

# Методичка по решению практических заданий по системам хранения FAT12 и FAT16

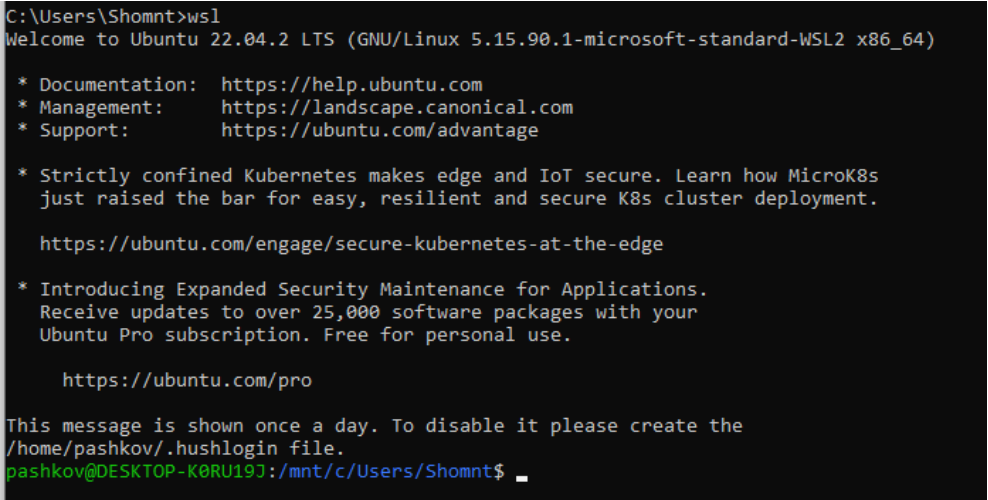
## Вступление

В данной методичке я буду описывать теорию и практику для каждого шага. Иными словами, сначала объяснять, что мы делаем и почему, а затем показывать этот процесс на примере. Итак, начнём.

## Шаг 0 – подготовка

Данный шаг предназначен для людей, которые впервые слышат о практической части зачёта по ос. Если вы уже знакомы с сайтом и знаете, что нужно подготовить для выполнения задания, то смело пропускайте этот шаг.

Перед началом решения на нужно сделать несколько вещей. Первое подготовить Linux систему. Это может быть, что угодно: ваша машина с этой ос, виртуальная машина с этой ос, Windows wsl, есть прецедент использования онлайн Linux консоли. Главное, чтобы вы могли взаимодействовать с файлами, используя команды linux. В примере будет использоваться wsl.



```
C:\Users\Shomnt>wsl
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

This message is shown once a day. To disable it please create the
/home/pashkov/.hushlogin file.
pashkov@DESKTOP-K0RU19J:~/mnt/c/Users/Shomnt$
```

Рисунок 1 – Windows wsl

Второе, что нужно сделать, это открыть сайт, на котором будет выполняться задание. Вот ссылка: <https://fat.bk252.ru>

После открытия перед нами предстанет следующая картина:

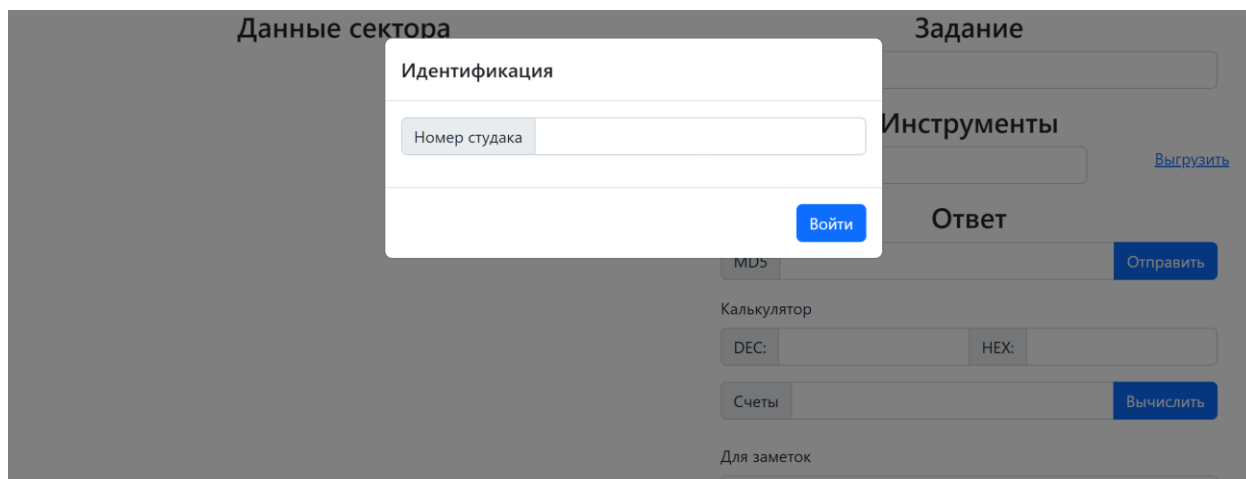


Рисунок 2 – Сайт для выполнения практических

Далее вводим «номер студака». Не обязательно вводить настоящий, можно ввести любой набор букв или цифр.

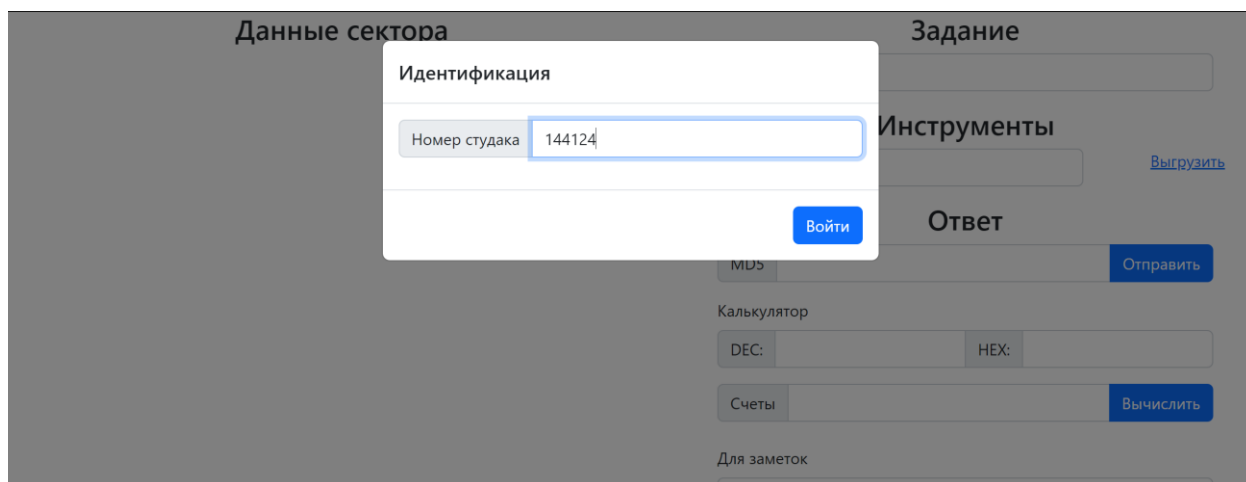


Рисунок 3 – Пример ввода «номера студака»

После нажатия войти ждем загрузки данных. После её завершения перед нами появится задание. Давайте для начала обратим

внимание на правую часть страницы.

