

## Дай мне таблицу для подбора fat\_sz и data\_sectors

☒ Думал 132 секунд ▾

Мы будем использовать итеративный подход для расчета `fat_sz` и `data_sectors`. Основная сложность в том, что размер FAT (`fat_sz`) зависит от количества кластеров, которое в свою очередь зависит от размера области данных, а область данных зависит от размера FAT.

Для FAT16 и FAT12 (FAT32 рассмотрим отдельно) можно использовать следующий алгоритм:

1. Задаем начальные параметры: общее количество секторов (`total_sectors`), размер сектора (`bytes_per_sector`), количество зарезервированных секторов (`reserved_sectors`), количество FAT-таблиц (`num_fats`), количество записей в корневом каталоге (`root_dir_entries`, только для FAT12/16) и размер кластера в секторах (`sectors_per_cluster`).
2. Для FAT12/16: вычисляем размер корневого каталога в секторах:  

$$\text{root\_dir\_sectors} = ((\text{root\_dir\_entries} * 32) + (\text{bytes\_per\_sector} - 1)) / \text{bytes\_per\_sector}$$
3. Обозначим:  

$$\text{data\_sectors} = \text{total\_sectors} - \text{reserved\_sectors} - \text{root\_dir\_sectors} - (\text{num\_fats} * \text{fat\_sz})$$
 Но `fat_sz` неизвестно. Нам нужно найти такое `fat_sz`, чтобы оно соответствовало количеству кластеров, которое поместится в `data_sectors`.
4. Количество кластеров =  $\text{data\_sectors} / \text{sectors\_per\_cluster}$ .
5. Размер FAT (в секторах) зависит от количества кластеров и типа FAT:  
 Для FAT16:  $\text{fat\_sz} = (\text{количество\_кластеров} * 2 + (\text{bytes\_per\_sector} - 1)) / \text{bytes\_per\_sector}$   
 Для FAT12:  $\text{fat\_sz} = (\text{количество\_кластеров} * 3 + 1) / (2 * \text{bytes\_per\_sector})$  (с округлением вверх)
6. Поскольку `fat_sz` и `data_sectors` зависят друг от друга, мы можем использовать итерационный метод:  
 Шаг 0: Предположим, что `fat_sz` = 0.  
 Шаг 1: Вычислить `data_sectors` =  $\text{total\_sectors} - \text{reserved\_sectors} - \text{root\_dir\_sectors} - (\text{num\_fats} * \text{текущее\_fat\_sz})$   
 Шаг 2: Вычислить количество кластеров =  $\text{data\_sectors} / \text{sectors\_per\_cluster}$ .  
 Шаг 3: Вычислить новое `fat_sz` по формуле для выбранного типа FAT.  
 Шаг 4: Если новое `fat_sz` не равно текущему, то обновить `текущее_fat_sz` и повторить с шага 1.  
 Итерации продолжаются до тех пор, пока `fat_sz` не перестанет меняться (обычно сходится за 2-3 итерации).
7. Для FAT32 корневой каталог находится в области данных, поэтому `root_dir_sectors` = 0, и мы используем другую формулу для `fat_sz`.  
 Для FAT32:  

$$\text{data\_sectors} = \text{total\_sectors} - \text{reserved\_sectors} - (\text{num\_fats} * \text{fat\_sz32})$$

$$\text{количество\_кластеров} = \text{data\_sectors} / \text{sectors\_per\_cluster}$$

$$\text{fat\_sz32} = (\text{количество\_кластеров} * 4 + (\text{bytes\_per\_sector} - 1)) / \text{bytes\_per\_sector}$$
 Итерации аналогичны, но без учета корневого каталога.
8. Однако, для FAT32 существует дополнительное условие: количество кластеров должно быть не менее 65525.  
 Также, для FAT12/16 есть свои ограничения на количество кластеров.
9. В процессе подбора может оказаться, что при выбранном размере кластера (`sectors_per_cluster`) не получается уложиться в ограничения по количеству кластеров для выбранного типа FAT. Тогда нужно увеличить размер кластера (если это возможно) или выбрать другой тип FAT.
10. Также, после вычислений нужно проверить, что количество кластеров соответствует выбранному типу FAT.
11. Для FAT12/16: если количество кластеров меньше 4085, то это FAT12, иначе если меньше 65525, то FAT16, иначе FAT32.

