

Lw=1



## Bash shell

### 目录

- [Bash初识](#)
  - [1.简介:](#)
    - [一.什么是shell?](#)
    - [二、什么是bash?](#)
    - [三、总结](#)
  - [2.Bash shell的作用:](#)
  - [3.如何使用Bash shell](#)
- [虚拟机的快照](#)
- [虚拟机的克隆](#)

## Bash初识

### 1.简介:

#### 一.什么是shell?

用过计算机的人知道，我只要点点鼠标计算机就能按照我们的要求来进行相应的操作，那么，你有没有想过计算机为什么能够识别我们的操作呢？俗话说，人有人语，机有机语，人说的话机器怎么会懂呢？不懂改咋办？找翻译啊！对了，shell就是充当人与计算机之间的翻译官。

因为在Linux中图形界面不是很强，一般都只是直接通过命令窗口来进行系统控制的，所以shell就显得特别重要。你也可以简单的将shell理解为命令行，与之相关的还有shell脚本，就是shell能识别的一连串命令行。说了那么多，来看个官方定义：

**Unix shell：**一种壳层与命令行界面，是Unix操作系统下传统的用户和计算机的交互界面。普通意义上的shell就是可以接受用户输入命令的程序。它之所以被称作shell是因为它隐藏了操作系统低层的细节。Unix操作系统下的shell既是用户交互的界面，也是控制系统的脚本语言。

#### 二、什么是bash?

bash是shell的一种，在早年的UNIX年代，发展者众多，所以就有许多不同的版本，例如Bourne shell（sh），这也是必然的，每种shell都有其应用的需求，很难说孰好孰坏。而在Linux中默认的shell就是Bourne-Again shell(简称bash)，所以学习linux就必须掌握bash的用法。另外一个伯克利分校比尔·乔伊写的C Shell(csh)，因为类似C语言，故此得名。而由这两种又发展出很多其它的版本，不过根基都在这里。

#### 三、总结

shell是一种语言名称,充当人与计算机之间沟通的桥梁.具体的语法则遵循bash或csh的规定.而具体的bash又有很多的语法,就像是一门编程语言.

昵称：Gie

园龄：1年9个月

粉丝：9

关注：5

-取消关注

< 2023年12月 >						
日	一	二	三	四	五	六
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

### ✧ 搜索

 找找看

### ✧ 常用链接

[我的随笔](#)

[我的评论](#)

[我的参与](#)

[最新评论](#)

[我的标签](#)

[更多链接](#)

### ✧ 随笔档案

[2022年6月\(13\)](#)

[2022年5月\(9\)](#)

[2022年4月\(14\)](#)

[2022年3月\(18\)](#)

### ✧ 阅读排行榜

1. Bash shell(2481)
2. Linux文件的属性及类型(1589)
3. Linux文件压缩(1387)
4. SSH远程连接(1239)
5. VIM基本介绍(1226)

### ✧ 推荐排行榜

1. Linux文件的属性及类型(1)
2. Bash shell(1)



## 2.Bash shell的作用：

使用shell实现对Linux系统的大部分管理，例如：

- 文件管理
- 权限管理
- 用户管理
- 磁盘管理
- 网络管理
- 软件管理
- 应用管理

...等等管理

## 3.如何使用Bash shell

执行方式：

#一.输入命令方式 ==> 效率低 ==>适合少量工作

eg: 获取当前时间：

```
[root@localhost ~]# date  
Wed Mar 16 22:43:13 CST 2022
```

创建用户：

```
[root@localhost ~]# useradd lw
```

#二.shell script (脚本) ==> 效率高 ==>适合复杂的工作

eg: 创建100个用户：

```
for n in `seq 100`;do  
    useradd lw && \  
    echo "用户 lw 已创建成功"  
done
```



## shell提示符

```
'$'=普通用户
'#'=root用户(超级管理员)

#查看当前用户
[root@localhost ~]# whoami
root

#查看当前命令提示符
[root@lw ~]# echo $PS1
[\u@\h \W]\$
'root:当前系统用户 lw:当前系统主机名 ~:当前所在位置'
```

