

# Размещение файлов

Непрерывное размещение  
файлов

# Непрерывное размещение файлов

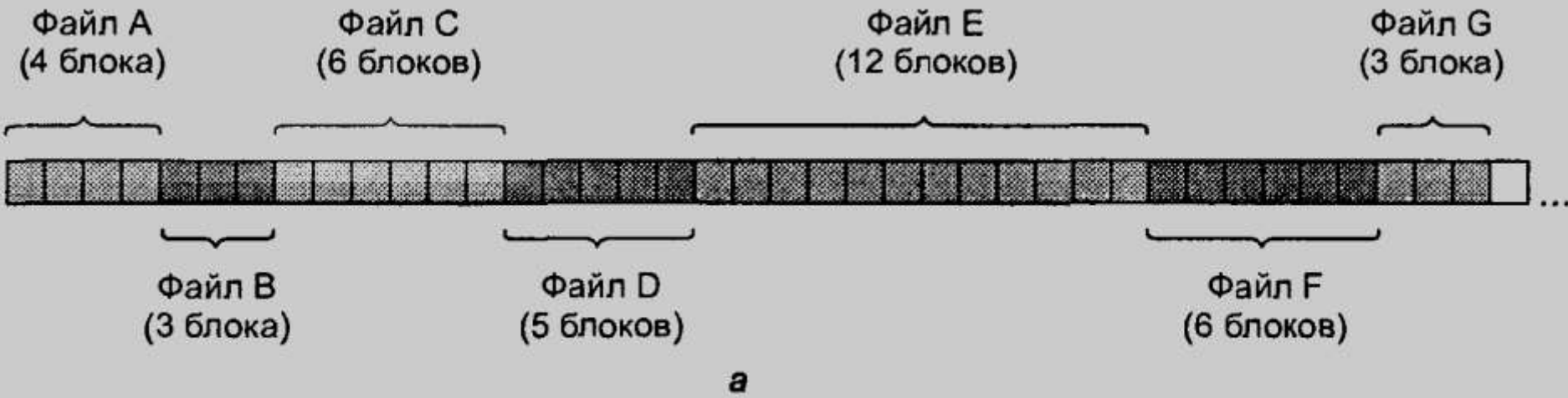
- Под файлы выделяются непрерывные участки адресов в пространствах устройств хранения
- Пользователь заранее указывает размер области, которую нужно выделить под файл
- Если непрерывную область заданного размера выделить невозможно, файл создан не будет

# Преимущества непрерывного размещения файлов

- Следующие друг за другом логические записи обычно оказываются размещенными рядом и физически
- Непрерывное размещение файлов увеличивает скорость доступа к данным
- В директориях нужно хранить только информацию о месте, в котором файл начинается и о длине файла

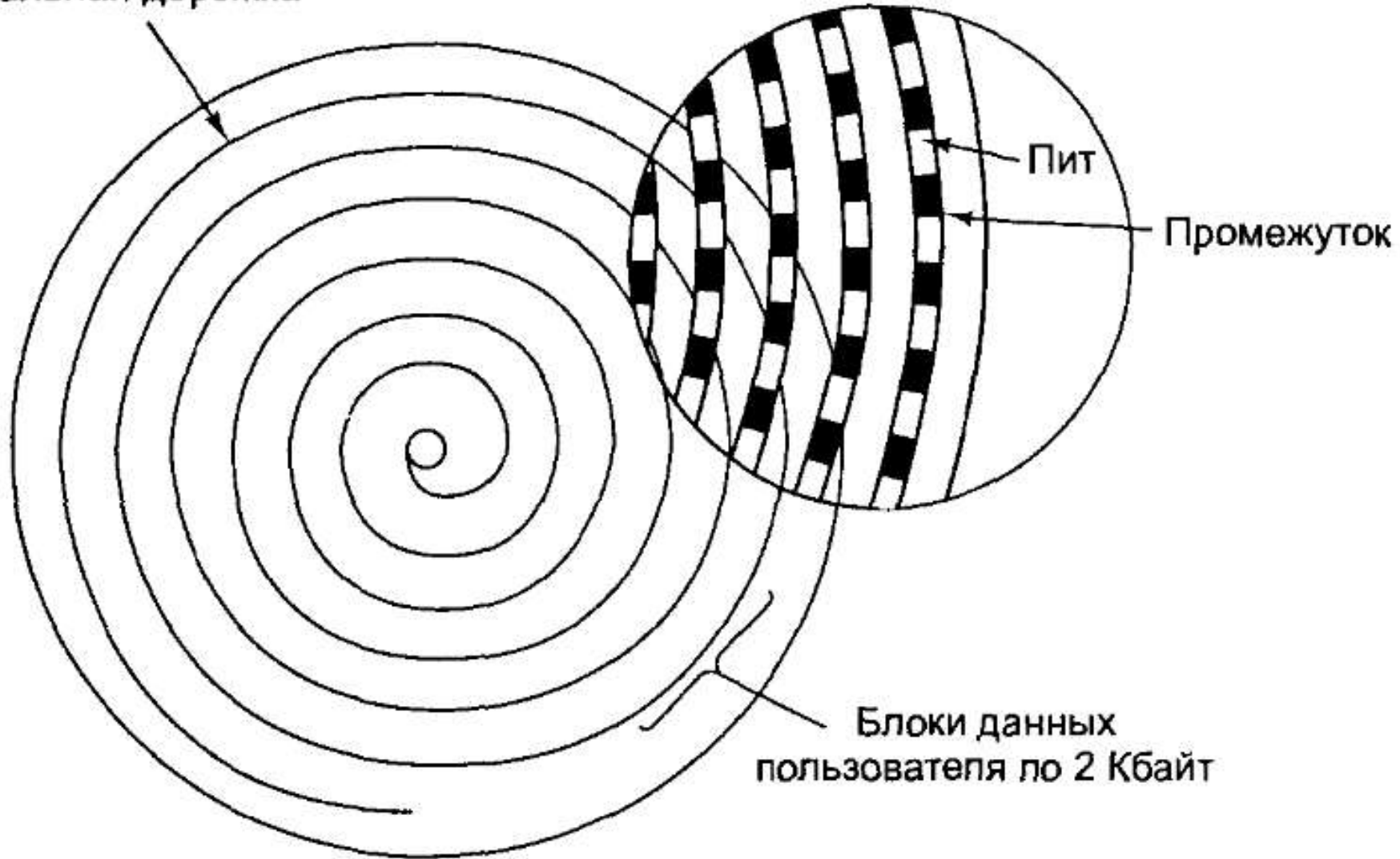
# Недостатки непрерывного размещения файлов

