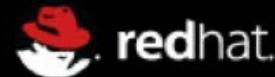


**RED HAT®
TRAINING**



Comprehensive, hands-on training that solves real-world problems

Red Hat System Administration I



东方瑞通 终身学习



第1章 访问命令行



关注微信公众号
享终身免费培训

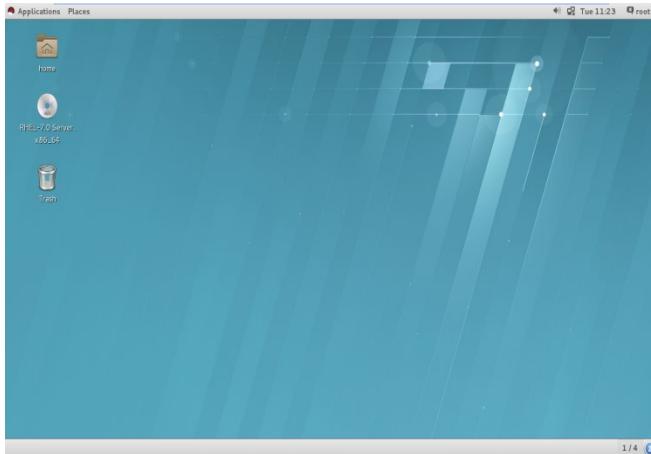
学习目标

- 登录Linux操作系统
- 认识 GNOME3 桌面环境
- 在Linux控制台输入命令
- 以便捷方式运行命令

两种登录方式

- 图形界面

- ◆ GNOME是RHEL默认环境
- ◆ 由X Window System提供图形框架



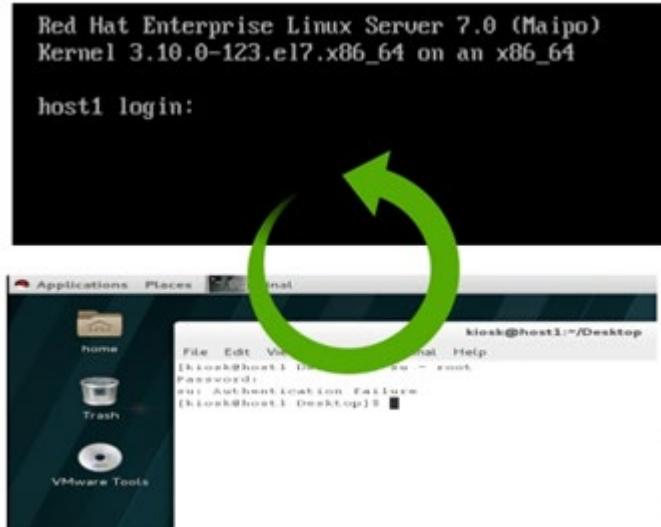
- 字符界面

```
Red Hat Enterprise Linux Server 7.4 (Maipo)
Kernel 3.10.0-693.el7.x86_64 on an x86_64

desktop0 login:
```

切换界面快捷键

- 图形界面切换字符界面
 - ◆ Ctrl+Alt+Fn (F2~F6)
 - ◆ 字符终端间切换
 - Alt+Fn
- 字符界面切换回图形界面
 - ◆ Ctrl+Alt+F1



命令行与Bash

- 命令行可以用来输入指令到计算机系统中基于文本的界面
- Linux 的命令行通过shell 程序提供，类似 UNIX系统
- RHEL系统默认使用 Bourne Again shell (bash)
- 以普通用户运行
 - ◆ [student@desktopX ~]\$
- 以超级用户运行
 - ◆ [root@desktopX ~]#

使用Bash Shell执行命令

- Linux命令：用于实现某一类功能的指令，执行依赖于shell程序
- 通用命令格式：
 - ◆ 命令 [选项] [参数]
 - 命令(command)：整条shell命令的主体
 - 选项(option)：调整命令的行为，通常以 -, -- 表示
 - 参数(arguments)：命令作用的对象
 - ◆ 例子：# ls -l /etc/hosts
- 取消命令执行
 - ◆ Ctrl+C

简单命令示例

```
[student@desktopX ~]$ date
Sat Apr  5 08:13:50 PDT 2014
[student@desktopX ~]$ date +%R
08:13
[student@desktopX ~]$ date +%x
04/05/2014
```

```
[student@desktopX ~]$ passwd
Changing password for user student.
Changing password for student.
(current) UNIX password: old_password
New password: new_password
Retype new password: new_password
passwd: all authentication tokens updated successfully.
```

Bash Shell使用技巧——Tab键

- Tab 键使用户能够快速完成命令输入
- 自动补全指定的命令或者目录、文件
- 如果输入的字符不唯一，需执行两次 Tab 键，显示指定输入字符的所有指令

```
[student@desktopX ~]$ pas<Tab><Tab>
passwd      paste      pasuspender
[student@desktopX ~]$ pass<Tab>
[student@desktopX ~]$ passwd
Changing password for user student.
Changing password for student.
(current) UNIX password:
```

```
[student@desktopX ~]$ ls /etc/pas<Tab>
[student@desktopX ~]$ ls /etc/passwd<Tab>
passwd    passwd-
```

Bash Shell使用技巧——历史记录

- 历史命令
 - ◆ 使用history显示命令记录
- 执行历史命令
 - ◆ 使用 !number 执行历史记录中指定编号命令
 - ◆ 使用 !string 执行记录中最后(最近)一条以指定字串起始的命令

Bash Shell 常用快捷键

| 快捷键 | 说明 |
|--------|--------------|
| ctrl+a | 光标跳至行首 |
| ctrl+e | 光标跳至行末 |
| ctrl+u | 从光标所在位置清空至行首 |
| ctrl+k | 从光标所在位置清空至行末 |
| ctrl+← | 光标向左跳一个单词 |
| ctrl+→ | 光标向右跳一个单词 |
| ctrl+w | 回删一个单词 |
| alt+d | 删除光标之后一个单词 |

Q&A



第2章 从命令行管理文本



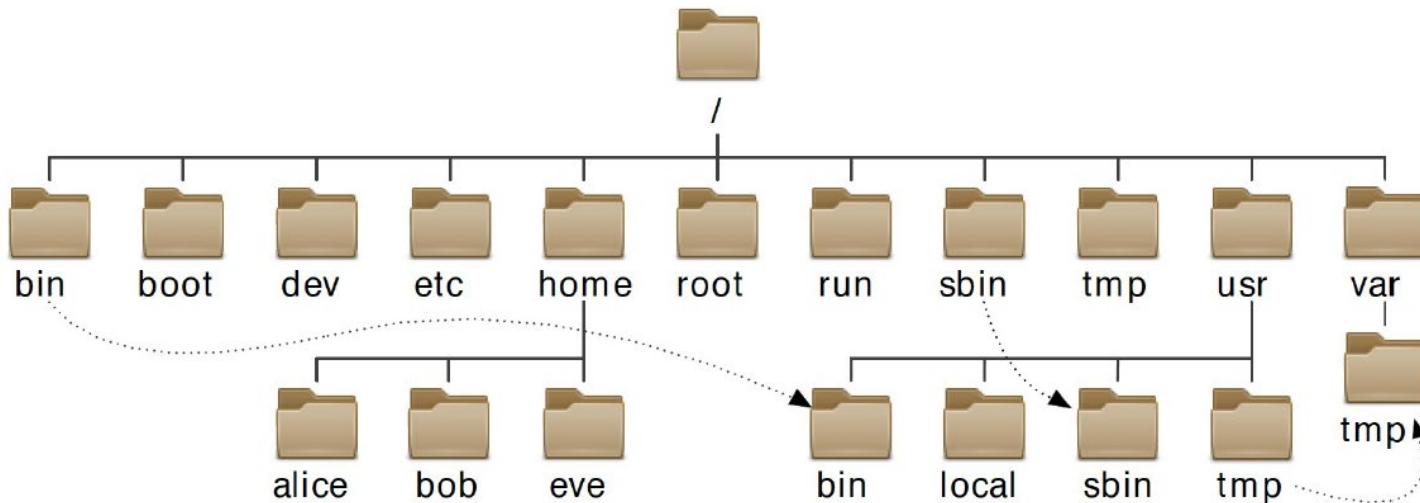
关注微信公众号
享终身免费培训

目标

- 认识Linux 系统目录结构
- 使用绝对路径和相对路径指定文件
- 使用命令行工具创建，复制，移动，删除文件和目录
- 使用 shell 中的通配符，匹配一个或多个文件

Linux 系统目录结构

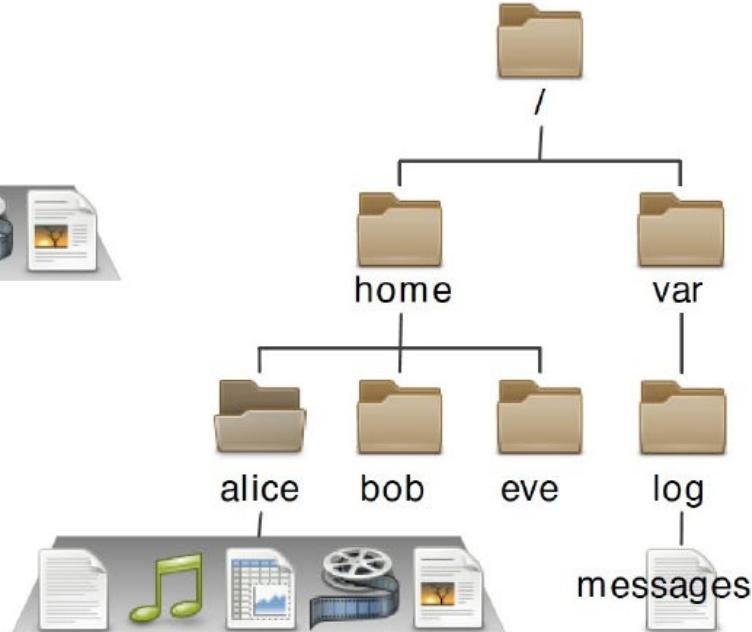
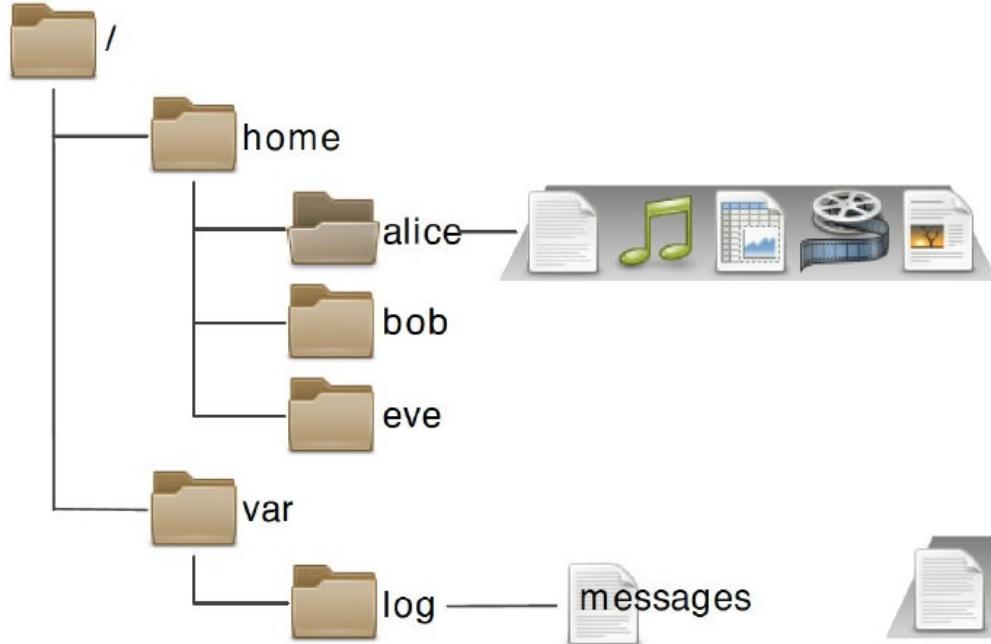
- 在Linux系统中一切以文件为基础进行管理
- 当用户登录系统后都将处在某一目录中



