Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук

Кафедра «Защита информации»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой,
к.т.н., доцент
_____ А.Н. Соколов
_____ 2017 г.

Контроль системы разграничения доступа в средствах защиты информации от несанкционированного доступа с использованием поведенческой модели пользователя

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Автор проекта,
студент группы КЭ- 431
_____ Предеин, А. С.
_____ 2017 г.

Нормоконтролер,
к.т.н., доцент
_____ В.П. Мартынов
_____ 2017 г.

Челябинск 2017

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В наше время компьютеры используются практически во всех сферах жизни. И часто обрабатываемая ими информация обладает определенной ценностью. А ценность в любом случае должна находится под защитой. Из этого вытекает вопрос, как операционная система компьютера будет определять кому предоставить доступ к запрашиваемой информации, а кому нет? Ведь за клавиатуру может сесть абсолютно любой человек, и этим человеком с определенной долей вероятности может оказаться злоумышленник.

Для разрешения этого вопроса в операционные системы были введен механизм произвольного управления доступом или дискреционный доступ. Наиболее подробно этот механизм был описан в стандарте Министерства обороны США «Trusted Computer System Evaluation Criteria» [1] сокращенно TCSEC в переводе на русский "Критерии оценки безопасности компьютерных систем" принятом в 1983 году. Этот стандарт также известен как «Оранжевая книга». Согласно TCSEC дискреционное управление доступом - это метод ограничения доступа к объектам, основанный на учете личности субъекта или группы, в которую субъект входит. Дискреционность управления состоит в том, что некоторое лицо (обычно владелец объекта) может по своему усмотрению давать другим субъектам или отбирать у них права доступа к объекту. Текущее состояние прав доступа при дискреционном управлении описывается матрицей, в строках которой перечислены субъекты, а в столбцах - объекты. В клетках, расположенных на пересечении строк и столбцов, записываются способы доступа, допустимые для субъекта по отношению к объекту, например: чтение, запись, выполнение.

Большинство операционных систем реализуют именно произвольное управление доступом. Главное его достоинство - гибкость, главные недостатки - рассредоточенность управления и сложность централизованного контроля, а также оторванность прав доступа от данных, что позволяет копировать конфиденциальную информацию в общедоступные файлы. От этих недостатков избавлен механизм принудительного управления доступом или мандатный доступ.

Этот механизм так же описан в TCSEC: мандатное управление доступом основано на сопоставлении меток безопасности субъекта и объекта. Субъект может читать информацию из объекта, если уровень конфиденциальности субъекта не ниже, чем у объекта, а все категории, перечисленные в метке безопасности объекта, присутствуют в метке субъекта. В таком случае говорят, что метка субъекта доминирует над меткой объекта. Смысл сформулированного правила понятен - читать можно только то, что положено. Субъект может записывать информацию в объект, если метка безопасности объекта доминирует над меткой субъекта. В частности, "конфиденциальный" субъект может писать в строго конфиденциальные файлы, но не может - в не конфиденциальные. Ни при каких операциях уровень конфиденциальности информации не должен понижаться, хотя обратный процесс вполне возможен. Посторонний человек может случайно узнать конфиденциальные сведения и сообщить их куда следует, однако лицо, допущенное к работе с конфиденциальными документами, несет полную ответственность за разглашение этих данных. Описанный способ управления доступом называется принудительным, поскольку он не зависит от воли субъектов.

После того как зафиксированы метки безопасности субъектов и объектов, оказываются зафиксированными и права доступа. В терминах принудительного управления нельзя выразить предложение "разрешить доступ к объекту X еще и для пользователя Y". Конечно, можно изменить уровень конфиденциальности пользователя Y, но тогда он, скорее всего, получит доступ ко многим дополнительным объектам, а не только к X.

Рассмотрим подробнее метки безопасности. Метки предусмотрены для субъектов (степень благонадежности) и объектов (степень конфиденциальности информации). Метки безопасности содержат данные об уровне конфиденциальности и категории, к которой относятся данные. Согласно «Оранжевой книге», метки безопасности должны состоять из двух частей — уровня конфиденциальности и списка категорий. Уровни конфиденциальности, поддерживаемые системой, образуют упорядоченное множество, которое может выглядеть, например, так:

- не конфиденциально;
- почти конфиденциально;
- конфиденциально;
- супер-мега конфиденциально.

Для разных систем набор уровней конфиденциальности может различаться. Категории образуют неупорядоченный набор. Их назначение — описать предметную область, к которой относятся данные. В военном окружении каждая категория может соответствовать, например, определенному виду вооружений. Механизм категорий позволяет разделить информацию по отсекам, что способствует лучшей защищенности. Субъект не может получить доступ к «чужим» категориям, даже если его уровень благонадежности «строго конфиденциально». Специалист по танкам не узнает тактико-технические данные самолетов.

Главная проблема, которую необходимо решать в связи с метками, это обеспечение их целостности. Во-первых, не должно быть непомеченных субъектов и объектов, иначе в меточной безопасности появятся легко используемые бреши. Во-вторых, при любых операциях с данными метки должны оставаться правильными. В особенности это относится к экспорту и импорту данных. Например, печатный документ должен открываться заголовком, содержащим текстовое и/или графическое представление метки безопасности. Аналогично при передаче файла по каналу связи должна передаваться и ассоциированная с ним метка, причем в таком виде, чтобы удаленная система могла её разобрать, несмотря на возможные различия в уровнях конфиденциальности и наборе категорий. Одним из средств обеспечения целостности меток безопасности является разделение устройств на многоуровневые и одноуровневые. На многоуровневых устройствах может храниться информация разного уровня конфиденциальности (точнее, лежащая в определенном диапазоне уровней). Одноуровневое устройство можно рассматривать как вырожденный случай многоуровневого, когда допустимый диапазон состоит из одного уровня. Зная уровень устройства, система может решить, допустимо ли записывать на него информацию с определенной меткой. Например, по-

пытка напечатать строго конфиденциальную информацию на принтере общего пользования с уровнем «Не конфиденциально» потерпит неудачу.

Принудительное управление доступом обычно реализуют в системах требующих повышенной защищенности. Независимо от практического использования принципы принудительного управления являются удобным методологическим базисом для начальной классификации информации и распределения прав доступа. Удобнее мыслить в терминах уровней конфиденциальности и категорий, чем заполнять неструктурированную матрицу доступа.

В Российской Федерации наличие мандатного разграничения прав доступа, а также контроль потоков информации в автоматизированных системах является требованием контролирующих органов и описан в [2]. Согласно [3] наличие мандатного доступа является критерием защищенности и необходимо для выполнения требований к классам защищённости начиная с 2А и выше.

Согласно [2] средство защиты информации должно реализовывать мандатный принцип контроля доступа применительно ко всем объектам при явном и скрытом доступе со стороны любого из субъектов:

- субъект может читать объект, только если иерархическая классификация в классификационном уровне субъекта не меньше, чем иерархическая классификация в классификационном уровне объекта, и неиерархические категории в классификационном уровне субъекта включают в себя все иерархические категории в классификационном уровне объекта;

- субъект осуществляет запись в объект, только если классификационный уровень субъекта в иерархической классификации не больше, чем классификационный уровень объекта в иерархической классификации, и все иерархические категории в классификационном уровне субъекта включаются в неиерархические категории в классификационном уровне объекта.

Реализация мандатных прав разграничения доступа должна предусматривать возможности сопровождения: изменения классификационных уровней субъектов и объектов специально выделенными субъектами.

В средстве вычислительной техники должен быть реализован диспетчер доступа, т.е. средство, осуществляющее перехват всех обращений субъектов к объектам, а также разграничение доступа в соответствии с заданным принципом разграничения доступа. При этом решение о санкционированности запроса на доступ должно приниматься только при одновременном разрешении его и дискреционными, и мандатными правами разграничения доступа. Таким образом, должен контролироваться не только единичный акт доступа, но и потоки информации.

Требования связанные с правами разграничения доступа включают в себя не только их наличие, но и их периодическое тестирование:

- согласно [3] должно проводиться периодическое тестирование функций средств защиты информации от несанкционированного доступа. при изменении программной среды и персонала автоматизированных систем с помощью тест - программ, имитирующих попытки несанкционированного доступа;

- согласно [4] обеспечивающие средства для систем разграничения доступа должны выполнять функцию тестирования;

- согласно [5] оценка эффективности реализованных в рамках системы защиты персональных данных мер по обеспечению безопасности персональных данных проводится оператором самостоятельно или с привлечением на договорной основе юридических лиц и индивидуальных предпринимателей, имеющих лицензию на осуществление деятельности по технической защите конфиденциальной информации;

- согласно [6] при проведении аттестационных испытаний должны применяться следующие методы проверок: испытания системы защиты информации путем осуществления попыток несанкционированного доступа к информационной системе в обход ее системы защиты информации.

Несмотря на то, что у многих средств защиты есть различные возможности самотестирования, не каждое из них поддерживают оценку корректности реализации средств разграничения доступа. Для решения этой задачи можно либо прибегнуть к помощи специализированного программного обеспечения, либо прове-

сти оценку вручную, что определенно не обеспечит точные результаты тестирования. Примером такого программного обеспечения является программный комплекс, состоящий из двух программ: «Ревизор 1 ХР» и «Ревизор 2 ХР» являющийся де-факто единственным сертифицированным средством подобного рода.

Целью данной выпускной квалификационной работы является создание алгоритма программы, реализующей контроль разграничения доступа согласно требованиям нормативно правовых актов и осуществляющей работу посредством средств операционной системы. Для достижения этой цели были поставлены задачи:

- проанализировать работу, преимущества и недостатки де-факто единственного сертифицированного средства контроля защищенности осуществляющего контроль полномочного разграничения доступа «Ревизор ХР».

- исследовать подход к системам разграничения доступа с точки зрения поведенческой модели пользователя;

- проанализировать реализацию полномочного доступа в различных средствах защиты информации;

- разработать и описать алгоритм программы, реализующей контроль разграничения доступа.

ГЛАВА 1 ОБЗОР СРЕДСТВА КОНТРОЛЯ ЗАЩИЩЕННОСТИ РЕВИЗОР XP

В данной главе рассмотрим функционирование, а также достоинства и недостатки наиболее популярного, на момент написания выпускной квалификационной работы, средства контроля защищенности «Ревизор XP».

Данное средство контроля защищенности представляет из себя программный комплекс компонентами которого являются:

- «Ревизор 1 XP» - средство создания модели системы разграничения доступа для автоматизации процесса создания избирательной модели системы разграничения доступа пользователей к ресурсам;

- «Ревизор 2 XP» - программа контроля полномочий доступа информационным ресурсам.

1.1 Начальные условия

В целях демонстрации контроля реализации прав разграничения доступа средством контроля защищенности «Ревизор XP» используем правила мандатного управления доступом настроенные с помощью средства защиты информации «Secret Net 7» установленном в операционной системе «Windows 7 Максимальная». Перед началом были созданы три папки имеющих разные метки конфиденциальности и активированный чек-бокс «Автоматически присваивать новым файлам» содержащие по одному текстовому файлу (Таблица 1). Все вышеупомянутые папки находятся в каталоге «C:\»; Были созданы 3 пользователя имеющие разные уровни допуска (Таблица 2).

1.2 Тестирование работы «Ревизор XP»

Для проведения тестирования, в первую очередь, создадим модель разграничения доступа. Это схематичное выражение правил, реализованных посредством средства зашиты «Secret Net 7» отражающее текущие правила разграничения до-

ступа, файлы, участвующие в тестировании, а также существующих пользователей. Создадим эту модель в программе «Ревизор 1 XP».

Таблица 1 – Описание тестовых файлов

Имя папки	Имя файла	Метка конфиденциальности
Не конфиденциально	тест1.txt	Не конфиденциально
Конфиденциально	тест2.txt	Конфиденциально
Строго конфиденциально	тест3.txt	Строго конфиденциально

Таблица 2 – Описание пользователей

Имя пользователя	Уровень допуска
Максимальный	Строго конфиденциально
Средний	Конфиденциально
Минимальный	Не конфиденциально

Первым шагом сканируем, выбранный пользователем, логический диск на предмет существующих файлов и пользователей, после чего программа отобразит их в виде списка, который необходимо отредактировать следующим образом:

- удалить не участвующих в тестировании пользователей;

- отметить плюсами все права R W D (Эти права обозначают R- read W- write D – delete и относятся к дискреционному управлению доступом которое не участвует в тестировании, и их установка в положение плюсов – все разрешено необходима что бы исключить конфликты в ходе проверки);

- выставить соответствующие метки файлам и пользователям. (Рисунок 1)

Метки конфиденциальности в программе «Ревизор 1 XP» не подлежат редактированию, в связи с этим, для выставления их в модель, доступно всего три

уровня конфиденциальности которые мы условно свяжем с уровнями допуска (Таблица 3). Результатом вышеописанных действий является файл с расширением «.arx», который необходимо загрузить в следующую программу комплекса – «Ревизор 2 XP». В функции этой программы входит непосредственно тестирование прав разграничения доступа, основанное на вышеупомянутой модели.

Таблица 3 – Название уровней конфиденциальности в программе «Ревизор 1 XP»

Название в программе	Уровень допуска
-	Не конфиденциально
С	Конфиденциально
СС	Строго конфиденциально

До запуска тестирования создадим его план, где удалим не участвующие в тестировании файлы и выберем пользователя чьи права необходимо проверить (Рисунок 2).

Результаты тестирования, в форме сравнения требуемых в модели и фактических прав доступа, будут выведены на экран (Рисунок 3). Так же существует возможность сформировать файл с результатами, имеющий расширение «.html» (Рисунок 4).

1.3 Результаты тестирования

Средство контроля защищенности «Ревизор XP» однозначно справляется с выполнением возложенных на него задач, тем не менее оно имеет некоторые недостатки, о которых пойдет речь далее.

Первым и главным недостатком этого комплекса программ на момент получения задания на выпускную квалификационную работу являются истекшие сертификаты Федеральной службы по техническому и экспортному контролю (ФСТЭК). Программный комплекс «Ревизор XP» имел два сертификата: один для

«Ревизор 1 XP» действующий до 08 февраля 2017 года и второй для «Ревизор 2 XP» соответственно действующий также до 08 февраля 2017 года.

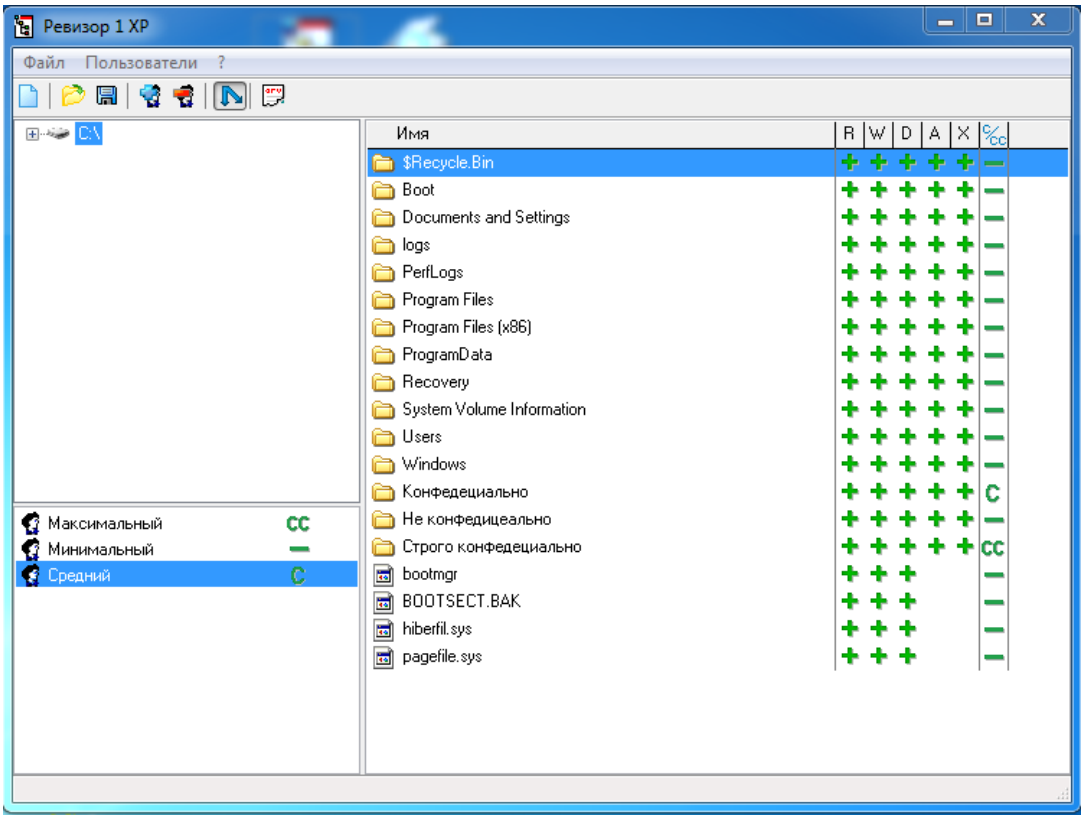


Рисунок 1 – Создание модели разграничения доступа в программе «Ревизор 1 XP».

Использование средств контроля защищенности без сертификатов может повлечь за собой определенные последствия, как в правовом аспекте, так и в аспекте безопасности.

Вторым недостатком является достаточно долгий период жизни и отсутствие поддержки. Программа, выпущенная в 2004 году, однозначно требует обновления, ведь компьютерная индустрия стремительно развивается каждый год и в настоящее время повсеместно используются операционные системы, которые не поддерживаются «Ревизор XP», такие как: «Windows 8», «Windows 10», «Windows server 2008», «Windows Server 2012».

Третьим недостатком, является отсутствие поддержки сертифицированных unix-подобных систем таких как: «Astra Linux Special Edition», «SUSE Linux» и «Альт Линукс СПТ 7.0».

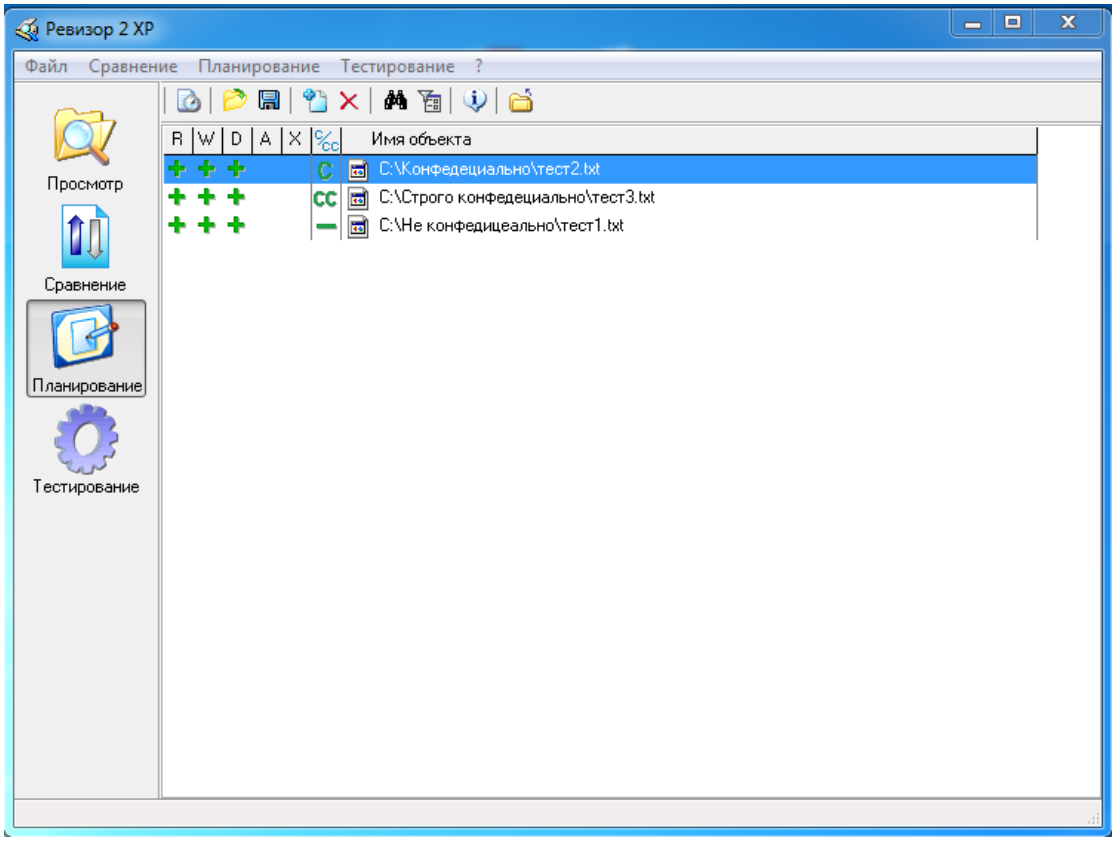


Рисунок 2 – Создание плана тестирования в программе «Ревизор 2 XP».

