Linux 基础

第 8 讲 IO 重定向和管道

问题

- 运行 `find /usr -iname *gnome*` 会在 `/usr` 目录搜索名称含有 `gnome` 的文件。
- 结果会输出到终端，并且错误信息也会一并输出。
- 如何保存搜索结果到一个文件?

IO 重定向

- 之前的问题可以使用 IO 重定向解决。
- IO 重定向就是更改输入输出数据的流向。
- 这在 shell 中，通过命令操作，形式上体现出来的仅仅是一个符号。

示例

- 保存结果到文件：

```
find /usr -iname *gnome* > findtmp
```

- 在 shell 中，> 表示输出重定向。

理解 IO 重定向

- 一个进程，在运行时，通过一个结构体数组关联打开的文件。
- 而作为数组下标的数字在 Linux/Unix 上被称为文件描述符。

理解 IO 重定向

- 文件描述符 0,1,2 分别表示:
  - 标准输入（stdin）
  - 标准输出（stdout）
  - 标准错误输出（stderr）

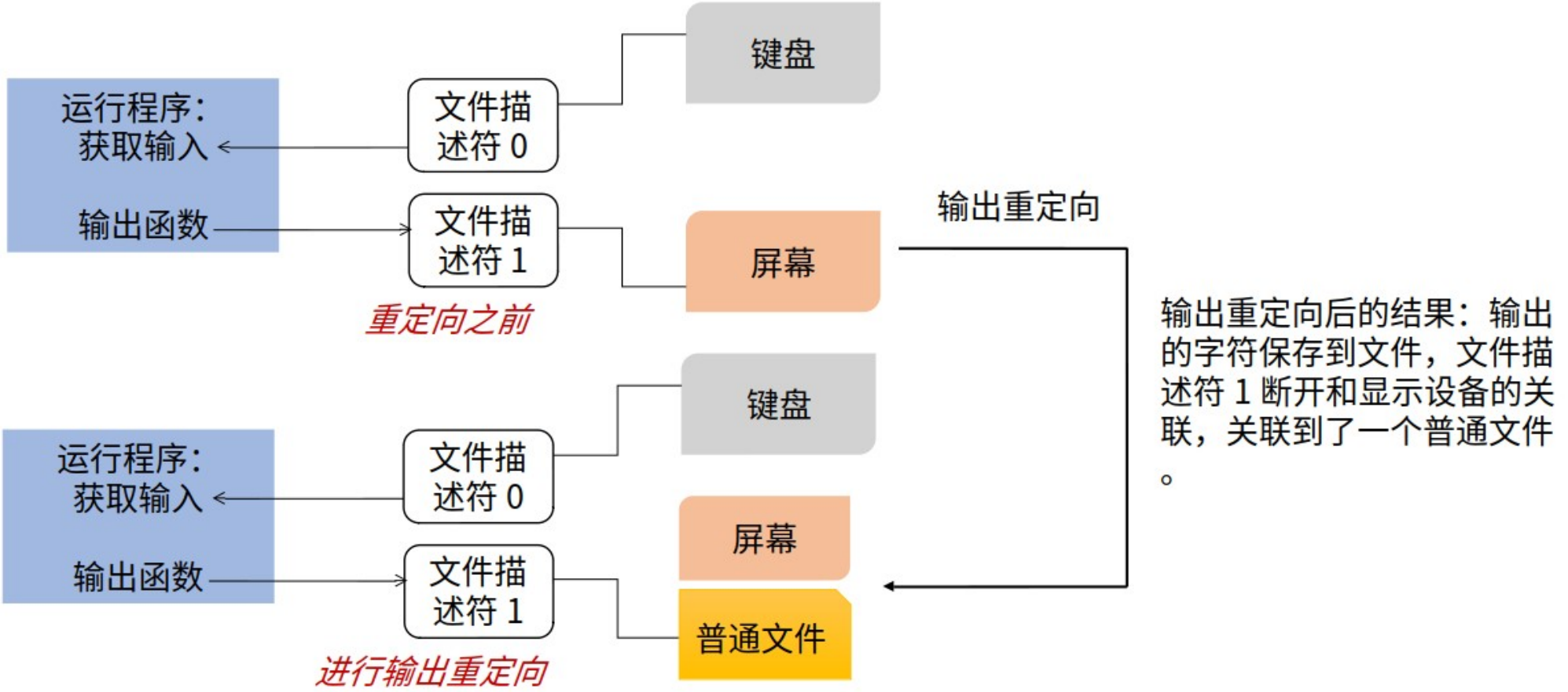
理解 IO 重定向

- 默认情况下，文件描述符 0 关联输入设备的设备文件，在 PC 上，为键盘对应的设备文件。
- 默认文件描述符 1,2 都关联输出设备的设备文件，在 PC 上，为当前显示设备对应的设备文件，可简单理解为向屏幕输出。

理解 IO 重定向

- 以输出重定向为例，如果程序运行时，文件描述符 1 不再关联输出设备的设备文件，而是关联到一个其他的文件。
- 比如一个普通的文件，则输出结果就写入到文件，而不是输出到屏幕，这就是输出重定向。

重定向图解


键盘
运行程序：
获取输入
文件描
述符 0
输出函数
文件描
述符 1
屏幕
输出重定向
重定向之前
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。
键盘
运行程序：
获取输入
文件描
述符 0
屏幕
输出函数
文件描
述符 1
普通文件
进行输出重定向

shell 与重定向

- IO 重定向是系统底层提供的功能，在编程时可以实现程序的重定向。
- 在 shell 中，控制 IO 重定向只需要通过以下几个符号：

  `>` `>>` `<` `2>` `2>>` `&>` `&>>`

shell 重定向符号说明

符号	作用	说明
> >>	输出重定向	重定向到指定文件，文件不存在则会创建，如果文件存在， > 会导致之前的内容丢失， >> 则会追加到文件末尾。
