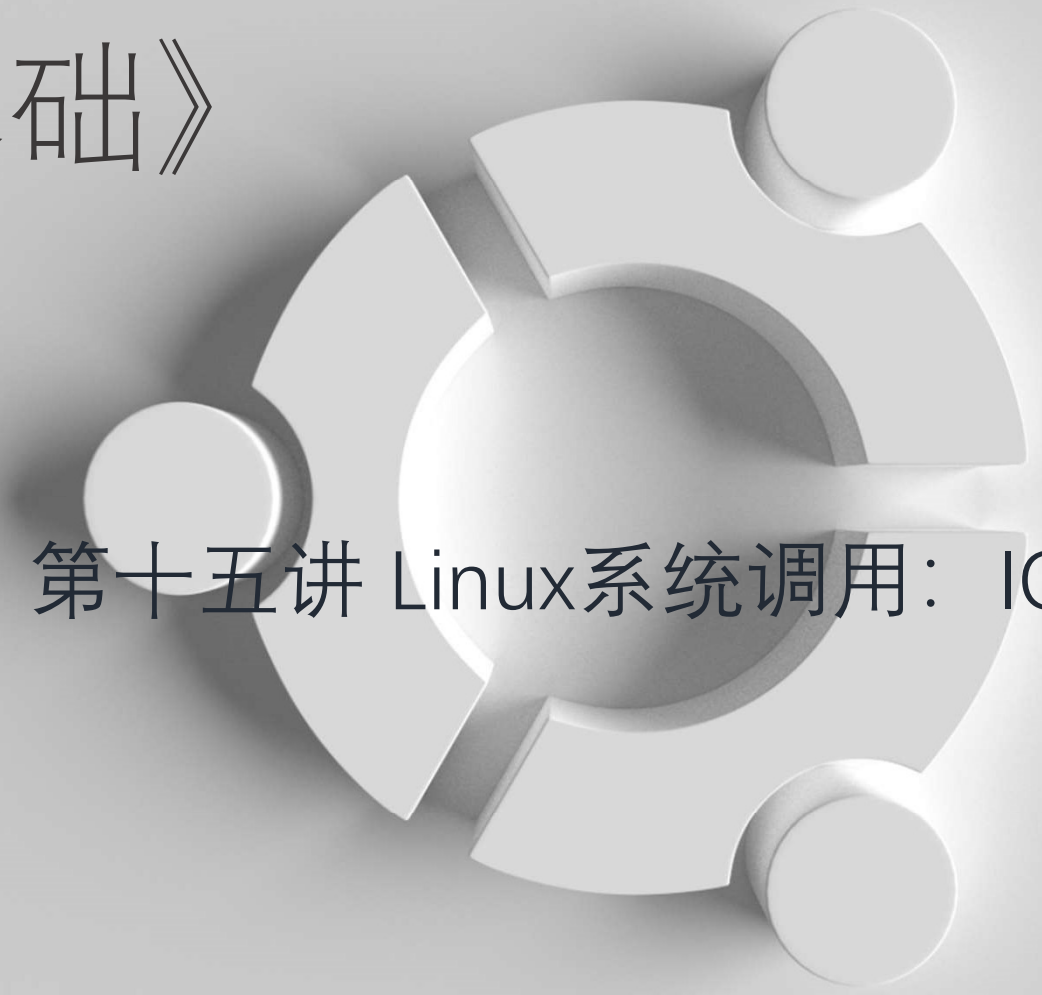
河北师范大学 软件学院
Software College of Hebei Normal University

《Linux基础》



第十五讲 Linux系统调用：IO

open函数

- open函数用于打开文件操作:

  int open(const char *pathname, int flags, mode_t mode); 参数依次为文件路径名称，标志位，模式。成功返回值为打开的文件描述符，错误返回-1。

  flags选项:

O_CREAT	没有则创建文件
O_WRONLY	写模式打开文件
O_RDONLY	只读模式打开文件
O_RDWR	读写方式打开

  mode选项:

S_IRWXU	文件所有者具有可读，可写，可执行的权限
S_IRUSR	文件所有者有可读权限
S_IWUSR	文件所有者具有可写权限

close函数

- 在文件操作最后要记得使用close关闭打开的文件：int close(int fd);
- close函数成功返回0，错误返回-1。

open函数示例

- 以下代码会打开名为buffer的文件，如果文件不存在则创建：

```
int main(int argc, char * argv[])
    int fd = open("buffer", O_CREAT, S_IRUSR|S_IWUSR|S_IRGRP);
    if (fd < 0) {
        perror("open");
        return -1;
    }
    close(fd);
    return 0;
}
```

read函数

- read函数从一个文件读取数据:

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

  参数依次为打开的文件描述符，指向数据的指针，要读取的字节数。

  返回值是成功读取的字节数，如果有错误则返回-1。

read函数示例

- 以下代码从buffer文件读取一些数据并放到变量buf中：

```
int fd = open("buffer", O_RDONLY);
//此处省略错误处理
char buf[1024] = {'\0'};
int count = 0;
count = read(fd, buf, 1000);
if (count<0) {
    perror("read");
} else {
    printf("%s\n",buf);
}
close(fd);
```

write函数

- write函数向一个文件写入数据:

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

  参数依次为打开的文件描述符，指向数据的指针，要写入的字节数。

  返回值是成功写入的字节数，错误则返回-1。

write函数示例

- 以下代码会向buffer文件写入一条数据:

```
char*buf = “Linux\nUnix\nC”;
int count = 0;
count = write(fd, buf, strlen(buf));
if (count<0)
    perror(“write”);
else
    printf(“bytes:%d\n”, count);
```

- 文件打开标志位: O_TRUNC会截断文件长度为0

```
int fd=open(“buffer”,O_RDWR|O_TRUNC, S_IWUSR|S_IRUSR);
```

如果buffer是普通文件，并且对当前用户可写，这种方式打开文件会导致buffer文件的内容被清空（文件长度被截断为0）。（O_WRONLY或O_RDWR配合O_TRUNC会导致这种效果）

IO重定向

- IO重定向基于这样一个设计原则：最低可用文件描述符（Lowest Available fd）原则。
- 文件描述符是一个数组索引号，每个进程都有一组打开的文件，这些打开的文件信息保存在一个数组中，文件描述符就是数组的索引号。
- 在打开文件时，分配的描述符总是数组中最低可用的索引位置（索引数字最小的位置）。
- 在Linux上，使用0，1，2作为程序的标准输入，标准输出，标准错误输出。而如果关闭描述符1，然后打开其他文件，这样文件就被分配了文件描述符1，于是标准输出就会写入到新打开的文件。这就是IO重定向。

IO重定向实现方式

- 编程实现的方式（以文件描述符1为例）：
  - close-open-close方式，先close(1)，然后open(filename, O_RDWR, S_IWUSR)；操作完成，close关闭新打开的文件。
  - open-close-dup-close方式，open打开文件，返回的文件描述符不是1，然后close(1)，现在最低可用描述符是1，dup(fd)会把新打开的描述符复制到1，然后close(fd)关闭新打开的文件。
  - open-dup2-close方式，dup2(oldfd, newfd)，关闭newfd，把oldfd复制到newfd，close关闭新打开的描述符。

IO重定向示例

- 以下示例代码printf不会在屏幕输出，而是输出到一个文件中：

```
int fd=open("riotest",O_CREAT|O_APPEND|O_RDWR,S_IRUSR|S_IWUSR);
if (fd<0) {
    perror("open");
    return -1;
}
dup2(fd, 1);
close(fd);
printf("PHP is best\n");
```

练习

- 编写一个简单的复制命令：接受两个参数，第一个参数作为要复制的文件，第二个参数是要复制的文件名。