- Этим системам присуща внешняя фрагментация после удаления файлов
- Если размер файла превысит изначально заданный, то файл должен быть целиком перенесен в новый участок подходящего размера



**а) Семь непрерывных файлов на диске**  
**б) Состояние диска после удаления двух файлов**

Спиральная дорожка



# Структура записи CD-ROM

# Вопрос для самопроверки

- Непрерывное размещение файлов не используется в современных системах?  
(Да/Нет)

# Вопрос для самопроверки

- Непрерывное размещение файлов не используется в современных системах? (Да/Нет)
- Нет. Непрерывное размещение файлов очень удобно и используется на однажды записываемых носителях, например, компакт-дисках и DVD, где размер файлов неизменен во времени.



# Вопрос для самопроверки

- Подходит ли непрерывное размещение файлов для записи данных на магнитной ленте? (Да/Нет)

# Вопрос для самопроверки

- Подходит ли непрерывное размещение файлов для записи данных на магнитной ленте? (Да/Нет)
- Да. Для файлов, хранящихся на магнитной ленте подходит непрерывное размещение файлов, поскольку магнитная лента – это носитель с последовательным доступом.

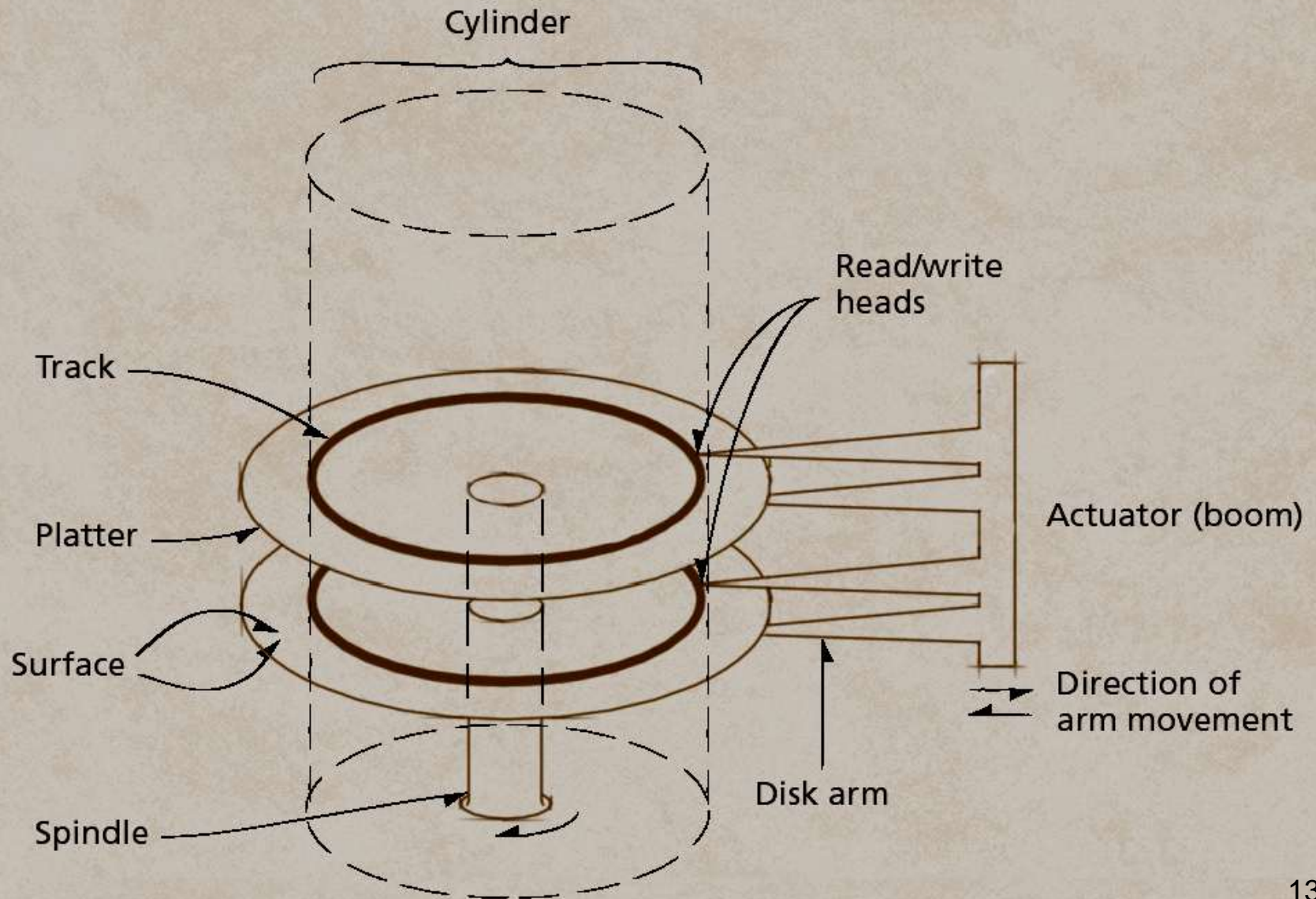
# Размещение файлов

Размещение файлов в виде  
СВЯЗНЫХ СПИСКОВ

# Дорожка

Опр. Дорожка (track) – кольцевая область на диске. Последовательности битов данных из файлов обычно размещаются в непрерывных областях на одной и той же дорожке, чтобы уменьшить потребность в операциях позиционирования головок.

