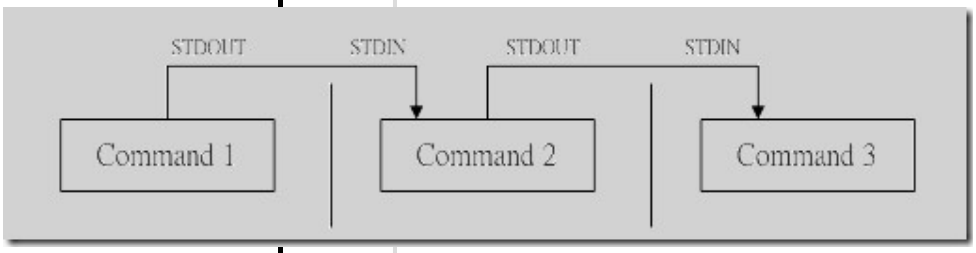
linux shell 管道命令(pipe)使用及与shell重定向区别

看了前面一节：linux shell数据重定向（输入重定向与输出重定向）详细分析 估计还有一些朋友是头晕晕的，好复杂的重定向了。这次我们看下管道命令了。shell管道，可以说用法就简单多了。

管道命令操作符是：”|”,它仅能处理经由前面一个指令传出的正确输出信息，也就是 standard output 的信息，对于 stdandard
error 信息没有直接处理能力。然后，传递给下一个命令，作为标准的输入 standard input.

- **管道命令使用说明：**

先看下下面图：



command1正确输出，作为command2的输入 然后comand2的输出作为，comand3的输入 ，comand3输出就会直接显示在屏幕上面了。

随笔分类
linux(36)
nginx(3)
php(16)
python(1)
web前端(1)
编码解码(1)
心得分享(6)
字符集(3)

随笔档案
2014年5月 (1)
2013年6月 (4)
2013年5月 (4)
2013年1月 (1)
2011年2月 (1)
2010年12月 (1)
2010年10月 (42)
2010年9月 (3)
2010年7月 (2)
2010年6月 (4)

友情连接
查错网
生活查询网

积分与排名
积分 - 90972
排名 - 3153

阅读排行榜
1. linux shell 字符串操作（长度，查找，替换）详解(263364)
2. linux shell 流程控制（条件if,循环【for,while】,选择【case】语句实例(150014)
3. linux shell数据重定向（输入重定向与输出重定向）详细分析(149735)
4. linux shell 自定义函数(定义、返回值、变量作用域）介绍(127688)
5. linux shell 数组建立及使用技巧(125818)

通过管道之后：comand1,comand2的正确输出不显示在屏幕上面

注意：

1、管道命令只处理前一个命令正确输出，不处理错误输出

2、管道命令右边命令，必须能够接收标准输入流命令才行。

实例：

```
[chengmo@centc
shell]$ cat
test.sh | grep
-n 'echo'
5:    echo "ve
good!";
7:    echo
"good!";
9:    echo
"pass!";
11:    echo "r
pass!";
#读出test.sh文
内容，通过管道转
给grep 作为输入
容

[chengmo@centc
shell]$ cat
test.sh test1.
| grep -n 'ech
cat: test1.sh:
有那个文件或目录
5:    echo "ve
good!";
7:    echo
"good!";
```

```
9:    echo "pass!";
11:    echo "r
pass!";
#cat test1.sh
在，错误输出打印
屏幕，正确输出通
管道发送给grep


[chengmo@centc
shell]$ cat test.sh test1.
2>/dev/null | grep -n 'echo'
5:    echo "ve
good!";
7:    echo "good!";
9:    echo "pass!";
11:    echo "r
pass!";
#将test1.sh 没
找到错误输出重定
输出给/dev/nul
文件，正确输出通
管道发送给grep


[chengmo@centc
shell]$ cat test.sh | ls
catfile
httprequest.tx
secure
test
testfdread.sh
testpipe.sh
testsh.sh
testwhile2.sh
envcron.txt
python
sh
```

```
testcase.sh
testfor2.sh
testselect.sh
test.txt
text.txt
env.txt
release
sms
testcronenv.sh
testfor.sh
test.sh
testwhile1.sh
#读取test.sh内
容，通过管道发送给
ls命令，由于ls 不
支持标准输入，因此
数据被丢弃
```