## Задание

Имя файла UIVwN/xhmUZ/yHGSv/xhmUZ

## Инструменты

Номер сектора 0

[Выгрузить](#)

## Ответ

MD5

[Отправить](#)

Калькулятор

DEC:

HEX:

Счеты

[Вычислить](#)

Для заметок

Рисунок 4 – Правая часть страницы сайта.

В поле «Имя файла» указан путь до файла, контрольную сумму которого нам необходимо посчитать.

В поле «Номер сектора» указан сектор, в котором мы сейчас находимся. Его мы также будем использовать для перемещения по секторам, вводя в его нужный нам номер.

Кнопка «выгрузить» скачивает сектор, в котором мы сейчас находимся.

В поле «MD5» мы будем вводить наш итоговый ответ.

В поле «Счёты» мы можем вводить какое либо число, и в полях «DEC» и «HEX» появятся его десятичная и шестнадцатиричная записи соответственно.

В поле «Для заметок» мы можем записывать любую информацию.

Важно учитывать, что на выполнение задания даётся один час. На этом подготовка завершена. Приступаем к следующему шагу.

### Шаг 1 – анализ загрузочного сектора

**Теория.** Загрузочный сектор – это сектор, в котором хранится вся основная информация об текущем диске. Из него мы будем доставать базовую информацию, необходимую для выполнения задания. Все данные в секторах указаны в 16-тиричной системе счисления и задом наперёд (чуть позже объясню, что имеется в виду). Каждые две цифры, не отделённые пробелом – это один байт.

Посмотрим на пример загрузочного сектора:

```
offset 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 5B 3C 50 6D 6B 66 73 2E 66 61 74 00 02 02 02 00| ë<mkfs.fat.....
00000010 02 00 02 00 20 F8 0C 00 20 00 02 00 00 00 00 00| ....ø..
00000020 00 00 00 00 80 00 29 A6 C8 79 5F 4E 4F 20 4E 41| ....).!Ëy NO NA
00000030 4D 45 20 20 20 20 46 41 54 31 32 20 20 20 0E 1F| ME FAT12 ..
00000040 BE 5B 7C AC 22 C0 74 0B 56 B4 0E BB 07 00 CD 10| %[|-"Àt.V´.»...í.
00000050 5E EB F0 32 E4 CD 16 CD 19 EB FE 54 68 69 73 20| ^ëð2äí.í.ëþThis
00000060 69 73 20 6E 6F 74 20 61 20 62 6F 6F 74 61 62 6C| is not a bootabl
00000070 65 20 64 69 73 6B 2E 20 20 50 6C 65 61 73 65 20| e disk. Please
00000080 69 6E 73 65 72 74 20 61 20 62 6F 6F 74 61 62 6C| insert a bootabl
00000090 65 20 66 6C 6F 70 70 79 20 61 6E 64 0D 0A 70 72| e floppy and..pr
000000A0 65 73 73 20 61 6E 79 20 6B 65 79 20 74 6F 20 74| ess any key to t
000000B0 72 79 20 61 67 61 69 6E 20 2E 2E 2E 2E 20 0D 0A 00| ry again ... ..
000000C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00| .....
```

Рисунок 5 – Пример загрузочного сектора

В зоне «1» у нас указывается система хранения. Если у вас там не FAT12 или не FAT16, то идите читать методичку по ntfs.

В зоне «2» указан размер одного сектора в байтах.

В зоне «3» указано сколько секторов в одном кластере.

В зоне «4» указано количество зарезервированных секторов.  
Минимальное значение равно единице, так как загрузочный сектор считается зарезервированным.

В зоне «5» указано количество FAT-таблиц.

В зоне «6» указан размер корневой директории.

В зоне «7» указан размер одной FAT-таблицы.

Теперь стоит отметить, как правильно считывать данные из таблицы.  
Посмотрим на зону «2» с рисунка 5. Значение хранящееся в ней следующее:  
00 02. Для того, чтобы узнать какое это число, сначала нужно поменять байты местами:

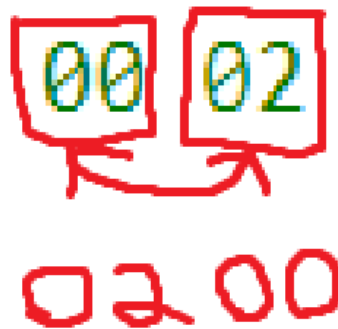


Рисунок 6 – Меняем байты местами.

Затем нужно перевести получившееся число в 10-тичную систему счисления. И получившееся число будет, тем что нам будет нужно.

**Практика.** Посмотрим на данный загрузочный сектор:

```

offset 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 EB 3C 90 6D 6B 66 73 2E 66 61 74 00 02 01 05 00| ë<mkfs.fat.....
00000010 04 00 02 00 08 F8 06 00 10 00 02 00 00 00 00 00| .....ø.....
00000020 00 00 00 00 80 00 29 25 94 4D D8 4E 4F 20 4E 41| ....).)%MONO NA
00000030 4D 45 20 20 20 20 46 41 54 31 32 20 20 20 0E 1F| ME FAT12 ..
00000040 BE 5B 7C AC 22 C0 74 0B 56 B4 0E BB 07 00 CD 10| %[|-"Àt.V´.»...í.
00000050 5E EB F0 32 E4 CD 16 CD 19 EB FE 54 68 69 73 20| ^ëð2äí.í.ëþThis
00000060 69 73 20 6E 6F 74 20 61 20 62 6F 6F 74 61 62 6C| is not a bootabl
00000070 65 20 64 69 73 6B 2E 20 20 50 6C 65 61 73 65 20| e disk. Please
00000080 69 6E 73 65 72 74 20 61 20 62 6F 6F 74 61 62 6C| insert a bootabl
00000090 65 20 66 6C 6F 70 70 79 20 61 6E 64 0D 0A 70 72| e floppy and..pr
000000A0 65 73 73 20 61 6E 79 20 6B 65 79 20 74 6F 20 74| ess any key to t
000000B0 72 79 20 61 67 61 69 6E 20 2E 2E 2E 2E 20 0D 0A 00| ry again ... ...
000000C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00| .....