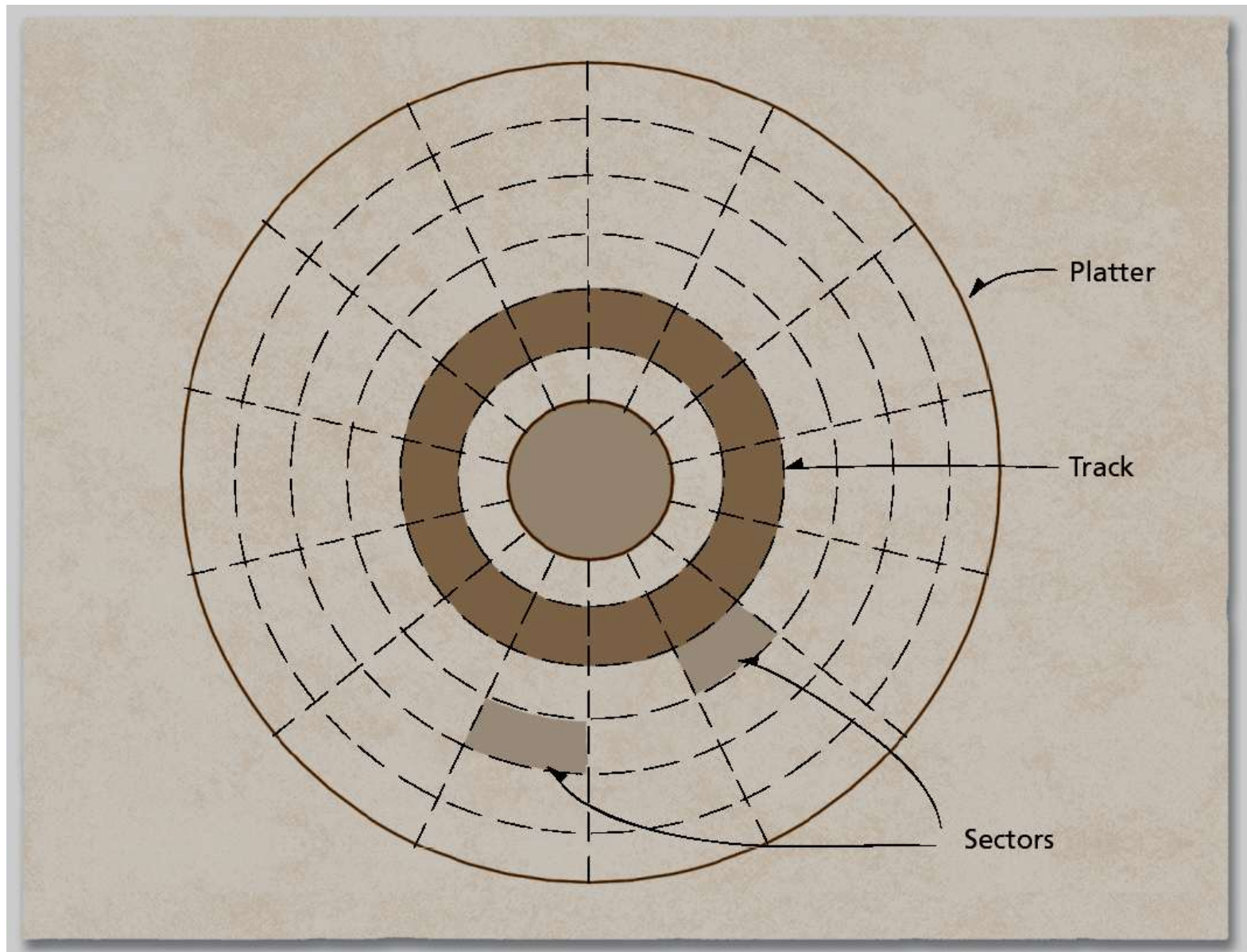
# Схема дискового накопителя с подвижными головками



# Сектор

Опр. Сектор (sector) – наименьшая часть дорожки, к которой может обращаться запрос ввода/вывода. Размер сектора обычно равен 512 байт.

# Схематическое изображение структуры поверхности диска



# Блок

Опр. Блок (block) – элемент данных фиксированного размера, состоящий из непрерывной последовательности нескольких секторов. Обычно используются блоки размером от 1 до 8 килобайт (от 2 до 16 секторов).



