

--- ФИРМА НИППЕЛЬ ПРЕДСТАВЛЯЕТ ---

ФОРМАТ ДОСТУПА/ХРАНЕНИЯ ИНФОРМАЦИИ "NIPFEL FILE SYSTEM"

Стандарт

Автор: Голов А.А.

Основные технические характеристики

Файловая система Nifils является стандартом фирмы Ниппель на долговременное хранение информации на носителях последовательно-прямого доступа,

Стандарт обеспечивает гибкость доступа к информации и высокую скорость. Обеспечивается адресация 16 М единиц информации (кластеров) в пределах одного тома, общий объем которого, при кластерах размером 512 байт, достигает 8 Гбайт, Максимально допустимый размер кластера равен 128 Кбайт, поэтому теоретически размер тома может достигать 2 Тбайт. Максимальный размер файла для кластера размером 512 байт равен 2,45 Гбайт, при кластерах большего размера 4 Гбайт. Максимальное число файлов в одном каталоге 65535. Максимальное число подкаталогов и уровней их вложения ограничено кол-вой единиц информации (16 М). Размер имени файла - 16 символов, расширение имени - 4 символа.

Стандарт обеспечивает хранение размера файла с точностью до байта, атрибутов доступа к файлу, даты и времени последней модификации файла, четырех байт для информации произвольного назначения (по усмотрению программы - владельца).

Стандарт допускает произвольную фрагментацию пространства кластеров, и поддерживает его целостность, обеспечивает выполнение функций прямого доступа к файлу, создание, модификации, удаление и обратное восстановление файлов.

Описание файловой системы

Область файловой системы представляет из себя последовательность кластеров (в первой версии **File Manager** кластер имеет фиксированный размер в один блок, поэтому можно говорить о последовательности блоков), с общим числом до 16 М и обозначающихся трехбайтовым числом. Кластер с номером \$FFFFFF отсутствует и в полях каталога выражает отсутствие продолжения. Нумерация блоков начинается с 0 в котором находится заголовок тома, являющийся, в том числе, заголовком корневого каталога. Вся информация для доступа к тому черпается из заголовка. Загрузчики записываются на известных драйверу диска секторах, которые не входят в состав области файловой системы. Таким же образом хранится информация о разбиении одного носителя информации на несколько томов, т.е. на уровне драйвера диска.

Каждый блок каталога (в том числе блоки с заголовками тома и подкаталога) содержит указатель на предыдущий кластер каталога (байты 0-2, причем значение \$FFFFFF означает, что текущий кластер - головной в каталоге), указатель на следующий кластер каталога (байты 3-5, причем значение \$FFFFFF означает, что текущий кластер - хвостовой в каталоге).

Первая запись в блоке располагается начиная с байта 6. Размер записи и число записей указано в заголовке тома. Для **File Manager** версии 1, запись имеет размер 42 байта, а в блоке размещается 12 записей. Первая запись в каталоге - заголовок либо тома, либо подкаталога.

Заголовок тома помимо имени тома определяет все общие параметры и значения для обработки этого тома программой **File Manager**. Заголовок занимает одну стандартную запись корневого каталога, но некоторые поля этой записи имеют другое назначение чем в записи о файле.

