

第2天-进程/管道/重定向

一、进程概念讲解

进程（Process）是计算机中的程序关于某数据集合上的一次运行活动，是系统进行资源分配和调度的基本单位，是操作系统结构的基础。在早期面向进程设计的计算机结构中，进程是程序的基本执行实体；在当代面向线程设计的计算机结构中，进程是线程的容器。程序是指令、数据及其组织形式的描述，进程是程序的实体。

二、查看进程

ps

查看当前终端上运行的进程

```
[root@qfedu.com ~]#ps
```

```
[root@qfedu.com ~]#ps aux
```

```
[root@qfedu.com ~]#ps auxf
```

a 只能查看所有终端进程

u 显示进程所有者

x 显示系统内所有进程

f 显示进程之间的父子关系(使用pstree查看更详细的父子关系)

user 进程所有者

pid process identify

TTY 进程在哪个终端运行

查看tty的方法：

```
[root@qfedu.com ~]#tty
```

? 表示这个进程开启的时候没有占用终端

time 进程占用cpu的总时间

cmd 进程名称

%cpu 进程占用的cpu百分比

%mem 进程占用memory百分比

VSZ 进程占用的虚拟内存大小

RSS 占用的物理内存大小

STAT 当前进程状态

按指定字段排序：

```
[root@qfedu.com ~]# ps aux --sort %cpu | less
```

```
[root@qfedu.com ~]# ps aux --sort %-cpu | less
```

显示指定字段：

```
[root@qfedu.com ~]# ps axo user,%cpu,command --sort -%cpu | less
```

```
[root@qfedu.com ~]# ps -eo user,%cpu,%mem,command --sort %cpu
```

```
[root@qfedu.com ~]# ps -elf
-e 显示所有进程
-l 长格式显示
-f 完整格式
[root@qfedu.com ~]# ps -e 常用
```

pstree

查看进程树

```
[root@qfedu.com ~]# pstree
-a 显示参数
-p 显示pid
-u 显示用户名
```

lsof

应用范围有限，只能查看带端口的进程)

```
[root@qfedu.com ~]# lsof -i:80
查看端口为80的进程
```

pidof

```
[root@qfedu.com ~]# pidof 进程名称
```

pgrep

```
[root@qfedu.com ~]# pgrep -l eyes
1179 xeyes
```

进程pid文件

```
[root@qfedu.com ~]# cd /var/run
[root@qfedu.com ~]# cat /var/run/httpd/httpd.pid
12265
```

w

```
[root@qfedu.com ~]# w
 16:51:43 up  8:30,  7 users,  load average: 0.11, 0.27, 0.30
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT
root      :0        :0             08:22    ?xdm?  29:39  0.23s
gdm-session-
root      pts/0    :0             14:13    1:43   0.30s  0.30s bash
root      pts/1    :0             11:19    5:19   0.62s  0.62s bash
root      pts/2    :0             14:13    1:39m  0.06s  0.01s less
root      pts/3    :0             14:54    7:00s  0.06s  0.01s w
root      pts/4    :0             16:25    25:51  0.06s  0.06s bash
root      pts/5    :0             16:40    7:11   0.24s  0.24s bash
```

who

```
[root@qfedu.com ~]# who
root      :0                2017-11-27 08:22 (:0)
root      pts/0              2017-11-27 14:13 (:0)
root      pts/1              2017-11-27 11:19 (:0)
```

```
root    pts/2      2017-11-27 14:13 (:0)
root    pts/3      2017-11-27 14:54 (:0)
root    pts/4      2017-11-27 16:25 (:0)
root    pts/5      2017-11-27 16:40 (:0)
```

users

```
[root@qfedu.com ~]# users
```

```
root root root root root root root
```

top

实时的查看进程的状态

h|? 帮助

> 往下翻页

< 往上翻页

M 按内存排序

P 按cpu排序

k 输入pid杀死进程

h 打印帮助

q 退出

```
[root@qfedu.com ~]# top -d 2 -p 1068 -bn1
```

```
[root@qfedu.com ~]# top -u apache
```

三、进程状态/生命周期

```
[root@qfedu.com ~]# man ps
```

R running

S sleep

T stop

Z zombie(僵死, 僵尸)

D Uninterruptible sleep (usually IO)

R Running or runnable (on run queue)

S Interruptible sleep (waiting for an event to complete)

T Stopped, either by a job control signal or because it is being traced.

W paging (not valid since the 2.6.xx kernel)

X dead (should never be seen)

Z Defunct ("zombie") process, terminated but not reaped by its parent.

