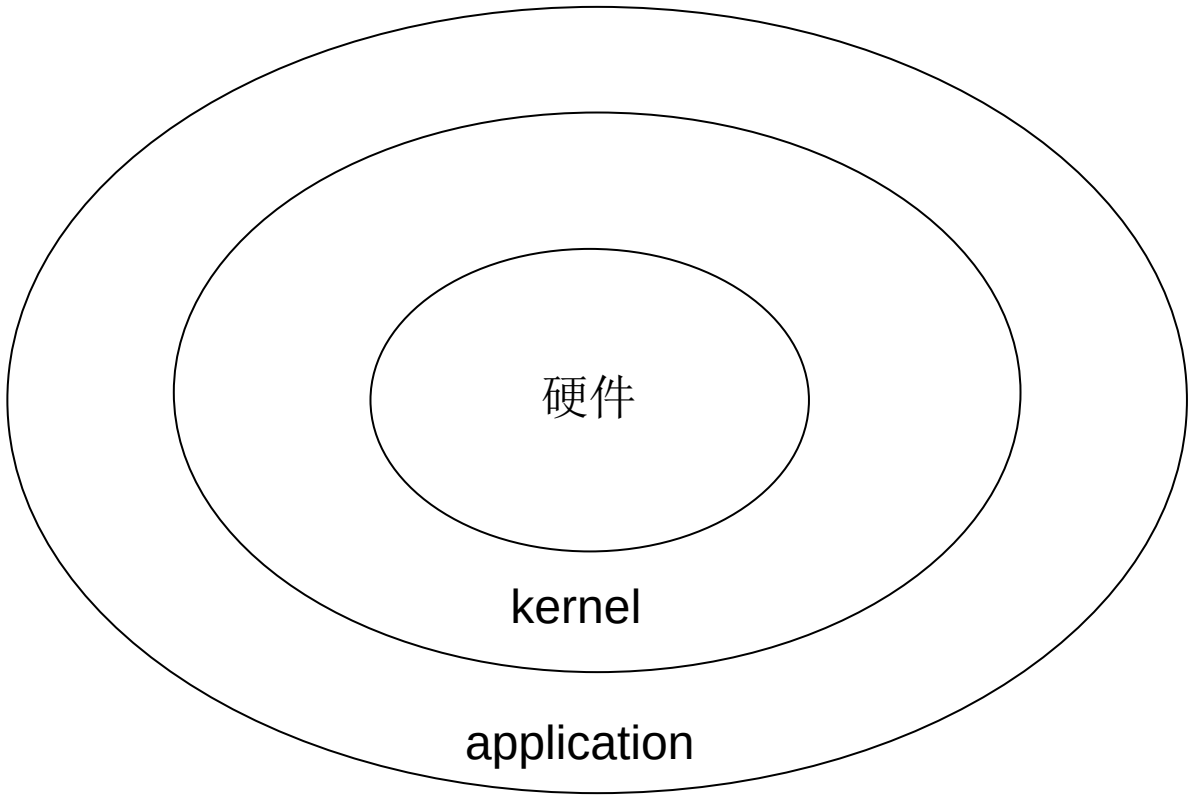
双星教育
Linux 系统管理员

Shell 基础篇

Terry Tsang
曾绍坤
signmem@netease.com

什么是 shell



shell

Shell 内部命令

外部命令

为什么要使用 shell

为内核工作
提供环境变量

思考：什么是环境变量，如何获得环境变量

fork 系统调用
wait 系统调用
exec 系统调用
exit 系统调用

Linux 下的 shell

/etc/shells

- /bin/sh
- /bin/bash
- /bin/ksh
- /bin/tcsh
- /bin/csh

思考：如何改变用户的 shell 环境

获得环境变量

环境变量常用配置文件

- /etc/bashrc
- /etc/profile
- ~/.bashrc
- ~/.bash_logout
- ~/.bash_profile
- ~/.bash_history

