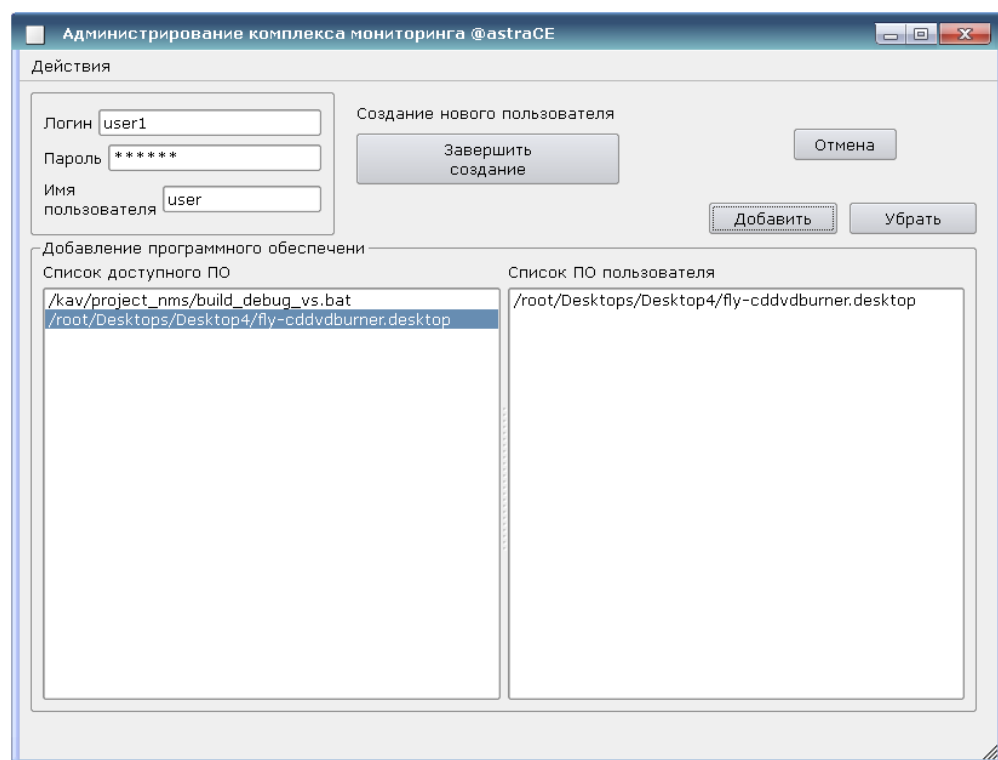


Рисунок 12 – Графическое окно для создания нового пользователя

Для редактирования учетных записей пользователей, необходимо выбрать одну из учетных записей пользователя и нажать на кнопку «Редактирование пользователя». Программа выведет на экран графическое окно, представленное на рисунке 13.

При редактировании можно изменить логин, пароль и имя пользователя, а также список разрешенных файлов программного обеспечения. Изменения сохраняются после нажатия кнопки «Завершить редактирование», а программа после нажатия данной кнопки возвращается к графическому окну работы с пользователем.

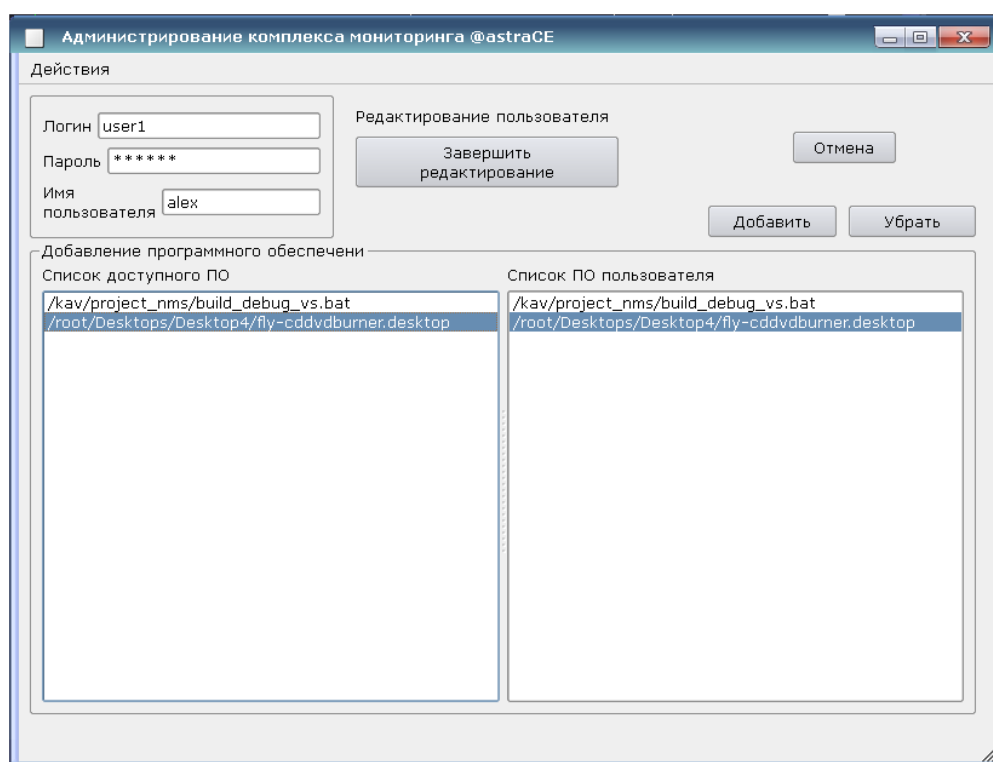


Рисунок 13 – Графическое окно для редактирования пользователя

Для удаления учетной записи пользователя необходимо выбрать пользователя и нажать кнопку «Удалить пользователя».

Учетную запись администратора комплекса мониторинга, также можно редактировать. Это необходимо при смене администратора системы или изменении

используемой базы данных. Для работы с учетной записью администратора, в разделе меню «Действия» необходимо выбрать пункт «Учетная запись».

После выбора пункта будет выведено графическое окно, представленное на рисунке 14.

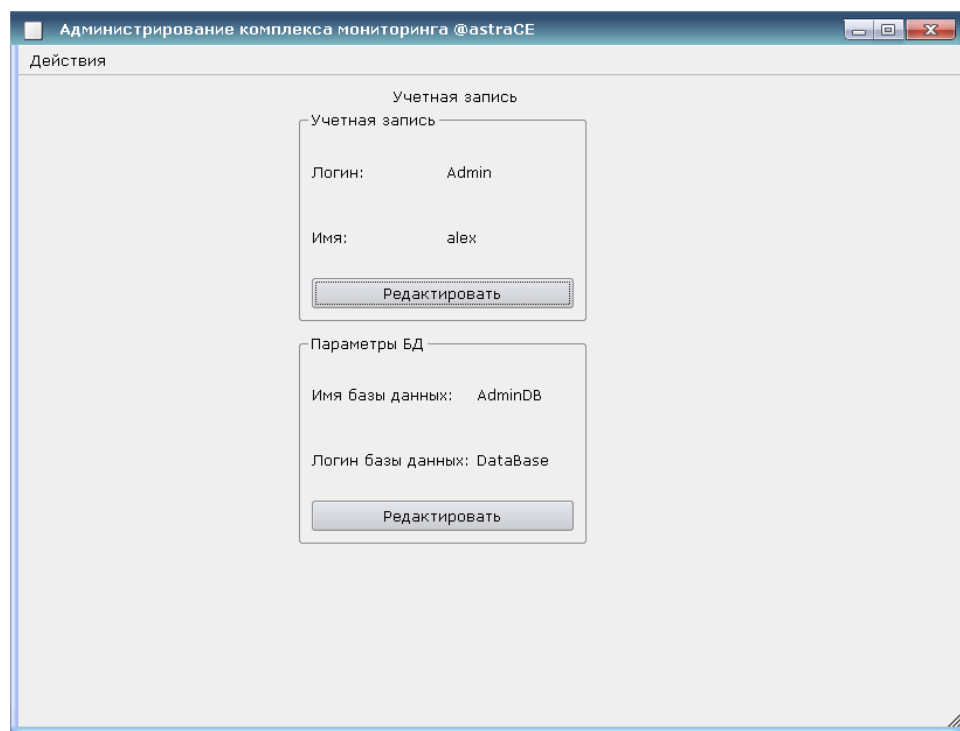


Рисунок 14 – Информация об учетной записи администратора

В данном окне есть возможность начать редактирование необходимой информации учетной записи.

Для этого нужно нажать на кнопку «Редактирование» в нужном поле.

В результате будет выведено графическое меню, где можно редактировать введенные данные.

При смене пароля от учетной записи администратора, необходимо также ввести старый пароль, для подтверждения действия.

При смене пароля для доступа к базе данных, необходимо помимо нового пароля указать пароль от учетной записи, для подтверждения действия.

Графические окна, выводимые для редактирования информации об учетной записи представлены на рисунках 15 и 16.

Для сохранения изменений необходимо нажать на кнопку «Завершить редактирование».

После завершения редактирования, программа вернется к окну информации об учетной записи.

The screenshot shows a Windows-style application window titled "Администрирование комплекса мониторинга @astraCE". Inside the window, there is a tab labeled "Действия" (Actions) and a sub-tab titled "Редактирование учетной записи" (Edit Account). The form is divided into two main sections: "Смена учетных данных" (Change Account Data) and "Смена пароля" (Change Password). The "Смена учетных данных" section contains three input fields: "Логин:" (Login) with the value "admin", "Имя:" (Name) with the value "alex", and "Введите пароль" (Enter password). Below these fields is an "Изменить" (Change) button. The "Смена пароля" section contains three input fields: "Введите новый пароль" (Enter new password), "Повторите новый пароль" (Repeat new password), and "Введите старый пароль" (Enter old password). Below these fields is another "Изменить" (Change) button. At the bottom of the form is a large button labeled "Завершить редактирование" (Finish editing).

