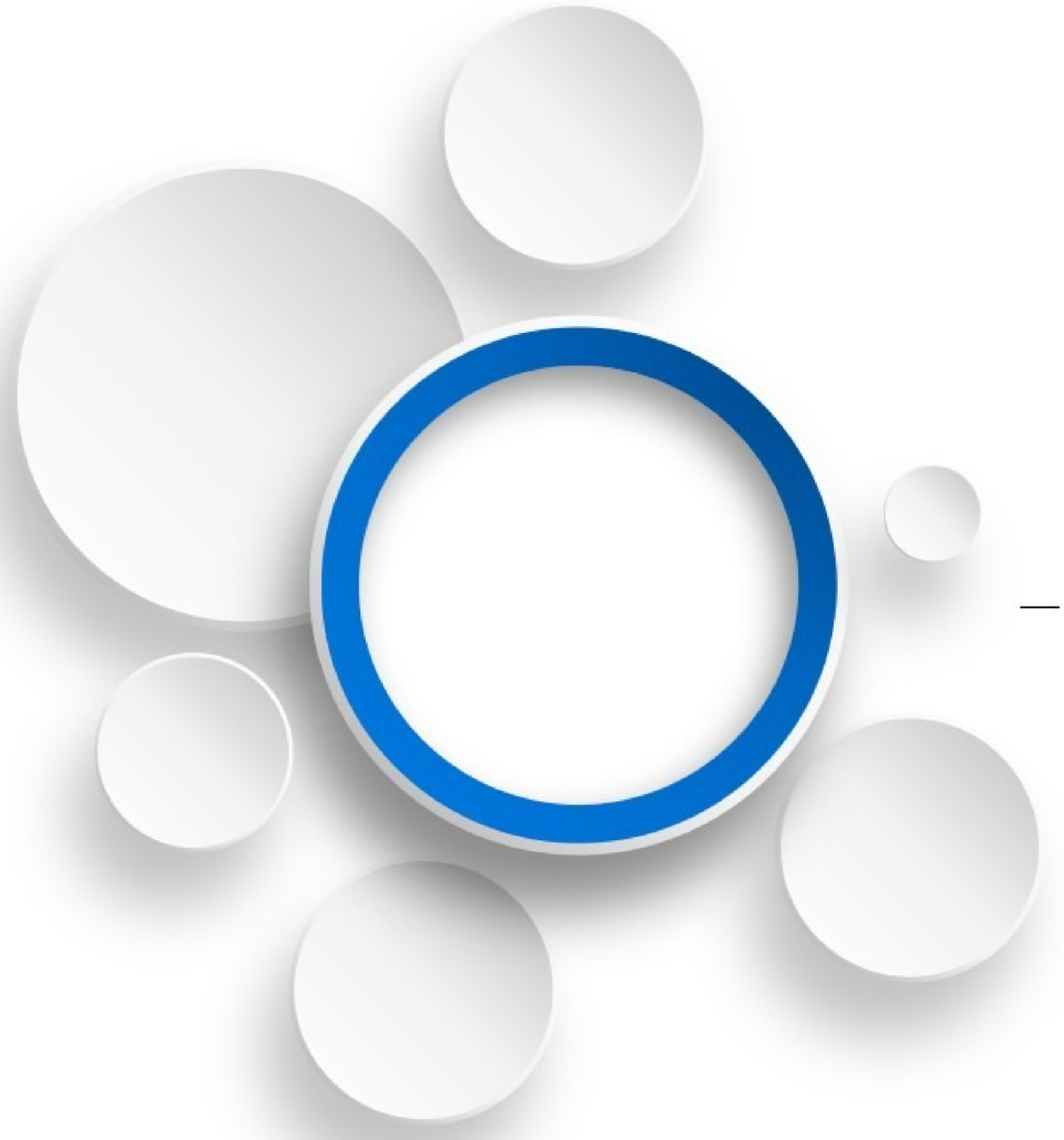


磁盘存储管理

大纲

01 外存的组织方式

02 磁盘空间管理

03 文件系统的可靠性

01

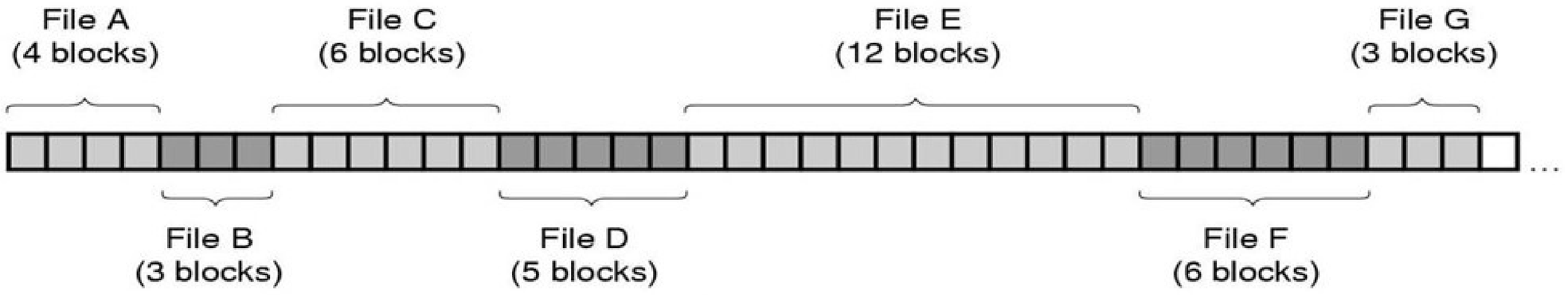
外存的组织方式

文件的实现

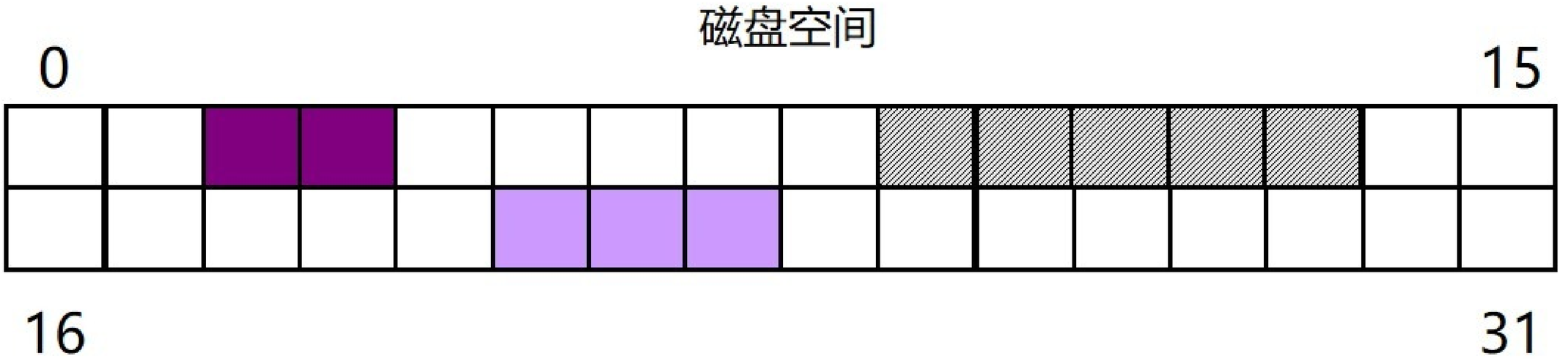
- 从系统的角度来看文件，从文件在物理介质上的存放方式来研究文件——文件的物理结构
- 文件的物理结构直接与外存的组织方式有关。对于不同的外存组织方式，将形成不同的文件物理结构
  - 连续分配
  - 链表分配
  - FAT分配
  - 索引分配

连续分配