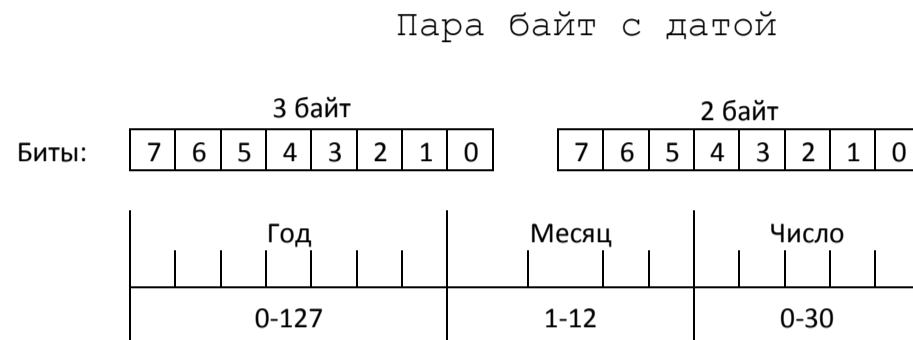
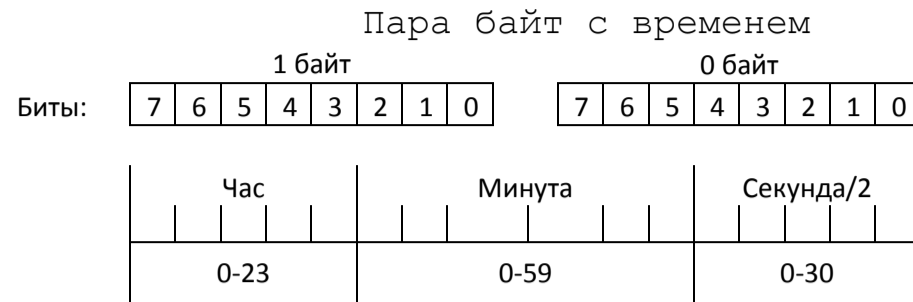
Байт	Наименование поля
1	Размер имени + признак заголовка тома
20	Имя тома в формате ГОСТ
1	Число блоков (512 байтных) в кластере
3	Общее число кластеров в томе
1	Атрибуты доступа к тому
2	Число задействованных записей в корневом каталоге
1	Размер записи каталога (в байтах)
1	Число записей в одном блоке (512 байтном)
4	Дата и время создания тома
1	Минимальный номер версии File Manager для доступа к тому
2	Не используется
1	Способ доступа к Bit Map
3	Указатель на кластер начала системы Bit Map
1	Не используется

Табл.1 Формат заголовка тома - первой записи

В Табл. 1 указан формат заголовка тома, а ниже рассматриваются все поля заголовка тона и указаны особенности их значений:

1. Размер имени + признак заголовка тома. Формат байта - :XXXYYYYY X - 7 признак заголовка тома Y - 1-\$14 размер имени (1-20 символов, соответственно)
2. Имя тома. Набор символов с кодами \$20-\$FF соответствующих кодировке ГОСТ.
3. Общее число кластеров в томе. Трехбайтовое число, выражающее номер последнего кластера, тома плюс один.
4. Число блоков в кластере. Означает кол-во блоков по 512 байт, скрытых за одним кластером. Может принимать значения от 0-\$FF, соответственно от 1 до 256 блоков в кластере. В **File Manager** версии 1 не используется, тем не менее должен иметь значение 0.
5. Атрибуты доступа. Набор единичных битов разрешающих различные операции доступа к тому:
 - бит 0 - том может быть прочитан
 - бит 1 - том может быть записан
 - бит 2 - том может быть переименован
 - бит 3 - том может быть уничтожен (проинициализирован)
6. Число задействованных записей каталога. Выражает кол-во активных записей в текущем каталоге, т.е. все записи содержащие информацию о файлах, исключая заголовки каталога и удаленные файлы. Ноль означает отсутствие файлов в каталоге.
7. Размер записи каталога. Число байт занимаемых одной записью каталога. В **File Manager** версии 1 - 42.
8. Число записей в одном блоке каталога. В **File Manager** версии 1 - 12.

9. Дата и время создания тома. Первая пара байт содержит дату, а вторая время создания тома.



Год образуется добавлением содержимого поля Год к 1980 базовому году.

10. Минимальный номер версии **File Manager** для доступа к тому. Том имеющий в этом поле значение больше чем допускающая к нему **File Manager** не будет обработан.

11. Способ доступа к Bit Map

0 - прямая последовательность блоков с Bit Map.

1 - первый блок системы Bit Map - Bit Map 1 - карта наличия Свободных блоков в соответствующих блоках Bit Map.

12. Указатель на кластер системы Bit Map. Указывает первый кластер системы Bit Map. Формат Bit Map будет описан ниже.