< high-priority (not nice to other users)

N low-priority (nice to other users)

L has pages locked into memory (for real-time and custom IO)

s is a session leader

l is multi-threaded (using CLONE_THREAD, like NPTL pthreads do)

+ is in the foreground process group

四、进程控制

```
[root@qfedu.com ~]# kill 信号(signal) pid
-1   HUP  重新加载进程或者重新加载配置文件
-9   KILL 强制杀死
-15  TERM 正常杀死(这个信号可以默认不写)
-18  CONT 激活进程
-19  STOP 挂起进程
[root@qfedu.com ~]# kill -HUP pid
[root@qfedu.com ~]# kill -STOP pid
```

```
[root@qfedu.com ~]# kill -l
[root@qfedu.com ~]# man 7 signal 所有信号的解释

[root@qfedu.com ~]# killall 信号 进程名称

[root@qfedu.com ~]# pkill -9 进程名称
[root@qfedu.com ~]# pkill -t 终端
不加-9只杀死在终端上运行的进程，加-9连终端本身一起干掉
[root@qfedu.com ~]# pkill -u 用户名称
```

五、进程管理实战

使用以下命令对Linux系统进程进行管理操作

```
ps
pstree(-p)
top
htop
atop
ss
pgrep
Nethogs
kill
killall
pkill
```

特例

```
[root@qfedu.com ~]# ps aux | grep cher[r]
```

进程启动用户一定不要是root比如Redis要用普通用户启动

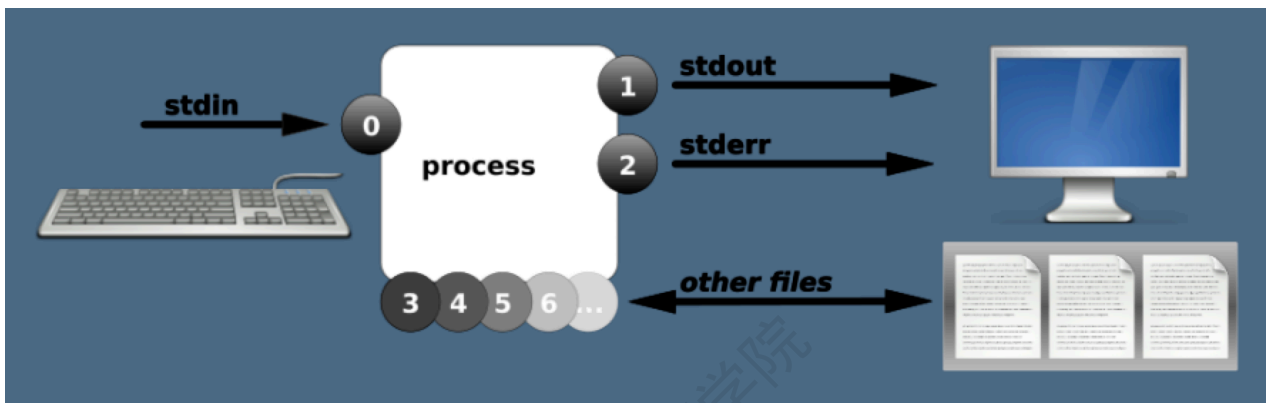
六、重定向工作原理

标准输入

标准输出

标准正确输出

标准错误输出



七、文件描述符

File Descriptors 简称fd 或 Process I/O channels

进程使用文件描述符来管理打开的文件

```
[root@qfedu.com ~]# ls /proc/$$/fd
0 1 2 3 4
0, 1, and 2, known as standard input, standard output, and standard error
```