Рисунок 15 - Редактирование учетной записи администратора системы мониторинга

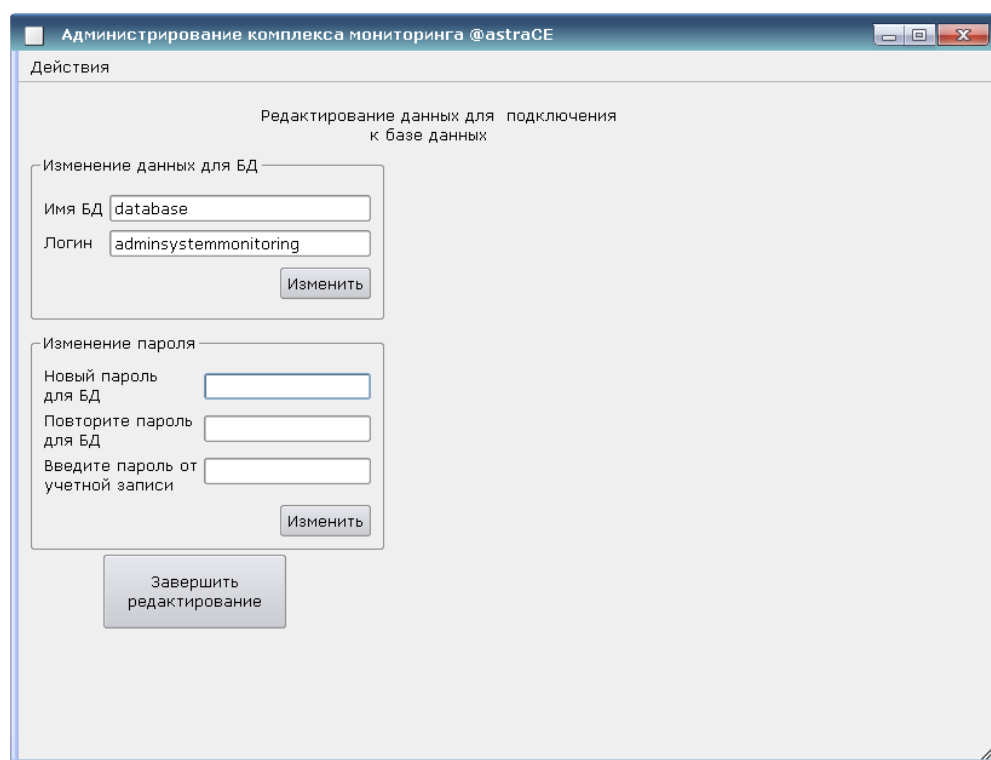


Рисунок 16 – Редактирование данных, для подключения к базе данных.

Для вывода меню настроек необходимо выбрать в меню «Действия» пункт «Настройка комплекса». Будет выведено графическое окно, представленное на рисунке 17.

В настройках можно задать шифрование журнала и выбрать алгоритм шифрования. Настроить политику мониторинга и работы с запускаемыми процессами информационной системы.

Работа с журналом мониторинга в комплексе осуществляется с помощью специального графического окна, представленного на рисунке 18.

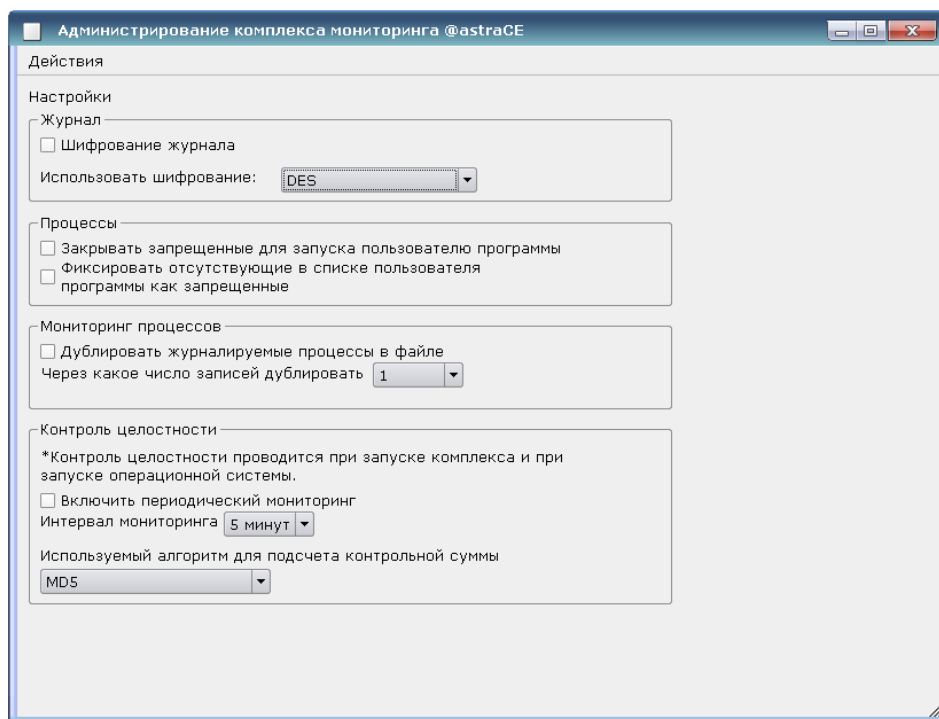


Рисунок 17 – Графическое окно с настройками комплекса.

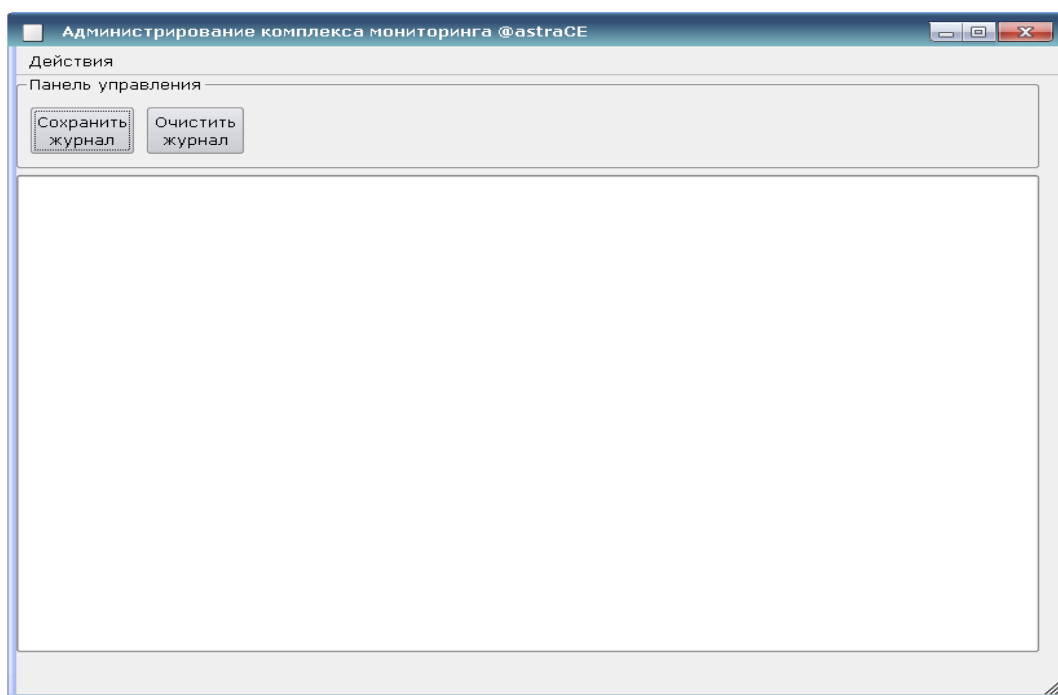


Рисунок 18 – Графическое окно работы с журналом мониторинга.

В данном окне можно просматривать журнал мониторинга, сохранять журнал в формате PDF, TXT, DOC.

4.5 Режим работы комплекса мониторинга «Работа»

После инициализации и настройки комплекса мониторинга он переходит в режим «Работа». В данном режиме комплекс выполняет мониторинг запускаемых процессов и контроль целостности файлов программного обеспечения.

Все зафиксированные события запуска процессов или нарушения целостности файлов, комплекс записывает в журнал мониторинга.

Для мониторинга в данном режиме запускается три фоновых процесса: агент мониторинга запускаемых процессов, агент мониторинга контроля целостности и агент обработки поступающей от агентов мониторинга информации с последующим журналированием.

За реализацию данных программ агентов отвечают классы `DaemonProcessMonitor`, `DaemonFileMonitor`, `DaemonAgentMonitoring` соответственно.

Агенты обладают возможностью самовосстановления при их завершении.

Настройки программ-агентов передаются через базу данных системы мониторинга.

Связующим звеном между агентами и базой данных служит агент обработки информации. Он подключается к базе данных с помощью учетной записи пользователя базы данных «`monitoringagent`». Из базы данных программа считывает информацию о текущих настройках комплекса мониторинга и передает их агентам мониторинга. От агентов мониторинга он получает информацию о результатах мониторинга, обрабатывает ее и формирует отчет, который затем записывает в базу данных.

Обмен информацией между агентами происходит с помощью механизма сокетов. Агент обработки информации является сервером, и ждет поступления информации от агентов мониторинга. Агенты мониторинга являются клиентами и устанавливают связь с сервером для передачи результатов мониторинга и получения настроек комплекса мониторинга.

Механизм сокетов реализован в классах `ServerSocketForDaemon`, `ClientSocketForDaemon`.

Работа с базой данных организована с помощью класса ConnectAndWorkWithDataBase.

Агент мониторинга запускаемых процессов работает с директорией операционной системы «proc». Данная директория в операционной системе предназначена для хранения информации о запущенных процессах. Для работы с данной директорией используется класс WorkWithProc. Он считывает информацию из директории «proc», анализирует и обрабатывает ее.

При запуске агент мониторинга запускаемых процессов запрашивает список разрешенных файлов для активного пользователя системы и сохраняет копию списка у себя. После считывания информации из директории proc он сверяет запущенные процессы со списком файлов. Если пользователю запрещено запускать данных процесс, то агент посылает сигнал завершения процессу и оповещает о выявленном инциденте агента обработки информации. При обнаружении новых записей в директории proc агент считывает и обрабатывает эти записи.

При запуске агент мониторинга контроля целостности запрашивает список файлов, находящихся под контролем целостности и контрольную сумму для них. Проверка контроля целостности заключается в следующем: агент считывает информацию из файла и находит контрольную сумму для считанной информации. Полученная контрольная сумма сравнивается с эталонной контрольной суммой. Результат сравнения передается агенту обработки информации.

Для нахождения контрольной суммы используется алгоритм MD5.

5 Тестирование комплекса мониторинга

Контролю качества должны быть подвергнут программно-аппаратный комплекс в целом, а также его отдельные части.

