# 管道、环境变量与常用命令

## 知识点

### 管道

#### 概念

管道类似于文件重定向，可以将前一个命令的stdout重定向到下一个命令的stdin。

#### 要点

管道命令仅处理stdout，会忽略stderr。

管道右边的命令必须能接受stdin。

多个管道命令可以串联。

#### 与文件重定向的区别

文件重定向左边为命令，右边为文件。

管道左右两边均为命令，左边有stdout，右边有stdin。

#### 举例

统计当前目录下所有python文件的总行数，其中find、xargs、wc等命令可以参考常用命令这一节内容。

find . -name '\*.py' | xargs cat | wc -l

### 环境变量

#### 概念

Linux系统中会用很多环境变量来记录配置信息。

环境变量类似于全局变量，可以被各个进程访问到。我们可以通过修改环境变量来方便地修改系统配置。

#### 查看

列出当前环境下的所有环境变量：

env # 显示当前用户的变量

set # 显示当前shell的变量，包括当前用户的变量;

export # 显示当前导出成用户变量的shell变量

输出某个环境变量的值：

echo $PATH

#### 修改

环境变量的定义、修改、删除操作可以参考3. shell语法——变量这一节的内容。

为了将对环境变量的修改应用到未来所有环境下，可以将修改命令放到~/.bashrc文件中。

修改完~/.bashrc文件后，记得执行source ~/.bashrc，来将修改应用到当前的bash环境下。

为何将修改命令放到~/.bashrc，就可以确保修改会影响未来所有的环境呢？

每次启动bash，都会先执行~/.bashrc。

每次ssh登陆远程服务器，都会启动一个bash命令行给我们。

每次tmux新开一个pane，都会启动一个bash命令行给我们。

所以未来所有新开的环境都会加载我们修改的内容。

#### 常见环境变量

HOME：用户的家目录。

PATH：可执行文件（命令）的存储路径。路径与路径之间用:分隔。当某个可执行文件同时出现在多个路径中时，会选择从左到右数第一个路径中的执行。下列所有存储路径的环境变量，均采用从左到右的优先顺序。

LD\_LIBRARY\_PATH：用于指定动态链接库(.so文件)的路径，其内容是以冒号分隔的路径列表。

C\_INCLUDE\_PATH：C语言的头文件路径，内容是以冒号分隔的路径列表。

CPLUS\_INCLUDE\_PATH：CPP的头文件路径，内容是以冒号分隔的路径列表。

PYTHONPATH：Python导入包的路径，内容是以冒号分隔的路径列表。

JAVA\_HOME：jdk的安装目录。

CLASSPATH：存放Java导入类的路径，内容是以冒号分隔的路径列表。

### 常用命令

Linux命令非常多，本节讲解几个常用命令。其他命令依赖于大家根据实际操作环境，边用边查。

#### 系统状况

1. top：查看所有进程的信息（Linux的任务管理器）

打开后，输入M：按使用内存排序

打开后，输入P：按使用CPU排序

打开后，输入q：退出

1. df -h：查看硬盘使用情况
2. free -h：查看内存使用情况
3. du -sh：查看当前目录占用的硬盘空间
4. ps aux：查看所有进程
5. kill -9 pid：杀死编号为pid的进程

传递某个具体的信号：kill -s SIGTERM pid

1. netstat -nt：查看所有网络连接
2. w：列出当前登陆的用户
3. ping www.baidu.com：检查是否连网

#### 文件权限

chmod：修改文件权限

chmod +x xxx：给xxx添加可执行权限

chmod -x xxx：去掉xxx的可执行权限

chmod 777 xxx：将xxx的权限改成777

chmod 777 xxx -R：递归修改整个文件夹的权限

#### 文件检索

1. find /path/to/directory/ -name '\*.py'：搜索某个文件路径下的所有\*.py文件
2. grep xxx：从stdin中读入若干行数据，如果某行中包含xxx，则输出该行；否则忽略该行。
3. wc：统计行数、单词数、字节数