- 文件存放在若干连续的物理块中
- 要求为每一个文件分配一组相邻接的盘块
  - 例如，第一个盘块的地址为b，则第二个盘块的地址为b+1，第三个盘块的地址为b+2，...



连续分配

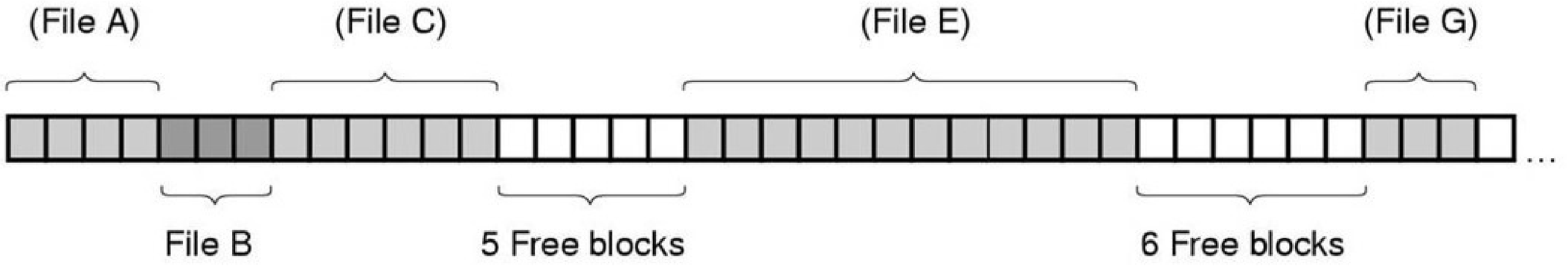


目录		
文件名	起始地址	大小
Hello.c	2	2
z1.cc	9	5
a.out	21	3

连续分配

- 文件存放在若干连续的物理块中
- 通常一个文件的盘块都位于一条磁道上，在进行读/写时，不必移动磁头
- 优点:
  - 简单——目录项中只需存放第一块的地址和块数
  - 支持顺序存取和随机存取
  - 顺序存取速度快
  - 所需的磁盘寻道次数和寻道时间最少