Четвертый недостаток - есть невозможность проверить всех нужных пользователей за одно тестирование. Это многократно увеличивает время проведения тестирования, и также увеличивает количество отчетов о его проведении, тем самым ухудшает наглядность результатов. Здесь имеет место антропогенный фактор и трудно не согласится, что проверяющий может не заметить ошибку в n-ном количестве отчетов гораздо с большей вероятностью чем в одном.

Пятый недостаток заключается в логике работы «Ревизор 2 XP». В начале тестирования прав доступа определенного пользователя программа запрашивает повышение уровня конфиденциальности для своей вспомогательной программы «Revizor2XP_tester.exe» и продолжает тестирование с помощью нее.

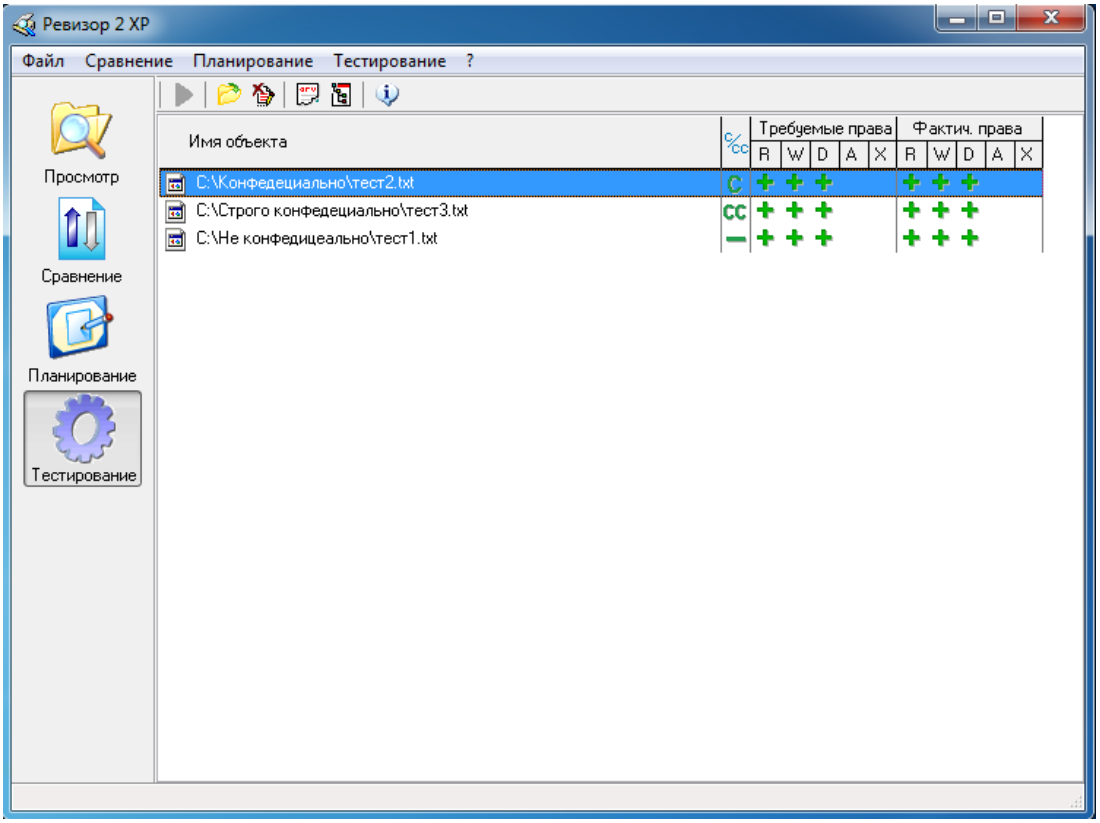


Рисунок 3 – Результаты тестирования в программе «Ревизор 2 XP».

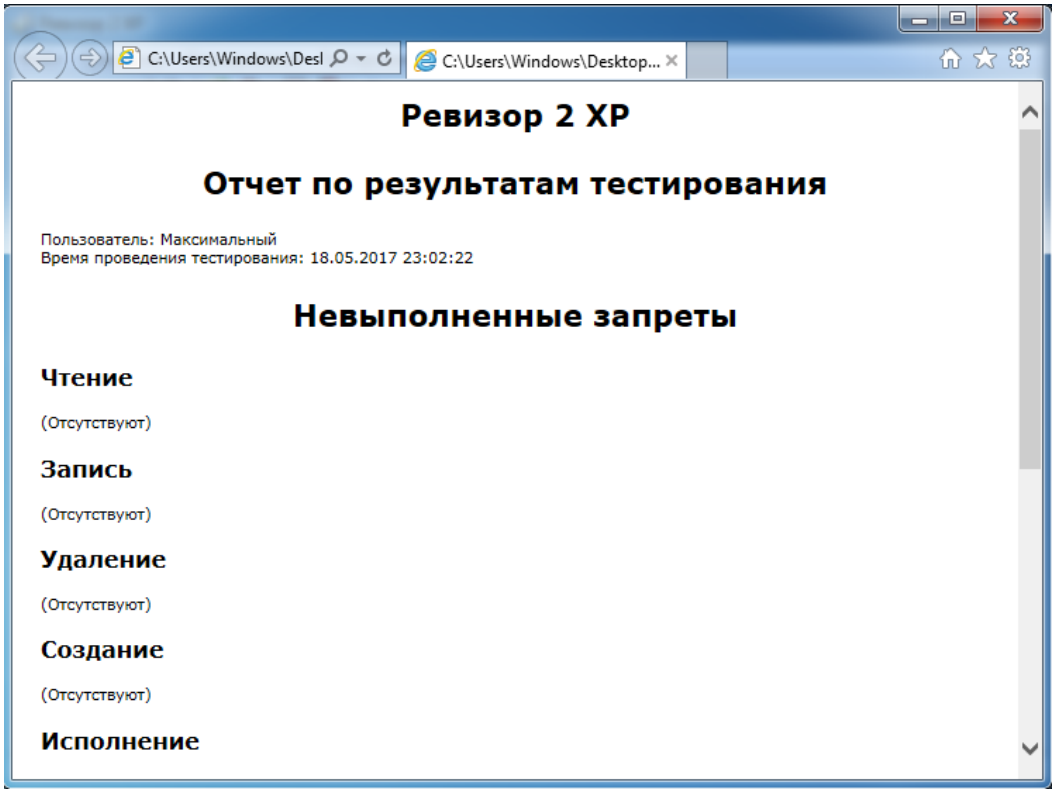


Рисунок 4 – Результаты тестирования прав доступа в программе «Ревизор 2 XP».

При этом уровень конфиденциальности не аннулируется при переходе к тестированию прав следующего уровня, что может привести к ошибкам в результатах (Рисунок 5).

1.4 Вывод по главе

Анализ работы средства контроля защищенности «Ревизор XP» успешно проведен.

В ходе анализа был выполнен полный цикл проверки прав полномочного доступа и выделены некоторые особенности. Опираясь на материалы данной главы можно сделать вывод о том некоторые недостатки де-факто единственного в своем роде средства не поддаются исправлению и существует необходимость создания инструмента решающего эту проблему.

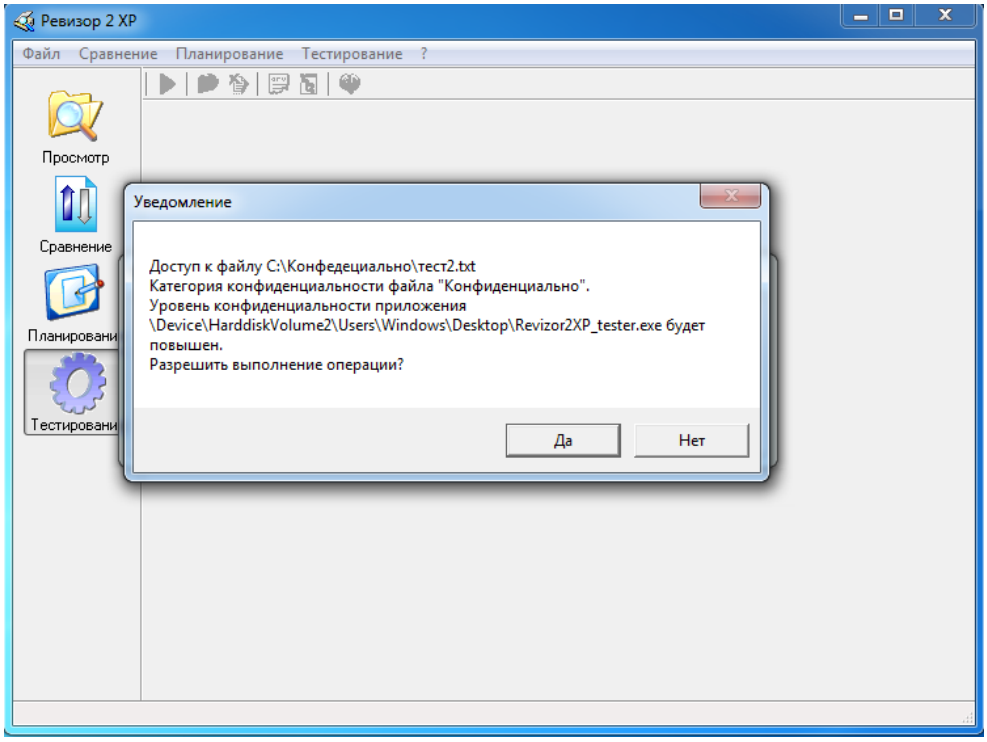


Рисунок 5– Запрос на повышение уровня конфиденциальности процесса «Revizor2XP_tester.exe».

ГЛАВА 2 ИССЛЕДОВАНИЕ СИСТЕМЫ РАЗГРАНИЧЕНИЯ ДОСТУПА НА ОСНОВЕ ПОВЕДЕНЧЕСКОЙ МОДЕЛИ ПОЛЬЗОВАТЕЛЯ

2.1 Постановка задачи

Для оценки охвата системы разграничения доступа сохранив при этом комплексность подхода, необходимо в ручном режиме провести тестирование всех элементов системы разграничения доступа поэтапно проходя все стадии технологического процесса автоматизированной системы от лица пользователей со всеми доступными уровнями допуска.

В данной ситуации от тестирующего требуется понимание технологического процесса, установленного в автоматизированной системе, а также знание тонкостей настройки средства защиты информации.

При проведении оценки специалистом, не обладающим необходимым объёмом знаний о связи настроек средства защиты информации и технологическим процессом, установленным в операционной системе, ошибки могут произойти с большой вероятностью

Однако существует возможность частично автоматизировать действия проверяющего при помощи специальной модели поведения. Подходящим решением будет формализация модели поведения пользователя в автоматизированной системе.

Модель такого рода была предложена в исследовании [7] в виде алгоритма, представленного на рисунке 6.

2.2 Описание эксперимента

В целях проверки работоспособности модели, авторами работы был проведён эксперимент, в котором, сравнивались результаты использования этого алгоритма не обладающими экспертными знаниями тестирующими пользователями, с выводами о корректной настройке средства защиты информации.

Общая схема алгоритма, моделирующего поведение пользователя, включает:

- выбор папок, к которым заранее настроены права доступа в средства защиты информации, с указанием уровневой метки доступа;

- выбор текущей уровня конфиденциальности;

- создание новых файлов в текущих папках;

- тестовая запись в созданные файлы;

- копирование отредактированных файлов;

- попытка чтения файлов, созданных пользователями, в тех папках, в которых создание новых файлов не удалось;

- копирование информации из файлов, чтение из которых удалось, в созданные файлы;

- печать всех созданных и изменённых файлов на всех возможных принтерах;

- удаление всех созданных файлов.

Тестирующие были разделены на две группы, первая из которых настраивала средство защиты информации «Secret Net 7», а другая – средство защиты информации «Страж NT 3.0». Обе группы:

- произвели установку и настройку предложенных средств защиты информации, пользуясь дополнительны ми инструкциями [8][9];

- сделали заключение о корректности настройки средства защиты информации другой группы;

2.3 Результаты

Результаты заключений групп сопоставлены с результатами выполнения скрипта, реализующего алгоритм (Таблица 4). Из приведенного вывода по эксперименту:

- ошибки, возникающие при создании папок и файлов связаны с неверной настройкой прав разграничения доступа (п. 1.1 таблица 4);

- ошибки настройке системы разграничения доступа обнаруженные при копировании папок связаны с особенностями средств защиты информации (п. 1.2 таблица 4);

- при создании и копировании файлов тестирующие неверно истолковывали полученные результаты, в некоторых случаях считая ошибочным отказ в создании файлов, что назовем ошибкой первого рода. (п. 2.1, 2.3 таблица 4);

- при создании и копировании папок и файлов ошибки тестирующих связаны с неверным истолкованием полученных результатов вследствие некорректной настройки другой группой механизма контроля потоков в выбранных средствах защиты информации что назовем ошибкой 2 рода (п. 2.2, 2.4 таблица 4).

Таблица 4. Результаты заключений, тестирующих и работы реализации алгоритма

№ п/п	Критерии сравнений	Результаты заключений тестирующих	Реализации алгоритма
1.	Ошибки настройки СРД		
1.1.	Кол-во ошибок при создании файлов и папок	1(10%)	7(70%)
1.2.	Кол-во ошибок при копировании папок и файлов	5(50%)	5(50%)
2.	Ошибки тестирующих и реализации алгоритма		
2.1.	Кол-во ошибок 1 рода при создании файлов и папок	3(20%)	0(0%)
2.2.	Кол-во ошибок 2 рода при создании файлов и папок	2(20%)	0(0%)
2.3.	Кол-во ошибок 1 рода при копировании файлов и папок	3(40%)	0(0%)
2.4.	Кол-во ошибок 2 рода при копировании файлов и папок	5(40%)	0(0%)

2.4 Вывод по главе

Исследование подхода к системам разграничения доступа с точки зрения поведенческой модели пользователя проведено и можно сделать вывод что даже те пользователи, в чьей работе присутствует ежедневное взаимодействие со средствами защиты информации совершают ошибки и делают не верные выводы. На основе этих наблюдений было принято решение что полная автоматизация этого процесса целесообразна. Исходя из этого следует что поставленная цель актуально, но для ее достижения еще необходимо подробно рассмотреть, как реализованы механизмы мандатного контроля доступа в различных средствах защиты информации.

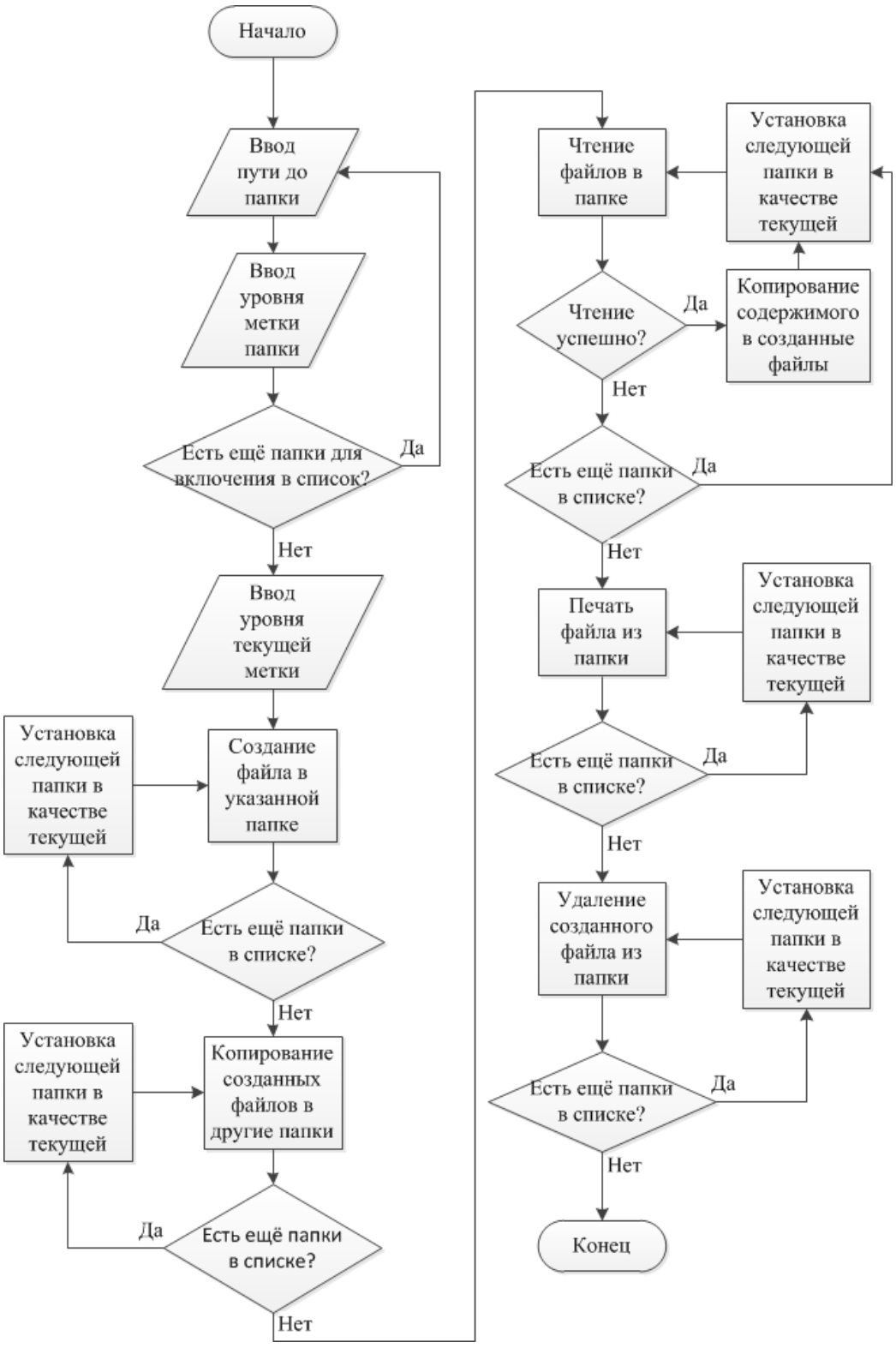


Рисунок 6 – Блок схема алгоритма поведенческой модели пользователя.

ГЛАВА 3 СРАВНЕНИЕ РЕАЛИЗАЦИЙ ПОЛНОМОЧНОГО РАЗГРАНИЧЕНИЯ ДОСТУПА В СРЕДСТВАХ ЗАЩИТЫ ИНФОРМАЦИИ.