常用目录结构介绍

- / 根目录
- /home、/root 用户的宿主目录
- /usr 命令程序文件
- /etc 配置文件
- /boot 系统启动文件
- /dev 设备文件
- /tmp、/var/tmp 临时文件
- /var 变量数据（日志、邮件、web内容等）
- /run 进程最后一次运行数据

通过文件名定位文件



路径的表示

- 绝对路径
 - ◆ 以 “/” 开始的路径表示形式
 - ◆ 文件系统中每个文件都有一个唯一绝对路径
- 相对路径
 - ◆ 指从当前工作目录到达指定文件所需的路径，
可以是以 “.” 或者“ ..” 开始的目录路径

导航路径

- **pwd命令**

- ◆ 用途：查看工作目录 (Print Working Directory)

- **cd命令**

- ◆ 用途：切换工作目录 (Change Directory)

- ◆ 格式：cd [目录位置]

- **ls命令**

- ◆ 用途：列表 (List) 显示目录内容

- ◆ 格式：ls [选项]... [目录或文件名]

使用命令行管理文件/目录

- 常见文件管理命令

- ◆ 查看文件内容: cat、 less、 more、 head、 tail
- ◆ 创建删除文件: touch、 rm
- ◆ 拷贝移动文件: cp、 mv

| 目的 | 实例1 | 实例2 |
|------|-----------------------------|---------------------------------------|
| 复制文件 | <code>cp file1 file2</code> | <code>cp file1 file2 file3 dir</code> |
| 移动文件 | <code>mv file1 file2</code> | <code>mv file1 file2 file3 dir</code> |
| 删除文件 | <code>rm file1</code> | <code>rm -f file1 file2 file3</code> |

`cp file1 file2`

`mv file1 file2`

`rm file1`

`cp file1 file2 file3 dir`

`mv file1 file2 file3 dir`

`rm -f file1 file2 file3`

使用命令行管理文件/目录

- 常见目录管理命令

- ◆ 创建删除目录: mkdir、 rm
- ◆ 复制移动目录: cp、 mv

目的

创建目录

复制目录

移动目录

删除目录

实例1

mkdir dir

cp -r dir1 dir2

mv dir1 dir2

rm -r dir1

实例2

mkdir -p par1/par2/dir

cp -r dir1 dir2 dir3 dir4

mv dir1 dir2 dir3 dir4

rm -rf dir1 dir2 dir3

使用通配符匹配文件名称

- 通配符

- ◆ 通配符是一种特殊语句，常用于模糊查找
- ◆ 当查找文件时，可以用来代替一个或多个真正字符
- ◆ 常用通配符
 - ◆ * 匹配多个任意字符
 - ◆ ? 匹配一个任意字符
 - ◆ [abc...] 匹配指定范围内的任意一个字符
 - ◆ {a,b} 或{a..c} 匹配括号中的连续字符

使用通配符匹配文件名称

●其它通配符

- ◆ [!abc...] 或者 [^abc...] 不包含字符
- ◆ [:alpha:] 任何字母字符
- ◆ [:lower:] 任何小写字符
- ◆ [:upper:] 任何大写字符
- ◆ [:alnum:] 任何字母或数字
- ◆ [:punct:] 除空格和字母、数字以外的任何可打印字符
- ◆ [:digit:] 任何数字，即0-9
- ◆ [:space:] 任何空白字符

请思考

- 把 /etc/fstab 拷贝到 /var/tmp/fstab
- 在/home下创建一个目录为sysadms

Q&A



第3章 在RHEL中获得帮助



关注微信公众号
享终身免费培训

目标

- 使用Linux帮助man手册
- 使用pinfo 帮助
- 使用红帽软件包（RPM）文档
- 使用线上帮助

查找帮助

- 本地帮助工具
 - ◆ Applications→Favorites→help
- 自身命令帮助
 - ◆ help history
 - ◆ mkdir --help

Man帮助手册

- 位置： /usr/share/man/
 - ◆ man 手册标准章节
 - ◆ 常用
 - 1 用户命令
 - 5 文件格式
 - 8 系统管理命令

| | |
|---|----------------------|
| 1 | User commands |
| 2 | System calls |
| 3 | Library functions |
| 4 | Special files |
| 5 | File formats |
| 6 | Games |
| 7 | Conventions |
| 8 | Maintenance commands |
| 9 | Linux Kernel API |

Man帮助手册

- 使用方法：

- ◆ # man [章节信息] 命令或者配置文件
- ◆ # man -k 关键字

- 基本操作：

- ◆ PgUp, PgDn : (向上, 向下) 翻页
- ◆ /, ? : (向下, 向上) 搜索关键字
- ◆ n, N : (向下, 向上) 查找下一个关键字
- ◆ q : 退出搜索页

Man手册结构说明

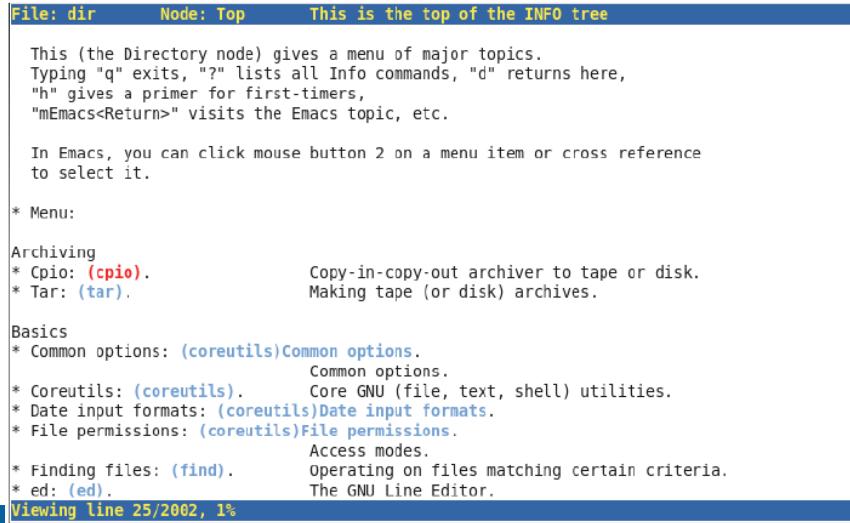
| 结构名称 | 描述 |
|-------------|------------|
| NAME | 名称及简要说明 |
| SYNOPSIS | 格式和使用方法说明 |
| DESCRIPTION | 详细说明 |
| OPTIONS | 可用选项及其介绍说明 |
| EXAMPLES | 示例（附带简单说明） |
| FILES | 相关文件 |
| ENVIRONMENT | 环境变量 |
| SEE ALSO | 其它帮助参考 |

Man帮助手册

- 请思考，以下两个 passwd 怎么使用 man 查找帮助？
 - ◆ /usr/bin/passwd
 - ◆ /etc/passwd

Pinfo帮助文档

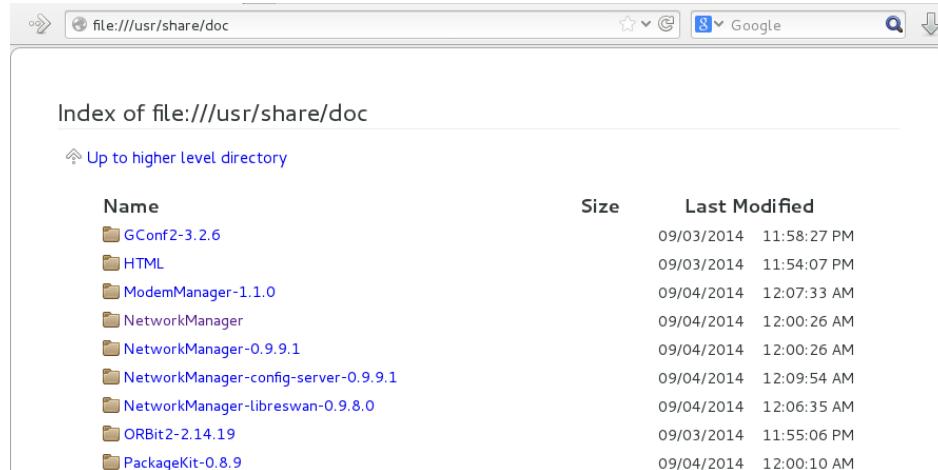
- 文档结构由超链接式的 info 节点组成，记录了很多复杂命令和概念的彻底说明
- 在 RHEL 中可以使用 info 和 pinfo 两个命令浏览 info page



RPM包文档

- 本地帮助文档

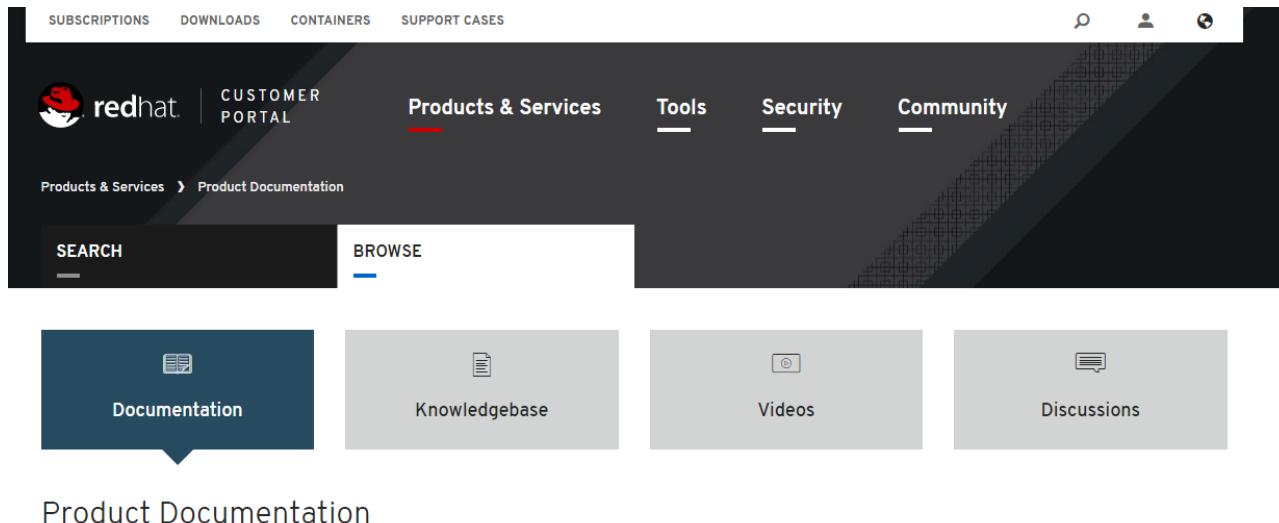
- ◆ 安装软件包后，在doc目录下释放文档
- ◆ `/usr/share/doc/packagename`
- ◆ 可使用浏览器查看



