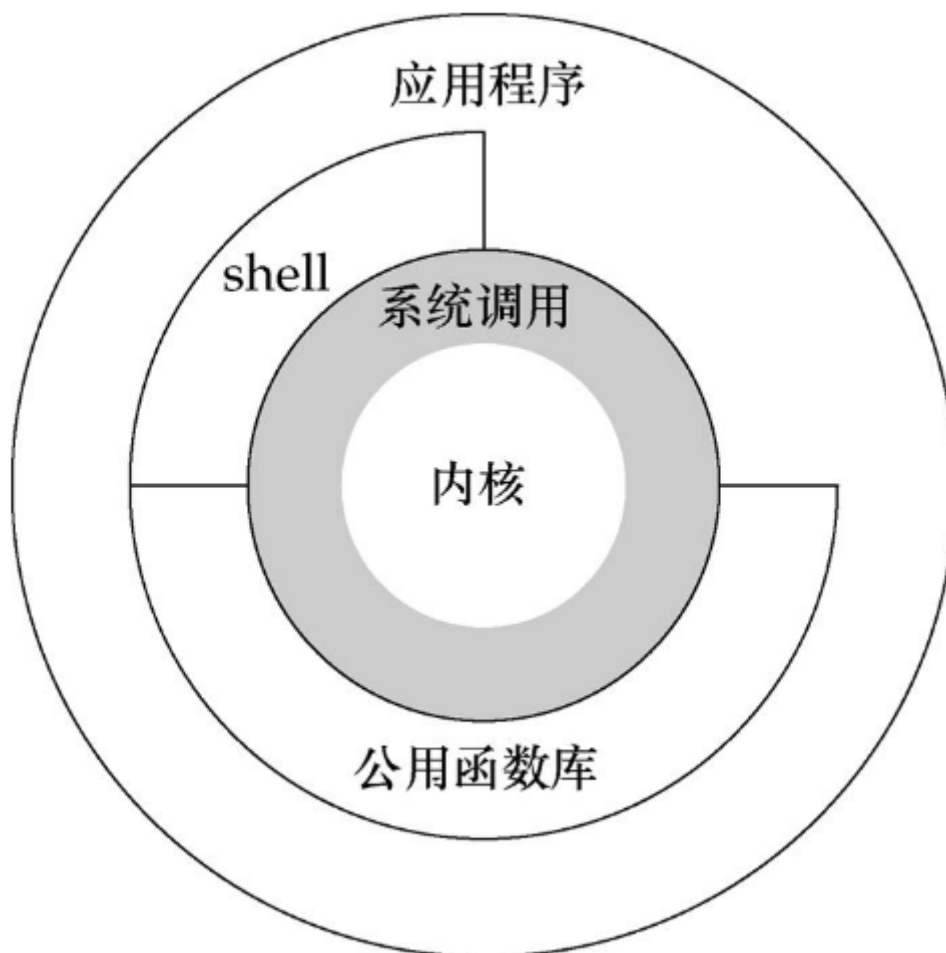


初识 Shell

虽然我们已经安装好了系统，但是光会安装不会操作是不够的。我们还要像玩手机一样熟悉并记忆操作方法。

Shell 是系统的**用户界面**，提供了用户与内核进行交互操作的一种接口。它接收用户输入的命令并把它送入内核去执行。实际上 Shell 是一个**命令解释器**，它解释用户输入的命令并且把用户的意图传达给内核。（可以理解为用户与内核之间的翻译官角色）



我们可以使用 Shell 实现对Linux系统的大部分管理，例如：

1. 文件管理
2. 用户管理
3. 权限管理
4. 磁盘管理
5. 软件管理
6. 网络管理 ...

使用 Shell 的两种方式

- 交互式命令行
 - 默认等待用户输入命令，输入一行回车后执行一行命令
 - 适合少量的工作

- Shell 脚本
 - 将需要执行的命令和逻辑判断语句都写入一个文件后执行该文件
 - 适合完成复杂，重复性工作

Bash shell 提示符

登录 Linux 系统之后，默认进入交互式的命令行界面，在光标前边会出现提示符

```
[root@localhost ~]#  
[用户名@主机名 目录名]权限标识
```

解释说明：

- 用户名：当前登录的用户
- 主机名：当前这台主机的名字，默认叫localhost
- 目录名：当前光标所在的目录；当前用户家目录表示成~
- 权限标识：超级管理员权限就表示为#；普通用户标识为\$