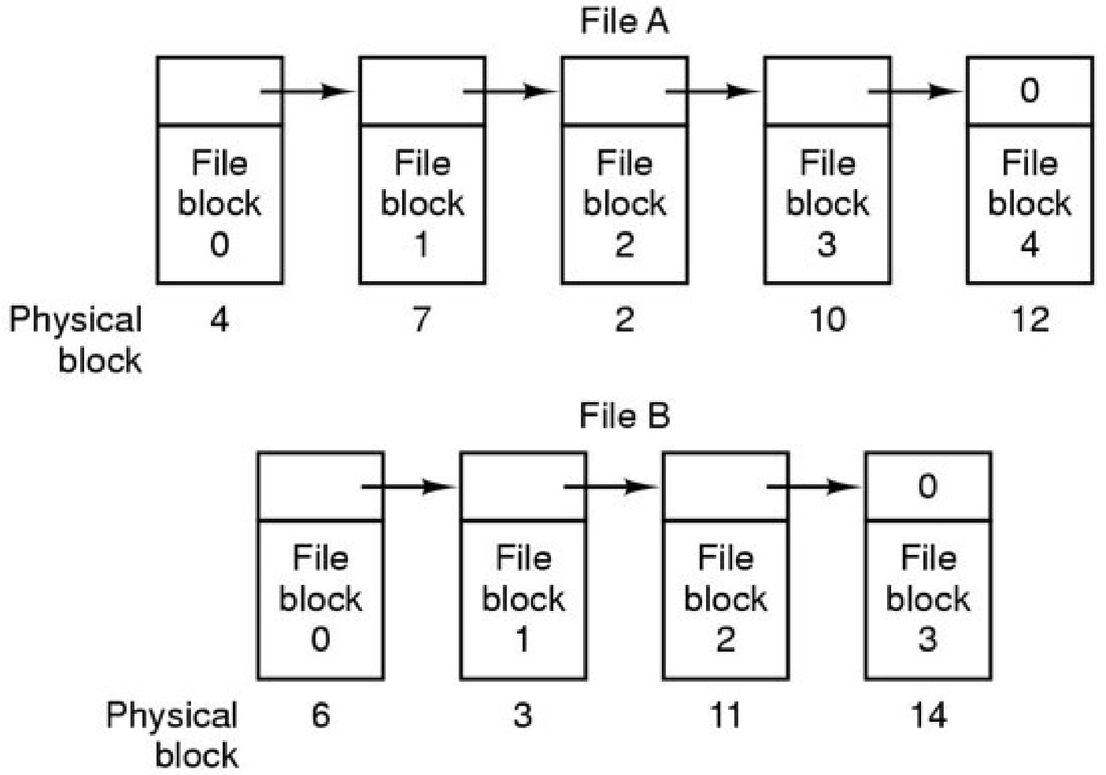
- 缺点:
  - 必须事先知道文件的长度
  - 不能灵活地删除和插入记录
  - 文件不能动态增长
  - 外部碎片问题



- 在CD-ROM文件系统中广泛使用
  - 文件大小事先已知
  - 在使用中文件大小不再改变

链表分配

- 一个文件的信息存放在若干不连续的磁盘块中，各块之间通过指针连接，前一个磁盘块指向下一个磁盘块
- 每个块的第1个字作为指向下一块的指针，块的其余部分存放数据
- 在目录项中，只要存放第一块的地址