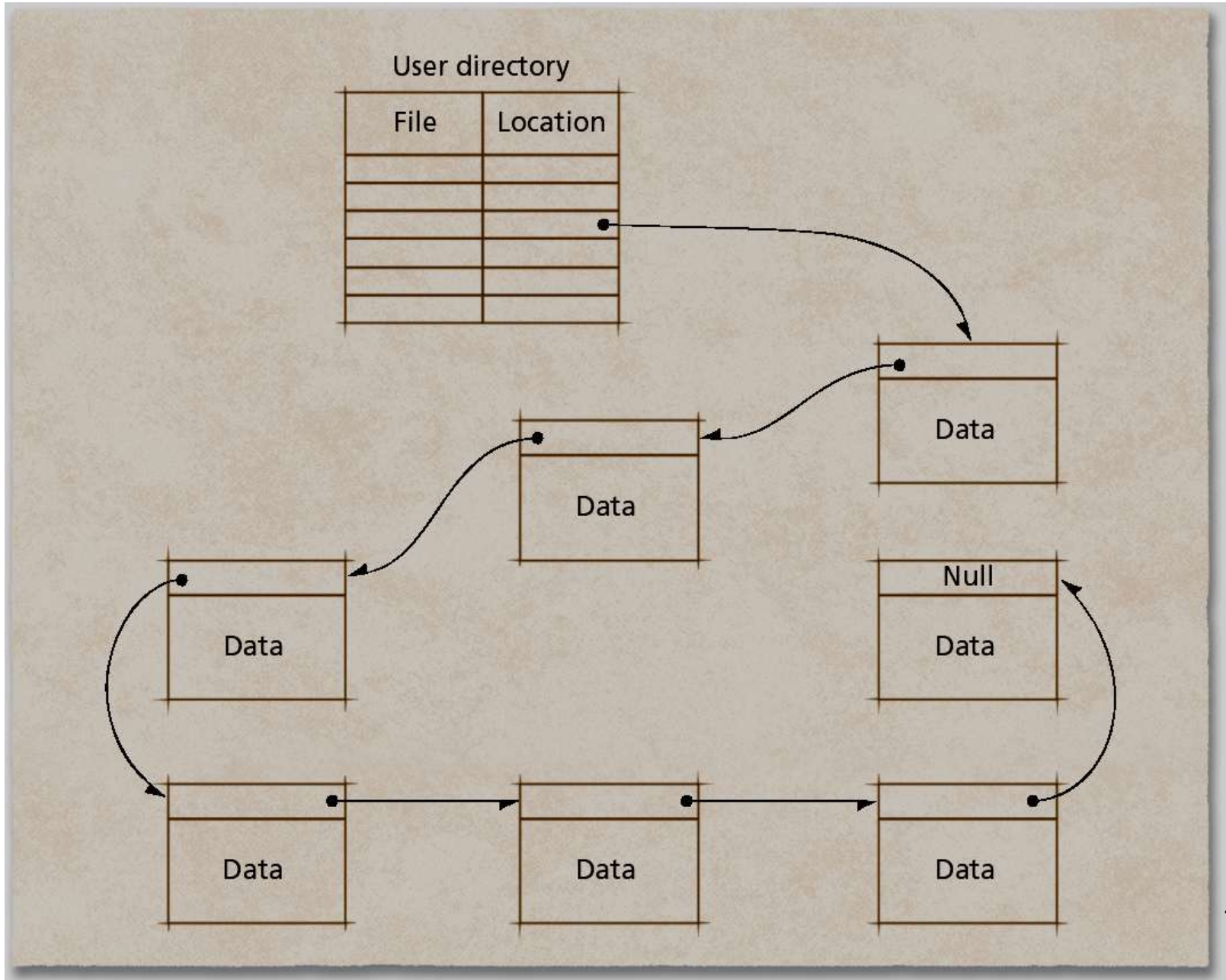
# Блочное размещение

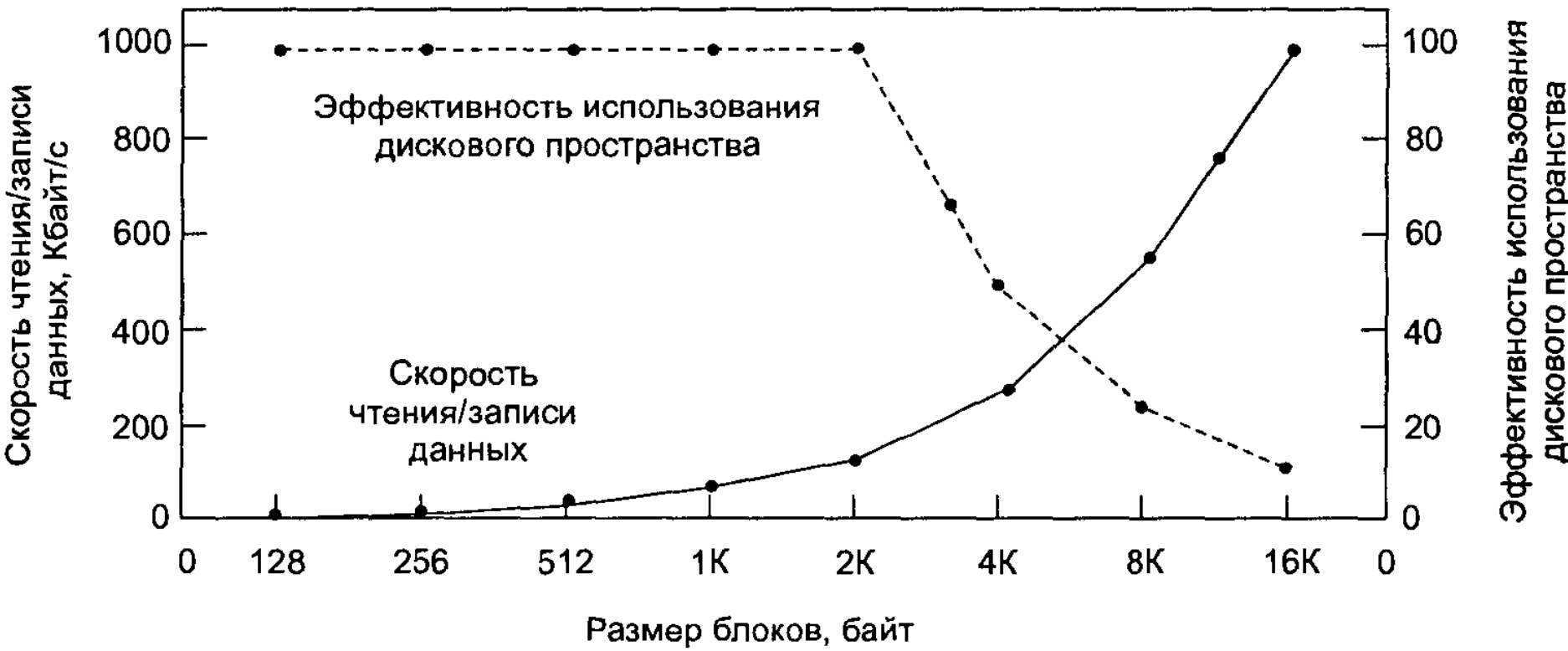
Опр. Блочное размещение (block allocation) – прием, позволяющий файловым системам более эффективно распоряжаться пространством накопителей и уменьшать накладные расходы при просмотре файлов за счет размещения файлов в блоках.

