实验五：文件备份实验

一、实验目的

（1）熟悉 Linux 文件系统的文件和目录结构
（2）掌握文件系统的基本特征
（3）掌握常用文件操作函数。

二、实验内容

编写 C 程序模拟实现 Linux 文件系统的简单 I/O 流操作：备份文件，将源文件 source.dat 备份为 target.dat 文件。要求：

（1）使用 C 库函数实现文件备份；
（2）使用系统调用函数实现文件备份。

三、实验指导

1. 使用 C 库函数实现文件备份

对于实验要求（1），涉及的 C 库函数有 fopen()、fclose()、fread()和 fwrite()，并需要经过以下步骤：

（1）使用 fopen()函数以只读方式打开想要备份的源文件 source 和以只写方式打开想要写入内容的目标文件 target。
（2）使用 fread()循环读取源文件一个缓冲区大小的内容，使用 fwrite()将内容写入目标文件。
（3）读取与写入完毕，使用 fclose()关闭读写文件流。

2. 使用系统调用实现文件备份

对于实验要求（2），涉及的 Linux 相关系统调用有有 open()、close()、read()和 write()，并需要经过以下步骤：

（1）使用 open()系统调用函数以只读方式打开想要备份的源文件 source 和以只写方式打开想要写入内容的目标文件 target。
（2）使用 read()循环读取源文件一个缓冲区大小的内容，使用 write()将内容写入目标文件。
（3）读取与写入完毕，使用 close()关闭读写文件流。

四、实验结果

五、实验思考及总结

（1）使用系统调用函数 open(),read(),write(),close()实现简单文件备份的原理是什么？
（2）使用 C 库函数 fopen(), fread(), fwrite(), fclose() 来实现简单文件备份的原理是什么？
（3）上述两者的区别是什么？

文件操作函数介绍

本小节介绍的是 Linux 提供的最常用的操作文件的系统调用，特殊情况：如设备驱动、目录读写、网络连接等特殊文件除外。需要注意的是，这些函数不是 C 语言库函数中提供的文件操作函数。C 库函数提供的函数，如 fopen()、fclose()、fread()、fwrite()等，请读者参见 C 语言中有关文件操作的帮助文档。

1. open()函数

功能描述：打开或创建文件，在打开或创建文件时可指定文件的属性及用户的权限等。

所需头文件:

```
#include <sys/types.h>
#include <sys/stat.h>,
#include <fcntl.h>
```

函数原型:

```
int open(const char *pathname,int flags,int perms)
```

返回值：成功：返回文件描述符；失败：返回-1。

2. close()函数

功能描述：关闭一个被打开的的文件。

所需头文件：

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

函数原型:

```
int close(int fd)
```

函数返回值：0 成功，-1 出错。

3. read()函数

功能描述： 从文件读取数据。

所需头文件：

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

函数原型：

```
ssize_t read(int fd, void *buf, size_t count);
```

返回值：返回所读取的字节数；0（读到 EOF）；-1（出错）。

4. write()函数

功能描述： 向文件写入数据。

所需头文件：

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

函数原型：

```
ssize_t write(int fd, void *buf, size_t count);
```

返回值：写入文件的字节数（成功）；-1（出错）。

5. lseek()函数

功能描述： 设置文件指针，即指定文件偏移量的位置，从而实现随机存取。

所需头文件：

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

函数原型：

```
off_t lseek(int fd, off_t offset, int whence);
```

返回值：从文件开头计算文件偏移量的值（成功）；-1（出错）

6. fcntl ()函数

功能描述： 根据文件描述符来操作文件的特性。
所需头文件：

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

函数原型：

```
int fcntl(int fd, int cmd);
int fcntl(int fd, int cmd, long arg);
```

返回值：正确返回值根据命令码而定,错误返回-1。

7. readv()
功能描述： 从多个缓冲区中读取数据。
所需头文件：

```
#include <sys/uio.h>
```

函数原型：

```
ssize_t readv (int fd, const struct iovec *iov,int count);
```

返回值：返回所读取的字节数； -1（出错）。

8. writev()
功能描述： 向多个缓冲区中写入数据。
所需头文件：

```
#include <sys/uio.h>
```

函数原型：

```
ssize_t writev (int fd, const struct iovec *iov,int count);
```

返回值：写入文件的字节数（成功）；-1（出错）。