

16.5 ФАЙЛОВА СИСТЕМА NTFS

В порівнянні з FAT або FAT32, NTFS (New Technology File System) надає користувачеві ціле поєднання переваг: ефективність, надійність і сумісність. Перечислимо деякі її особливості.

1. 64-розрядні адреси, тобто теоретично може підтримувати $2^{64} * 2^{16}$ байт (18 446 744 073 Тб).
2. Розміри блоку (кластеру) від 512 байт до 64 Кб, частіше використовується 4 Кб.
3. Підтримка великих файлів.
4. Імена файлів обмежені 255 символами Unicode.
5. Довжина шляху обмежується 32767 (2^{15}) символами Unicode.
6. Імена чутливі до регістра, my.txt і MY.TXT це різні файли.
7. Файлова система з журналом, тобто не потрапить в суперечливий стан після збоїв.
8. Контроль доступу до файлів і каталогів.
9. Підтримка жорстких і символічних посилань.
10. Підтримка стискування і шифрування файлів.
11. Підтримка дискових квот.

Система NTFS розроблена для швидкого виконання операцій як стандартних файлових (читання, запис і пошук), так і удосконалених (наприклад, відновлення файлової системи) на дуже великих жорстких дисках.

Підтримуючи управління доступом до даних, NTFS дає гарантії безпеки, необхідні для файлових серверів і персональних комп'ютерів в корпоративному середовищі. Це важливо для цілісності корпоративних даних.

Система NTFS проста, але дуже потужна розробка, для якої вся інформація на томі NTFS – файл або частина файлу. Кожен розподілений на томі NTFS сектор належить деякому файлу. Частиною файлу є навіть метадані файлової системи (інформація, що описує безпосередньо файлову систему). Файлова система підтримує об'єктно-орієнтовані додатки, обробляючи всі файли як об'єкти з атрибутами, обумовленими користувачем і системою.

16.5.1 Ключові можливості NTFS

До основних особливостей NTFS належать:

1. **Здатність відновлення даних.** Це досягається за допомогою використання моделі обробки транзакцій для операцій обміну у файловій системі. Кожен обмін розглядається як атомарна дія, яка або виконується повністю, або не виконується зовсім. Окрім цього, NTFS використовує надмірне зберігання критичних даних файлової системи.

2. **Безпека.** Для забезпечення безпеки NTFS використовує об'єктну модель W2K. Відкритий файл реалізується як файловий об'єкт з дескриптором, що визначає атрибути безпеки.

3. **Диски і файли великих об'ємів.** NTFS підтримує дуже великі диски і файли ефективніше, ніж FAT і інші системи.

4. **Узагальнена індексація.** NTFS зв'язує з кожним файлом набір атрибутів. Набір описів файлів в системі управління файлами організований як реляційна база даних, тому файли можуть бути індексовані за будь-яким атрибутом.

5. **Множинні потоки даних.** Файл в NTFS – це не просто лінійна послідовність байтів, як файли в системах FAT-32 і UNIX. Замість цього файл складається з декількох атрибутів, кожен з яких представлений у вигляді потоку байтів. Більшість файлів мають декілька коротких потоків, таких як ім'я файлу і його 64-бітовий ідентифікатор, плюс один довгий (неіменованний) потік з даними. У файлу може бути і декілька довгих потоків даних.

16.5.2 Том NTFS і файлова структура

NTFS використовує такі концепції дискового зберігання:

1. **Сектор.** Найменша одиниця фізичного зберігання на диску. Розмір даних у байтах є степенем двійки і майже завжди рівний 512 байт.

2. **Кластер.** Один або декілька послідовних секторів на одній доріжці. Звернення до блоків (кластерів) здійснюється за їх зміщенням від початку тому.

3. **Том.** Логічний розділ диска, що складається з деякої кількості кластерів. Том може займати як увесь диск, так і його частину або охоплювати декілька дисків (до 2^{64} байт).

Кластер є фундаментальною одиницею розміщення у файловій системі NTFS, яка не розпізнає сектори. Нині максимальний розмір файлу складає 2^{48} байт.

Використання кластерів при розміщенні файлів робить систему NTFS незалежною від розмірів фізичних секторів. Це дозволяє їй без перешкод підтримувати нестандартні диски з розміром сектора, не рівним 512 байт. У таблиці. 16.4 приведені розміри кластерів системи NTFS за умовчанням.

