第53－54讲 文件的物理组织和外存空闲空间管理

§5.6 Secondary Storage Management

Secondary Storage Management

❖ **Space must be allocated to files.**

❖ **Must keep track of the space available for allocation.**

Space Allocated to Files

❖ **When a new file is created，is the maximum space required for the file allocated at once?**

❖ **What size of portion should be used for file allocation ?**

❖ **What sort of data structure or table (FAT) is used to keep track of the portions assigned to a file?**

Preallocation

❖ **Need the maximum size for the file at the time of creation.**

❖ **Difficult to reliably estimate the maximum potential size of the file.**

❖ **Tend to overestimated file size so as not to run out of space.**

❖ **Allocates space to a file in portions as needed.**

Portion Size （分区大小）

❖ **两个极端情况：**

  ▪ **足够大：能容纳整个文件**

  ▪ **最小： 1个磁盘块大小**

❖ **Tradeoff：**

  ▪ **单个文件的效率 vs. 整个系统的效率**

❖ **Contiguity of space increase performance, especially for Retrieve_Next operations, and greatly for transactions running in a transaction-oriented operating system.**

❖ **Having a large number of small portions increase the size of tables needed to manage the allocation information.**

❖ **Having fixed-size portions (for example,blocks) simplifies the reallocation of space.**

❖ **Having variable-size or small fixed-size portions minimizes waste of unused storage due to overallocation.**

Methods of File Allocation

❖ **Contiguous allocation** （连续分配）
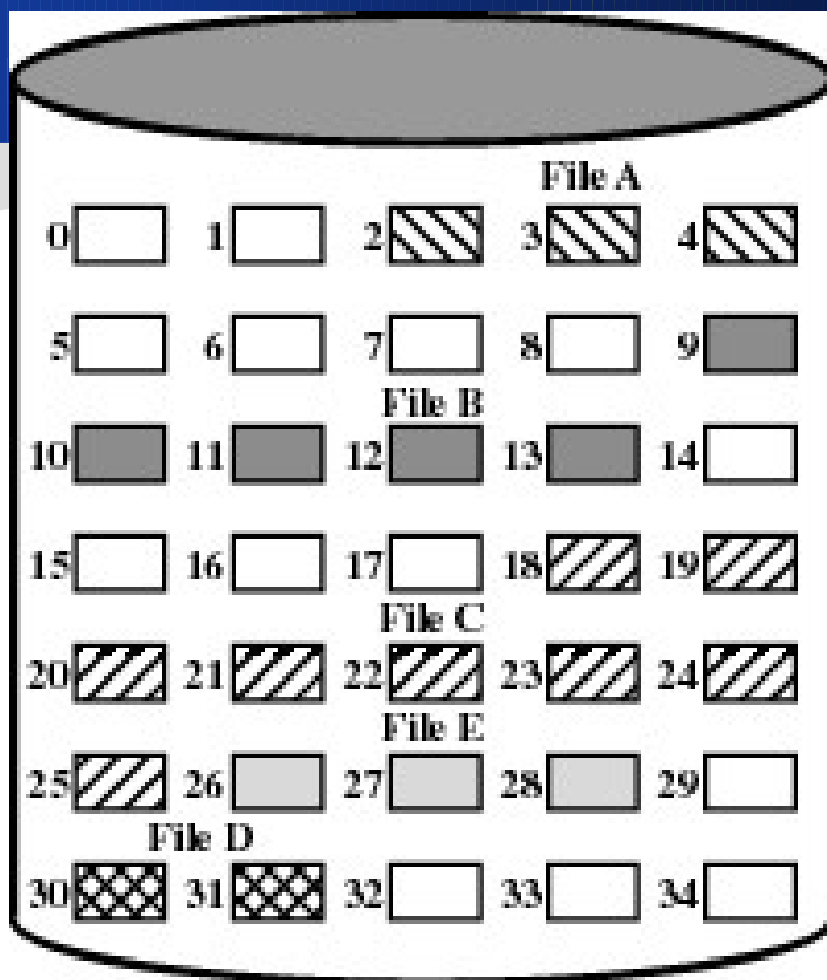
❖ **Chained allocation** （链接分配）