这里实例就是对上面2点注意的验证。作用接收标准输入的命令才可以用作管道右边。否则传递过程中数据会抛弃。 常用来作为接收数据管道命令有：sed,awk,cut,head,top,less,more,wc,join,sort,split等等，都是些文本处理命令。

- **管道命令与重定向区别**

区别是：

1、左边的命令应该有标准输出 | 右边的命令应该接受标准输入

　左边的命令应该有标准输出 > 右边只能是文件

　左边的命令应该需要标准输入 < 右边只能是文件

2、**管道**触发两个子进程执行"|"两边的程序；而**重定向**是在一个进程内执行

这些都是网上总结很多的，其实只要多加清楚用法，也一定有自己的一份不同描述。

实例：

```
#可以相互转换情
#输入重定向

[chengmo@centc
shell]$ cat
test.sh| grep
'echo'
5:    echo "ve
good!";
7:    echo
"good!";
9:    echo
"pass!";
11:    echo "r
pass!";
#"|"管道两边都
是shell命令


[chengmo@centc
shell]$ grep -
'echo'
<test.sh
5:    echo "ve
good!";
7:    echo
"good!";
9:    echo
"pass!";
11:    echo "r
pass!";

```

```
#"重定向"符号，
边只能是文件（普
文件，文件描述符
文件设备）


[chengmo@centos
shell]$ mail -
'test'
8292669@qq.com
<test.sh
[chengmo@centos
shell]$ cat
test.sh|mail -
'test'
8292669@qq.com
#以上2个也相同，
test.sh内容发送
指定邮箱。


[chengmo@centos
shell]$ (sed -
'1,$p'|grep -n
'echo')<test.s
5:    echo "ve
good!";
7:    echo
"good!";
9:    echo
"pass!";
11:    echo "n
pass!";
#这个脚本比较有
思了。由于前面是
道，后面需要把
test.sh内容重定
到 sed ，然后se
输出通过管道，输
给grep.需要将前
用"()"运算符括起
来。在单括号内的
令，可以把它们看
一个象一个命令样
```

```
如果不加括号
test.sh就是gre
的输入了。


#上面一个等同于
个
[chengmo@cento
shell]$ sed -n
'1,$p'<test.sh
grep -n 'echo'
5:    echo "ve
good!";
7:    echo
"good!";
9:    echo
"pass!";
11:    echo "n
pass!";

#重定向运算符，
shell命令解析前
首先检查的（一个
令，执行前一定检
好它的输入，输出
也就是0,1,2 设备
否准备好），所以
先级会最高


[chengmo@cento
shell]$ sed -n
'1,10p'<test.s
| grep -n 'ech
<testsh.sh
10:echo $total
18:echo $total
21:     echo
"ok";
#哈哈，这个grep
接受管道输入，又
testsh.sh输入，
是不是2个都接收
呢。刚才说了"<"
```

```
算符会优先，管道
没有发送数据前，
grep绑定了
testsh.sh输入，
样sed命令输出就
抛弃了。这里一定
小心使用

#输出重定向

[chengmo@centos
shell]$ cat
test.sh>test.t
[chengmo@centos
shell] cat
test.sh|tee
test.txt
&>/dev/null
#通过管道实现将
果存入文件,还需
借助命令tee，它
把管道过来标准输
写入文件
test.txt ,然后
标准输入复制到标
输出(stdout),所
重定向
到/dev/null 不
示输出
#">"输出重定向,
往在命令最右边，
收左边命令的，输
结果，重定向到指
文件。也可以用到
令中间。


[chengmo@centos
shell]$ ls
test.sh test1.
testsh.sh
2>err.txt | gr
'test'
test.sh
```

```
testsh.sh
#目录下面有：test,testsh文件，test1.sh不存在，因此将ls 命令错误输出输入到 err.txt 正确输出，还会通过管道发送到grep命令。
[chengmo@centos5 shell]$ ls test.sh test1.sh testsh.sh &>err.txt | grep 'test'
#这次打印结果是空，&代表正确与错误输出 都输入给 err.txt，通过管道继续往下面传递数据为空,所以没有什么显示的

#同样">"输出重定向符，优先级也是先解析，当一个命令有这个字符，它就会与左边命令标准输出绑定。准备好了这些，就等待命令执行输出数据，它就开始接收
```