# СВЯЗНЫЙ СПИСОК

Опр. Связный список (linked list) – метод размещения файлов, состоящих из цепочек блоков, при котором в каждом блоке одна из записей зарезервирована под указатель на другой блок. Директория содержит указатель на первый блок в цепочке.

# Размещение файлов в виде СВЯЗНЫХ СПИСКОВ





Зависимость скорости чтения/записи данных диска (жирная линия, левая шкала) и эффективности использования дискового пространства (штриховая линия, правая шкала) от размера блоков. Все файлы по 2 Кбайт

# Вопрос для самопроверки

- Размер сектора обычно равен 512 байт? (Да/Нет)

# Вопрос для самопроверки

- Размер сектора обычно равен 512 байт? (Да/Нет)
- Да. Размер сектора обычно равен 512 байт.

# Вопрос для самопроверки

- В худшем случае система со связным списком должна обратиться ко всем блокам файла при поиске данных?  
(Да/Нет)

# Вопрос для самопроверки

- В худшем случае система со связным списком должна обратиться ко всем блокам файла при поиске данных?  
(Да/Нет)
- Да. Если используется связный список, в худшем случае системе нужно обратиться к каждому блоку файла, чтобы найти нужные данные.



# Вопрос для самопроверки

- Большие блоки в файлах лучше маленьких? (Да/Нет)

# Вопрос для самопроверки

- Большие блоки в файлах лучше маленьких? (Да/Нет)
- Нет. При большом размере блока для извлечения нужной записи требуется меньше операций ввода/вывода, но на внутреннюю фрагментацию расходуется более заметная часть пространства накопителя.

# Размещение файлов

Табличное фрагментированное  
размещение

# Таблица выделения блоков

Опр. Таблица выделения блоков (block allocation table) – таблица, используемая в файловых системах с табличным фрагментированным размещением файлов для ускорения доступа к данным. Обычно загружается в кэш. В этой таблице в качестве индекса используются номера блоков на накопителе, а записи хранят указатели на блоки файлов. Записи в директориях указывают на первые блоки файлов.

User directory

File	Location
A	8
B	6
C	2

Block allocation table

0	22
1	Null
2	5
3	26
4	9
5	20
6	10
7	Free
8	17
9	1
10	14
11	Free
12	3
13	4
14	0
15	Free
16	Free
17	12
18	13
19	Null
20	23
21	Free
22	18
23	19
24	Free
25	Free
26	Null
27	Free



Physical blocks on secondary storage

Block 0 B(4)	Block 1 B(10)	Block 2 C(1)	Block 3 A(4)	Block 4 B(8)	Block 5 C(2)	Block 6 B(1)
Block 7 Free	Block 8 A(1)	Block 9 B(9)	Block 10 B(2)	Block 11 Free	Block 12 A(3)	Block 13 B(7)
Block 14 B(3)	Block 15 Free	Block 16 Free	Block 17 A(2)	Block 18 B(6)	Block 19 C(5)	Block 20 C(3)
Block 21 Free	Block 22 B(5)	Block 23 C(4)	Block 24 Free	Block 25 Free	Block 26 A(5)	Block 27 Free

# Табличное фрагментиро- ванное размещение файлов

# FAT

Опр. FAT (File Allocation Table, таблица размещения файлов) – реализация файловой системы с табличным фрагментированным размещением, разработанная фирмой Microsoft. Известны версии FAT12, FAT16, FAT32, где цифры указывают на число бит в табличных записях.

<b>Размер кластера, Кбайт</b>	<b>FAT-12, Мбайт</b>	<b>FAT-16, Мбайт</b>	<b>FAT-32, Тбайт</b>
0,5	2		
1	4		
2	8	128	
4	16	256	1
8		512	2
16		1024	2
32		2048	2

Максимальный размер накопителя в зависимости от размера блока (кластера) для разных версий FAT

# Вопрос для самопроверки

- Табличное фрагментированное размещение файлов эффективнее размещения в виде СВЯЗНЫХ СПИСКОВ? (Да/Нет)



# Вопрос для самопроверки

- Табличное фрагментированное размещение файлов эффективнее размещения в виде связанных списков? (Да/Нет)
- Да. Таблицы выделения блоков могут хранить информацию о размещении данных в непрерывном виде, и количество операций позиционирования, необходимых для доступа к данным, уменьшается.