在线上获取帮助

- 在线查找帮助，一般用于查找新技术的官方参考文档
 - ◆ 红帽官方文档

- <https://access.redhat.com>



Q&A



第4章 创建、查看编辑文本



关注微信公众号
享终身免费培训

目标

- vi编辑器
- gedit文本编辑
- 数据流重定向

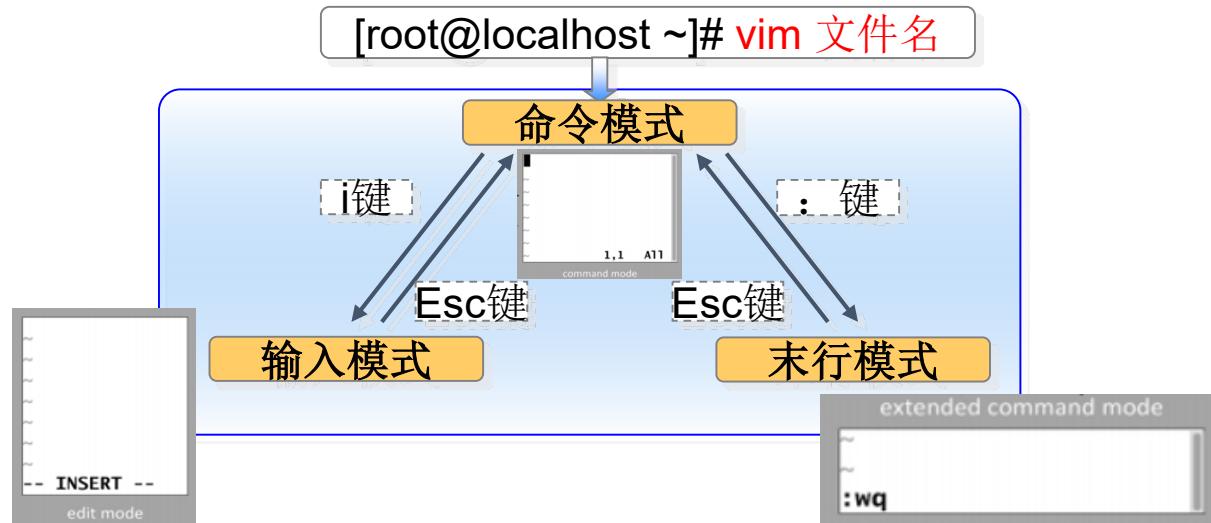
文本编辑器

- 文本编辑器作用
 - ◆ 创建或修改文本文件
 - ◆ 维护 Linux 系统中的各种配置文件
- Linux中最常用的文本编辑器
 - ◆ vi : 类 Unix 系统中默认的文本编辑器
 - ◆ vim: vi编辑器的增强版本, 习惯上也称为 vi

VI编辑器的工作模式

- 三种工作模式

- ◆ 命令模式 (command) 、输入模式(insert)、末行模式(ex)



VI编辑器的典型操作

- 命令模式下的基本操作

- ◆ 光标移动

- 方向键
 - h、j、k、l
 - 1G、G

- ◆ 撤销编辑

- u

- ◆ 删除

- 单个字符 x
 - 整行 dd

VI编辑器的典型操作

● 末行模式下的基本操作

◆ 保存退出

- :w 保存
- :q 退出
- :q! 退出不保存
- :wq 保存并退出

gedit文本编辑器

- 用于编辑文本文件的图形工具

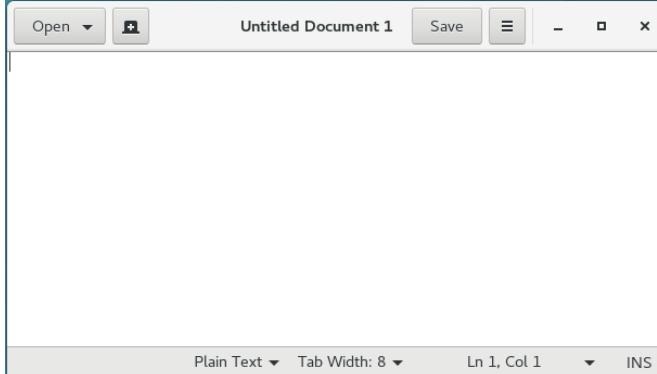
- ◆ 与windows记事本类似

- ◆ 在GNOME中启动：

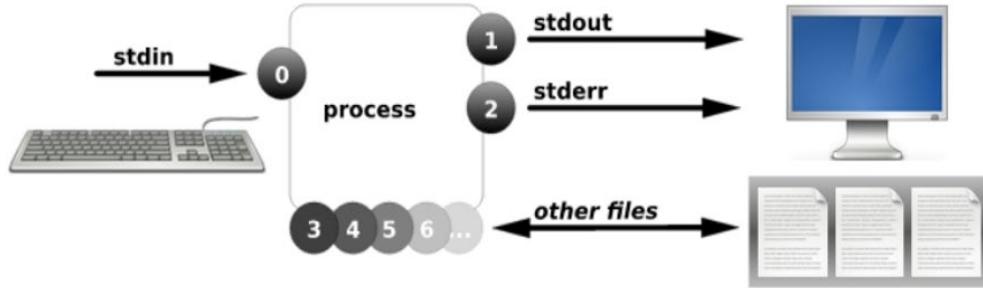
- Applications->Accessories->Text Editor

- ◆ 在命令行中启动：

- gedit



标准输入输出

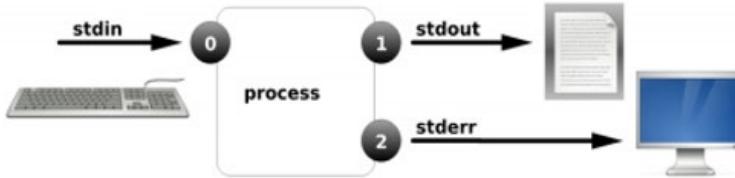


| 输入输出文件 | 文件编号 | 默认设备 |
|-----------------|------|------|
| 标准输入 (stdin) | 0 | 键盘 |
| 标准输出 (stdout) | 1 | 显示器 |
| 标准错误输出 (stderr) | 2 | 显示器 |

I/O重定向操作

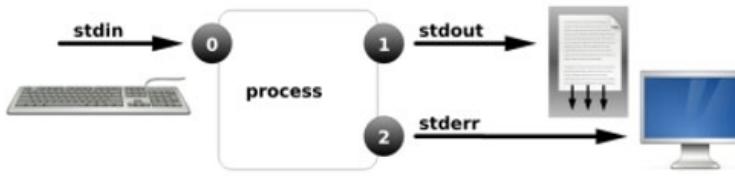
- >file

- ◆ 标准重定向



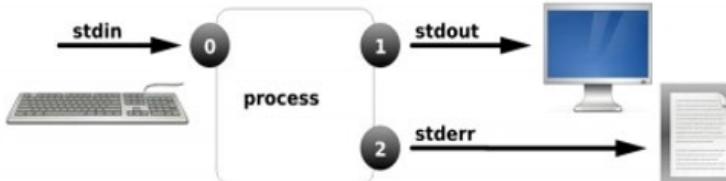
- >>file

- ◆ 追加重定向



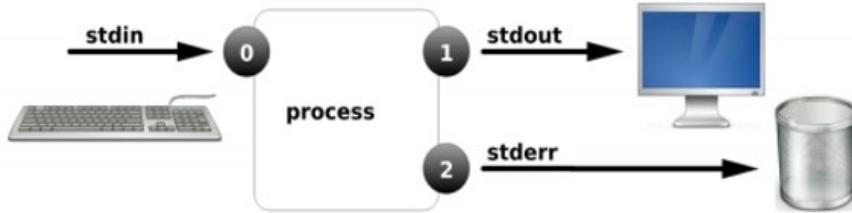
- 2>file

- ◆ 错误重定向

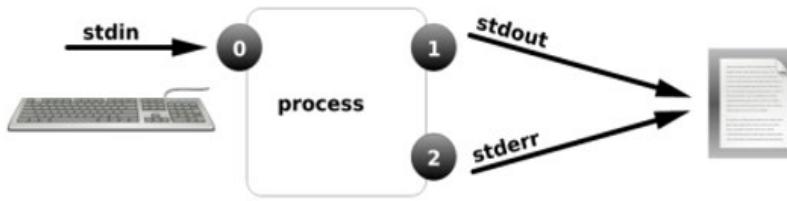


I/O重定向到指定文件或程序

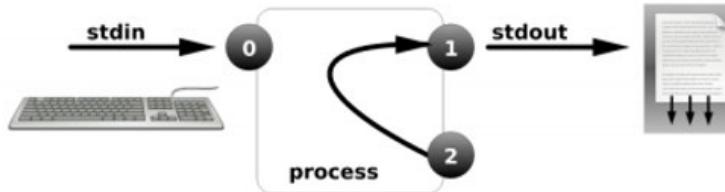
- 2>/dev/null
 - ◆ stderr 重定向到 /dev/null



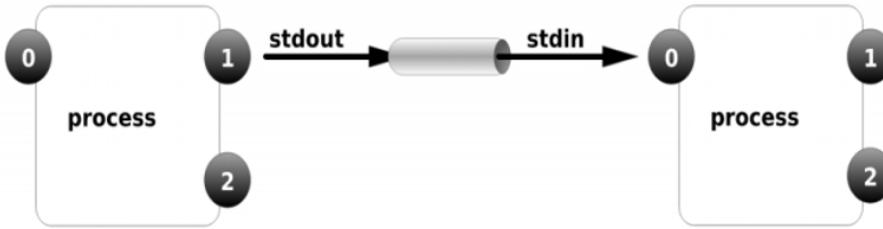
- &>file
 - ◆ 结合 stdout 和 stderr 到一个文件



- &>>file
 - ◆ 结合 stdout 和 stderr 追加到当前文件内容后

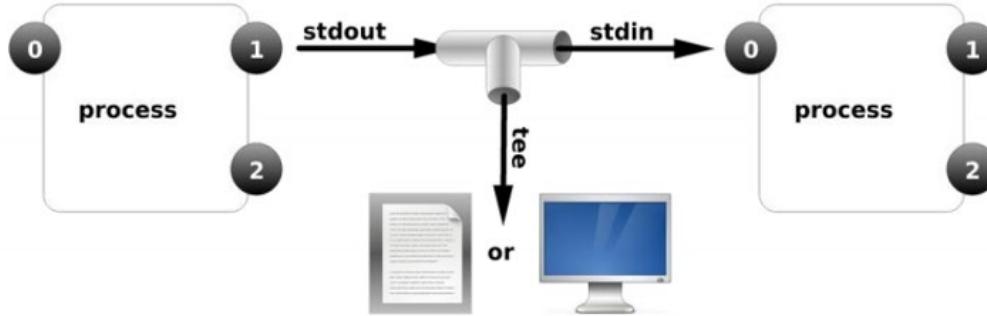


管道符



- “|” 左边命令执行结果作为 “|” 右边命令的输入
 - ◆ 格式: cmd1 | cmd2
 - ◆ 例子:
 - # ls -l /usr/bin | more

