

Day2

Vi/Vim编辑器、系统
管理操作、rpm、
yum

1813075 刘茵

一、 虚拟机

- 1. 安装虚拟机
- 2. [在虚拟机中安装Centos7.0](#)

二、 Linux

- 1. Linux 简介：
Linux 是一套免费使用和自由传播的类 Unix 操作系统

比较	Windows	Linux
界面	界面统一，外壳程序固定所有 Windows 程序菜单几乎一致，快捷键也几乎相同	图形界面风格依发布版不同而不同，可能互不兼容。GNU/Linux 的终端机是从 UNIX 传承下来，基本命令和操作方法也几乎一致。
驱动程序	驱动程序丰富，版本更新频繁。默认安装程序里面一般包含有该版本发布时流行的硬件驱动程序，之后所出的新硬件驱动依赖于硬件厂商提供。对于一些老硬件，如果没有了原配的驱动有时很难支持。另外，有时硬件厂商未提供所需版本的 Windows 下的驱动，也会比较头痛。	由志愿者开发，由 Linux 核心开发小组发布，很多硬件厂商基于版权考虑并未提供驱动程序，尽管多数无需手动安装，但是涉及安装则相对复杂，使得新用户面对驱动程序问题（是否存在和安装方法）会一筹莫展。但是在开源开发模式下，许多老硬件尽管在Windows下很难支持的也容易找到驱动。HP、Intel、AMD 等硬件厂商逐步不同程度支持开源驱动，问题正在得到缓解。
使用	使用比较简单，容易入门。图形化界面对没有计算机背景知识的用户使用十分有利。	图形界面使用简单，容易入门。文字界面，需要学习才能掌握。
学习	系统构造复杂、变化频繁，且知识、技能淘汰快，深入学习困难。	系统构造简单、稳定，且知识、技能传承性好，深入学习相对容易。
软件	每一种特定功能可能都需要商业软件的支持，需要购买相应的授权。	大部分软件都可以自由获取，同样功能的软件选择较少。

- 2. 虚拟机内安装 linux
- 3. Linux 文档目录结构：

在 Linux 或 Unix 操作系统中，所有的文件和目录都被组织成以一个根节点开始的倒置的树状结构。

文件系统的最顶层是由根目录开始的，系统使用 / 来表示根目录。在根目录之下的既可以是目录，也可以是文件，而每一个目录中又可以包含子目录文件。如此反复就可以构成一个庞大的文件系统。

在 Linux 文件系统中有两个特殊的目录，一个用户所在的工作目录，也叫当前目录，可以使用一个点 . 来表示；另一个是当前目录的上一级目录，也叫父目录，可以使用两个点 .. 来表示。

. : 代表当前的目录, 也可以使用 **/** 来表示;

.. : 代表上一层目录, 也可以 **../** 来代表。

如果一个目录或文件名以一个点 **.** 开始, 表示这个目录或文件是一个隐藏目录或文件(如: **.bashrc**)。即以默认方式查找时, 不显示该目录或文件。

系统启动必须:

/boot: 存放的启动 Linux 时使用的内核文件, 包括连接文件以及镜像文件。

/etc: 存放所有的系统需要的**配置文件和子目录列表**, 更改目录下的文件可能会导致系统不能启动。

/lib: 存放基本代码库(比如 **c++**库), 其作用类似于 Windows 里的 DLL 文件。几乎所有的应用程序都需要用到这些共享库。

/sys: 这是 linux2.6 内核的一个很大的变化。该目录下安装了 2.6 内核中新出现的一个文件系统 **sysfs**。**sysfs** 文件系统集成了下面 3 种文件系统的信息: 针对进程信息的 **proc** 文件系统、针对设备的 **devfs** 文件系统以及针对伪终端的 **devpts** 文件系统。该文件系统是内核设备树的一个直观反映。当一个内核对象被创建的时候, 对应的文件和目录也在内核对象子系统中

指令集合:

/bin: 存放着最常用的程序和指令

/sbin: 只有系统管理员能使用的程序和指令。

外部文件管理:

/dev : Device(设备)的缩写, 存放的是 Linux 的外部设备。**注意:** 在 Linux 中访问设备和访问文件的方式是相同的。

/media: 类 windows 的**其他设备**, 例如 U 盘、光驱等等, 识别后 linux 会把设备放到这个目录下。

/mnt: 临时挂载别的文件系统的, 我们可以将光驱挂载在 **/mnt/**上, 然后进入该目录就可以查看光驱里的内容了。

临时文件:

/run: 是一个临时文件系统, 存储系统启动以来的信息。当系统重启时, 这个目录下的文件应该被删掉或清除。如果你的系统上有 **/var/run** 目录, 应该让它指向 **run**。

/lost+found: 一般情况下为空的, 系统非法关机后, 这里就存放一些文件。

/tmp: 这个目录是用来存放一些临时文件的。

账户:

/root: 系统管理员的用户主目录。

/home: 用户的主目录, 以用户的账号命名的。

/usr: 用户的很多应用程序和文件都放在这个目录下, 类似于 windows 下的 **program files** 目录。

/usr/bin: 系统用户使用的应用程序与指令。

/usr/sbin: 超级用户使用的比较高级的管理程序和系统守护程序。

/usr/src: 内核源代码默认的放置目录。

运行过程中要用:

/var: 存放经常修改的数据, 比如程序运行的日志文件 (/var/log 目录下)。

/proc: 管理**内存空间**! 虚拟的目录, 是系统内存的映射, 我们可以直接访问这个目录来, 获取系统信息。这个目录的内容不在硬盘上而是在内存里, 我们也可以直接修改里面的某些文件来做修改。

扩展用的:

/opt: 默认是空的, 我们安装额外软件可以放在这个里面。

/srv: 存放服务启动后需要提取的数据 (不用服务器就是空)

4. Linux 问卷基本属性

Linux的权限不是很细致, 只有 RWX 三种:

- r(Read, 读取): 对**文件**而言, 具有**读取文件内容**的权限; 对目录来说, 具有**浏览**目录信息的权限。
- w(Write, 写入): 对**文件**而言, 具有**新增, 修改 文件内容**的权限(但不含删除该文件); 对目录来说, 具有新建, 删除, 修改, 移动目录内文件的权限。
- x(eXecute, 执行): 对**文件**而言, 具有**执行文件**的权限; 对**目录**来说该用户具有**进入目录**的权限。

文件 类型	属主 权限			属组 权限			其他用户 权限		
0	1	2	3	4	5	6	7	8	9
d	rwX			r-X			r-X		
目录 文件	读	写	执行	读	写	执行	读	写	执行

文件类型:

当为[**d**]则是目录

当为[**-**]则是文件;

若是[**l**]则表示为链接文档(link file);

若是[**b**]则表示为装置文件里面的可供储存的接口设备(可随机存取装置);

若是[**c**]则表示为装置文件里面的串行端口设备, 例如键盘、鼠标(一次性读取装置)。

第 0 位确定文件类型, 第 1-3 位确定属主 (该文件的所有者) 拥有该文件的权限。

第 4-6 位确定属组 (所有者的同组用户) 拥有该文件的权限, 第 7-9 位确定其他用户拥有该文件的权限。

其中, 第 1、4、7 位表示读权限, 如果用"r"字符表示, 则有读权限, 如果用 "-" 字符表示, 则没有读权限;

第 2、5、8 位表示写权限，如果用"w"字符表示，则有写权限，如果用 "-" 字符表示没有写权限；第 3、6、9 位表示可执行权限，如果用"x"字符表示，则有执行权限，如果用 "-" 字符表示，则没有执行权限。