# Вопрос для самопроверки

- FAT не подходит для работы с дисковыми накопителями большого объема? (Да/Нет)

# Вопрос для самопроверки

- FAT не подходит для работы с дисковыми накопителями большого объема? (Да/Нет)
- Да. Современные дисковые накопители имеют большой объем и, следовательно, состоят из большого числа блоков. При этом увеличивается размер таблицы размещения файлов, за счет чего растет время доступа к файлам и затраты памяти на кэширование таблицы.

# Вопрос для самопроверки

- Верно ли, что FAT не используется в современных системах? (Да/Нет)

# Вопрос для самопроверки

- Верно ли, что FAT не используется в современных системах? (Да/Нет)
- Нет. FAT эффективна и сейчас для носителей малой емкости, например, USB Flash накопителей.

# Размещение файлов

Индексированное  
фрагментированное размещение

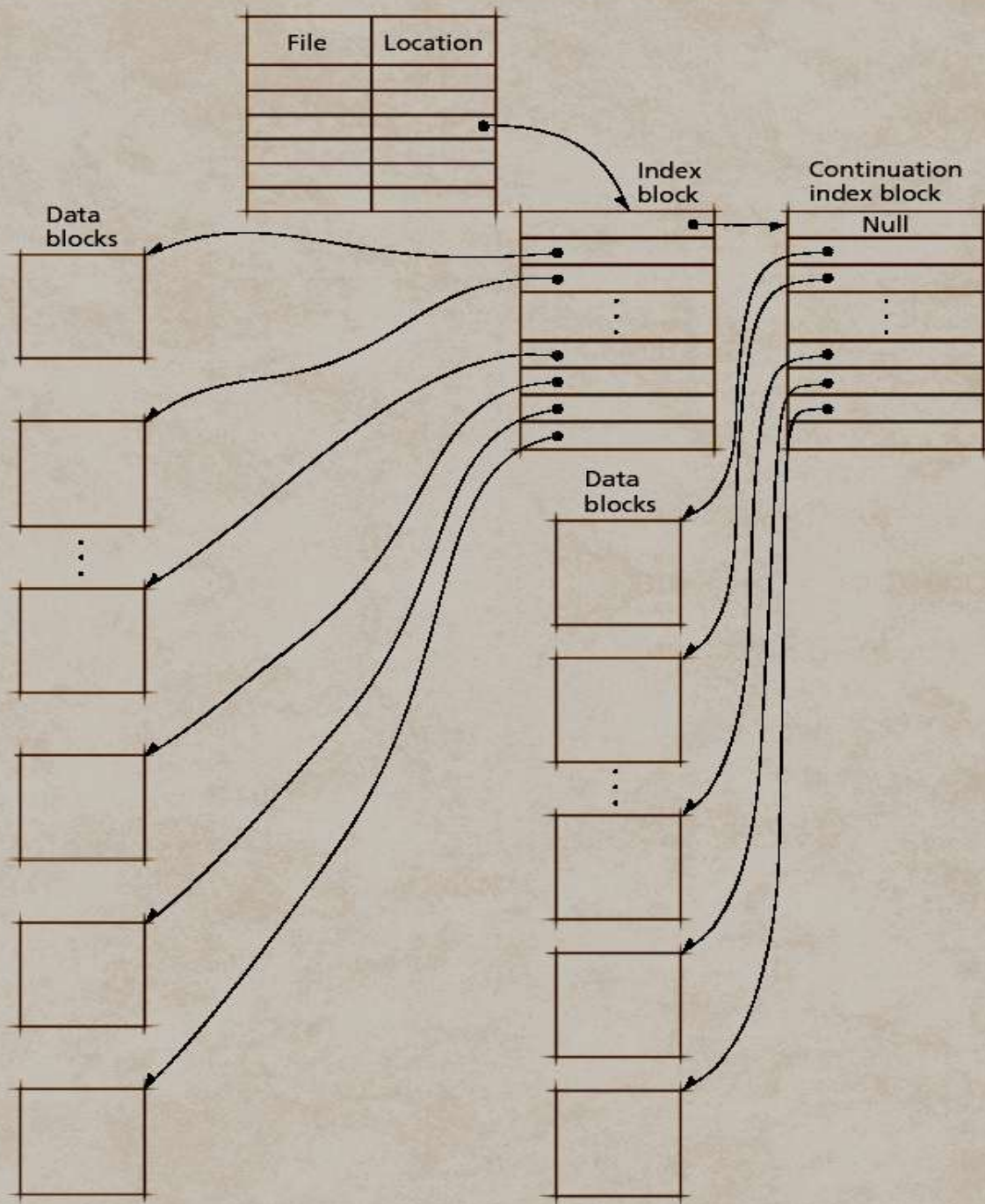
# Индексный блок

Опр. Индексный блок (index block) – блок, содержащий список указателей на блоки с данными файла. Запись о файле в его директории содержит указатель на индексный блок этого файла. Индексные блоки используются, например, в Unix, Linux, NTFS Windows XP.

# Сцепление

Опр. Сцепление (chaining) – метод индексированного фрагментированного размещения, при котором в каждом индексном блоке последние несколько записей зарезервированы под указатели на другие индексные блоки, которые в свою очередь указывают на блоки данных. Сцепление позволяет индексным блокам указывать размещение больших файлов, распределяя указатели на блоки данных по нескольким индексным блокам.