```

Рисунок 7 – Загрузочный сектор

Проанализировав его, делаем вывод, что:

1. Система хранения – FAT12
2. Размер сектора – 0x0200 – 512 байт
3. Количество секторов в кластере – 0x01 – 1 сектор
4. Количество зарезервированных секторов – 0x0005 – 5 секторов
5. Количество FAT-таблиц – 0x04 – 4 штуки
6. Размер корневой директории – 0x0200 – 512 байт
7. Размер одной FAT-таблицы – 0x0006 – 6 ссекторов

Для удобства всю информацию будем записывать в поле для заметок:

Для заметок

1. Система хранения – FAT12
2. Размер сектора – 0x0200 – 512 байт
3. Количество секторов в кластере – 0x01 – 1 сектор
4. Количество зарезервированных секторов – 0x0005 – 5 секторов
5. Количество FAT-таблиц – 0x04 – 4 штуки
6. Размер корневой директории – 0x0200 – 512 байт
7. Размер одной FAT-таблицы – 0x0006 – 6 ссекторов

Рисунок 8 – Поле для заметок после анализа загрузочного сектора

На этом анализ загрузочного сектора окончен.

## Шаг 2 – нахождение корневой директории

**Теория.** Чтобы не запутаться в этом шаге важно понимать, как выглядит FAT диск в общем случае. Взглянем схему:



Рисунок 9 – FAT диск в общем случае

Из схемы видно, что корневая директория хранится сразу после FAT-таблиц, а они в свою очередь – сразу после зарезервированных секторов. Следовательно, чтобы попасть в корневую директорию нужно узнать сколько секторов занимают зарезервированные сектора и FAT-таблицы, сложить эти значения, прибавить к 0 (т. к. индексация секторов начинается с 0) и наконец перейти в сектор с номером равном получившемуся значению. С

зарезервированными секторами всё просто, берём значение, которое мы выяснили на шаге 1. Чтобы узнать сколько занимают FAT-таблицы нужно умножить их количество на размер одной из них. Для упрощения можно сказать, что первый сектор корневой папке считается по следующей формуле:

Количество зарезервированных секторов + количество FAT-таблиц \* Размер одной FAT-таблицы.

**Практика.** Считаем по формуле сектор корневой папки:  $5 + 4 * 6 = 29$ . Перейдём в 29 сектор:

```

offset 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00003A00 E5 55 4E 4B 20 20 20 20 20 20 10 08 00 AC 82 | åUNK      ...~
00003A10 D1 56 D1 56 00 00 AC 82 D1 56 02 00 00 00 00 | ÑVÑV...~ÑV.....
00003A20 41 68 00 6E 00 49 00 57 00 6F 00 0F 00 9D 00 | Ah.n.I.W.o...~..
00003A30 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF | üüüüüüüüüü..üüüü
00003A40 48 4E 49 57 4F 20 20 20 20 20 20 10 00 00 | HNIWO      ...~
00003A50 D1 56 D1 56 00 00 AC 82 D1 56 2D 00 00 00 00 | ÑVÑV...~ÑV-.....
00003A60 41 64 00 65 00 66 00 55 00 62 00 0F 00 EA 00 | Ad.e.f.U.b...ê..
00003A70 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF | üüüüüüüüüü..üüüü
00003A80 44 45 46 55 42 20 20 20 20 20 20 10 00 00 | DEFUB      ...~
00003A90 D1 56 D1 56 00 00 AC 82 D1 56 05 00 00 00 00 | ÑVÑV...~ÑV.....
00003AA0 41 4F 00 42 00 61 00 56 00 59 00 0F 00 7E 00 | AO.B.a.V.Y...~..
00003AB0 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF | üüüüüüüüüü..üüüü
00003AC0 4F 42 41 56 59 20 20 20 20 20 20 20 00 00 | OBAVY      ...~
00003AD0 D1 56 D1 56 00 00 AC 82 D1 56 0C 00 89 09 00 | ÑVÑV...~ÑV...~...
00003AE0 41 78 00 6F 00 4F 00 4F 00 56 00 0F 00 3F 00 | Ax.o.O.O.V...?..
00003AF0 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF | üüüüüüüüüü..üüüü
00003B00 58 4F 4F 4F 56 20 20 20 20 20 20 20 00 00 | XOOOV      ...~
00003B10 D1 56 D1 56 00 00 AD 82 D1 56 46 00 E9 0A 00 | ÑVÑV...~ÑVF.é...
00003B20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....

```

Рисунок 10 – Корневая директория

### Шаг 3 – нахождение начала области данных

**Теория.** После того, как мы нашли начало корневой директории, надо найти начало области данных, по-другому первый сектор области данных. Если мы вспомним, как выглядит FAT-диск в общем случае, то поймём, что формула начала области данных следующая: Первый сектор корневой директории + размер корневой директории в секторах.



Формула размера корневой директории, следующая: Размер корневой директории в байтах \* 32 / Размер одного сектора в байтах.

Число 32 – это количество строк в секторе, можете посчитать сами и увидите, что их 32. Так как количество строк никогда не меняется, то 32 – это константа.

Чтобы проверить, что мы действительно в корневой директории, нужно вернуться на сектор назад и проверить все ли значения там нулевые. То есть, если начало области равно 255, то нужно переключиться на сектор 254 и удостоверится, что он заполнен нулями.