Таблиця 16.4 – Розділи NTFS і розміри кластерів

| Розмір тому | Секторів у кластері | Розмір кластера |
|---------------|---------------------|-----------------|
| <= 512 Мб | 1 | 512 байт |
| 512 Мб – 1 Гб | 2 | 1 Кб |
| 1 Гб – 2 Гб | 4 | 2 Кб |
| 2 Гб – 4 Гб | 8 | 4 Кб |
| 4 Гб – 8 Гб | 16 | 8 Кб |
| 8 Гб – 16 Гб | 32 | 16 Кб |
| 16 Гб – 32 Гб | 64 | 32 Кб |
| > 32 Гб | 128 | 64 Кб |

16.5.3 Схема тому NTFS

NTFS використовує простий і в той же час потужний підхід в організації інформації на томі диска. Кожен елемент тома є файлом, і кожен файл складається з набору атрибутів (навіть дані, що зберігаються у файлі, розглядаються як атрибут). При такій простій структурі достатньо невеликої кількості функцій загального призначення для організації і управління файловою системою. На рис. 16.10 показана схема тому, що складається з 4-х областей.

| | | | |
|------------------------------------|----------------------------|----------------|----------------|
| Завантажувальний сектор розділу | Головна файлова таблиця | Системні файли | Область файлів |
|------------------------------------|----------------------------|----------------|----------------|

Рисунок 16.10 – Схема тому NTFS

Перші декілька секторів будь-якого тому займає **завантажувальний сектор розділу** (розміром до 16 секторів), що містять інформацію про схему тому і структури файлової системи, а також початкову завантажувальну інформацію і код завантаження. Потім іде **головна файлова таблиця MFT** (Master File Table), що є

лінійною послідовністю записів фіксованого розміру (1 Кб). Вона містить інформацію про всі файли і каталоги цього тому, а також інформацію про вільний простір. За областю MFT іде область, завдовжки 1 Мб, що містить *системні файли* (див. рис. 16.10).

16.5.4 Головна файлова таблиця

Кожен файл на томі NTFS представлений записом фіксованого розміру в спеціальному файлі – *головній файловій таблиці MFT (Master File Table)*. NTFS резервує перші 16 записів таблиці для спеціальної інформації, файлів метаданих (рис. 16.11).

Перший запис таблиці описує безпосередньо головну файлову таблицю. За нею йде дзеркальний запис MFT. Якщо перший запис MFT зруйнований, NTFS прочитає другий запис, щоб відшукати дзеркальний файл MFT, перший запис якого ідентичний першому запису MFT.



Рисунок 16.11 – Головна файлова таблиця MFT, кожен запис посилається на файл або каталог

Місце розташування сегментів даних MFT і дзеркального файлу MFT записане в секторі початкового завантаження. Дублікат сектора початкового завантаження

знаходиться в логічному центрі диска. Третій запис MFT – файл реєстрації, який застосовується для відновлення файлів.

Сімнадцятий і подальші записи головної файлової таблиці використовуються файлами і каталогами на томі. Призначення цих файлів описане в показаній нижче таблиці MFT (табл. 16.5), що забезпечує дуже швидкий доступ до файлів.

Якщо файл дуже великий, то іноді використовується два і більше записів головної файлової таблиці, щоб вмістити список усіх блоків файлу. В цьому випадку 1-й запис MFT, що називається *базовим записом*, вказує на інші записи MFT.

Сама головна файлова таблиця є файлом і, як будь-який файл, може розташовуватися в будь-якому місці тому, тим самим усувається проблема дефектних секторів на першій доріжці дискового розділу.

Кожен атрибут розпочинається із заголовка, що ідентифікує цей атрибут і значення, що повідомляє довжину, оскільки деякі атрибути, наприклад, ім'я файлу або дані, можуть мати змінну довжину. Якщо значення атрибуту занадто довге, воно розташовується в іншому місці диска, а в запис MFT поміщається покажчик на нього.