既可以从stdin中直接读入内容；也可以在命令行参数中传入文件名列表；

wc -l：统计行数

wc -w：统计单词数

wc -c：统计字节数

1. tree：展示当前目录的文件结构

tree /path/to/directory/：展示某个目录的文件结构

tree -a：展示隐藏文件

1. ag xxx：搜索当前目录下的所有文件，检索xxx字符串
2. cut：分割一行内容

从stdin中读入多行数据

echo $PATH | cut -d ':' -f 3,5：输出PATH用:分割后第3、5列数据

echo $PATH | cut -d ':' -f 3-5：输出PATH用:分割后第3-5列数据

echo $PATH | cut -c 3,5：输出PATH的第3、5个字符

echo $PATH | cut -c 3-5：输出PATH的第3-5个字符

1. sort：将每行内容按字典序排序

可以从stdin中读取多行数据

可以从命令行参数中读取文件名列表

1. xargs：将stdin中的数据用空格或回车分割成命令行参数

find . -name '\*.py' | xargs cat | wc -l：统计当前目录下所有python文件的总行数

#### 查看文件内容

1. more：浏览文件内容

回车：下一行

空格：下一页

b：上一页

q：退出

1. less：与more类似，功能更全

回车：下一行

y：上一行

Page Down：下一页

Page Up：上一页

q：退出

1. head -3 xxx：展示xxx的前3行内容

同时支持从stdin读入内容

1. tail -3 xxx：展示xxx末尾3行内容

同时支持从stdin读入内容

#### 用户相关

history：展示当前用户的历史操作。内容存放在~/.bash\_history中

#### 工具

1. md5sum：计算md5哈希值

可以从stdin读入内容

也可以在命令行参数中传入文件名列表；

1. time command：统计command命令的执行时间
2. ipython3：交互式python3环境。可以当做计算器，或者批量管理文件。

! echo "Hello World"：!表示执行shell脚本

1. watch -n 0.1 command：每0.1秒执行一次command命令
2. tar：压缩文件

tar -zcvf xxx.tar.gz /path/to/file/\*：压缩

tar -zxvf xxx.tar.gz：解压缩

1. diff xxx yyy：查找文件xxx与yyy的不同点

#### 安装软件

1. sudo command：以root身份执行command命令
2. apt-get install xxx：安装软件
3. pip install xxx --user --upgrade：安装python包

## 2. 创建作业 & 测试作业的正确性

homework 7 create 可以重新创建所有lesson\_7的作业

homework 7 create id 可以单独创建lesson\_7的第id个作业. e.g.

homework 7 create 0 可以只重新创建lesson\_7的第0个作业

homework 7 test 可以评测lesson\_7的所有作业

## 3. 作业

创建好作业后，先进入文件夹/home/acs/homework/lesson\_7/，然后：

(0) 进入homework\_0文件夹，然后：

[1] 编写可执行shell脚本，文件名为mydu，使得在任意目录执行该脚本后，可以输出当前目录所占用的磁盘空间大小，格式化输出（例如 1K, 234M, 2G）。

[2] 修改环境变量PATH，使得在任意目录下输入mydu均可执行上述脚本。

(1) 进入homework\_1文件夹，然后：

[1] 给当前目录下的所有文件添加可读权限。

[2] 统计当前目录下共有多少个以.cpp结尾的文件，将结果写入ans.txt中。

(2) 进入homework\_2文件夹，然后：

[1] 统计当前目录下所有cpp文件的总行数，将结果写入ans1.txt中。

[2] 遍历当前目录下所有py文件，统计共有多少行包含字符串thrift，将结果写入ans2.txt中。

(3) 进入homework\_3文件夹，然后删除当前目录下的所有py文件。

(4) 进入homework\_4文件夹，然后：

[1] 当前目录下包含一个文件scores.txt，文件中有若干行数据，每行数据包含用空格隔开的3个字段：姓名、数学成绩、算法成绩。

[2] 将scores.txt文件的内容以空格为分隔符分隔成3列，将第一列内容写入names.txt，第二列内容写入mathematics.txt，第三列内容写入algorithm.txt。

