МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ
компьютерной безопасности и
криптографии

Восстановление удаленных файлов в ext3-4fs для случаев косвенной адресации области данных

АВТОРЕФЕРАТ
дипломной работы

студента 6 курса 631 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Зеляка Сергея Андреевича

Научный руководитель
к.ю.н., доцент _____ А. В. Гортинский
22.01.2022 г.

Заведующий кафедрой
д. ф.-м. н., доцент _____ М. Б. Абросимов
22.01.2022 г.

Саратов 2022

ВВЕДЕНИЕ

Количество хранимой и вырабатываемой во всём мире информации растёт с каждым годом. По прогнозам IDC глобальная сфера данных вырастет к 2025 году до 173 зеттабайт[1]. Мир уже вошёл в эпоху повсеместного использования больших данных и программное обеспечение подстраивается под сегодняшние реалии позволяя использовать всё более вместительные хранилища данных.

Семейство файловых систем Extended File System начало своё развитие с появлением первой версии в далёком 1992. Вдохновлённая структурами метаданных UFS ext является первой файловой системой для операционной системы Linux, которая расширяла возможности Minix File System добавив поддержку разделов и файлов размером до 2Гб, а также имён длиной до 255 символов. С тех пор появились 2, 3 и 4 версии данной системы каждая из которых привносила множество новых изменений, что позволяет ext оставаться актуальной и востребованной сегодня. На данный момент ext4 – четвёртая версия файловой системы является стандартной во многих дистрибутивах Linux и также используется в ОС Android.

Различия 2 и 3 версии системы ext не так велики и в основном заключаются в появлении журналирования. Однако, в 4 версии существенно изменились многие структуры, что в совокупности позволяет поддерживать разделы размером до 1 эксбибайта (2^{60} байт) и ускоряет работу с большими файлами[2].

Целью данной работы будет исследование возможностей восстановления удалённых файлов в двух версиях файловой системы ext3 и ext4, а также реализация на их основе алгоритмов восстановления удалённых файлов для случаев косвенной адресации области данных.

---

[1] Reinsel, D. The Digitization of the World. From Edge to Core [Электронный ресурс] / D. Reinsel // David Reinsel – URL: https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf (дата обращения 12.12.2021). – Загл. с экрана. – Яз. англ.

[2] Wong, D. Ext4 Data Structures and Algorithms [Электронный ресурс] / D. Wong // Darrick Wong – URL: https://www.kernel.org/doc/html/latest/filesystems/ext4/index.html (дата обращения 24.10.2021). – Загл. с экрана. – Яз. англ.

Дипломная работа состоит из введения, 5 разделов, заключения, списка использованных источников и 3 приложений. Общий объем работы – 56 страниц, из них 29 страниц – основное содержание, включая 21 рисунок и 5 таблиц, список использованных источников из 6 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

Первый раздел содержит описание общего строения, а также структурных компонентов файловой системы ext3. Простота и надёжность – идеи, стоявшие превыше всего при разработке файловой системы, благодаря чему все ассоциированные с файлом данные локализуются, а важные для ФС данные дублируются.

Ядром и основным элементом для хранения метаданных в ext является суперблок. Суперблок хранит следующую необходимую для работы информацию:

- размер блока файловой системы;
- размер индексного узла;
- общее количество блоков;
- и другие служебные данные.

Файловая система ext разбита на блоки одинаковой длины, а те в свою очередь объединены в группы блоков, каждая из которых имеет собственный набор метаданных:

- копия суперблока;
- битовая карта блоков данных;
- и другие служебные данные.

