

Еремина Е.А.

студент

РГУ нефти и газа (НИУ) имени И.М. Губкина

Простова А.Н.

студент

РГУ нефти и газа (НИУ) имени И.М. Губкина

ВОПРОСЫ БЕЗОПАСНОСТИ REFS

***Аннотация:** Исследование посвящено оценке механизмов целостности и отказоустойчивости файловой системы ReFS в условиях имитации сбоя носителя. Анализируются возможности встроенных средств защиты данных, включая контрольные суммы, автоматическое восстановление и зеркалирование на уровне пула хранения. Эксперименты проведены в среде Windows Server 2022 с использованием PowerShell, DiskPart для развертывания виртуальных дисков, создания зеркального тома ReFS и последующего моделирования отказа физического диска. Выявлена эффективность работы функции File Integrity и механизма Storage Spaces при сохранении доступности данных.*

***Ключевые слова:** ReFS, целостность данных, отказоустойчивость, зеркалирование, Storage Spaces, виртуальный диск, File Integrity, Windows Server, пул хранения, PowerShell, DiskPart, безопасность данных.*

Eremina E.A.

student

Gubkin Russian State University (NRU) of Oil and Gas

Prostova A.N.

student

REFS SECURITY ISSUES

Abstract: *The study is devoted to evaluating the mechanisms of integrity and fault tolerance of the ReFS file system in conditions of simulated media failure. The capabilities of the built-in data protection tools, including checksums, automatic recovery and mirroring at the storage pool level, are analyzed. The experiments were conducted in a Windows Server 2022 environment using PowerShell, DiskPart to deploy virtual disks, create a mirrored ReFS volume, and then simulate a physical disk failure. The efficiency of the File Integrity function and the Storage Spaces mechanism while maintaining data availability has been revealed.*

Keywords: *ReFS, Data Integrity, Fault Tolerance, Mirroring, Storage Spaces, Virtual Disk, File Integrity, Windows Server, Storage Pool, PowerShell, DiskPart, Data Security.*

Введение

Актуальность проблемы обеспечения целостности и доступности данных в современных условиях многократно возрастает в связи с распространением сложных угроз информационной безопасности (APT-атаки, атаки «нулевого дня» и т.д.). В сфере информационной безопасности особое внимание уделено этой проблеме, так как нарушение целостности часто оказывается более разрушительным, чем нарушение конфиденциальности. Повреждение или удаление критически важных данных может привести к остановке работы целой организации.

Традиционные файловые системы (например, NTFS) были разработаны в конце XX века, когда основную угрозу составляли случайные сбои оборудования, а не кибератаки. Их архитектура основана на реактивной модели, то есть повреждения обнаруживаются постфактум, но в современных условиях такой подход неэффективен.

Объект исследования: механизмы обеспечения целостности и доступности данных в файловых системах Windows.

Предмет исследования: средства безопасности и алгоритмы восстановления данных после сбоев файловой системы ReFS в операционной среде Windows 11.

Цель исследования: проведение экспериментального анализа устойчивости файловой системы ReFS к нарушениям целостности данных и оценка функции File Integrity и механизма Storage Spaces для сохранения и восстановления данных.

Файловая система (file system) – описание способа хранения, распределения, наименования и обеспечения доступа к информации, хранящейся на машинном носителе информации [5].

Resilient File System (ReFS) – это новейшая файловая система Microsoft, созданная для обеспечения целостности данных, устойчивости к повреждениям и максимальной доступности данных. Она решает несколько растущих проблем и закладывает основу для более современных технологических усовершенствований [7]. Файловая система ReFS создавалась с использованием исходных кодов NTFS. Ключевой причиной создания файловой системы ReFS стала принципиальная недостаточность возможностей традиционных файловых систем для работы в условиях современных нагрузок и угроз. NTFS столкнулась с проблемами при управлении эксабайтными массивами данных, где время проверки целостности после сбоя могло достигать неприемлемых величин [7]. Кроме того, архитектура NTFS была уязвима к «повреждению распространяющихся данных», когда незамеченная ошибка на физическом носителе могла привести к порче файла или метаданных, а в худшем случае – к краху всей файловой системы. ReFS была разработана именно для преодоления этих ограничений.

Рассмотрим ключевые преимущества ReFS, такие как устойчивость, производительность и масштабируемость.