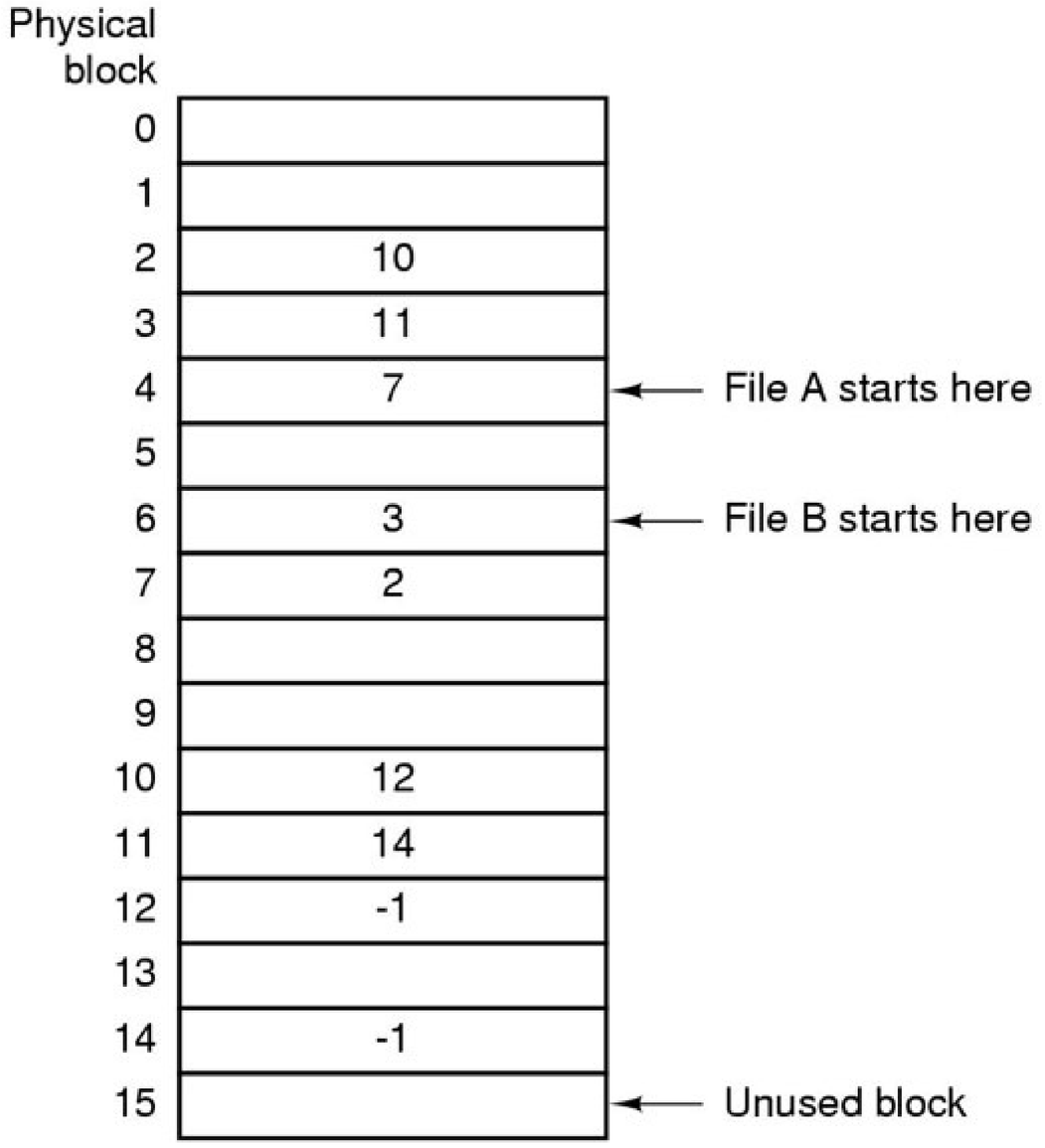


链表分配

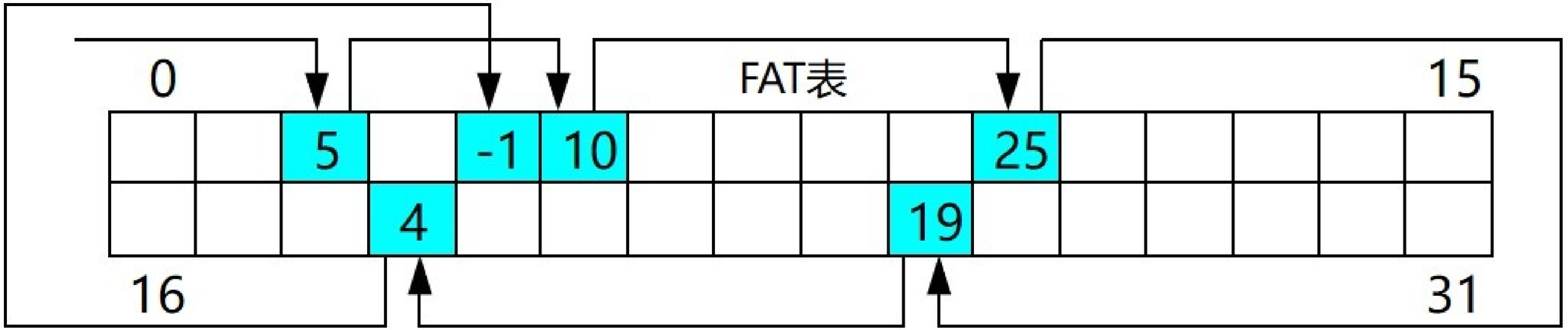
- 优点
  - 充分利用磁盘空间，没有外碎片
  - 顺序访问速度快
  - 对插入、删除和修改记录都非常容易
  - 能适应文件的动态增长，无需事先知道文件的大小
- 缺点
  - 随机访问性能差

FAT分配

- 取出每个磁盘块的指针字，存放在内存中一个单独的表中——FAT(File Allocation Table，文件分配表)
- 在目录项中只需记录起始块号，其余的块可以从FAT表中获得



