

C语言程序设计(第8章输入输出和文件系统)2 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/135/2021_2022_C_E8_AF_AD_E8_A8_80_E7_A8_8B_c97_135061.htm 8.2 非缓冲文件系统

前面介绍的缓冲文件系统是借助文件结构体指针来对文件进行管理，通过文件指针来对文件进行访问，既可以读写字符、字符串、格式化数据，也可以读写二进制数据。非缓冲文件系统依赖于操作系统，通过操作系统的功能对文件进行读写，是系统级的输入输出，它不设文件结构体指针，只能读写二进制文件，但效率高、速度快，由于ANSI标准不再包括非缓冲文件系统，因此建议大家最好不要选择它。本书只作简单介绍。

1. 文件的打开与关闭 非缓冲文件系统不是ANSI标准定义的，是UNIX型I/O系统的一员，所以，其原型位于io.h文件中。打开文件：`int open(char *fname,int access)`. 打开文件名为fname，以access方式访问：access的值为：`O_RDONLY`只读 `O_WRONLY`只写 `O_RDWR`读写 关闭文件：`close(int fd)`；下述程序用UNIX系统打开和关闭一个文件：`#include "io.h" #include "fcntl.h" #include "sys\stat.h" main(argc, argv) int argc. char *argv[] { int fd. if((fd=open(argv[1],O_RDONLY))==-1) /* 以只读方式打开文件*/ { printf("cantt open file!"). exit(0). } printf("file existent!"). if(close(fd)) printf("error in closing file\n"). }`

2. 文件的读写 对非缓冲文件系统的读写函数的原型在io.h头文件中，其调用形式为：`int read(int fd,void *buf,int count)` `read()`函数从fd说明的文件中读取count个字节到buf所指向的缓冲区。函数的返回值是实际读写的字节数。`int write(int fd,void *buf,int count)` `write()`

函数把count个字节从buf写入到fd说明的文件中。函数的返回值是实际写入的字节数。下面例子从文件TEST.TST中读取它的前半100个字节并放到数组buffer中。

```
#include "io.h" #include "stdio.h" #include "fcntl.h" main( ) { int fd. char buffer[100]. if ((fd=open("TEST.TST",O_RDONLY))==-1) /* 打开文件* / { printf("cannot open file !\n"). exit(0). } if(read(fd,buffer,100)!=100) /*判断读写的字节数是否正确*/ printf("Possible read error."). }
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com