《操作系统综合实验》 实验报告

实验名称	实验日期	实验序号	实验人
系统调用基础	2024.3.23	1	周仙辉

一、 实验题目

在 2.3 节中，我们描述了**一个将一个文件的内容复制到目标文件的程序。**该程序首先提示用户输入源文件和目标文件的名称。使用 Windows 或 **POSIX API** 编写此程序。请务必包括所有必要的错误检查，包括确保源文件是否存在。

正确设计和测试程序后，如果你使用支持它的系统，请使用跟踪系统调用的实用程序运行该程序。Linux 系统提供 **strace** 实用程序，Solaris 和 Mac OS X 系统使用 dtrace 命令。由于 Windows 系统不提供此类功能，您将不得不使用调试器跟踪此程序的 Windows 版本。

二、 相关原理与知识

系统调用和linux使用。

三、 实验过程

打开源文件，读取数据，循环读取，拷贝到目标文件，最后在屏幕输出文件内容。要求在发生错误时，能够反馈一些信息：当源文件不存在；当目标文件存在，是否进行覆盖。

四、 实验结果与分析

输出结果1

这个结果是源文件和目标文件存在，并对目标文件进行覆盖。

openat(AT_FDCWD, "b", O_WRONLY|O_CREAT|O_EXCL, 0644) = -1 EEXIST (File exists) 用了O_CREAT和O_EXCL参数,在文件存在时会返回-1.

openat(AT_FDCWD, "b", O_WRONLY|O_CREAT|O_TRUNC, 0644) = 4 如果选择了覆盖文件，就会执行这句，这句用了O_TRUNC,直接进行覆盖。

```
☺  strace ./copy_file
execve("./copy_file", ["./copy_file"], 0x7ffe85840b30 /* 29 vars */) = 0
brk(NULL)                               = 0x55780cda7000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fff93973030) = -1 EINVAL (Invalid argument)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f4f11b59000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=95243, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 95243, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f4f11b41000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0\0"...,
832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"...,
784, 64) = 784
```

```
pread64(3, "\4\0\0\0 \0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0"...,
48, 848) = 48
pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\302\211\332Pq\2439\235\350\223\322\257\201\326\2
43\f"..., 68, 896) = 68
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2220400, ...}, AT_EMPTY_PATH) =
0
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"...,
784, 64) = 784
mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f4f11918000
mprotect(0x7f4f11940000, 2023424, PROT_NONE) = 0
mmap(0x7f4f11940000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f4f11940000
mmap(0x7f4f11ad5000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1bd000) = 0x7f4f11ad5000
mmap(0x7f4f11b2e000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) = 0x7f4f11b2e000
mmap(0x7f4f11b34000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f4f11b34000
close(3)                                = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f4f11915000
arch_prctl(ARCH_SET_FS, 0x7f4f11915740) = 0
set_tid_address(0x7f4f11915a10)         = 1015
set_robust_list(0x7f4f11915a20, 24)     = 0
rseq(0x7f4f119160e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7f4f11b2e000, 16384, PROT_READ) = 0
mprotect(0x55780b2f6000, 4096, PROT_READ) = 0
mprotect(0x7f4f11b93000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY})
= 0
munmap(0x7f4f11b41000, 95243)           = 0
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...},
AT_EMPTY_PATH) = 0
getrandom("\x7c\x62\xa2\xaf\xac\xdd\x2b\x1f", 8, GRND_NONBLOCK) = 8
brk(NULL)                               = 0x55780cda7000
brk(0x55780cdc8000)                     = 0x55780cdc8000
write(1, "please input the source file:\n", 30please input the source file:
) = 30
newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...},
AT_EMPTY_PATH) = 0
read(0, a
"a\n", 1024)                    = 2
write(1, "please input the target file:\n", 30please input the target file:
) = 30
read(0, b
"b\n", 1024)                    = 2
openat(AT_FDCWD, "a", O_RDONLY)         = 3
openat(AT_FDCWD, "b", O_WRONLY|O_CREAT|O_EXCL, 0644) = -1 EEXIST (File exists)
write(1, "target file already exits\n", 26target file already exits
) = 26
write(1, "input 1 to abort the program or "..., 69input 1 to abort the program or
input 2 to replace the existing file
) = 69
read(0, 2
"2\n", 1024)                    = 2
openat(AT_FDCWD, "b", O_WRONLY|O_CREAT|O_TRUNC, 0644) = 4
read(3, "t", 1)                         = 1
```

```
write(4, "t", 1)                        = 1
read(3, "o", 1)                         = 1
write(4, "o", 1)                        = 1
read(3, "d", 1)                         = 1
write(4, "d", 1)                        = 1
read(3, "a", 1)                         = 1
write(4, "a", 1)                        = 1
read(3, "y", 1)                         = 1
write(4, "y", 1)                        = 1
read(3, " ", 1)                         = 1
write(4, " ", 1)                        = 1
read(3, "i", 1)                         = 1
write(4, "i", 1)                        = 1
read(3, "s", 1)                         = 1
write(4, "s", 1)                        = 1
read(3, " ", 1)                         = 1
write(4, " ", 1)                        = 1
read(3, "t", 1)                         = 1
write(4, "t", 1)                        = 1
read(3, "o", 1)                         = 1
write(4, "o", 1)                        = 1
read(3, "d", 1)                         = 1
write(4, "d", 1)                        = 1
read(3, "a", 1)                         = 1
write(4, "a", 1)                        = 1
read(3, "y", 1)                         = 1
write(4, "y", 1)                        = 1
read(3, "\n", 1)                        = 1
write(4, "\n", 1)                       = 1
read(3, "", 1)                          = 0
write(1, "the end of file\n", 16the end of file
)       = 16
close(4)                                = 0
lseek(3, 0, SEEK_SET)                   = 0
read(3, "today is today\n", 1000)       = 15
write(1, "the contents of the file:\n", 26the contents of the file:
) = 26
write(1, "today is today\n\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"..., 1000today is
today
) = 1000
close(3)                                = 0
lseek(0, -1, SEEK_CUR)                  = -1 ESPIPE (Illegal seek)
exit_group(0)                           = ?
+++ exited with 0 +++
```