Так в частности, должно быть проведено тестирование:

- программно - аппаратный комплекс в целом;
- отдельные компоненты программы на тестовых стендах;
- руководство оператора.

Виды проводимых тестов:

- функциональное тестирование;
- тестирование бизнес цикла;
- конфигурационное тестирование;
- тестирование производительности;
- стресс тестирование;
- юзабилити тестирование;
- тестирование инсталляции.

Качество программного обеспечения может быть оценено следующими характеристиками [7]:

- функциональные возможности;
- надежность;
- практичность;
- эффективность;
- сопровождаемость;
- мобильность.

Для каждой характеристики качества были определены подхарактеристики или комплексных показателей. Это является необходимым этапом в определении качества с использованием модели процесса оценивания качества, описанной в стандарте ГОСТ Р ИСО/МЭК 9126-93 [7].

Для характеристики «Функциональные возможности» определены следующие подхарактеристики:

- пригодность;
- правильность;
- способность к взаимодействию.

Для характеристики «Надежность» определены следующие подхарактеристики:

- стабильность;
- устойчивость к ошибке;
- восстанавливаемость.

Для характеристики «Практичность» определены следующие подхарактеристики:

- понятность;
- обучаемость;
- простота использования.

Для характеристики «Эффективность» определены следующие подхарактеристики:

- характер изменения во времени;
- характер изменения ресурсов.

Для характеристики «Сопровождаемость» определены следующие подхарактеристики:

- анализируемость;
- изменяемость;
- устойчивость;
- тестируемость.

Для характеристики «Мобильность» определены следующие подхарактеристики:

- адаптируемость;
- простота внедрения.

Функциональное тестирование – используется для контроля качества «Функциональных возможностей» в части «Пригодности», «Правильности» и «Способности к взаимодействию».

Функциональное тестирование является основным видом тестирования. Проводится вручную через интерфейс пользователя.

Тестирование бизнес цикла – используется для контроля качества «Функциональных возможностей» в части «Пригодности», «Правильности» и «Способности к взаимодействию».

В первую очередь применяется для оценки готовности прототипа и оценки полноты функциональных требований.

Конфигурационное тестирование – используется для контроля качества «Мобильности» в части «Адаптируемости».

Должна быть проверена работоспособность приложения для ОС Astra Linux и СУБД PostgreSQL.

Рекомендованный метод – объединение с функциональным тестированием. В этом случае на каждом рабочем месте тестировщика рекомендуется установка своей конфигурации.

Тестирование производительности – используется для контроля качества «Эффективности».

Для анализа поведения пользовательского интерфейса на реальных объемах данных используется ручное тестирование.

Стресс тестирование – используется для контроля качества «Надежности» в части «Стабильности» и «Устойчивости к ошибке».

Тестирование на удобство использования – используется для контроля качества «Практичности» в части «Понятности», «Обучаемости», «Простоты использования».

Тестирование инсталляции – используется для контроля качества «Мобильности» в части «Простоты внедрения». Проводится вручную.

В ходе проведенных тестов, было подтверждено, что программа соответствует всем характеристикам качества.

6 Разработка программной документации

Единая система программной документации (далее – ЕСПД) — комплекс государственных стандартов, устанавливающих взаимоувязанные правила разработки, оформления и обращения программ и программной документации.

В стандартах ЕСПД устанавливают требования, регламентирующие разработку, сопровождение, изготовление и эксплуатацию программ, что обеспечивает возможность:

- унификации программных изделий для взаимного обмена программами и применения ранее разработанных программ в новых разработках;
- снижения трудоемкости и повышения эффективности разработки, сопровождения, изготовления и эксплуатации программных изделий;
- автоматизации изготовления и хранения программной документации.

Сопровождение программы включает анализ функционирования, развитие и совершенствование. Правила и положения, установленные в стандартах ЕСПД, распространяются на программы и программную документацию для вычислительных машин, комплексов и систем независимо от их назначения и области применения. Разработка организационно-методической документации, определяющей и регламентирующей деятельность организаций по разработке, сопровождению и эксплуатации программ, должна проводиться на основе стандартов ЕСПД.

Одним из нормативных документов, входящих в состав ЕСПД, является стандарт ГОСТ 19.505-79 ЕСПД «Руководство оператора. Требования к содержанию и оформлению», определяющий требования к документу «Руководство оператора» [8].

Руководство оператора обеспечивает процедуру общения оператора с вычислительной системой в процессе выполнения программы.

Стандарт ГОСТ 19.505-79 ЕСПД «Руководство оператора. Требования к содержанию и оформлению» [8] является основным стандартом, определяющим

требования руководству оператора. Структура и оформление документа устанавливаются в соответствии с этим стандартом.

Таблицы, расчеты и другие документы, используемые при разработке, оформляются в соответствии со стандартом ГОСТ 2.111-2013 [9].

Перед разработкой руководства оператора был разработан шаблон руководства оператора.

При подготовке шаблона руководства оператора, термины и определения основных понятий устанавливает ГОСТ 19.781-90 «Обеспечение систем обработки информации. Термины и определения», дополнительно используются в стандарте ГОСТ 15.971-90 «Системы обработки информации. Термины и определения» [10].

Исходя из того описания содержания документа, в соответствии с ГОСТ 19.505-79 «Руководство оператора. Требования к содержанию и оформлению», содержит:

- назначение программы;
- условия выполнения программы;
- выполнение программы;
- сообщения оператору.

В разделе «Назначение программы» должны быть указаны сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

В разделе «Условия выполнения программы» должны быть указаны условия, необходимые для выполнения программы (минимальный и (или) максимальный состав аппаратных и программных средств и т.п.).

В разделе «Выполнение программы» должна быть указана последовательность действий оператора, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведено описание функций, формата и возможных вариантов команд, с помощью которых оператор осуществляет загрузки и управляет выполнением программы, а также ответы программы на эти команды.

В разделе «Сообщения оператору» должны быть приведены тексты сообщений, выдаваемых в ходе выполнения программы, описание их содержания и соответствующие действия оператора (действия оператора в случае сбоя, возможности повторного запуска программы и т.п.).

Все основные надписи руководства оператора выполнены в соответствии с ГОСТ 2.104-2006 «Единая система конструкторской документации. Основные надписи» [11].

В перечень терминов руководства оператора были включены термины: программа, ввод данных, вывод данных, программные и технические средства, данные, диалоговый режим, интерактивный режим. Данные термины составляют основную группу терминов.

При работе с программой оператор, может выполнять ввод-вывод данных. Поэтому в перечень терминов были включены термины «данные», «ввод данных», «вывод данных».

Условия выполнения программы, содержат сведения об необходимых программных и технических средствах, и их характеристиках, поэтому необходимо включить термин «программные и технические средства».

Взаимодействие оператора с программой может осуществляться в различных режимах. Термины, обозначающие режимы взаимодействия «диалоговый режим» и «интерактивный режим» были включены в перечень терминов.

Термин «программа» содержит в своем определении термин «система обработки информации», которое в свою очередь содержит термин «обработка информации». Оба термина отсутствуют в ГОСТ 19.781-90 [12], данные термины взяты из ГОСТ 15.971-90 «Системы обработки информации. Термины и определения». Эти термины так же включены в перечень дополнительных терминов.

Таким образом, перечень дополнительных терминов содержит следующие термины: «система обработки информации» «обработка информации».

В перечень основных включены термины, представленные ниже.

Программа – данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определенного алгоритма (ГОСТ 19.781-90) .

Диалоговый режим – режим взаимодействия человека с системой обработки информации, при котором человек и система обмениваются информацией в темпе, который соизмерим с темпом обработки информации человеком.

Интерактивный режим – режим взаимодействия процесса обработки информации системы обработки информации с человеком, выражающийся в разного рода воздействиях на этот процесс, предусмотренных механизмом управления конкретной системы и вызывающих ответную реакцию процесса.

Технические средства системы обработки информации – все оборудование, включая носителя данных, предназначенное для автоматизированной обработки информации (ГОСТ 15.971-90).

Данные – информация, представленная в виде, пригодном для обработки автоматическими средствами при возможном участии человека (ГОСТ 15.971-90).

Ввод данных – операция чтения данных с носителя, запись этих данных в память данных (ГОСТ 15.971-90).

Вывод данных – операция чтения данных в памяти данных и последующая их запись на носитель данных или отображение на экране (ГОСТ 15.971-90).

В перечень дополнительных включены термины, представленные ниже.

Обработка информации – систематическое выполнение операций над данными, представляющими предназначенную для обработки информацию (ГОСТ 15.971-90).

Система обработки информации – совокупность технических средств и программного обеспечения, а также методов обработки информации и действий персонала, обеспечивающая выполнение автоматизированной обработки информации (ГОСТ 15.971-90).

В результате был разработан шаблон руководства оператора, на основании которого было разработано руководство оператора комплекса мониторинга запускаемых процессов. Руководство оператора приведено в приложении К.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломного проекта были решены следующие задачи:

- произведен анализ операционной системы для работы, в которой предназначается комплекс мониторинга;
- произведена разработка модели угроз для разрабатываемого программного комплекса мониторинга;
- сформированы требования к комплексу мониторинга;
- произведено проектирование комплекса мониторинга;
- произведена программная реализация компонентов комплекса мониторинга;
- произведено тестирование комплекса мониторинга;
- написано руководство оператора для комплекса мониторинга.

В результате дипломного проектирования все пункты технического задания были выполнены, был разработан комплекс мониторинга процессов функционирования распределенной вычислительной системы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ключевые особенности Astra Linux Special Edition по реализации требований безопасности информации [Электронный ресурс]. – Режим доступа свободный: <http://www.astralinux.com/klyuchevye-osobennosti.html>;
2. Qt Application Development. Особенности библиотеки Qt [Электронный ресурс]. – Режим доступа свободный: <https://www.qt.io/ru/application-development>;
3. PostgreSQL [Электронный ресурс]. – Режим доступа свободный: <https://www.wikiwand.com/ru/PostgreSQL>
4. Мониторинг, анализ и управление ИБ [Электронный ресурс]. – Режим доступа свободный: www.itsec.ru/articles2/control/monitoring_analiz_upravl_ib;
5. Методически документ «Методика определения актуальных угроз безопасности персональных данных при их обработке в информационных системах персональных данных. ФСТЭК России, 2008 год;
6. ГОСТ Р 34.11-2012 «Информационная технология. Криптографическая защита информации. Функция хэширования»;
7. ГОСТ Р ИСО/МЭК 9126-93 «Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению»;
8. ГОСТ Р 19.505-79 «Руководство оператора»;
9. ГОСТ 19.781-90 «Обеспечение систем обработки информации. Термины и определения»;
10. ГОСТ 19.103-77 «Единая система программной документации. Обозначения программ и программных документов»;
11. ГОСТ 15.971-90 «Системы обработки информации. Термины и определения»;
12. ГОСТ 2.104-2006 «Единая система конструкторской документации. Основные надписи».