ReFS имеет функции, которые могут точно обнаруживать повреждения, а также устранять эти повреждения, оставаясь в сети, помогая обеспечить повышенную целостность и доступность данных. ReFS использует

контрольные суммы для метаданных, что позволяет ReFS надежно обнаруживать повреждения. При использовании зеркального пространства ReFS может автоматически восстанавливать обнаруженные повреждения с помощью альтернативной копии данных, предоставляемых дисковыми пространствами. Процессы восстановления локализованы в области повреждения и выполняются в сети, не требуя простоя тома.

Если том становится поврежден и копия поврежденных данных не существует, ReFS удаляет поврежденные данные из пространства имен. ReFS поддерживает том в рабочем состоянии, пока обрабатывает большинство неисправимых повреждений, но существуют редкие случаи, когда ReFS необходимо вывести том из сети.

Помимо проверки данных перед чтением и записью, ReFS представляет сканер целостности данных. Этот сканер периодически сканирует том, выявляя скрытые повреждения, и активирует восстановление поврежденных данных [2].

Файловая система ReFS также совместима с технологией Storage Spaces, которая позволяет объединять несколько физических дисков и применять зеркалирование [2].

ReFS предназначена для поддержки чрезвычайно больших наборов данных – миллионов терабайтов – без отрицательного влияния на производительность [1].

В таблице 1 представлены конкретные значения предельных значений различных атрибутов для файловой системы ReFS.

Таблица 1. Предельные значения для ReFS [1]

Атрибут	Предел для ReFS
Максимальный размер единого файла	$2^{64}-1$ байт
Максимальный размер единого тома	Формат поддерживает 2^{78} байт с размером кластеров 16 КБ ($2^{64} * 16 * 2^{10}$)
Максимальная длина имени файла	32 тысячи символов Unicode
Максимальный размер любого пула носителей	4 ПБ

Максимальное число пулов носителей в системе	Не ограничено
--	---------------

Обзор и методология исследования

В официальной документации Microsoft ReFS определяется как современная файловая система, разработанная для обеспечения целостности данных, устойчивости к повреждениям и высокой доступности через механизмы Integrity Streams совместно с Storage Spaces. Система позиционируется как решение для виртуализированных и высоконагруженных сред, поддерживающее масштабируемость до миллионов терабайт [3].

Существенный пласт исследований посвящен сравнительному анализу производительности файловых систем Windows. В работе Поддубной Я.С. и Чуреиной С.С. [6], проводится комплексное тестирование FAT32, NTFS и ReFS на базе Windows Server 2022. Авторы, используя утилиту CrystalDiskMark и скрипты PowerShell, оценивают скорость линейного и случайного доступа, а также эффективность базовых файловых операций (создание, копирование, перемещение, удаление).

Исследование Zöchling (2017) [4] показало, что ReFS без File Integrity уступает NTFS в производительности, однако при активации механизмов целостности ReFS демонстрирует устойчивость к повреждениям за счет увеличения размера операций ввода-вывода. Эксперимент с имитацией отказа диска проводился в контролируемой среде, что по методологии близко к нашему исследованию, где также используется виртуализация и программная имитация сбоев носителей.

В статье Д. Соловьева [7] подробно описаны архитектурные особенности ReFS, ее преимущества и ограничения по сравнению с NTFS.

Однако, несмотря на обширный теоретический анализ и сравнительные тесты производительности, представленные в литературе, исследований, моделирующих отказ носителя в изолированной среде с полной цепочкой действий от развертывания ReFS до проверки доступности данных, остается

недостаточно, что связано с относительной новизной файловой системы и ее недостаточным внедрением.

Тип исследования: эксперимент с тестированием на устойчивость и практическим моделированием нарушения целостности данных в изолированной лабораторной среде с Windows 11.

Характеристика среды исследования: Windows Server 2022 Standart.
Инструменты: PowerShell, DiskPart, Server Manager.

Процедура проведения эксперимента:

1. Развертывание и настройка изолированной лабораторной среды.
2. Создание пула хранения из нескольких виртуальных дисков и формирование в нем зеркального виртуального диска.
3. Инициализация виртуального диска, создание в нем раздела, отформатированного в файловую систему ReFS, активация функции проверки целостности данных File Integrity.
4. Подготовка текстового набора данных путем создания файлов определенного размера и заполнения их содержимым.
5. Имитация отказа физического носителя путем перевода одного из дисков в аварийное состояние.
6. Проверка доступности и целостности данных на томе после сбоя, наблюдение за реакцией файловой системы.
7. Сравнительный анализ и формирование выводов.

Методы обработки данных: количественный анализ целостности данных, качественный анализ поведения файловой системы, практическая оценка.