5. Linux 常用基本命令

1) 创建文件

touch+file

2) 文件目录

ls: 列出目录及文件名

cd: 切换目录

`cd ..` 表示去到目前的上一级目录

`cd ~` 表示返回家目录

pwd: 显示目前的目录

mkdir: 创建一个新的目录

`mkdir -p test1/test2/test3/test4` 创建多重目录

rmdir: 删除一个空的目录

`rmdir -p test1/test2/test3/test4` 删除多重目录

cp: 复制文件或目录

用 root 身份，将 root 目录下的 .bashrc 复制到 /tmp 下，并命名为 bashrc

`cp ~/.bashrc /tmp/bashrc`

rm: 移除文件或目录

-f : 就是 force 的意思，忽略不存在的文件，不会出现警告信息；

-i : 互动模式，在删除前会询问使用者是否动作

-r : 递归删除啊！最常用在目录的删除了！这是非常危险的选项！！

mv: 移动文件与目录，或修改文件与目录的名称

可以使用 **man [命令]** 来查看各个命令的使用文档，如：**man cp**。

3) 更改属性等

chgrp: 更改文件属组

chown: 更改文件属主，也可以同时更改文件属组

进入 /root 目录 (~) 将 install.log 的拥有者改为 bin 这个账号：`# chown bin install.log`

4) linux 文件内容查看

cat 由第一行开始显示文件内容

tac 从最后一行开始显示，可以看出 tac 是 cat 的倒着写！

nl 显示的时候，顺道输出行号！

more 一页一页的显示文件内容

在 **more** 这个程序的运行过程中，你有几个按键可以按的：

空白键 (**space**): 代表向下翻一页；

Enter : 代表向下翻『一行』；

/字符串 : 代表在这个显示的内容当中，向下搜寻『字符串』这个关键字；

:f : 立刻显示出档名以及目前显示的行数；

q : 代表立刻离开 **more**，不再显示该文件内容。

b 或 **[ctrl]-b** : 代表往回翻页，不过这动作只对文件有用，对管线无用。

less 与 **more** 类似，但是比 **more** 更好的是，他可以往前翻页！

head 只看头几行

`head -n 20 /<file>` 显示文件的前 20 行

tail 只看尾巴几行

6. Linux 用户账号的管理

- 添加账户 `useradd` 选项 用户名
- 删除账户 `userdel` 选项 用户名: 常用的选项是 `-r`，它的作用是把用户的主目录一起删除。
- 修改账号 `usermod` 选项 用户名
- 修改口令 `passwd` : 锁定某一用户，使其不能登录# `passwd -l sam`

三、VI/VIM编辑器

vi 是老式的字处理器，不过功能已经很齐全了，但是还是有可以进步的地方。**vim** 则可以说是程序开发者的一项很好用的工具。

version 1.1
April 1st, 06
翻译: 2006-5-21

vi / vim 键盘图

Esc
命令
模式

~ 转换大小写 \ 跳转到标注	! 外部过滤器	@ 运行宏	# prev ident	\$ 行尾	% 括号匹配	^ "软"行首	& 重复 :s	* next ident	(句首) 下一句首	"soft" bol down	+ 后一行首
1	2	3	4	5	6	7	8	9	0	"硬"行首	- 前一行首	= 自动格式 ³
Q 切换至 ex 模式	W 下一单词	E 词尾	R 替换模式	T back 'till	Y 拷贝行	U 撤销命令	I 到行首插入	O 分段(前)	P 粘贴(前)	{ 段首	}	段尾
q 录制宏	w 下一单词	e 词尾	r 替换字符	t 'till	y 拷贝 ^{1,3}	u 撤销命令	i 插入模式	o 分段(后)	p 粘贴(后) ¹	. 杂项	.	杂项
A 在行尾附加	S 删除行并插入	D 删除至行尾	F 行内字符反向查找	G 文尾/行号	H 屏幕顶行	J 合并两行	K 帮助	L 屏幕底行	: ex 命令	" 寄存器 ¹ 标识	行首/列	
a 附加	s 删除字符并插入	d 删除 ^{1,3}	f 行内字符查找	g 附加命令 ⁶	h ←	j ↓	k ↑	l →	; 重复 :v/T/f/F	' 跳转到标注的行首	\ 未用!	
Z 退出 ⁴	X 退格	C 修改至行末	V 可视行模式	B 前一单词	N 查找上一处	M 屏幕中间行	< 反缩进 ³	> 缩进 ³	? 向前搜索			
Z 附加命令 ⁵	x 删除(字符)	c 修改 ^{1,3}	v 可视模式	b 前一单词	n 查找下一处	m 设置标注	, 反向 :v/T/f/F	.	重复命令	/ 向后搜索		

动作 移动光标, 或者定义操作的范围

命令 直接执行的命令,
红色命令 进入编辑模式

操作 后面跟随表示操作范围的指令

extra 特殊功能,
需要额外的输入

q· 后跟字符参数

w,e,b 命令

小写(b): quux(foo, bar, baz);
大写(B): QUUX(FOO, BAR, BAZ);

主要 ex 命令:

:w (保存), :q (退出), :q! (不保存退出)
:ef (打开文件 f),
:%s/x/y/g ('y' 全局替换 'x'),
:h (帮助 in vim), :new (新建文件 in vim),

其它重要命令:

CTRL-R: 重复 (vim),
CTRL-F/-B: 上翻/下翻,
CTRL-E/-Y: 上滚/下滚,
CTRL-V: 块可视模式 (vim only)

可视模式:

漫游后对选中的区域执行操作 (vim only)

备注:

(1) 在 拷贝/粘贴/删除 命令前使用 "x (x=a..z,*)" 使用命令的寄存器("剪贴板")
(如: "ay\$ 拷贝剩余的行内容至寄存器 'a')

(2) 命令前添加数字
多遍重复操作
(e.g.: 2p, d2w, 5i, d4j)

(3) 重复本字符在光标所在行执行操作
(dd = 删除本行, >> = 行首缩进)

