

《Linux基础》



第四讲 IO重定向与管道

标准输入输出

- 每个进程都至少有3个信道：标准输入-STDIN、标准输出-STDOUT、标准错误-STDERR。

标准输入：文件描述符为0，默认指向终端（在虚拟终端中即为键盘）

标准输出：文件描述符为1，默认指向终端（在虚拟终端中即为屏幕）

标准错误：文件描述符为2，默认指向终端（在虚拟终端中即为屏幕）

在Linux中一切皆文件，所以标准输入、标准输出、标准错误都有对应的文件，而这种对应关系是通过文件描述符来实现的，我们在重定向的时候还会用到它们。

- 对于一个进程来说，它从描述符0指向的文件读取输入信息，把执行结果送到描述符1指向的文件，把出错信息送到描述符2指向的文件，而不关心这些文件具体是什么。
- 关于文件描述符：一个进程有一个结构保存一组打开的文件，每个打开的文件用一个数字标识，此数字一般被称为文件描述符，而在Linux/Unix上设备也被映射为文件，数字0，1，2都会分别关联到键盘，屏幕，屏幕。

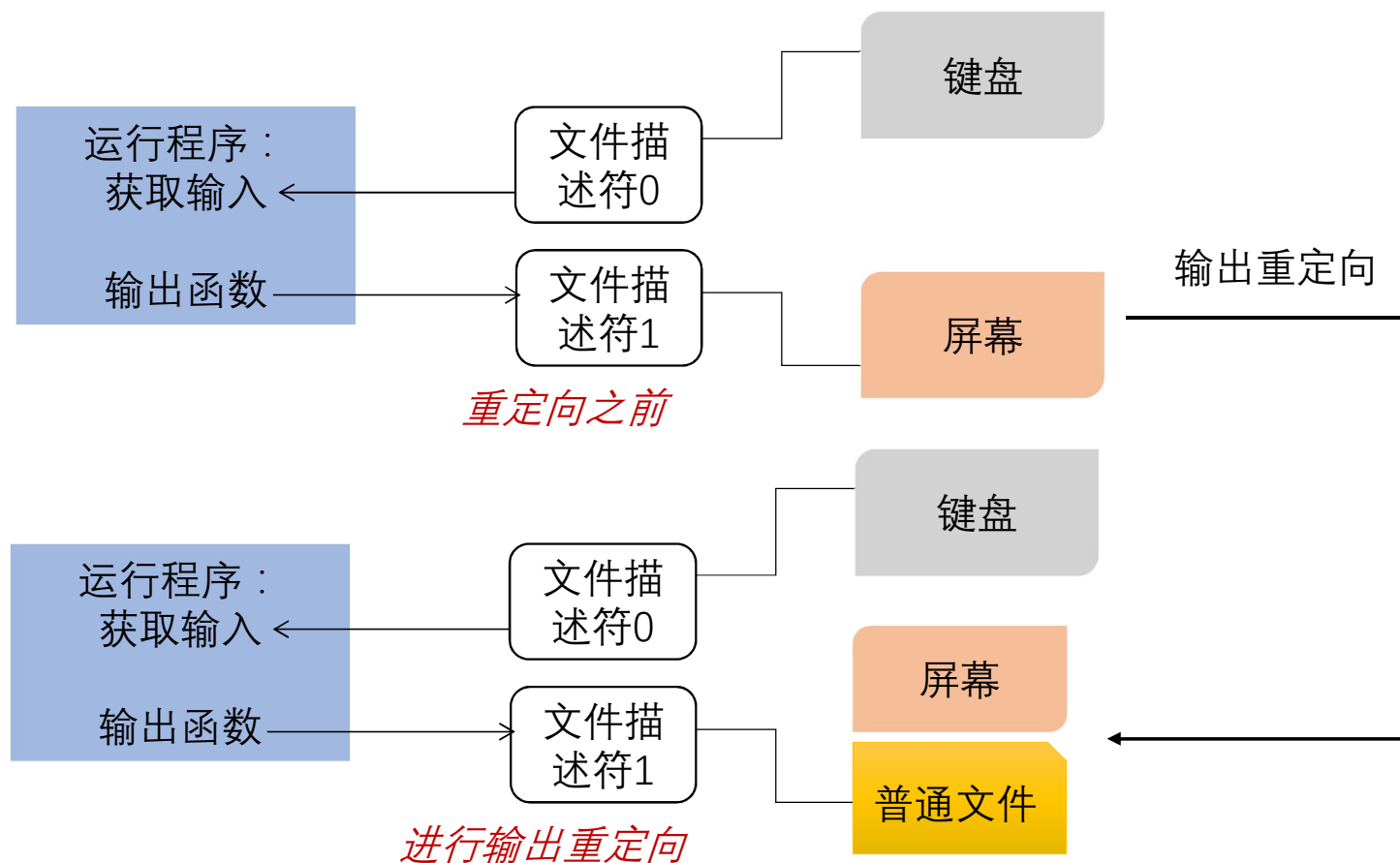
IO重定向

- 在shell中输入命令运行程序，程序的正常输出信息（标准输出）和一些出错信息（标准错误）会通过shell显示在屏幕上。
- 有时候我们并不需要把这些输出信息（包括标准输出和标准错误）显示在屏幕上，或需要把这些输出信息保存在一个文件中，这时就需要进行输出重定向。
- 如果一个命令要获取的输入来自文件，或者是在shell脚本编程中，需要从文件获取输入，则需要输入重定向。

重定向示例

- `echo 'abc'` 会输出abc到屏幕。`echo 'abc' > tmp/buff`会把abc输出到tmp/buff这个文件。
- 如果没有此文件则会创建这个文件并写入。但是如果文件存在并且不为空，则重定向会导致之前的数据丢失，只保存重定向的数据。
- `echo 'abc' >> /tmp/buff` 会把abc追加到文件末尾，之前的数据不会丢失。

重定向图解



输出重定向后的结果：输出的字符保存到文件，文件描述符1断开和显示设备的关联，关联到了一个普通文件。

谁控制重定向

- 在shell中使用 $>$ ， $<$ ， $>>$ 等重定向符号时，执行重定向操作的是shell，而不是程序。
- shell把重定向符号解释成指令，将标准输入、输出、错误输出指向其他文件，而不是所关联的设备。
- 手动编写的程序也是可以进行重定向的，但是在shell中输入命令，shell会创建子进程去执行命令，此时进行重定向操作的是shell。