Таблиця 16.5 – Головна файлова таблиця NTFS (0-15 – файли метаданих)

| Номер запису | Системний файл | Ім'я файлу | Призначення файлу |
|--------------|---------------------------------|------------|---------------------------------------------------------------------------------------|
| 0 | Головна таблиця | \$Mft | Містить повний список файлів тому NTFS |
| 1 | Копія головної таблиці файлів | \$MftMirr | Дзеркальна копія перших трьох записів MFT |
| 2 | Файл журналу | \$LogFile | Список транзакцій, який використовується для відновлення файлової системи після збоїв |
| 3 | Том | \$Volume | Ім'я тому, версія NTFS, інформація про том |
| 4 | Таблиця атрибутів | \$AttrDef | Таблиця імен, номерів і описів атрибутів |
| 5 | Індекс кореневого каталогу | \$. | Кореневий каталог |
| 6 | Бітова карта кластерів | \$Bitmap | Розмітка використаних кластерів тому |
| 7 | Завантажувальний сектор розділу | \$Boot | Адреса завантажувального сектора розділу |
| 8 | Файл поганих кластерів | \$BadClus | Файл, що містить список усіх виявлених на томі поганих кластерів |
| 9 | Таблиця квот | \$Quota | Квоти використовуваного простору на диску для кожного користувача |

| | | | |
|--------|----------------------------------------|----------|---------------------------------------------------------------------------|
| 10 | Таблиця перетворення регістра символів | SUppcase | Використовується для перетворення регістра символів для кодування Unicode |
| 11 | Розширення | | Квоти |
| 12-15 | Зарезервовані | | |
| 16 ... | | | Перший файл користувача і так далі |

Перші 16 записів MFT зарезервовані для файлів метаданих NTFS. Кожен запис описує нормальний файл, в якого є атрибути і блоки даних, як у будь-якого файлу. У кожного такого файлу є ім'я, яке розпочинається з символу долара, що вказує на те, що це файл метаданих.

Перший запис описує сам файл MFT. Зокрема, він містить інформацію про розташування блоків файлу MFT, що дозволяє системі знайти файл MFT. Очевидно, щоб знайти всю іншу інформацію про систему, в ОС має бути деякий спосіб знаходження першого блоку файлу MFT, який міститься в завантажувальному блоці, куди він поміщається при розмітці тому.

Запис 1 є дублікатом першої частини файлу MFT. Ця інформація є настільки цінною, що наявність другої копії може бути потрібна, якщо один з перших блоків головної файлової таблиці раптом стане дефектним.

Запис 2 є журналом. Коли у файловій системі робляться структурні зміни, такі як додавання нового каталогу або видалення каталогу, інформація про майбутню операцію реєструється в журналі. Таким чином, збільшується ймовірність коректного відновлення файлової системи в разі збою під час виконання операції.

Запис 3 містить інформацію про том, наприклад, його розмір, мітка.

Запис 4 містить файл \$AttrDef, в якому визначаються атрибути.

Запис 5 містить дані про кореневий каталог, який сам є файлом і може доволно збільшуватися в розмірах.

Запис 6 враховує вільне місце на диску за допомогою бітового масиву, що є також файлом.

Запис 7 вказує на файл початкового завантаження.

Запис 8 використовується для того щоб зв'язати разом усі дефектні блоки і гарантувати, що вони ніколи не зустрінуться у файлах.

Запис 9 містить інформацію про захист.

Запис 10 використовується для перетворення регістра. Для символів латинського алфавіту від А до Z перетворення регістра не представляє проблем. Для інших мов (наприклад, грузинської, грецької) це питання не таке очевидне, тому файл містить необхідні інструкції.

Запис 11 є каталогом, що містить різні файли для дискових квот, ідентифікаторів об'єктів тощо.

Кожен запис MFT має *заголовок запису*, за яким іде послідовність пар (заголовок атрибуту, значення). Заголовок запису містить:

- магічне число, яке використовується для перевірки дійсності запису;
- порядковий номер, що оновлюється кожного разу, коли запис використовується для нового файлу;
- лічильник звернень до файлу;
- кількість байт, використуваних в записі;
- ідентифікатор (індекс, порядковий номер) базового запису, який використовується тільки для запису розширення, а також інші поля.

У файловій системі NTFS визначені 13 атрибутів, які можуть з'являтися в записах MFT (табл. 16.6).

Всі записи таблиці MFT складаються з послідовності заголовків атрибутів, кожен з яких ідентифікує наступний за ним атрибут, а також містить довжину і розташування поля значення з різноманітними прапорами та іншою інформацією.