(4) ZZ 保存退出, ZQ 不保存退出

(5) zt: 移动光标所在行至屏幕顶端,
zb: 底端, zz: 中间

(6) gg: 文首 (vim only),
gf: 打开光标处的文件名 (vim only)

原图: www.viemu.com 翻译: fdl (linuxsir)

1. VI/VIM 的使用

基本上 vi/vim 共分为三种模式, 分别是命令模式 (Command mode), 输入模式 (Insert mode) 和底线命令模式 (Last line mode)

命令模式:

用户刚刚启动 vi/vim, 便进入了命令模式。

此状态下敲击键盘动作会被 Vim 识别为命令, 而非输入字符。比如我们此时按下 i, 并不会输入一个字符, i 被当作了一个命令。

以下是常用的几个命令:

- i 切换到输入模式, 以输入字符。
- x 删除当前光标所在处的字符。
- : 切换到底线命令模式, 以在最底一行输入命令。

输入模式:

在命令模式下按下 i 就进入了输入模式。

在输入模式中, 可以使用以下按键:

- 字符按键以及 Shift 组合, 输入字符
- ENTER, 回车键, 换行

- **BACK SPACE**，退格键，删除光标前一个字符

底线命令模式：

在命令模式下按下：（英文冒号）就进入了底线命令模式。

底线命令模式可以输入单个或多个字符的命令，可用的命令非常多。

在底线命令模式中，基本的命令有（已经省略了冒号）：

- q 退出程序
- w 保存文件

进入模式必须 **vi+<filename>**

2. vi/vim 按键说明

光标移动：↑ ↓ ← → 等

搜索替换：/word 光标向下查找 word ?word 光标向上查找 word

n 代表重复前一个搜寻的动作

N 为『反向』进行前一个搜寻动作

替换：

- 100 到 200 行之间搜寻 vbird 并取代为 VBIRD 则：
[:100,200s/vbird/VBIRD/g]
- :1,\$s/word1/word2/g 或 :%s/word1/word2/g 从第一行到最后一行寻找 word1 字符串，并将该字符串取代为 word2
- :1,\$s/word1/word2/gc 或 :%s/word1/word2/gc 从第一行到最后一行寻找 word1 字符串，并将该字符串取代为 word2 ！且在取代前显示提示字符给用户确认 (confirm) 是否需要取代！（常用）

删除：

- x, X: x 为向后删除一个字符，X 为向前
- nx: n 为数字，连续向后删除 n 个字符
- dd: 删除光标所在的那一整行(常用)

复制：

- yy: 复制光标所在的那一行(常用)
- J: 将光标所在行与下一行的数据结合成同一行
- u: 复原前一个动作
- [Ctrl]+r 重复上一个动作

四、Rpm

RPM 套件管理方式的出现，让 Linux 易于安装，升级，间接提升了 Linux 的适用度。

rpm 常见命令参数:

用法: rpm [选项...]

- -a: 查询所有套件;
- -b<完成阶段><套件档>+或-t <完成阶段><套件档>+: 设置包装套件的完成阶段, 并指定套件档的文件名称;
- -c: 只列出组态配置文件, 本参数需配合"-l"参数使用;
- -d: 只列出文本文件, 本参数需配合"-l"参数使用;
- -e<套件档>或--erase<套件档>: 删除指定的套件;
- -f<文件>+: 查询拥有指定文件的套件;
- -h 或--hash: 套件安装时列出标记;
- -i: 显示套件的相关信息;

如何安装 rpm 软件包:

```
rpm -ivh your-package
```

如何卸载 rpm 软件包:

```
[root@localhost ~]# rpm -ql tree      # 查询
[root@localhost ~]# rpm -e tree      # 卸载
[root@localhost ~]# rpm -ql tree      # 查询
```

列出所有安装的包# rpm -qa

五、[yum](#)

yum 是一个在 Fedora 和 RedHat 以及 SUSE 中的 Shell 前端软件包管理器。

yum 常用命令

1.列出所有可更新的软件清单命令: yum check-update

2.更新所有软件命令: yum update

3.仅安装指定的软件命令: yum install <package_name>

4.仅更新指定的软件命令: yum update <package_name>

5.列出所有可安装的软件清单命令: yum list

6.删除软件包命令: yum remove <package_name>

7.查找软件包 命令: yum search <keyword>

8.清除缓存命令:

yum clean packages: 清除缓存目录下的软件包

yum clean headers: 清除缓存目录下的 headers

yum clean oldheaders: 清除缓存目录下旧的 headers

yum clean, yum clean all (= yum clean packages; yum clean oldheaders) :清除
缓存目录下的软件包及旧的 headers