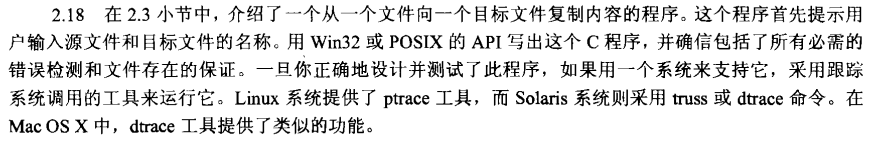
## 实验报告

姓名：王迎旭

学号：16340226

班级：16级软件工程6班

1. **实验目的**



1. **实验条件**

系统：Fedora-Linux

编程语言：C

1. **编程代码**

#include <unistd.h>

#include <assert.h>

#include <sys/mman.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <errno.h>

#include <string.h>

#include <stdio.h>

#include <stdlib.h>

#include <sys/ptrace.h>

#include <sys/wait.h>

#include <sys/reg.h>

#include <sys/syscall.h>

#include <sys/user.h>

void mmapcopy(int src\_fd, size\_t src\_len, int dst\_fd);

int main(int argc, char \*argv[])

{

if (argc != 3) {

printf("Usage: %s<src\_file><dst\_file>\n", argv[0]);

return -1;

}

//文件复制模块

int src\_fd ;

int dst\_fd;

if ((src\_fd = open(argv[1], O\_RDONLY)) < 0){

printf("file1 open failed！\n");

return -1;

}

if ((dst\_fd = open(argv[2], O\_RDWR | O\_CREAT | O\_TRUNC, S\_IRWXU)) < 0) {

printf("file2 open failed! \n");

return -1;

}

struct stat stat;

fstat(src\_fd, &stat); // 获取文件信息

truncate(argv[2], stat.st\_size); // 设置文件大小

mmapcopy(src\_fd, stat.st\_size, dst\_fd);

Close(src\_fd);

close(dst\_fd);

return 0;

}

void mmapcopy(int src\_fd, size\_t src\_len, int dst\_fd)

{

void \*src\_ptr, \*dst\_ptr;

src\_ptr = mmap(NULL, src\_len, PROT\_READ, MAP\_PRIVATE, src\_fd, 0);

dst\_ptr = mmap(NULL, src\_len, PROT\_WRITE | PROT\_READ, MAP\_SHARED, dst\_fd, 0);

if (dst\_ptr == MAP\_FAILED) {

printf("mmap error: %s\n", strerror(errno));

return ;

}

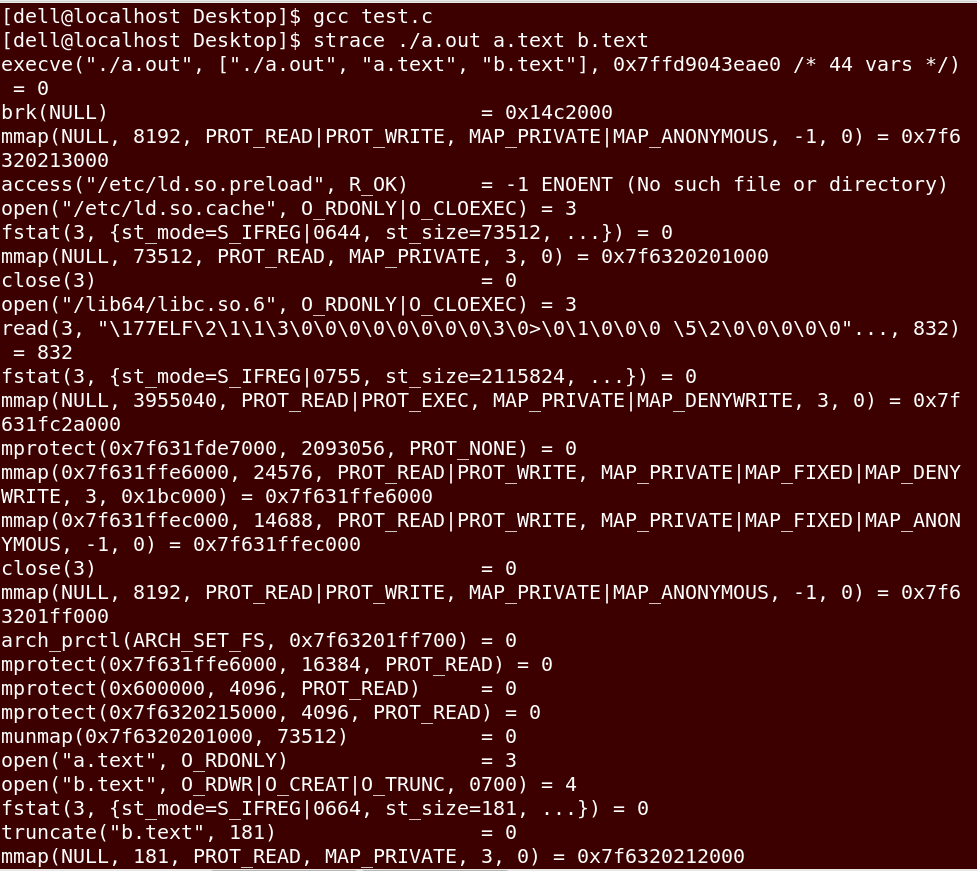
memcpy(dst\_ptr, src\_ptr, src\_len); // 实现拷贝