❖ **Indexed allocation** （索引分配）

Contiguous allocation （连续分配）

❖ **Preallocatin strategy using variable-size portions.**

❖ **Single set of blocks is allocated to a file at the time of creation.**

❖ **Only a single entry in the file allocation table.**
- **Starting block and length of the file.**

Figure 12.7 Contiguous File Allocation

Allocation Strategies (分配策略：可变分区)

- ❖ ***First fit***： **Choose the first unused contiguous group of blocks of sufficient size.**

- ❖ ***Best fit***： **Choose the smallest unused group of sufficient size.（内零头最小）**

- ❖ ***Nearest fit***： **Choose the unused group of sufficient size that is closest to the previous allocation for the file to increase locality.**

Contiguous Allocation

❖ **External fragmentation will occur.**

❖ **It is necessary to perform a compaction algorithm to free up additional space on the disk .**
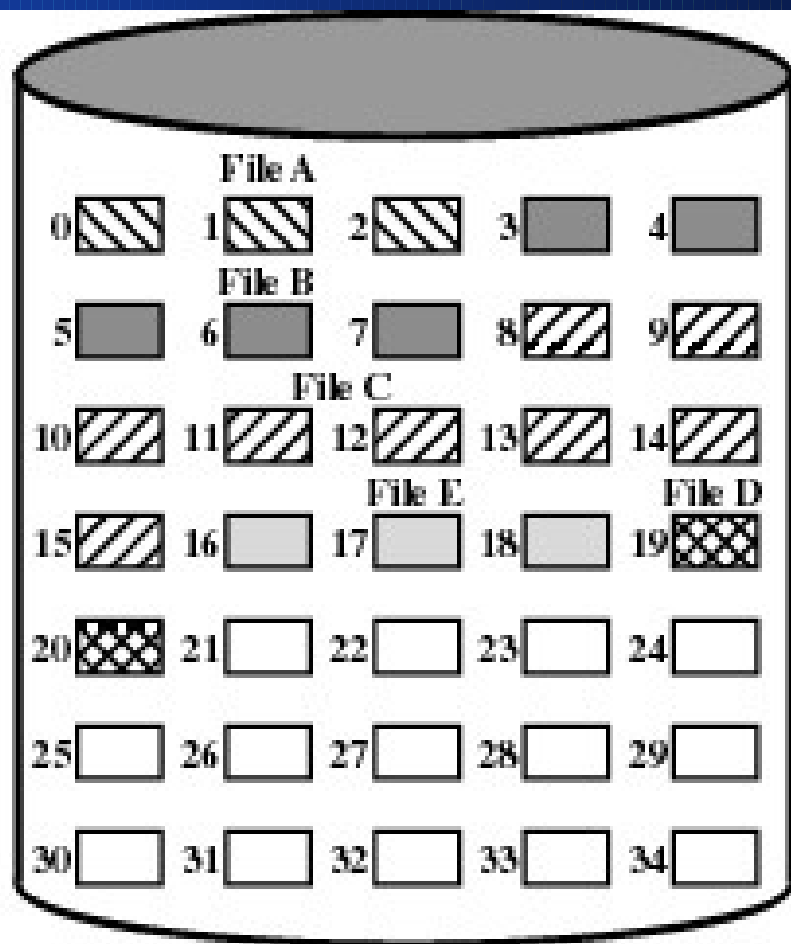
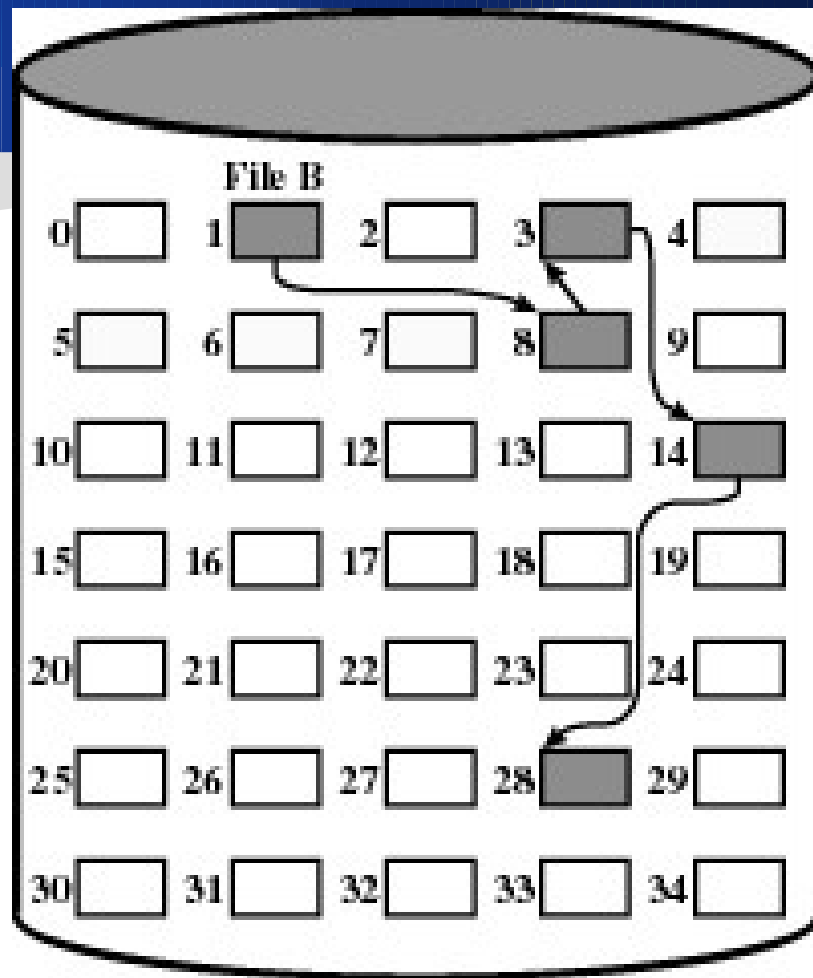| File Allocation Table		
File Name	Start Block	Length
File A	0	3
File B	3	5
File C	8	8
File D	19	2
File E	16	3