Основные моменты экспериментального исследования

Экспериментальная среда на базе Windows Server 2022 включала в себя:

1. Пул хранения Storage Spaces, состоящий из двух виртуальных дисков по 10 ГБ каждый.
2. Зеркальный виртуальный диск размером 8 ГБ, созданный в пуле хранения.

3. Том с файловой системой ReFS, на котором были подготовлены и размещены тестовые данные.

Перешли в утилиту diskpart и создали два виртуальных диска (disk1.vhd и disk2.vhd) размером по 10 ГБ каждый [8, стр. 157]:

```
DISKPART> create vdisk file=C:\VHD/disk1.vhd maximum=10240  
type=expandable
```

```
DISKPART> create vdisk file=C:\VHD/disk2.vhd maximum=10240  
type=expandable
```

Присоединили созданные виртуальные диски к системе с помощью:

```
DISKPART> attach vdisk
```

После этого диски стали видимыми в Windows как физические устройства. Выполнили инициализацию диска №1 с таблицей разделов GPT.

См. рис. 1.

```
PS C:\WINDOWS\system32> Get-Disk
```

Number	Friendly Name	Serial Number	HealthStatus	OperationalStatus	Total Size	Partition Style
1	Msft Virtu...		Healthy	Online	10 GB	GPT
2	Msft Virtu...		Healthy	Online	10 GB	RAW
0	VBOX HARDDISK	VB9972dd26-423878fe	Healthy	Online	50 GB	MBR

```
PS C:\WINDOWS\system32> Initialize-Disk -Number 1 -PartitionStyle GPT
```

Рисунок 1 – Инициализация диска

Создали пул хранения ReFSPool из доступных дисков, которые могут быть объединены, с помощью команд:

```
> $disks = Get-PhysicalDisk | Where-Object CanPool -eq $True
```

```
> New-StoragePool -FriendlyName "ReFSPool" -StorageSubSystem  
FriendlyName "Windows Storage" -PhysicalDisks $disks
```

На рисунке 2 представлен результат выполнения команд в Server Manager. Пул хранения ReFSPool был создан и содержит в себе два диска.

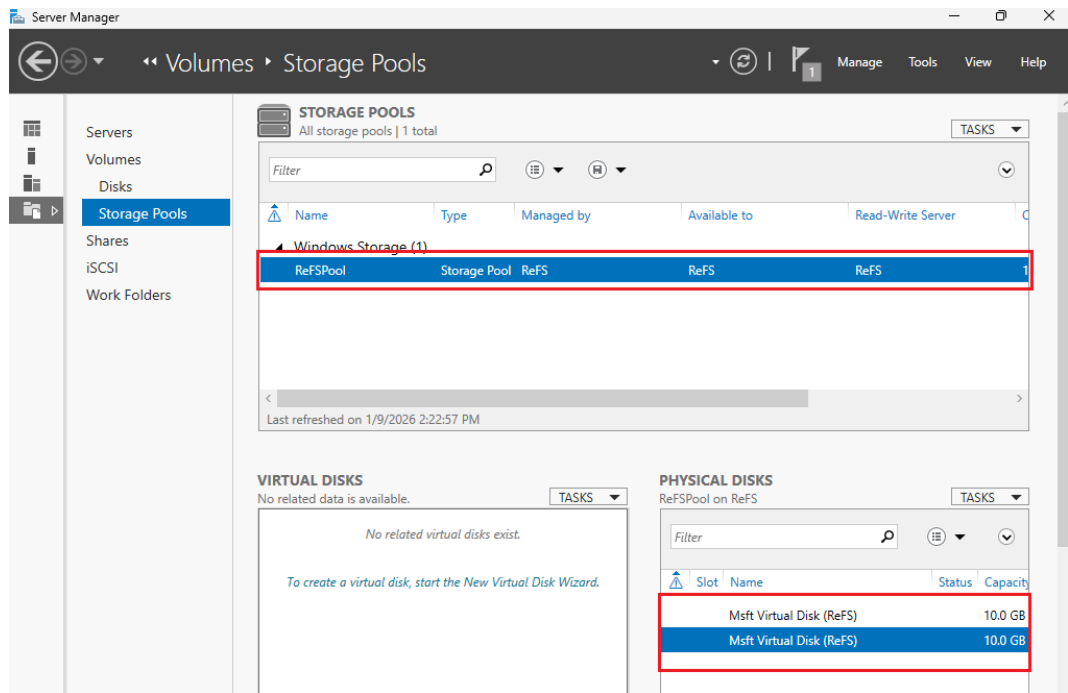


Рисунок 2 – Созданный пул хранения

Создали виртуальный диск ReFS_VDisk с зеркальной системой хранения размером 8 ГБ с помощью команды:

```
> New-VirtualDisk -StoragePoolFriendlyName "ReFSPool" -FriendlyName "ReFS_VDisk" -Size 8GB -ResiliencySettingName Mirror -ProvisioningType Fixed
```