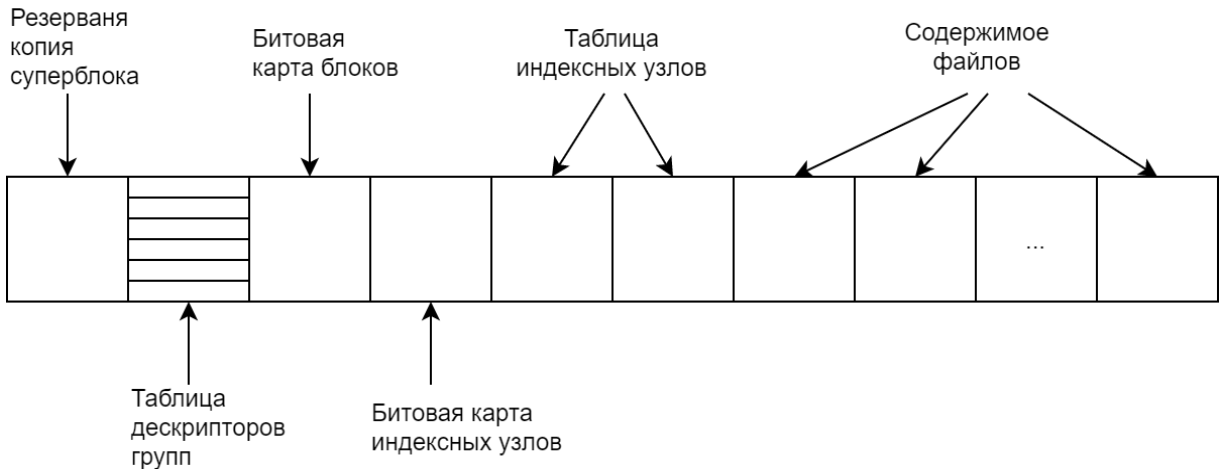

Рисунок 1 – Строение группы блоков

Для того чтобы определить местоположение приведённых выше служебных данных группы блоков необходимо прочитать таблицу дескрипторов групп, которая находится в области, следующей за суперблоком[3].

Метаданные файлов хранятся в структурах индексных узлов. При проектировании файловой системы ext3 упор был сделан на эффективную работу с небольшими файлами, поэтому каждый узел может хранить адреса первых 12 блоков, выделяемых файлу. Данные 12 блоков называются прямыми указателями. В тех случаях, когда чтобы сохранить файл необходимо большее количество блоков, выделяется специальный блок для хранения остальных адресов. Данный блок называется блоком косвенной адресации первого уровня.

Если размер файла настолько велик, что для его хранения недостаточно 12 прямых указателей и блока косвенной адресации первого уровня, используется двойная косвенная адресация. При этом в индексном узле содержится ссылка на блок косвенной адресации второго уровня.

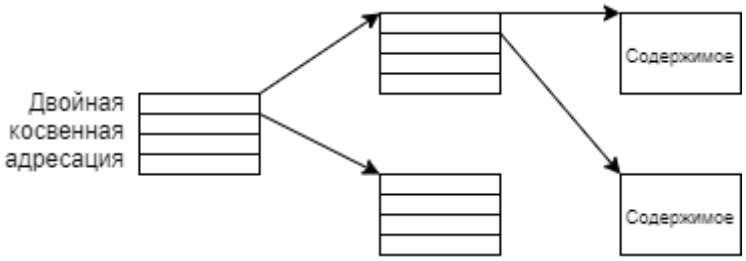

Рисунок 2 – Графическое представление двойной косвенной адресации

Если и таких мер недостаточно, то файловая система прибегает к механизму тройной косвенной адресации.

В расширенной файловой системе каталоги мало отличаются от обычного файла, помимо установленного в индексном узле специального значения. В выделенных под каталог блоках находится список особых структур данных с именем файла и адресом индексного узла, содержащего метаданные файла.

С 3 версии ext система стала журналируемой. В журнале регистрируются предстоящие изменения файловой системы, а после их завершения создаётся запись об успешной операции, в противном случае происходит откат изменений.

---

[3] Кэрриэ Б. «Криминалистический анализ файловых систем» / Б. Кэрриэ – СПб.: Питер,2007. – 480 с.: ил.