ПРИЛОЖЕНИЕ А

Текст файла mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QTreeWidgetItem>
#include <QFileDialog>
#include <QVector>
#include <QListWidget>

#include "workwithdatabase.h"
#include "smcore.h"

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();
    bool isAutorized() { return isUserAutorized;}
private:
    Ui::MainWindow *ui;
    WorkWithDataBase *dataBase;
    SMCore *core;
    // Авторизация
    bool isUserAutorized;

    // Поведение
    bool isValidUser(QString , QString);
    void initUserInfo();

private:
    // Меню работы пользователями
    void setListUser();
    void setListProgramForUser(QString ,QTreeWidgetItem*);
    void setListProcessForUser(QString ,QListWidget*);

    // имитация БД
    QMap<QString, UserInfo> userList;
    QMap<QString, FileInfo> programList;
    QVector<UserInfo> listUser;
    QVector<FileInfo> listAllProgramm;

    QString loginAdmin;
    QString passAdmin;
    QString nameAdmin;
```

```

QString passRoot;

void setListAllFile(QListWidget *List)
{
    List->clear();
    foreach (QString key, programList.keys())
    {
        List->addItem(key);
    }
}

void setListFile(QListWidget *List, QMap<QString, FileInfo> map)
{
    List->clear();
    foreach (QString key, map.keys())
    {
        List->addItem(key);
    }
}

void setListUser(QListWidget *List)
{
    List->clear();
    foreach (QString key, userList.keys())
    {
        List->addItem(key);
    }
}

private slots:
    /// \brief Слоты для регистрации
    /// \brief Слот для обработки нажатия кнопки "Начать регистрацию"
    void slotStartRegistration();
    /// \brief Слот для обработки нажатия кнопки "Далее" при создании учетной
записи. Этап 1
    void slotRegistrationPoin1();
    /// \brief Слот для обработки нажатия кнопки "Далее" при создании учетной
записи. Этап 2
    void slotRegistrationPoin2();
    /// \brief Слот для обработки нажатия кнопки "Далее" при создании учетной
записи. Этап 3
    void slotRegistrationPoin3();
    /// \brief Слот для обработки нажатия кнопки "Завершить" на последнем этапе
регистрации
    void slotEndRegistration();

    /// \brief Слоты для авторизации
    /// \brief Слот для обработки нажатия на кнопку "Авторизироваться"
    void slotAutorizationClick();

    /// \brief Слоты для выбора меню для работы
    /// \brief Выбрано меню работы с журналом
    void slotWorkWithJournalChange();
    /// \brief Выбрано меню работы с программами
    void slotWorkWithProgramChange();
    /// \brief Выбрано меню работы с пользователями
    void slotWorkWithUserChange();
    /// \brief Выбрано меню работы с настройками
    void slotWorkWithSettings();
    /// \brief Выбрано меню работы с аккаунтом администратора
    void slotWorkWithAccount();

    /// \brief Слоты для работы с пользователем

```

```

    /// \brief Выбран пункт создать нового пользователя
    void slotCreateNewUserClick();
    /// \brief Меню создания нового пользователя
    /// \brief Слот для обработки нажатия кнопки "Завершить создание"
    /// в меню создания нового пользователя
    void slotEndCreateNewUserClick();
    /// \brief Выбран пункт редактирования пользователя
    void slotRedactUserClick();
    /// \brief Слот для обработки нажатия кнопки "Завершить редактирование" в
меню редактирования пользователя
    void slotEndRedactUserClick();
    /// \brief Нажата клавиша добавить в меню создания нового пользователя
    void slotAddProgramInUserList();
    /// \brief Нажата клавиша убрать в меню создания нового пользователя
    void slotTakeProgramFromUserList();
    /// \brief Отмена создания или редактирования пользователя
    void slotCancelWorkUser();
    /// \brief Удаление пользователя
    void slotDelUserClick();


    /// \brief Слоты для работы с аккаунтом
    /// \brief Нажата клавиша редактировать учетную запись
    void slotEditAccount();
    /// \brief Нажата клавиша изменить в поле логин
    void slotEditLoginInfo();
    /// \brief Нажата клавиша изменить в поле пароль
    void slotEditPasswordInfo();
    /// \brief Нажата клавиша завершить редактирование
    void slotEndEditAccount();
    /// \brief Нажата клавиша изменить в поле логин в БД
    void slotEditDBLoginInfo();
    /// \brief Нажата клавиша изменить в поле пароль в БД
    void slotEditDBPasswordInfo();
    /// \brief Нажата клавиша завершить редактирование в БД
    void slotEndEditDBInfo();
    /// \brief Нажата клавиша редактировать информацию о базе данных
    void slotEditDBInfo();


    /// \brief Слоты для работы с файлами
    /// \brief Нажата клавиша добавить файл
    void slotAddFileClick();
    /// \brief Нажата клавиша удалить файл
    void slotDelFileClick();
    /// \brief Нажата клавиша сохранить изменения
    void saveFileList();
};

#endif // MAINWINDOW_H

```

ПРИЛОЖЕНИЕ Б

Текст файла smcore.h

```
#ifndef SMCORE_H
#define SMCORE_H
//Qt
#include <QVector>
#include <QFile>
#include <QByteArray>
#include <QString>
#include <QCryptographicHash>
#include <QRegExp>
//Syst
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <pwd.h>
#include <crypt.h>

//Project
#include "workwithdatabase.h"

class SMCore
{
public:
    SMCore();

    WorkWithDataBase *getAccessInDataBase() { return DataBase; }

    void initSM(QString p_nameAdmin, QString p_passwordAdmin ,
                QString p_DataBaseName, QString p_DataBasePassword,
                QString p_DataBaseUser);

    bool verPassword(QString name, QString password);

    bool isAdminInit() { return this->adminIsCreate; }

    QString createPassHash(QString );
    QString createCSHash(QString );
    QString criptData(QString, QString key);
    QString readDataFileInStr(QString nameFile);

private:
    QString criptInitFile(QString );
    QString decryptInitFile(QString );

private:
    bool adminIsCreate;
    QString keyForInitFile;
    QString DataBaseName;
    QString DataBasePassword;
    QString DataBaseAdminName;
    WorkWithDataBase *DataBase;
};

#endif // SMCORE_H
```

ПРИЛОЖЕНИЕ В

Текст файла workwithdatabase.h

```
#ifndef WORKWITHDATABASE_H
#define WORKWITHDATABASE_H
// Qt moduls for work
#include <QSqlDatabase>
#include <QSqlQuery>
#include <QVector>
//Syst
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <pwd.h>

// For debbuging
#include <QDebug>
#include <QStringList>
#include <QSqlError>

class WorkWithDataBase : ConnectAndWorkWithDataBase
{
    /*****
    /*****      МЕТОДЫ      *****/
    /*****/

public:
    WorkWithDataBase();

    /*****      ИНТЕРФЕЙС      *****/
    int addProgram(QString nameProgram, QString path, QVector<FileInfo>
programFile);
    void addUser(QString newUserLogin, QString newUserPassword, QString
newUserName/*, QLinkedList<FileInfo>*/);

    void getProgramInfo(int programId);
    void getUserInfo(QString nameUser);

    void updateProgramFile(int programId, QVector<FileInfo> programFile);

    void getAdminData(QString &login, QString &hashPass);

    QVector<UserInfo> getAllUser();

    QVector<FileInfo> getAllProgram();

    void setAdminAccount(QString login, QString pass, QString name, QString
passRoot);

    void setAdminBDInfo(QString p_login, QString p_pass);
    bool connectDataBase(QString namedataBase, QString passwordDatabase, QString
nameAdmin);
    bool isAdminInit();
private:
    bool createDataBase(); // Если база данных еще не создана
    int getLastProgramId();
    int getLastFileId();

    /*****/
};
```

```

        /*****                АТРИБУТЫ                *****/
        /*****/

public:

private:
    QSqlDatabase dbForSMonitor;    // DataBaseForSystemMonitor, переменная для
    работы с базой данных Postgres
    QVector<int > programIdBase; // Хранилище
};

#endif // WORKWITHDATABASE_H

```

ПРИЛОЖЕНИЕ Г

Текст файла abstractdaemon.h

```
#ifndef ABSTRACTDAEMON_H
#define ABSTRACTDAEMON_H
// Project
#include "smtypes.h"
#include "servicemodule.h"

//! \class AbstractDaemon
//! \brief Базовый класс для всех демонов
class AbstractDaemon
{
public:
    AbstractDaemon();
    virtual ~AbstractDaemon() {}

    //! Этот метод осуществляет основную работу (назначение) демона
    //! Каждый демон должен реализовать его в соответствии с назначением
    virtual int mainLoop() = 0;

    //! Метод запуска работы демона
    void exec() { this->createDaemon(); }

private:
    //! Создает и запускает демона
    void createDaemon();
    //! Следит за работой демона
    void daemonMonitor();
    //! Обработчик сигналов для борьбы с процессами-зомби
    void signalHandler(int signum) { waitpid(0, 0, WNOHANG); }
};

//! \class AbstractNotifyingDaemon
//! \brief Базовый класс для демонов которые должны оповещать о чем то
class AbstractNotifyingDaemon : public AbstractDaemon
{
public:
    AbstractNotifyingDaemon();
    ~AbstractNotifyingDaemon() {}
    void commit();
    void commitInFile();
    void setDblCommit(bool enable, int numberRegProcess);
    void setErrorCommit(bool enable) { if(enable) flagCommitInFile =
true; }

private:
    void openFileForCommit();

private:
    bool flagCommitInFile;
};