FAT分配



Hello.c: 2->5->10->25->19->4

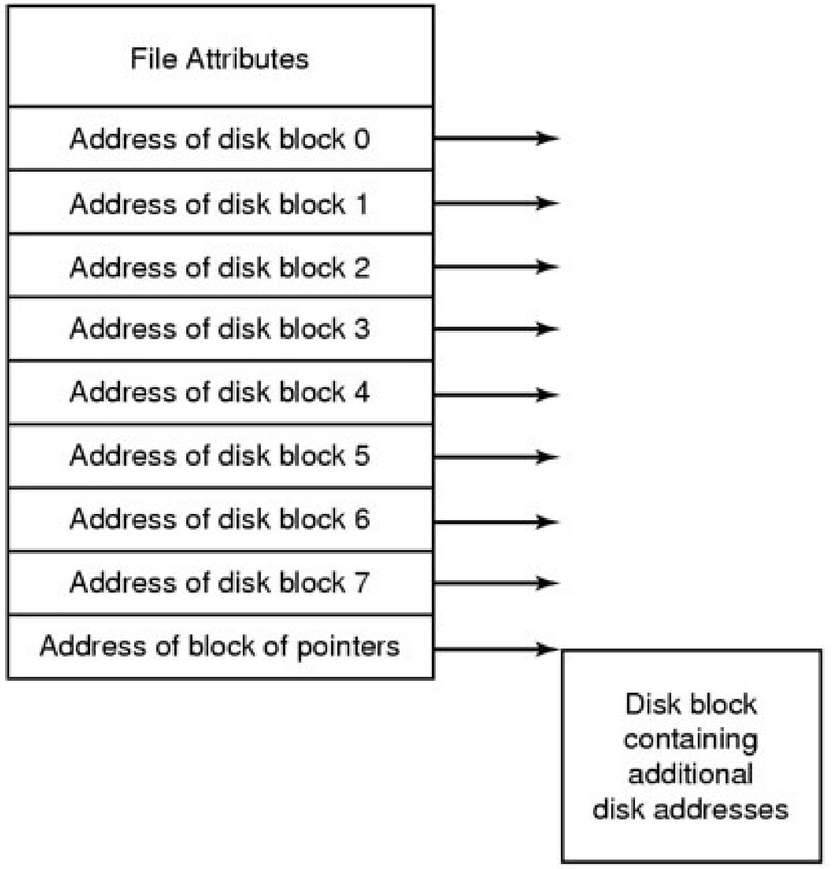
目录	
文件名	起始地址
Hello.c	2

FAT分配

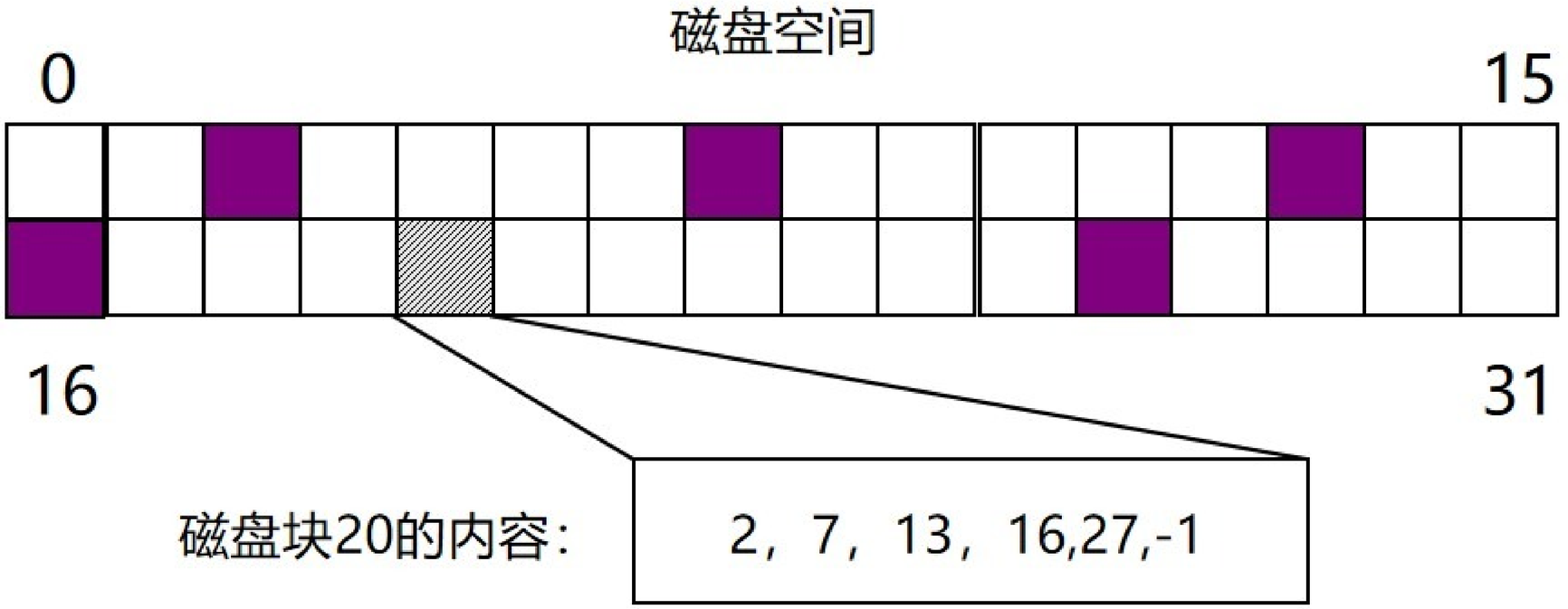
- 优点
  - 在链表分配的基础上，解决了随机访问性能的问题
- 缺点
  - 不能支持高效的直接存取，要对一个较大的文件进行存取，须在FAT中顺序地查找许多盘块号
  - FAT占用内存较大，且一个文件占用的盘块号是随机地分布的，只有将整个FAT调入内存，才能保证在FAT中找到一个文件的所有盘块号

索引分配

- 一个文件的信息存放在若干不连续的磁盘块中，系统为每个文件建立一个专用数据结构——i节点(索引节点)，并将这些块的块号存放在一个i节点中
- 一个i节点就是磁盘块地址数组，其中第n个条目指向文件的第n块
- 当一个文件包含的磁盘块数超出i节点所能容纳的数目时，最后一个项不是指向数据块，而是指向包含附加磁盘地址的磁盘块



索引分配



目录	
文件名	i节点地址
Hello.c	20

- 优点
  - 即能顺序存取，又能随机存取
  - 满足了文件动态增长、插入删除的要求
  - 能充分利用外存空间
- 缺点：
  - 较多的寻道次数和寻道时间
  - 索引表本身带来了系统开销