三通管道



- 管道末尾使用tee命令把数据重定向到指定文件和屏幕上

◆ 例子：

- # ls -l | tee /tmp/files-save

文本内容筛选

- grep工具
- 用途：在文件中查找并显示包含指定字符串的行
 - ◆ 命令格式：grep [选项]... 查找条件 目标文件
 - 常用命令选项
 - -i：查找时忽略大小写
 - -v：反转查找，输出与查找条件不相符的行
 - ◆ **查找条件设置**
 - 要查找的字符串以双引号括起来
 - “^..” 表示以..开头， “..\$” 表示以..结尾

文本内容筛选示例

- grep示例：

- ◆ grep "root" /etc/passwd
- ◆ grep "^root" /etc/passwd
- ◆ grep "bash\$" /etc/passwd
- ◆ grep -v "^#" /etc/resolv.conf

请思考

- 从文件 /etc/resolv.conf 中找出包含 na 的行写入到文件 /root/lines.txt 中，要求顺序与原文件中一致，并且不包含注释行

Q&A



第5章 管理本地用户和组



关注微信公众号
享终身免费培训

目标

- 理解 Linux 用户和组
- 管理本地用户和组

用户账户和组账户概述

- Linux 基于用户身份对资源访问进行控制

- ◆ 用户帐号： UID 标识

- 超级用户 **root**
 - 普通用户
 - 程序用户

- ◆ 组帐号： GID 标识

- 基本组(私有组)
 - 附加组 (公共组)

用户账户文件—/etc/passwd

- 用于保存用户的帐户基本信息

① username: ② password: ③ UID: ④ GID: ⑤ GECOS: ⑥ /home/dir: ⑦ shell

- ◆ 字段1：用户帐号的名称
- ◆ 字段2：密码占位符 “x”
- ◆ 字段3：用户帐号的UID号
- ◆ 字段4：所属基本组帐号的GID号
- ◆ 字段5：用户描述
- ◆ 字段6：宿主目录
- ◆ 字段7：登录Shell信息

本地用户账户管理

- 添加用户帐号
 - ◆ 格式: useradd [选项]... 用户名
- 设置/更改用户口令
 - ◆ 格式: passwd [选项]... 用户名
- 修改用户帐号的属性
 - ◆ 格式: usermod [选项]... 用户名
- 删除用户帐号
 - ◆ 格式: userdel [-r] 用户名

请思考

- 创建用户 fred，用户ID为 1111，密码为 fred1111

组账户文件—/etc/group

- 组账户分类：
 - ◆ 基本组
 - ◆ 附加组
- 保存组账户基本信息

`groupname:password:GID:list,of,users,in,this,group`

- ◆ 创建组帐号
 - 格式: groupadd [-g GID] 组帐号名
- ◆ 修改现有组
 - 格式: groupmod [-n | -g]
- ◆ 添加、删除组成员
 - 格式: gpasswd [-a|-M|-d] 用户名... 组帐号名
- ◆ 删除组账户
 - 格式: groupdel 组帐号名

超级用户权限

- 使用 su - 切换用户
 - ◆ 切换至 root 用户，使用 root 用户密码验证
 - ◆ 切换至其他用户，使用目标用户密码验证
 - ◆ 命令格式
 - su [-] [username]
- 使用sudo执行命令
 - ◆ 基于 /etc/sudoers 配置，允许以超级用户(或其他用户)身份执行命令
 - 使用用户自己的密码验证
 - ◆ 命令格式
 - sudo [-u username | UID] command

用户密码文件 —/etc/shadow

- 用于保存密码字串、密码有效期等信息

① name: ② password: ③ lastchange: ④ minage: ⑤ maxage: ⑥ warning: ⑦ inactive: ⑧ expire: ⑨ blank

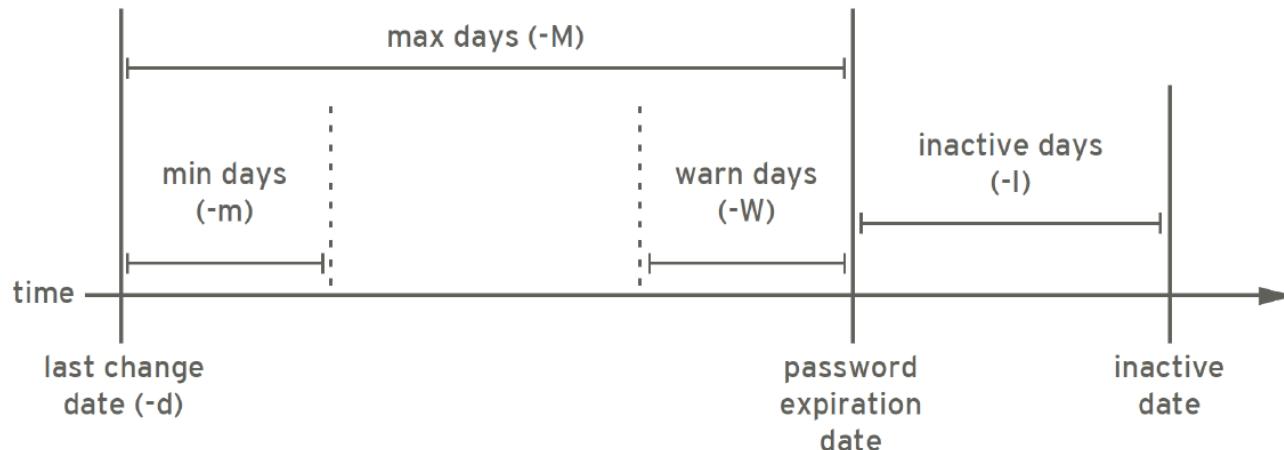
- ◆ 字段1：用户帐号的名称
- ◆ 字段2：加密的密码字串信息
- ◆ 字段3：最近一次修改密码的时间，表示从1970.01.01至今的天数
- ◆ 字段4：密码的最短使用天数，默认值为0，没有要求
- ◆ 字段5：密码的最长有效天数，默认值为99999
- ◆ 字段6：密码即将到期警告天数，默认值为7
- ◆ 字段7：在密码过期之后账号保持活动的天数，指定天数后账号被锁定，成为无效
- ◆ 字段8：帐号失效时间，默认值为空，以1970.01.01的天数表示
- ◆ 字段9：保留字段（未使用）

密码老化时间轴

- 修改用户帐号密码有效期限

- ◆ 语法: chage [选项] 用户

- 修改老化时间 # chage -m 0 -M 90 -W 7 -I 14 username
 - 强制下次登录修改密码 # chage -d 0 username



请思考

- 创建组sysadm
- 创建用户harry, natasha, tom
- 要求harry, natasha的附加组为sysadm
- 要求tom用户的登陆shell为非交互式shell
- 三个用户的密码为redhat

Q&A



第6章 Linux文件权限



关注微信公众号
享终身免费培训

目标

- 理解Linux文件权限
- 基于命令行管理文件权限及所属
- 了解文件默认权限
- 文件访问控制管理

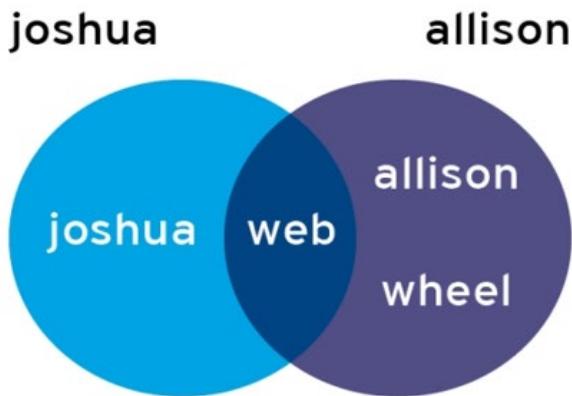
Linux文件权限简介

- 权限的分类

- ◆ r (读)

- ◆ w (写)

- ◆ x (执行)



- 查看权限方法

- ◆ 文件

- # ls -l 文件名

- ◆ 目录

- # ls -ld 目录名

Linux系统的权限表示

```
# ls -l test
```

- rw-r--r--. 1 stu1 class1 35 May 21 14:09 test

访问权限

属主

属组

| 权限分配 | 文件所有者 | | | 文件所属组用户 | | | 其他用户 | | |
|------|-------|---|----|---------|---|----|------|---|----|
| 字符表示 | r | w | x | r | w | x | r | w | x |
| 权限项 | 读 | 写 | 执行 | 读 | 写 | 执行 | 读 | 写 | 执行 |
| 数字表示 | 4 | 2 | 1 | 4 | 2 | 1 | 4 | 2 | 1 |

Linux系统的权限的作用

- 对文件和目录的影响

| 权限 | 对文件的影响 | 对目录的影响 |
|--------|------------|-----------------|
| r (读) | 可以读取文件的内容 | 可以列出目录的内容（文件名） |
| w (写) | 可以更改文件的内容 | 可以创建或删除目录中的任一文件 |
| x (执行) | 可以作为命令执行文件 | 可以访问目录的内容 |

使用符号方式修改文件权限

- chmod命令用于更改文件对于某类用户的操作权限
- chmod who what which FILE...

设置权限的对象
可以是：

u (用户)
g (组)
o (其他人)
a (所有人)

属性操作符
+增加权限
-减少权限
=精确设置

权限内容，可以是：
r (读)
w (写)
x (执行)
以及这些权限的组合

被设置属性的文件或目录

使用数字方式修改文件权限

- chmod nnn FILE...