使用lsof查看某个进程所使用的所有文件

```
[root@wing fd]# lsof -p 3626
```

八、输出重定向

正确输出: 1> 1>> 等价于 > >>

错误输出: 2> 2>>

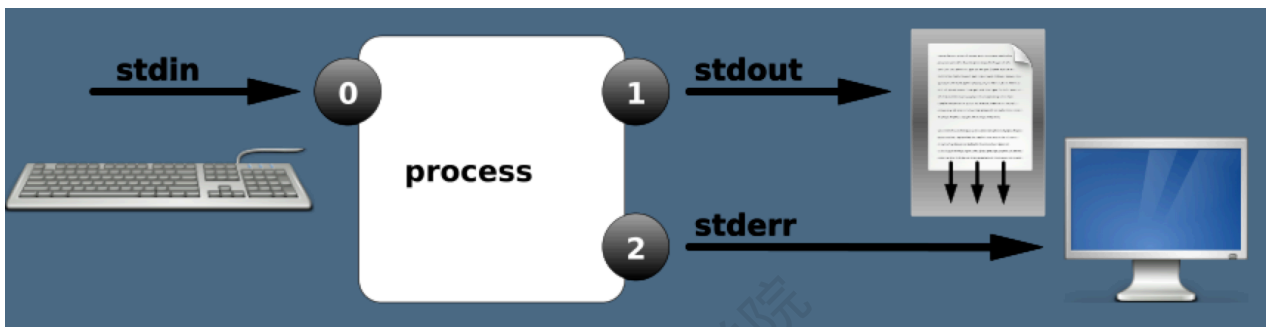
案例1: 输出重定向 (覆盖)

```
[root@qfedu.com ~]# date 1> date.txt
```



案例2：输出重定向（追加）

```
[root@qfedu.com ~]# date >> date.txt
```



案例3：错误输出重定向

```
[root@qfedu.com ~]# ls /home/ /aaaaaaaa >list.txt
```

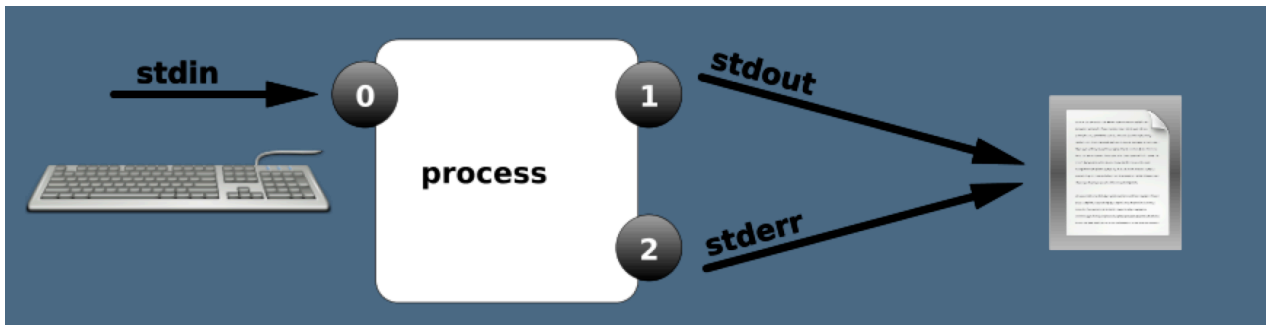
ls: 无法访问/aaaaaaaa: 没有那个文件或目录

```
[root@qfedu.com ~]# ls /home/ /aaaaaaaa >list.txt 2>error.txt //重定向到不同的位置
```



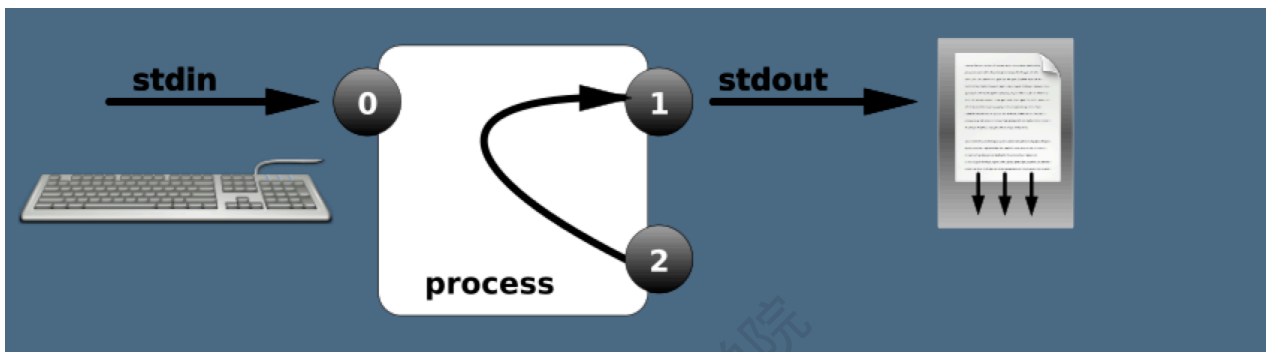
案例4：正确和错误都输入到相同位置

```
[root@qfedu.com ~]# ls /home/ /aaaaaaaa &>list.txt //混合输出
```