Как уже говорилось выше, заголовок тома является одновременно и заголовком корневого каталога. Остальные записи каталога представляют собой описания файлов. Формат записи о файле представлен в Табл. 2:

Байт	Наименование поля
1	Размер имени файла + способ доступа к файлу
20	Имя файла в формате ГОСТ
4	Расширение имени файла
1	Атрибуты доступа к файлу + вид хранимой информации
4	Размер файла (в байтах)
4	Дата и время последней модификации
4	Дополнительная информация
3	Указатель на кластер файла, Index 1,2,3
1	Поле для хранения способа доступа у стертого файла

Табл.2 Формат активной записи каталога, описывающей файл

1. Размер имени файла + способ доступа к файлу. Формат байта: XXXYYYYY

X - 0 - удаленный файл

1 - файл без Index кластера (т.е. размером в 1 кластер)

2 - файл с уровнем Index 1

3 - файл с уровнем Index 2

4 - файл с уровнем Index 3

5 - файл подкаталог

6 - заголовок подкаталога

7 - заголовок тома

Y - 1-\$14 размер имени (1-20 символов, соответственно)

2. Имя файла (до 20 символов). Набор символов с кодами \$20-\$FF соответствующих кодировке ГОСТ.

3. Расширение имени файла (до 4 символов). Набор символов с кодами \$20-\$FF соответствующих кодировке ГОСТ.

4. Атрибуты доступа + вид хранимой информации. Байт вида :XXXXYUUU:

X – вид хранимой информации (справочная информация):

- 0 – подкаталог
- 1 – системный файл
- 2 – настроечный файл
- 3 – программный файл
- 4 – оттранслированный перемещаемый файл
- 5 – текстовый файл
- 6 – текстовый документ
- 7 – текст программы
- 8 – текст иного рода
- 9 – графический матричный файл
- A – графический векторный файл
- B – графический иного рода
- C – база данных
- D – индексный файл
- E – метаданные
- F – двоичный (неклассифицируемые данные)

Y – набор единичных битов разрешающих различные операции доступа к файлу:

- бит 0 – файл может быть прочитан
- бит 1 – файл может быть записан
- бит 2 – файл может быть переименован
- бит 3 – файл может быть уничтожен.

5. Размер файла (в байтах). 32-разрядное число указывающее на полный размер файла с точностью до байта.

6. Дата и время последней модификации файла. Формат поля такой же как и у заголовка тома.

7. Дополнительная информация. Четырехбайтное поле содержащее произвольную информацию, по усмотрению создавшей файл программы.

8. Указатель на кластер файла, Index 1,2,3. Поле осуществляющее связь записи каталога с пространством файла. Формат системы индексов файла будет описан ниже.

9. Поле для хранения способа доступа стерттого файла. При удалений файла заносится 0 в поле способа доступа к файлу и соответственно теряется информация о способе доступа. Для возможности последующего восстановления файла байт размера имени и способа доступа буферизуется в данном поле.

Из формата поля 'способ доступа к файлу' можно вывести, что файлы бывают двух типов обычные и подкаталоги. Структура обычного файла будет описана ниже, а сейчас мы рассмотрим формат подкаталога.

В записи о файле-подкаталоге имеется указатель на кластер с заголовком подкаталога. Формат заголовка подкаталога описан в Табл. 3.

Байт	Наименование поля
1	Размер имени подкаталога + признак заголовка подкаталога
20	Имя подкаталога в формате ГОСТ
4	Не используется
1	Атрибуты доступа
2	Число задействованных записей в подкаталоге
3	Указатель на заголовок родительского каталога
5	Не используется
1	Номер блока кластера, с родительским файлом
1	Номер записи в блоке, с родительским файлом
3	Указатель на кластер каталога с родительским файлом
1	Не используется