- ◆ 使用三个数字 (nnn) 模式
- ◆ 通过把数值相加来计算权限
 - 第一个数字代表所属主 (u) 的权限
 - 第二个数字代表属组 (g) 的权限
 - 第三个数字代表其它人 (o) 的权限

| rwx.. | | |
|-------|--------|-------|
| 000 | 000 =0 | --- |
| 001 | 001 =1 | --x.. |
| 010 | 020 =2 | -w- |
| 011 | 021 =3 | -wx.. |
| 100 | 400 =4 | r--.. |
| 101 | 401 =5 | r-x.. |
| 110 | 420 =6 | rw- |
| 111 | 421 =7 | rwx.. |

文件权限设置实例

- 使用chmod命令设置文件权限

- ◆ 增加文件属主的执行权限 (x)

- \$ chmod u+x test

- ◆ 去除文件属主与属组的写权限 (w)

- \$ chmod u-w,g-w test

- ◆ 设置其他用户的文件权限为可执行

- \$ chmod o=x test

- ◆ 设置属主完全控制，属组和其他人读权限

- \$ chmod 744 test

设置文件属主和属组

- chown命令用于设置文件的属主和属组

- ◆ 命令格式

- chown OWNER[:GROUP] FILE...

- ◆ 例子：

- 设置文件test的属主为用户stu2

- # chown stu2 test

- 设置文件test的属组为用户组class2

- # chown :class2 test

- 设置文件test的属主为stu3，并设置文件的属组为class3

- # chown stu3:class3 test

文件默认权限

- 使用umask查找或设定
 - ◆ 定义文件生成时的默认权限
 - ◆ 文件生成的默认权限为“去除” umask后的值
- 命令格式
 - ◆ umask [-S] [mode]
 - ◆ 例子：

```
[student@desktop0 ~]$ umask  
0002  
[student@desktop0 ~]$ umask -S  
u=rwx,g=rwx,o=rx
```

特殊权限

- SET位权限

- ◆ SET位权限类型：

- SUID：
 - 为可执行（有 x 权限的）文件设置，权限字符为 “s”
 - 应用示例：/usr/bin/passwd
 - 其他用户执行该文件时，将拥有者用户的权限
 - SGID：
 - 为可执行（有 x 权限的）文件或目录设置，权限字符为 “s”
 - 其他用户执行该文件时，将拥有所属组的权限
 - 拥有特殊权限的目录中创建目录或文件会自动继承目录所属组

特殊权限

- 粘滞位权限 (Sticky)

- ◆ 主要用途：

- 为公共目录（权限为777的）设置，权限字符为“t”
 - 用户不能删除该目录中其他用户的文件

- ◆ 应用示例：

- /tmp
 - /var/tmp

设置特殊权限

- 设置SET位、粘滞位权限

- ◆ 用字符表示

- chmod ug±s 可执行文件...
 - chmod o±t 目录名...

- ◆ 用数字表示 (第四位)

- chmod mnnn 可执行文件...
 - m为 SUID=4, SGID=2, Sticky=1, 可累加

请思考

- 在 /home 下创建一个目录为 sysadms
- 要求所属的组为 sysadm
- 组成员可以读写，其他用户没有任何权限
- 同组成员在目录下创建的文件所属组也为 sysadm 组

Q&A



第7章 进程监控及管理



关注微信公众号
享终身免费培训

目标

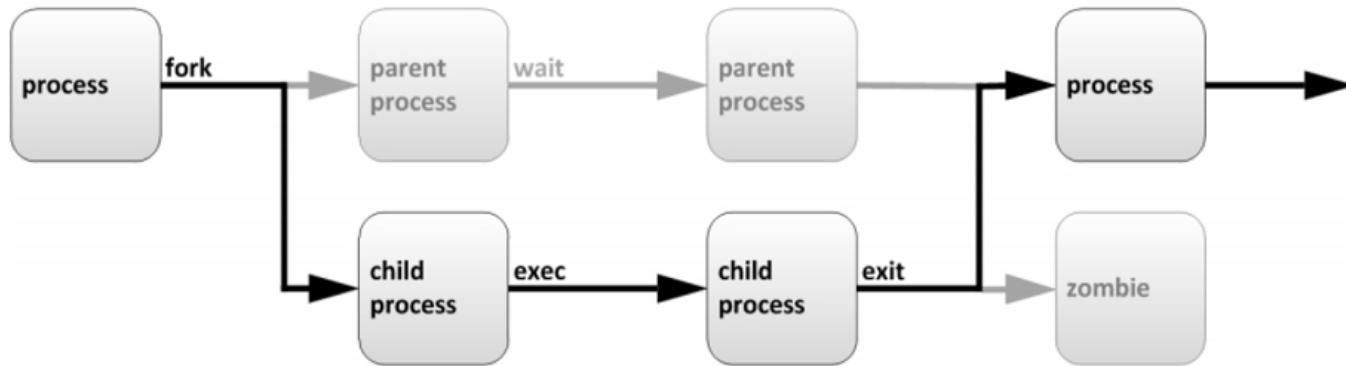
- 系统进程的基本信息
- 使用作业控制进程
- 使用信号终止进程

程序与进程

- 程序是保存在磁盘中机器代码指令和数据的集合
- 进程可以被认为是执行中的计算机程序
- Linux 是多进程操作系统，每个程序启动时，可以创建一个或几个进程

进程简介

- 进程 ID (Process ID, PID) 号码用来标记各个进程

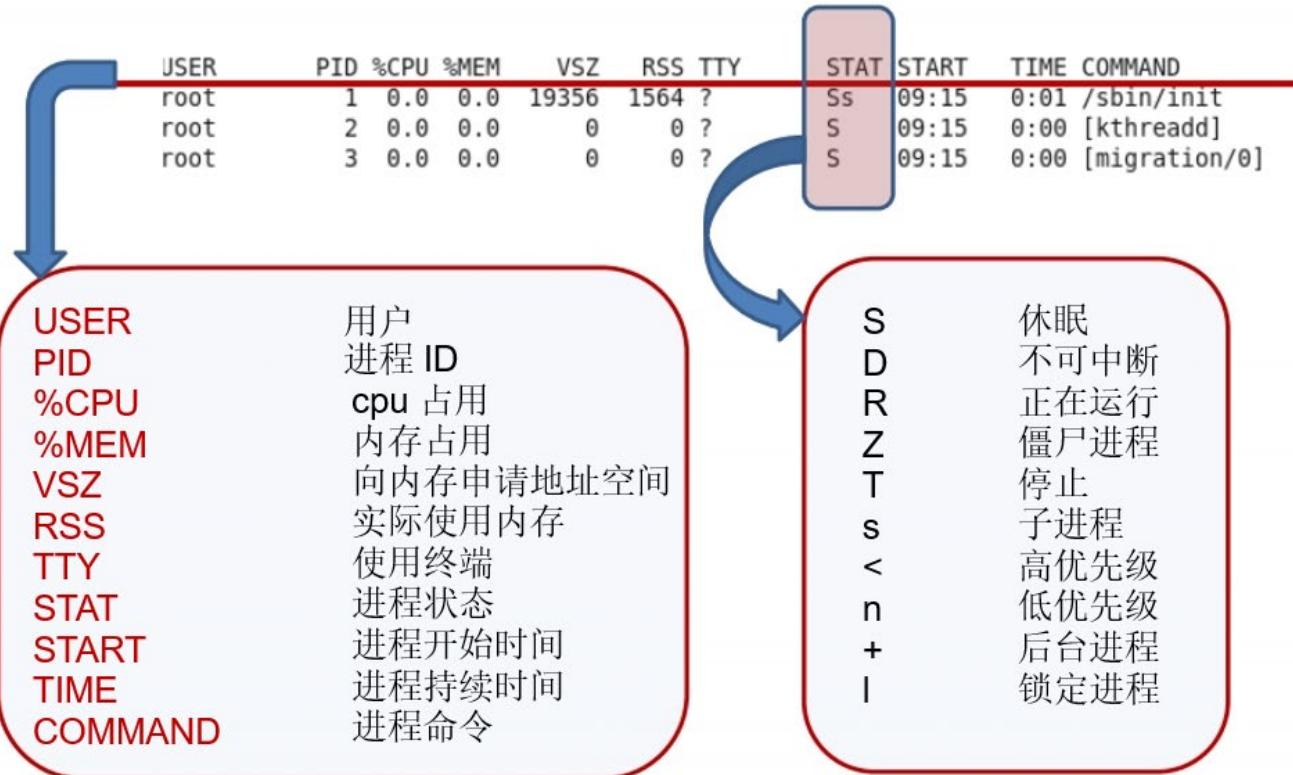


查看进程

- 使用 ps 来查看进程信息
 - ◆ 最灵活： ps 选项 | 其它命令
 - 如： ps aux |grep httpd

- 使用 top 来查看进程信息
 - ◆ 实时显示进程信息
 - ◆ 允许排序、中止运行
 - M : 按内存使用情况排序
 - P : 按CPU使用情况排序
 - h : 显示帮助

进程菜单与状态



进程状态

| 状态 | 标志 |
|-------|----------|
| 运行 | R |
| 睡眠/休眠 | S |
| | D (不可中断) |
| | K |
| 已停止 | T |
| 僵停 | Z |
| | X |

信号

- 对进程的操作可以通过发送不同的信号来实现

- ◆ 信号来源

- 外部信号，如ctrl+c终止
 - 通过kill等命令向进程发送信号
 - 程序错误，如非法访问内存

- ◆ 信号可以使用名称或号码表示

- 信号 1, SIGHUP – 重读配置文件
 - 信号 9, SIGKILL – 立即终止
 - 信号 15, SIGTERM - 整洁终止
(缺省信号)

- ◆ man 7 signal 显示了完整列表

终止进程

- 发送信号

- ◆ 按 PID :

- kill [-9] *PID*

- ◆ 按名称 :

- killall [- signal] 名称
 - pkill 名称

管理登录用户

```
[root@desktop0 ~]# pgrep -l -u student
```

```
31764 ping
```

```
31819 sshd
```

```
31820 bash
```

```
31947 bash
```

```
[root@desktop0 ~]# w -h -u student
```

```
student  ttym1    14:21   36.00s  0.02s  0.02s -bash
```

```
student  pts/0    14:18    4.00s  0.12s  0.00s w -h -u student
```

```
[root@desktop0 ~]# pkill -SIGKILL -t ttym1
```

```
[root@desktop0 ~]# w -h -u student
```

```
student  pts/0    14:18    7.00s  0.13s  0.00s w -h -u student
```

作业控制

- 后台运行进程

- ◆ 在命令行后添加一个 “&” 符号:

- firefox &

- 暂时停止某个运行程序

- ◆ Ctrl-z

- 管理后台作业

- ◆ jobs
 - ◆ bg [%作业号码]
 - ◆ fg [%作业号码]
 - ◆ kill [-signal] [%作业号码]

Q&A



第8章 控制服务与守护进程



关注微信公众号
享终身免费培训

目标

- 使用 systemctl 控制网络服务与守护进程

systemd

- init 进程

- ◆ 在之前版本的 RHEL 中，使用 init 进程
- ◆ PID为1
- ◆ 系统最先启动的进程，为所有进程的父进程

- systemd 进程

- ◆ 在 RHEL7 中，使用 system 进程代替 init
- ◆ PID为1

systemd特性

- 系统引导时实现服务并行启动
- 按需激活进程
- 基于依赖关系定义服务控制逻辑
- 向后兼容 sysv init 脚本

单元(unit)