В данной главе будут рассматриваться реализации мандатного доступа в средствах защиты информации: «Secret Net 7», «Страж NT 3.0», «Astra Linux Special Edition» релиз «Смоленск»» с использованием материалов [10]. С целью углубления знаний в сфере реализации мандатного доступа. Выбор средств защиты основан на двух критериях: доступность экземпляра средства защиты для эксперимента и различие в технологических подходах к решению задач разграничения доступа. Для тестирования средств защиты информации, работающих в семействе операционных систем «Windows» использовались программные продукты от «Sysinternals»:

– «Autoruns» (отражает перечень драйверов, служб, модулей оболочки и входа в операционную систему, и другое);

– «Process Explorer» (показывает работающие процессы, а также их подчинённость, используемые файлы и директории);

– «Process Monitor» (отслеживает действия всех процессов в системе, в том числе драйверов и библиотек, позволяет установить драйвер мониторинга с самого начала загрузки операционной системы).

С помощью этих программ проанализировано взаимодействие средств защиты информации с операционной системой, в том числе влияние на запросы программного обеспечения к защищаемой информации и устойчивость к сбоям.

3.1 «Secret Net 7»

Рассмотрим реализацию мандатного разграничения доступа в средстве защиты информации «Secret Net 7». Механизм разграничения доступа работает как служба операционной системы и реализован в форме драйвера-фильтра. Этот драйвер начинает работу еще до загрузки операционной системы и благодаря ему настройка мандатного доступа интегрируется в файловые операции и стандарт-

ные службы настройки системы, но при желании можно использовать вариант автоматической настройки мандатного разграничения доступа. С функциональной стороны это средство защиты предусматривает два режима работа в аспекте реализации мандатного разграничения доступа:

- со включённым контролем потоков;
- с отключённым контролем потоков.

Режим со включённым контролем потоков описывается разработчиком следующим образом: Используется режим контроля потоков конфиденциальных документов при работе механизма полномочного разграничения доступа: обеспечивается строгое соблюдение принципов полномочного разграничения доступа и предотвращение копирования/перемещения конфиденциальной информации с понижением ее уровня. Возможность работы с конфиденциальными ресурсами определяется уровнем конфиденциальности сессии, который устанавливается при входе пользователя в систему. Для включенного режима контроля потоков можно определить дополнительные параметры:

- «строгий контроль терминальных подключений» — на компьютерах, осуществляющих терминальный доступ к компьютеру с включенным режимом контроля потоков, также должен быть включен режим контроля потоков. При этом уровень конфиденциальности терминальной сессии при подключении автоматически устанавливается равным уровню локальной сессии на клиенте (соответственно, режим контроля потоков также должен быть включен на клиенте);

- «автоматический выбор максимального уровня сессии» - включает принудительное назначение максимально возможных уровней конфиденциальности для сессий пользователей. Сессии будет назначаться уровень конфиденциальности, равный уровню допуска пользователя, который выполняет вход в систему.

Другими словами, средство защиты информации жестко контролирует единоразовое присвоение метки пользователю при входе и полностью гарантирует неизменность этой метки до завершения сессии. (Рисунок 7). Так же выполняются требования [3] то есть пользователи с доминирующей над ресурсом меткой не

имеют к нему полного доступа. Они лишены возможности копировать, изменять или же удалять ресурс, имеется только возможность чтения – это гарантирует сохранность текущего уровня конфиденциальности ресурса, и обеспечивает предотвращение понижения его уровня конфиденциальности. Включенный режим потоков так же ограничивает настройку средства защиты информации, когда у сессии любой уровень допуска кроме «Не конфиденциально».

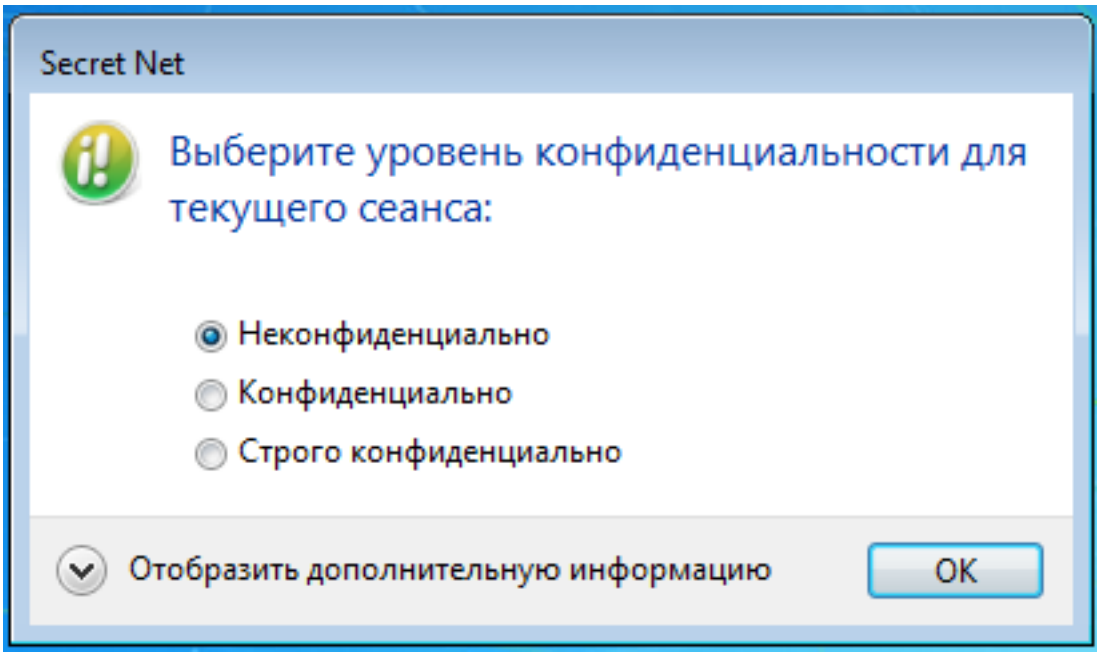


Рисунок 7 – Окно выбора уровня конфиденциальности сессии в «Secret Net 7» со включенным контролем потоков.

Если же контроль потоков отключен, то доступ пользователей к конфиденциальным ресурсам осуществляется на основе уровней допуска пользователей (Рисунок 8). Разрешается доступ ко всем ресурсам, категория которых не выше уровня допуска пользователя. После получения доступа к конфиденциальному ресурсу пользователь может выполнять с ним любые действия.

В этом режиме процессы операционной системы не привязаны к сессии, и если какой-либо процесс нуждается в допуске к ресурсу, то если он запущен от лица пользователя с доминирующей меткой, ему необходимо запросить разрешение на

повышение допуска не в интерактивном режиме. (Рисунок 9). И лишь после того как пользователь подтвердит запрос доступ к ресурсу будет предоставлен. Настройки средства защиты информации в этом режиме могут проводится администратором безопасности с любым уровнем допуска.

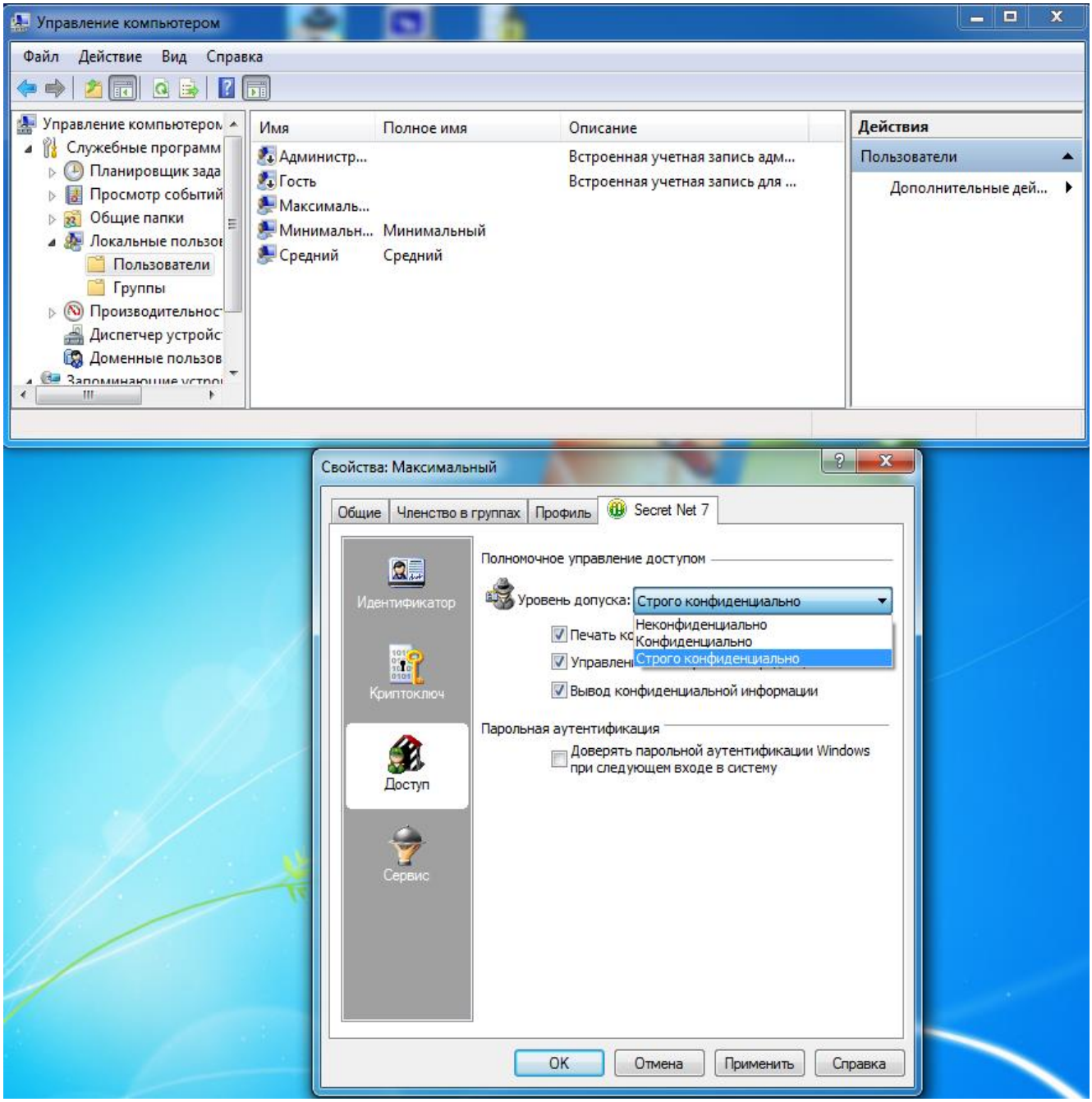


Рисунок 8 – Выбор уровня допуска в средстве защиты информации в средстве защиты информации «Secret Net 7».

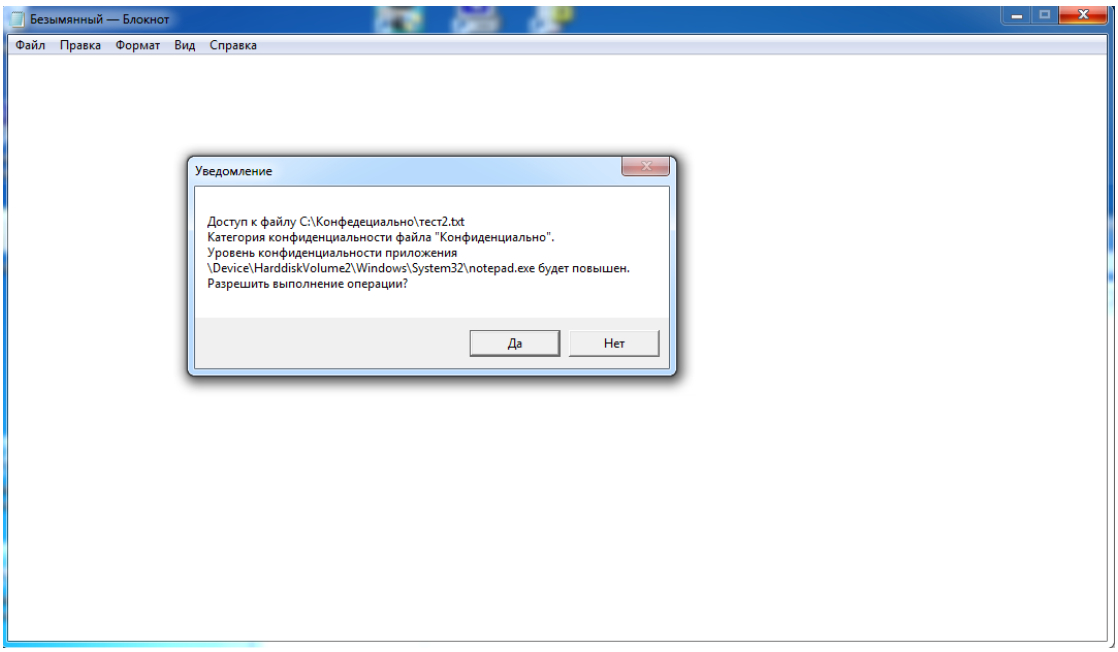


Рисунок 9 – Запрос на повышение уровня конфиденциальности процесса.

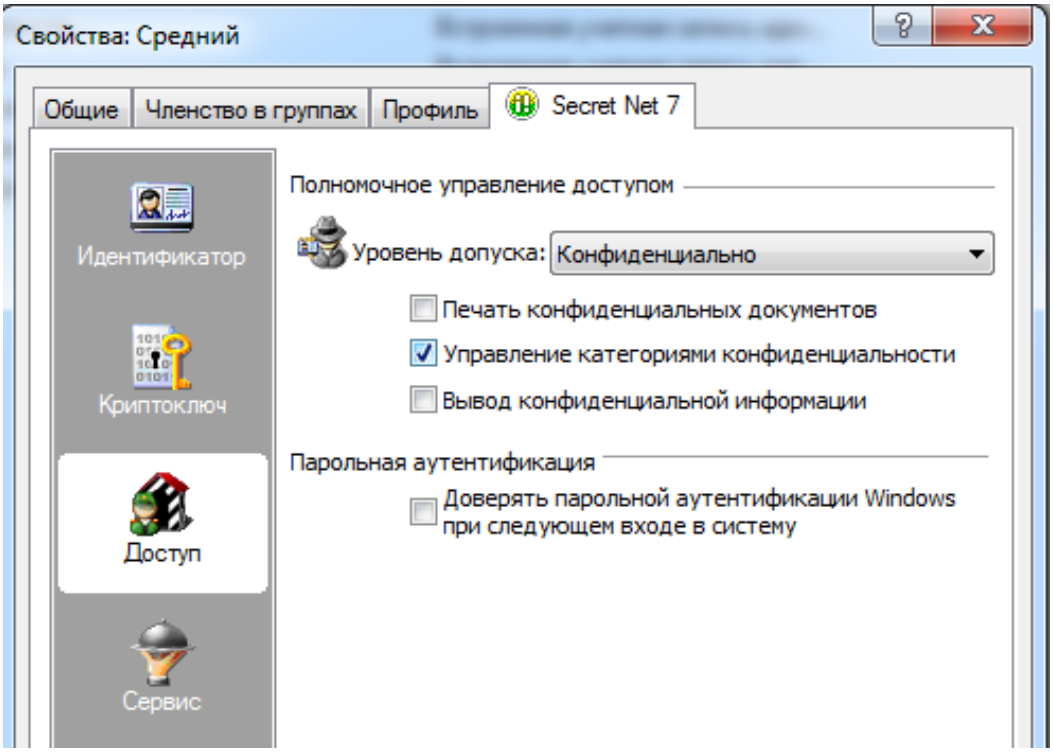


Рисунок 10 – Настройка управления категориями конфиденциальности и разрешением на печать конфиденциальных документов.

3.2 «Страж NT 3.0»

Для разделения доступа так же, как и в «Secret Net 7», используются драйверы-фильтры, которые перенаправляют запросы в случае обращений к файлам и устройствам внутренним механизмам средства защиты информации. «Страж NT 3.0» интегрирован в операционную систему и просмотр метки конфиденциальности доступен из стандартного контекстного меню «свойства» (Рисунок 11).

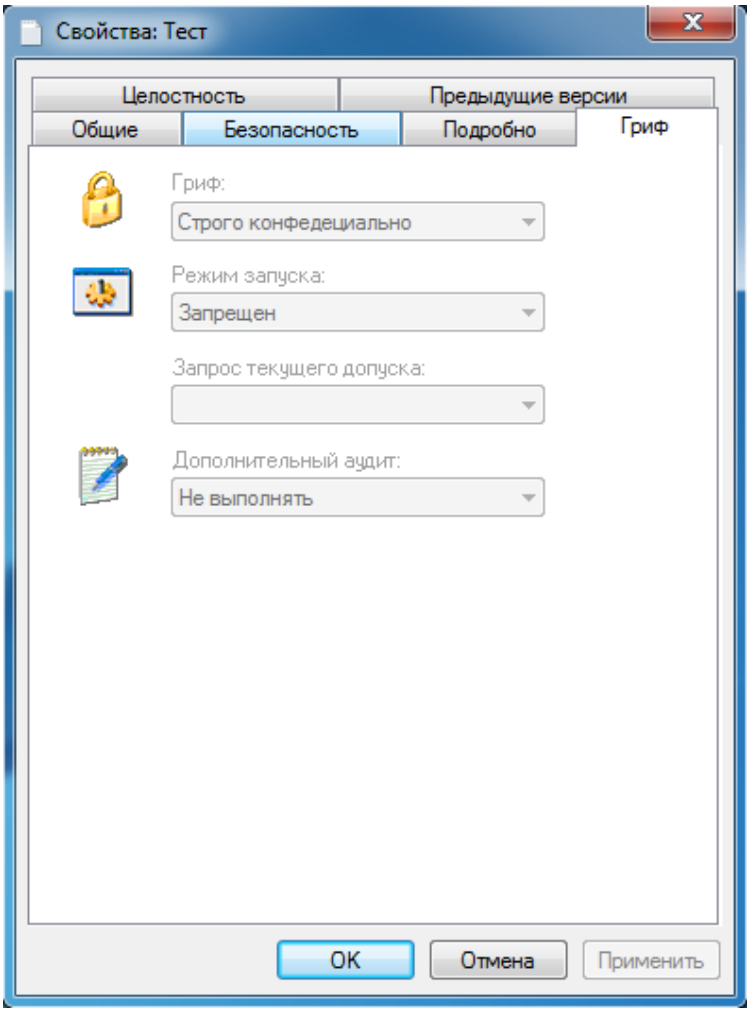


Рисунок 11 – Стандартное контекстное меню свойства.

В то время как управление механизмами защиты происходит из специальных программ, входящих в состав средства защиты информации. Управление конкретно механизмом мандатного разграничения доступа осуществляется с помощью программ «Менеджер файлов» (Рисунок 12) и «Менеджер пользователей»

(Рисунок 13). При этом должен быть активирован режим администрирования. Возможность войти в режим администрирования может быть предоставлена любому пользователю на усмотрение назначенного при установке администратора.

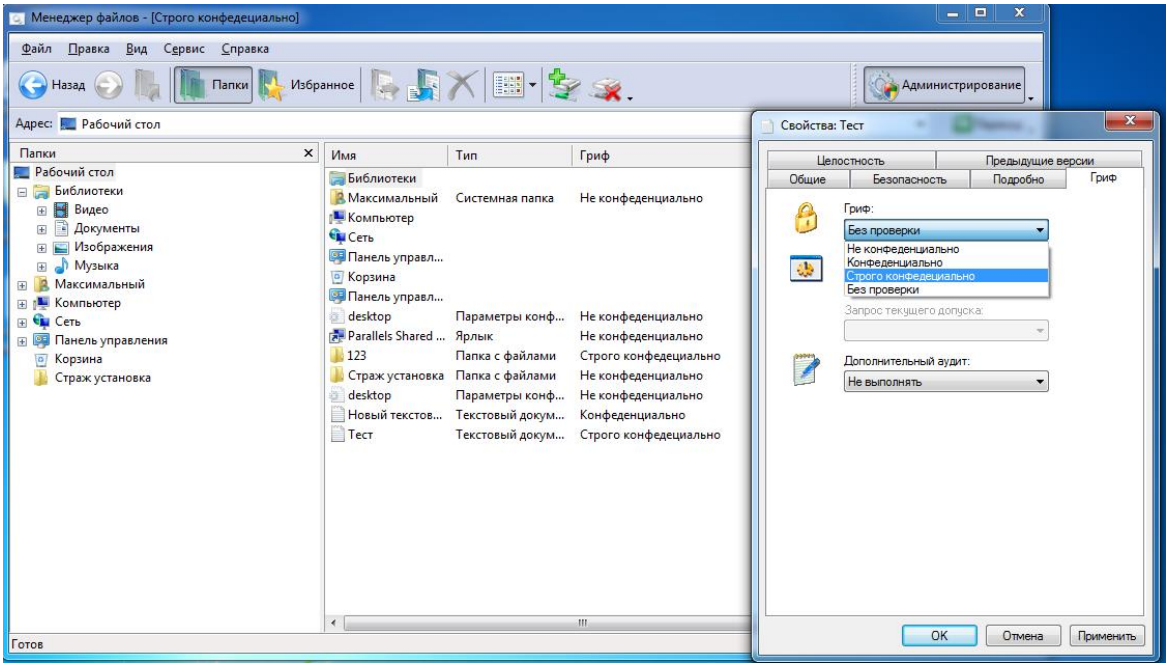


Рисунок 12 – Редактирование метки конфиденциальности в «Страж NT 3.0».