<	输入重定向	从指定文件获取输入，而不是通过键盘。
2> 2>>	错误输出重定向	错误输出重定向到文件，其他参考输出重定向。
&> &>>	输出和错误输出重定向	标准输出和标准错误输出重定向。

IO 重定向示例

- `wc -l n.c` 或 `wc -l < n.c`
  - 统计文件的行数，第一种情况是读取文件统计，第二种是获取输入并统计，但是输入重定向到 `n.c`。

- `find / -iname gcc* > find_tmp`
  - 全盘搜索名称为 `gcc` 开头的文件，把结果保存到 `find_tmp`，但是错误信息会输出到屏幕。

重定向示例

- `grep 'runlevel' /etc -R > greptmp 2> /dev/null`
  - 在 /etc 目录递归搜索文件中的 runlevel，把结果保存到 greptmp，错误信息重定向到 /dev/null。
- /dev/null 是一个特殊的设备文件，称为空设备文件，是系统提供的一个功能。对此文件写入的数据会被立即丢弃，读取会立即返回 EOF。所以也被称为黑洞文件。

谁控制重定向

- 在 shell 中通过 > < 等进行重定向操作，实际是 shell 控制程序的重定向。
- shell 在解析命令字符串之后，遇到重定向符号，会对要运行命令的进程设置重定向。

管道

- 管道用于连接一个程序的输出和另一个程序的输入。
- 通过管道可以把多个命令组合在一起完成复杂的功能。

shell 中的管道

- 在 shell 中使用管道只需要符号： |
- shell 解析命令字符串，遇到 | 会创建管道。
- 并且把前一个程序的输出重定向到管道，后一个程序的输入重定向到管道。

管道示例

- `cat a.c | less`
  - 获取 `a.c` 文件的内容并使用 `less` 分页查看。
- `find / -iname *gnome* | wc -l`
  - 全盘搜索名称含有 `gnome` 的文件并通过 `wc -l` 统计结果。

管道与 IO 重定向

- shell 在使用管道连接进程时，就是采用重定向到管道的方式。
- 并且，只有标准输出才会重定向到管道，标准错误输出不会。

其他说明

- grep 是正则表达式匹配命令，可以搜索文件内容，也可以从用户的输入进行匹配，经常通过管道和其他命令组合使用。
- 管道可以连续使用，组合多个命令：

```
ps -aux | grep '^root.*nginx.*master' | grep -v
grep | awk -F' ' '{printf $2}'
```