02

磁盘空间管理

- 空闲表法
- 为每个文件分配一块连续的存储空间
- 系统为外存上的所有空闲区建立一张空闲表
  - 每个空闲区对应于一个空闲表项
  - 表项序号
  - 该空闲区的第一个盘块号
  - 该区的空闲盘块数
- 将所有空闲区按其起始盘块号递增的次序排列，形成空闲盘块表

■ 空闲盘块表

序号	第一空闲盘块号	空闲盘块数
1	2	4
2	9	3
3	15	5
4	—	—

- 存储空间的分配与回收
- 与内存的分区(动态)分配类似
- 分配空闲盘块时，先顺序地检索空闲表的各表项，找到一个大小满足要求的空闲区，再将该盘区分配给用户(进程)，同时修改空闲表
  - 首次适应算法
  - 最佳适应算法

空闲链表法

- 空闲盘块链
  - 将磁盘上的所有空闲空间以盘块为单位组成一条链，其中的每一个盘块都有指向后继盘块的指针
- 空闲盘区链
  - 将磁盘上的所有空闲盘区(每个盘区可包含若干个盘块)组成一条链
  - 在每个盘区上除含有用于指示下一个空闲盘区的指针外，还应有能指明本盘区大小(盘块数)的信息

空闲块位示图

- 利用二进制的一位来表示磁盘中一个盘块的使用情况
- 当其值为“0”时，表示对应的盘块空闲
- 为“1”时，表示已分配
- 磁盘上的所有盘块都有一个二进制位与之对应
- 由所有盘块所对应的位构成的集合，称为位示图

空闲块位示图

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

03

文件系统的可靠性

文件系统的可靠性

- 可靠性——系统抵抗和预防各种物理性破坏和人为性破坏的能力
- 比起计算机的损坏，文件系统破坏的损失要大得多

文件备份

- 文件备份主要是为了解决两个问题：
  - 从意外的灾难中恢复
  - 从错误的操作中恢复
- 文件备份通常是将磁盘文件转储到另一个磁盘上保存。转储有两种方案：
  - 物理转储
  - 逻辑转储