//! \class DaemonProcessMonitor
```

```

//! \brief Класс осуществляющий мониторинг процессов
class DaemonProcessMonitor : public AbstractNotifyingDaemon, public
ClientSocketForDaemon
{
public:
    DaemonProcessMonitor();
    ~DaemonProcessMonitor(){}
    int mainLoop();
    void setListPermittingProcess(QStringList );
    void setListForbiddenProcess(QStringList );
    void setUsername(QString );
    void setTimeRepeat(int );

private:
    void readAllProcess();
    void searchNewProcess();
    void pidMonitoring();
    void closeProcess(int pidForClose);
    void parsingProcessName();

private:
    QString userName;
    QStringList PermittingProcess;
    QStringList ForbiddenProcess;
    QStringList PermittingDir;
    QStringList SystemProcess;

};

//! \class DaemonFileMonitor
//! \brief Класс осуществляющий проверку контроля целостности
class DaemonFileMonitor : public AbstractNotifyingDaemon, public
ClientSocketForDaemon
{
public:
    DaemonFileMonitor();
    ~DaemonFileMonitor(){}
    int mainLoop();

private:

};

//! \class DaemonAgentMonitoring
//! \brief Класс порождающий демонов для мониторинга DaemonFileMonitor,
DaemonProcessMonitor
//! и получающий от них результаты мониторинга, для обработки и записи этих
результатов в журнал
class DaemonAgentMonitoring : public AbstractDaemon, public ServerSocketForDaemon
{
public:
    DaemonAgentMonitoring();
    ~DaemonAgentMonitoring(){}

    bool isConnectDataBase(){ return this->dataBaseConnected;}
    int mainLoop();

private:
    //! инициализация при запуске
    void connectDataBase();
    QString detectedUser();

```

```

void readListPermitItem(QString pUserName);
void readSettings();

// Запуск демонов
void createFileMonitorDaemon();
void createProcessMonitorDaemon();

// Обработка и анализ
void parsingMessage(QString );

// Журналирование событий
void writeInFileJournal(QString message);
void writeInDBJournal(QString message);

// Дублирование информации
void writeTempSettingsAndData(); //!< Дублирует всю информацию для
мониторанга в Файле.
private:
    QStringList PermittingProcess;
    QStringList ForbiddenProcess;
    QStringList SystemProcess;
    QStringList PermittingDir;
    bool dataBaseConnected;
    QString userName;
};

#endif // ABSTRACTDAEMON_H

```

ПРИЛОЖЕНИЕ Д

Текст файла serversocketfordaemon.h

```
//! \class ServerSocketForDaemon
//! \brief Реализация сервера для работы демона-агента
class ServerSocketForDaemon
{
private:
    QTcpServer* m_ptcpServer;
    quint16      m_nNextBlockSize;

private:
    void sendToClient(QTcpSocket* pSocket, const QString& str);
    void getInfo();

public:
    ServerSocketForDaemon(int nPort);

public slots:
    virtual void slotNewConnection();
    void slotReadClient();
};
```

ПРИЛОЖЕНИЕ Е

Текст файла clientsocketfordaemon.h

```
class ClientSocketForDaemon
{
private:
    QTcpSocket* m_pTcpSocket;
    quint16      m_nNextBlockSize;

public:
    ClientSocketForDaemon(const QString& strHost, int nPort);
    void sendInfo(QString info){}

private slots:
    void slotReadyRead();
    void slotError(QAbstractSocket::SocketError);
    void slotSendToServer();
    void slotConnected();
};
```

ПРИЛОЖЕНИЕ Ж

Текст файла workwithproc.h

```
class WorkWithProc
{
    WorkWithProc();
    ~WorkWithProc();

    void readDir();
    QString getDirInfo();
    void readSysInfo();
    QString getSysInfo();

private:
    QString systinfo;
    QString dirinfo;
    void parsingDirInfo();
    vector<float> readCpuStats();
    int getCpuLoad(double dt);
    void update();
};
```

ПРИЛОЖЕНИЕ 3

Текст файла cryptmodule.h

```
class CriptModule
{
public:
    CriptModule();
    ~CriptModule();

    QByteArray createHash(QByteArray , long int *);
    QByteArray cryptData(QByteArray , long int *);

private:
    void cryptDES(QByteArray);
    void cryptMod(QByteArray);
};
```

ПРИЛОЖЕНИЕ И

Текст файла connectandworkwithdatabase.h

```
class ConnectAndWorkWithDataBase
{
public:
    ConnectAndWorkWithDataBase();
    ~ConnectAndWorkWithDataBase();

    bool connectDataBase(QString namedataBase, QString passwordDatabase, QString
nameAdmin);
    bool removeDataBase(QString namedataBase);

    QStringList getAllRow(QString tableName);
    QString getLastRow(QString tableName);