**Практика.** Сначала найдём размер корневой папки в секторах:  $512 * 32 / 512 = 32$ . Теперь найдём область начала данных:  $29 + 32 = 61$ . Перейдём в сектор 61:

```
offset 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00007A00 2E 20 20 20 20 20 20 20 20 20 20 10 00 00 AC 82| .          ...~
00007A10 D1 56 D1 56 00 00 AC 82 D1 56 02 00 00 00 00 00| ÑVÑV...ÑV.....
00007A20 2E 2E 20 20 20 20 20 20 20 20 20 20 10 00 00 AC 82| ..          ...~
00007A30 D1 56 D1 56 00 00 AC 82 D1 56 00 00 00 00 00 00| ÑVÑV...ÑV.....
00007A40 E5 6C 00 67 00 76 00 66 00 56 00 0F 00 CC 00 00| âl.g.v.f.v...Ï..
00007A50 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF| ýýýýýýýýýý...ýýýý
00007A60 E5 55 00 5A 00 77 00 70 00 63 00 0F 00 CC 4B 00| âU.Z.w.p.c...ÏK.
00007A70 5A 00 56 00 51 00 6E 00 67 00 00 00 75 00 63 00| Z.V.Q.n.g...u.c.
00007A80 E5 42 00 45 00 59 00 7A 00 4F 00 0F 00 CC 6A 00| âB.E.Y.Z.O...Ïj.
00007A90 68 00 4D 00 47 00 55 00 45 00 00 00 56 00 41 00| h.M.G.U.E...V.A.
00007AA0 E5 54 00 5A 00 76 00 73 00 64 00 0F 00 CC 67 00| âT.Z.v.s.d...Ïg.
00007AB0 52 00 6E 00 4F 00 57 00 7A 00 00 00 6F 00 4D 00| R.n.O.W.z...o.M.
00007AC0 E5 5A 56 53 44 47 7E 31 20 20 20 20 00 00 AC 82| âZVSDG~1      ...~
00007AD0 D1 56 D1 56 00 00 AC 82 D1 56 03 00 00 02 00 00| ÑVÑV...ÑV.....
00007AE0 E5 50 00 61 00 52 00 72 00 6F 00 0F 00 16 72 00| âP.a.R.r.o....r.
00007AF0 6C 00 51 00 79 00 69 00 46 00 00 00 50 00 4C 00| l.Q.y.i.F...P.L.
00007B00 E5 78 00 61 00 75 00 6A 00 73 00 0F 00 16 74 00| âx.a.u.j.s...t.
00007B10 46 00 61 00 55 00 67 00 75 00 00 00 4A 00 6E 00| F.a.U.g.u...J.n.
00007B20 E5 50 00 48 00 46 00 41 00 75 00 0F 00 16 50 00| âP.H.F.A.u....P.
00007B30 67 00 61 00 54 00 44 00 79 00 00 00 64 00 55 00| g.a.T.D.y...d.U.
00007B40 E5 48 46 41 55 50 7E 31 20 20 20 20 00 00 AC 82| âHFAUP~1      ...~
00007B50 D1 56 D1 56 00 00 AC 82 D1 56 04 00 00 02 00 00| ÑVÑV...ÑV.....
00007B60 E5 61 00 54 00 73 00 46 00 59 00 0F 00 83 6E 00| âa.T.s.F.Y...ñn.
00007B70 47 00 54 00 00 00 FF FF FF FF 00 00 FF FF FF FF| G.T...ýýýý...ýýýý
00007B80 E5 71 00 47 00 7A 00 78 00 66 00 0F 00 83 70 00| âq.G.z.x.f...ñp.
00007B90 53 00 50 00 41 00 5A 00 4F 00 00 00 55 00 4F 00| S.P.A.Z.O...U.O.
00007BA0 E5 56 00 62 00 75 00 70 00 70 00 0F 00 83 73 00| âV.b.u.p...ñs.
00007BB0 75 00 76 00 48 00 46 00 41 00 00 00 4C 00 73 00| u.v.H.F.A...L.s.
00007BC0 E5 77 00 4D 00 56 00 51 00 51 00 0F 00 83 52 00| âw.M.V.Q.Q...ñR.
00007BD0 6D 00 53 00 72 00 62 00 4B 00 00 00 78 00 50 00| m.S.r.b.K...x.P.
00007BE0 E5 4D 56 51 51 52 7E 31 20 20 20 20 00 00 AC 82| âMVQQR~1      ...~
00007BF0 D1 56 D1 56 00 00 AC 82 D1 56 05 00 00 02 00 00| ÑVÑV...ÑV.....
```

Рисунок 11 – Начало области данных

## Шаг 4 – нахождение нужного файла

**Теория.** Для того чтобы найти нужный файл нужно пройти полный путь до него. Сначала нужно посмотреть какой у нас путь по заданию. В нём опущена корневая директория. То есть полный путь выглядит так:

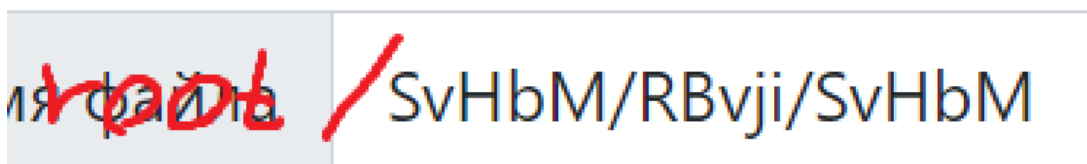


Рисунок 12 – Пример полного пути к файлу

Для того, чтобы перемещаться по данным, необходимо в папке найти запись, отвечающую за директорию, в которую нам надо переместиться. Вот пример подобной записи:

```
0001FE40 53 56 48 42 4D 20 20 20 20 20 20 10 00 00 EC 5E | SVHBM 1 ...i^
0001FE50 D1 56 D1 56 00 00 EC 5E D1 56 4C 00 00 00 00 00 ÑVÑV..i^ÑVL.....
          3          4
```

Рисунок 13 – Пример записи директории

В зоне «1» указано имя директории, за которую отвечает запись.

В зоне «2» указан атрибут. Если он равен 10, то это папка, если он равен 20, то это файл.

В зоне «3» указан номер первого кластера директории в 16-тиричной системе счисления.

В зоне «4» указан размер директории в 16-тиричной системе счисления. У папок он всегда равен нулю.

Далее нужно понять несколько вещей. Первое, кластер – это объединение нескольких смежных секторов. То есть один кластер состоит из нескольких секторов, из скольких именно мы узнавали в загрузочном секторе.

Все данные в FAT хранятся в кластерах. Индексация кластеров начинается с 2. То есть нулевой кластер будет иметь номер 2, первый кластер – 3, второй кластер – 4, и так далее. Счёт кластеров начинается от начала данных.

Таким образом, чтобы попасть в какую-либо директорию необходимо взять номер её первого кластера, вычесть из него 2, умножить на количество секторов в кластере и прибавить к началу данных. Таким образом мы получим первый сектор директории. То есть формула перемещения такая: (Номер кластера – 2) \* Количество секторов в кластере + Начало данных.