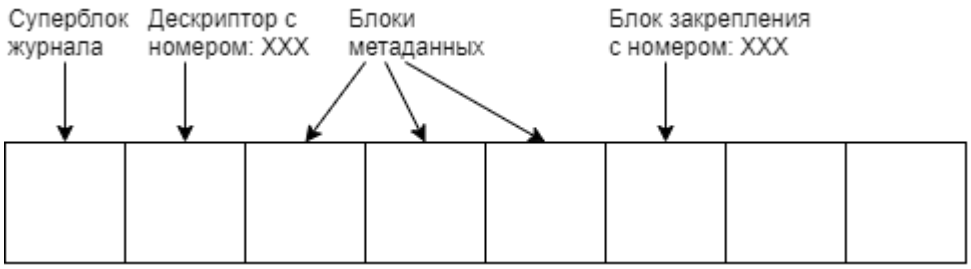

Рисунок 4 – Журнал файловой системы ext3 с одной транзакцией

Второй раздел посвящён строению файловой системы ext4 и описывает отличия 4 версии файловой системы от ext3. Главным нововведением ext4 является поддержка носителей больших размеров, что отражается в добавленных в каждую структуру дополнительных полях. Начиная с этой версии появилась новая структура – гибкие группы блоков, которые объединяют несколько групп блоков в одну логическую.

Главным изменением в индексных узлах стал новый механизм адресации блоков данных файла. Теперь все 60 байт отведённых под указатели могут быть использованы для хранения содержимого файла, если его размер не превышает размер этого поля. В иных случаях данное поле хранит дерево экстентов – ещё одну новую структуру ext4. Дерево экстентов описывает не перечень адресов блоков данных на диске, а последовательности блоков.

Как ясно из названия экстенты имеют древовидную структуру. При последовательной записи файла размером до 512Мб достаточно дерева с одним уровнем, которое полностью умещается в индексном узле, но для больших или сильно фрагментированных файлов может потребоваться большее количество экстентов. В таких случаях экстенты индексного узла ссылаются на блоки косвенной адресации, которые хранят последующие уровни дерева экстентов[2].

В подразделах третьего раздела приводится описание алгоритмов восстановления удалённых файлов в ext3. Начиная с 3 версии файловой системы ext при удалении файла индексный узел не только помечается как свободный, но

[2] Wong, D. Ext4 Data Structures and Algorithms [Электронный ресурс] / D. Wong // Darrick Wong – URL: https://www.kernel.org/doc/html/latest/filesystems/ext4/index.html (дата обращения 24.10.2021). – Загл. с экрана. – Яз. англ.

также происходит обнуление всех указателей на блоки прямой и косвенной адресации файла. Это накладывает ограничения на возможности восстановления удалённых файлов.

Первый подраздел описывает алгоритм восстановления файлов, основанный на поиске наиболее актуальной копии удалённых метаданных файла в журнале.

Данный способ восстановления файла является наиболее точным, так как доступны все те же данные, которые содержались в индексном узле до удаления. Однако, данный способ имеет весомый минус – восстановление необходимо проводить сразу же после удаления файла.

В следующем подразделе рассматривается механизм выделения места под новые файлы в ОС Linux, который старается размещать все данные файла в одной группе блоков файловой системы. В то же время в ext3 блоки косвенной адресации являются непрерывной последовательностью номеров блоков размером 4 байта, что позволяет построить алгоритм восстановления удалённых файлов с использованием блоков косвенной адресации.

Данный алгоритм проходится по блокам данных файловой системы и ищет в них паттерны, схожие с содержимым блоков косвенной адресации. Если блок с обнаруженным паттерном содержит блок косвенной адресации первого уровня, то будет возможно восстановить содержимое удалённого файла.

Такой способ восстановления гораздо менее точен по сравнению с использованием журнала, так как данные на носителе могут быть довольно хаотичными и невозможно полностью исключить ложные срабатывания[4].

Также, если блок содержал ссылки на блоки косвенной адресации 2 и 3 уровней, то невозможно точным образом определить взаимосвязанные части файлов.

---

[4] Pomeranz, H. EXT3 File Recovery via Indirect Blocks [Электронный ресурс] / H. Pomeranz // Hal Pomeranz – URL: http://www.deerrun.com/~hal/EXT3FileRecovery.pdf (дата обращения 15.11.2021). – Загл. с экрана. – Яз. англ.