private:
    QSqlDatabase dataBase;
};
```

ПРИЛОЖЕНИЕ К

(обязательное)

Протокол проверки ВКР на оригинальность

Протокол

проверки на оригинальность в системе «Антиплагиат.ВУЗ» выпускной квалификационной работы на тему «Программный комплекс мониторинга процессов функционирования распределенной вычислительной системы» студента шестого курса А.В.Кузнецова специальности 10.05.02 «Информационная безопасность телекоммуникационных систем».

Руководитель ВКР Козлов Андрей Юрьевич.

Информация о документе:

Имя исходного файла: Отчет по диплому Кузнецов А.В..doc
Имя компании: Пензенский гос. Университет
Тип документа: Прочее
Имя документа: Группа 11ПТ1 ВКР Кузнецов АВ
Дата проверки: 09.01.2017 17:42
Модули поиска: Интернет (Антиплагиат), Университетская библиотека онлайн, Модуль поиска ЭБС БиблиоРоссика, Модуль поиска ЭБС "Лань", Диссертации и авторефераты РГБ, Пензенский гос. Университет, Цитирования, Кольцо вузов

Текстовые статистики:

Индекс читаемости: сложный
Неизвестные слова: в пределах нормы
Макс. длина слова: в пределах нормы
Большие слова: в пределах нормы

Оригинальные блоки: 80,34%

Заимствованные блоки: 19,38%

Заимствование из "белых" источников: 0,28%

Итоговая оценка оригинальности: **80,62%**

09.01.2017 г

_____ А.Ю. Козлов

ПРИЛОЖЕНИЕ Л

Руководство оператора

<АО НПП "РУБИН">

УТВЕРЖДЕН

7241000. 425760.001.34.X–X.M.

**<КОМПЛЕКС МОНИТОРИНГА ЗАПУСКАЕМЫХ
ПРОЦЕССОВ>**

РУКОВОДСТВО ОПЕРАТОРА

7241000. 425760.001.34.X–X.M.

На 13 листах

Ине. № подл.	Подпись и дата	Взам. инв. №	Ине. № дубл.	Подпись и дата

Пенза, 2017

СОДЕРЖАНИЕ

Термины и определения.....2

1 Назначение программы.....3

2 Условия выполнения программы4

3 Выполнение программы5

4 Сообщения оператору.....6

Первое применение

Справ.№

Подпись и дата

Инв. № дубл.

Взам. инв. №

Подпись и дата

Инв. № подл.

Изм	Лист	№ документа	Подпись	Дата
Разработ.	Кузнецов А.В.			
Проверил				
Т.контр.				
Н.контр.				
Утвердил				

7241000. 425760.001.34.X–X.M.

Наименование АС
Руководство оператора

Литера	Лист	Листов
М	1	98

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Программа – данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определенного алгоритма (ГОСТ 19.781-90).

Диалоговый режим – Режим взаимодействия человека с системой обработки информации, при котором человек и система обмениваются информацией в темпе, который соизмерим с темпом обработки информации человеком.

Интерактивный режим – режим взаимодействия процесса обработки информации системы обработки информации с человеком, выражающийся в разного рода воздействиях на этот процесс, предусмотренных механизмом управления конкретной системы и вызывающих ответную реакцию процесса.

Технические средства системы обработки информации – все оборудование, включая носителя данных, предназначенное для автоматизированной обработки информации (ГОСТ 15.971-90).

Данные – информация, представленная в виде, пригодном для обработки автоматическими средствами при возможном участии человека (ГОСТ 15.971-90).

Ввод данных – операция чтения данных с носителя, запись этих данных в память данных (ГОСТ 15.971-90).

Вывод данных – операция чтения данных в памяти данных и последующая их запись на носитель данных или отображение на экране (ГОСТ 15.971-90).

Обработка информации – систематическое выполнение операций над данными, представляющими предназначенную для обработки информацию (ГОСТ 15.971-90).

Система обработки информации – совокупность технических средств и программного обеспечения, а также методов обработки информации и действий персонала, обеспечивающая выполнение автоматизированной обработки информации (ГОСТ 15.971-90).

<div>Изм.</div> <div>Лист</div> <div>№ документа</div> <div>Подпись</div> <div>Дата</div>	Инва. № подл.	Подпись и дата	Взам. инв. №	Инва. № дубл.	Подпись и дата
	<p>Данные – информация, представленная в виде, пригодном для обработки автоматическими средствами при возможном участии человека (ГОСТ 15.971-90).</p>				
	<p>Ввод данных – операция чтения данных с носителя, запись этих данных в память данных (ГОСТ 15.971-90).</p>				