**Практика.** Переходим в конечную директорию. Затем посмотрим на путь к файлу:



Рисунок 14 – Путь к файлу

Находим в корневой директории запись, отвечающую за папку defUb:

```
00003A80 44 45 46 55 42 20 20 20 20 20 10 00 00 AC 82| DEFUB      ...~
00003A90 D1 56 D1 56 00 00 AC 82 D1 56 05 00 00 00 00| ÑVÑV...~ÑV.....
```

Рисунок 15 – Запись папки defUb

Отсюда выясняю первый кластер этой папки. Он равен  $0x0005 = 5$ . Считаем первый сектор этого кластера:  $(5-2) * 1 + 61 = 64$ . Перейдём в этот сектор:

```

offset 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00008000 2E 20 20 20 20 20 20 20 20 20 20 10 00 00 AC 82| .          ...~
00008010 D1 56 D1 56 00 00 AC 82 D1 56 05 00 00 00 00 00| ÑVÑV...~ÑV.....
00008020 2E 2E 20 20 20 20 20 20 20 20 20 10 00 00 AC 82| ..          ...~
00008030 D1 56 D1 56 00 00 AC 82 D1 56 00 00 00 00 00 00| ÑVÑV...~ÑV.....
00008040 41 68 00 6E 00 49 00 57 00 6F 00 0F 00 9D 00 00| Ah.n.I.W.o...~..
00008050 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF| üüüüüüüüüü..üüüü
00008060 48 4E 49 57 4F 20 20 20 20 20 20 10 00 00 AC 82| HNIWO          ...~
00008070 D1 56 D1 56 00 00 AC 82 D1 56 28 00 00 00 00 00| ÑVÑV...~ÑV(.....
00008080 41 64 00 65 00 66 00 55 00 62 00 0F 00 EA 00 00| Ad.e.f.U.b...ê..
00008090 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF| üüüüüüüüüü..üüüü
000080A0 44 45 46 55 42 20 20 20 20 20 20 20 00 00 AC 82| DEFUB          ...~
000080B0 D1 56 D1 56 00 00 AC 82 D1 56 13 00 81 0B 00 00| ÑVÑV...~ÑV...~...
000080C0 41 78 00 6F 00 4F 00 4F 00 56 00 0F 00 3F 00 00| Ax.o.O.O.V...?..
000080D0 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF| üüüüüüüüüü..üüüü
000080E0 58 4F 4F 4F 56 20 20 20 20 20 20 10 00 00 AD 82| XOOOV          ...~
000080F0 D1 56 D1 56 00 00 AD 82 D1 56 06 00 00 00 00 00| ÑVÑV...~ÑV.....
00008100 41 4F 00 42 00 61 00 56 00 59 00 0F 00 7E 00 00| AO.B.a.V.Y...~..
00008110 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF| üüüüüüüüüü..üüüü
00008120 4F 42 41 56 59 20 20 20 20 20 20 20 00 00 AD 82| OBAVY          ..~
00008130 D1 56 D1 56 00 00 AD 82 D1 56 59 00 5B 0B 00 00| ÑVÑV...~ÑVY.[ ...
-----

```

Рисунок 16 – Содержимое папки defUb

Теперь ищем запись, отвечающую за папку хоOOV:

```

000080E0 58 4F 4F 4F 56 20 20 20 20 20 20 10 00 00 AD 82| XOOOV          ...~
000080F0 D1 56 D1 56 00 00 AD 82 D1 56 06 00 00 00 00 00| ÑVÑV...~ÑV.....

```

Рисунок 17 – Запись папки хоOOV

Отсюда выясняю первый кластер этой папки. Он равен  $0x0006 = 6$ . Считаем первый сектор этого кластера:  $(6-2) * 1 + 61 = 65$ . Перейдём в этот сектор:

```

offset 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00008200 2E 20 20 20 20 20 20 20 20 20 20 10 00 00 AD 82 | .          ...
00008210 D1 56 D1 56 00 00 AD 82 D1 56 06 00 00 00 00 00 | ÑVÑV..ÑV.....
00008220 2E 2E 20 20 20 20 20 20 20 20 20 20 10 00 00 AD 82 | ..          ...
00008230 D1 56 D1 56 00 00 AD 82 D1 56 05 00 00 00 00 00 | ÑVÑV..ÑV.....
00008240 41 68 00 6E 00 49 00 57 00 6F 00 0F 00 9D 00 00 | Ah.n.I.W.o...
00008250 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF | üüüüüüüüüü..üüüü
00008260 48 4E 49 57 4F 20 20 20 20 20 20 20 10 00 00 AD 82 | HNIWO      ...
00008270 D1 56 D1 56 00 00 AD 82 D1 56 35 00 00 00 00 00 | ÑVÑV..ÑV5.....
00008280 41 78 00 6F 00 4F 00 4F 00 56 00 0F 00 3F 00 00 | Ax.o.O.O.V...?..
00008290 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF | üüüüüüüüüü..üüüü
000082A0 58 4F 4F 4F 56 20 20 20 20 20 20 20 10 00 00 AD 82 | XOOOV      ...
000082B0 D1 56 D1 56 00 00 AD 82 D1 56 17 00 00 00 00 00 | ÑVÑV..ÑV.....
000082C0 41 4F 00 42 00 61 00 56 00 59 00 0F 00 7E 00 00 | AO.B.a.V.Y...~..
000082D0 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF | üüüüüüüüüü..üüüü
000082E0 4F 42 41 56 59 20 20 20 20 20 20 20 20 00 00 AD 82 | OBAVY      ..
000082F0 D1 56 D1 56 00 00 AD 82 D1 56 6F 00 E8 08 00 00 | ÑVÑV..ÑVo.è...
00008300 41 64 00 65 00 66 00 55 00 62 00 0F 00 EA 00 00 | Ad.e.f.U.b...ê..
00008310 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF | üüüüüüüüüü..üüüü
00008320 44 45 46 55 42 20 20 20 20 20 20 20 20 00 00 AD 82 | DEFUB      ..
00008330 D1 56 D1 56 00 00 AD 82 D1 56 03 00 6E 0B 00 00 | ÑVÑV..ÑV..n...
00008340 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....

```

Рисунок 18 – Содержимое папки хоOOV

Теперь ищем запись, отвечающую за файл defUb:

```

00008320 44 45 46 55 42 20 20 20 20 20 20 20 20 00 00 AD 82 | DEFUB      ..
00008330 D1 56 D1 56 00 00 AD 82 D1 56 03 00 6E 0B 00 00 | ÑVÑV..ÑV..n...

```

Рисунок 19 – Запись файла defUb

Переходим к следующему шагу.

## Шаг 5 – анализ файла

**Теория.** После того как мы нашли запись, отвечающую за файл, необходимо её проанализировать.

Первое, нужно посчитать размер файла в байтах. Для этого вспоминаем зону, где хранится размер (посмотрите теорию предыдущего шага если забыли) и переводим значение, хранящееся там в десятичную.

Второе, что нужно сделать, это посчитать в скольких кластерах хранится файл. Для этого нужно посчитать сколько весит один кластер и поделить размер файла на это число. Нам известно сколько секторов в кластере и сколько весит один сектор, следовательно формула размера кластера будет следующая: Размер сектора \* Количество секторов в кластере. Далее делим размер файла на это число и округляем вверх. Полученный результат будет количеством кластеров, которые занимает файл.