Интересной особенностью данного средства защиты информации является тот факт, если пользователю не доступен какой-либо ресурс, то к этому ресурсу не только ограничивается доступ, но еще и само наличие этого ресурса скрывается от пользователя. Метка допуска приложений на действия с конфиденциальными файлами может выбираться пользователем, но имеет верхнюю границу равную метке этого пользователя (Рисунок 14) и даже при доминировании метки пользователя доступ к ресурсу будет ограничен.

3.3 «Astra Linux Special Edition» релиз «Смоленск»»

Данное средство защиты информации отличает от остальных то, что оно, само по себе, является операционной системой. Решение задачи мандатного разграни-

чения доступа процессов к ресурсам основано на реализации соответствующего механизма в ядре операционной системы.

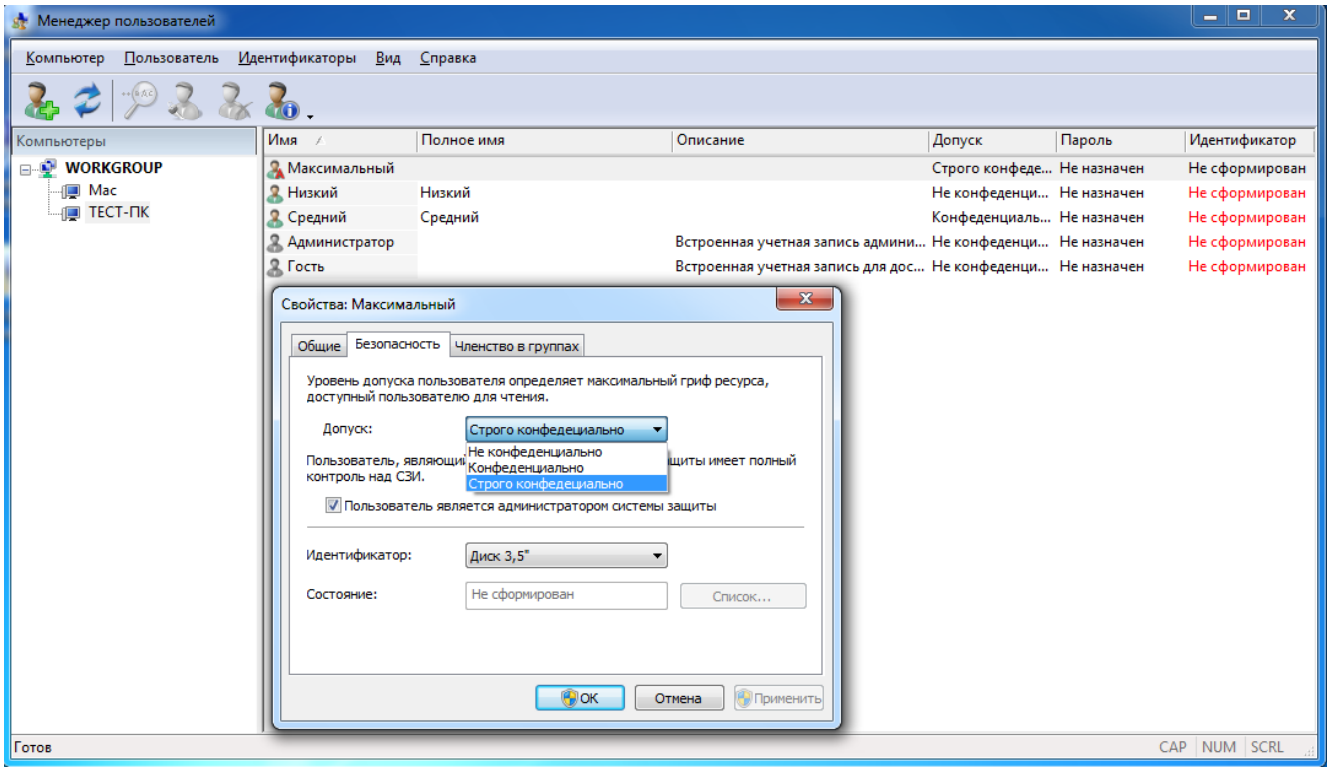


Рисунок 13 – Редактирование уровня допуска пользователя в «Страж NT 3.0».

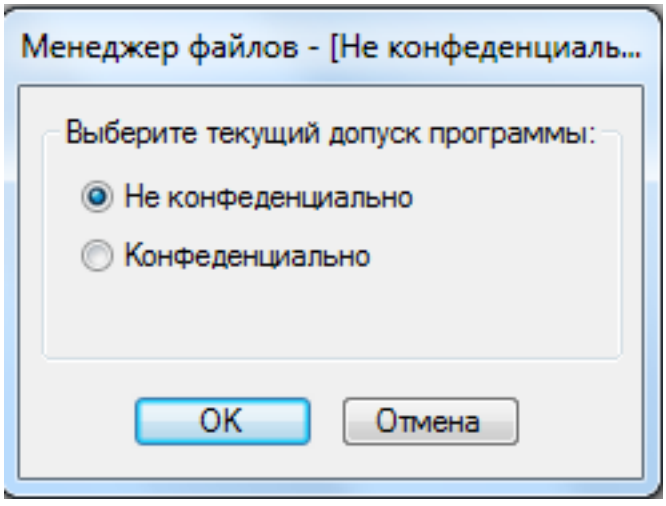


Рисунок 14 – Выбор уровня допуска приложения.

Согласно [11] механизм управления мандатным доступом, встроен в рабочий стол «Fly», он позволяет администратору устанавливать, разрешённый диапазон уровней конфиденциальности и категорий конфиденциальности отдельно для каждого пользователя. Категории введены с целью разделить «горизонтальные» уровни допуска «вертикально». То есть, если пользователь имеет допуск «Строго конфиденциально» и категорию «А», то он не сможет прочитать файл, имеющий метку «Конфиденциально» и категорию «Б». Для настройки данного механизма используется графическая утилита «fly-admin-smc» (Рисунок 15). В этой программе можно настроить как сами уровни конфиденциальности, так их принадлежность пользователям.

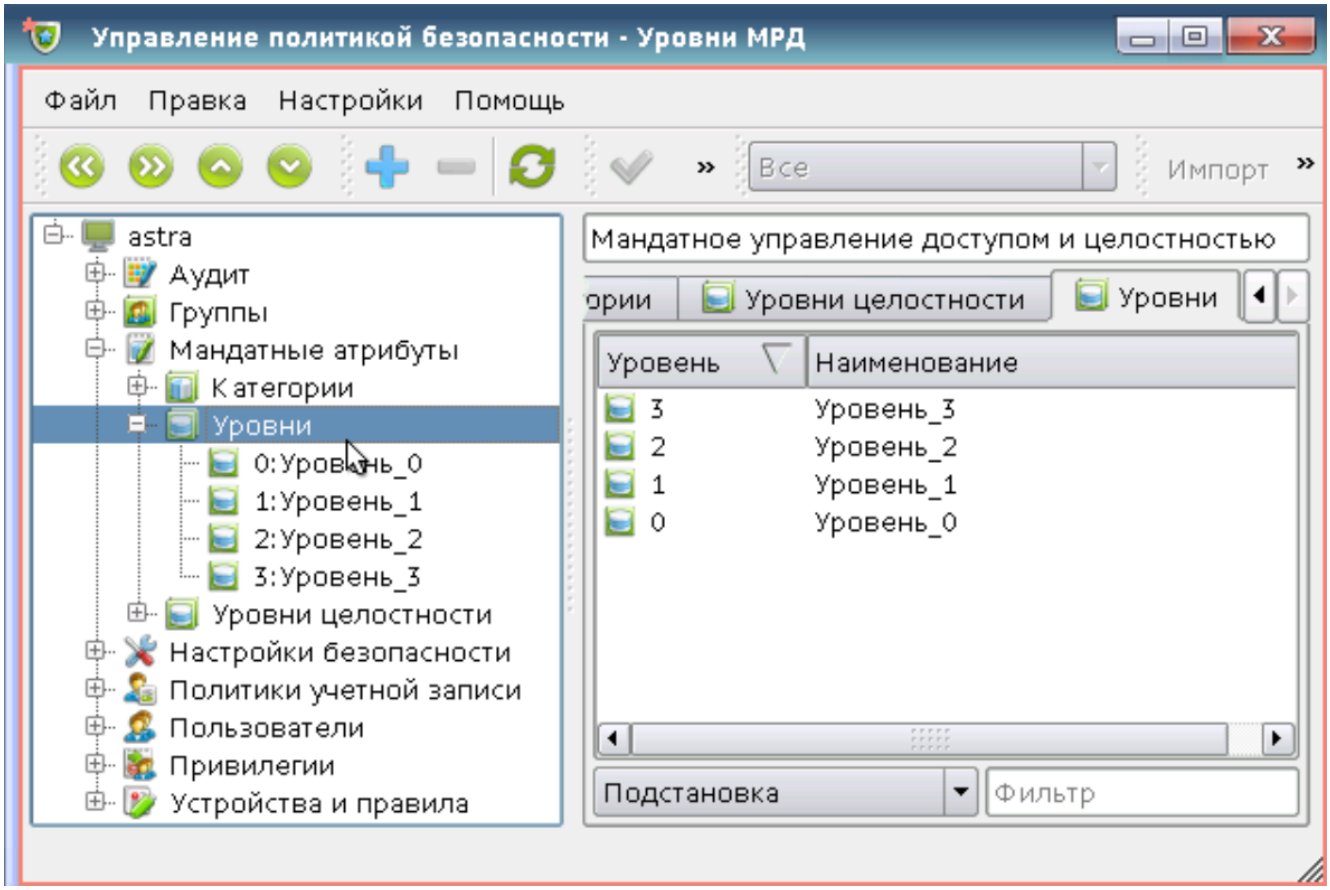


Рисунок 15- Программа «fly-admin-smc».

После того, как пользователь, для которого установлены определенные мандатные уровни и категории, отличные от нуля, войдёт в систему, ему будет пре-

дложено установить конкретный уровень допуска и конкретную категорию для данной сессии в пределах разрешенных администратором диапазонов. Выбранные значения этих параметров можно будет проверить с помощью индикатора в виде кружка с числом внутри, расположенного в области уведомлений на панели задач в правом нижнем углу рабочего стола (Рисунок 16).

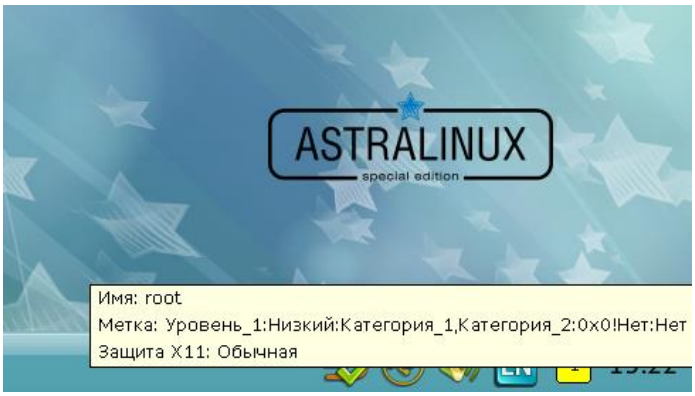


Рисунок 16 – Индикатор уровня допуска в «Astra Linux Special Edition» релиз «Смоленск»».

Подробней рассмотрим механизм принятия решений. Правила принятия решения могут быть записаны следующим образом: Пусть контекст безопасности субъекта содержит уровень L0 и категории C0, а мандатная метка объекта содержит уровень L1 и категории C1. В «Astra Linux Special Edition» релиз «Смоленск»» определены следующие операции сравнения уровней и категорий:

– уровень L0 меньше уровня L1 если численное значение L0 меньше численного значения L1;

– уровень L0 равен уровню L1 если численные значения L0 и L1 совпадают;

– категории C0 меньше категорий C1 если все биты набора C0 являются подмножеством набора бит C1;

– категории C0 равны категориям C1 если значения C0 и C1 совпадают.

Таким образом, в механизме мандатного разграничения доступа действуют следующие правила:

– операция записи разрешена, если L0=L1 и C0=C1;

– операция чтения разрешена, если L0>=L1 и C0>=C1;

– операция исполнения разрешена, если L0>=L1 и C0>=C1.

Уровни доступа наследуются процессами от своих родителей. Процессы, запущенные от имени администратора, независимо от наличия у них определенных прав имеют возможность осуществлять все привилегированные действия.

3.4 Сводная таблица сравнения

Далее приведена таблица сравнения характеристик выше рассмотренных средств защиты информации с точки зрения реализации мандатного доступа основная на материалах данной главы.

Таблица 5 – Сводная таблица сравнения средств защиты информации

	«Secret Net 7»	«Страж NT 3.0»	«Astra Linux»
Реализация контроля доступа к файлам	Реализованы как отдельные драйверы-фильтры и службы операционной системы	Реализованы как отдельные драйверы-фильтры	Элемент ядра операционной системы
Автоматическая настройка мандатного доступа	Имеется отдельная утилита для автоматической настройки с малым количеством шаблонов	Самообучаем, имеется самообучение и режим автозапуска, очень большое количество шаблонов	На данный момент авто настройка отсутствует, но существуют дополнения с такой функцией
Дополнительный аудит	Контроль целостности проводится при загрузке операционной системы, при входе в систему или же по запросу администратора. Проводится аудит при обращении к ресурсу, ведется журнал. Имеется отдельное ПО для настройки	Контроль целостности проводится при загрузке операционной системы, или же по запросу администратора. Проводится аудит при обращении к ресурсу. Настраивается через свойства ресурса	Контроль целостности проводится при загрузке операционной системы, при входе в систему или же по запросу администратора. Проводится аудит при обращении к ресурсу, ведется журнал. Имеется отдельное по для настройки
Настройка	Интегрирована в операционную систему	Отдельная программа настройки	Интегрирована в операционную систему
Контроль потоков	С возможностью отключения	+	+

3.3 Вывод по главе

Проанализирована реализация полномочного доступа в трех средствах защиты информации: «Secret Net 7», «Страж NT 3.0», ««Astra Linux Special Edition» релиз «Смоленск»» и отмечены ее особенности. Исходя из полученных данных, примем решение о разработке алгоритма программы, реализующей контроль настройки полномочного разграничения доступа, осуществленного в «Astra Linux», так как на данный момент средство контроля, как для этой операционной системы, так и для других Unix-подобных систем отсутствует, и не имеется барьеров для разработки скриптов в данной среде, связанных с обучением логике работы и основными структурами интерпретатора команд.

ГЛАВА 4 ОПИСАНИЕ АЛГОРИМА РАБОТЫ ПРОГРАМНОГО СРЕДСТВА ОСУЩЕСТВЛЯЮЩЕГО КОНТРОЛЬ РЕАЛИЗАЦИИ МАНДАТНОЙ СИСТЕМЫ РАЗГРАНИЧЕНИЯ ДОСТУПА

Для начального варианта алгоритма предусмотрим наличие только трех уровней конфиденциальности, так как это количество чаще всего используется на практике. В качестве средства защиты информации в котором будет реализовано тестируемое мандатное разграничение доступа выберем «Astra Linux». Этот выбор обусловлен тем, что, во-первых, на данный момент средство контроля для этой системы отсутствует. А во-вторых средство, разработанное под одну из Unix –подобных систем с большой вероятностью будет работать и в остальных, так как они имеют схожий командный интерпретатор. Реализация представленного алгоритма предусматривает использование штатных средств операционной системы, а именно командных интерпретаторов, таких, например, как «Power shell» для семейства «Windows» и «bash» для Unix-подобных систем. В командных интерпретаторах используются скриптовые языки программирования. Реализация алгоритма посредством штатных средств операционной системы повысит доверие к конечному программному обеспечению и не будет нуждаться во внедрении в систему дополнительных сторонних элементов, что положительно скажется на безопасности.

В ходе разработки алгоритма было принято решение что средство контроля мандатного разграничения доступа должно состоять из двух частей:

– инициализирующей (блок-схема алгоритма представлена в приложении А);

– осуществляющей проверку (блок-схема алгоритма представлена в приложении Б).

Инициализирующий алгоритм выполняет задачу формирования начальных данных путем считывания введенных проверяющим с клавиатуры путей до тестовых папок с предустановленными метками конфиденциальности, и формирования инициализирующих файлов, в которых введенная аудитором информация будет

сохранена и структурирована для использования вторым скриптом при запуске проверки. Далее будут перечислены инициализирующие файлы представленных в данной работе алгоритмов, их структура описана в приложении В:

- «nconf.txt» – предназначен для хранения списка пользователей, имеющих метку «Не конфиденциально»;

- «nconf1.txt» – предназначен для хранения пути до папки, метка которой «Не конфиденциально»;

- «conf.txt» – предназначен для хранения списка пользователей, метка которой «Конфиденциально»;

- «conf1.txt» – предназначен для хранения пути до папки, метка которой «Конфиденциально»;

- «sconf.txt» – предназначен для хранения списка пользователей, метка которой «Строго конфиденциально»;

- «sconf1.txt» – предназначен для хранения пути до папки, метка которой «Строго конфиденциально».

Так же в функционал инициализирующего скрипта входит создание во всех тестируемых каталогах нового файла, имеющего аналогичную с папкой метку конфиденциальности и запись в единый файл отчета сведений о невыполненных разрешениях на создание файла.

Опишем используемые в инициализирующем алгоритме переменные:

- «level» - строчная переменная предназначенная для назначения текущего уровня конфиденциальности;

- «temp» - строчная переменная предназначенная для хранения пути до файла с информацией о пользователях, принадлежащих к текущему уровню конфиденциальности;

- «temp1» - строчная переменная предназначенная для хранения пути до файла с информацией о нахождении ресурса, принадлежащего к текущему уровню конфиденциальности;

- «i» - переменная счетчик необходимая для выбора инициализируемого уровня конфиденциальности.

В функционал второго скрипта входит непосредственно сама проверка настройка правил мандатного разграничения доступа в средстве защиты информации. Далее будут перечислены переменные использующиеся в представленном в данной работе алгоритме, их:

- «path» - строчная переменная для хранения текущей тестируемой папки;

- «users» - массив для хранения учетных данных тестируемых пользователей для текущего уровня конфиденциальности;

- «i» - переменная счетчик необходимая для выбора следующего пользователя из массива users.

В алгоритме предусмотрен подход к проверке с точки зрения уровней конфиденциальности, другими словами поэтапная проверка проходит не каждого тестируемого пользователя в порядке очереди, а каждый уровень конфиденциальности что увеличивает достоверность результатов, а так же при проверке каждого пользователя создается отдельный, обособленный процесс, который завершается по окончанию прохождения проверки, что предотвращает возможные ошибки с наложением меток друг на друга и искажения конечного результата проверки. В ходе алгоритма встречаются 3 ситуации:

- метка ресурса совпадает с меткой пользователя;

- метка ресурса доминирует над меткой пользователя;

- метка пользователя доминирует над меткой ресурса.

