1. 文件描述符

先解释输入输出重定向

文件描述符 0 1 2

0标准输入 默认接受来自键盘的输入 stdin in

1标准输出 默认输出到屏幕终端 stdout out

2错误输出 默认输出到屏幕终端 stderr err

其实1其实我理解了一点，将交互时的内容输出到文件

eg: echo “a b” 1>a.txt

这样就实现了标准输出，将内容输出到文本

或者

ls –al > a.txt

是一种将屏幕内容输出到文本的操作

但是0，我几乎没用过

因为这是将文本输出到屏幕。

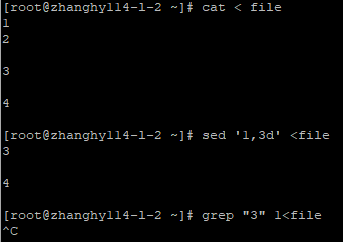
那有时候可以脑子就不好使了，输出屏幕用cat file 不久行了嘛？

cat file 何必写成 cat 1< file呢。

其实举一反三，输出可以将任何输出到屏幕内容哪怕时命令内容都输出到文件

那么输入可不可以将文件内容全部输入到屏幕再给拿出来使用呢。

如果将屏幕内容使用，这个问题就有点难度了。过滤可以吗？

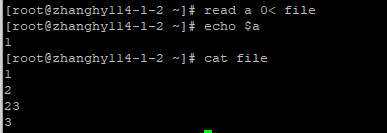


好像sed可以，但也是有点多余了，因为可以直接sed “1,3d” file

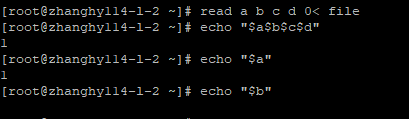
但是有一个可以，交互式的时候需要我们输入内容到屏幕啊。

比如read –p “nihao” a

这时候需要将内容输入到屏幕。但我们可以直接将文本内容赋值给变量a



但是只会赋值文本第一行内容给变量a



如何将文本内容（假如5行）依次赋值给一个变量呢？

read a b c d e < file （这个是错误的，因为read没有这个命令）

肯定用到循环了

while read i

do

done<file

键盘----标准输入stdin----process进程------------stdont标准输出到文本

错误输出到屏幕stdeer当然也可以输出到文本

文件描述符就是每打开一个文件都会有一个与之对应的描述符。Linux一切都是文件

其中0就是输入打开啊

1就是输出文件

2就是错误输出

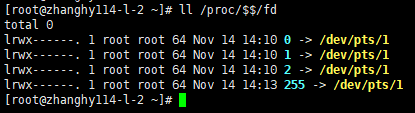
3就是其他文件了

如何查看一个进程打开那些描述符？

ll /proc/进程id/fd

$$当前shell的进程ID，如果在子shell中打印这个结果就是子shell的进程号。

但是在shell中这就是shell的进程号。因为我是用xshell打开的，那么就是



这是什么意思呢？



其实就是四个文件对应了我们打开的文件

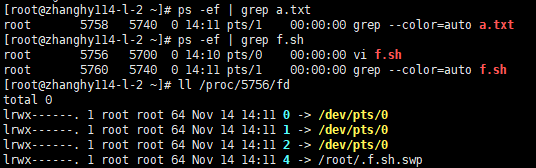
这表示我打开的终端号文件描述符



测试其他文件描述符！

打开一个终端，并且打开一个文件f.sh

再打开一个终端，查看你文件进程，并查看文件标识符

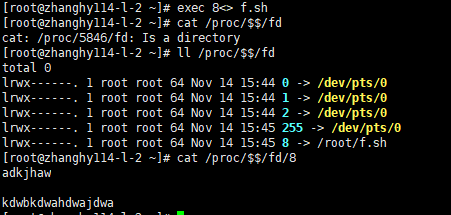


文件标识符内容和文件内容是一样的。但是不知道为什么我的文件标识符打开时错码？

会不会是用不同终端导致的结果。一个终端修改了文件描述符，打开这个文件的确实另一个终端的缘故？

所以接下里我在一个终端打开文件。

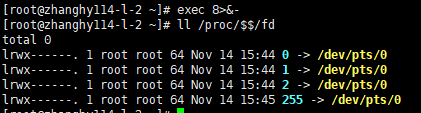
用上exec 8<> file 这个命令是将file打开而且将8作为他的文件描述符



