

Shell 能做什么？

1. 自动化批量系统初始化程序（update, 软件安装, 时区设置, 安全策略...）
2. 自动化批量软件部署程序（LAMP, LNMP, Tomcat, LVS, Nginx）
3. 应用管理程序（KVM, 集群管理扩容, MySQL, DELL R720 批量 RAID）
4. 日志分析处理程序（PV, UV, 200, !200, top 100, grep/awk）
5. 自动化备份恢复程序（MySQL 完全备份/增量 + Crond）
6. 自动化管理程序（批量远程修改密码, 软件升级, 配置更新）
7. 自动化信息采集及监控程序（收集系统/应用状态信息 CPU, Mem, Disk, Net, TCP Status, Apache, MySQL）
8. 配合 Zabbix 信息采集（收集系统/应用状态信息 CPU, Mem, Disk, Net, TCP Status, Apache, MySQL）
9. 自动化扩容（增加云主机——>业务上线）zabbix 监控 CPU 80%+|-50% Python API AWS/EC2（增加/删除云主机）+ Shell Script（业务上线）
10. 俄罗斯方块, 打印三角形, 打印圣诞树, 打印五角星, 运行小火车, 坦克大战, 排序算法实现
11. Shell 可以做任何事（一切取决于业务需求）

程序语言执行：

C —— 编译 —— 二进制机器码（CPU X86 复杂指令集, Power 精简指令集, APM）

Java —— 编译 —— 字节码（Java 虚拟机 JDK） Tomcat/Hadoop

shell —— 解释（/usr/bin/bash）

perl —— 解释（/usr/bin/perl）

expect —— 解释（/usr/bin/expect）

Python —— 解释（/usr/bin/python）

—— 编译 —— 字节码（Python 虚拟机）

程序是由哪些组程序： 逻辑 + 数据

BASH SHELL1

=====

一、SHELL 技术

1. shell 特性
2. shell 变量
3. shell 条件测试
4. shell 数值运算
5. 流控、循环

if

case

for

while

until

break

continue

exit

shift

array

function

6. 企业项目

二、shell? 命令解释器

shell 命令 cp ls date

Linux 支持的 shell

cat /etc/shells

chsh -l

三、GNU/bash shell 特点

1. 命令和文件自动补齐
2. 命令历史记忆功能 上下键、!number、!string、!\$、!!、^R
3. 别名功能 alias、unalias cp、~username/.bashrc、\cp -rf /etc/hosts .
4. 快捷键 ^R、^D、^A、^E、^L、^U、^K、^Y、^S、^Q
5. 前后台作业控制 &、nohup、^C、^Z、bg %1、fg %1、kill %3、screen
6. 输入输出重定向 0,1,2> >> 2> 2>> 2>&1 &> cat < /etc/hosts cat <<EOF cat >file1 <<EOF

[root@tianyun ~]# ll /dev/std*

lrwxrwxrwx 1 root root 15 Sep 1 2015 /dev/stderr -> /proc/self/fd/2

lrwxrwxrwx 1 root root 15 Sep 1 2015 /dev/stdin -> /proc/self/fd/0

lrwxrwxrwx 1 root root 15 Sep 1 2015 /dev/stdout -> /proc/self/fd/1

7. 管道 | tee

ip addr |grep 'inet ' |grep eth0

ip addr |grep 'inet ' |tee test |grep eth0 覆盖

ip addr |grep 'inet ' |tee -a test |grep eth0 -a 追加

df |grep '/\$'

df |tee df.txt |grep '/\$'

[root@tianyun ~]# date > date.txt

[root@tianyun ~]# date |tee date.txt

Fri Aug 25 15:30:20 CST 2017

8. 命令排序

; 不具备逻辑判断

cd; eject

&& || 具备逻辑判断

./configure && make && make install （命令返回值 echo \$?）

mkdir /var/111/222/333 && echo ok

mkdir -p /var/111/222/333 && echo ok

ls /home/111/222/333/444 || mkdir -p /home/111/222/333/444

[-d /home/111/222/333/444] || mkdir -p /home/111/222/333/444

ping -c1 10.18.42.1 &>/dev/null && echo up || echo down

=====

注意:

command & 后台执行

command &>/dev/null 混合重定向（标准输出 1，错误输出 2）

command1 && command2 命令排序，逻辑判断

=====

9. shell 通配符（元字符）表示的不是本意

* 匹配任意多个字符 ls in* rm -rf * rm -rf *.pdf find / -iname "*-eth0"

? 匹配任意一个字符 touch love loove live l7ve; ll l?ve

[] 匹配括号中任意一个字符 [abc] [a-z] [0-9] [a-zA-Z0-9] [^a-zA-Z0-9] ll l[io]ve ll l[^a-z]ve ll /dev/sd[a-z]

() 在子 shell 中执行(cd /boot;ls) (umask 077; touch file1000)

{ } 集合 touch file{1..9}

mkdir /home/{111,222} mkdir -pv /home/{333/{aaa,bbb},444}

cp -rv /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth0.old

cp -rv /etc/sysconfig/network-scripts/{ifcfg-eth0,ifcfg-eth0.old}

cp -rv /etc/sysconfig/network-scripts/ifcfg-eth0{,.old}

\ 转义符，让元字符回归本意

```
# echo *
```

```
# echo \*
```

```
# touch yang\sheng
```

```
mkdir \
```

```
echo -e "atb"
```

```
echo -e "a\tb"
```

```
echo -e "anb"
```

```
echo -e "a\nb"
```

10. 变量

11. shell script 脚本

小知识：

echo 输出带颜色文本

printf 格式化输出文本