12. Но на практике, иногда для томов чуть больше 512 Мб используют FAT16 с большими кластерами (64 сектора по 512 байт = 32 Кб), но это не поддерживается некоторыми системами. Обычно для FAT16 максимальный размер тома с 32Кб кластерами - 2 Гб.
13. Для FAT32 минимальный размер тома - 65525 кластеров. При размере сектора 512 байт и кластере 512 байт (1 сектор) это около 32 Мб. Но обычно для FAT32 используют том больше 512 Мб.
14. Приведем таблицу для подбора размера кластера в зависимости от размера тома (по рекомендациям Microsoft):  
 Для FAT12 (дискеты):  
 3.5" 1.44 Мб: sectors\_per\_cluster = 1, размер кластера = 512 байт.  
 Для FAT16:  
 от 0 до 32 Мб: spc = 1 (512 байт)  
 от 32 до 64 Мб: spc = 2 (1 Кб)  
 от 64 до 128 Мб: spc = 4 (2 Кб)  
 от 128 до 256 Мб: spc = 8 (4 Кб)  
 от 256 до 512 Мб: spc = 16 (8 Кб)  
 от 512 Мб до 1 Гб: spc = 32 (16 Кб)  
 от 1 Гб до 2 Гб: spc = 64 (32 Кб)  
 Для FAT32:  
 от 512 Мб до 8 Гб: spc = 8 (4 Кб)  
 от 8 Гб до 16 Гб: spc = 16 (8 Кб)  
 от 16 Гб до 32 Гб: spc = 32 (16 Кб)  
 свыше 32 Гб: spc = 64 (32 Кб)
15. Однако, для FAT32 при томах больше 32 Гб некоторые ОС (Windows) не позволяют форматировать стандартными средствами, но технически это возможно.
16. Приведем пример алгоритма для FAT16:  
 Дано:  
 total\_sectors = 409600 (200 Мб, при 512 байт/сектор)  
 bytes\_per\_sector = 512  
 reserved\_sectors = 1  
 num\_fats = 2  
 root\_dir\_entries = 512  
 sectors\_per\_cluster = 4 (2 Кб кластер)  
 Итерация 0:  
 root\_dir\_sectors =  $(512 * 32 + 511) / 512 = 32$   
 fat\_sz = 0  
 data\_sectors =  $409600 - 1 - 32 - 2 * 0 = 409567$   
 total\_clusters =  $409567 / 4 = 102391$   
 Но для FAT16 максимальное количество кластеров 65524, а у нас 102391 - это слишком много.  
 Значит, нужно увеличить размер кластера.  
 Попробуем sectors\_per\_cluster = 32 (16 Кб кластер):  
 data\_sectors =  $409600 - 1 - 32 - 2 * 0 = 409567$   
 total\_clusters =  $409567 / 32 = 12798$  (подходит для FAT16, т.к. меньше 65524)  
 Теперь вычисляем fat\_sz для FAT16:  
 fat\_sz =  $(12798 * 2 + 511) / 512 = (25596 + 511) / 512 = 26107 / 512 = 51$  (округление вверх)  
 Итерация 1:  
 data\_sectors =  $409600 - 1 - 32 - 2 * 51 = 409600 - 1 - 32 - 102 = 409465$   
 total\_clusters =  $409465 / 32 = 12795$   
 fat\_sz\_new =  $(12795 * 2 + 511) / 512 = (25590 + 511) / 512 = 26101 / 512 = 51$  (округление вверх)  
 fat\_sz не изменился. Итак, fat\_sz = 51.  
 Проверяем:  
 Занимаемое место: reserved\_sectors + root\_dir\_sectors + num\_fats \* fat\_sz + data\_sectors =  $1 + 32 + 2 * 51 + 409465 = 409600$  (совпало).
17. Для FAT32 итерации аналогичны, но без root\_dir\_sectors.
18. Для FAT12 расчет аналогичен, но с формулой для fat\_sz (1.5 байта на запись).
19. Важно: для FAT12 и FAT16 корневой каталог фиксированного размера, а для FAT32 - нет.