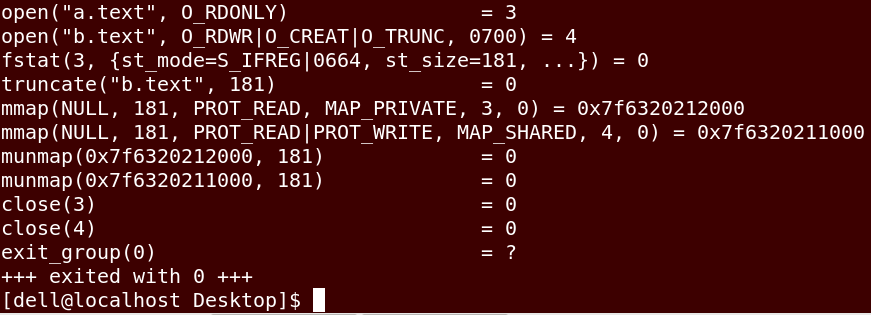
munmap(src\_ptr, src\_len);

munmap(dst\_ptr, src\_len);

}

1. **实验截图**
2. 新建 a.text与b.text，然后随意向a.text文件中输入字符
3. gcc编译test.c(源文件)
4. 运行并监控信息
5. 截图分析





①每一行都是一次系统调用。等号左边是系统调用函数及其参数，右边是返回值。

②系统首先调用execve开始一个新的进程，最后调用exit\_group退出进程，完成 整个程序的执行过程。对于命令行执行的程序，execve或者exec系列中的任何一 个，均为strace输出系统调用中的第一个。strace首先调用fork或者clone函 数新建一个子进程，然后在子进程中调用exec载入需要执行的程序。

③  brk(0)   = 0x8803000,  以0作为参数的调用brk，返回值是内存管理的起始 地址，如果程序调用调用malloc，内存管理分配将从0x8803000地址开始。

④access是检查文件是否存在

⑤使用nmap2函数进行匿名内存映射，以此来获取内存空间，其第二参数就是需要 获取内存空间的长度，返回值为内存空间的起始地址。匿名内存映射就是为了不涉 及具体的文件名，避免了文件的创建和打开，只能用于具有亲缘关系的进程间通信。 真正能与源码对应上的只有write系统调用，其他系统调用基本用于系统初始化工 作，装载被执行程序，载入libc函数库，设置内存映射等。

1. **实验心得**

①编写普通的C程序实现文件的拷贝比较容易，但是由于第一次使用追踪工具，不太懂原理，耗费了比较多的时间在尝试不同的代码嵌套上面，效率也低了很多。

②关于进程监控，此次小实验，在Linux环境下我只是简单的看出系统进行了那些简单的调用，而室友在windows下使用进程追踪工具可以更加清楚的进行进程监控，我觉得对于初学者来说可能更需要去了解一些深入性内在的东西。

③在做题的过程中，多多和周围的人进行交流，有时候会有额外的收获，比如本次的作业，我一开始的做法是在代码中加入追踪工具，后来舍友说代码可以简化，直接用strace工具进行追踪，果然减少了代码量，而且更加直观的反映了追踪过程。