案例5：正确和错误都输入到相同位置

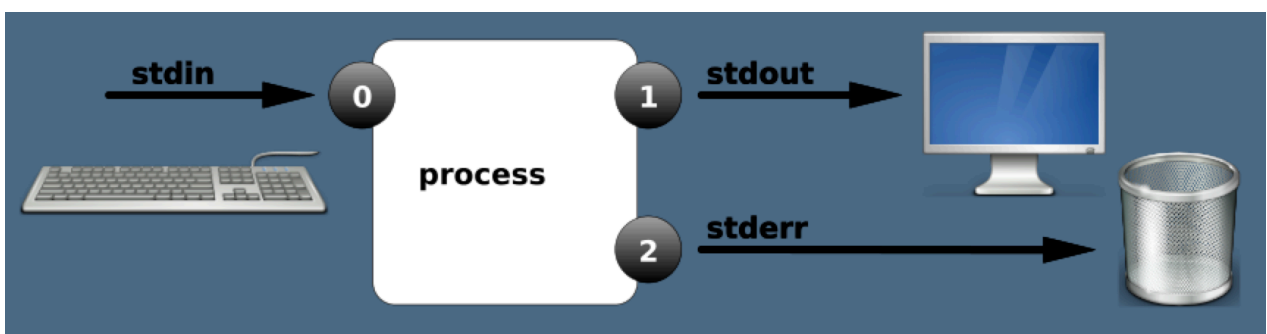
```
[root@qfedu.com ~]# ls /home/ /aaaaaaaa >list.txt 2>&1 //重定向到相同的位置
```



案例6：重定向到空设备/dev/null

```
[root@qfedu.com ~]# ls /home/ /aaaaaaaa >list.txt 2>/dev/null //空设备，即将产生的输出丢掉
```

```
[root@qfedu.com ~]# ls /home/ /aaaaaaaa &>/dev/null //空设备，即将产生的输出丢掉
```



九、输入重定向

标准输入：

< 等价 0<

案例1：

```
[root@qfedu.com ~]# mail alice //没有改变输入的方向，默认键盘
```

Subject: hello

```

1111
2222
3333
.
EOT

[root@qfedu.com ~]# su - alice
[alice@wing ~]$ mail
Heirloom Mail version 12.5 7/5/10. Type ? for help.
"/var/spool/mail/alice": 1 message 1 new
>N 1 root Mon Jul 31 15:16 20/617 "hello"

[root@qfedu.com ~]# mail -s "test01" alice < /etc/hosts //输入重定向, 来自于文件

案例2:
[root@qfedu.com ~]# grep 'root' //没有改变输入的方向, 默认键盘, 此时等待输入...
yang sss
sssrootssss..
sssrootssss..

[root@qfedu.com ~]# grep 'root' < /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin

案例3:
[root@qfedu.com ~]# dd if=/dev/zero of=/file1.txt bs=1M count=2
[root@qfedu.com ~]# dd </dev/zero >/file2.txt bs=1M count=20

案例4: mysql表结构导入
[root@qfedu.com ~]# mysql -uroot -p123 < bbs.sql

案例5: at
[root@qfedu.com ~]# at now +5 min
at> useradd yang99
at> <EOT>
job 1 at Mon Jul 31 15:29:00 2017

[root@qfedu.com ~]# vim at.txt
sudo useradd yang100
sudo useradd yang102
[root@qfedu.com ~]# at now +2 min <a.txt
job 2 at Mon Jul 31 15:27:00 2017

```

十、重定向命令

tee命令

Linux tee命令用于读取标准输入的数据, 并将其内容输出成文件。

tee指令会从标准输入设备读取数据，将其内容输出到标准输出设备，同时保存成文件。

语法

```
[root@qfedu.com ~]# tee [-ai][--help][--version][文件...]
```

参数：

- -a或--append 附加到既有文件的后面，而非覆盖它。
- -i或--ignore-interrupts 忽略中断信号。
- --help 在线帮助。
- --version 显示版本信息。

实例

使用指令"tee"将用户输入的数据同时保存到文件"file1"和"file2"中，输入如下命令：

```
[root@qfedu.com ~]# tee file1 file2 #在两个文件中复制内容
```

