2.2 io 接口 - read/write 等_物联网 / 嵌入式工程师 - 慕课网

> 慕课网慕课教程 2.2 io 接口 – read/write 等涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

2.Linux 文件 io 接口 – read/write/lseek

函数头文件

#include <unistd.h>

函数原型

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

函数功能

从文件中读取数据保存缓冲区中

函数参数

fd : 文件描述符

buf : 数据缓冲区

count : 能够读取的最大字节数

函数返回值

成功 : 返回实际读取的字节数

失败 : –1, 并将错误编码设置到 errno 中

示例: 从指定文件中读取 10 个字节数据，并进行打印

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <errno.h>
#include <unistd.h>

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

    int fd;
    char buffer[64]  = {0};
    ssize_t rbytes;

    if (argc != 2){
        fprintf(stderr,"Usage : < %s > < pathname >\n",argv[0]);
        return -1;
    }



    fd = open(argv[1],O_RDONLY);

    if (fd == -1){
        perror("Open(): ");
        return -1;
    }
```

```
    rbytes = read(fd,buffer,10);
    if (rbytes == -1){
        perror("Read(): ");
        return -1;
    }

    printf("Buffer : %s\n",buffer);


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

函数头文件

#include <unistd.h>

函数原型

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

函数参数

- fd : 文件描述符
- buf : 缓冲区地址
- count : 需要写入的字节数

函数返回值

- 成功: 返回实际成功写入的字节数
- 失败: 返回 –1, 并设置 errno

示例: 将 ABCDE12345 字符串写入到指定文件中, 并验证是否写入正确

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <errno.h>
#include <unistd.h>

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

    int fd;
    char buffer[64]  = "ABCED12345";
    ssize_t wbytes;

    if (argc != 2){
        fprintf(stderr,"Usage : < %s > < pathname >\n",argv[0]);
        return -1;
    }



    fd = open(argv[1],O_RDWR|O_CREAT);

    if (fd == -1){
        perror("Open(): ");
        return -1;
    }

    wbytes = write(fd,buffer,10);
    if (wbytes == -1){
        perror("Write(): ");
        return -1;
    }



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

函数头文件

#include <sys/types.h>

#include <unistd.h>

函数功能

- 成功: 返回 0
- 失败：返回 -1, 并设置 errno

函数原型

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

函数参数

- fd：文件描述符
- offset：偏移量, 可以为正数或者负数
- whence：偏移相对位置
    - SEEK_CUR：相对于文件当前偏移
    - SEEK_SET：相对于文件开始位置
    - SEEK_END：相对于文件尾偏移

函数返回值

- 成功: 返回 0
- 失败：返回 -1, 并设置 errno
- 当前文件的偏移量决定下次 io 操作时的起始位置
- 对于同一个文件描述符，共享同一个偏移量

示例: 将一个字符串 "hello,linux io" 写入到文件中，在读取出来

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <errno.h>
#include <unistd.h>
#include <string.h>

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

    int fd;
    char buffer[64]  = "hello,Linux io";
    char rbuffer[64] = {0};
    ssize_t wbytes = 0,rbytes = 0;

    if (argc != 2){
        fprintf(stderr,"Usage : < %s > < pathname >\n",argv[0]);
        return -1;
    }



    fd = open(argv[1],O_RDWR|O_CREAT);

    if (fd == -1){
        perror("Open(): ");
        return -1;
    }

    wbytes = write(fd,buffer,strlen(buffer));
    if (wbytes == -1){
        perror("Write(): ");
        return -1;
```

```
    }


    lseek(fd,0,SEEK_SET);

    rbytes = read(fd,rbuffer,wbytes);
    if (rbytes == -1){
        perror("Read(): ");
        return -1;
    }

    printf("rbuffer : %s\n",rbuffer);

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

练习：使用 Linux 文件 io 接口实现 文件复制

全文完