Як правило, значення атрибутів розташовуються відразу ж за заголовками. Але, якщо довжина значення занадто велика, щоб поміститися в запис таблиці MFT, вона може бути поміщена в окремий блок диска. Такий атрибут називають *нерезидентним атрибутом*. Наприклад, таким атрибутом є атрибут даних.

Ім'я потоку даних, якщо він є присутнім, міститься в заголовку атрибуту даних. Слідом за цим заголовком розташовується або список дискових адрес, що визначає місце розташування файлу на диску, або, для файлів завдовжки всього в декілька сотень байтів, сам файл.

Ім'я потоку даних, якщо він є присутнім, міститься в заголовку атрибуту даних. Слідом за цим заголовком розташовується або список дискових адрес, що визначає місце розташування файлу на диску, або, для файлів завдовжки всього в декілька сотень байтів, сам файл. Метод поміщення самого утримуваного файлу в запис MFT

називається *безпосереднім файлом*. Як правило, всі дані файлу не поміщаються в запис MFT. Тому дискові блоки файлам призначаються заможливістю у вигляді *серій послідовних блоків в (сегментів файлів)*. В ідеалі файл має бути записаний в одну серію (нефрагментований файл), файл, що складається з n блоків, може бути записаний від 1 до n серій.

Таблиця 16.6 – Типи атрибутів файлів і каталогів в NTFS

| Атрибут | Опис |
|--------------------------------|---------------------------------------------------------|
| Стандартна інформація | Біти прапорів, тимчасові штампи тощо |
| Ім'я файлу | У кодуванні Unicode |
| Описувач захисту | Застарів. Тепер знаходиться в атрибуті \$Extend\$Secure |
| Список атрибутів | Розташування додаткових записів MFT |
| Ідентифікатор об'єкту | 64-розрядний ідентифікатор файлу |
| Точка повторного аналізу | Для монтування і символічних посилань |
| Назва тому | Назва тому (тільки у \$Volume) |
| Інформація про том | Версія тому (тільки у \$Volume) |
| Кореневий індекс | Використовується для каталогів |
| Розміщення індексу | Використовується для дуже великих каталогів |
| Потік даних утиліти реєстрації | Управляє реєстрацією у файлі |
| Дані | Потокові дані; можуть повторюватися |

На рисунку 16.12 представлений запис MFT для 9-блокового файлу, що складається з трьох сегментів (серій).

Увесь запис поміщається в один запис MFT. Заголовок містить кількість блоків (9 блоків). Кожна серія записується у вигляді пари, дискова адреса – кількість блоків (20-4, 64-2, 80-3). Кожна пара, за відсутності стискування, це два 64-розрядні числа (16 байт на пару). Багато адрес містять велику кількість нулів, стискування робиться за рахунок прибирання нулів в старших байтах. У результаті для пари потрібно частіше всього 4 байти.

