缓冲流是处理流的一种

①BufferedInputStream

②BufferedOutputStream

③BufferedReader

④BufferedWriter

一，使用BufferedInputStream和BufferedOutputStream实现文件的复制

写完之后加上flush：：：bos.flush();//进行刷新

使用缓冲流可以提高文件操作的效率

long end = System.currentTimeMillis();得到程序运行的时间，返回一个长整型的数据

```
//对文件的复制进行一个加速，相对于fileinputstream
        //和fileoutputstream复制文件的时候更加快了
        FileInputStream  fis=null;
        FileOutputStream  fos = null;
        BufferedInputStream bis = null;
        BufferedOutputStream  bos  =null;
        try {
                //提供读入和写出的文件
                File f1 = new File("e://1.jpg");
                File f2 = new File("e://linmingjun.jpg");
                //创建读取文件的流
                //先创建相应的节点流
                  fis = new FileInputStream(f1);
                 fos = new FileOutputStream(f2);
                //将创建的节点流的对象作为参数传递给缓冲流的构造器中
                 bis = new BufferedInputStream(fis);
                  bos = new BufferedOutputStream(fos);
                //具体的实现文件复制的操作
                byte b [] = new byte[1024];
                int len;
                while((len = bis.read(b)) != -1){
                        bos .write(b,0, len);
                       bos.flush();//进行刷新
```

```
        }
} catch (Exception e) {
        // TODO: handle exception
        e.printStackTrace();
}finally{
        //关闭相应的文件流
        if (bis != null) {
                try {
                        bis.close();
                } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
        }
        if (bos != null) {
                try {
                        bos.close();
                } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
        }
        if (fis != null) {
                try {
                        fis.close();
                } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
        }if (fos != null) {
                try {
                        fos.close();
                } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
```

```
                    }
                }
            }
```