修改提示符：

```
# 这个提示符格式被`$PS1`控制，我们可以查看这个变量  
# \u表示是用户名 \h表示的是主机名 \W表示的当前所在目录 \h是权限标识  
[root@localhost ~]# echo $PS1  
[\u@\h \W]\h  
  
# 可以通过export命令修改PS1变量，让提示符可以根据你的习惯变化  
[root@localhost ~]# export PS1="{\u@\h}\W \h"  
{root@localhost}~ $  
  
# 修改回原来的样子  
{root@localhost}~ $ $export PS1="[\u@\h \W]\h "
```

常用命令

Linux 常见命令比较多，这边只列出初学者最常用的部分命令，大家可以根据命令有意去进行练习。

注意 Linux 会准确的识别出命令的大小写，所以大家需要注意大小写的问题。命令选项和参数之间是用空格进行分隔，请大家在输入的时候注意不要缺失空格。

学习 Linux 最重要的就是以下三个方面

1. 命令的积累
2. 原理的掌握
3. 大量的实战

下面就是开始第一步，积累基础的命令

ls

用于显示指定工作目录下之内容（列出目前工作目录所含之文件及子目录）

```
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
```

常用选项

选项	含义
-a	显示所有文件及目录 (.开头的隐藏文件也会列出)
-l	除文件名称外，亦将文件型态、权限、拥有者、文件大小等资讯详细列出
-r	将文件以相反次序显示(原定依英文字母次序)
-t	将文件依建立时间之先后次序列出
-A	同 -a，但不列出 . (目前目录) 及 .. (父目录)
-F	在列出的文件名称后加一符号；例如可执行档则加 *, 目录则加 /，链接加@
-R	若目录下有文件，则以下之文件亦皆依序列出
-h	将显示出来的文件大小以合适的单位显示出来

案例演示

```
# 查看当前目录下的文件
[root@localhost ~]# ls

# 查看根目录下的文件，查看/usr目录下的文件
[root@localhost ~]# ls /
[root@localhost ~]# ls /usr

# 查看当前目录下所有文件，包括隐藏文件
[root@localhost ~]# ls -a

# 查看当前目录下文件详情，包括隐藏文件
[root@localhost ~]# ls -lha

# 查看 /usr/sbin/ 目录下的文件，并且显示出目录，文件，程序的区别
# 可以看到普通文件只有文件名，可执行文件后面带*，文件夹后面带/
[root@localhost ~]# ls -F /usr/sbin/

# 查看 /etc/ssh/ 目录下的文件，如果有文件夹，那么将文件夹中的文件也显示出来
[root@localhost ~]# ls -FR /etc/ssh/

# 显示详细的信息
```

```
[root@localhost ~]# ls -FRl
```

cd

用于切换当前工作目录

```
cd: cd [-L|[-P [-e]] [-@]] [dir]
    Change the shell working directory.
```

案例演示

```
# 跳转到 /usr/bin 目录下
[root@localhost ~]# cd /usr/bin
# 跳到自己的 home 目录
[root@localhost bin]# cd ~
# 跳到目前目录的上一层
[root@localhost ~]# cd ..
```

pwd

显示工作目录

```
pwd: pwd [-LP]
    Print the name of the current working directory.
```

clear

用于清除屏幕

```
Usage: clear [options]
```

echo

用于字符串的输出

```
echo [option]... [string]...
```

常用选项

选项	含义
-n	不输出行尾的换行符
-e	允许对下面列出的加反斜线转义的字符进行解释
\\	反斜线
\a	报警符(BEL)
\b	退格符
\c	禁止尾随的换行符
\f	换页符
\n	换行符
\r	回车符
\t	水平制表符
\v	纵向制表符
-E	禁止对在 STRINGs 中的那些序列进行解释

案例演示

```
# 显示出 hello world
[root@localhost ~]# echo "hello world"
# 用两行显示出 hello world
[root@localhost ~]# echo -e "hello\nworld"
# 输出 hello world 的时候让系统发出警报音
[root@localhost ~]# echo -e "hello\aworld"
```

系统命令

poweroff

用于关闭计算机并切断电源

```
poweroff [OPTIONS...]
Power off the system.
```

常用选项

选项	含义
----	----

选项	含义
-n	这个选项用于在关机时不执行文件系统的同步操作，即不调用 <code>sync()</code> 系统调用。通常，系统在关机时会自动同步所有挂载的文件系统，以确保所有挂起的磁盘写入操作都完成，从而避免数据丢失。使用 <code>-n</code> 参数可以跳过这个同步过程。
-w	仅记录关机信息到 <code>/var/log/wtmp</code> 文件中，但并不实际执行关机操作。
-d	不把记录写到 <code>/var/log/wtmp</code> 文件里
-f	强制关机。此参数会立即停止所有进程并关闭系统，而不是正常关机流程。