环境变量

什么是环境变量

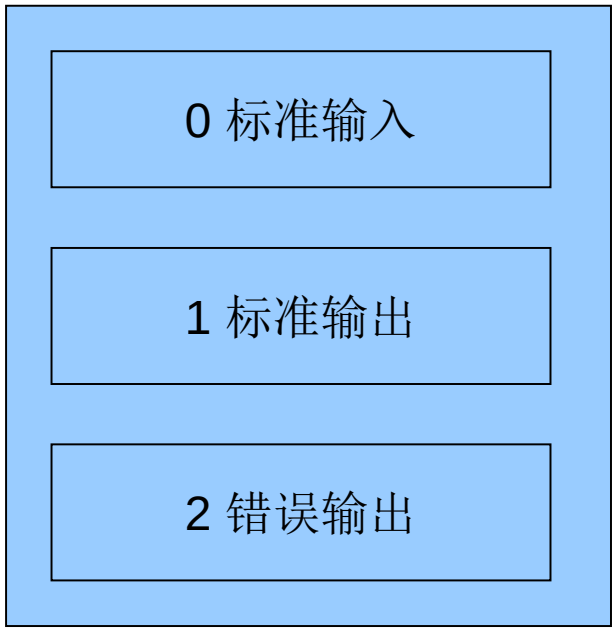
如何查看环境变量

env，set，echo, export

bash 下常用的环境变量

HOME，USER, HOSTNAME, SHELL, LANG, HISTSIZE

Shell 下的重定向



思考

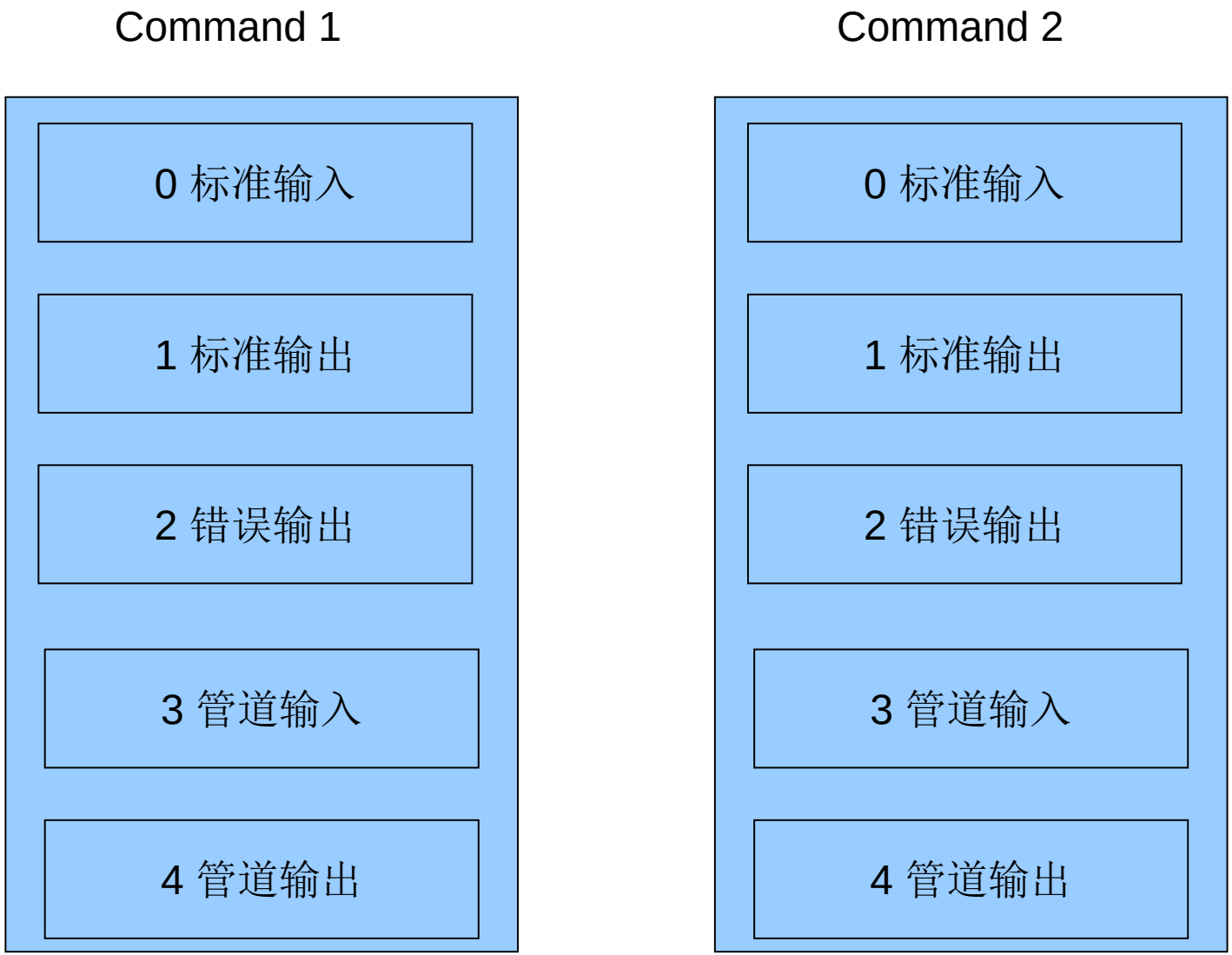
思考：什么时候需要改变输出效果

例子

参考例子

```
ls -l  /etc  >  /tmp/etc_file
locate  passwd  >  /tmp/locate
tail  -f  /var/log/messages  >  /dev/tty8
mail  -s subject   user@localhost   <  file
ls  -l   abc   2>   err_file
ls  -l   abc   >>    std_file
ls  -l   abc  > std_file  2> err_file
ls  -l   abc  > std_file  2>&1
ls  -l   abc  &>  all output
ls  -l   abc  2 >>   err_file
```

管道


Command 1
0 标准输入
1 标准输出
2 错误输出
3 管道输入
4 管道输出
Command 2
0 标准输入
1 标准输出
2 错误输出
3 管道输入
4 管道输出

思考

什么时候用到了管道?