Третье, что нужно сделать – это выписать номер первого кластера.

**Практика.** Из записи файла (рисунок 19) видно, что его размер равен  $0x00000B6E = 2926$  байт, а номер первого кластера –  $0x0003 = 3$ .

Теперь посчитаем размер одного кластера:  $512 * 1 = 512$  байт. Посчитаем сколько кластеров занимает файл:  $2926 / 512 = 5,72 \sim 6$  кластеров занимает файл.

Переходим к следующему шагу.

## Шаг 6 – поиск всех кластеров файла

**Теория.** Структура хранения FAT является односвязным списком. То есть каждый кластер хранит ссылку на следующий кластер. FAT-таблицы хранят информацию о том, на какой кластер указывает каждый кластер.

Вне зависимости от количества FAT-таблиц они всегда будут одинаковыми. Далее для удобства будем работать с первой таблицей. Чтобы найти её нам надо прибавить к 0 количество зарезервированных секторов (вспоминаем теорию шага 2).



В FAT-таблице в FAT12 каждый кластер занимает 1,5 байта (в FAT16 2 байта). То есть если поделить таблицу на кластеры она будет выглядеть так:

```
offset 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000E00 F8 FF FF 00 60 03 56 F0 FF 00 F0 FF 3E 50 07 00| øÿÿ.`.vðÿ.ðÿ>P..
00000E10 F0 FF 14 F0 00 00 E0 01 38 F0 FF 22 00 00 67 A0| ðÿ.ð..à.8ðÿ"..g
```

Рисунок 20 – FAT-таблица FAT12 разделённая на кластеры

В FAT-таблицах нумерация кластеров, совпадает с их нумерацией в области данных. То есть 10 кластер в области данных, также будет 10 кластером в FAT-таблице.

Чтобы найти на что указывает какой-либо кластер мы будем прибавлять его номер к смещению таблицы, переводить результат в 16-тиричную систему счисления и искать результат в таблице. Сначала найдём смещение таблицы, оно находится здесь:

```
offset 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000E00 F8 FF FF 00 60 03 56 F0 FF 00 F0 FF 3E 50 07 00| øÿÿ.`.vðÿ.ðÿ>P..
00000E10 F0 FF 14 F0 00 00 E0 01 38 F0 FF 22 00 00 67 A0| ðÿ.ð..à.8ðÿ"..g
00000E20 03 FF 0F 00 65 A0 04 31 00 00 FF DF 02 93 00 00| .ÿ..e .1..ÿß.ð..
00000E30 2A 00 00 25 00 00 44 B0 08 87 F0 FF 00 20 09 53| *..%..D°.ðÿ. .s
00000E40 00 00 70 E0 02 71 00 00 BC D0 07 54 30 11 00 90| ..pà.q..%ð.Tð..ð
00000E50 03 6A 00 00 9A F0 FF 5B 30 0B 00 80 09 7A D0 06| .j..ðÿ[ð...ð.zð.
00000E60 00 F0 FF 79 F0 FF FF 0F 00 8A E0 08 4E 00 00 74| .ðÿvðÿÿ...ðà.N..t
```

Рисунок 21 – Смещение FAT-таблицы

Переводим данное число в десятичную, прибавляем к нему номер кластера умноженный на 1,5 (В FAT16 умножаем на 2). Если число получилось целое, то просто переводим в 16-тиричную, если же оно получилось дробное, то округляем в меньшую сторону, запоминаем данный факт и переводим в 16-тиричную. Полученное число делим подобным образом:

0x E27

Рисунок 22 – Деление адреса

Левая часть отвечает за строку, а правая за столбец. То есть согласно рисунку 14 наш кластер указывает на следующий участок:

offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000E00	F8	FF	FF	00	60	03	56	F0	FF	00	F0	FF	3E	50	07	00	øÿÿ.`.
00000E10	F0	FF	14	F0	00	00	E0	01	38	F0	FF	22	00	00	67	A0	ðÿ.ð..
00000E20	03	FF	0F	00	65	A0	04	31	00	00	FF	DF	02	93	00	00	.ÿ..e
00000E30	2A	00	00	25	00	00	44	B0	08	87	F0	FF	00	20	09	53	*..%..
00000E40	00	00	70	E0	02	71	00	00	BC	D0	07	54	30	11	00	90	..pà.q.

Рисунок 23 – Байт, на который показал адрес

Теперь если у нас FAT12 действуем следующим образом: вспоминаем был ли адрес целым или дробным. В первом случае мы выделяем следующую область:

offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000E00	F8	FF	FF	00	60	03	56	F0	FF	00	F0	FF	3E	50	07	00	øÿÿ.`.Vð
00000E10	F0	FF	14	F0	00	00	E0	01	38	F0	FF	22	00	00	67	A0	ðÿ.ð..à.
00000E20	03	FF	0F	00	65	A0	04	31	00	00	FF	DF	02	93	00	00	.ÿ..e .1
00000E30	2A	00	00	25	00	00	44	B0	08	87	F0	FF	00	20	09	53	*..%..D°

Рисунок 24 – Область выделения если адрес целый

Во втором случае выделяем следующую область:

offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000E00	F8	FF	FF	00	60	03	56	F0	FF	00	F0	FF	3E	50	07	00	øÿÿ.`.Vð
00000E10	F0	FF	14	F0	00	00	E0	01	38	F0	FF	22	00	00	67	A0	ðÿ.ð..à.
00000E20	03	FF	0F	00	65	A0	04	31	00	00	FF	DF	02	93	00	00	.ÿ..e .
00000E30	2A	00	00	25	00	00	44	B0	08	87	F0	FF	00	20	09	53	*..%..[
00000E40	00	00	70	E0	02	71	00	00	BC	D0	07	54	30	11	00	90	..pà.q.
00000E50	03	6A	00	00	9A	F0	FF	5B	30	0B	00	80	09	7A	00	06	.i..mð

Рисунок 25– Область выделения если адрес дробный

Затем разворачиваем выделенную область. Делим её пополам:



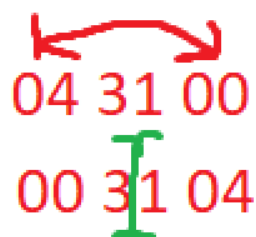


Рисунок 26 – Разбиение области

После этого, если адрес был целым, то берём правую часть, если адрес был дробным, то берём левую часть. Переводим получившееся значение в десятичную систему счисления, и это будет номером следующего кластера.