我们想删除这个8文件软连接可以吗？

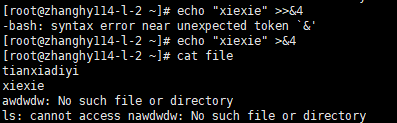


不可以，怎么办？



好了，学会了文件描述符。我发现在一个终端下面，操作文件描述符（肯定必须在一个终端下，因为你这个文件进程在$$下面）

那接下来就简单了，操作软连接文件，也就是描述符，看文件内容是否修改



果然修改了

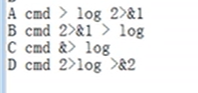
输出

ls 文件 >/root.log 2>&1

将2的输出到1，这样不管错误还是正确都输入到log文件下面

那么我想输出到描述符4上面呢

ls 文件 > &4



上面的那个是与其他三个不同。

B其中第四个是将标准输出到错误输出，然后错误输出到log文件

那么文件描述符是否可以无限打开呢，就是说在/dev/pts这个终端进程下，可以有多少个文件被打开？

ulimit –n

1024

这也就是说，可以有1024个被打开

当然可以设置 ulimit –n 2048

1. exec命令

xargs 和 –exec有点像，将之前命令的输出作为后一个命令的输入

ls –al | xargs rm –rf

ls –al –exec rm -rf

用上exec 8<> file 这个命令是将file打开而且将8作为他的文件描述符

ls 文件 > &4

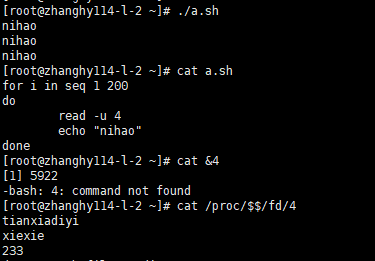
就是将内容输出到4被描述符，当然，内容也会随着修改

删除描述符

exec >&4-

1. read –u 文件描述符

这是读取文件描述符，当文件描述符内容有几行，就会读取几次，读取完毕就会停止



1. 管道

特征1.只能用一次，也就是说内容放进去，拿出来，管道里面就空了

2.先进先出，先放进去内容，出来的总是先进去的

匿名管道：

类似| 管道命令符 ，其实这也是文件

ls | grep “a.sh”

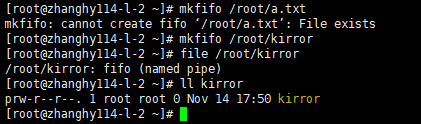
将ls的内容放进管道文件中 其实可以看做 grep “a.sh” | 这个|里面有了ls的内容，用完给grep后，就失去作用了，所以|可以一直使用

实名管道：

可以将一个文件作为管道

mkfifo + 文件路径（但是这个文件不能存在）

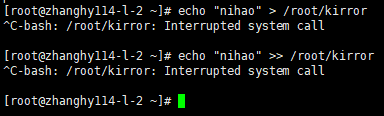
然后file + 文件路径就可以查看状态了



named pipe命令管道

可以看见我创建成功了

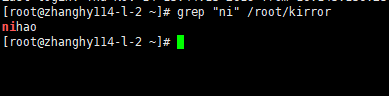
那么接下来接下来就是运用



我尝试将内容写入管道，但是失败了，所以我猜想，应该是写入无法退出状态

，应该就是在管道里面





一个终端输入管道内容，一个终端运行