На виртуальном диске создали раздел, форматировали его в файловую систему ReFS, присвоили букву E: и метку тома ReFS_Test:

```
> Get-VirtualDisk -FriendlyName "ReFS_VDisk" | Get-Disk | Initialize-Disk -PartitionStyle GPT
```

```
> Get-VirtualDisk -FriendlyName "ReFS_VDisk" |
  >> Get-Disk |
  >> New-Partition -UseMaximumSize -AssignDriveLetter E |
  >> Format-Volume -FileSystem ReFS -NewFileSystemLabel "ReFS_Test" -Confirm:$false
```

После выполнения команд том готов к использованию.

Включили функцию проверки целостности данных (FileIntegrity) для тома E:\. Изначально она была отключена, после выполнения команды Set-

FileIntegrity -Enable \$true – включена и принудительно применяется. См. рис. 3.

```
PS C:\WINDOWS\system32> Get-FileIntegrity E:\

FileName Enabled Enforced
-----
E:\      False   True

PS C:\WINDOWS\system32> Set-FileIntegrity E:\ -Enable $true
PS C:\WINDOWS\system32> Get-FileIntegrity E:\

FileName Enabled Enforced
-----
E:\      True    True
```

Рисунок 3 – Включение проверки целостности данных

В папке E:\TestData создали два тестовых файла по 10 МБ каждый (file1.bin, file2.bin) с помощью утилиты fsutil:

- > mkdir E:\TestData
- > fsutil file createnew E:\TestData\file1.bin 10485760
- > fsutil file createnew E:\TestData\file2.bin 10485760

После успешного создания заполнили файл file1.bin случайными данными для имитации реального содержимого с помощью команды:

- > 1..10000 | Get-Random -Count 10000 | Set-Content E:\TestData\file1.bin

Пометили физический диск №1 как Retired (выведенный из эксплуатации), имитируя его отказ. См. рис. 4.

```
PS C:\WINDOWS\system32> Get-PhysicalDisk -DeviceNumber 1 | Set-PhysicalDisk -Usage Retired
PS C:\WINDOWS\system32> Get-PhysicalDisk

Number FriendlyName      SerialNumber      MediaType      CanPool OperationalStatus HealthStatus Usage      Size
-----
2      Msft Virtual Disk      Unspecified      False         OK        Healthy      Auto-Select 10 GB
0      VBOX HARDDISK          VB9972dd26-423878fe Unspecified      False         OK        Healthy      Auto-Select 50 GB
1      Msft Virtual Disk      Unspecified      False         OK        Healthy      Retired     10 GB
```

Рисунок 4 – Имитация отказа диска.

В списке дисков видно, что диск №1 приобрел статус Retired.

После имитации отказа проверили наличие и размер тестовых файлов в папке E:\TestData. Оба файла присутствуют. Их размеры соответствуют исходным (10 МБ). Это говорит о том, что данные не были потеряны благодаря зеркалированию. См. рис. 5.

```
PS C:\WINDOWS\system32> Get-ChildItem E:\TestData

Directory: E:\TestData

Mode                LastWriteTime         Length Name
----                -
-a-----         1/9/2026   2:48 PM     10485760 file1.bin
-a-----         1/9/2026   2:34 PM     10485760 file2.bin
```

Рисунок 5 – Проверка целостности файлов

Результаты исследования

В ходе экспериментального исследования была успешно симитирована ситуация отказа одного из дисков в зеркальном пуле хранения с файловой системой ReFS. После перевода физического диска в состояние Retired было зафиксировано полное сохранение доступности и целостности данных на томе E:\TestData. Оба тестовых файла остались доступными, их размеры соответствовали исходным, а содержимое не повредилось.

Наблюдение за реакцией файловой системы показало, что механизм Storage Spaces в сочетании с функцией File Integrity обеспечил непрерывную работу тома при сбое без необходимости его отключения или вмешательства администратора.

Таким образом, в ходе эксперимента были подтверждены следующие ключевые характеристики ReFS:

1. Отказоустойчивость: данные оставались доступными даже после отказа одного из носителей.
2. Целостность: контрольные суммы и механизмы предотвратили повреждение файлов.