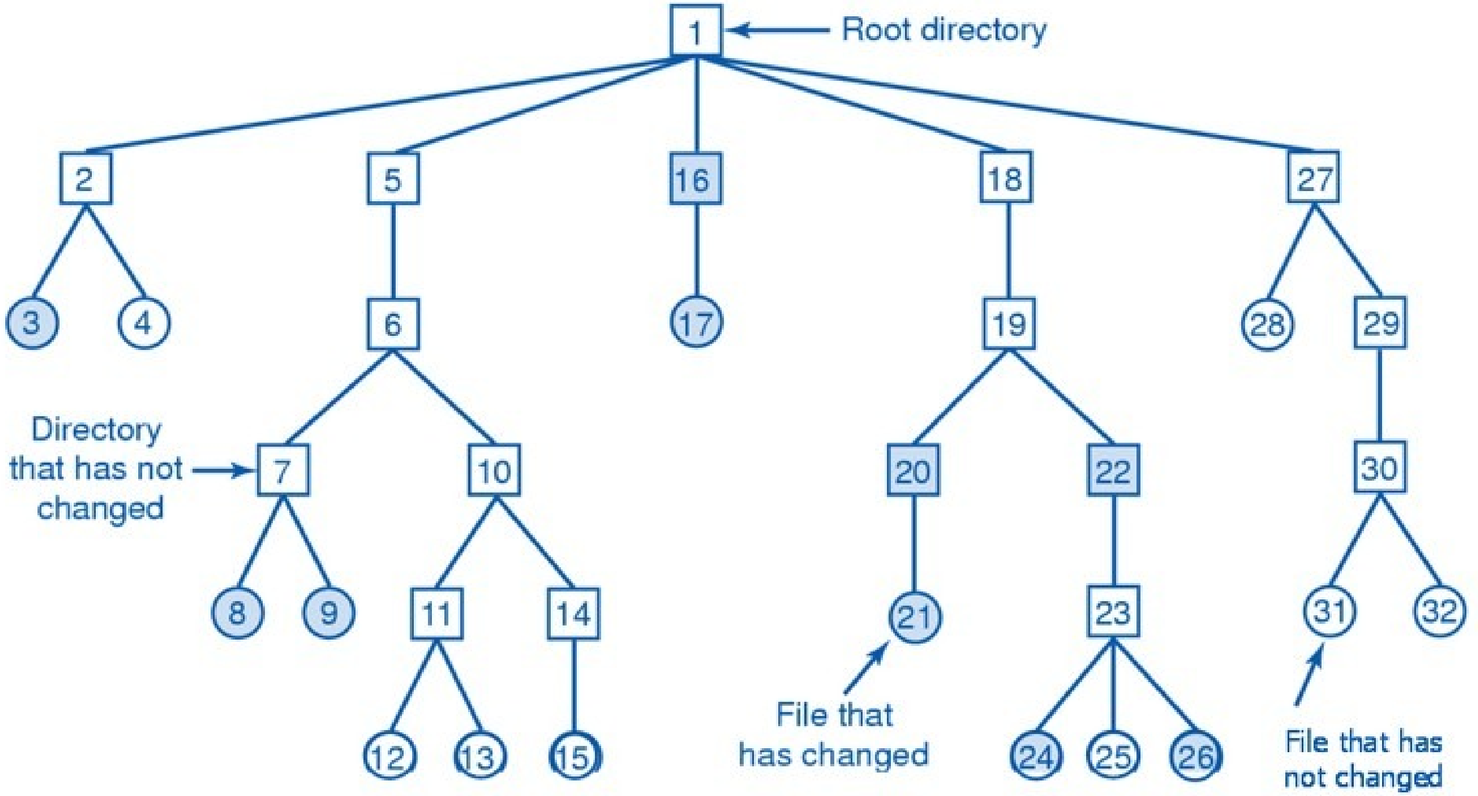
物理转储

- 从磁盘的第0块开始，将全部的磁盘块按顺序输出到另一个磁盘上
- 优点：简单
- 缺点：不能跳过选定的目录、不能实现增量转储、不能恢复单个的文件
- 增量转储：周期性地做全面备份，而每次只对最近更改的数据做备份

逻辑转储

- 从一个或几个指定的目录开始，递归地转储自给定基准日期后有所更改的全部文件和目录
- 优点：很容易满足恢复特定文件或目录的请求
- 缺点：实现复杂

- 被阴影覆盖的项目代表自基准日期以来修改过，因此需要转储，无阴影的不需要转储



逻辑转储

- 需要转储通向修改过的文件或目录上的所有目录(即使其本身未修改过)
  - 方便将这些转储的文件和目录恢复到另一台计算机上
  - 可以对单个文件进行增量恢复

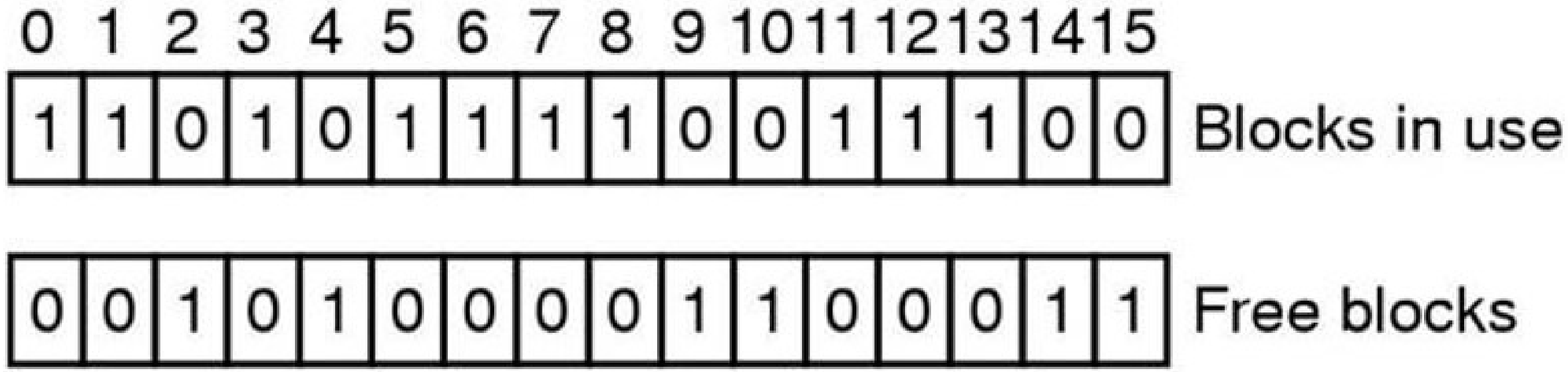
文件系统的一致性

- 文件的修改操作：读取磁盘块，修改，写回磁盘
- 如果修改过的磁盘块全部写回之前系统崩溃，则文件系统可能处于不一致状态
- 文件系统一致性检查
  - 块的一致性检查
  - 文件的一致性检查