	<p>Вывод данных – операция чтения данных в памяти данных и последующая их запись на носитель данных или отображение на экране (ГОСТ 15.971-90).</p>				
	<p>Обработка информации – систематическое выполнение операций над данными, представляющими предназначенную для обработки информацию (ГОСТ 15.971-90).</p>				
<p>Система обработки информации – совокупность технических средств и программного обеспечения, а также методов обработки информации и действий персонала, обеспечивающая выполнение автоматизированной обработки информации (ГОСТ 15.971-90).</p>					
7241000. 425760.001.34.X–X.M.					Лист
					87

A.1 НАЗНАЧЕНИЕ ПРОГРАММЫ

A1.1. Функциональное назначение программы

Основной функцией программного комплекса мониторинга является определение запуска процессов в операционной системе Astra Linux, обработка и протоколирование всех событий запуска процессов в системе, а также контроль целостности файлов программного обеспечения системы, указанных в программе.

A1.2. Эксплуатационное назначение программы

Основное назначение программного комплекса мониторинга – отслеживать и протоколировать события запуска процессов в операционной системе, а также события нарушения целостности файлов программного обеспечения, чем повысить безопасность операционной системы и упростить проведение расследований инцидентов.

А1.3. Состав функций

Функция определения запуска процессов в операционной системе и идентификация запущенного процесса заключается в постоянном отслеживании занятых PID идентификаторов в системе, и в случае появления нового идентификатора, определение пользователя в сеанс которого запущен процесс, название файла программы, который породил процесс.

Функция обработки и протоколирования событий запуска процессов в операционной системе заключается в проверке прав доступа текущего пользователя на запуск данного процесса и записи результатов обработки в журнал мониторинга.

Функция контроля целостности заключается в нахождении контрольной суммы для файлов программного обеспечения и периодической проверки соответствия контрольных сумм находимых для файла на соответствие эталонной.

[illegible]

А.2 УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

А2.1 Минимальный состав аппаратных средств

Минимальный состав аппаратных средств:

- ОЗУ более 128 Мбайт;
- 128 Мбайт видеопамяти и выше;
- наличие свободного места на жестком диске 500 Мбайт и выше.

А2.2 Минимальный состав программных средств

Системные программные средств, используемые программным комплексом мониторинга должны быть представлены операционной системой Astra Linux CE или Astra Linux SE.

Также для функционирования программного комплекса необходимо наличие предустановленной СУРБД PostgreSQL.

Ине. № подл.	Подпись и дата	Взам. инв. №	Ине. № дубл.	Подпись и дата

А.3 ВЫПОЛНЕНИЕ ПРОГРАММЫ

А.3.1 Загрузка и запуск программы.

Загрузка и запуск программного комплекса мониторинга осуществляется способами, деталильные сведения о которых изложены в «Руководстве пользователя операционной системы».

А.3.2 Выполнение программы

А.3.2.1 Функция определения запуска процессов в операционной системе

Программный комплекс мониторинга проверяет систему на наличие новых выполняемых процессов. Проверка происходит в автоматическом режиме. Все новые запущенные процессы сверяют со списком процессов к которым у пользователя есть доступ.

А.3.2.2 Функция обработки и протоколирования событий запуска процессов

После обнаружения запуска нового процесса в операционной системе, происходит обработка информации о процессе и запустившем его файле, на основании информации о процессе и текущем пользователе, в ходе обработки, выявляются дальнейшие действия по отношению к процессу. Если процесс запрещен на запуск пользователю, то он завершается. Результаты обработки записываются в журнал мониторинга.

А.3.2.3 Функция контроля целостности

В процессе функционирования комплекс проверяет целостность файлов, заданных специальным списком. Целостность проверяется с определенной периодичностью. Если целостность была нарушена, то это фиксируется в журнале мониторинга.