以上命令执行后，将提示用户输入需要保存到文件的数据，如下所示：

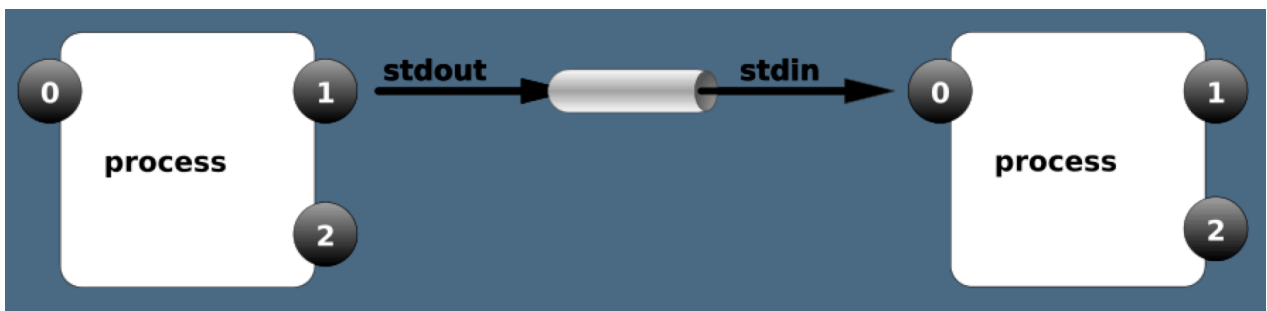
```
My Linux #提示用户输入数据
My Linux #输出数据，进行输出反馈
```

此时，可以分别打开文件"file1"和"file2"，查看其内容是否均是"My Linux"即可判断指令"tee"是否执行成功。

十一、管道 | 基本原理

用法

```
command1 | command2 | command3 | ...
```



```
[root@qfedu.com ~]# ll /dev/ | less
[root@qfedu.com ~]# ps aux | grep 'sshd'
[root@qfedu.com ~]# rpm -qa | grep 'httpd' //查询所有安装的软件包，
过滤包含httpd的包
[root@qfedu.com ~]# yum list | grep 'httpd'
```

实例

案例1: 将/etc/passwd中的用户按UID大小排序

```
[root@qfedu.com ~]# sort -t":" -k3 -n /etc/passwd //以: 分隔, 将第三列按数字升序
```

[root@qfedu.com ~]# sort -t":" -k3 -n /etc/passwd -r //逆序

```
[root@qfedu.com ~]# sort -t":" -k3 -n /etc/passwd | head
```

-t 指定字段分隔符--field-separator
-k 指定列
-n 按数值

案例2: 统计出最占CPU的5个进程

```
[root@qfedu.com ~]# ps aux --sort=-%cpu | head -6
```

案例3: 统计当前/etc/passwd中用户使用的shell类型

思路: 取出第七列 (shell) | 排序 (把相同归类) | 去重

```
[root@qfedu.com ~]# awk -F: '{print $7}' /etc/passwd
```

```
[root@qfedu.com ~]# awk -F: '{print $7}' /etc/passwd | sort
```

```
[root@qfedu.com ~]# awk -F: '{print $7}' /etc/passwd | sort | uniq
```

```
[root@qfedu.com ~]# awk -F: '{print $7}' /etc/passwd | sort | uniq -c
```

```
131  /bin/bash
1    /bin/sync
1    /sbin/halt
63   /sbin/nologin
1    /sbin/shutdown
```

-F: 指定字段分隔符
\$7 第七个字段

案例4: 统计网站的访问情况 top 20

思路: 打印所有访问的连接 | 过滤访问网站的连接 | 打印用户的IP | 排序 | 去重

```
[root@qfedu.com ~]# yum -y install httpd
```

```
[root@qfedu.com ~]# systemctl start httpd
```

```
[root@qfedu.com ~]# systemctl stop firewallld
```

```
[root@qfedu.com ~]# ss -an | grep :80 | awk -F":" '{print $8}' | sort | uniq -c
```

```
4334 192.168.0.66
1338 192.168.10.11
1482 192.168.10.125
44   192.168.10.183
3035 192.168.10.213
375  192.168.10.35
362  192.168.10.39
```

```
[root@qfedu.com ~]# ss -an | grep :80 | awk -F":" '{print $8}' | sort | uniq -c | sort -k1 -rn | head -n 20
```

案例5: 打印当前所有IP

```
[root@qfedu.com ~]# ip addr | grep 'inet ' | awk '{print $2}' | awk -F"/" '{print $1}'
```

```
127.0.0.1
192.168.2.115
```

案例6：打印根分区已用空间的百分比（仅打印数字）

```
[root@qfedu.com ~]# df -P |grep '/$' |awk '{print $5}' |awk -F"%" '{print $1}'
```

十二、重定向实战

- 1、熟记各种重定向操作符
- 2、利用各种重定向操作符进行重定向操作

特例

```
[root@qfedu.com ~]# tr '\n' '|' < /etc/passwd
```

干锋云计算学院