例子

参考数据

ls -l /etc | grep passwd

grep boot /etc * -R | more

ls -l /etc | more

grep

-v Invert the sense of matching, to select non-matching lines.

-c count of matching lines for each input file

-C NUM -A NUM -B NUM

--with-filename

--no-filename

-m --max-count=

-n line number

-l show the detail of the file

-h show the filename

-i Ignore case distinctions in both the PATTERN and the input files

-w matches that form whole words

正则表达式

? The preceding item is optional and matched at most once.

* The preceding item will be matched zero or more times.

+ The preceding item will be matched one or more times.

{n} The preceding item is matched exactly n times.

{n,} The preceding item is matched n or more times.

{,m} The preceding item is matched at most m times.

{n,m} The preceding item is matched at least n times, but not more than m times.

^

$

.

[]

find

-name
-perm
-readable || -writable || -executable
-user || -group
-nogroup || -nouser
-type (b d c p l f)
-size n[c
-follow || -P
-maxdepth N 目录深度
-atime || -ctime || -mtime (+-n)
-amin || -cmin || -mmin (+-n)

Find格式化

-print0
-printf
 %A
    小时 H 00..24 l 01..12 k 0..23 l 0..12 p AM PM
    分钟 M 00..59 l 0 .. 59
    r hh:mm:ss[AM|PM] T hh:mm:ss
    S 00..60
    a sun..sat A sunday..saturday w 0..6
    b jan..dec B January..December
    d 01..31 D mm/dd/yy
    m 01..12
    Y yyyy
%d 目录
%f 文件名 %p
%s 文件大小
%g groupname %G gid
%u username %U uid

shell 常用命令

tee

lspci -v | tee lspci.out | less

paste

wc

wc -l, wc -w, wc -c

tr

tr ‘A-Z’ ‘a-z’ < file

uniq

cat file | uniq

sort

cat file | sort

shell 常用命令

head / tail

grep

find

diff

cut

cut -f4 file

cut -f3 -d:

cut -f2 -d“ “ file

cut -c2-5 file

Shell 基础语法

{} [] ()

&& || ; 的用法

“” ‘’ `` 的用法

正则表达式

^abc

abc$

[Ss]ingl[eE]

^terry$

te..y

[.*a]

[^$]

^…..$

[a-zA-Z]

[a-zA-Z0-9]

\.

\^terry

[0-9]\{2\}-[0-9]\{2\}-[0-9]\{4\}

[0-9]\{3\}\.[0-9]\{3\}\.[0-9]\{3\}\.[0-9]\{3\}

Shell 常见语法

$#	程序命令行参数的数量
$0	程序名
$*	以(“$1 $2...”)的形式保存所有输入的命令行参数
$@	以(“$1”“$2”...)的形式保存所有输入的命令行参数
$$	获得程序 PID
$?	前一个命令的返回码
$!	返回后台进程 ID

返回值

0	no errors
1	file system error corrected
2	system should be rebooted
4	file system errors left uncorrected
8	operational error
16	usage or syntax error
128	shared library error

程序的运行

回顾: 进程管理

- 父进程
- 子进程
- fork-and-exec 机制
- 进程运行的特性

回顾

shell 中进程的运行

常见 shell 语法

重定向

“”, ‘’, ``符号作用

`` 符号等价于 $() 功能

`hostname` == $(hostname)

`expr 23+33`

$(23 + 33)

反义字符使用 \

练习

显示系统中默认以 /bin/bash 登陆的用户

在 /etc/passwd 中找到以 “daemon”开头的行

显示 /etc/passwd 中除了以 /bin/bash 登陆的用户? 并计算总数量?

计算 /usr/share/doc 中包含多少子目录

计算 /usr/share/doc 子目录中包含多少个README 文件

注意：不要把 README.xxxx 文件计算在内

计算出 /etc 下连同子目录中有多少个 *.conf 的配置文件总数

列出系统中文件被修改超过 10 小时的文件

计算 /bin/bash 在 /etc/passwd 出现了多少次

计算 daemon 单词在 /etc/passwd 共出现了多少次

谢谢