

---

# 文件I/O (二)

主讲：大海老师

# 课程目标

读取文件（熟练）

写入文件（熟练）

定位文件（熟练）

小结

# 文件I/O – read

read函数用来从文件中读取数据:

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

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

→ fp    s    n    size    判  
fd    buf    count    判

- 成功时返回实际读取的字节数；出错时返回EOF
- 读到文件末尾时返回0
- buf是接收数据的缓冲区
- count不应超过buf大小

# 文件I/O – read – 示例

从指定的文件(文本文件)中读取内容并统计大小

```
int main(int argc, char *argv[]) {  
    {  
        int fd, n, total = 0;  
        char buf[64];  
        if (argc < 2) {  
            printf("Usage : %s <file>\n", argv[0]); return -1;  
        }  
        if ((fd = open(argv[1], O_RDONLY)) < 0) {  
            perror("open"); return -1;  
        }  
        while ((n = read(fd, buf, 64)) > 0) {  
            total += n;  
        }  
        .....  
    }
```

文件

open

不间断, 接口, 加和

xf\_改 核心 辅助 例证

# 文件I/O – write

write函数用来向文件写入数据:

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

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

→ fp & n size  
fd buf count.

- 成功时返回实际写入的字节数；出错时返回EOF
- buf是发送数据的缓冲区
- count不应超过buf大小

# 文件I/O – write – 示例

将键盘输入的内容写入文件，直到输入quit

```
int fd;  
char buf[20];  
  
if ((fd = open(argv[1], O_WRONLY|O_CREAT|O_TRUNC, 0666)) < 0) {  
    perror( "open" ); return -1;  
}  
while (fgets(buf, 20, stdin) != NULL) {  
    if (strcmp(buf, "quit\n" ) == 0) break;  
    write(fd, buf, strlen(buf));  
}  
.....
```

( if argc )

open

→ gets buf

if write buf  
(while)

(re)

(close)

# 文件IO – lseek

lseek函数用来定位文件:

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

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

*fp whence offset.*

*fd whence offset.*

➤ 成功时返回当前的文件读写位置；出错时返回EOF

➤ 参数offset和参数whence同fseek完全一样

*(arg)*

lseek 例子: 定位到末尾  
加个 +



27 open  
lseek  
fwrite.  
fclose  
(re).

## 标准I/O - 小结

read

write

lseek



# 标准I/O – 思考和练习

(argc)

(文件)

利用文件IO实现文件的复制

文件名通过命令行参数指定

打开文件的方式?

如何判断读到源文件的末尾?

(说天可让ai写, 自己研究).

open fds

open fdt

→

read buf

write buf

(while)

c close)

(re)