对应的输出：

```
hx@DESKTOP-IQF56KM /mnt/d/OS/2
☻  ./copy_file
please input the source file:
a
please input the target file:
b
target file already exits
input 1 to abort the program or input 2 to replace the existing file
2
the end of file
the contents of the file:
today is today
```

分析

brk() 改变数据段的结尾位置。

arch_prctl() sets architecture-specific process or thread state.

mmap()在虚拟空间创建一个新的映射。

access("/etc/ld.so.preload", R_OK)：检查调用进程是否可以访问文件路径名。如果路径名是符号链接，则取消引用它。这里检查是否预加载了特定的共享库。

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3：打开对应的加载器的缓冲文件，文件以只读，以及执行新程序时不会保持打开状态。句柄为3。

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=95243, ...}, AT_EMPTY_PATH) = 0：获取加载器的状态信息。3代表加载器。

后面再打开了libc库

pread64():读入libc数据，并写入缓冲区。

mprotect():对应地址空间赋予权限。

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0：限制进程使用的资源，这里限制的是栈的大小。

munmap():取消相应地址的映射

getrandom():获取随机数字

scanf对应read，puts，printf对应write

open对应openat

lseek把文件的偏移置0

五、 问题总结

程序运行的前面部分功能是什么：

打开加载器，打开libc库，内存映射，内存保护属性，资源限制等一些功能。

openat()和open()区别：

主要区别在于 `openat()` 可以相对于任意目录打开文件，而 `open()` 只能相对于当前工作目录打开文件。

六、 源代码

代码先要求输入两个文件名字，然后对源文件和目标文件的存在性进行判断，并进行相应操作。open相关的参数的使用参考[open(2) - Linux manual page (man7.org)](open(2) - Linux manual page (man7.org))。

```
#include<stdio.h>

#include<unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#define lenth 1000
```

```
char input[lenth],output[lenth];
char screen[lenth];

int main(){
        printf("please input the source file:\n");
        scanf("%s",input);
        printf("please input the target file:\n");
        scanf("%s",output);
        int fd1 = open(input,O_RDONLY);
        if(fd1 == -1){
                puts("source file not exists");
                exit(0);
        }
        int fd2 = open(output, O_WRONLY|O_EXCL|O_CREAT ,0644);
        //printf("%d",fd2);
        if(fd2 == -1){
                puts("target file already exits");
                puts("input 1 to abort the program or input 2 to replace the
existing file");
                while(1){
                        int choice;
                        scanf("%d",&choice);
                        if(choice == 1){
                                close(fd1);
                                exit(0);
                        }else if(choice == 2){
                                fd2 =
open(output,O_WRONLY|O_CREAT|O_TRUNC,0644);
                                //printf("%d\n",fd2);
                                break;
                        }else{
                                puts("invalid input...please input again");
                                continue;
                        }
                }
        }

        char x[2];
        int mark = 1;
        while(mark != -1){
                mark = read(fd1,&x,1);
                if(mark == 0){
                        puts("the end of file");
                        break;
                }else if(mark == -1){
                        puts("input wrong");
                        close(fd1);
                        close(fd2);
                        exit(0);
                }
                mark = write(fd2,&x,1);
                if(mark == -1){
                        puts("no more disk space ");
                        close(fd1);
                        close(fd2);
                        exit(0);
                }
        }
```

```
        close(fd2);

        mark = lseek(fd1, 0, SEEK_SET);//将偏移重置
        if(mark == -1){
                puts("Error seeking file");
                exit(0);
        }

        mark = read(fd1,screen,lenth);
        if(mark == -1){
                puts("read to screen wrong");
                close(fd1);
                exit(0);
        }

        puts("the contents of the file:");
        mark = write(1,screen,lenth);
        if(mark == -1){
                puts("write to screen wrong");
                close(fd1);
                exit(0);
        }

        close(fd1);
        return 0;
}
```