Figure 12.8   Contiguous File Allocation (After Compaction

Chained Allocation （链接分配）

❖ **Allocation on basis of individual block.**

❖ **Each block contains a pointer to the next block in the chain.**

❖ **Only single entry in the file allocation table.**

- **Starting block and length of file**

❖ **No external fragmentation.**

❖ **Best for sequential files.**
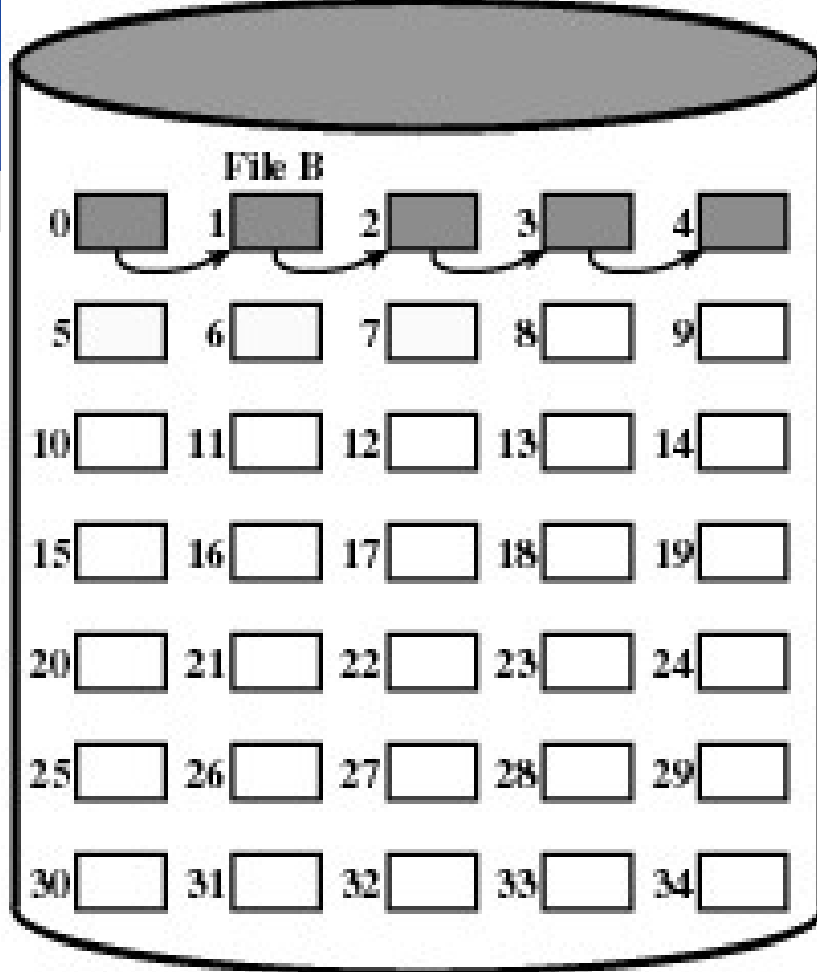
❖ **No accommodation of the principle of locality.**

Figure 12.9   Chained Allocation

❖ **To select an individual block of a file requires tracing trough the chain to the desired block.**

❖ **To access many blocks at a time, a series of accesses to different parts of the disk are required.**

❖ **To overcome this problem , some system periodically consolidate files.**

File Allocation Table

| File Name	Start Block	Length
• • •	• • •	• • •
File B	0	5
• • •	• • •	• • •

Figure 12.10   Chained Allocation (after consolidation)

Indexed Allocation （索引分配）

❖ **File allocation table contains a separate one-level index for each file.(一级索引)**

❖ **The index has one entry for each portion allocated to the file.**

❖ **The file index for a file is kept in a separate block, and the file allocation table contains block number for the index.**