Если же у нас FAT16, то нам нужно выделить следующую область:

---

offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000E00	F8	FF	FF	00	60	03	56	F0	FF	00	F0	FF	3E	50	07	00	øÿÿ.~.
00000E10	F0	FF	14	F0	00	00	E0	01	38	F0	FF	22	00	00	67	A0	ðÿ.ð..
00000E20	03	FF	0F	00	65	A0	04	31	00	00	FF	DF	02	93	00	00	.ÿ..e
00000E30	2A	00	00	25	00	00	44	B0	08	87	F0	FF	00	20	09	53	*..%..

Рисунок 27 – Область выделения в FAT16

Затем получившееся значение надо развернуть и перевести в десятичную. Это будет номером следующего кластера.

Подобный процесс надо повторить для каждого кластера файла.

Последний кластер файла всегда должен указывать на «FFF» в случае FAT12 и на «FF» в случае FAT16.

**Практика.** Найдём сектор первой FAT-таблицы:  $0+5 = 5$ . Перейдём в этот сектор:

```

offset 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000A00 F8 FF FF 00 80 04 76 F0 FF FF 0F 00 FF AF 00 41| øÿÿ.▯.vðÿÿ..ÿ~.A
00000A10 00 00 68 F0 FF 1B F0 01 00 00 02 4C A0 06 00 30| ..hðÿ.ð....L ..ø
00000A20 02 24 F0 FF 00 80 0F 54 40 07 00 40 09 2C 20 06| . $ðÿ.▯.T@...@., .
00000A30 87 00 00 4E 40 03 FF 0F 00 2F C0 07 FF 0F 00 43| ▯..N@.ÿ../À.ÿ..C
00000A40 C0 09 A0 F0 FF 00 90 04 50 A0 07 00 D0 04 FF FF| À. ðÿ.▯.P ..Ð.ÿÿ
00000A50 FF 00 80 03 97 70 0D 00 F0 FF 3F 20 05 00 00 10| ÿ.▯.▯p..ðÿ? ....
00000A60 6E 40 04 00 B0 0C 5F 00 07 55 00 00 5D E0 0A 00| n@...°. _...U..]à..
00000A70 00 00 65 60 08 D4 00 00 5A F0 FF 7B 00 00 FF BF| ..e`.ô...Zðÿ{...ÿ¿
00000A80 08 5E C0 0A 00 00 08 E9 70 0C 00 A0 0C B0 F0 07| .^À....ép... °ð.
00000A90 00 B0 0F 83 50 08 00 80 07 93 80 0D 6C 00 00 B8| .°.▯P...▯.▯▯.l...
00000AA0 50 0A BF 00 00 82 E0 07 FF 0F 00 FF FF FF FF 0F| P.¿...▯à.ÿ..ÿÿÿÿ.
00000AB0 00 8A E0 0D FF 0F 00 90 10 0F B6 00 00 8C F0 FF| .▯à.ÿ..▯...¶...▯ðÿ
00000AC0 98 00 00 8F D0 09 00 10 0A 88 90 09 BD 00 00 E3| ▯...▯▯....▯▯.‰...ã
00000AD0 90 0A B1 F0 FF 00 F0 FF A8 F0 0D 00 60 09 C3 00| ▯.±ðÿ.ðÿ"ð...` .Ã.
00000AE0 00 A6 50 0D C2 00 0D 00 90 0F A2 F0 09 00 F0 FF| .!P.Â...▯.¢ð..ðÿ
00000AF0 C5 A0 0A C1 00 00 FF 2F 0D FF 0F 00 DB B0 0B AD| Å .Á..ÿ/.ÿ...Û°.
00000B00 00 00 B4 A0 0B B2 00 00 BE 30 0D C6 00 00 FF 0F| ..´ .²...‰ø.Æ..ÿ.
00000B10 0E FF 0F 00 F4 70 0E FF 3F 0F 00 80 0E DC E0 0F| .ÿ..ôp.ÿ?...▯.Ûà.
00000B20 00 90 0D E4 F0 FF 00 80 0C FF 5F 0F CC 00 00 E2| .▯.äðÿ.▯.ÿ_.Ì..â
00000B30 F0 FF FF 0F 00 CF F0 FF 10 01 00 FF FF FF FF 7F| ðÿÿ...İðÿ...ÿÿÿÿ
00000B40 0F 00 F0 0E EC F0 FF 00 60 0E FF 0F 00 13 F1 FF| ..ð.ìðÿ.`.ÿ...ñÿ
00000B50 F0 00 00 09 D1 0E 08 01 00 FF 5F 10 FF BF 0E 00| ð...Ñ....ÿ_.ÿ¿..
00000B60 C0 0F FF FF FF 00 F0 FF 21 F1 FF 00 F0 FF 01 C1| À.ÿÿÿ.ðÿ!ñÿ.ðÿ.Á
00000B70 10 00 E0 10 02 B1 10 00 D0 0F FF FF FF FF 0F 00| ..à...±..Ð.ÿÿÿÿ..
00000B80 FF 4F 10 FF 0F 00 FF 6F 10 FF 0F 00 FF FF FF 00| ÿo.ÿ..ÿo.ÿ..ÿÿÿ.
00000B90 F0 FF 0D F1 FF FF 0F 00 FF 0F 00 00 C0 12 00 00| ðÿ.ñÿÿ..ÿ...À...
-----

```

Рисунок 28 – FAT-таблица

Сразу посмотрим, чему равно её смещение:  $0xA00 = 2560$ .

Найдём на что указывает первый кластер:  $2560 + 3 \cdot 1.5 = 2564,5 = 2564 = 0xA04$ . Найдём эту точку и её выделяемую область: 00 80 04. Проводим преобразования и берём левую часть, получаем:  $0x048 = 72$ . 72 – номер второго кластера.

Найдём на что указывает второй кластер:  $2560 + 72 \cdot 1.5 = 2668 = 0xA6C$ . Найдём эту точку и её выделяемую область: 5D E0 0A. Проводим

преобразования и берём левую часть, получаем:  $0x05D = 93$ . 93 – номер третьего кластера.

Проведём подобные вычисления для каждого сектора и получаем:

3 – первый кластер

72 – второй кластер

93 – третий кластер

202 – четвёртый кластер

226 – пятый кластер

265 – шестой кластер

### **Шаг 7 – выгрузка всех кластера файла**

**Теория.** Теперь, когда мы узнали все кластеры файла, нам надо их все скачать. Для того, чтобы скачать кластер, надо выгрузить каждый его сектор. Делается это следующим образом. Сначала переходим в нужный кластер (подобно тому, как мы перемещались по директориям в шаге 4). Затем выгружаем каждый его сектор. То есть если первый сектор в кластере равен 255, а количество секторов в кластере равно 4, то надо выгрузить сектора: 255, 256, 257, 258.