## Сцепление ИНДЕКСНЫХ блоков

# Вопрос для самопроверки

- Индексированное фрагментированное размещение для дисков большого объема эффективнее табличного? (Да/Нет)

# Вопрос для самопроверки

- Индексированное фрагментированное размещение для дисков большого объема эффективнее табличного? (Да/Нет)
- Да. Индексированное фрагментированное размещение аналогично хранению отдельной таблицы выделения блоков для каждого файла. Такой подход может быть эффективнее, поскольку ссылки на блоки каждого файла хранятся в виде непрерывной последовательности в пределах индексного блока.

# Вопрос для самопроверки

- Размещение индексных блоков рядом с блоками данных уменьшает время доступа? (Да/Нет)

# Вопрос для самопроверки

- Размещение индексных блоков рядом с блоками данных уменьшает время доступа? (Да/Нет)
- Да. При считывании индексного блока головка будет близко от блоков данных, на которые ссылается индексный блок, и потребность в позиционировании будет уменьшена, или вообще исчезнет.

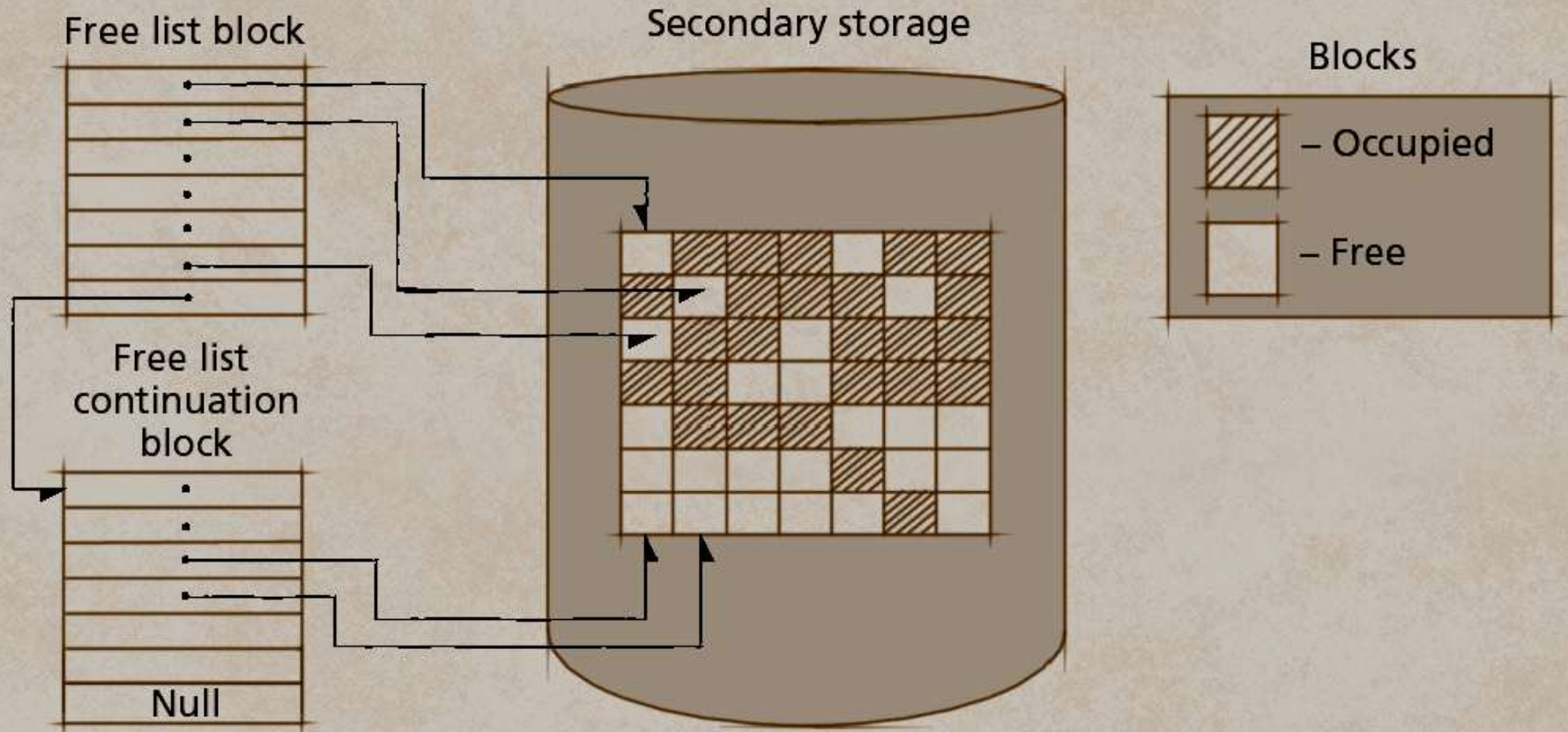
# Размещение файлов

Управление свободным  
пространством

# Список свободных блоков

Опр. Список свободных блоков (free list) – связный список блоков, содержащих адреса свободных блоков на накопителе.