注意：scores.txt文件本身不要修改。

(5) 进入homework\_5文件夹，然后：

[1] 当前目录下包含一个文件scores.txt，文件中有若干行数据，每行数据包含用空格隔开的3个字段：姓名、数学成绩、算法成绩。

[2] 将scores.txt文件的内容以空格为分隔符分隔成3列，将第一列内容按字典序从小到大排序后写入names.txt。

注意：scores.txt文件本身不要修改。

(6) 进入homework\_6文件夹，然后将scores.txt文件的前5行写入top.txt，后4行写入bottom.txt。

(7) 进入homework\_7文件夹，计算scores.txt文件的md5sum，将结果写入ans.txt。注意：ans.txt文件中只包含长度为32的md5sum，不包含其他内容。

(8) 进入homework\_8文件夹，然后：

[1] 将dir\_a文件夹压缩为project\_a.tar.gz。

[2] 将project\_b.tar.gz压缩文件解压到当前目录。

注意：不要删除dir\_a文件夹和project\_b.tar.gz文件。

(9) 进入homework\_9文件夹，然后计算 2^112+3^78，将结果写入ans.txt中。

作业代码：

homework\_0

vim mydu

#! /bin/bash

du -sh

:wq

cd

vim .bashrc

G

export PATH=/home/acs/homework/lesson\_7/homework\_0:$PATH

:wq

source ~/.bashrc

进入homework\_0文件夹

vim mydu

\*\*\*

#! /bin/bash

du -sh

\*\*\*

[1] 添加可执行权限：

chmod +x mydu

[2] 在~/.bashrc文件末尾添加一行：

export PATH=/home/acs/homework/lesson\_7/homework\_0:$PATH

应用该修改：

source ~/.bashrc

homework\_1

chmod +r \* -R

find . -name '\*.cpp' | wc -l > ans.txt

进入homework\_1文件夹；

[1] 添加可读权限：

chmod +r \* -R

[2] 统计行数:

find . -name '\*.cpp' | wc -l > ans.txt

homework\_2

find . -name '\*.cpp' | xargs cat | wc -l > ans1.txt

find . -name '\*.py' | xargs cat | grep thrift | wc -l > ans2.txt

进入homework\_2文件夹。

[1] 统计cpp文件总行数

find . -name '\*.cpp' | xargs cat | wc -l > ans1.txt

[2] 统计py文件包含thrift字符串的总行数

find . -name '\*.py' | xargs cat | grep thrift | wc -l > ans2.txt

homework\_3

find . -name '\*.py' | xargs rm

进入homework\_3文件夹。

find . -name '\*.py' | xargs rm

homework\_4

cat scores.txt | cut -d ' ' -f 1 > names.txt

cat scores.txt | cut -d ' ' -f 2 > mathematics.txt

cat scores.txt | cut -d ' ' -f 3 > algorithm.txt

进入homework\_4文件夹。

cat scores.txt | cut -d ' ' -f 1 > names.txt

cat scores.txt | cut -d ' ' -f 2 > mathematics.txt

cat scores.txt | cut -d ' ' -f 3 > algorithm.txt

homework\_5

cat scores.txt | cut -d ' ' -f 1 | sort > names.txt

进入homework\_5文件夹。

cat scores.txt | cut -d ' ' -f 1 | sort > names.txt

homework\_6

head -5 scores.txt > top.txt

tail -4 scores.txt > bottom.txt

进入homework\_6文件夹。

[1] 选取前5行

head -5 scores.txt > top.txt

[2] 选取后4行

tail -4 scores.txt > bottom.txt

homework\_7

md5sum scores.txt | cut -c 1-32 > ans.txt

进入homework\_7文件夹。

md5sum scores.txt | cut -c 1-32 > ans.txt

homework\_8

tar -zcvf project\_a.tar.gz dir\_a/\*

tar -zxvf project\_b.tar.gz

进入homework\_8文件夹。

[1] 压缩文件

tar -zcvf project\_a.tar.gz dir\_a/\*

[2] 解压文件

tar -zxvf project\_b.tar.gz

homework\_9

ipython3

res = 2\*\*112 + 3\*\*78

!echo $res > ans.txt

进入homework\_9文件夹

ipython3 # 打python3交互环境

res = 2\*\*112 + 3\*\*78

! echo $res > ans.txt # ! 表示后面的内容是shell脚本