В случае совпадения метки пользователя и метки ресурса пользователь получает к ресурсу полный доступ, то есть права на запись, чтение, печать и удаление что соответствует [3]. Поэтому во всех блоках выбора успешности действия если действие (запись, чтение, печать и удаление) успешно то в случае «да» выполняется запись в отчет «выполненное разрешение» и при всех «нет» не выполненное разрешение соответственно. Выполненных или не выполненных запретов в этом случае быть не может. (Рисунок 17)

В случае доминирования метки ресурса над меткой пользователя, пользователь имеет право только на запись, это связано с тем, что из информация которую подготавливают пользователи, не имеющие высокого уровня конфиденциальности, может составлять часть ресурсов, имеющих более высокий уровень. Поэтому запись в отчет «Выполненное разрешение» осуществляется только если пользователь успешно совершил запись. Во всех остальных случаях записывается толь либо «Выполненный запрет» при отказе в совершенных действиях, либо не «Выполненный запрет» в противоположной ситуации. (Рисунок 18)

В случае доминирования метки пользователя над меткой ресурса, пользователь получает только права на чтение. Это предотвращает понижение уровня конфиденциальности ресурса, для которой это не приемлемо. Поэтому «Выполненное разрешение» будет записываться в отчет только при успешном чтении и печати, во всех остальных случаях речь будет идти о запретах. (Рисунок 19)

Для демонстрации возможности реализации алгоритма в разных операционных системах сопоставим команды интерпретаторов с основными его элементами. Для семейства Windows это будет «Powershell», для Unix-подобных «bash».

Таблица 6 – соответствие команд интерпретаторов элементам алгоритма

Элемент алгоритма	Командлет «Powershell»	Команда «bash»
Создание файла	«new-item»	«touch»
Запись	«out-file»	«echo > »
Чтение	«get-content»	«cat»
Печать	«out-printer»	«lpr»
Перемещение	«move-item»	«mv»
Копирование	«copy-item»	«cp»
Удаление	«remove-item»	«rm»
Создание дочернего процесса от имени пользователя	«start-process –credential»	«sudo –u»
Закрытие дочернего процесса	«stop-process»	«kill»

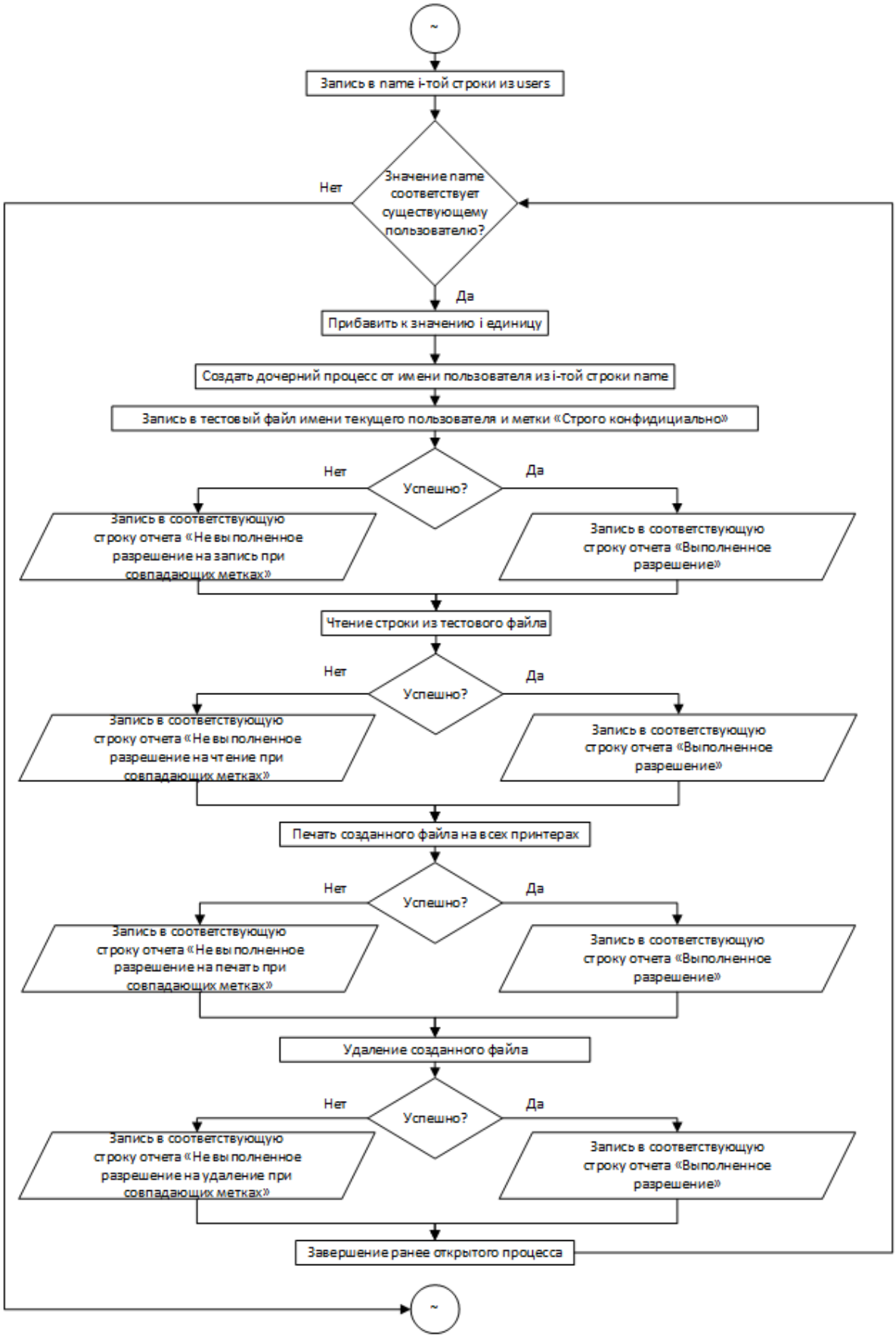


Рисунок 17 – Фрагмент блок-схемы алгоритма, демонстрирующий проверку прав доступа в случае совпадения меток пользователя и ресурса.

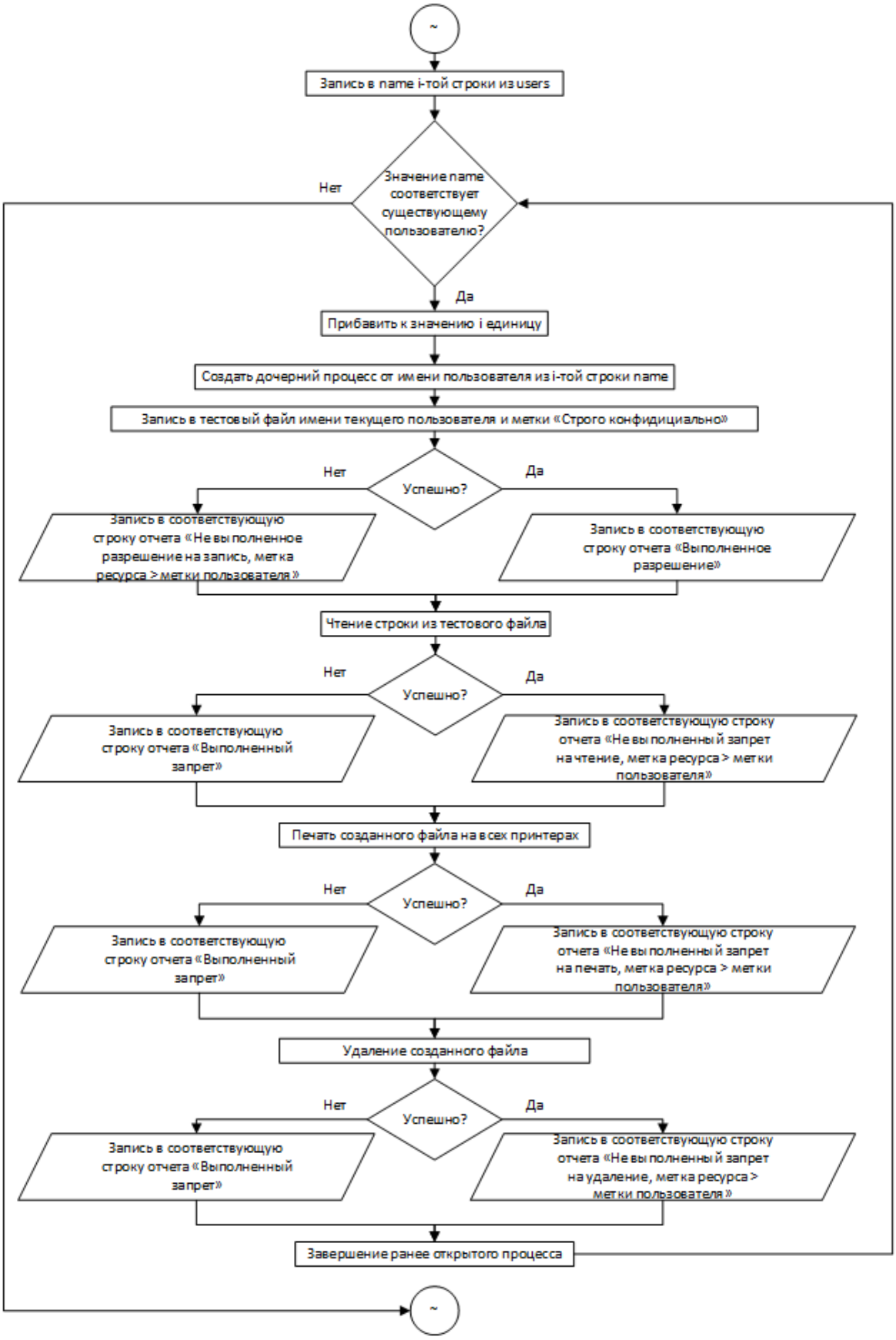


Рисунок 18 – Фрагмент блок-схемы алгоритма, демонстрирующий проверку прав доступа в случае доминирования метки ресурса над меткой пользователя.

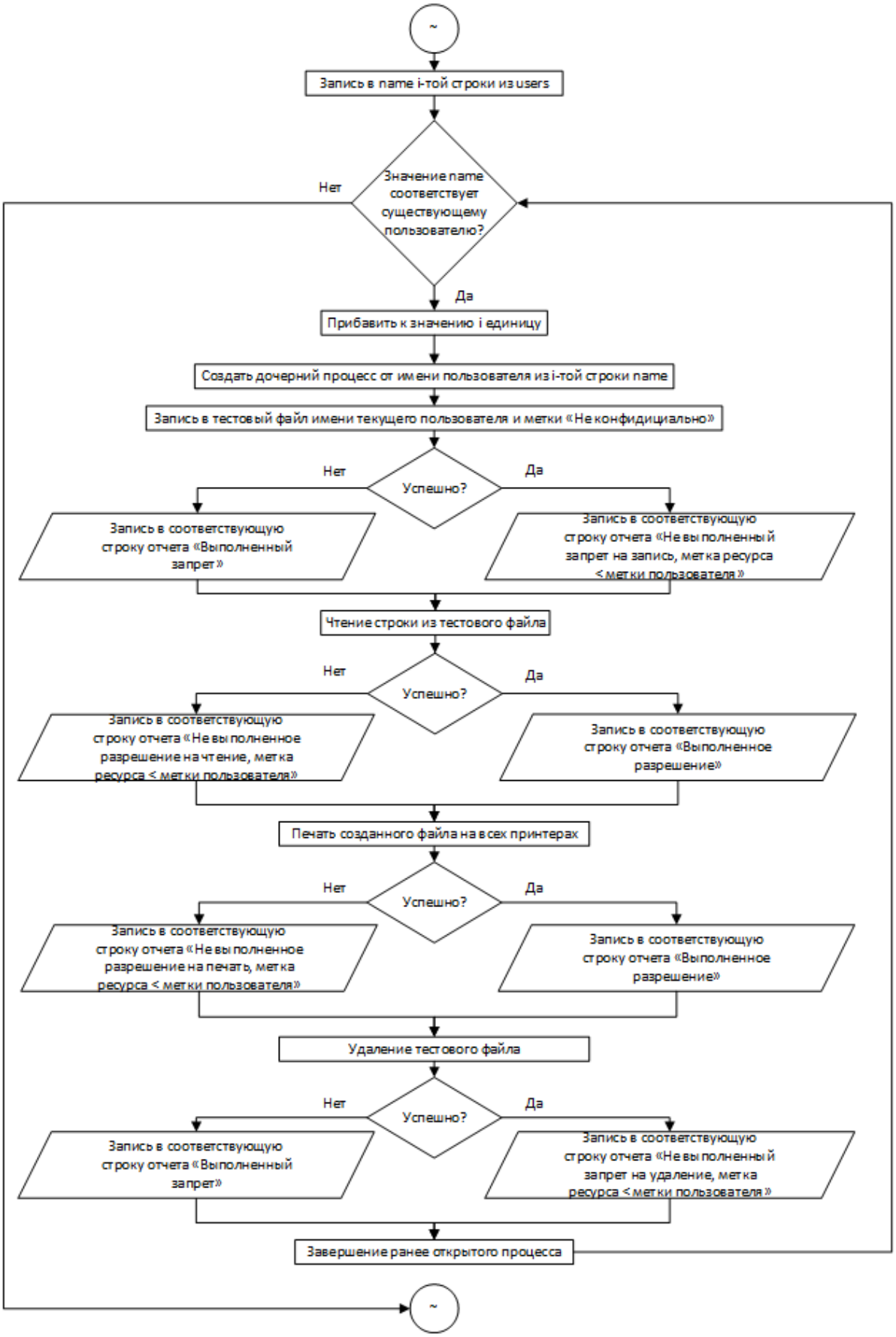


Рисунок 19 – Фрагмент блок-схемы алгоритма, демонстрирующий проверку прав доступа в случае доминирования метки пользователя.

4.1 Вывод по главе

Был разработан и описан алгоритм, реализующих контроль настройки мандатного разграничения доступа согласно требованиям, нормативно правовых актов. Были подробно разобраны основные «угловые» моменты, которые встретятся при работе реализации алгоритма, а также инициализирующие файлы и используемые переменные.

ЗАКЛЮЧЕНИЕ

Алгоритм программы, реализующей контроль разграничения доступа согласно требованиям, нормативно правовых актов и осуществляющей работу посредством средств операционной системы разработан и описан, что означает достижение поставленной к выпускной квалификационной работе цели. Реализация алгоритма подразумевает использование штатных средств операционной системы, а именно командных интерпретаторов, что наделяет ее повышенным доверием к конечному программному обеспечению, так как конечным средством является не исполняемый файл, а простой набор команд. Упор в разработке был сделан на то, что реализовать алгоритм можно как в семействе операционных систем «Windows», так и в Unix-подобных системах. Выше перечисленное определенно является весомыми достоинствами. Ограничениями на данный момент являются направленность алгоритма на средство защиты информации «Astra Linux» и только на 3 уровня конфиденциальности. Модернизация данного алгоритма может включать в себя ориентированность на большее количество средств защиты информации и уровней конфиденциальности, добавление оконного режима, где это возможно, а также добавление форматизированного файла отчета.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Критерии оценки безопасности компьютерных систем. – 1983 год [Электронный ресурс]. –Режим доступа: http://csrc.nist.gov/publications/history/dod85.pdf.

2. Руководящий документ. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации. — Утверждено решением председателя Государственной технической комиссии при Президенте Российской Федерации от 30 марта 1992 г. [Электронный ресурс]. Режим доступа:http://fstec.ru/component/attachments/download/297.

3. Руководящий документ. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации. — Утверждено решением председателя Государственной технической комиссии при Президенте Российской Федерации от 30 марта 1992 г. [Электронный ресурс]. — Режим доступа: http://fstec.ru/normativnye-i-metodicheskie-dokumenty-tzi/114- deyatelnost/ tekushchaya/ tekhnicheskaya-zashchita-informatsii/ normativnye- i-metodicheskie-dokumenty/ spetsialnye-normativnye-dokumenty /384-rukovodyashchij-dokument-reshenie-predsedatelya – gostekhkomissii - rossii-ot-30 -marta-1992-g.

4. Руководящий документ. Концепция защиты средств вычислительной техники и автоматизированных систем от несанкционированного доступа к информации. — Утверждена решением Государственной технической комиссии при Президенте Российской Федерации от 30 марта 1992 г. [Электронный ресурс]. – Режим доступа:http://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty/114-spetsialnye -normativnye-dokumenty/387-rukovodyashchij-dokument-reshenie-predsedatelya-gostekhkomissii-rossii-ot-30-marta-1992-g4.

5. Приказ ФСТЭК России от 18.02.2013 N 21 (ред. от 23.03.2017) "Об утверждении Состава и содержания организационных и технических мер по обеспечению безопасности персональных данных. [Электронный ресурс]. – Режим досту-

па: http://fstec.ru/normotvorcheskaya/akty/53-prikazy/691-prikaz-fstek-rossii-ot-18-fevralya-2013-g-n-21.

6. Приказ ФСТЭК России от 11 февраля 2013 г. N 17 (ред. от 15.02.2017 г. N 27) «Об Утверждении требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах» [Электронный ресурс]. – Режим доступа: http://fstec.ru/ normotvorcheskaya/akty/53-prikazy/702-prikaz-fstek-rossii-ot-11-fevralya-2013-g-n-17

7. Исследование системы разграничения доступа на основе поведенческой модели пользователя. //Информационное противодействие угрозам терроризма: научно-практический журнал. Материалы XIV научно-практической конференции «Информационная безопасность – 2015». Таганрог, 4 – 7 июня 2015 г. – Таганрог: Изд. Южного федерального университета, 2015. – №24.

8. Руководство администратора. Настройка механизмов защиты. [Электронный ресурс]: https://www.securitycode.ru/upload/documentation/secret_net/Secret_Net_Admin_Guide_Security_Settings.pdf (Дата обращения 12.04.2017).

9. Система защиты информации от несанкционированного доступа «СТРАЖ NT» Версия 3.0. Описание применения. [Электронный ресурс] URL: http://guardnt.ru/download/doc/ app_guide_nt_3_0.pdf (дата обращения 13.04.2014).

10. Технические решения, применяемые для защиты от НСД в системах классов 3А и 2А. //Вестник УрФО. Безопасность в информационной сфере. — Челябинск: Изд. центр ЮУрГУ, 2014. — № 1(11).

11. Операционная система специального назначения «Astra Linux Special Edition». Описание применения РУСБ.10015-01 31 01 Листов 36. 2015 г.

12. ГОСТ 19.701-90 ИСО 5807-85 Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. М.: Издательство стандартов, 1992. — 23 с.

ПРИЛОЖЕНИЕ А

Инициализирующая часть.