- 单元
 - ◆ 不同类型的 system 对象
 - ◆ 使用 systemctl 管理
- 查看系统中有效的 unit
 - ◆ 使用命令：systemctl -t help
- 常见单元类型
 - ◆ service : 扩展名为.service，代表系统服务
 - ◆ socket: 扩展名为.socket，表述进程间通信(IPC)

服务状态

- 查看服务状态

◆ 命令：`systemctl status NAME.service`

```
[root@desktop0 ~]# systemctl status sshd.service
sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
  Active: active (running) since Wed 2018-05-16 12:21:33 CST; 2h 20min ago
    Process: 1088 ExecStartPre=/usr/sbin/sshd-keygen (code=exited, status=0/SUCCESS)
   Main PID: 1136 (sshd)
     CGroup: /system.slice/sshd.service
             └─1136 /usr/sbin/sshd -D
```

```
May 16 12:21:33 localhost systemd[1]: Started OpenSSH server daemon.
May 16 12:21:33 localhost sshd[1136]: Server listening on 0.0.0.0 port 22.
May 16 12:21:33 localhost sshd[1136]: Server listening on :: port 22.
May 16 14:18:25 desktop0.example.com sshd[31816]: Accepted publickey for stud...
Hint: Some lines were ellipsized, use -l to show in full.
```

服务状态关键字

| 关键字 | 描述 |
|-----------------|---------------------|
| loaded | 配置文件被处理 |
| active(running) | 一个或多个进程持续运行 |
| active(exited) | 成功完成一个“一次性”配置 |
| active(waiting) | 运行但等待“事件” |
| inactive | 未运行 |
| enabled | 开机启动 |
| disabled | 不随开机启动 |
| static | 不能被启用，但可以被已启用单元自动启动 |

查看服务状态

- 查看服务启动启用状态

- ◆ systemctl is-active service
- ◆ systemctl is-enabled service

```
[root@desktop0 ~]# systemctl is-active sshd.service
active
[root@desktop0 ~]# systemctl is-enabled sshd.service
enabled
```

管理系统服务

- 启动与停止服务
 - ◆ `systemctl {start | stop | restart |reload} NAME.service`
 - `systemctl start sshd.service`
- 启用与停用开机自动启动
 - ◆ `systemctl {enable | disable} NAME.service`
 - `systemctl enable sshd.service`

管理系统服务

- 更多命令

- ◆ `systemctl mask NAME.service`
- ◆ `systemctl unmask NAME.service`

Q&A



第9章 OpenSSH服务



关注微信公众号
享终身免费培训

目标

- 使用 SSH 登录到远程系统
- 基于密钥认证的 SSH 登录
- 配置 sshd 服务

SSH远程管理

- 什么是 OpenSSH?
 - ◆ 是指系统中使用的 Secure Shell 软件的实施，用于在远程系统上安全运行 Shell
- 使用 SSH 进行远程管理会带来哪些好处
 - ◆ 所有数据在传输过程中都被加密
 - ◆ 客户端为每一个成功连接的服务器保存了钥匙，可以防止假冒

SSH的简单登录实例

- SSH创建远程交互 shell，结束时使用 exit
 - ssh remotehost [command]
 - ssh user@remotehost [command]

```
[student@host ~]$ ssh remotehost
student@remotehost's password:
[student@remotehost ~]$ exit
Connection to remotehost closed.
[student@host ~]$
```

```
[student@host ~]$ ssh remoteuser@remotehost
remoteuser@remotehost's password:
[remoteuser@remotehost ~]$
```

```
[student@host ~]$ ssh remoteuser@remotehost hostname
remoteuser@remotehost's password:
remotehost.example.com
[student@host ~]$
```

基于密钥认证的SSH

- 用户使用“密钥”验证身份登录，可以不使用用户密码
 - ◆ 公钥-私钥方案进行身份验证
 - 私钥：认证凭据
 - 公钥：用于验证私钥（需要拷贝到被登录系统）
- 使用命令 ssh-keygen 生成密钥
 - ◆ ~/.ssh/id_rsa
 - ◆ ~/.ssh/id_rsa.pub

基于密钥认证的SSH

```
[root@desktop0 ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
1d:38:98:38:52:27:d7:8d:cf:bc:e7:45:d4:dd:7e:93 root@desktop0.example.com
The key's randomart image is:
+--[ RSA 2048]----+
|   o . o     .o|
| . = oo..    . +|
| . o o o+.   . ..|
| . .   o+.   .Eo|
|      S . . . o|
|           . . . |
|               o . |
|                   .|
+-----+
```

基于密钥认证的SSH

- 将公钥拷贝至被登录主机
 - ◆ 使用命令 ssh-copy-id
- 命令格式：
 - ◆ ssh-copy-id user@remotehost
 - ◆ 公钥信息会被记录在在被登录主机 ~/.ssh/authorized_keys 文件中

配置sshd服务

- OpenSSH配置文件
 - ◆ /etc/ssh/sshd_config
- 对OpenSSH的安全措施
 - ◆ 配置是否允许root账户SSH登录
 - PermitRootLogin yes|no
 - ◆ 配置仅使用基于密钥认证登录root用户
 - PermitRootLogin without-password
 - ◆ 配置只允许密钥登录（即，禁止用户密码进行身份验证）
 - PasswordAuthentication no

Q&A



第10章 日志分析与存储



关注微信公众号
享终身免费培训

目标

- 描述 RHEL7 的基本日志结构
- 根据日志信息维护系统
- systemd-journald 日志存储
- 维护系统时间

日志文件

- 日志的功能

- ◆ 日志文件是记录 Linux 中系统消息的文件
- ◆ 不同的日志文件记载不同类型的信息，包括内核、服务以及在系统中运行的应用程序的信息
- ◆ 日志文件有助于诊断和解决系统中的问题

日志守护进程

- **systemd-journald**

- ◆ 从内核收集消息
- ◆ 包括启动进程、标准输出及错误、进程运行启动等记录
- ◆ 重启不保存

- **rsyslog**

- ◆ 根据类别(facility)及优先级别(severity)筛选系统记录
- ◆ 将信息写入 /var/log 下对应的日志文件，统一管理日志文件

系统中的日志文件

- 日志文件的位置

- ◆ 日志文件保存在目录 “/var/log/” 中

Overview of system log files

| Log file | Purpose |
|-------------------|---|
| /var/log/messages | Most syslog messages are logged here. The exceptions are messages related to authentication and email processing, that periodically run jobs, and those which are purely debugging-related. |
| /var/log/secure | The log file for security and authentication-related messages and errors. |
| /var/log/maillog | The log file with mail server-related messages. |
| /var/log/cron | The log file related to periodically executed tasks. |
| /var/log/boot.log | Messages related to system startup are logged here. |

rsyslog服务管理的日志文件

- 主要配置文件：/etc/rsyslog.conf

```
##### RULES #####
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                     /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none      /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                    /var/log/secure

# Log all the mail messages in one place.
mail.*                                         -/var/log/maillog

# Log cron stuff
cron.*                                         /var/log/cron

# Everybody gets emergency messages
*.emerg                                         :omusrmsg: *
```

rsyslog配置文件

- 类别(facility)

- ◆ Kern 内核
- ◆ authpriv 授权和安全
- ◆ cron 计划任务
- ◆ mail 邮件
- ◆ daemon 系统守护进程
- ◆ syslog 由rsyslog生成的信息
- ◆ local0~local7 自定义本地策略

rsyslog配置文件

- 日志消息的级别(severity)

- | | |
|------------------|---------------|
| ◆ 0 EMERG (紧急) | 会导致主机系统不可用的情况 |
| ◆ 1 ALERT (警告) | 必须马上采取措施解决的问题 |
| ◆ 2 CRIT (严重) | 比较严重的情况 |
| ◆ 3 ERR (错误) | 运行出现错误 |
| ◆ 4 WARNING (提醒) | 可能会影响系统功能的事件 |
| ◆ 5 NOTICE (注意) | 不会影响系统但值得注意 |
| ◆ 6 INFO (信息) | 一般信息 |
| ◆ 7 DEBUG (调试) | 程序或系统调试信息等 |

分析日志条目

- 由rsyslog记录的所有日志都有一个标准的格式

◆ 例如：

```
① Feb 11 20:11:48 ② localhost ③ sshd[1433]: ④ Failed password for student from  
172.25.0.10 port 59344 ssh2
```

- 1. 记录该日志条目的时间戳
- 2. 发送该日志消息的主机
- 3. 产生该日志信息的程序或进程
- 4. 实际发送的日志消息

- 使用tail -n 0 -f 命令监控日志的末尾

Logger命令测试

- 使用logger向系统日志写入信息
 - ◆ 默认将消息发送至user类以notice级别记录，可使用-p选项另外指定
 - ◆ 命令格式
 - *logger -p Facility.Severity Messages*
 - 例子：
 - # logger -p authpriv.info "test test"
 - # tail -1 /var/log/secure

日志转储

- logrotate 程序是一个日志文件管理工具
- 用来把旧的日志文件删除，并创建新的日志文件
- 可以根据日志文件的大小，也可以根据其天数来转储
- 这个过程一般通过 cron 程序来执行

日志转储

- 转储日志一般存放在：/var/log
 - ◆ 转储日志会在原日志后加上日期后缀
 - 如：/var/log/messages-20180523
 - ◆ 转储日志， 默认每周生成一次并且最多生成四个转储文件， 第五个出现时第一个自动删除
 - ◆ 转储日志文件， 有时也会因为达到设定的日志大小而提前（默认一周）产生。

Logrotate配置文件

- # cat /etc/logrotate.conf

```
# rotate log files weekly
weekly                                //日志将按周轮询
# keep 4 weeks worth of backlogs
rotate 4                               //指定日志文件删除之前保留的日志文件个数
# create new (empty) log files after rotating old ones
create                                //自动建立新日志文件，不建立使用nocreate
# use date as a suffix of the rotated file
dateext                               //使用日期作为日志轮询文件的后缀
include /etc/logrotate.d
# no packages own wtmp and btmp -- we'll rotate them here
/var/log/wtmp {                         //每月转储/var/log/wtmp一次最小为1M的例子
    monthly
    create 0664 root utmp               //指定权限创建全新的日志文件
    minsize 1M                          //日志轮询的最小值
    rotate 1                            //一次存储1个归档日志
}
```

systemd日志

- systemd日志数据存储在带有索引的二进制文件中
 - ◆ 默认保存在/run/log/journal/目录中
 - ◆ 重启系统清空
 - ◆ Journalctl命令查看systemd日志信息

systemd日志

- 使用journalctl命令以粗体文本显示优先级为notice或warning的消息，以红色文本突出显示error和更高的消息

- ◆ 查看方法：

- journalctl -n [number]
 - 默认查看最后10行，-n 选项指定最后多少个条目
 - journalctl -p severity|level
 - 按消息级别名称或级别编号查看日志
 - journalctl -f
 - 默认实时显示最后10行，类似tail -f
 - Journalctl --since “yyyy-mm-dd hh:mm:ss” --until “yyyy-mm-dd hh:mm:ss”
 - 按时间范围查找具体事件