20. Приведем таблицу для быстрого подбора (для стандартных параметров: 512 байт/сектор, 2 FAT, зарезервировано 1 сектор (для FAT12/16) или 32 (для FAT32), корневой каталог 512 записей (для FAT12/16)):

Размер тома (Мб)	Тип FAT	Секторов на кластер (spc)	Размер кластера
1.44	FAT12	1	0.5
16	FAT16	4	2
32	FAT16	2	1
64	FAT16	4	2
128	FAT16	8	4
256	FAT16	16	8
512	FAT16	32	16
1024 (1 Гб)	FAT16	64	32
2048 (2 Гб)	FAT16	64	32
4096 (4 Гб)	FAT32	8	4
8192 (8 Гб)	FAT32	16	8
16384 (16 Гб)	FAT32	32	16
32768 (32 Гб)	FAT32	64	32
65536 (64 Гб)	FAT32	128	64

21. Однако, для точного расчета необходимо учитывать, что размер FAT (fat\_sz) может быть дробным для FAT12, и его нужно округлять вверх до целого сектора.
22. Для программирования: напишите функцию, которая принимает параметры диска и возвращает структуру с вычисленными параметрами (тип FAT, spc, fat\_sz, root\_dir\_sectors и т.д.)
23. Учтите, что для FAT32 зарезервированная область обычно 32 сектора, а для FAT12/16 - 1.
24. Также, для FAT32 нужно вычислить номер корневого кластера (обычно 2) и заполнить FSInfo.

25. Приведем псевдокод для расчета параметров FAT16:

```
function calculate_fat16(total_sectors, bytes_per_sector, reserved_sectors, num_fats,
    root_dir_entries, sectors_per_cluster):
    root_dir_sectors = ceil((root_dir_entries * 32) / bytes_per_sector)
    fat_sz = 0
    do:
        prev_fat_sz = fat_sz
        data_sectors = total_sectors - reserved_sectors - root_dir_sectors - num_fats * fat_sz
        total_clusters = data_sectors / sectors_per_cluster
        fat_sz = ceil((total_clusters * 2) / bytes_per_sector)
    while fat_sz != prev_fat_sz
    return fat_sz, data_sectors, total_clusters

Для FAT12 замените формулу fat_sz на: fat_sz = ceil((total_clusters * 3) / (2 *
    bytes_per_sector))

Для FAT32:
function calculate_fat32(total_sectors, bytes_per_sector, reserved_sectors, num_fats,
    sectors_per_cluster):
    fat_sz32 = 0
    do:
        prev_fat_sz32 = fat_sz32
        data_sectors = total_sectors - reserved_sectors - num_fats * fat_sz32
        total_clusters = data_sectors / sectors_per_cluster
        fat_sz32 = ceil((total_clusters * 4) / bytes_per_sector)
```

```
while fat_sz32 != prev_fat_sz32
return fat_sz32, data_sectors, total_clusters
```

26. Важно: total\_clusters должно быть целым, поэтому data\_sectors должно делиться на sectors\_per\_cluster. В расчетах мы используем целочисленное деление.
27. Также, для FAT12/16: если total\_clusters выходит за границы для выбранного типа, то нужно изменить тип FAT или размер кластера.
28. На практике, при создании файловой системы, сначала определяют тип FAT на основе размера тома и желаемого размера кластера, а затем уточняют параметры.
29. Для дискет (FAT12) параметры фиксированы:  
1.44 Мб: total\_sectors=2880, bytes\_per\_sector=512, reserved\_sectors=1, num\_fats=2, root\_dir\_entries=224, sectors\_per\_cluster=1 (для 3.5" 2HD).  
Тогда:  

$$\text{root\_dir\_sectors} = (224 * 32 + 511) / 512 = 14$$

$$\text{data\_sectors} = 2880 - 1 - 14 - 2 * \text{fat\_sz}$$

$$\text{total\_clusters} = \text{data\_sectors} / 1 = \text{data\_sectors}$$

$$\text{fat\_sz} = (\text{total\_clusters} * 3 + 1) / (2 * 512) = (\text{total\_clusters} * 3 + 1) / 1024$$
Итерационно находим: fat\_sz = 9, total\_clusters = 2847, data\_sectors=2847.
30. Для жестких дисков и разделов расчеты аналогичны, но с другими начальными параметрами.
31. В случае, если расчет не сходится (например, слишком много итераций), следует остановиться и выдать ошибку.
32. Также, важно проверить, что количество кластеров для FAT12 не превышает 4084, для FAT16 - 65524, а для FAT32 - не менее 65525 и не более  $2^{28}-1$  (примерно 268 миллионов).
33. Для FAT32 максимальный размер кластера 64 сектора (32 Кб), поэтому максимальный теоретический объем:  $268435455 * 32 \text{ Кб} = 8 \text{ Тб}$ . Но из-за ограничений в реализации многих систем ограничиваются 2 Тб.
34. Приведем таблицу для FAT32 с разными размерами кластера и максимальными объемами:

Размер кластера (Кб)	Максимальное количество кластеров (примерно)	Макс
4	268435455	1 Тб
8	268435455	2 Тб
16	268435455	4 Тб
32	268435455	8 Тб

Но на практике из-за ограничений MBR (2 Тб) и других, часто ограничиваются 2 Тб.

35. Для современных больших дисков (больше 2 Тб) используют GPT, а файловую систему — exFAT или NTFS.
36. В заключение, алгоритм расчета параметров FAT нетривиален и требует итераций. Для реализации необходимо аккуратно выполнять шаги и проверять граничные условия.
37. Приведем пример кода на Python для расчета параметров FAT16 (без учета граничных условий и ошибок):

```
python

def calculate_fat16(total_sectors, bytes_per_sector=512, reserved_sectors=
1, num_fats=2, root_dir_entries=512, sectors_per_cluster=4):
    root_dir_sectors = (root_dir_entries * 32 + bytes_per_sector - 1) // b
ytes_per_sector
    fat_sz = 0
    while True:
        data_sectors = total_sectors - reserved_sectors - root_dir_sectors
        - num_fats * fat_sz
        total_clusters = data_sectors // sectors_per_cluster
        new_fat_sz = (total_clusters * 2 + bytes_per_sector - 1) // bytes_
per_sector
        if new_fat_sz == fat_sz:
            break
        fat_sz = new_fat_sz
    return fat_sz, data_sectors, total_clusters
```

```
# Пример для 200 Мб (409600 секторов) и spc=32
fat_sz, data_sectors, total_clusters = calculate_fat16(409600, sectors_per_
_cluster=32)
print("FAT size in sectors:", fat_sz)
print("Data sectors:", data_sectors)
print("Total clusters:", total_clusters)
```

38. Для FAT32:

```
python

def calculate_fat32(total_sectors, bytes_per_sector=512, reserved_sectors=
32, num_fats=2, sectors_per_cluster=8):
    fat_sz32 = 0
    while True:
        data_sectors = total_sectors - reserved_sectors - num_fats * fat_s
z32
        total_clusters = data_sectors // sectors_per_cluster
        new_fat_sz32 = (total_clusters * 4 + bytes_per_sector - 1) // byte
s_per_sector
        if new_fat_sz32 == fat_sz32:
            break
        fat_sz32 = new_fat_sz32
    return fat_sz32, data_sectors, total_clusters

# Пример для 4 Гб (8388608 секторов) и spc=8
fat_sz32, data_sectors, total_clusters = calculate_fat32(8388608, sectors_
per_cluster=8)
print("FAT32 size in sectors:", fat_sz32)
print("Data sectors:", data_sectors)
print("Total clusters:", total_clusters)
```

39. Надеюсь, это подробное руководство поможет вам в написании программы для форматирования.

Вот таблица для итерационного расчета `fat_sz` и `data_sectors`. Это **ключевая часть алгоритма**, требующая подбора из-за взаимозависимости параметров.

## Таблица итерационного расчета для FAT16

Предположим параметры:

- `total_sectors` = 409600 (200 МБ при 512 байт/сектор)
- `bytes_per_sector` = 512
- `reserved_sectors` = 1
- `num_fats` = 2
- `root_dir_entries` = 512
- `sectors_per_cluster` = 4 (2 КБ кластер)

Итерация	Шаг	Расчет	Значение	Г
0	1. Начальное приближение	$fat\_sz = 0$	0	Г
	2. Корневой каталог	$root\_dir\_sectors = ((512 \times 32) + 511) / 512$	32 сектора	5
	3. Data Sectors	$data\_sectors = 409600 - 1 - 32 - 2 \times 0$	409567	
	4. Total Clusters	$total\_clusters = 409567 / 4$	102391	C
	5. Новый FAT Size	$fat\_sz = (102391 \times 2 + 511) / 512$	401 секторов	C
1	3. Data Sectors	$data\_sectors = 409600 - 1 - 32 - 2 \times 401$	408765	C
	4. Total Clusters	$total\_clusters = 408765 / 4$	102191	
	5. Новый FAT Size	$fat\_sz = (102191 \times 2 + 511) / 512$	401 секторов	Г
Результат		$fat\_sz = 401$ , $data\_sectors = 408765$ , $total\_clusters = 102191$		

Проверка:  $1 + 32 + 2 \times 401 + 408765 = 409600$  ✓

Таблица для FAT32 (более простая)

Параметры:

- $total\_sectors = 8,388,608$  (4 ГБ при 512 байт/сектор)
- $bytes\_per\_sector = 512$
- $reserved\_sectors = 32$
- $num\_fats = 2$
- $sectors\_per\_cluster = 8$  (4 КБ кластер)

Итерация	Расчет	Значение	Примечание
0	$data\_sectors = 8,388,608 - 32 - 2 \times 0$	8,388,576	Начальное приближение
	$total\_clusters = 8,388,576 / 8$	1,048,572	
	$fat\_sz32 = (1,048,572 \times 4 + 511) / 512$	8,193 секторов	
1	$data\_sectors = 8,388,608 - 32 - 2 \times 8,193$	8,372,190	
	$total\_clusters = 8,372,190 / 8$	1,046,523	
	$fat\_sz32 = (1,046,523 \times 4 + 511) / 512$	8,177 секторов	
2	$data\_sectors = 8,388,608 - 32 - 2 \times 8,177$	8,372,222	
	$total\_clusters = 8,372,222 / 8$	1,046,527	
	$fat\_sz32 = (1,046,527 \times 4 + 511) / 512$	8,177 секторов	Сходимость достигнута

Таблица быстрого подбора (эмпирические значения)

Для стандартных параметров (512-байтные сектора, 2 FAT таблицы):

Объем тома	Тип FAT	Секторов на кластер	Ориентировочный fat_sz	Комментарий
1.44 МБ	FAT12	1	9 секторов	Для дискет 3.5"
16 МБ	FAT16	4	~25 секторов	
32 МБ	FAT16	2	~50 секторов	
64 МБ	FAT16	4	~50 секторов	
128 МБ	FAT16	8	~50 секторов	
256 МБ	FAT16	16	~50 секторов	
512 МБ	FAT16	32	~50 секторов	
1 ГБ	FAT16	64	~50 секторов	Предел FAT16
2 ГБ	FAT32	8	~4,000 секторов	
4 ГБ	FAT32	8	~8,000 секторов	
8 ГБ	FAT32	16	~8,000 секторов	
16 ГБ	FAT32	32	~8,000 секторов	
32 ГБ	FAT32	64	~8,000 секторов	

### Алгоритм подбора в псевдокоде:

python

```
def calculate_fat_params(total_sectors, bytes_per_sector, fat_type):
    # Начальные значения
    if fat_type == "FAT12/16":
        reserved = 1
        root_entries = 512
        root_sectors = ceil((root_entries * 32) / bytes_per_sector)
    else: # FAT32
        reserved = 32
        root_sectors = 0

    # Подбор sectors_per_cluster
    for spc in [1, 2, 4, 8, 16, 32, 64, 128]:
        if spc подходит для данного объема:
            break

    # Итерационный расчет fat_sz
    fat_sz = 0
    while True:
        old_fat_sz = fat_sz

        # Data sectors без учета FAT
        data_sectors = total_sectors - reserved - root_sectors

        if fat_type == "FAT12/16":
            # Учитываем уже известную FAT
            data_sectors -= num_fats * fat_sz
            total_clusters = data_sectors // spc
            # Новый размер FAT
            if fat_type == "FAT16":
                fat_sz = ceil((total_clusters * 2) / bytes_per_sector)
            else: # FAT12
                fat_sz = ceil((total_clusters * 3) / (2 * bytes_per_sector))
        else: # FAT32
            data_sectors -= num_fats * fat_sz
            total_clusters = data_sectors // spc
            fat_sz = ceil((total_clusters * 4) / bytes_per_sector)
```

```
if fat_sz == old_fat_sz:
    break # Сходимость
return fat_sz, data_sectors, total_clusters
```

[Продолжить с DeepSeek](#)