А.3.3. Завершение работы программы

Программный комплекс мониторинга завершает свою работу вместе с операционной системой.

Также завершить работу комплекса может администратор в режиме «Администрирование», с помощью графического интерфейса.

Ине. № подл.	Подпись и дата	Взам. инв. №	Ине. № дубл.	Подпись и дата

А.4 СООБЩЕНИЯ ОПЕРАТОРУ

А.4.1 Сообщения выводимые при регистрации

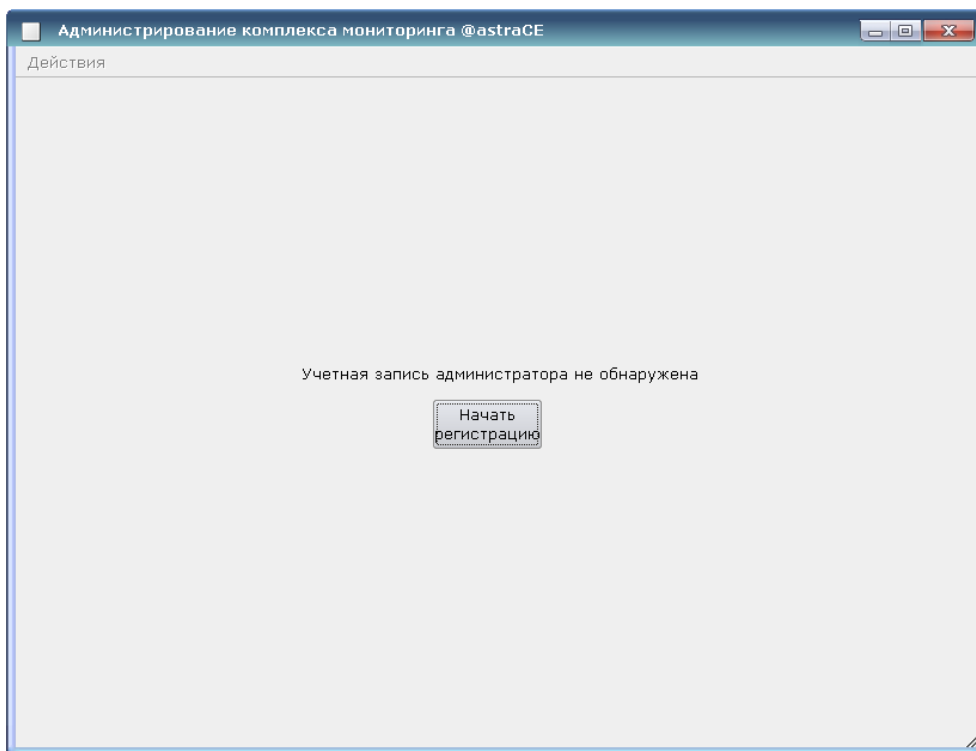


Рисунок 1 – Графическое окно начала регистрации

Рисунок 2 – Графическое окно с формой для регистрации, этап 1

Име. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

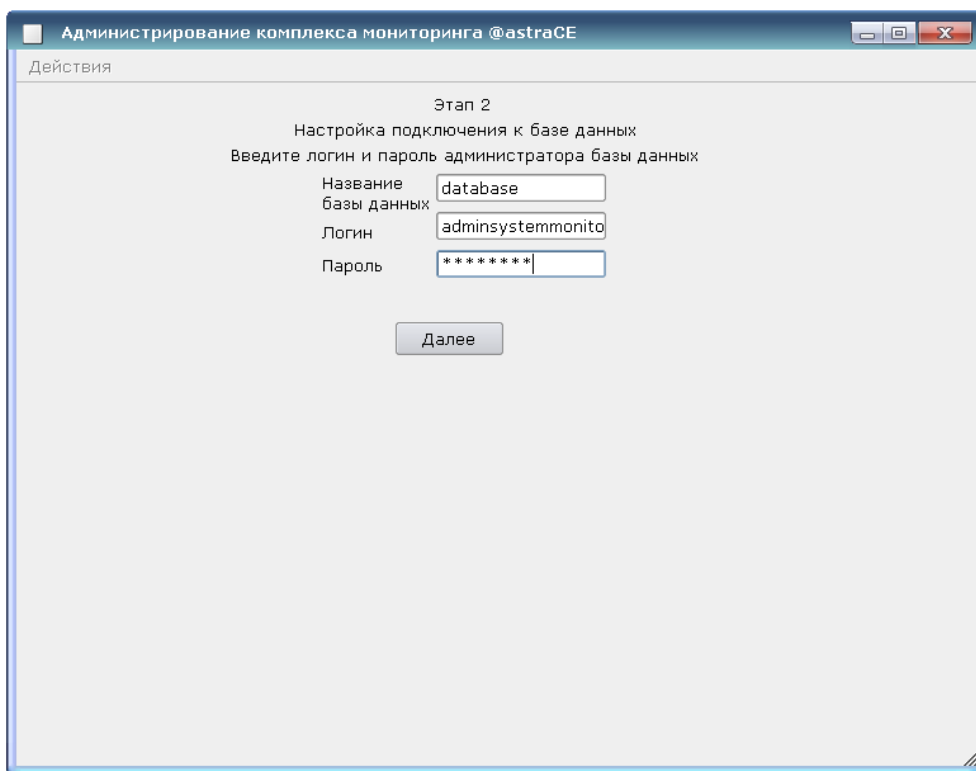


Рисунок 3 - Графическое окно с формой для регистрации, этап 2

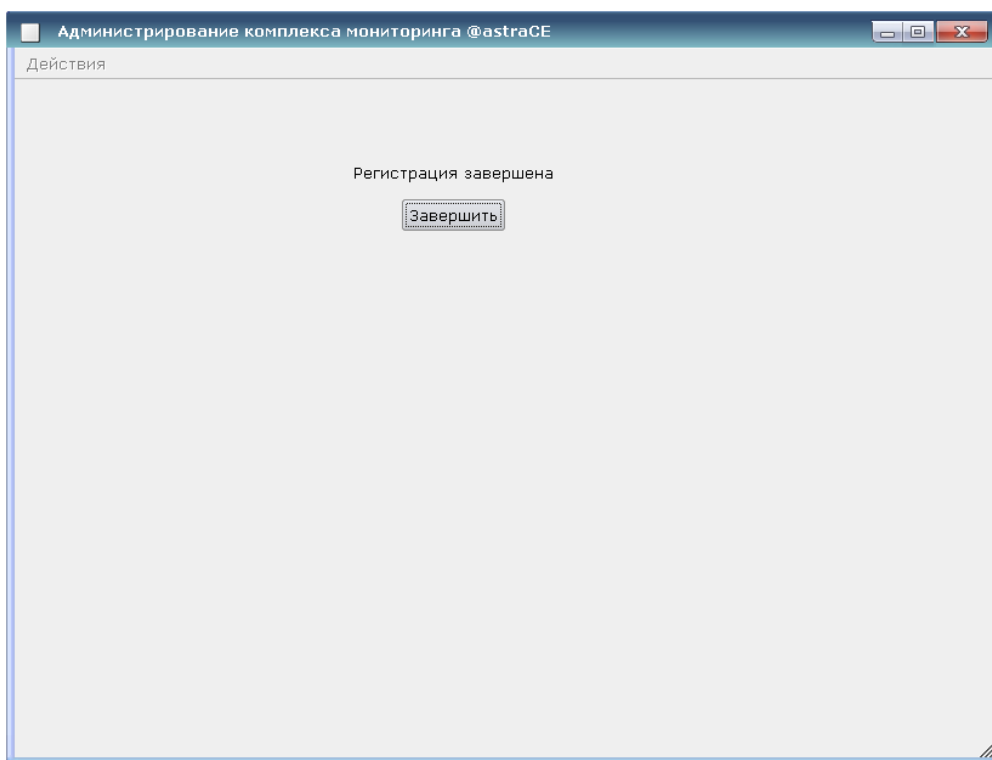


Рисунок 4 – Графическое окно завершения регистрации

Ине. № подл.	Подпись и дата	Взам. инв. №	Ине. № дубл.	Подпись и дата

А.4.2 Сообщения выводимые при авторизации

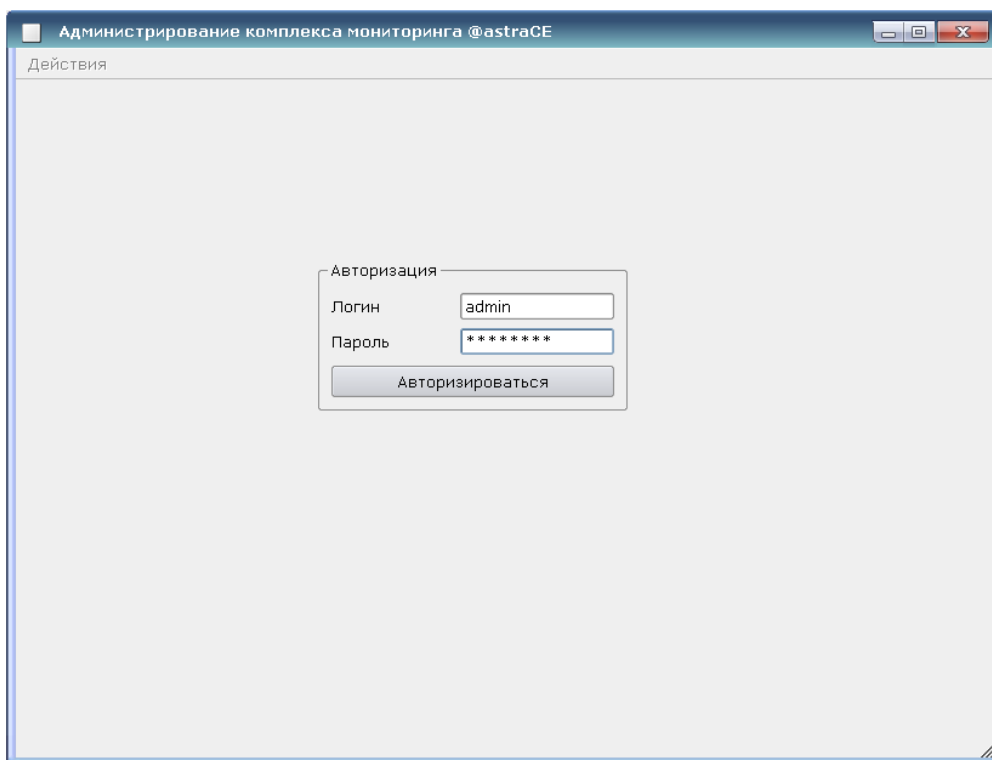


Рисунок 5 – Окно для введения данных для авторизации.

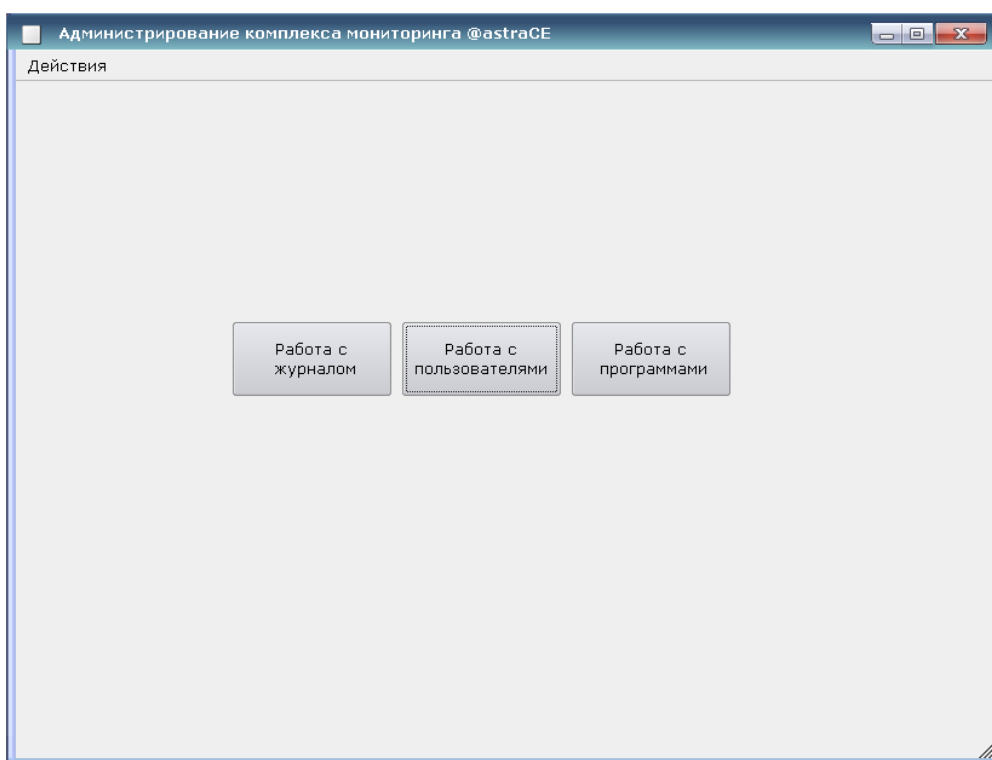


Рисунок 6 – Графическое окно с действиями для пользователя после прохождения авторизации

Ине. № подл.	Подпись и дата	Взам. инв. №	Ине. № дубл.	Подпись и дата

А.4.3 Сообщения выводимые работе с файлами программ

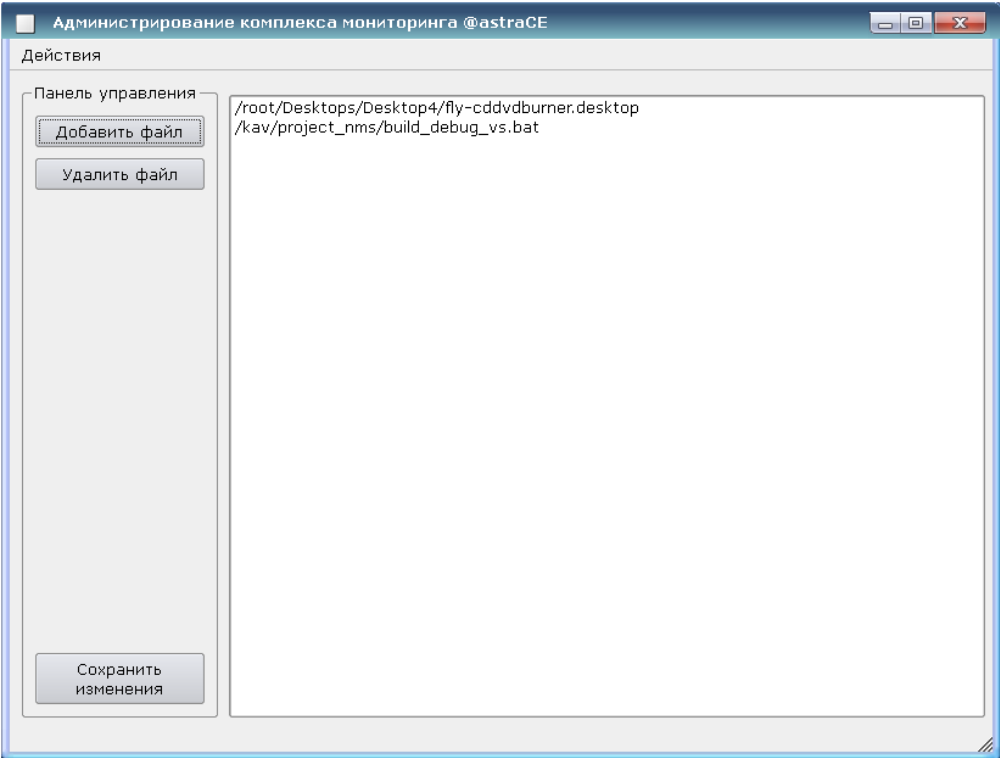


Рисунок 7 – Графическое окно для работы с файлами

А.4.4 Сообщения выводимые при работе с пользователями

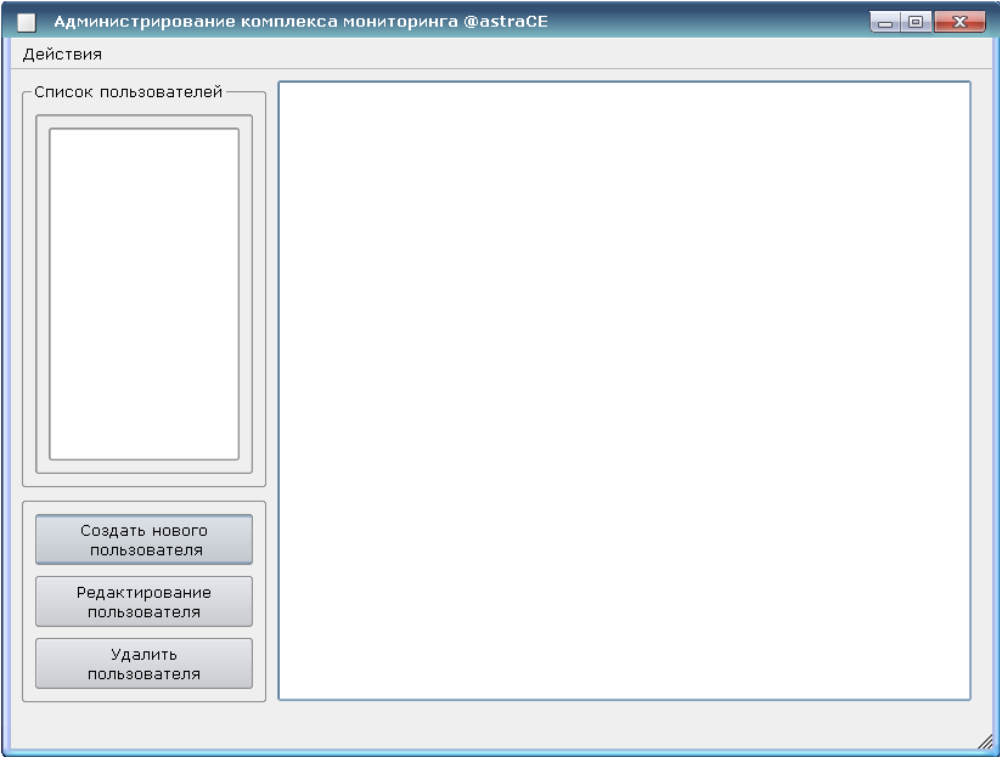


Рисунок 8 – Графическое окно для работы с пользователями

Ине. № подл.	Подпись и дата	Взам. инв. №	Ине. № дубл.	Подпись и дата

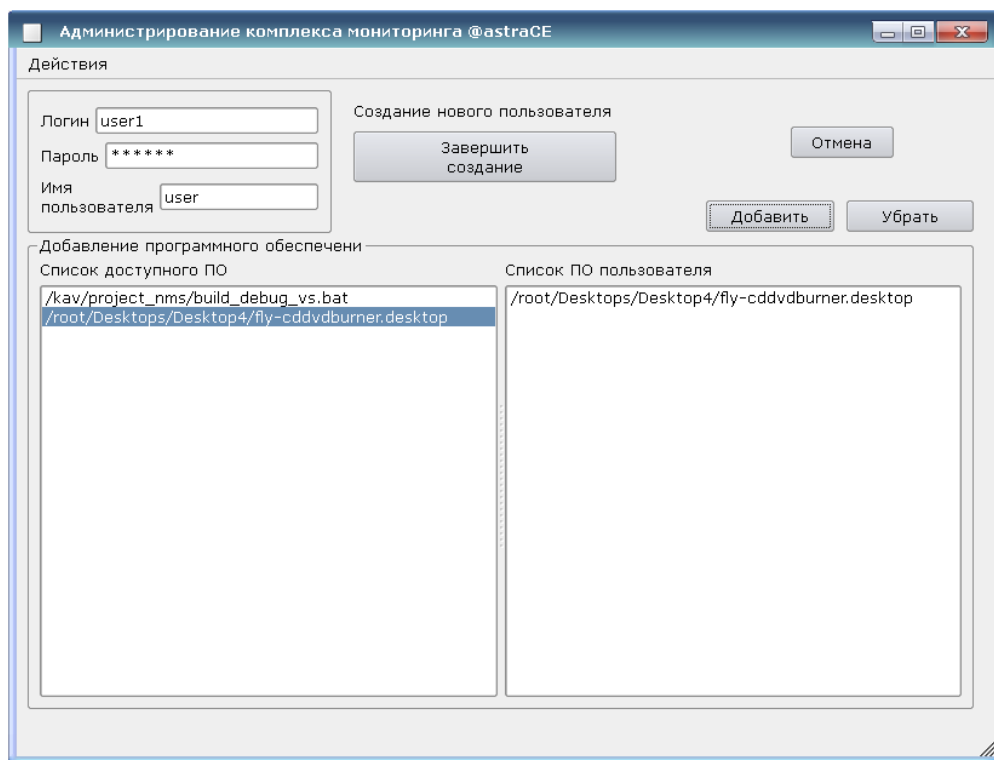


Рисунок 9 – Графическое окно для создания нового пользователя

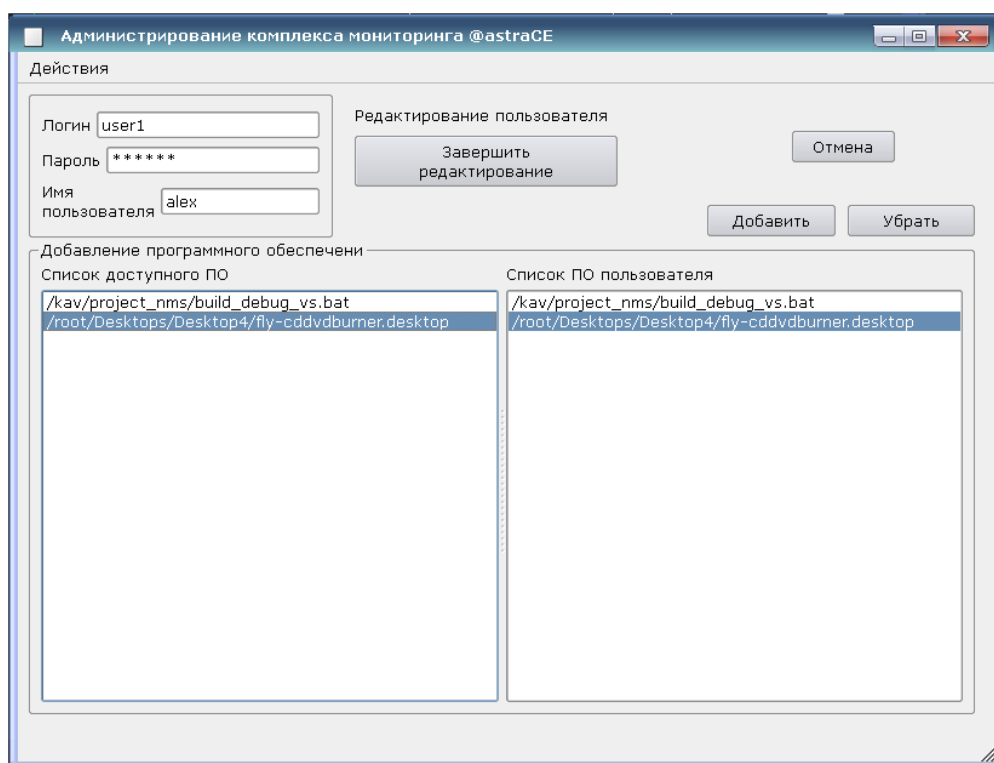


Рисунок 10 – Графическое окно для редактирования пользователя

Ине. № подл.	Подпись и дата	Взам. инв. №	Ине. № дубл.	Подпись и дата

А.4.5 Сообщения выводимые при работе с учетной записью администратора

Администрирование комплекса мониторинга @astraCE

Действия

Учетная запись

Учетная запись

Логин: Admin

Имя: alex

Редактировать

Параметры БД

Имя базы данных: AdminDB

Логин базы данных: DataBase

Редактировать

Рисунок 11 – Информация об учетной записи администратора

Администрирование комплекса мониторинга @astraCE

Действия

Редактирование учетной записи

Смена учетных данных

Логин: admin

Имя: alex

Введите пароль

Изменить

Смена пароля

Введите новый пароль

Повторите новый пароль

Введите старый пароль

Изменить

Завершить редактирование

Рисунок 12 – Редактирование учетной записи администратора системы мониторинга

Ине. № подл.	Подпись и дата
Взам. инв. №	Ине. № дубл.
Подпись и дата	

Администрирование комплекса мониторинга @astraCE

Действия

Редактирование данных для подключения к базе данных

Изменение данных для БД

Имя БД:

Логин:

Изменение пароля

Новый пароль для БД:

Повторите пароль для БД:

Введите пароль от учетной записи:

Рисунок 13 – Редактирование данных, для подключения к базе данных

А.4.6 Сообщения выводимые при настройке

Администрирование комплекса мониторинга @astraCE

Действия

Настройки

Журнал

☐ Шифрование журнала

Использовать шифрование:

Процессы

☐ Закрывать запрещенные для запуска пользователю программы

☐ Фиксировать отсутствующие в списке пользователя программы как запрещенные

Мониторинг процессов

☐ Дублировать журналируемые процессы в файл

Через какое число записей дублировать:

Контроль целостности

*Контроль целостности проводится при запуске комплекса и при запуске операционной системы.

☐ Включить периодический мониторинг

Интервал мониторинга:

Используемый алгоритм для подсчета контрольной суммы

Рисунок 14 – Графическое окно с настройками комплекса

Ине. № подл.	Подпись и дата	Взам. инв. №	Ине. № дубл.	Подпись и дата

А.4.7 Сообщения выводимые при работе с журналом

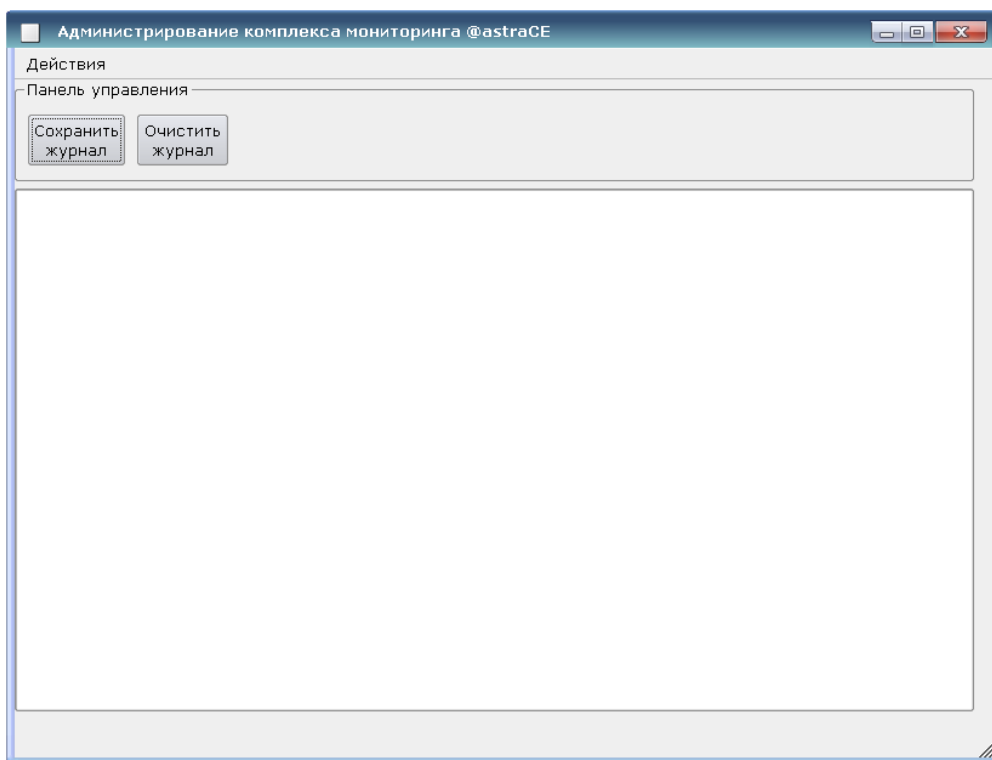


Рисунок 15 – Графическое окно для работы с журналом

Ине. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата