操作系统

Operating system

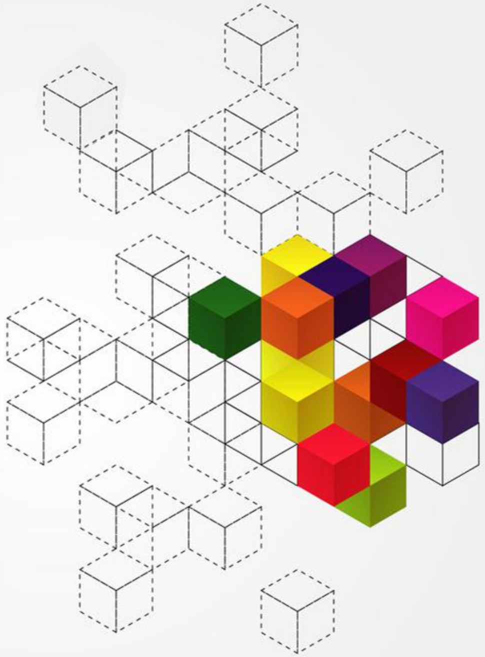
孔维强

大连理工大学

内容纲要　　10.2 文件操作与访问模式

一、 文件操作

二、 文件访问模式

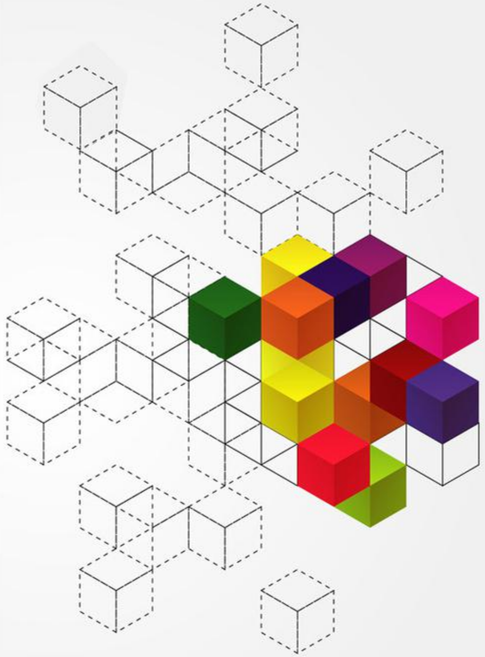


一、文件操作

- 文件是抽象数据类型（**abstract data type**）基本操作
  - **Create** (find a space, entry of the file in directory)
  - **Write –** at **write pointer** location
  - **Read –** at **read pointer** location
  - **Reposition within file - seek**
  - **Delete** (release file space, erase directory entry)
  - **Truncate** (revise file's contents but keeps attributes)

为避免反复搜索目录以查找文件

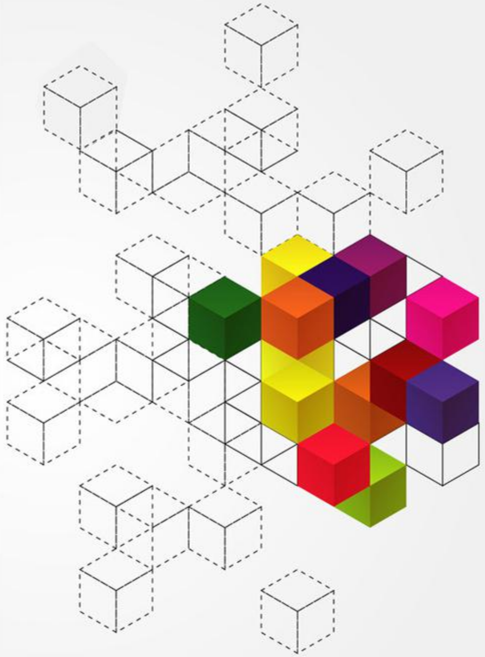
- ***Open(F_i)*** – search the directory structure on disk for entry ***F_i***, and move the content of entry to memory
- ***Close (F_i)*** – move the content of entry ***F_i*** in memory to directory structure on disk



一、文件操作

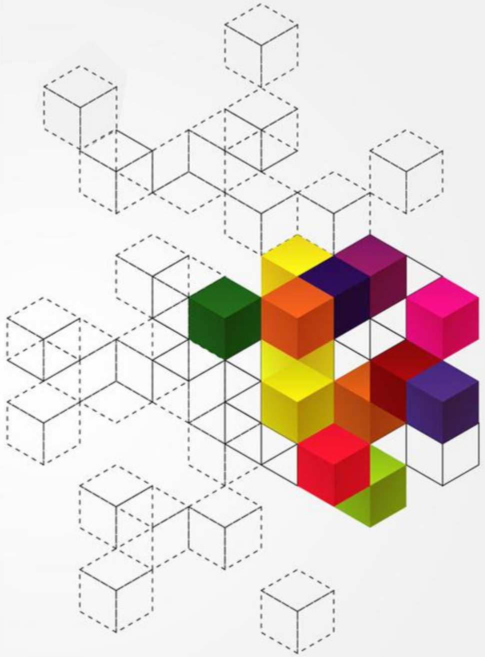
文件常规操作对应的操作内容

文件操作	操作内容
Open	首次打开，创建并返回句柄；非首次打开，在打开文件表中找到句柄并返回
Close	进程关闭文件后，文件句柄在相应进程中不再可用
Create	在磁盘上分配空间，存放文件内容；在目录结构内增加新目录项
Read	读文件内容，并自动调整文件指针
Write	写入文件，并自动调整文件指针
Seek	在文件内重新定位文件指针
Truncate	文件截短（释放文件所占部分空间，调整文件size属性）
Delete	删除文件（删除文件及其目录项）



一、文件操作

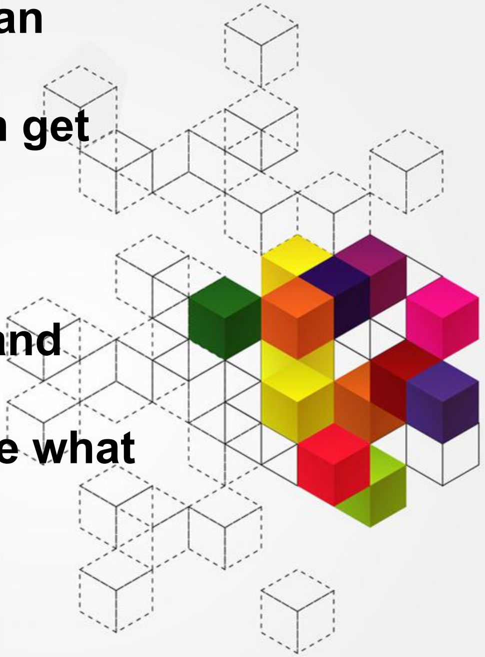
- 管理文件打开需要多个数据:
  - **Open-file table**: tracks open files
  - File pointer: pointer to last read/write location (as the current-file-position), per process that has the file open
  - **File-open count**: counter of number of times a file is open – to allow removal of data from open-file table when last processes closes it
  - Disk location of the file: cache of data access information
  - Access rights: per-process access mode information



一、文件操作——文件打开锁

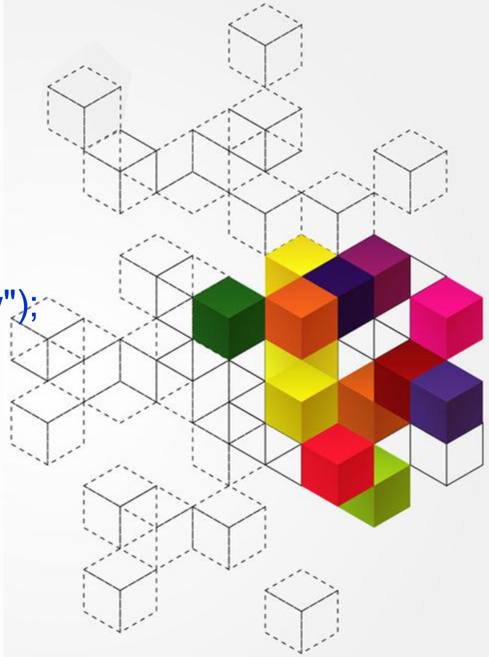
- 某些操作系统和文件系统提供文件打开锁
  - Similar to reader-writer locks
  - Shared lock similar to reader lock – several processes can acquire concurrently
  - Exclusive lock similar to writer lock – only 1 process can get

- 强制性 or 建议性:
  - Mandatory – access is denied depending on locks held and requested (OS decides)
  - Advisory – processes can find status of locks and decide what to do (programmers decide)



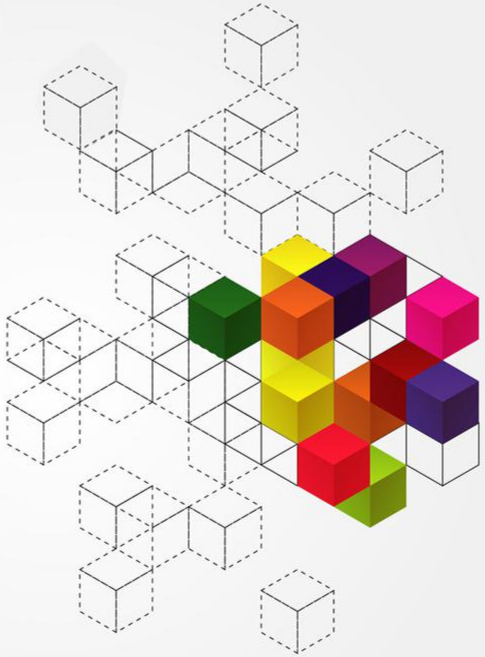
一、文件操作——文件打开锁

```
import java.io.*;
import java.nio.channels.*;
public class LockingExample {
   public static final boolean EXCLUSIVE = false;
   public static final boolean SHARED = true;
   public static void main(String arsg[]) throws IOException {
            FileLock sharedLock = null;
            FileLock exclusiveLock = null;
            try {       RandomAccessFile raf = new RandomAccessFile("file.txt", "rw");
                        // get the channel for the file
                        FileChannel ch = raf.getChannel();
                        // this locks the first half of the file - exclusive
                        exclusiveLock = ch.lock(0, raf.length()/2, EXCLUSIVE);
                        /** Now modify the data . . . */
                        // release the lock
                        exclusiveLock.release();
```



一、文件操作——文件打开锁

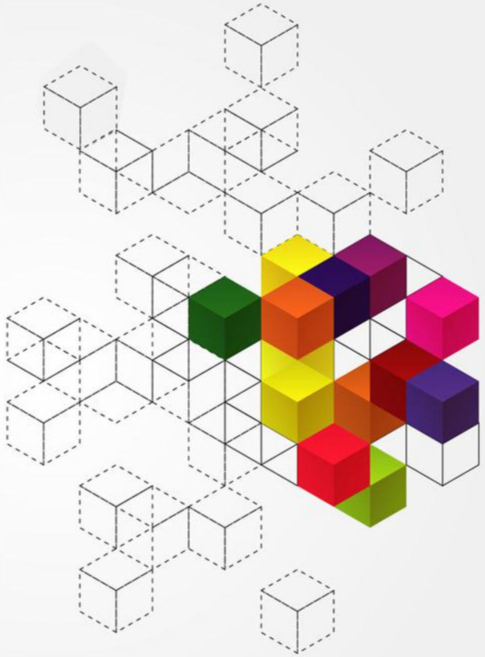
```
            // this locks the second half of the file - shared
            sharedLock = ch.lock(raf.length()/2+1, raf.length(),
            SHARED);
            /** Now read the data . . . */
            // release the lock
            sharedLock.release();
      } catch (java.io.IOException ioe) {
            System.err.println(ioe);
      }finally {
            if (exclusiveLock != null)
            exclusiveLock.release();
            if (sharedLock != null)
            sharedLock.release();
      }
} }
```



二、文件访问模式

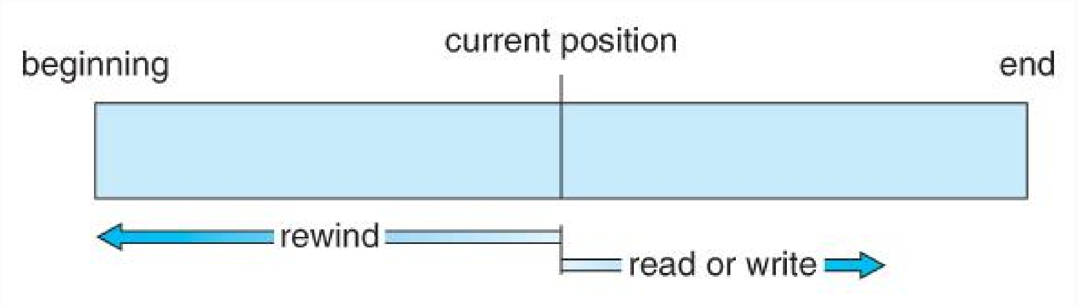
文件访问的三种典型模式

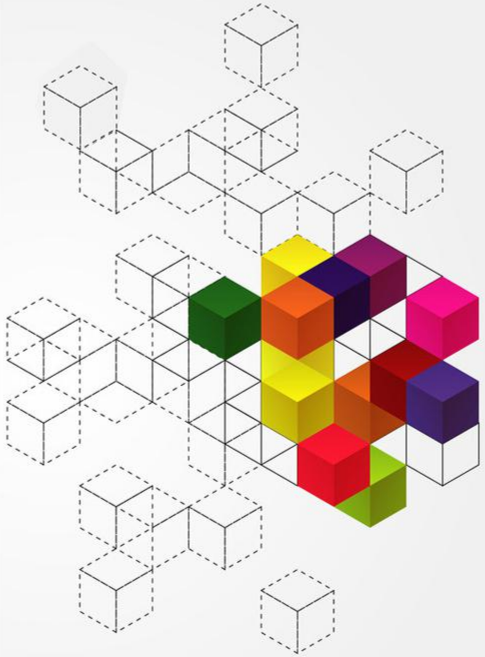
- 顺序访问
- 随机访问
- 索引访问



二、文件访问模式 -顺序访问

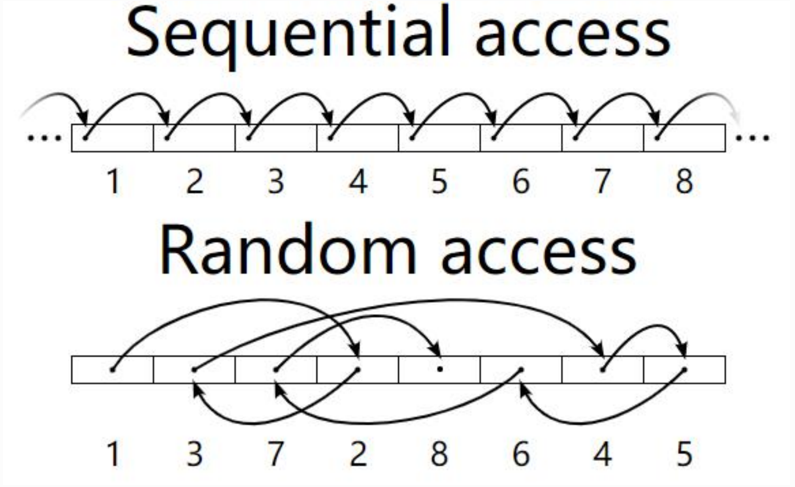
- 从当前文件指针所指地址开始，沿逻辑地址增长的方向依次访问文件内容
  - 基本操作：read next, write next, rewind
  - 典型存储介质: 磁带（Tape）

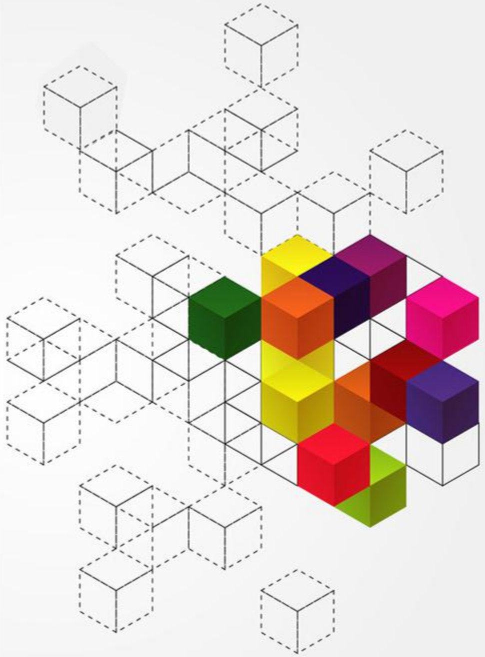




二、文件访问模式 -随机访问

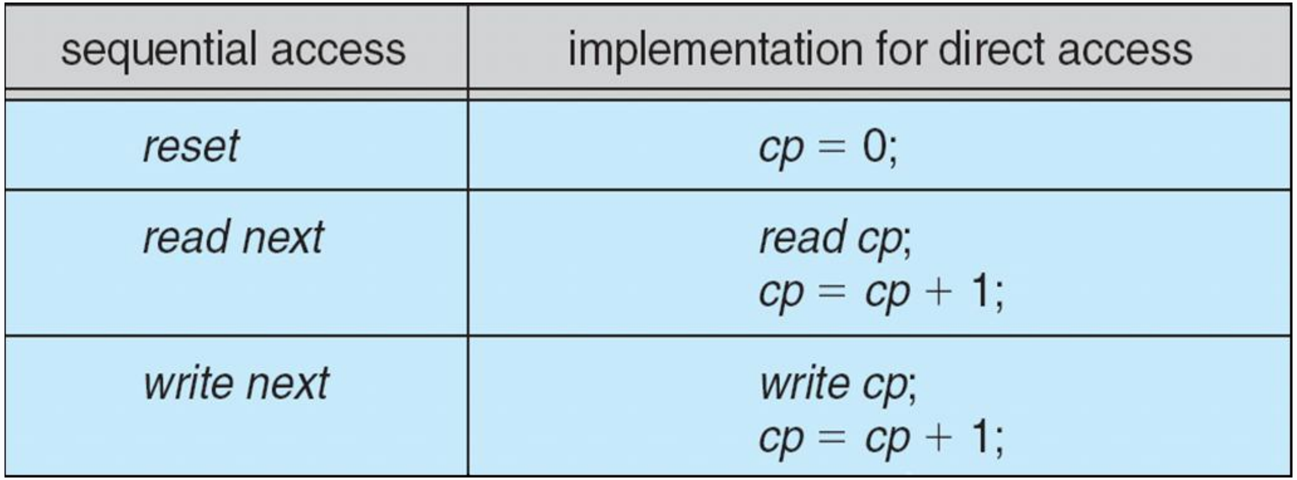
- 直接访问给定逻辑地址的文件内容
  - 基本操作：read(n), write(n), seek(n)
  - 典型存储介质: 磁盘（Magnetic Disk）



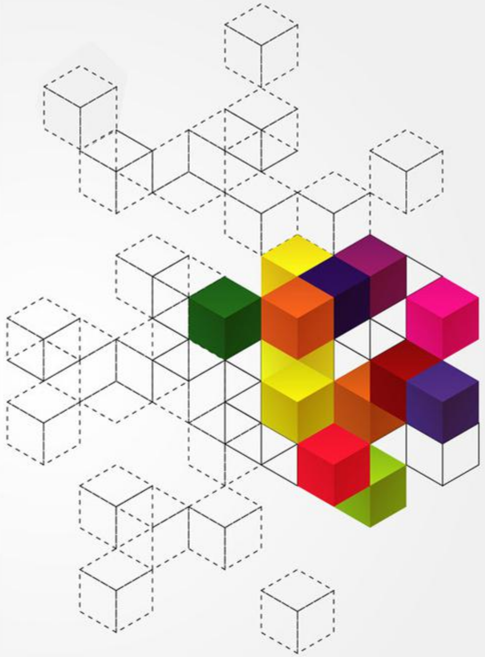


二、文件访问模式 -随机访问

- 直接访问给定逻辑地址的文件内容
  - 基本操作：read(n), write(n), seek(n)
  - 典型存储介质: 磁盘（Magnetic Disk）

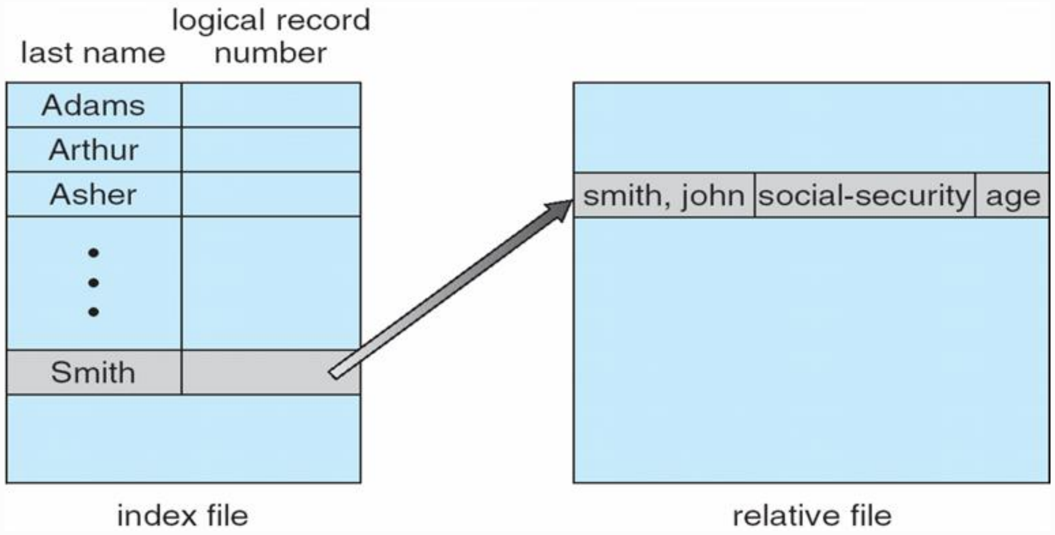


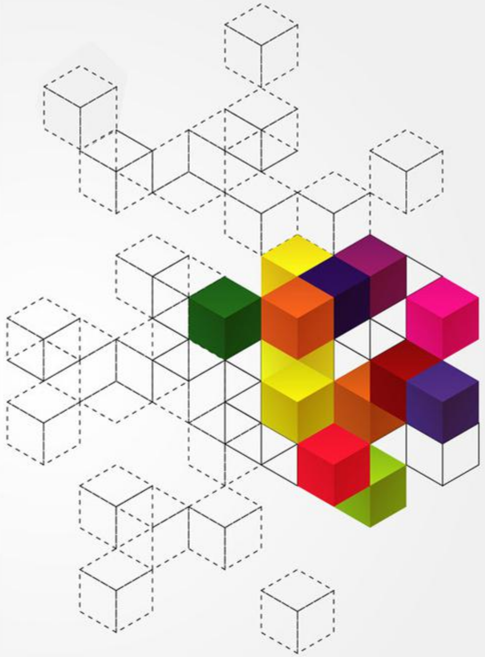
sequential access	implementation for direct access
reset	*cp* = 0;
read next	*read cp*; *cp* = *cp* + 1;
write next	*write cp*; *cp* = *cp* + 1;



二、文件访问模式 -索引访问

- 基于记录关键字建立索引，以索引方式访问文件内容
  - 基本操作：read(key), write(key)
  - 典型应用: 数据库表（DBMS Table Access）





本讲小结

- 文件操作
- 文件访问模式