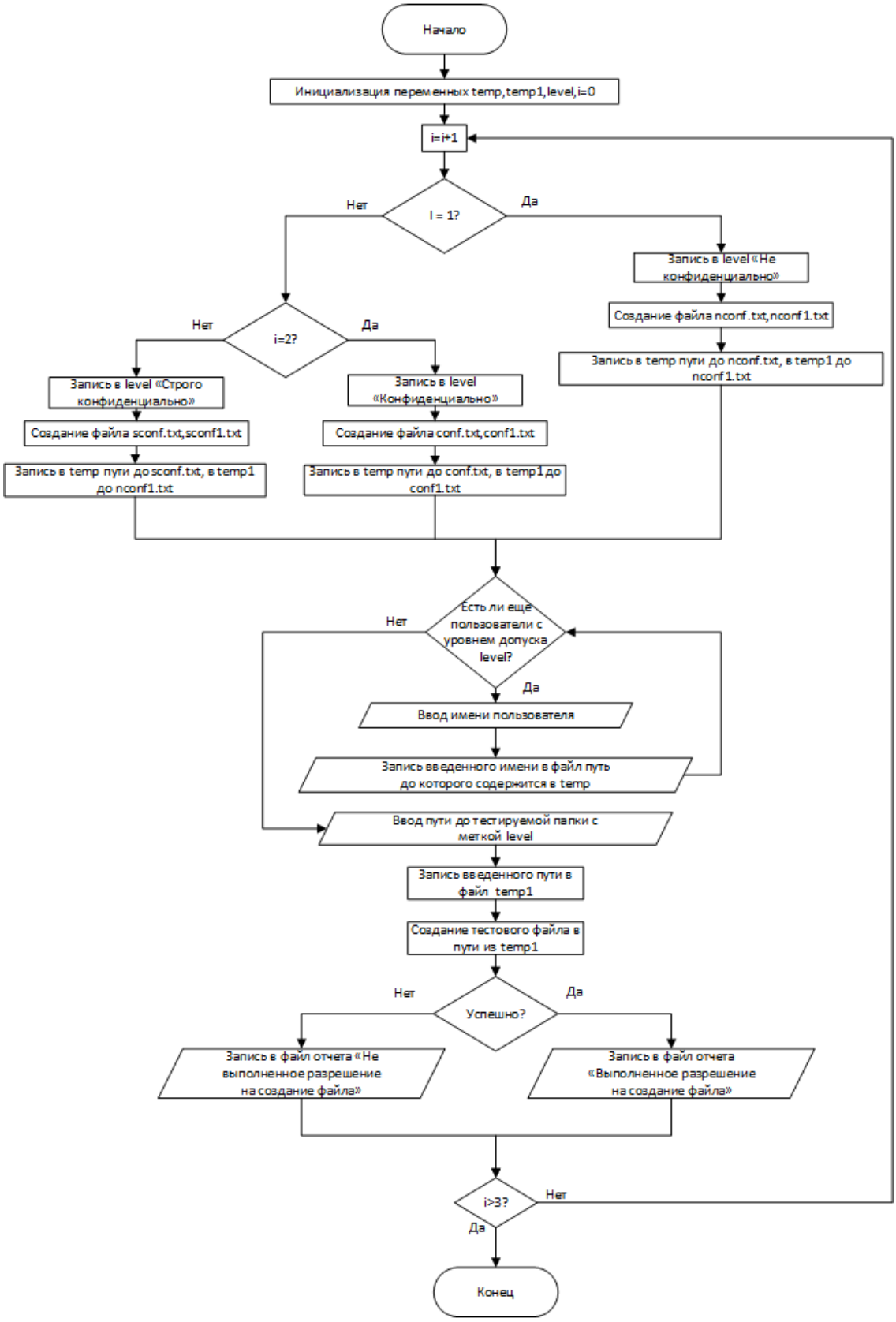


Рисунок А.1 – Блок-схема инициализирующей части алгоритма.

ПРИЛОЖЕНИЕ Б

Блок-схема схема части алгоритма, осуществляющей проверку.

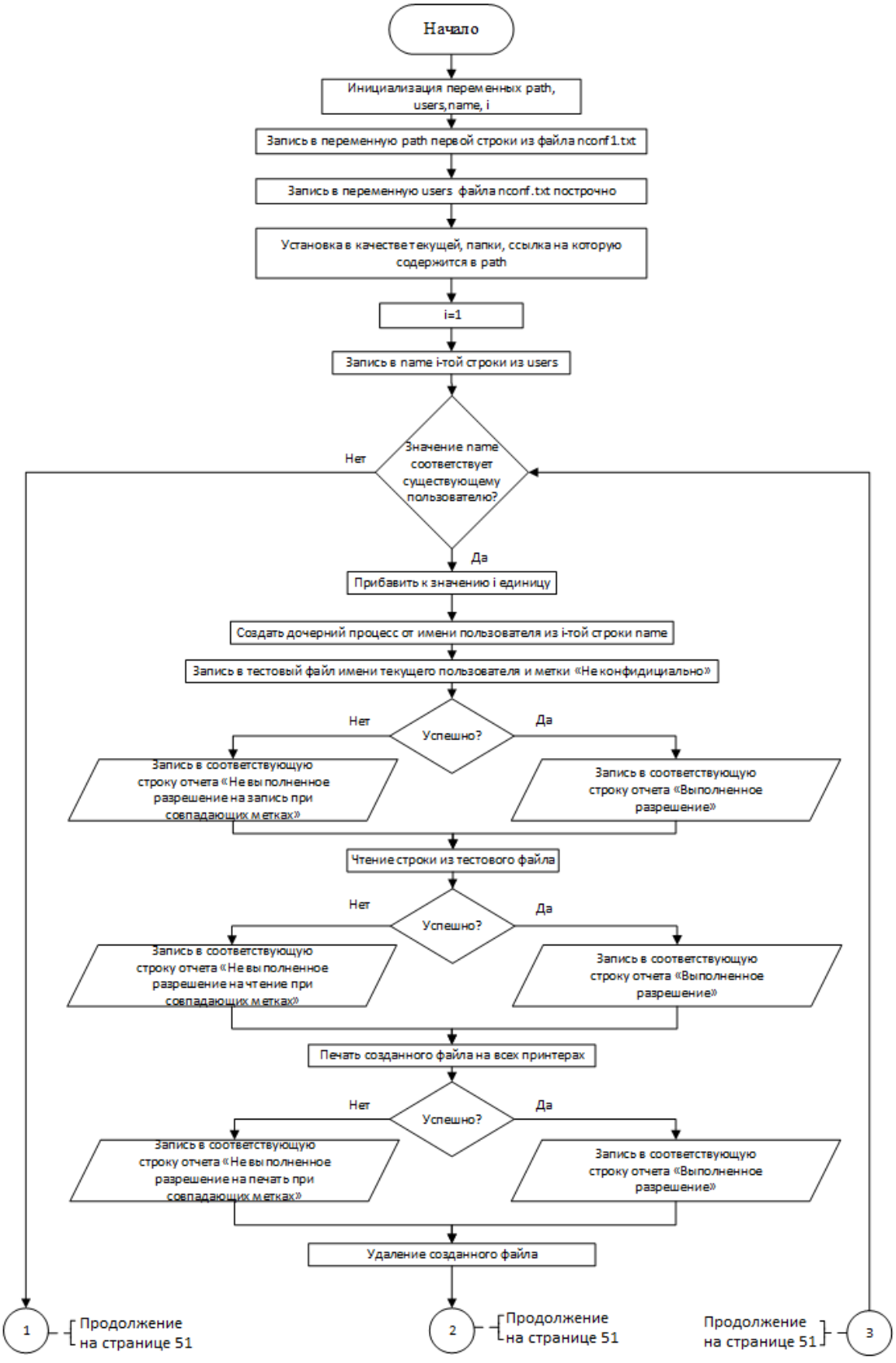
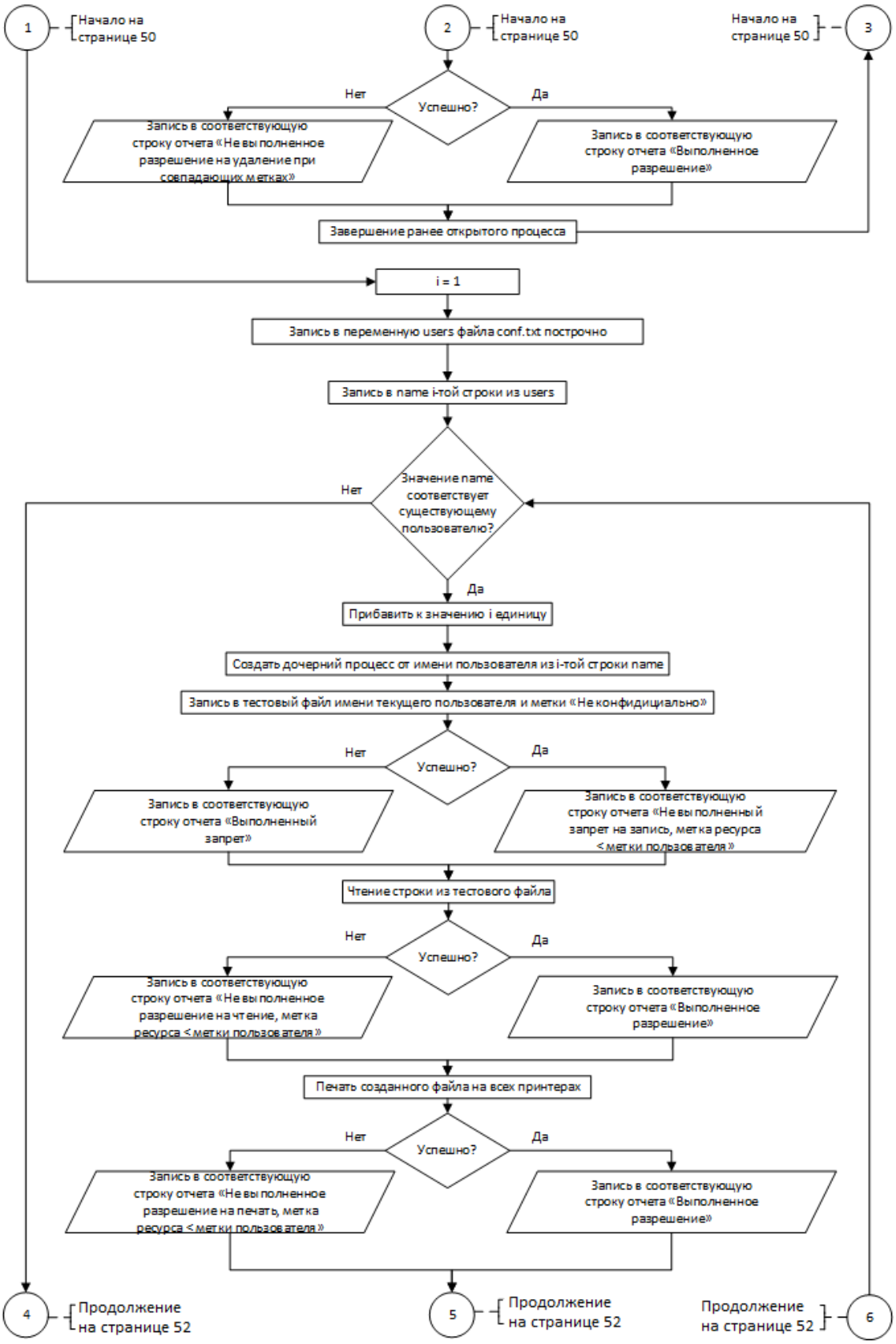
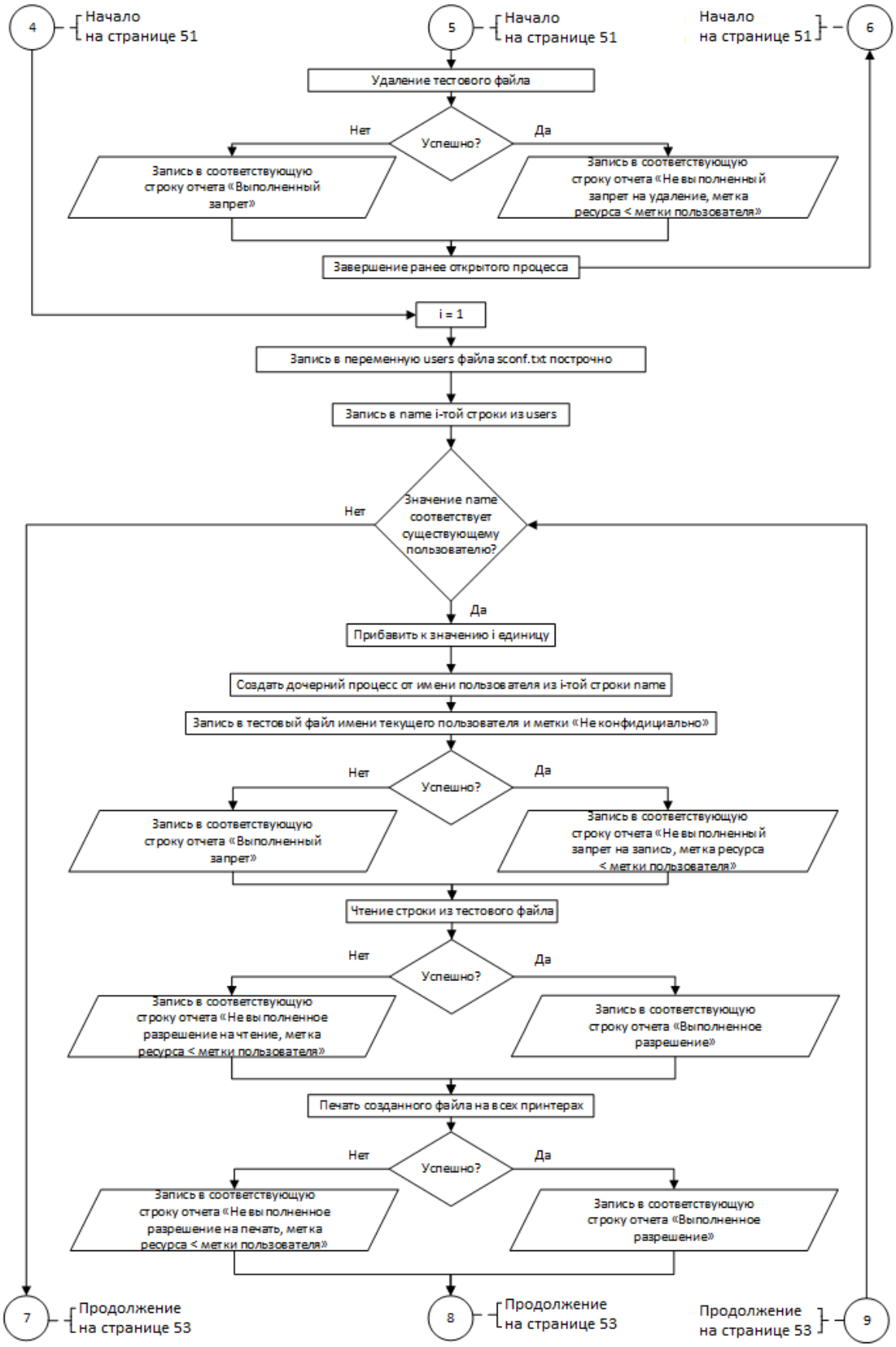


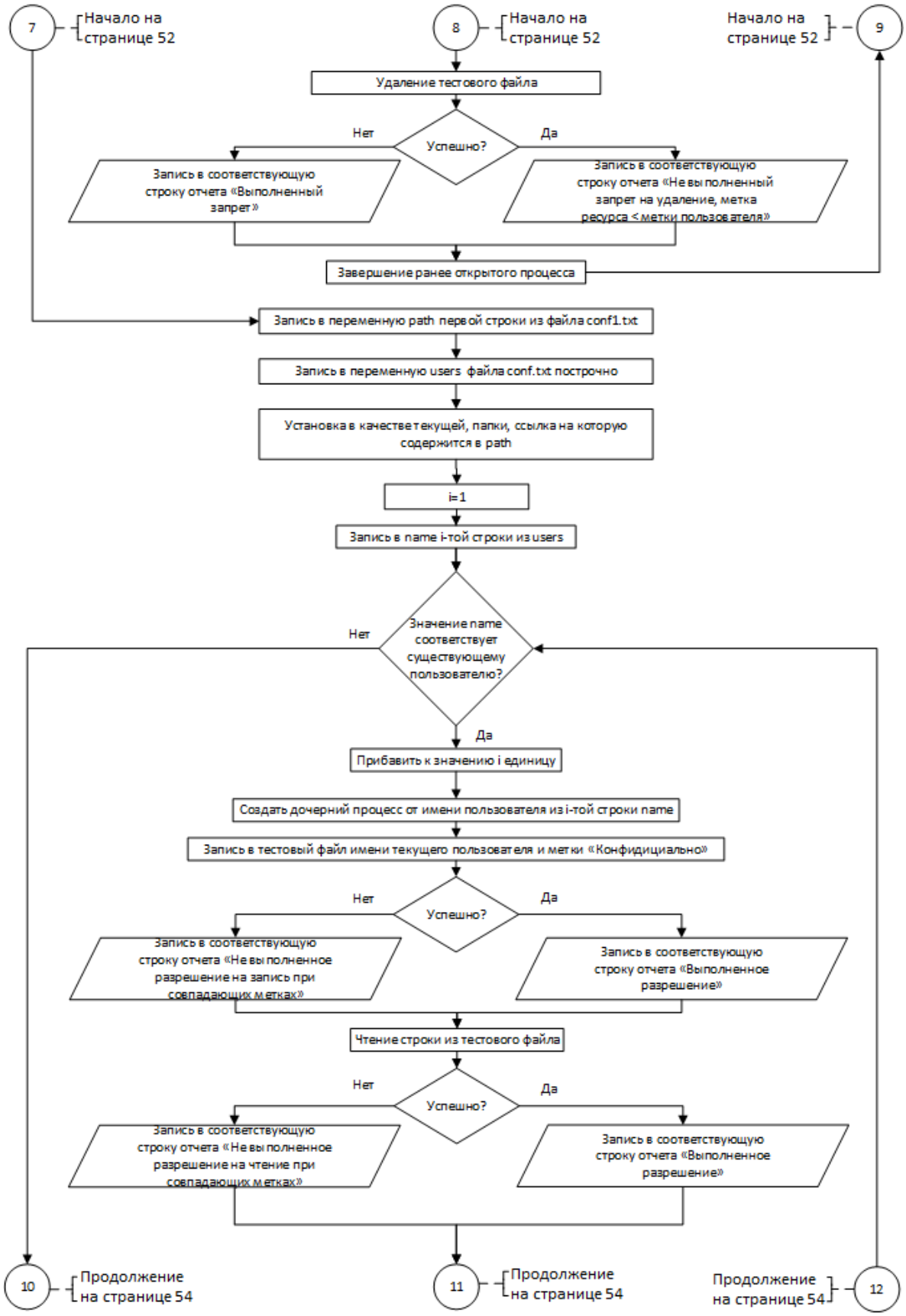
Рисунок Б.1 – Блок-схема части алгоритма, осуществляющего проверку.