Процесс восстановления данных выполнялся в фоновом режиме без прерывания работы.

Для наглядной оценки эффективности механизмов ReFS в условиях сбоя носителя в таблице 2 представлен сравнительный анализ ключевых аспектов работы системы до и после имитации отказа.

Таблица 2. Сравнительный анализ работы ReFS

Критерий оценки	До имитации отказа диска	После имитации отказа диска
Доступность данных	Полная	Полная благодаря зеркалированию
Целостность данных	Контрольные суммы активны	Файлы не повреждены, проверки пройдены
Состояние дисков (пула хранения)	Оба диска активны	Один диск в состоянии Retired

Заключение

Проведенное исследование подтвердило высокую эффективность файловой системы ReFS в обеспечении отказоустойчивости и целостности данных в условиях сбоя носителя. Встроенные механизмы Storage Spaces и File Integrity позволили системе автоматически локализовать сбой, сохранить доступность данных и продолжить работу в штатном режиме.

ReFS демонстрирует преимущества перед традиционными файловыми системами (такими, как NTFS) в средах с высокими требованиями к надежности и доступности данных, особенно в условиях современных угроз и больших объемов хранимой информации. Архитектура файловой системы делает ее перспективным решением для корпоративных инфраструктур, виртуальных сред и систем хранения критически важной информации.

Практические рекомендации:

1. Использовать ReFS совместно с зеркальными пулами хранения для систем, требующих высокой доступности и целостности данных.
2. Функцию File Integrity следует включать на всех томах ReFS для обеспечения постоянного контроля целостности.
3. Реализация ReFS целесообразна в средах Windows Server 2022 и выше, где поддерживаются все современные функции системы.

Направления дальнейшего исследования:

1. Анализ производительности ReFS при различных рабочих нагрузках (виртуализация, базы данных, резервное копирование).
2. Исследование отказоустойчивости ReFS при множественных сбоях носителей в пуле хранения.

3. Сравнительный анализ эффективности ReFS и других современных файловых систем в различных ситуациях.

Использованные источники

1. Chand, M. ReFS replacing NTFS in Windows 8 [Электронный ресурс] / М. Chand // C# Corner. – 2012. – URL: <https://www.c-sharpcorner.com/blogs/refs-replacing-ntfs-in-windows-8> (дата обращения: 25.12.2025).

2. Hetman Software. Файловая система ReFS (Resilient File System) [Электронный ресурс] // Дзен. – 2020. – URL: <https://dzen.ru/a/X1qKETtKATv43F12> (дата обращения: 10.01.2026).

3. Microsoft Learn. Resilient File System (ReFS) overview (Обзор файловой системы ReFS) [Электронный ресурс]. – 2025. – URL: <https://learn.microsoft.com/ru-ru/windows-server/storage/refs/refs-overview> (дата обращения: 05.01.2026).

4. Zöchling, F. Server 2016: Speed comparison between ReFS and NTFS (Part 1) [Электронный ресурс] / F. Zöchling. – 2017. – URL: <https://www.frankysweb.de/en/server-2016-speed-comparison-between-refs-and-ntfs-part-1/> (дата обращения: 15.12.2025).

5. ГОСТ Р 57429-2017. Судебная компьютерно-техническая экспертиза. Термины и определения. – Введ. 2017-09-01. – М. : Стандартинформ, 2018. – 12 с.

6. Поддубная, Я. С. Оценка производительности файловых систем Windows Server: FAT32, NTFS и ReFS [Электронный ресурс] / Я. С. Поддубная, С. С. Чуреина // Форум молодых ученых. – 2025. – № 5(105). – С. 142-149. – URL: <https://cyberleninka.ru/article/n/otsenka-proizvoditelnosti-faylovyh-sistem-windows-server-fat32-ntfs-i-refs> (дата обращения: 06.01.2026).

7. Соловьев, Д. Что такое файловая система ReFS [Электронный ресурс] / Д. Соловьев. – 2026. – URL: <https://recoverit.wondershare.com.ru/file-system/refs-file-system.html> (дата обращения: 20.12.2025).

8. Уймин, А. Г. Периферийные устройства ЭВМ : практикум / А. Г. Уймин. – Москва : Ай Пи Ар Медиа, 2023. – 429 с. – ISBN 978-5-4497-2079-5. – Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. – URL: <https://www.iprbookshop.ru/128551.html> (дата обращения: 10.12.2025). – Режим доступа: для авторизир. пользователей. – DOI: <https://doi.org/10.23682/128551>. – С. 143-163.