设置永久的systemd日志

- 若/var/log/journal目录存在，则可以将systemd日志保存在此，重启后仍有历史数据
 - 使用在/etc/systemd/journald.conf中的日志占用空间百分比来进行轮转
 - 设置方法
 - ◆ 建立目录/var/log/journal
 - ◆ 修改目录所属及权限
 - # chown root:systemd-journal /var/log/journal;chmod 2755 /var/log/journal
 - ◆ 重启系统或发信号USR1至systemd-journald进程
 - # killall -USR1 systemd-journald
 - 使用journalctl -b 仅显示系统上一次启动以来的日志消息

保持准确的系统时间

- 正确的时间同步，在多个系统环境中是非常重要的
 - ◆ 网络时间协议（NTP）是从网上获得正确时间的标准方法

```
[student@serverX ~]$ timedatectl
      Local time: Thu 2014-02-13 02:16:15 EST
      Universal time: Thu 2014-02-13 07:16:15 UTC
            RTC time: Thu 2014-02-13 07:16:15
           Timezone: America/New_York (EST, -0500)
         NTP enabled: yes
    NTP synchronized: no
      RTC in local TZ: no
        DST active: no
Last DST change: DST ended at
                  Sun 2013-11-03 01:59:59 EDT
                  Sun 2013-11-03 01:00:00 EST
Next DST change: DST begins (the clock jumps one hour forward) at
                  Sun 2014-03-09 01:59:59 EST
                  Sun 2014-03-09 03:00:00 EDT
```

维护系统时间

- 查看系统时钟与时区
 - ◆ 使用timedatectl
 - ◆ 使用timedatectl list-timezones列出可用的时区
- 设定系统时钟与时区
 - ◆ timedatectl set-timezone timezone
 - ◆ timedatectl set-time hour:min:sec
- 设定是否启用网络时间同步
 - ◆ timedatectl set-ntp true|false

维护系统时间

- chrony
 - ◆ RHEL7使用chrony同步时间，不再使用ntp服务
 - ◆ chronyd服务通过与配置的NTP服务器同步
- 配置方式
 - ◆ 修改配置文件：/etc/chrony.conf
 - ◆ 图形界面： system-config-date

指定时钟服务器

- 配置文件/etc/chrony.conf

- ◆ server 指定网络时钟服务器，类似ntp

```
# Use public servers from the pool.ntp.org project.  
# Please consider joining the pool (http://www.pool.ntp.org/join)  
server 0.rhel.pool.ntp.org iburst  
server 1.rhel.pool.ntp.org iburst  
server 2.rhel.pool.ntp.org iburst  
server 3.rhel.pool.ntp.org iburst
```

- 查看当前网络时钟服务器信息

- ◆ chronyc sources -v

请思考

- 配置时间服务器为 classroom.example.com

Q&A



第11章 RHEL网络管理



关注微信公众号
享终身免费培训

目标

- 计算机网络的基本概念
- 查看并测试当前网络配置
- 使用nmcli管理网络
- 通过编辑配置文件修改网络设置
- 配置并测试主机名与名称解析

认识IPv4地址

- IP地址是一个32位的二进制数

- 地址分为两个部分

- ◆ 网络部分
- ◆ 主机部分

IP Address:

172.17.5.3 = 10101100.00010001.00000101.00000011

Prefix: /16

Netmask:

255.255.0.0 = 11111111.11111111.00000000.00000000

10101100.00010001.00000101.00000011

Network

Host

IP Address:

192.168.5.3 = 11000000.10101000.00000101.00000011

Prefix: /24

Netmask:

255.255.255.0 = 11111111.11111111.11111111.00000000

11000000.10101000.00000101.00000011

Network

Host

认识IPv4地址

- 分类

| | Bits: | 1 | 8 9 | 16 17 | 24 25 | 32 |
|-----------------|-------|-----------------|-----------------|-----------------|-----------------|----|
| Class A: | | 0NNNNNNN | Host | Host | Host | |
| | | Range (1-126) | | | | |
| Class B: | Bits: | 10NNNNNN | Network | Host | Host | |
| | | Range (128-191) | | | | |
| Class C: | Bits: | 110NNNNN | Network | Network | Host | |
| | | Range (192-223) | | | | |
| Class D: | Bits: | 1110MMMM | Multicast Group | Multicast Group | Multicast Group | |
| | | Range (224-239) | | | | |

子网掩码的作用

Calculating the network address for 192.168.1.107/24

| | | |
|----------------|---------------------|--|
| Host addr | 192.168.1.107 | 11000000.10101000.00000001.01101011 |
| Network prefix | /24 (255.255.255.0) | 11111111.11111111.11111111.00000000 |
| Network addr | 192.168.1.0 | 11000000.10101000.00000001.00000000 |
| Broadcast addr | 192.168.1.255 | 11000000.10101000.00000001.11111111 |

Calculating the network address for 10.1.1.18/8

| | | |
|----------------|----------------|--|
| Host addr | 10.1.1.18 | 00001010.00000001.00000001.00010010 |
| Network prefix | /8 (255.0.0.0) | 11111111.00000000.00000000.00000000 |
| Network addr | 10.0.0.0 | 00001010.00000000.00000000.00000000 |
| Broadcast addr | 10.255.255.255 | 00001010.11111111.11111111.11111111 |

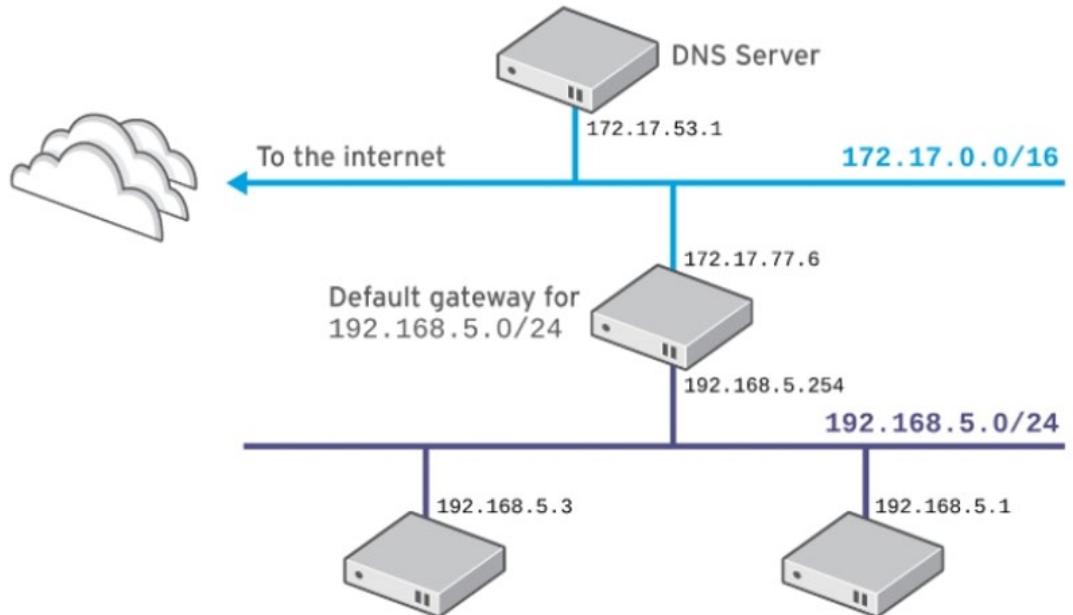
Calculating the network address for 172.16.181.23/19

| | | |
|----------------|---------------------|--|
| Host addr | 172.168.181.23 | 10101100.10101000.10110101.00010111 |
| Network prefix | /19 (255.255.224.0) | 11111111.11111111.11100000.00000000 |
| Network addr | 172.168.160.0 | 10101100.10101000.10100000.00000000 |
| Broadcast addr | 172.168.191.255 | 10101100.10101000.10111111.11111111 |

网关应用

- 网关

- ◆ 当需要和其它网段通信时需要设置网关
- ◆ 网关通常指向本地路由器接口的IP



Example routing table

| Destination | Interface | Router (if needed) |
|---------------------|-----------|--------------------|
| 192.0.2.0/24 | wl01 | |
| 192.168.5.0/24 | enp3s0 | |
| 0.0.0.0/0 (default) | enp3s0 | 192.168.5.254 |

网络接口名称

- 传统上，Linux 的网络接口列举为 eth0、eth1、eth2 ...
- RHEL7 的默认命名是基于固件基础上分配的固定名称，如ens33
 - ◆ 接口类型
 - 以太网有线接口为 en
 - 无线局域网接口为 wl
 - 无线广域网接口为 ww
 - ◆ 适配器类型
 - s为热插拔插槽
 - o为板载
 - p代表对 PCI 类型的支持
 - ◆ 数字N代表索引、ID或端口
- 如果固定名称不能确定，传统的名称如 ethN 将被使用

验证网络配置

1. 接口状态 2. 硬件（MAC）地址 3. IPv4 地址和前缀
4. 广播地址，作用域和设备的名称 5. IPv6 信息

```
[student@desktopX ~]$ ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,①UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    ②link/ether 52:54:00:00:00:0a brd ff:ff:ff:ff:ff:ff
    ③inet 172.25.0.10/24 brd ④172.25.0.255 scope global eth0
        valid_lft forever preferred_lft forever
    ⑤inet6 fe80::5054:ff:fe00:b/64 scope link
        valid_lft forever preferred_lft forever
```

验证网络配置

- ip命令还可以用于显示关于网络性能的统计信息。所接收（RX）和发送（TX）数据包，错误和丢弃计数器可以被用来识别所造成堵塞，内存不足，和超支网络问题

```
[student@desktopX ~]$ ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
link/ether 52:54:00:00:00:0a brd ff:ff:ff:ff:ff:ff
    RX: bytes    packets    errors    dropped    overrun    mcast
      269850        2931        0        0        0        0
    TX: bytes    packets    errors    dropped    carrier    collsns
      300556        3250        0        0        0        0
```

端口与服务

- 1. 监听所有SSH服务IPv4 地址端口；“*” 来表示 “所有的地址”
- 2. 监听SMTP 服务IPv4 环回接口 (127.0.0.1)
- 3. 建立的 SSH 连接，本地172.25.x.10 和对端172.25.254.254
- 4. 监听所有SSH服务IPv6 地址端口；“ :: ” 代表所有 IPv6 接口
- 5. 监听SMTP 服务的IPv6 环回接口 (::1)

```
[student@desktopX ~]$ ss -ta
State      Recv-Q Send-Q      Local Address:Port          Peer Address:Port
LISTEN      0      128          *:sunrpc                  *:*
LISTEN      0      128          ①*:ssh                   *:*
LISTEN      0      100          ②127.0.0.1:smtp          *:*
LISTEN      0      128          *:36889                  *:*
ESTAB       0      0            ③172.25.X.10:ssh        172.25.254.254:59392
LISTEN      0      128          :::sunrpc                 :::*
LISTEN      0      128          ④:::ssh                  :::*
LISTEN      0      100          ⑤:::1:smtp                :::*
LISTEN      0      128          :::34946                  :::*
```