❖ **Allocation may be on the basis of fixed-size blocks.**
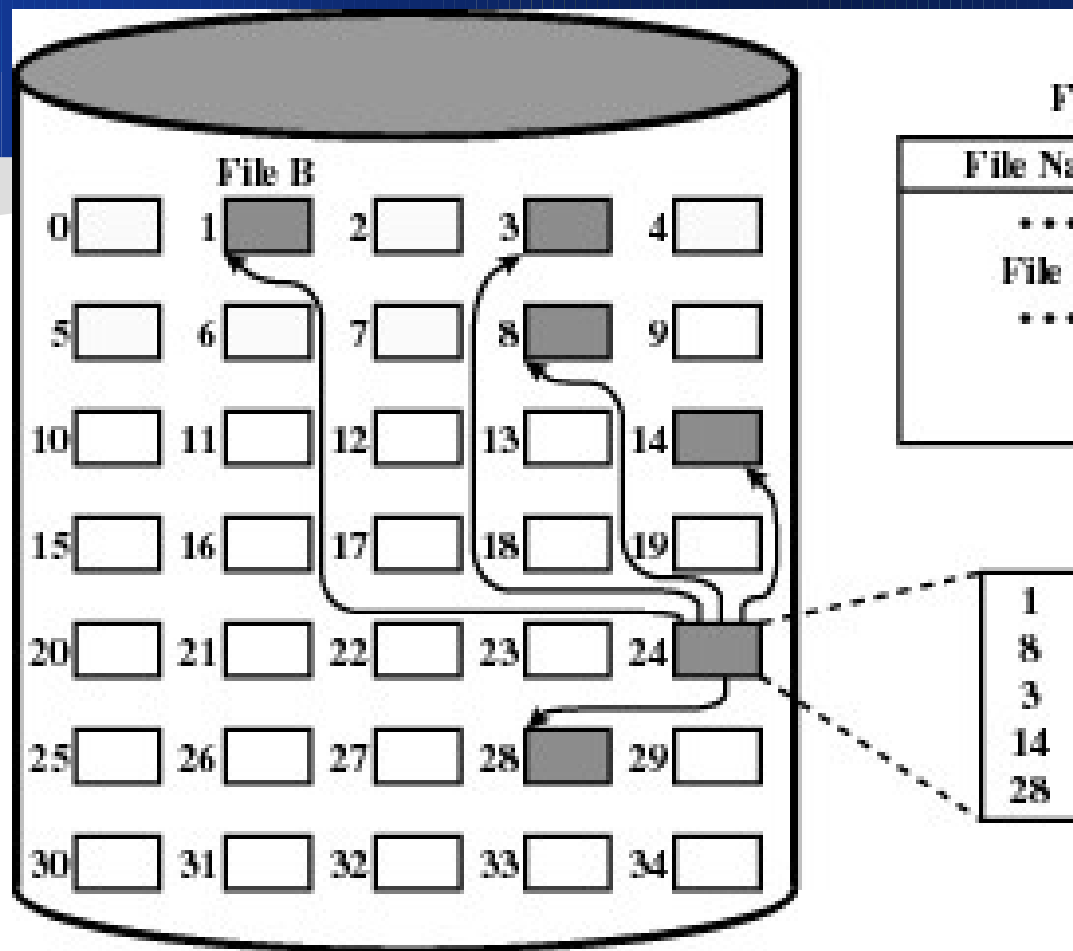
Figure 12.11   Indexed Allocation with Block Portions