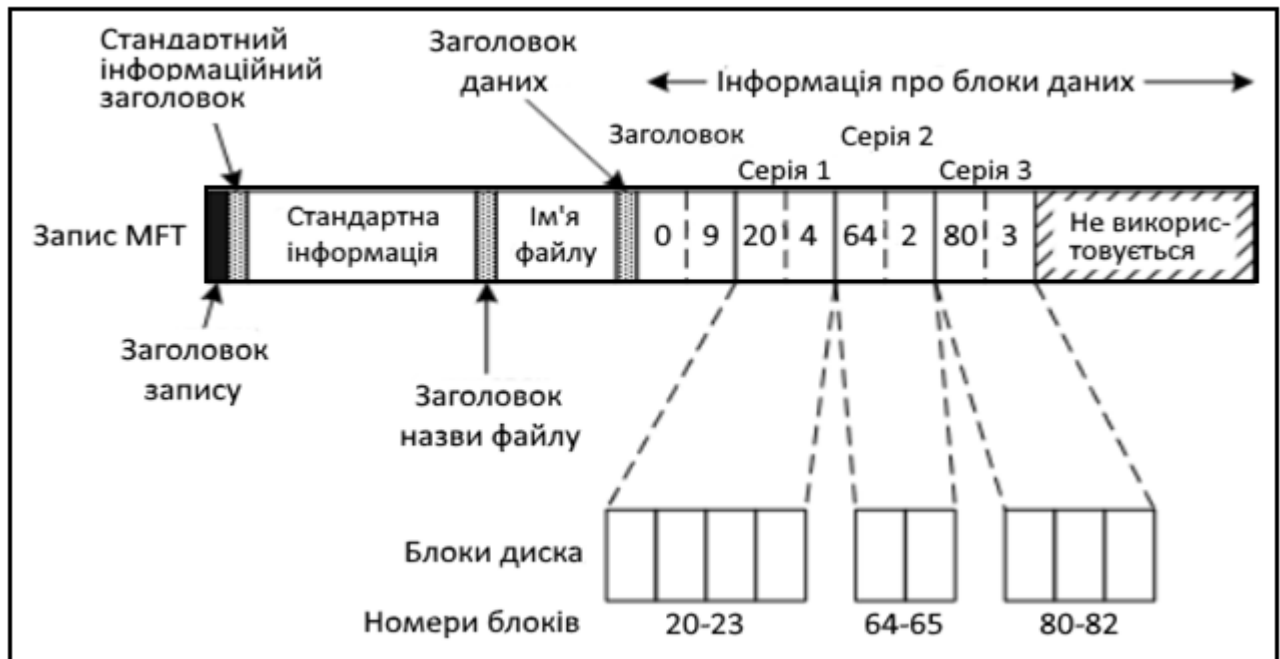


Рисунок 16.12 – Запис MFT для 9-блочного файлу

16.5.5 Цілісність даних і відновлення в NTFS

NTFS – це відновлювана файлова система. Кожна операція введення-виведення, що змінює файл на томі NTFS, розглядається файловою системою як транзакція і може виконуватися як неподільний блок. При модифікації файлу користувачем сервіс файлу реєстрації фіксує всю інформацію, необхідну для повторення або відкату транзакції. Якщо транзакція завершена успішно, робиться модифікація файлу. Якщо ні, NTFS виконує відкат транзакції згідно інструкціями в інформації скасування. Якщо в транзакції виявлена помилка, то вона виконується в зворотному порядку.

При збої системи NTFS виконує три проходи: аналізу, повторів і відкатів. У процесі аналізу на підставі інформації файлу реєстрації NTFS оцінює ушкодження і точно визначає, які кластери треба модифікувати. Під час повторного проходу виконуються всі етапи транзакції від останньої контрольної точки. При відкаті відбувається повернення всіх незакінчених транзакцій.

Важлива особливість NTFS – *відкладена передача*, що дозволяє, подібно до відкладеного запису, мінімізувати витрати на реєстрацію транзакцій. Замість використання ресурсів для негайної відмітки транзакції як успішно завершеної, ця інформація заноситься в кеш і записується у файл реєстрації як фоновий процес. Якщо збій відбувається до того, як інформація про транзакцію була зареєстрована,

NTFS зробить повторну перевірку транзакції для визначення її успішності. Якщо NTFS не може гарантувати, що транзакція завершилася успішно, робиться відкат транзакції. Ніякі незавершені модифікації тома не дозволені.

Кожні декілька секунд NTFS перевіряє кеш, щоб визначити стан відкладеного запису і відмітити його у файлі реєстрації як контрольну точку. Якщо вслід за визначенням контрольної точки наступить збій, система має можливість повернутися до стану, зафіксованого контрольною точкою. Цей метод призначений для захисту метаданих і забезпечує оптимальний час відновлення, зберігаючи чергу подій, яка може знадобитися в процесі відновлення. Дані користувача в разі збою системи можуть бути зруйновані.

16.5.6 Компресія файлів і каталогів

Особливість NTFS – можливість динамічного стискування файлів і каталогів. Файлова система NTFS підтримує прозоре стискування файлів. Файл може бути створений в стислому режимі. Це означає, що система NTFS автоматично намагатиметься стиснути блоки цього файлу при записі їх на диск і автоматично розпаковувати їх при читанні.

Стискування є новим атрибутом файлу або каталогу і подібно до будь-якого атрибуту може бути зняте або встановлене в будь-який момент часу. Стискування даних файлу відбувається таким чином. Коли система NTFS записує на диск файл, помічений для стискування, вона вивчає перших 16 логічних блоків файлу. Потім до цих блоків застосовується алгоритм стискування. Якщо отримані на виході блоки можуть поміститися в 15 або менше блоків, то стиснуті дані записуються на диск. Якщо отримати виграш хоч би в один блок не вдається, то дані 16 блоків так і записуються в нестиснутому виді. Потім увесь алгоритм повторюється для наступних 16 блоків і так далі.