

文件编程

文件编程

- 1, 打开/创建文件 (open)
- 2, 读取/写入操作 (write)
- 3, 文件的读取(read)

函数学习:

问题分析:

1, 打开/创建文件 (open)

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
int main(void)
{
    int fd;

    fd = open("./file1", O_RDWR);

    if(fd == -1){

        printf("open file1 fail\n");
        fd = open("./file1", O_RDWR|O_CREAT, 0600);
        if(fd > 0){
```

```

        printf("creat file1\n");
    }

}

return 0;
}

```

touch file1 //建立一个文件

文件描述符:

0600 可读可写, 读取4 写入2 执行1

2, 读取/写入操作 (write)

```

#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>
int main(void)
{
    int fd;
    char *buf = "一颗例子";
    fd = open("./file1", O_RDWR);

    if(fd == -1){

        printf("open file1 fail\n");
        fd = open("./file1", O_RDWR|O_CREAT, 0600);
        if(fd > 0){

            printf("creat file1\n");

        }
    }
}

```

```

    }

    printf("success! fd = %d\n", fd);

    write(fd, buf, strlen(buf));

    close(fd);

    return 0;
}

```

3, 文件的读取(read)

函数学习:

```

#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int main(void)
{
    int fd;
    char *buf = "一颗例子";
    fd = open("./file1", O_RDWR);

    if(fd == -1){

        printf("open file1 fail\n");
        fd = open("./file1", O_RDWR | O_CREAT, 0600);
    }
}

```

```

        if(fd > 0){

            printf("creat file1\n");
        }

    }

    printf("success! fd = %d\n",fd);

    int n_write = write(fd,buf,strlen(buf));
    if(n_write != -1){
        printf("write %d byte to file\n",n_write);

    }
    //修改后, 补充的代码
    //close(fd);
    //fd = open("./file1",O_RDWR);

    char *readbuf;
    readbuf = (char *)malloc(sizeof(char)*n_write + 1);

    int n_read = read(fd, readbuf, n_write);

    printf("read %d,context:%s\n",n_read, readbuf);

    close(fd);

    return 0;
}

```

```

lizi@ubuntu:~/桌面/linux/1.document$ ./a.out
open file1 fail
creat file1
success! fd = 3
write 12 byte to file
read 0,context:

```

问题分析：

重点在于光标位置，其实在上面代码写完之后，光标就在最后了，这个时候有两个办法，第一个就是关闭重新读取，第二个就是改变光标位置。

这是使用的是第一种方式，关闭重新读取。

```
lizi@ubuntu:~/桌面/linux/1.document$ ./a.out  
open file1 fail  
creat file1  
success! fd = 3  
write 12 byte to file  
read 12,context:一颗例子
```