❖ **Or allocation may be on the basis of variable-size portions.**
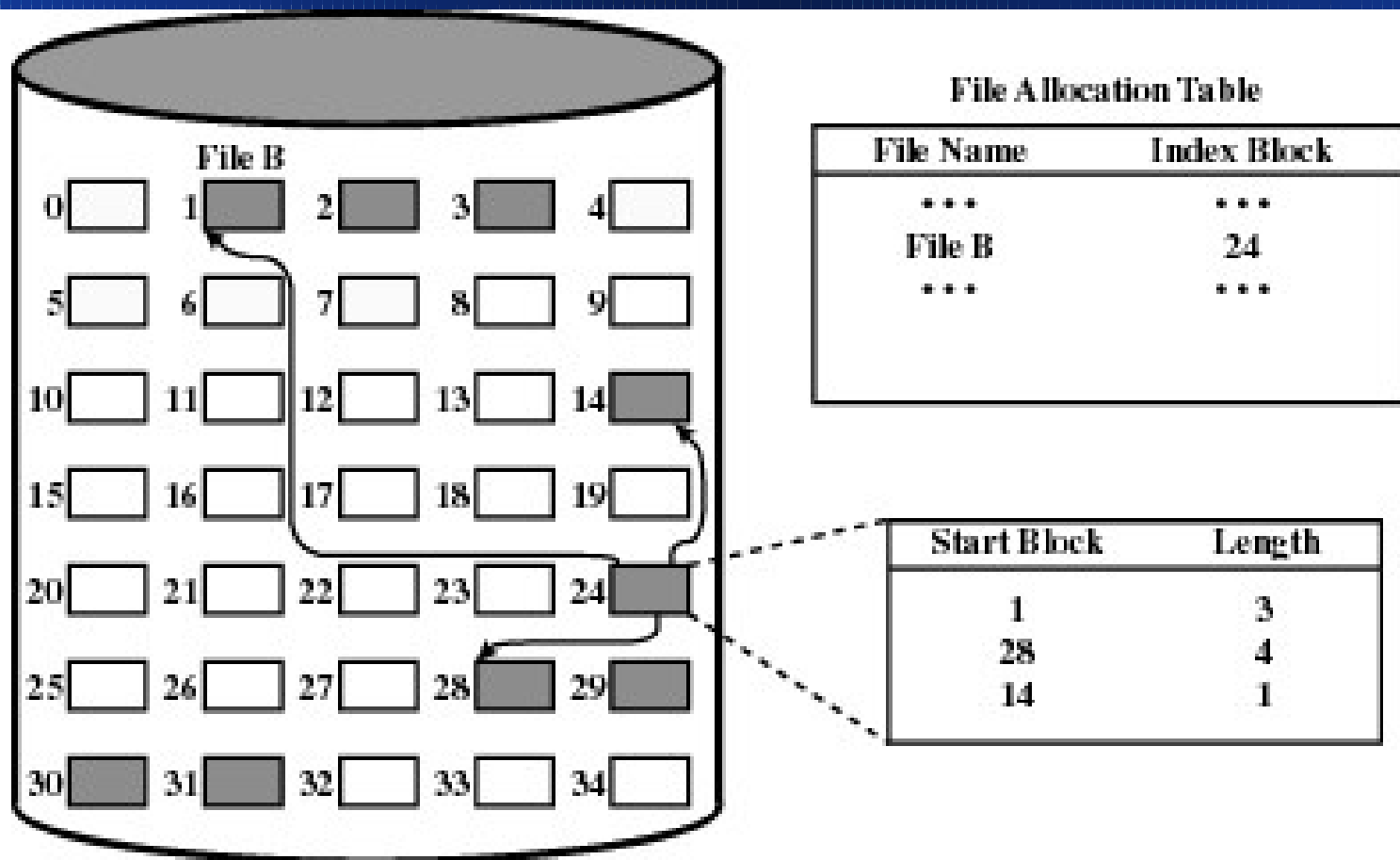