

《操作系统原理》实验报告			
实验名称	系统调用基础	实验序号	1
实验日期	2023/03/20	实验人	盖乐
<p>一、实验题目</p> <p>实现一个将一个文件的内容复制到另外一个文件的程序，并且程序通过用户传入的第一个参数作为目的地，第二个参数作为数据来源，完成这个程序时请使用 Posix 或者 windows api 来实现，并且做好相应所有必要的错误的检测。</p>			
<p>二、相关原理与知识</p> <p>通过用户键入的命令行参数来获得目标文件和工作源文件的地址，计算出文件的大小，再通过相应的 open read write 系统调用来实现文件的打开读写，就能实现文件的复制了</p>			
<p>三、实验过程</p> <p>1. 读取参数，参数判断并提示</p> <pre>if(argc != 3) { usage(argv[0]); return 0; }</pre> <pre>static void usage(char *name) { fprintf(stderr, "Usage: %s source [dest]\n", name); }</pre> <p>2. 打开源文件并检查错误</p> <pre>int src = open(argv[1], O_RDONLY); if(src < 0) { fprintf(stderr, "Error: cant open source file.\n"); return 0; }</pre> <p>3. 打开目标文件并检查错误</p> <pre>int dest = open(argv[2], O_WRONLY O_CREAT O_TRUNC, 0644); if(dest < 0) { fprintf(stderr, "Error: cant open dest file.\n"); return 0; }</pre>			

4. 使用循环读取到缓冲区并写到新地址

```
int pos = 0;
while((pos = read(src, buf, 1024)) > 0)
{
    write(dest, buf, pos);
}
```

四、实验结果与分析

1. 使用 gcc 连接并编译可执行文件

```
gaile@gaile-virtual-machine:~/Code/operation system experiment$ gcc op1.c -o op1
```

2. Usage 提示

```
gaile@gaile-virtual-machine:~/Code/operation system experiment$ ./op1
Usage: ./op1 source [dest]
```

3. 然后使用运行程序，进行文件复制操作

```
gaile@gaile-virtual-machine:~/Code/operation system experiment$ ./op1 op1.c cp1.txt
```

- 4.通过 vi 查看目标文件复制结果

```
gaile@gaile-virtual-machine:~/Code/operation system experiment$ vi cp1.txt
```

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
static void usage(char*);
int main(int argc, char* argv[])
{
    char buf[1024];
    if(argc != 3) {
        usage(argv[0]);
        return 0;
    }
    int src = open(argv[1], O_RDONLY);
    if(src < 0) {
        fprintf(stderr, "Error: cant open source file.\n");
        return 0;
    }
    int dest = open(argv[2], O_WRONLY|O_CREAT|O_TRUNC, 0644);
    if(dest < 0) {
        fprintf(stderr, "Error: cant open dest file.\n");
        return 0;
    }
    int pos = 0;
    while((pos = read(src, buf, 1024)) > 0)
    {
        write(dest, buf, pos);
    }
    close(src);
    close(dest);
    return 0;
}

static void usage(char *name)
{
    fprintf(stderr, "Usage: %s source [dest]\n", name);
}
```

此程序使用了 Linux 系统调用来打开、读取、写入文件，并且在必要时关闭文件。Linux 系统调用是操作系统提供给用户程序的接口，可以访问底层系统资源（如文件、网络、进程、内存等），并且可以在用户空间和内核空间之间进行数据传输。

在此程序中使用的 Linux 系统调用包括：

① `open()`：打开文件，并返回一个文件描述符（file descriptor），表示对该文件的访问权限和状态。它接收文件路径、访问模式和文件权限作为参数。本程序使用该系统调用打开源文件和目标文件。

② `read()`：从文件描述符指定的文件中读取数据，并将数据存储到指定的缓冲区中。它接收文件描述符、缓冲区和要读取的字节数作为参数。本程序使用该系统调用从源文件中读取数据。

③ `write()`：向文件描述符指定的文件中写入数据。它接收文件描述符、缓冲区和要写入的字节数作为参数。本程序使用该系统调用将从源文件中读取的数据写入目标文件中。

④ `close()`：关闭文件描述符，并释放与该文件描述符相关的系统资源。它接收文件描述符作为参数。本程序使用该系统调用关闭打开的源文件和目标文件。

五、问题总结

1.源文件错误报错，无法找到源文件地址

```
gaile@gaile-virtual-machine:~/Code/operation system experiment$ ./op1 o1.c cp
Error: cant open source file.
```

2.目标文件错误报错，无法找到目标文件地址

```
gaile@gaile-virtual-machine:~/Code/operation system experiment$ ./op1 op1.c ./
Error: cant open dest file.
```

3.使用 `strace` 分析系统调用

[illegible]

六、源代码

```
#include <stdio.h>
```

```
#include <fcntl.h>
```

```
#include <unistd.h>
```

```
static void usage(char*);
```

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

{

```
char buf[1024];
```

```
if(argc != 3) {
```

```
usage(argv[0]);
```

```
return 0;
```

```
}  
  
int src = open(argv[1], O_RDONLY);  
if(src < 0) {  
    fprintf(stderr, "Error: cant open source file.\n");  
    return 0;  
}  
  
int dest = open(argv[2], O_WRONLY|O_CREAT|O_TRUNC, 0644);  
if(dest < 0) {  
    fprintf(stderr, "Error: cant open dest file.\n");  
    return 0;  
}  
  
int pos = 0;  
while((pos = read(src, buf, 1024)) > 0)  
{  
    write(dest, buf, pos);  
}  
  
close(src);  
close(dest);  
return 0;  
}  
  
static void usage(char *name)  
{  
    fprintf(stderr, "Usage: %s source [dest]\n", name);  
}
```