Плюсом данного алгоритма является возможность восстановления частей удалённых файлов для упрощения дальнейшей ручной обработки.

В подразделах четвёртого раздела приводится описание алгоритмов восстановления удалённых файлов в ext4.

Первый подраздел отведён под восстановление удалённых файлов с использованием журнала, но так как журналирование в 4 версии ext не отличается от такого же в 3 версии, то алгоритм является идентичным и описывать его нет необходимости.

Во втором подразделе вновь указывается факт того, что в ext4 адресация данных файлов происходит не при помощи последовательности указателей на блоки, а с использованием дерева экстентов[5]. Обнаружить блоки с косвенной адресацией в таком случае гораздо легче, чем блоки косвенной адресации в ext3, так как заголовок экстента содержит специальную сигнатуру.

Это позволяет построить алгоритм, суть которого – поиск в блоках данных файловой системы сигнатуры экстента и последующее восстановление файла из него.

Такой алгоритм позволяет с большой точностью восстановить данные удалённых файлов. Основным же минусом является маленькая вероятность восстановления небольших файлов (размером менее 512Мб).

Пятый раздел содержит описание программы для восстановления удалённых файлов на носителях с файловой системой ext3 или ext4. Программа реализована на языке Go версии 1.17. Для описания структур файловых систем использовалась библиотека Kaitai Struct, позволяющая использовать декларативный синтаксис для описания бинарных структур. Программа имеет графический интерфейс, реализованный при помощи библиотеки Fyne.

---

[5] Stroll, S. Analysis of the process of deleting files in Linux [Электронный ресурс] / S. Stroll // Star Stroll – URL: https://www.fatalerrors.org/a/analysisof-the-process-of-deleting-files-in-linux.html (дата обращения 10.11.2021). – Загл. с экрана. – Яз. англ.

Для демонстрации работы программы использовались носители с ext3 и ext4, на которые были записаны по 3 файла размером 12 байт, 666Кб и 1Гб, после чего файлы удалялись. Разница в размерах этих файлов поможет лучше продемонстрировать разницу в работе различных алгоритмов восстановления удалённых файлов.

Для ext3 программа предлагает 3 алгоритма восстановления:

- восстановление с использованием журнала;
- восстановление с использованием блоков косвенной адресации первого уровня;
- восстановление частей файлов с использование блоков косвенной адресации любого уровня.

При поиске с использованием журнала программа смогла обнаружить все удалённые файлы. Восстановить удалось файлы размером в 12 байт и 666Кб. Файл размером 1Гб удалось восстановить только частично.

При поиске блоков косвенной адресации программа нашла и смогла успешно восстановить только файл размером 666Кб, остальные обнаруженные блоки косвенной адресации содержали только фрагменты и не могут быть использованы для полного восстановления файлов.

При поиске фрагментов удалённых файлов для их дальнейшей ручной обработки программа обнаружила и восстановила множество фрагментов, среди которых удалось идентифицировать фрагмент файла размером 666Кб.

Для ext4 предложен алгоритм восстановления с использованием блоков косвенной адресации. При использовании данного способа удалось обнаружить только блок косвенной адресации файла размером 1Гб, что позволило его успешно восстановить.

Анализ результатов работы программы на практике подтвердил описанные при разработке плюсы и минусы алгоритмов восстановления удалённых файлов на носителях с ext3 и ext4.

Листинг модулей программы приведён в приложениях А, Б и В.

ЗАКЛЮЧЕНИЕ

Общее количество информации в мире непрерывно растёт, и рост этот ускоряется. Однако, стоимость преднамеренной или случайной потери не становится меньше, поэтому так важно исследовать возможности восстановления утерянных файлов.

В данной работе было проведено исследование строения файловых систем ext3-4 на основе чего были построены алгоритмы позволяющие при различных условиях восстановить информацию, а также были исследованы их плюсы и минусы.

В ходе работы на языке Go была реализована программа позволяющая восстанавливать удалённые файлы в файловых системах ext3-4 в случаях косвенной адресации блоков данных.

Таким образом, все поставленные задачи полностью решены, а, следовательно, цель дипломной работы достигнута.