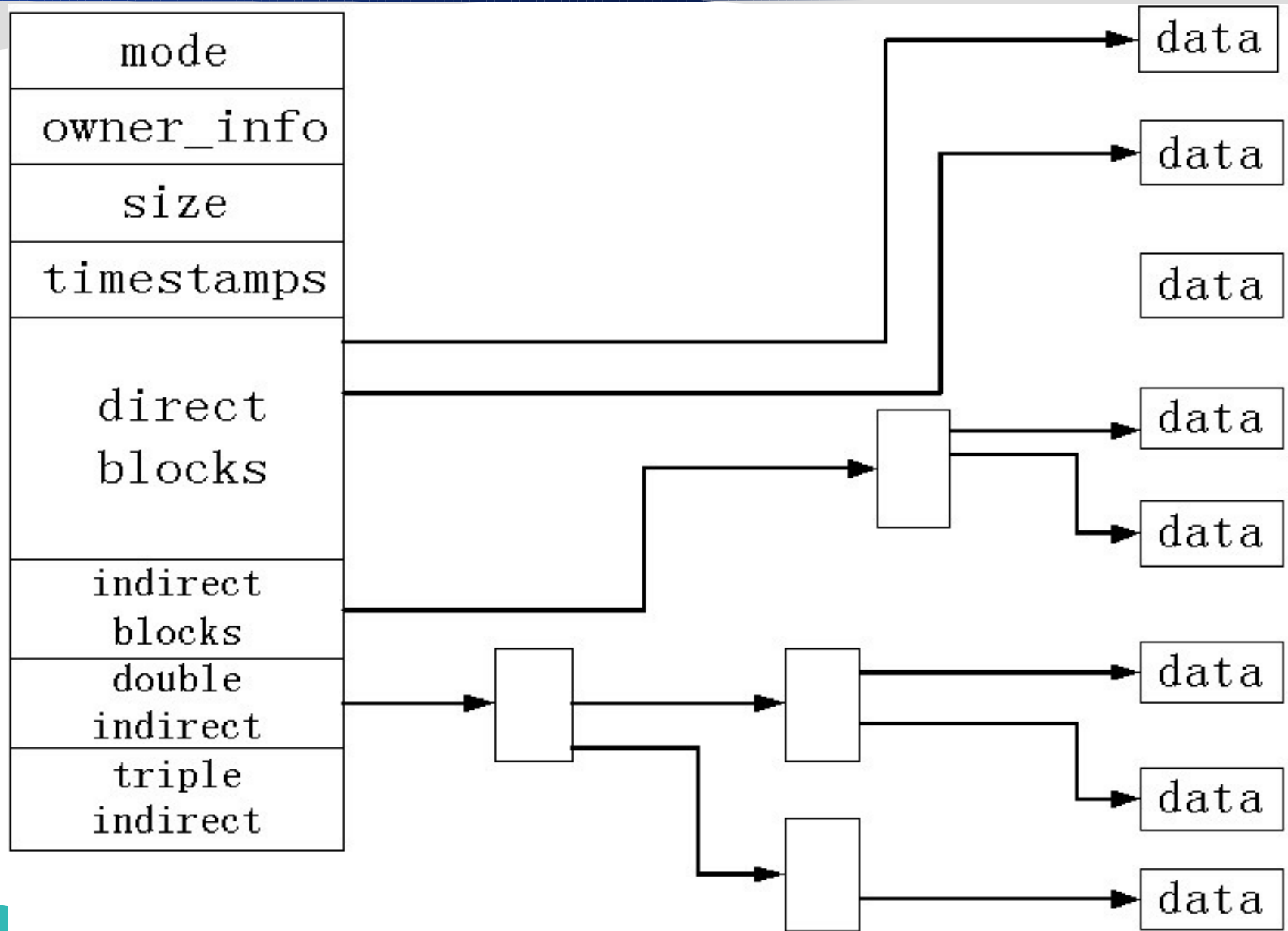
Figure 12.12   Indexed Allocation with Variable-Length Portions