reboot

用来重新启动计算机

```
reboot [OPTIONS...] [ARG]
Reboot the system
```

常用选项

选项	含义
-n	这个选项用于在关机时不执行文件系统的同步操作，即不调用 <code>sync()</code> 系统调用。通常，系统在关机时会自动同步所有挂载的文件系统，以确保所有挂起的磁盘写入操作都完成，从而避免数据丢失。使用 <code>-n</code> 参数可以跳过这个同步过程。
-w	仅记录重启信息到 <code>/var/log/wtmp</code> 文件中，但并不实际执行重启操作。
-d	不把记录写到 <code>/var/log/wtmp</code> 档案里（ <code>-n</code> 这个参数包含了 <code>-d</code> ）
-f	强迫重开机，不呼叫 <code>shutdown</code> 这个指令

whoami

用于显示自身用户名称

```
[root@localhost ~]# whoami
root
```

快捷键

快捷键	作用
<code>^C</code>	终止前台运行的程序
<code>^D</code>	退出 等价exit
<code>^L</code>	清屏
<code>^A</code>	光标移动到命令行的最前端
<code>^E</code>	光标移动到命令行的后端
<code>^U</code>	删除光标前所有字符
<code>^K</code>	删除光标后所有字符
<code>^R</code>	搜索历史命令，利用关键词

帮助命令

history

```
history [n]  n为数字，列出最近的n条命令
```

常用选项

选项	含义
<code>-c</code>	将目前shell中的所有history命令消除
<code>-a</code>	将目前新增的命令写入histfiles, 默认写入 <code>~/.bash_history</code>
<code>-r</code>	将histfiles内容读入到目前shell的history记忆中
<code>-w</code>	将目前history记忆的内容写入到histfiles

案例演示

```
# 将history的内容写入一个新的文件中
[root@localhost ~]# history -w histfiles.txt
# 清理所有的history记录，注意并不清空 ~/.bash_history 文件
[root@localhos t ~]# history -c
[root@localhost ~]# history
1  history
2  cat .bash_history
3  ping -c 3 baidu.com
4  history
# 这里是执行第三条命令的意思
```

```
[root@localhost ~]# !3
# 从最近的命令查到以`command`开头的命令执行
[root@localhost ~]# !ping
# 执行上一条命令
[root@localhost ~]# !!
```

help

显示命令的帮助信息

```
help [-dms] [内置命令]
```

常用选项

选项	含义
-d	输出每个主题的简短描述
-m	以伪 man 手册的格式显示使用方法
-s	为每一个匹配 PATTERN 模式的主题仅显示一个用法

案例演示

```
# 查看echo的帮助信息
[root@localhost ~]# help echo
```

man

显示在线帮助手册页

```
man 需要帮助的命令或者文件
```

快捷键

按键	用途
空格键	向下翻一页
PaGe down	向下翻一页
PaGe up	向上翻一页
home	直接前往首页
end	直接前往尾页

按键	用途
/	从上至下搜索某个关键词，如“/linux”
?	从下至上搜索某个关键词，如“?linux”
n	定位到下一个搜索到的关键词
N	定位到上一个搜索到的关键词
q	退出帮助文档

手册的结构

结构名称	代表意义
NAME	命令的名称
SYNOPSIS	参数的大致使用方法
DESCRIPTION	介绍说明
EXAMPLES	演示（附带简单说明）
OVERVIEW	概述
DEFAULTS	默认的功能
OPTIONS	具体的可用选项（带介绍）
ENVIRONMENT	环境变量
FILES	用到的文件
SEE ALSO	相关的资料
HISTORY	维护历史与联系方式

alias

用于设置指令的别名

案例演示

```
# 查看系统当前的别名
[root@localhost ~]# alias
alias cp='cp -i'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias mv='mv -i'
alias rm='rm -i'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --
```

```
show-tilde'
[root@localhost ~]# ll
总用量 4
-rw-----. 1 root root 1241 8月 22 2018 anaconda-ks.cfg
drwxr-xr-x. 2 root root 19 8月 21 12:15 home
# 查看命令类型
# ls 是 `ls --color=auto' 的别名
# ls 是 /usr/bin/ls
[root@xwz ~]# type -a ls
# 修改别名, 比如使用wl来查看IP地址相关信息
[root@localhost ~]# alias wl='ip address'
[root@localhost ~]# wl
# 为了让别名永久生效, 可以将修改别名的命令写入 bashrc 文件, 这个文件中的命令会在每次登陆
命令行的时候执行
[root@localhost ~]# echo "alias wl='ip address'" >> /etc/bashrc
```