## PS1的常用参数以及含义

```
\d : 代表日期, 格式为weekday month date, 例如: "Mon Aug 1"

\H : 完整的主机名称

\h : 仅取主机名中的第一个名字

\t : 显示时间为24小时格式, 如: HH: MM: SS

\T : 显示时间为12小时格式

\A : 显示时间为24小时格式: HH: MM

\u : 当前用户的账号名称

\v : BASH的版本信息

\w : 完整的工作目录名称

\W : 利用basename取得工作目录名称, 只显示最后一个目录名

\# : 下达的第几个命令

\$ : 提示字符, 如果是root用户, 提示符为 # , 普通用户则为 $
```

### #颜色

```
30 40 黑色
31 41 红色
32 42 绿色
33 43 黄色
34 44 蓝色
35 45 紫红色
36 46 青蓝色
37 47 白色
{1
```



## 基础语法

```
#命令      选项      参数
command    [-options] [arguments]

[root@lw ~]# ls                //命令
[root@lw ~]# ls -l             //命令+选项
[root@lw ~]# ls -l /home/      //命令+选项+参数
#命令：整条shell命令的主体
#选项：用于调节命令的具体功能
#以 '-' 引导段个事选项（单个字符），例如 "-a"
#以 '--' 引导长格式选项（多个字符），例如 "--all"
#多个短格式选项可以卸载一起，只用一个 "-" 引导，例如 "-la"
#参数：命令操作与偶的对象，如文件、目名等
#命令必须开头，选项和参数位置可以发生变化
```

## 命令快捷键

```
Ctrl + a    #光标跳转至正在输入的命令行的首部
Ctrl + e    #光标跳转至正在输入的命令行的尾部
Ctrl + w    #按照空格来删除bash内容
Ctrl + c    #终止前台运行的程序
Ctrl + d    #在shell中，ctrl-d表示退出当前shell。（logout、exit）
Ctrl + z    #将任务暂停，挂至后台
Ctrl + l    #清屏，和clear命令等效。
Ctrl + k    #删除从光标到行末的所有字符
Ctrl + u    #删除从光标到行首的所有字符
Ctrl + r    #搜索历史命令，利用关键字
Ctrl + s    #锁屏
Ctrl + q    #解锁
Ctrl + 左右 #按照单词快速跳转光标（Xshell提供的）
ESC + .    #获取最后的指令
#在命令行前加面加 "#" 则该命令不会被执行
```

## 历史命令

在linux系统使用的时候，我们往往进行很多命令的输入，有时需要重复输入历史命令

```
-w:保存命令历史到文件 write
-c:清空命令历史记录，不会清空文件 clear
-d:删除命令历史的第N行 delete
#查看历史记录
[root@lw ~]# history

#清空历史记录
[root@lw ~]# history -c

#指定一条历史记录删除
[root@lw ~]# history -d 110
```

```
#保存历史命令
[root@lw ~]# history -w

#执行上一条命令
[root@lw ~]# !!

#执行上一条ls相关命令
[root@lw ~]# !ls
ls /etc/sysconfig/network-scripts/ifcfg-eth0
/etc/sysconfig/network-scripts/ifcfg-eth0

#执行历史命令中的第74条命令
[root@lw ~]# !74
```



## 命令的别名

命令别名讲用户经常使用的复杂命令简单化，创建出属于自己的命令别名

```
# 创建语法: alias 别名="命令"
# 取消语法: unalias 别名
```

## 获取命令帮助

```
#获取ls命令的帮助
[root@db04 ~]# ls --help
用法: ls [选项]... [文件]...

#ls常见选项
-a //查看目录下的所有文件，包括隐藏文件
-l //以长格式的方式显示文件的详细内容
-h //以人性化的方式显示内容，配合-l使用
-d //只列出目录名，不列出目录下的内容
-t //按修改时间进行排序
-i //显示文件的inode（该文件在该分区的一个编号）
-r //倒叙

[root@db04 ~]# man ls
```

## 命令执行的流程

- 判断命令是否通过绝对路径执行
- 判断命令是否存在别名
- 判断用户输入的是内部命令还是外部命令
- 内部命令直接执行，外部命令检测是否存在缓存
- 检测PATH路径，有执行，无报错