Табл.3 Формат заголовка подкаталога – 1-й записи 1-го кластера подкаталога

1. Размер имени + признак заголовка подкаталога. Формат байта - :XXXYYYYY:
X - 6 признак заголовка подкаталога
Y - 14 размер имени (1-20 символов, соответственно)
2. Имя подкаталога. Набор символов с кодами \$20-\$FF соответствующих кодировке ГОСТ.
3. Атрибуты доступа. Набор единичных битов разрешающих различные операции доступа к подкаталогу:
бит 0 - подкаталог может быть прочитан
бит 1 - подкаталог быть записан
бит 2 - подкаталог быть переименован
бит 3 - подкаталог быть уничтожен
4. Число задействованных записей подкаталога. Выражает кол-во активных записей в подкаталоге, т.е. все записи содержащие информацию с файлах, исключая заголовков подкаталога и удаленные файлы. Ноль означает отсутствие файлов в подкаталоге.
5. Указатель на заголовок родительского подкаталога. Указывает *на* кластер с заголовком каталога содержащего соответствующий файл-подкаталог. Предназначен для обратного хода по уровням подкаталогов.
6. Номер блока кластера с родительским файлом. Используется в том же с размером кластера больше 1 блока (512 байтного), для указания точного местоположения записи о файле подкаталоге. Может принимать значения от 0-\$FF, соответственно от 0-го до 255-го блока кластера. В **File Manager** версии 1 не используется, но тем не менее должен быть равен 0.
7. Номер записи в блоке с родительским файлом. Предназначен для точной идентификации записи о соответствующем файле-подкаталоге. Нумерация производится от 0 до 11 (в **File Manager** версии 1), причем в расчете учитывается и заголовок тома или подкаталога.
8. Указатель на кластер каталога с родительским файлом. Предназначен для идентификации кластера каталога содержащего запись о соответствующем файле-подкаталоге.

Обычный файл обладает более сложной и развитой структурой. Для идентификации составляющих файл кластеров используется система индексных списков. Поле записи о файле 'способ доступа' предоставляет четыре варианта индексации файла по значению поля:

1 - Означает отсутствие индекса, т.е. файл состоит из одного кластера и его номер непосредственно указан в записи о файле.

2 - Присутствует кластер-индекс файла - Index 1. Этот кластер содержит номера кластеров файла. Всего содержится 170 трехбайтовых записей. Младшие байты хранятся с адреса 0 по \$A9, средние с адреса \$AA по \$ 153, старшие с адреса \$ 154 по \$1FD. Таким образом Index 1 при кластере размером в 1 блок (512 байт) указывает максимум 170 кластеров файла. При кластерах большего размера в соответствующее кол-во раз увеличивается число индексируемых кластеров.

3 - Присутствует кластер-индекс второго уровня - Index 2. Этот кластер содержит номера кластеров с Index 1. Другими словами это индекс индексов. Кол-во адресуемых кластеров файла возрастает в 170 раз, а время доступа увеличивается только на одно обращение. Способ хранения записей Index 2 аналогичен Index 1.

4 - Присутствует кластер-индекс третьего уровня - Index 3. Этот кластер как и Index 2 также является индексом индексов, только он индексирует кластеры с Index 2. Кол-во адресуемых кластеров файла возрастает, еще в 170 раз и при размере кластера в 1 блок позволяет использовать файлы до 2.45 Гбайт.

Теперь расскажем о том, каким образом распределяется пространство кластеров тома. Для этого существует карта свободных кластеров (Bit Map) посредством сканирования которой можно находить неиспользованные кластеры. Последовательность нумерации кластеров в блоке Bit Map установлена от старшего бита байта к младшему и от начала блока к концу. Свободным считается кластер обозначенный 1.

Для обращения к Bit Map в заголовке тома имеется два поля: 'способ доступа к Bit Map' и 'указатель на кластер начала системы Bit Map'. Bit Map может иметь два варианта исполнения в зависимости от состояния поля 'способ доступа':

0 - Bit Map представляет из себя просто последовательность кластеров начиная с указанного в заголовке тома. Bit Map не индексируется т.к. создается один раз при инициализации и размеров своих не меняет. Последний значащий бит четко выражен через размер тома, указанный в заголовке тома.

1 - Первый блок является вторым уровнем Bit Map (Bit Map 1), т.е. карта карты. Дело в том, что при 16 М единиц информации размер Bit Map может достигать 2 Мбайт (4096 блоков), и просматривать такой объем информации для поиска свободного блока, при уже основательно заполненном пространстве, неэкономично. Этот блок (Bit Map 1) содержит 1 в полях соответствующих блоку Bit Map в котором есть свободные блоки. Такой способ позволяет осуществлять быстрый доступ к блокам на носителях чрезвычайно большой емкости.