Подобный процесс надо повторить для каждого кластера файла.

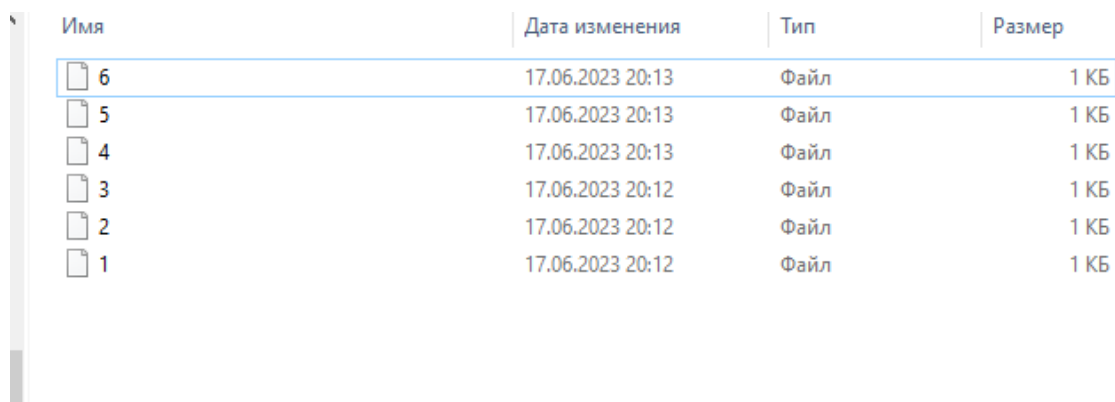
**Практика.** Переходим в каждый кластер и скачиваем каждый сектор, используя кнопку выгрузить.

### **Шаг 8 – подготовка файлов**

**Теория.** Теперь, когда мы выгрузили все сектора, надо их подготовить. Это довольно просто и быстро. Для начала переименуем все файлы в порядке их скачки. То есть 1 скачанный файл будет называться «1», второй – «2» и так

далее. Далее поместим все файлы в какую-нибудь папку, чтобы нам не мешали сторонние файлы.

**Практика.** Переименуем все скачанные файлы:



Имя	Дата изменения	Тип	Размер
6	17.06.2023 20:13	Файл	1 КБ
5	17.06.2023 20:13	Файл	1 КБ
4	17.06.2023 20:13	Файл	1 КБ
3	17.06.2023 20:12	Файл	1 КБ
2	17.06.2023 20:12	Файл	1 КБ
1	17.06.2023 20:12	Файл	1 КБ

Рисунок 29 – Переименованные файлы

## Шаг 9 – вычисление контрольной суммы

**Теория.** Теперь надо объединить все файлы в один с помощью команды *cat*. Затем проверяем размер получившегося файла с помощью команды *ls -l*. После этого обрезаем его до нужного размера, который мы узнали в шаге 5. Делаем это с помощью команды *dd if=название объединённого файла of=final bs = 1 count=\$((нужный размер))*. Далее вычисляем контрольную сумму с помощью команды *md5sum final*.

**Практика.** Выполняем все команды:

```
pashkov@DESKTOP-K0RU19J:/mnt/c/Users/Shomnt/os$ ls
1 2 3 4 5 6
pashkov@DESKTOP-K0RU19J:/mnt/c/Users/Shomnt/os$ cat 1 2 3 4 5 6 > out
pashkov@DESKTOP-K0RU19J:/mnt/c/Users/Shomnt/os$ ls -l
total 4
-rwxrwxrwx 1 pashkov pashkov 512 Jun 17 20:12 1
-rwxrwxrwx 1 pashkov pashkov 512 Jun 17 20:12 2
-rwxrwxrwx 1 pashkov pashkov 512 Jun 17 20:12 3
-rwxrwxrwx 1 pashkov pashkov 512 Jun 17 20:13 4
-rwxrwxrwx 1 pashkov pashkov 512 Jun 17 20:13 5
-rwxrwxrwx 1 pashkov pashkov 512 Jun 17 20:13 6
-rwxrwxrwx 1 pashkov pashkov 3072 Jun 17 20:14 out
pashkov@DESKTOP-K0RU19J:/mnt/c/Users/Shomnt/os$ dd if=out of=final bs=1 count=$((2926))
2926+0 records in
2926+0 records out
2926 bytes (2.9 kB, 2.9 KiB) copied, 1.71937 s, 1.7 kB/s
pashkov@DESKTOP-K0RU19J:/mnt/c/Users/Shomnt/os$ ls -l
total 8
-rwxrwxrwx 1 pashkov pashkov 512 Jun 17 20:12 1
-rwxrwxrwx 1 pashkov pashkov 512 Jun 17 20:12 2
-rwxrwxrwx 1 pashkov pashkov 512 Jun 17 20:12 3
-rwxrwxrwx 1 pashkov pashkov 512 Jun 17 20:13 4
-rwxrwxrwx 1 pashkov pashkov 512 Jun 17 20:13 5
-rwxrwxrwx 1 pashkov pashkov 512 Jun 17 20:13 6
-rwxrwxrwx 1 pashkov pashkov 2926 Jun 17 20:15 final
-rwxrwxrwx 1 pashkov pashkov 3072 Jun 17 20:14 out
pashkov@DESKTOP-K0RU19J:/mnt/c/Users/Shomnt/os$ md5sum final
1976be81969903a4275eb18c90d622dc final
pashkov@DESKTOP-K0RU19J:/mnt/c/Users/Shomnt/os$ _
```

Рисунок 30 – Выполнение команд

Копируем контрольную сумму: 1976be81969903a4275eb18c90d622dc

Вставляем её в поле ответа и нажимаем отправить и получаем уведомление:

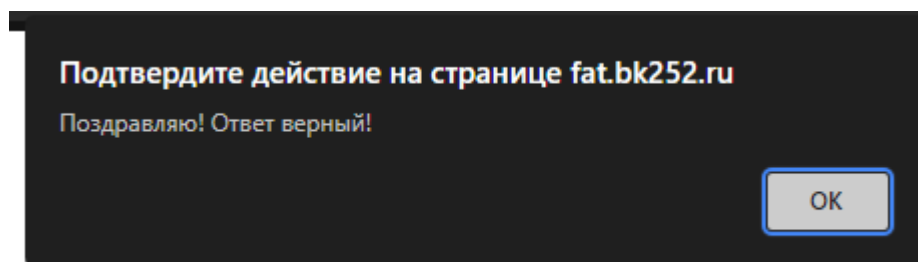


Рисунок 31 – Уведомление от сайта