再概括下：

从上面例子可以看，重定向与管道在使用时候很多时候可以通用，其实，在shell里面，经常是【条条大路通罗马】的。一般如果是命令间传递参数，还是管道的好，如果处理输出结果需要重定向到文

件，还是用重定向输出比较好。

命令执行顺序可以看下：Linux Shell 通配符、元字符、转义符使用实例介绍

- **shell脚本接收管道输入**

  有意思的问题：

  既然作用管道接收命令，需要可以接收标准的输入，那么我们shell脚本是否可以开发出这样的基本程序呢？（大家经常看到的，都是一些系统的命令作为管道接收方）

  实例（testpipe.sh）：

```
#!/bin/sh

 if [ $# -gt 0 ];then
     exec 0<$1;
#判断是否传入参数：文件名，如果传入，将该文件绑定到标准输入
 fi

 while read line
 do
```

```
        echo
    $line;
     done<&0;
    #通过标准输入
    循环读取内容
     exec 0&-;
    #解除标准输入
    绑定
```

运行结果：

```
[chengmo@centc
shell]$ cat
testpipe.txt
1,t,est pipe
2,t,est pipe
3,t,est pipe
4,t,est pipe
#testpipe.txt
是需要读取的测试
本

[chengmo@centc
shell]$ cat
testpipe.txt |
testpipe.sh
1,t,est pipe
2,t,est pipe
3,t,est pipe
4,t,est pipe
#通过cat 读取
testpipe.txt 发
给testpipe.sh
准输入

[chengmo@centc
shell]$ sh
testpipe.sh
testpipe.txt
1,t,est pipe
2,t,est pipe
3,t,est pipe
4,t,est pipe
```

```
#testpipe.sh 通过
出入文件名读取文件
内容
```

分类: linux

标签: linux, shell

好文要顶 关注我 收藏该文

程默
关注 - 0
粉丝 - 319
+加关注

9 0

« 上一篇：php 实现进制转换(二进制、八进制、十六进制) 互相转换
» 下一篇：linux shell 脚本实现tcp/upd协议通讯 (重定向应用)

posted on 2010-10-21 09:53 程默 阅读(94259) 评论(9) 编辑 收藏

评论

#1楼 2010-10-21 11:14 David Young 杨博华

呃..... 之前真没去想这个东西有多复杂

支持(0) 反对(0)

#2楼[楼主] 2010-10-21 12:37 程默

shell基本，很多命令都有常用方法，还有些特殊方法，这就是它的复杂所在。

支持(0) 反对(0)

#3楼 2014-05-13 10:00 析冥

cmd1 2>&1 | cmd2 可以使管道接收标准错误。。。

支持(0) 反对(0)

#4楼 2015-04-20 16:32 森度

拜读了

支持(0) 反对(0)

#5楼 2015-11-25 11:19 少林功夫好

@ 析冥
这个有点意思。

支持(0) 反对(0)

#6楼 2015-11-25 11:22 少林功夫好

testpipe.sh 脚本还不太理解。

支持(0) 反对(0)

#7楼 2016-01-23 14:03 GoingMyWay

不错

支持(0) 反对(0)

#8楼 2017-01-04 10:01 bob.dong

围绕这个主题非常详尽，描述清楚。能否请教一下 下面命令行中 "-" 是什么意思呢？是用来表示前一个管道的标准输出吗？不用会有什么不同吗？我搜索了半天，没发现相

关的信息。谢谢了！

./a.sh r1.csv r2.csv \
| ./b.sh ref.csv - 2> out.log \
| ./c.sh convert - > out.dat;

支持(0) 反对(0)

#9楼 2017-01-04 10:50 bob.dong

搞清楚了，- 是b.sh, c.sh 的参数占位符，详见 http://weibo.com/1652171664/EpeGfbt6d。

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。