

```

public class BufferExample {
    public static void main(String[] args) throws Exception {
        //입출력 스트림 생성
        String originalFilePath1 =
            BufferExample.class.getResource("originalFile1.jpg").getPath();
        String targetFilePath1 = "C:/Temp/targetFile1.jpg";

        FileInputStream fis = new FileInputStream(originalFilePath1);
        FileOutputStream fos = new FileOutputStream(targetFilePath1);

        //입출력 스트림 + 버퍼 스트림 생성
        String originalFilePath2 =
            BufferExample.class.getResource("originalFile2.jpg").getPath();
        String targetFilePath2 = "C:/Temp/targetFile2.jpg";
        FileInputStream fis2 = new FileInputStream(originalFilePath2);
        FileOutputStream fos2 = new FileOutputStream(targetFilePath2);
        BufferedInputStream bis = new BufferedInputStream(fis2);
        BufferedOutputStream bos = new BufferedOutputStream(fos2);
    }
}

```

실행 결과

버퍼 미사용: 3937437000 ns

버퍼 사용: 24195000 ns

```

//버퍼를 사용하지 않고 복사
long nonBufferTime = copy(fis, fos);
System.out.println("버퍼 미사용:\t" + nonBufferTime + " ns");

//버퍼를 사용하고 복사
long bufferTime = copy(bis, bos);
System.out.println("버퍼 사용:\t" + bufferTime + " ns");

fis.close();
fos.close();
bis.close();
bos.close();
}

public static long copy(InputStream is, OutputStream os) throws Exception {
    //시작 시간 저장
    long start = System.nanoTime();

    //1 바이트를 읽고 1 바이트를 출력
    while(true) {
        int data = is.read();
        if(data == -1) break;
        os.write(data);
    }
    os.flush();

    //끝 시간 저장
    long end = System.nanoTime();
    //복사 시간 리턴
    return (end-start);
}
}

```

※ originalFile1.jpg 와 originalFile2.jpg 파일 복사후 실행하기