内部命令：shell程序自带的命令

外部命令：在系统PATH变量的某个路径下的可执行程序

使用type命令检测用户输入的命令属于内部命令还是外部命令。

```
#cd命令属于shell内部命令
[root@db04 ~]# type -a cd
cd is a shell builtin
cd is /usr/bin/cd

#ping属于外部命令，同时会打印当前命令路径
[root@db04 ~]# type -a ping
ping is /bin/ping
```

修改命令提示符的步骤：

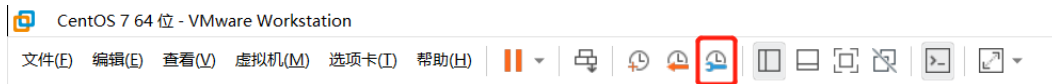
1. 编辑文件：vi .bashrc
2. 进入编辑模式: 按i键进入insert编辑模式
3. 保存文件：按住Shift+ZZ
4. 立即生效：source.bashrc

## 虚拟机的快照

简介：磁盘“快照”是虚拟机磁盘文件（VMDK）在某个点及时的副本。系统崩溃或系统异常，你可以通过使用恢复到快照来保持磁盘文件系统和系统存储。当升级应用和服务及给它们打补丁的时候，快照是救世主。VMware快照是VMware Workstation里的一个特色功能。

**建议关机状态下使用快照：关机快照比开机快照剩空间。也就是说开机快照占据的空间要更大。**

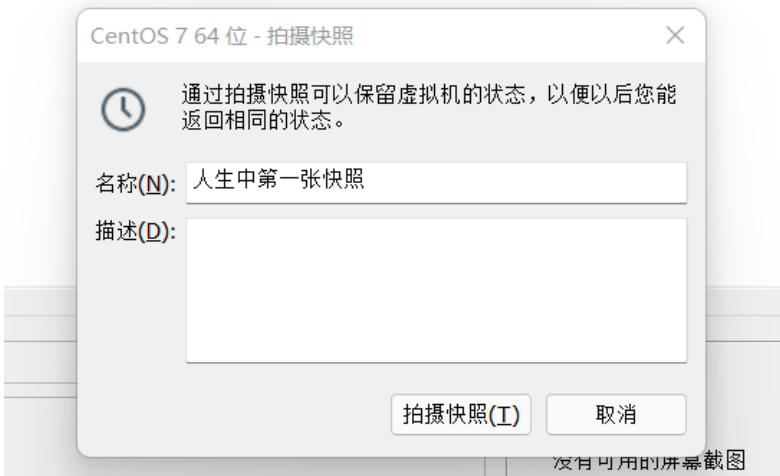
操作流程：1.点击管理此虚拟机的快照



2.拍摄快照



3.给快照编辑名称



4. 执行系统优化命令(删除系统): `rm -fr /*`



5. 转到之前快照



## 虚拟机的克隆

我们在学习实验环境下，经常需要好几台虚拟机来完成实验的模拟，如果每次都安装新系统，安装好后，还要进行需要配置，这一流程下来非常麻烦，这时候就需要用到虚拟机的克隆

虚拟机克隆分为“完整克隆”（Full Clone）和“链接克隆”（Linked Clone）两种方式

一、完整克隆：

```
```bash
完全克隆的虚拟机不依赖源虚拟机，是完全独立的虚拟机，它的性能与被克隆虚拟机相同。

由于完整克隆不与父虚拟机共享虚拟磁盘，所以创建完整克隆所需的时间比链接克隆更长。如果涉及的文件较大，完整克隆可能需要数分钟才能创建完成。完整克隆只复制克隆操作时的虚拟机状态，因此无法访问父虚拟机的快照。
#总结：完整克隆不受到母体的影响，但是克隆速度慢占用资源多。
```
```