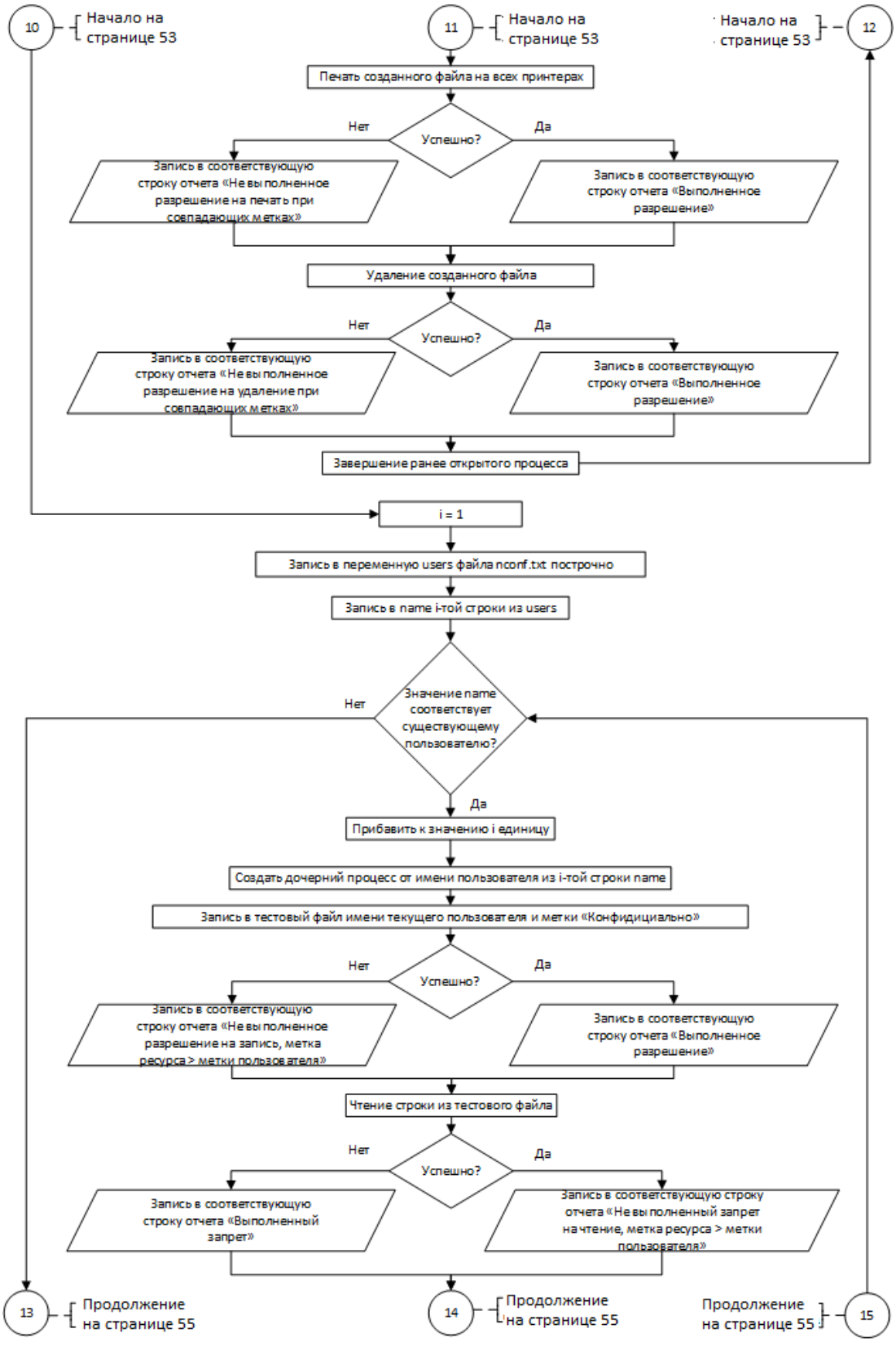
Продолжение рисунка Б.1



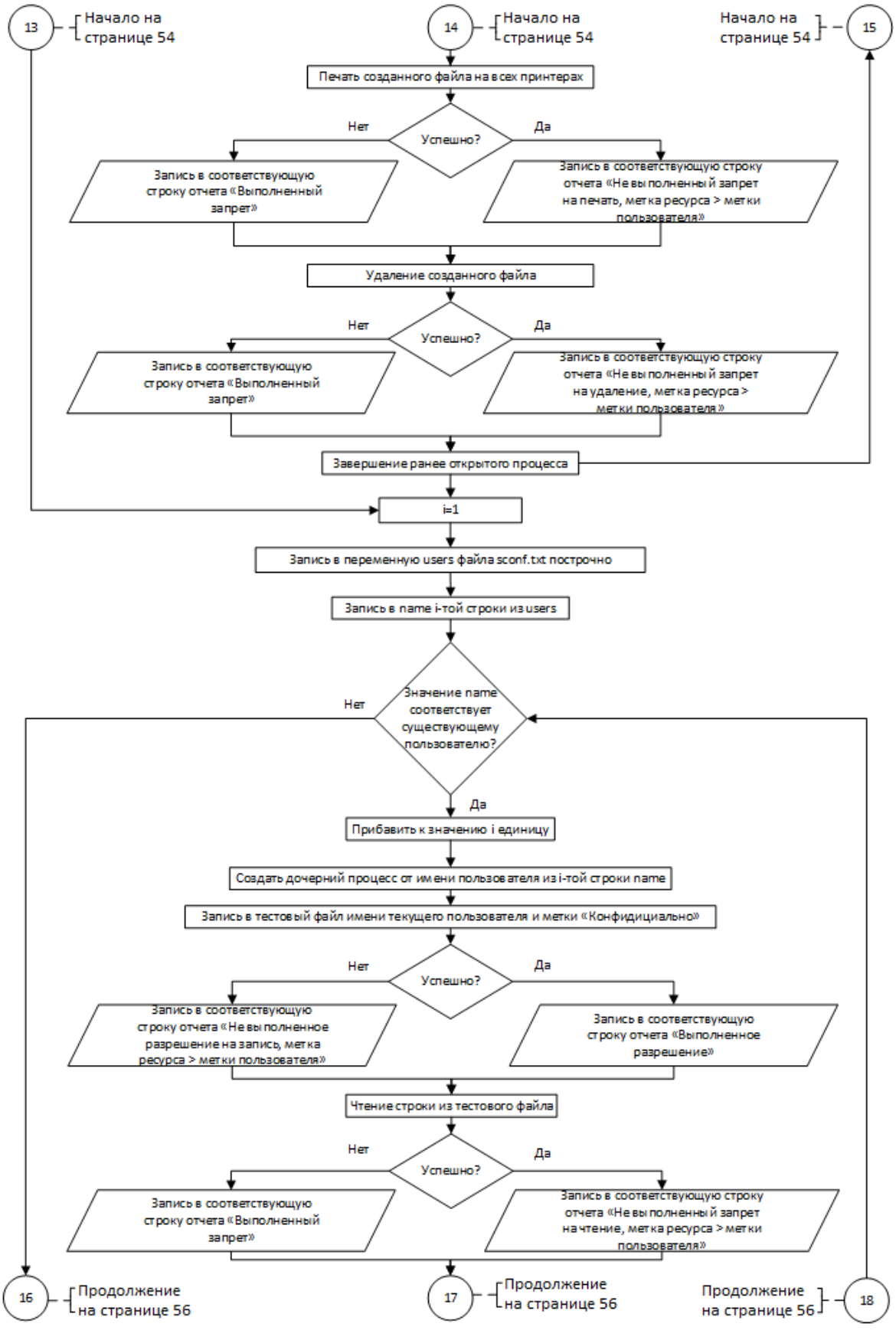
Продолжение рисунка Б.1



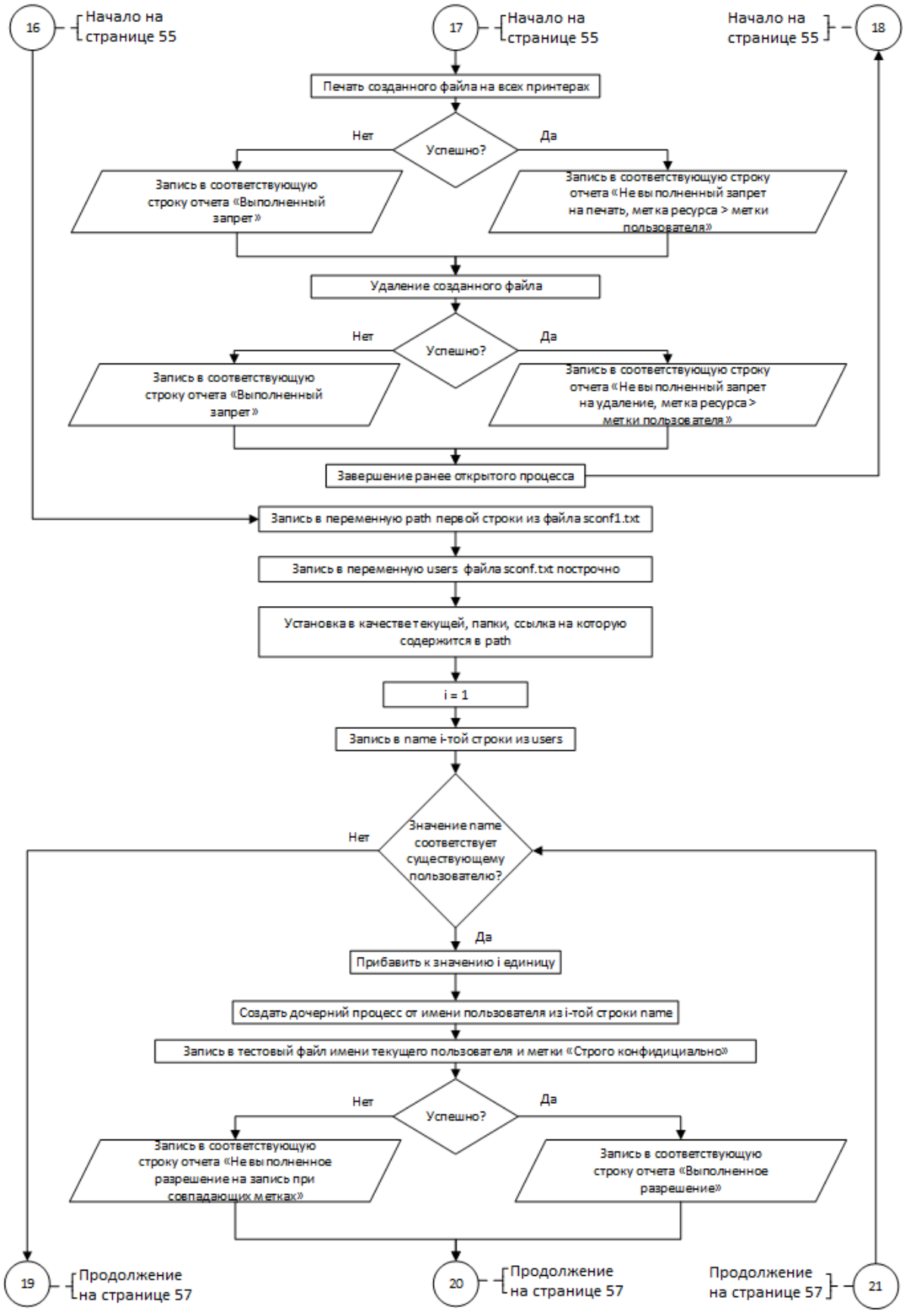
Продолжение рисунка Б.1.



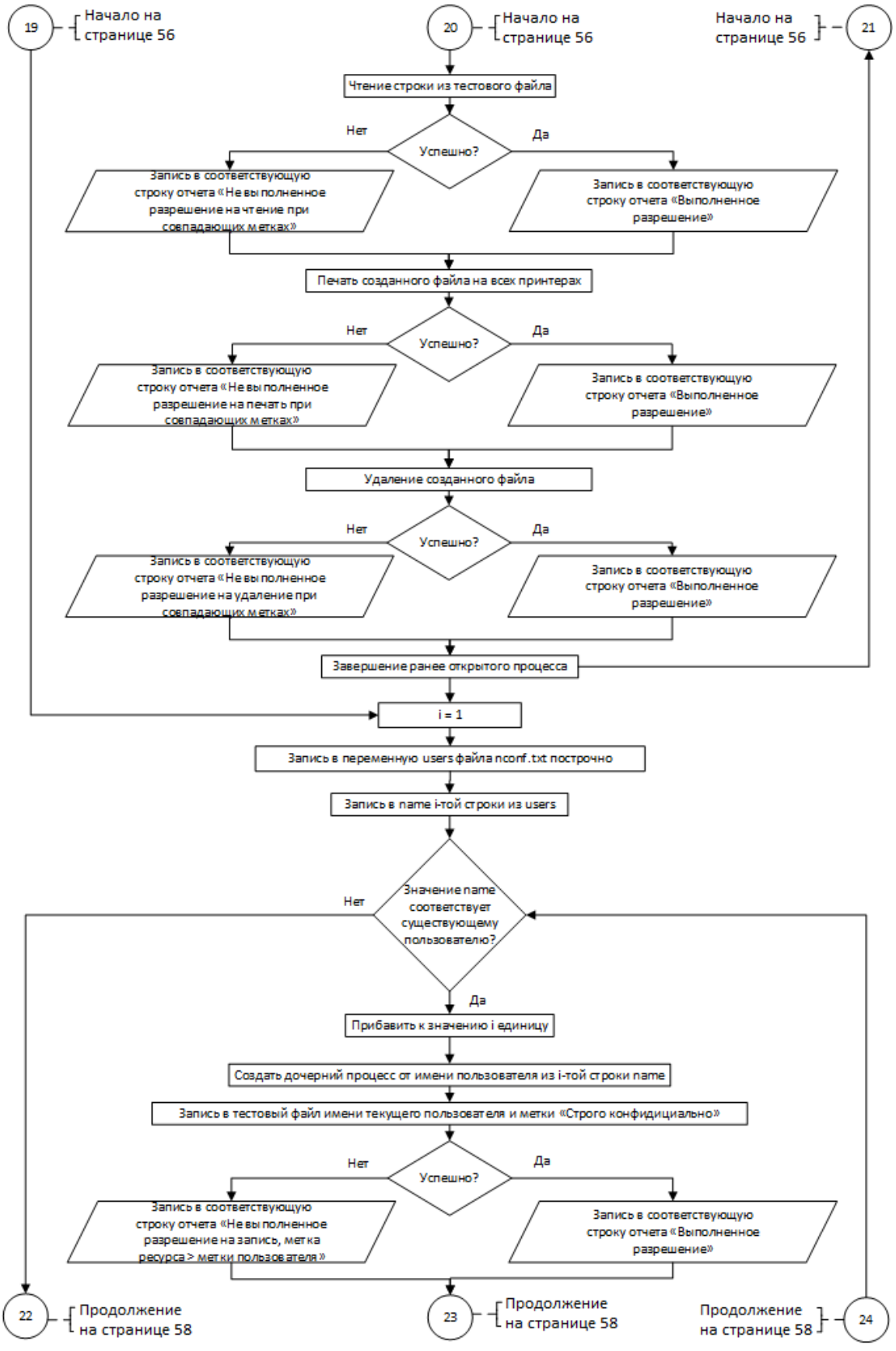
Продолжение рисунка Б.1.



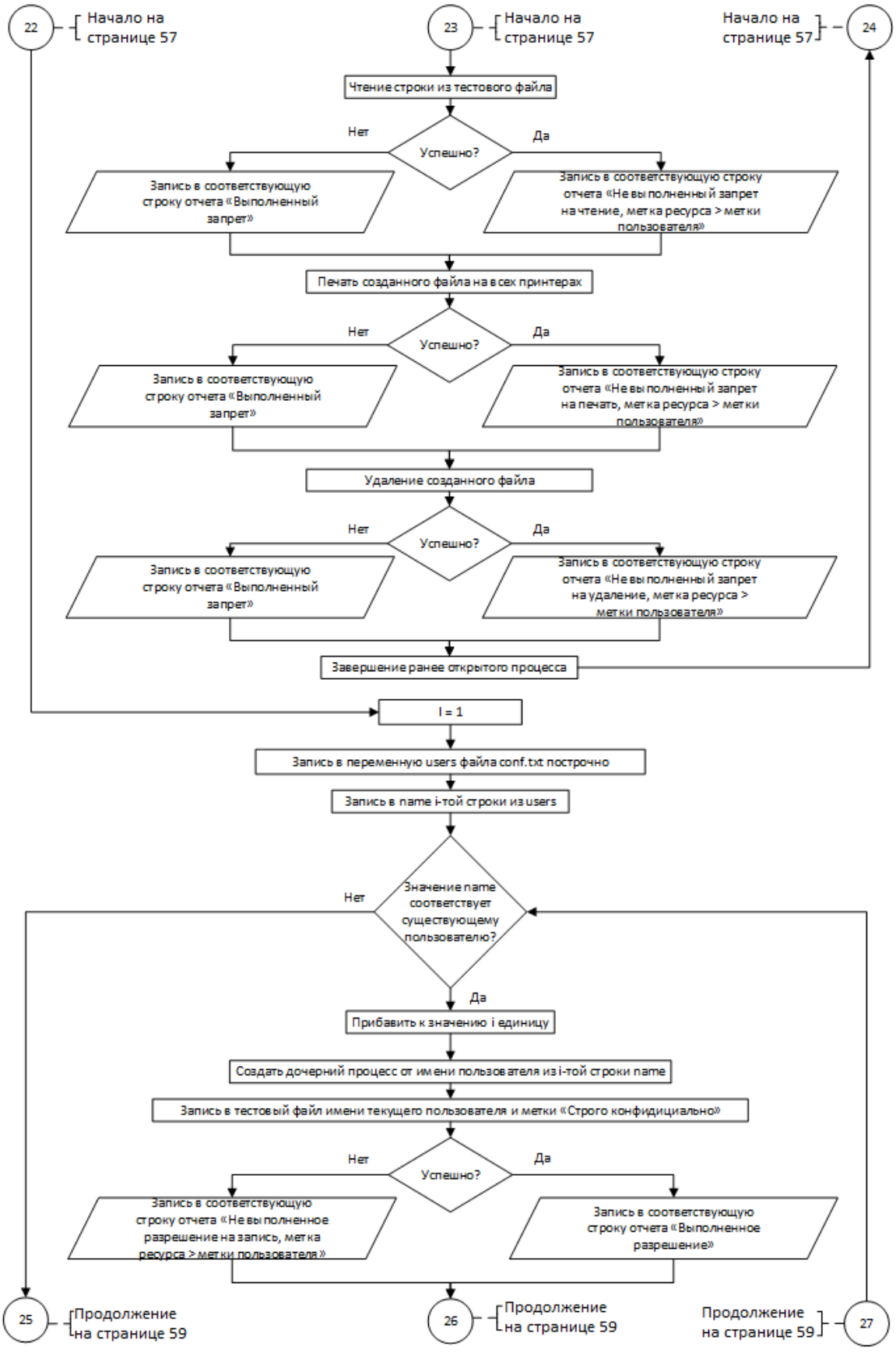
Продолжение рисунка Б.1.



Продолжение рисунка Б.1.

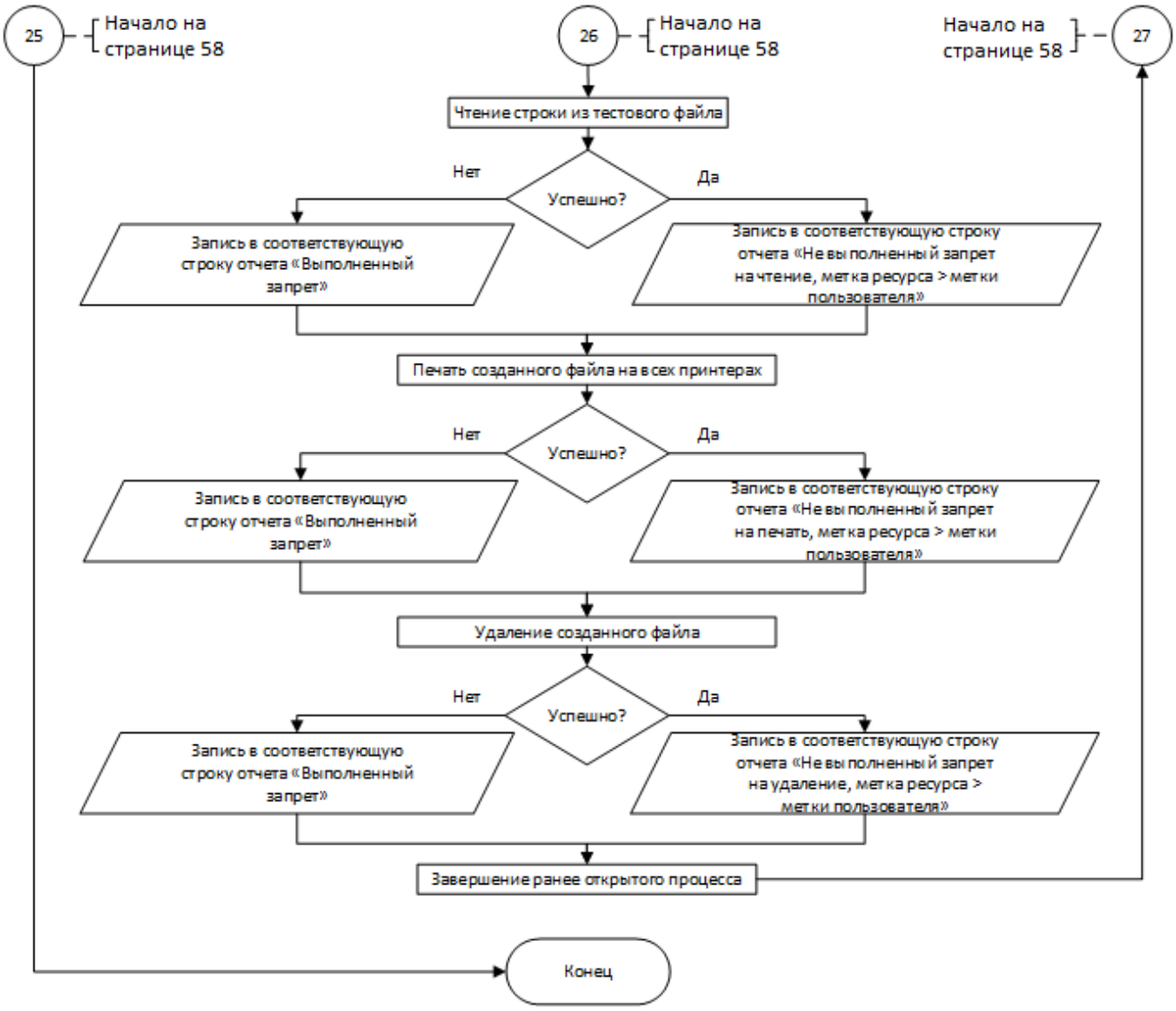


Продолжение рисунка Б.1.