# Управление свободным пространством с помощью списка свободных блоков





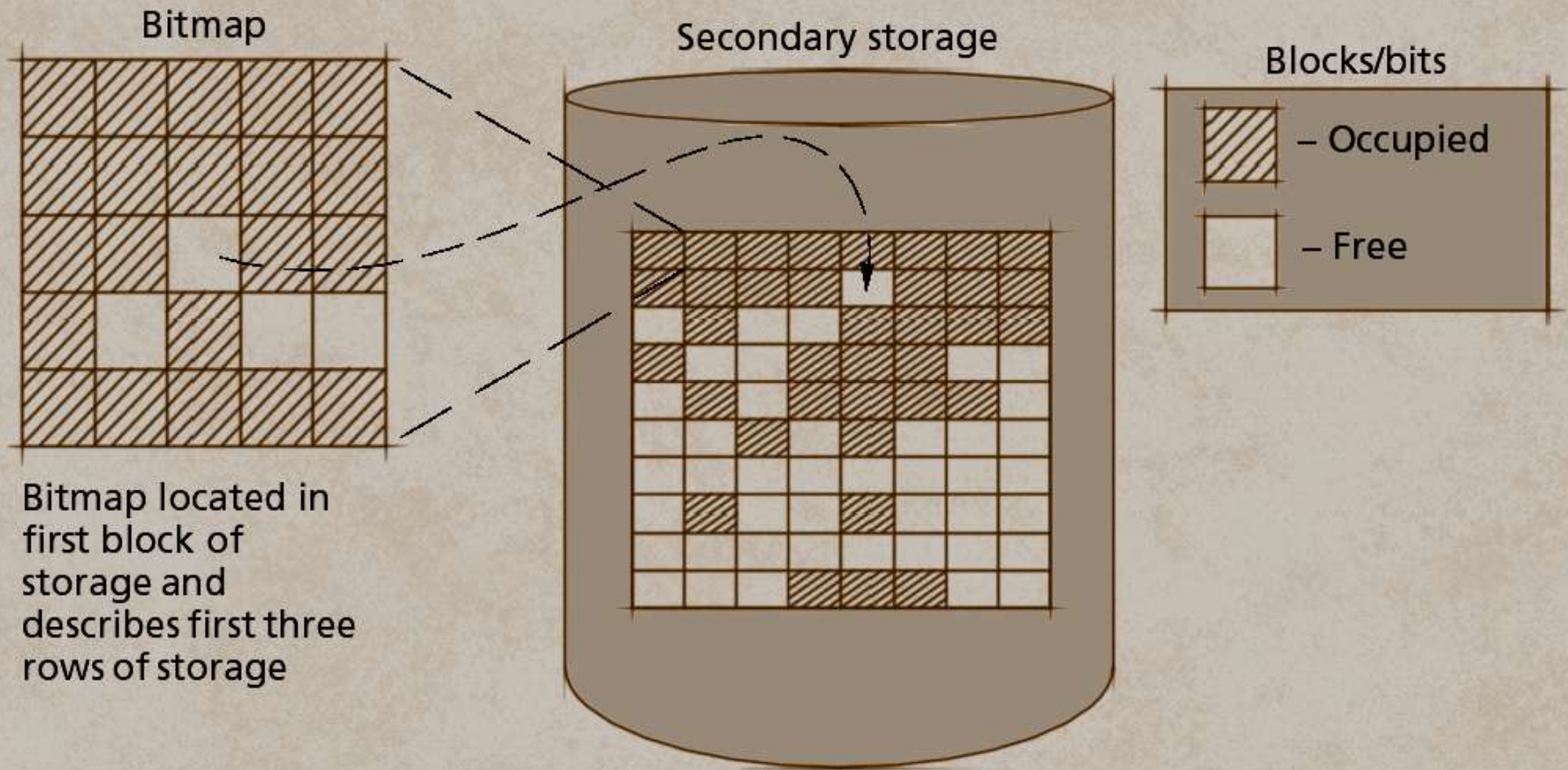
# Список свободных блоков

- Когда системе нужно выделить для файла новый блок, она находит адрес свободного блока в списке свободных блоков, записывает данные в этот блок и удаляет запись об этом блоке из списка
- Обычно файловая система выделяет блоки из начала списка и добавляет освободившиеся блоки в его конец
- Указатели на начало и конец списка могут храниться в суперблоке файловой системы<sup>49</sup>

# БИТОВЫЙ МАССИВ

Опр. Битовый массив (bitmap) – средство управления свободным пространством накопителя, в котором каждому блоку накопителя соответствует один бит, причем номер бита соответствует номеру блока. Битовые массивы позволяют выделять непрерывные блоки эффективнее, чем списки свободных блоков, но на поиск свободных участков с помощью битовых массивов может потребоваться заметное время.

# Управление свободным пространством с помощью битового массива



# Вопрос для самопроверки

- Битовый массив эффективнее списка свободных блоков при выделении одного свободного блока? (Да/Нет)

# Вопрос для самопроверки

- Битовый массив эффективнее списка свободных блоков при выделении одного свободного блока? (Да/Нет)
- Нет. Список свободных блоков эффективнее при выделении одного свободного блока, поскольку для этого в нем нужно только выполнить переход по указателю в начало списка.

# Вопрос для самопроверки

- Битовый массив эффективнее списка свободных блоков при поиске непрерывной последовательности блоков? (Да/Нет)

# Вопрос для самопроверки

- Битовый массив эффективнее списка свободных блоков при поиске непрерывной последовательности блоков? (Да/Нет)
- Да. При поиске непрерывной последовательности свободных блоков битовые массивы эффективнее – их можно просматривать в поисках соответствующих участков, а списки свободных блоков нужно сортировать, на что требуется заметное дополнительное время.

# Вопрос для самопроверки

- Битовый массив всегда занимает меньше места, чем список свободных блоков?  
(Да/Нет)



# Вопрос для самопроверки

- Битовый массив всегда занимает меньше места, чем список свободных блоков?  
(Да/Нет)
- Нет. Часто битовый массив занимает меньше места, поскольку в нем каждому блоку соответствует один бит, а в списке – 32 или даже 64 бита. Однако, если на накопителе мало свободных блоков, то список будет занимать меньше места, чем битовый массив.