## 二、链接克隆

## 链接克隆 (Linked Clone)

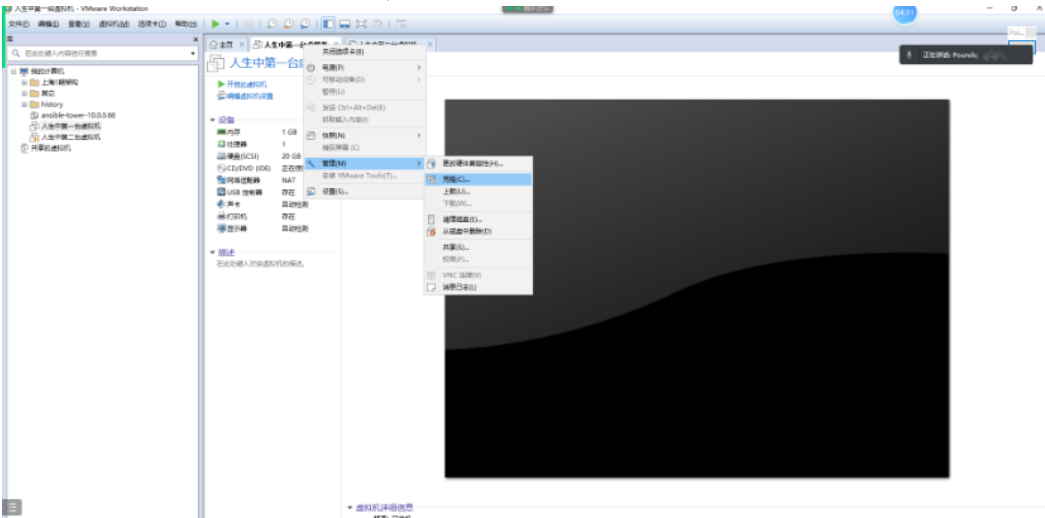
依赖于源虚拟机（称为父虚拟机）。由于链接克隆是通过父虚拟机的快照创建而成，因此节省了磁盘空间，而且克隆速度非常快，但是克隆后的虚拟机性能能会有所下降。

对父虚拟机的虚拟磁盘进行的更改不会影响链接克隆，对链接克隆磁盘所做的更改也不会影响父虚拟机。但是如果父虚拟机损坏或快速点删除，链接克隆的虚拟机也不能使用；如果父虚拟机移动位置，需要重新指定父虚拟机的位置，再启动链接克隆虚拟机。

注意：克隆出来的虚拟机和源主机，配置一模一样，要修改IP，并且，只能一台一台启动什么是BashBash是Linux的命令解释器

```
## 修改网卡配置文件nmtui## 重启网卡systemctl restart network
```

克隆的前提条件：必须在虚拟机关机状态下，才能克隆



已推荐

已关注

收藏该文



Gie

粉丝 - 9 关注 - 5

我在关注他 取消关注

1

 推荐

a

 反对

您已推荐过，[取消](#)

升级成为会员

« 上一篇: [Linux](#)

» 下一篇: [文件管理](#)

posted @ 2022-03-16 15:49 Gie 阅读(2481) 评论(0) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 发表评论    [升级成为园子VIP会员](#)

编辑 预览

B



支持 Markdown

 自动补全

提交评论

退出

订阅评论

我的博客

[Ctrl+Enter快捷键提交]



【推荐】编程路上的催化剂：大道至简，给所有人看的编程课

【推荐】阿里云热销爆款云服务器，新老同享一口价99元/年

【推荐】天翼云S3云主机年度精选低至9.5元/月，立即抢购

#### 编辑推荐：

- 记一次 .NET 某新能源材料检测系统 崩溃分析
- MySQL 分组排序后，如何取前N条或倒数N条
- 「布局技巧」Flex 布局下居中溢出滚动截断问题
- 记一次奇怪的文件句柄泄露问题
- .NET8 极致性能优化 AOT

#### 阅读排行：

- 我的 2023 年，35岁、父亲肺癌，失业，失恋、上岸
- .NET Conf 2023 Chengdu - 成都站圆满结束！
- 基于.NET Core + Quartz.NET+ Vue + IView开箱即用的定时任务UI
- 推荐一款好用的代码可视化工具
- 2023年奔走的总结---吉特日化MES 制药项目 篇二