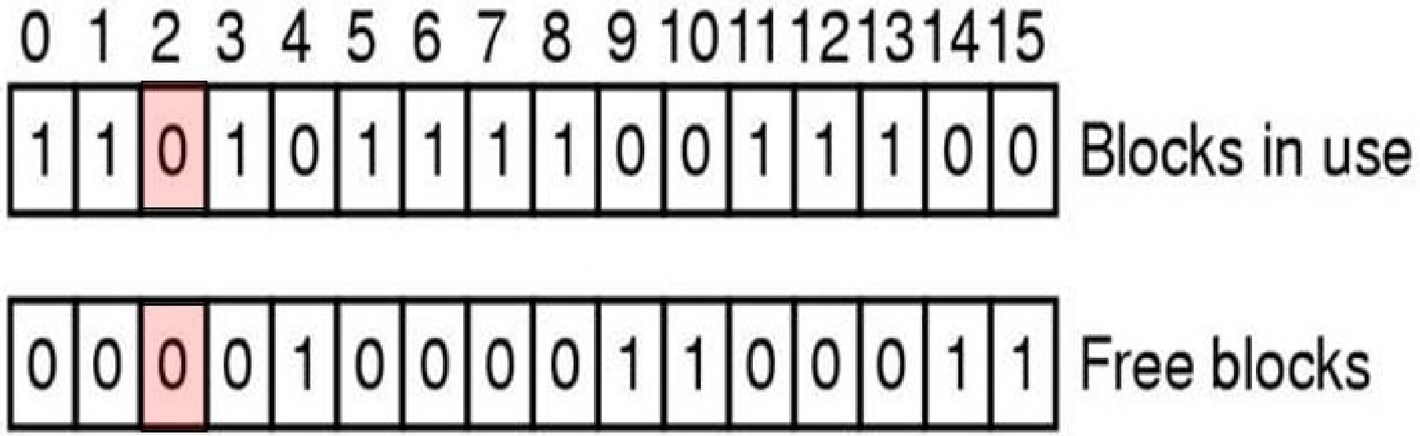
块的一致性检查

- 构造两张表，每张表中为每个块设立一个计数器。第一个表中的计数器跟踪该块在文件中出现的次数；第二个表中的计数器跟踪该块在空闲表中出现的次数
- 两张表每个块对应的计数初值为0
- 检验程序首先读取全部i节点，得到相应文件使用的块的块号。每当读到一个块时，第一个表中的计数值增1；然后检验程序检查空闲表，找到全部未使用的块，并使第二个表中对应的计数值增1

- 一致的情况

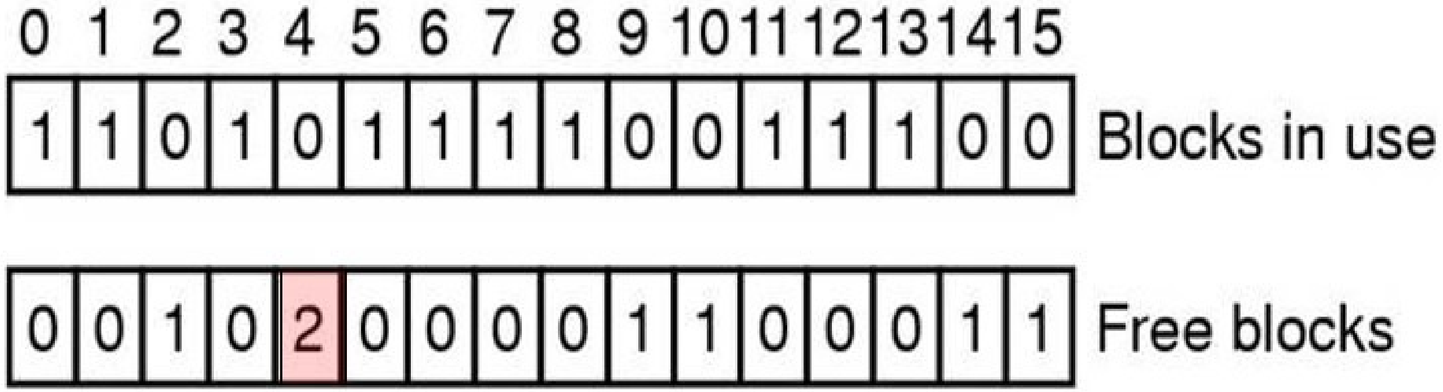

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1 1 0 1 0 1 1 1 1 0 0 1 1 1 0 0
Blocks in use
0 0 1 0 1 0 0 0 0 1 1 0 0 0 1 1
Free blocks

- 块丢失的情况
  - 不会造成大的损害，但是浪费磁盘空间
  - 解决方法：加到空闲表中

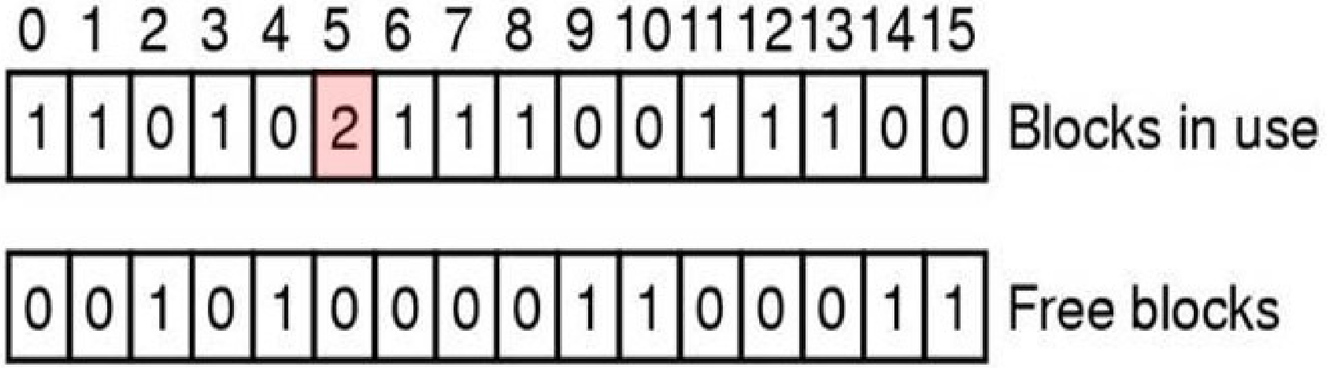


空闲表中有重复块的情况

- 解决方法：重建空闲表



- 重复数据块的情况
  - 解决方法：分配一个空闲块，将块5的内容复制到其中，并将该块插入到一个文件中。这样虽然可以保证文件系统的一致性，但是错误并未消除，因此应该报告这一错误



大纲

01 外存的组织方式

02 磁盘空间管理

03 文件系统的可靠性