Продолжение рисунка Б.1

Конец приложения Б.



Продолжение рисунка Б.1.

ПРИЛОЖЕНИЕ В

Структура инициализирующих файлов, используемых в алгоритмах, представленных в данной работе.

Файлы «nconf.txt», «conf.txt», «sconf.txt» предназначены для хранения списка пользователей, имеющих различный уровень допуска. Структура этих файлов аналогична и различия заключаются только в названии и собственно самом списке пользователей, в связи с этим, структура будет представлена на примере одного файла. Для демонстрации был выбран файл «nconf.txt». (Рисунок В.1)

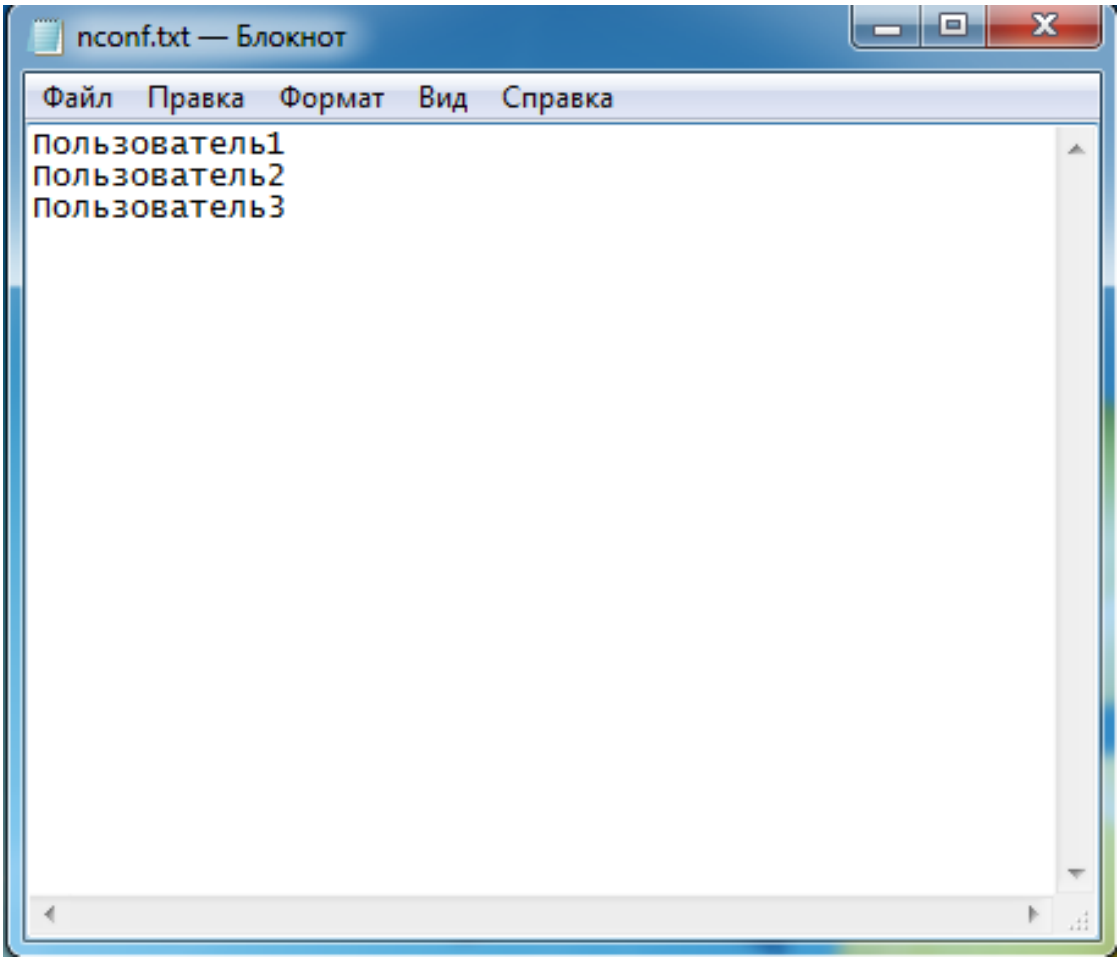


Рисунок В.1 – Структура файла «nconf.txt»

Конец приложения В.

Файлы для хранения пути до ресурсов, имеющих различные метки конфиденциальности. Структура этих файлов аналогична и различия заключаются только в названии и собственно самом пути до ресурса, в связи с этим, структура будет представлена на примере одного файла. Для демонстрации был выбран файл «nconf1.txt». (Рисунок В.2)

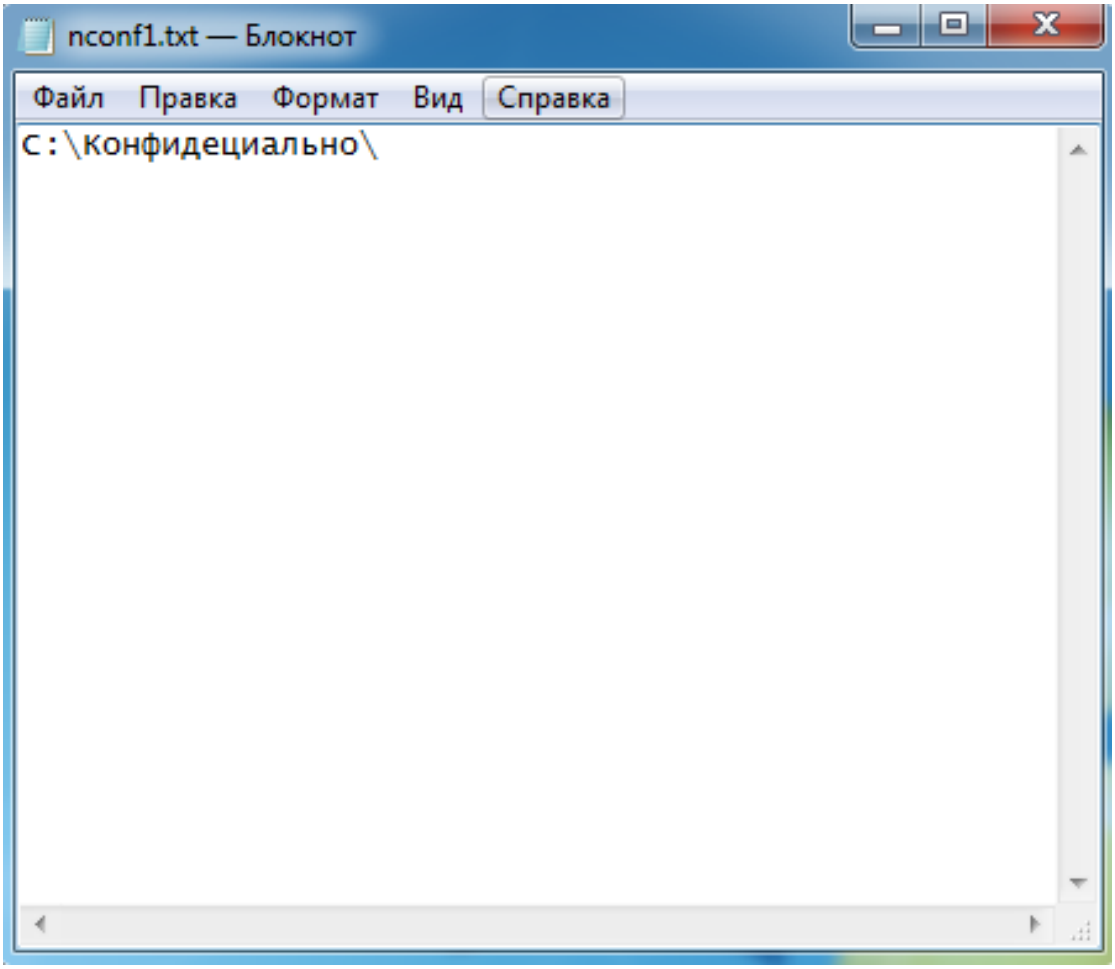


Рисунок В.2 – Структура файла «nconf1.txt»

ПРИЛОЖЕНИЕ Г

Листинг кода реализации инициализирующей части алгоритма в «bash»

```
#!/bin/bash
i=0
main() {
    i=$(($i+1))
    case $i in
        1)level='Не конфиденциально'
             touch ./nconf
            touch ./nconf1
            temp=./nconf
            temp1=./nconf1;;
        2)level='Конфиденциально'
             touch ./conf
             touch ./conf1
             temp=./conf
             temp1=./conf1;;
        3)level='Строго Конфиденциально'
            touch ./sconf
            touch ./sconf1
            temp=./sconf
            temp1=./sconf1;;
    esac
}

prov(){
    answ=1
    while [ $answ -eq 1 ] do
        echo 'Введите имя пользователя с уровнем допуска ' $level
        read a
        echo $a >> $temp
        echo 'Есть ли еще пользователи с таким же уровнем допуска? '
        read answ
    done
    echo 'Введите путь до папки имеющей метку конфиденциальности' $level
    read a
    echo $a >> $temp1
    read a < $temp
    read b < $temp1
    if sudo -u $a touch $b/test
        then echo 'Выполненное разрешение на создание файла для ' $a 'Уровень:' $level
        else echo 'Невыполненное разрешение на создание файла для ' $a 'Уровень:' $level
     fi
}

for c in 1 2 3 do
    main
    prov
done

exit 0;
```

ПРИЛОЖЕНИЕ Д

Листинг кода реализации проверяющей части алгоритма в «bash»

```
#!/bin/bash
temp=./nconf1
temp1=./nconf
read path < $temp
read users < $temp1
level='Не конфиденциально'
cd $path
echo ''
sudo -u $users /bin/bash - << usercodeblock
    if echo '$users $level' > $path/test
        then echo 'Выполненное разрешение на запись у '$users'уровень:'$level
        else echo 'Не выполненное разрешение на запись у '$users'уровень:'$level
    fi
    if read tst < $path/test
        then echo 'Выполненное разрешение на чтение у '$users'уровень:' $level
        else echo 'Не выполненное разрешение на чтение у '$users'уровень:'$level
    fi
    if lpr $path/test
        then echo 'Выполненное разрешение на печать у' $users 'уровень:' $level
        else echo 'Не выполненное разрешение на печать у'$users'уровень:'$level
    fi
    if rm $path/test
        then echo 'Выполненное разрешение на удаление у' $users'уровень:'$level
        touch /home/user1/nconf/test&echo $users 'уровень:'$level>/home/user1
/nconf/test
        else echo 'Не выполненное разрешение на удаление у'$users 'уровень:'$level
    fi
usercodeblock
echo ''
temp1=/home/user/conf
read users < $temp1
sudo -u $users /bin/bash - << usercodeblock
    if echo '$users $level' > $path/test
        then echo 'Невыполненный запрет на запись у'$users'уровень:'$level
        else echo 'Выполненный запрет на запись у' $users'уровень:'$level
    fi
    if read tst < $path/test
        then echo 'Выполненное разрешение на чтение у' $users 'уровень:'$level
        else echo 'Не выполненное разрешение на чтение у' $users'уровень:'$level

    fi
    if lpr $path/test
        then echo 'Выполненное разрешение на печать у'$users 'уровень:' $level
        else echo 'Не выполненное разрешение на печать у'$users'уровень:'$level
    fi
    if rm $path/test
        then echo 'Невыполненный запрет на удаление у'$users'уровень:' $level
        else echo 'Выполненный запрет на удаление у'$users'уровень:'$level
    fi
usercodeblock
sudo -u user1touch/home/user1/nconf/test & echo $users 'уровень: '$level> /home/ user1
/nconf/test
echo ''
```

```
temp1=/home/user/sconf
read users < $temp1
sudo -u $users /bin/bash - << usercodeblock
    if echo '$users $level' > $path/test
        then echo 'Невыполненный запрет на запись у'$users'уровень:'$level
        else echo 'Выполненный запрет на запись у' $users 'уровень:'$level
    fi
    if read tst < $path/test
        then echo 'Выполненное разрешение на чтение у' $users'уровень:'$level
        else echo 'Не выполненное разрешение на чтение у' $users 'уровень:' $level
    fi
    if lpr $path/test
        then echo 'Выполненное разрешение на печать у'$users'уровень:' $level
        else echo 'Не выполненное разрешение на печать у'$users'уровень:'$level
    fi
    if rm $path/test
        then echo 'Невыполненный запрет на удаление у'$users'уровень:'$level
        else echo 'Выполненный запрет на удаление у' $users 'уровень:'$level
    fi
usercodeblock
temp=/home/user/conf1
temp1=/home/user/conf
read path < $temp
read users < $temp1
level='Конфиденциально'
cd $path
echo ''
sudo -u $users /bin/bash - << usercodeblock
    if echo '$users $level' > $path/test
        then echo 'Выполненное разрешение на запись у'$users'уровень:'$level
        else echo 'Не выполненное разрешение на запись у'$users'уровень:'$level
    fi
    if read tst < $path/test
        then echo 'Выполненное разрешение на чтение у' $users 'уровень:'$level
        else echo 'Не выполненное разрешение на чтение у'$users'уровень:'$level
    fi
    if lpr $path/test
        then echo 'Выполненное разрешение на печать у'$users'уровень:'$level
        else echo 'Не выполненное разрешение на печать у'$users'уровень:'$level
    fi
    if rm $path/test
        then echo 'Выполненное разрешение на удаление у' $users'уровень:'$level
        touch /home/user2/conf/test & echo $users'уровень:'$level>/home/user2/conf/test
        else echo 'Не выполненное разрешение на удаление у'$users'уровень:'$level
    fi
usercodeblock
echo ''
temp1=/home/user/nconf
read users < $temp1
sudo -u $users /bin/bash - << usercodeblock
    if echo '$users $level' > $path/test
        then echo 'Выполненное разрешение на запись у'$users'уровень:'$level
        else echo 'Не выполненное разрешение на запись у'$users'уровень:'$level
    fi
    if read tst < $path/test
        then echo 'Не выполненный запрет на чтение у'$users'уровень:'$level
```

```
        else echo 'Выполненный запрет на чтение у' $users'уровень:'$level

    fi
    if lpr $path/test
        then echo 'Не выполненный запрет на печать у'$users'уровень:'$level
        else echo 'Выполненный запрет на печать у'$users'уровень:'$level
    fi
    if rm $path/test
        then echo 'Не выполненный запрет на удаление у' $users'уровень:'$level
        else echo 'Выполненный запрет на удаление у' $users'уровень:'$level
    fi
usercodeblock
sudo  -u  user2  touch  /home/user2/conf/test&echo  $users'уровень:'  $level>/home/user2/
conf/test
echo ''
temp1=/home/user/sconf
read users < $temp1
sudo -u $users /bin/bash - << usercodeblock
    if echo '$users $level' > $path/test
        then echo 'Невыполненный запрет на запись у'$users'уровень:'$level
        else echo 'Выполненный запрет на запись у' $users 'уровень:'$level
    fi
    if read tst < $path/test
        then echo 'Выполненное разрешение на чтение у'$users'уровень:'$level
        else echo 'Не выполненное разрешение на чтение у'$users'уровень:'$level

    fi
    if lpr $path/test
        then echo 'Выполненное разрешение на печать у'$users'уровень:'$level
        else echo 'Не выполненное разрешение на печать у'$users'уровень:'$level
    fi
    if rm $path/test
        then echo 'Невыполненный запрет на удаление у'$users'уровень:'$level
        else echo 'Выполненный запрет на удаление у'$users'уровень:'$level
    fi
usercodeblock
echo ''
temp=/home/user/sconf1
temp1=/home/user/sconf
read path < $temp
read users < $temp1
level='Строго конфиденциально'
cd $path
echo ''
sudo -u $users /bin/bash - << usercodeblock
    if echo '$users $level' > $path/test
        then echo 'Выполненное разрешение на запись у'$users'уровень:'$level
        else echo 'Не выполненное разрешение на запись у'$users'уровень:'$level
    fi
    if read tst < $path/test
        then echo 'Выполненное разрешение на чтение у'$users'уровень:'$level
        else echo 'Не выполненное разрешение на чтение у'$users'уровень:'$level

    fi
    if lpr $path/test
        then echo 'Выполненное разрешение на печать у'$users'уровень:'$level
        else echo 'Не выполненное разрешение на печать у'$users'уровень:'$level
    fi
```

```
if rm $path/test
       then echo 'Выполненное разрешение на удаление у'$users'уровень:'$level
       touch /home/user3/sconf/test & echo $users 'уровень:' $level>/home/user3/
    sconf/test
       else echo 'Не выполненное разрешение на удаление у'$users'уровень:' $level
    fi
usercodeblock
echo ''
temp1=/home/user/nconf
read users < $temp1
sudo -u $users /bin/bash - << usercodeblock
    if echo '$users $level' > $path/test
       then echo 'Выполненное разрешение на запись у'$users'уровень:'$level
       else echo 'Не выполненное разрешение на запись у'$users'уровень:'$level
    fi
    if read tst < $path/test
       then echo 'Не выполненный запрет на чтение у' $users 'уровень:' $level
       else echo 'Выполненный запрет на чтение у' $users 'уровень:' $level

    fi
    if lpr $path/test
       then echo 'Не выполненный запрет на печать у' $users'уровень:'$level
       else echo 'Выполненный запрет на печать у' $users'уровень:'$level
    fi
    if rm $path/test
       then echo 'Не выполненный запрет на удаление у'$users 'уровень:'$level
       else echo 'Выполненный запрет на удаление у' $users 'уровень:' $level
    fi
usercodeblock
sudo -u user3 touch /home/user3/sconf/test&echo $users 'уровень:'$level >/home/user3/
sconf/test
echo ''
temp1=/home/user/conf
read users < $temp1
sudo -u $users /bin/bash - << usercodeblock
    if echo '$users $level' > $path/test
       then echo 'Выполненное разрешение на запись у'$users'уровень:'$level
       else echo 'Не выполненное разрешение на запись у'$users'уровень:'$level
    fi
    if read tst < $path/test
       then echo 'Не выполненный запрет на чтение у'$users'уровень:'$level
       else echo 'Выполненный запрет на чтение у' $users 'уровень:'$level

    fi
    if lpr $path/test
       then echo 'Не выполненный запрет на печать у'$users'уровень:'$level
       else echo 'Выполненный запрет на печать у'$users'уровень: $level
    fi
    if rm $path/test
       then echo 'Не выполненный запрет на удаление у'$users'уровень:'$level
       else echo 'Выполненный запрет на удаление у' $users'уровень:'$level
    fi
usercodeblock

exit 0;
```