# 文件描述符

## 简介

注：库函数即标准IO所在的位置。

在库函数层面表示的标准I/O用流来表示，在系统调用层面用文件描述字/符表示。不论是流还是文件描述字，表示的都是进程中与硬盘中文件的连接关系。

文件描述字：

另一种连接机制（程序与文件）

系统调用（底层基本的IO操作）

特定设备（流不行）

特殊方式进行IO（流不行）

## 操作

### 创建文件

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <fcntl.h>

int main(int argc,char \*\*argv)

{

int ret = -1;

//创建新文件b.c，权限读写执行都有

ret = creat("b.c",0777);

if( -1 == ret )

{

perror("create");

return -1;

}

return 0;

}

### 打开文件

原型：

int open(const char \*pathname, int flags);//只是执行打开

int open(const char \*pathname, int flags, mode\_t mode);//还可以创建

数据类型（返回值）：int（范围有限制）

预先打开的文件描述字：0,1,2

示例（打开文件）：

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <fcntl.h>

int main(int argc,char \*\*argv)

{

int ret = -1;

//打开文件b.c

ret = open("b.c",O\_RDWR);//流 r+

//ret = open("b.c",O\_RDWR|O\_TRUNC);//流 w+

//ret = open("b.c",O\_RDWR|O\_APPEND);//流 a+

if( -1 == ret )

{

perror("create");

return -1;

}

close(ret);

return 0;

}

示例（打开并创建）：

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <fcntl.h>

int main(int argc,char \*\*argv)

{

int ret = -1;

//创建新文件b.c，权限读写执行都有,并且打开文件

ret = open("b.c",O\_RDWR|O\_CREAT,0777);

if( -1 == ret )

{

perror("create");

return -1;

}

close(ret);

return 0;

}

示例（文件存在正常打开，不存在则创建）：

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <fcntl.h>

#include <errno.h>

int main(int argc,char \*\*argv)

{

int ret = -1;

//文件存在打开,不存在就创建打开

//创建新文件b.c，权限读写执行都有,并且打开文件

ret = open("b.c",O\_RDWR|O\_CREAT|O\_EXCL,0777);

if( -1 == ret )

{

if ( EEXIST == errno )//是否存在

{

ret = open("b.c",O\_RDWR);

if ( -1 == ret )

{

perror("open");

return -1;

}

}

return -1;

}

close(ret);

return 0;

}

### 关闭文件

### 读写

### 定位

### 重复

### 与流互换

### 文件控制

### 缓冲

# 非阻塞IO