Indexed Allocation

❖ **Eliminates external fragmentation.**

❖ **File consolidation may be done from time to time ， reducing the size of the index in the case of variable-size portions.**

❖ **Supports both sequential and direct access to the file, and thus is the most popular form of file allocation.**

多级索引文件结构

Disk Allocation Table:

❖ **Bit table(位表）**

❖ **Chained free portion(空闲分区链）**

❖ **Indexing(索引）**

❖ 位表，又称为位示图，其中的每一位对应一个磁盘块。位的值为 0 或 1，分别表示磁盘块空闲，或磁盘块已分配

❖ 利用位表容易找到一个或一组空闲盘块

❖ 位表适合于以上各种文件分配方法

❖ 位表很小，可以装入内存

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	1	0	1	1	0	0	0
4																
.																
.																
.																
16																

位示图

Chained Free Portion

❖ **每个空闲分区包含一个指向下一个分区的指针，并记载分区大小**

❖ **无空闲分区表空间开销**

❖ **适合于各种文件分配方法**

❖ **若每次分配一个磁盘块，则可取空闲分区链的第一个盘块进行分配，并调整空闲分区链首指针和分区链大小**

❖ **若采用可变分区法，可用首次适应算法，从链表头开始查找，找到的第一个适合的分区则可分配，然后调整空闲分区链首指针和分区链大小**

Indexing

❖ **将空闲分区视为文件，按文件存储空间分配法为空闲分区建立索引**

❖ **索引表中为每一个空闲分区建立一个索引项**

❖ **为可变分区建立索引比为磁盘块建立索引效率高**

❖ **适合于各种文件分配法**