重定向符号

- shell将<、>、>>解释成指令，用来把一条命令的输入或输出重定向到一个文件。重定向STDERR的话，则用2>。

类型	操作符	用途
重定向标准输入	<	将命令中接收输入的途径由默认的键盘更改为指定的文件
重定向标准输出	>	以替换的方式将命令的执行结果输出到指定的文件，而不是直接显示在屏幕上
	>>	将命令执行的结果追加输出到指定文件
重定向标准错误	2>	清空指定文件的内容，并将标准错误信息保存到该文件中
	2>>	将标准错误信息追加输出到指定的文件中
重定向标准输出和标准错误	&>或>&	将标准输出、标准错误的内容全部保存到指定的文件中，而不是直接显示在屏幕上

重定向示例

- `wc -l newfile` 与 `wc -l < newfile`

`wc`是一个计算行数的程序，第一个是对文件进行操作；第二个是对文件内容进行操作，把文件的内容作为输入。

- `find / -iname gcc* > find_tmp`

标准输出重定向到`find_tmp`文件，但是错误信息会显示在屏幕上，如果需要把错误信息也重定向到文件，可以运行：`find / -iname gcc* &> find_tmp`

- `1>` 等同于`>`，`0<` 等同于`<`，即前导的0和1可以省略；如果不省略，数字和重定向符号之间没有空格，前导2不可以省略。

管道

- 把一条命令的STDOUT连接到另一条命令的STDIN上，可以用 | 这个符号，在shell中表示管道。
- shell在解释命令遇到 | 时会创建管道，并创建两个进程，把标准输入输出重定向到管道，前一个进程向管道写数据，后一个进程从管道读数据。
- 在管道中只有标准输出才传递给下一个命令 标准错误输出直接输出到终端。

管道和重定向示例

- 查找名称含有curl的文件进行计数，并把错误信息重定向到/dev/null

```
find / -iname *gcc* 2> /dev/null | wc -l
```

- 查找名称含有ssh的进程

```
ps -ef | grep ssh
```

- 分页查看内容

```
ls -l -R /usr/share | less
```

- 排序文件

```
ls | sort -r
```

- 分页查看帮助文档/文件

```
help test | less
```

```
cat /etc/passwd | less
```