项目一：实现简单的文件复制

1 学时

- 2 学时

2 实验目的

- 理解、掌握、应用文件的基本概念，以及 open、close、read、write 等函数的使用。

3 实验内容

- 编写程序，实现“cp 文件名 1 文件名 2”命令的基本功能，将该功能做成一个独立的函数 copy 去调用。

4 实验原理

4.1 命令行参数

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

argc：代表命令行中输入项的个数（包括可执行文件名）

argv：数组长度为 argc。argv[0]~argv[argc-1]指向命令行各输入项的字符串首地址，argv[argc]存放 NULL。

如执行时输入：

./a.out –f foo bar

argc = 4

argv 所指向的数组元素个数为 5

4.2 perror 函数的用法

void perror(const char *msg);

功能：显示当前 errno 对应的错误信息

参数：msg-用户自定义的提示信息

输出格式“msg：错误信息”

假设函数 func 若返回-1 并设置错误代码，一般可以这样使用：

```
if(func()==-1)
{
    perror(“func”);
    exit(1);//此处根据程序具体情况决定是否退出进程
}
```

这样一旦 func 执行出错，程序员就可以直接看到对应的错误信息，从而解决错误。

4.3 open、close、read、write 函数的用法

（1）open

#include <sys/stat.h>

#include <fcntl.h>

int open(const char *path, int flags, mode_t perms);

参数：

path：文件名，其他目录需要写明路径

flags：打开文件的方式，用或运算

perms：文件权限，仅创建时需要

返回值：

成功返回最小可用的文件描述符，即访问当前文件所用的文件描述符。

失败返回-1(置 errno)

例子：

1）以读写方式打开文件 f1

int fd;

fd=open(“f1”, O_RDWR);

if(fd==-1) { perror(“open”); exit(1); }

2）以读写方式打开文件/home/f2，文件存在则清 0，不存在则创建，设置权限为用户可读可写，同组可读，其他可读。

int fd;

fd=open(“/home/f2”,O_RDWR|O_TRUNC|O_CREAT,0644);

if(fd==-1) { perror(“open”); exit(1); }

（2）close

#include <unistd.h>

int close(int fd);

功能：关闭文件，释放文件描述符，使之可再利用

参数：fd，文件描述符

返回值：成功返回 0，否则返回-1（置 errno）

（3）write

#include <unistd.h>

ssize_t write(int fd, const void *buf, size_t nbytes) ;

// ssize_t 相当于 int

参数：

fd：文件描述符

buf：要写入文件的数据区首地址

nbytes：要写入文件的数据字节个数

返回值：成功返回已写字节数，失败返回-1(置 errno)

例子：

```
char buff[20]=”hello”;
write(fd,  buff,  sizeof(buff));
```

（4）read

```
#include <unistd.h>
ssize_t read(int fd, void *buf, size_t nbytes );
```

参数：

fd：文件描述符

buf：保存读取数据的缓冲区

nbytes：要读取数据的字节数

返回值：成功返回已读取字符数，失败返回-1(置 errno)

例子：

```
char buff[20];
write(fd,  buff,  sizeof(buff)-1);
```

5 预习要求和技术准备工作

- 掌握 main 函数命令行参数的用法
- 掌握 open、close、read、write 函数的使用
- 掌握系统调用函数调用出错的判断方法及错误输出函数 perror 的用法

6 实验环境

- PC 机
- 在 Windows 环境中的 VMware 虚拟机上运行 RedhatLinux9.0 操作系统或者独立的 Redhat Linux 9.0 操作系统
- 基于 Linux 的 vi 编辑器和 gcc 编译器

7 实验设计及操作步骤

7.1 以 root 身份登录系统，在/home 目录中创建目录 exp41

```
cd  /home
mkdir  exp41
```

7.2 进入刚创建的目录

```
cd  exp41
```

7.3 使用 vi 编辑文件，文件名为 mycp.c

```
vi  mycp.c
```

编写程序，实现“cp 文件名 1 文件名 2”的功能，即执行“mycp 文件名 1 文件名 2”能够实现将文件 1 的内容复制到文件 2 的功能。

★要求：

将复制文件 1 到文件 2 的功能独立做成函数 copy，原型为：

void copy(char *from, char *to);

参数代表被拷贝的文件 1 和要拷入的文件 2。

★注意：

（1）根据要求，本程序要通过命令行参数实现

（2）可以使用 diff 命令（格式：diff 文件名 1 文件名 2）查看两个文件是否有区别

★编程思路

void copy(char *from, char *to)

{

①以只读的方式打开被拷贝的文件*from

②以写的方式打开要写入的文件*to（对于打开方式再思考一下，还有什么补充没？）

③从*from 文件中读出数据写入到*to 文件中，使用循环，每次完成一部分数据的读出和写入（注意结束条件）

④关闭文件

}

main(int argc,char *argv[])

{

①判断 argc 是否符合要求，不符合则给出用法提示，退出程序。

②调用 copy 函数完成文件复制

}

7.4 编译 mycp.c 为可执行文件 mycp，并能正确执行。

编译：gcc mycp.c -o mycp

7.5 执行“./mycp /etc/profile ./profile1”，然后执行命令“ls –l /etc/profile ./profile1”，将执行结果屏幕截屏。参考示例如下：

```
cst@ubuntu:~/linuxsys2021/41/teacher$ ./mycp /etc/profile ./profile1
cst@ubuntu:~/linuxsys2021/41/teacher$ ls -l /etc/profile ./profile1
-rw-r--r-- 1 root root 581 12月  5  2019 /etc/profile
-rw------- 1 cst  cst  581 7月  23 10:48 ./profile1
```

8 实验报告提交要求：

将实验操作每个步骤中的命令、源程序以及截图写入实验报告，实验报告命名为“学号姓名-实验 41.doc”，并提交。

9　项目思考

如果被拷贝的是目录（可能非空），该如何做？现在的知识够吗？上网查查资料吧！