标准服务端口

- TCP用主机的IP地址加上端口号作为TCP连接的端点
- 标准服务端口配置文件

◆ /etc/services

```

# Each line describes one service, and is of the form:
#
# service-name    port/protocol  [aliases ...]  [# comment]

tcpmux          1/tcp
tcpmux          1/udp
rje             5/tcp
rje             5/udp
echo            7/tcp
echo            7/udp
discard         9/tcp      sink null
discard         9/udp      sink null
systat          11/tcp     users
systat          11/udp     users
daytime         13/tcp
daytime         13/udp

```

网络管理工具

- NetworkManager

- ◆ 目的是让linux用户能更容易的应对现在网络管理的需求
- ◆ nmcli 工具

- 查看网络设备
- 创建网络连接
- 修改网络配置

使用nmcli管理网络

- 使用nmcli con show 命令显示主机网卡信息
 - ◆ 使用--active选项 查看活动的设备

```
[root@desktopX ~]# nmcli con show
NAME           UUID                                  TYPE      DEVICE
static-eth0    f3e8dd32-3c9d-48f6-9066-551e5b6e612d 802-3-ethernet  eth0
System eth0    5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03 802-3-ethernet  --
guest         f601ca8a-6647-4188-a431-dab48cc63bf4   802-11-wireless wlp3s0
[root@desktopX ~]# nmcli con show --active
NAME           UUID                                  TYPE      DEVICE
static-eth0    f3e8dd32-3c9d-48f6-9066-551e5b6e612d 802-3-ethernet  eth0
guest         f601ca8a-6647-4188-a431-dab48cc63bf4   802-11-wireless wlp3s0
```

使用nmcli创建网络连接

1. Define a new connection named "default" which will autoconnect as an Ethernet connection on the eth0 device using DHCP.

```
[root@desktopX ~]# nmcli con add con-name "default" type ethernet ifname eth0
```

2. Create a new connection named "static" and specify the IP address and gateway. Do not autoconnect.

```
[root@desktopX ~]# nmcli con add con-name "static" ifname eth0 autoconnect no type ethernet ip4 172.25.X.10/24 gw4 172.25.X.254
```

3. The system will autoconnect with the DHCP connection at boot. Change to the static connection.

```
[root@desktopX ~]# nmcli con up "static"
```

4. Change back to the DHCP connection.

```
[root@desktopX ~]# nmcli con up "default"
```

使用nmcli修改网络配置

1. Turn off autoconnect.

```
[root@desktopX ~]# nmcli con mod "static" connection.autoconnect no
```

2. Specify a DNS server.

```
[root@desktopX ~]# nmcli con mod "static" ipv4.dns 172.25.X.254
```

3. Some configuration arguments may have values added or removed. Add a +/- symbol in front of the argument. Add a secondary DNS server.

```
[root@desktopX ~]# nmcli con mod "static" +ipv4.dns 8.8.8.8
```

4. Replace the static IP address and gateway.

```
[root@desktopX ~]# nmcli con mod "static" ipv4.addresses "172.25.X.10/24  
172.25.X.254"
```

5. Add a secondary IP address without a gateway.

```
[root@desktopX ~]# nmcli con mod "static" +ipv4.addresses 10.10.10.10/16
```

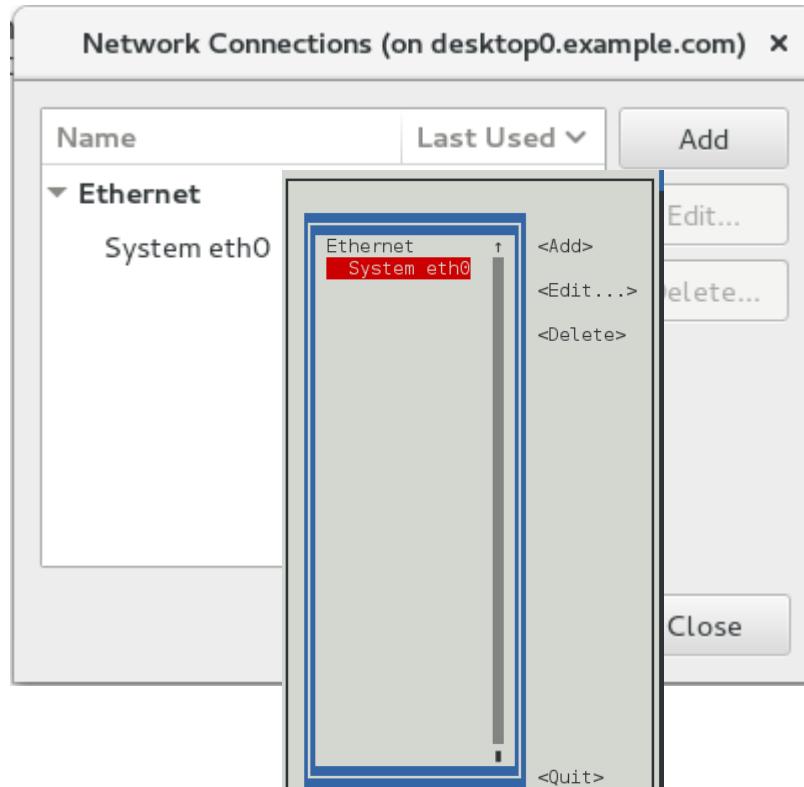
nmcli 命令总结

| Command | Use |
|--------------------------|---|
| nmcli dev status | List all devices. |
| nmcli con show | List all connections. |
| nmcli con up "<ID>" | Activate a connection. |
| nmcli con down "<ID>" | Deactivate a connection. The connection will restart if autoconnect is yes. |
| nmcli dev dis <DEV> | Bring down an interface and temporarily disable autoconnect. |
| nmcli net off | Disable all managed interfaces. |
| nmcli con add ... | Add a new connection. |
| nmcli con mod "<ID>" ... | Modify a connection. |
| nmcli con del "<ID>" | Delete a connection. |

管理网卡配置

- 图形化管理工具

- ◆ nm-connection-editor
- ◆ nmtui-edit



网卡配置文件

- 配置文件路径

- ◆ /etc/sysconfig/network-scripts/ifcfg-<name>

| Static | Dynamic | Either |
|-----------------------|----------------|--------------------|
| BOOTPROTO=none | BOOTPROTO=dhcp | DEVICE=eth0 |
| IPADDR0=172.25.X.10 | | NAME="System eth0" |
| PREFIX0=24 | | ONBOOT=yes |
| GATEWAY0=172.25.X.254 | | UUID=f3e8dd32-3... |
| DEFROUTE=yes | | USERCTL=yes |
| DNS1=172.25.254.254 | | |

nmcli 配置与 ifcfg-* 配置文件

- nmcli配置与ifcfg-*配置文件对比

| nmcli con mod | ifcfg-* | 目的 |
|---|---|----------------------------|
| ipv4.method manual 或static | BOOTPROTO=none或static | 静态指定IPv4地址 |
| ipv4.method auto | BOOTPROTO=dhcp | 使用DHCPv4配置, |
| ipv4.addresses "192.0.2.1/24 192.0.2.254 " | IPADDR0=192.0.2.1 PREFIX0=24 GATEWAY0=192.0.2.254 | 设定IPv4地址, 前缀及网关, 可配置多个值 |

nmcli配置与ifcfg-*配置文件

- nmcli配置与ifcfg-*配置文件对比

| nmcli con mod | ifcfg-* | 目的 |
|---------------------------------|--------------|---------------------|
| ipv4.dns 8.8.8.8 | DNS1=8.8.8.8 | 指定DNS服务器地址 |
| connection.id eth0 | NAME=eth0 | 配置网络连接名 |
| connection.interface-name eth0 | DEVICE=eth0 | 配置网络连接所绑定的网络接口名 |
| 802-3-ethernet.mac-address XXXX | HWADDR=XXXX | 配置网络连接所绑定的网络接口MAC地址 |

主机名

- 查看主机名

```
[root@desktopX ~]# hostname  
desktopX.example.com
```

- 主机名配置文件 /etc/hostname

- 配置主机名

```
[root@desktopX ~]# hostnamectl set-hostname desktopX.example.com  
[root@desktopX ~]# hostnamectl status  
    Static hostname: desktopX.example.com  
          Icon name: computer  
            Chassis: n/a  
        Machine ID: 9f6fb63045a845d79e5e870b914c61c9  
          Boot ID: aa6c3259825e4b8c92bd0f601089ddf7  
Virtualization: kvm  
Operating System: Red Hat Enterprise Linux Server 7.0 (Maipo)  
      CPE OS Name: cpe:/o:redhat:enterprise_linux:7.0:beta:server  
        Kernel: Linux 3.10.0-97.el7.x86_64  
Architecture: x86_64  
[root@desktopX ~]# cat /etc/hostname  
desktopX.example.com
```

验证网络配置

- 本地解析/etc/hosts
- 域名解析配置文件

```
[root@desktopX ~]# cat /etc/resolv.conf
# Generated by NetworkManager
domain example.com
search example.com
nameserver 172.25.254.254
```

- 使用host命令测试主机名解析
 - ◆ host hostname|ip
- 使用nslookup测试域名解析
 - ◆ nslookup hostname|ip

请思考

- 配置主机名为desktop0.example.com
- 配置网卡信息
 - ◆ IP: 172.25.0.10/24
 - ◆ 网关: 172.25.0.254
 - ◆ DNS: 172.25.254.254

Q&A



第12章 归档与系统间复制文件



关注微信公众号
享终身免费培训

目标

- 使用tar创建与释放归档文件
- 通过sshd在系统间安全拷贝文件
- 在本地及远端安全同步文件

归档及压缩

- 什么是tar
 - ◆ 归档是将多个文件汇集成为一个文件
 - ◆ 用于备份、打包存储
 - ◆ 支持使用gzip或bzip2或xz压缩

归档及压缩

- tar命令

- ◆ 用途：制作归档文件、释放归档文件

- ◆ 执行以下三种操作之一

- c：创建一个归档
 - t：列出归档的内容
 - x：提取存档

- ◆ 格式：

- tar [选项]... 归档文件名 源文件或目录
 - tar [选项]... 归档文件名 [-C 目标目录]

- ◆ 常用选项

- v：输出详细信息
 - f：表示使用归档文件
 - p：提取存档文件时保留原始文件及目录的权限

创建压缩的tar归档文件

- tar可以选择以下选项进行压缩
 - ◆ -z: 调用gzip压缩 (filename.tar.gz 或 filename.tarz)
 - ◆ -j: 调用bzip2压缩 (filename.tar.bz2)
 - ◆ -J: 调用xz压缩 (filename.tar.xz)

Create (c option) a gzip-compressed (z option) tar archive **/root/etcbackup.tar.gz** of the **/etc** directory on serverX:

```
[root@serverX ~]$ tar czf /root/etcbackup.tar.gz /etc
```

Create (c option) a bzip2-compressed (j option) tar archive **/root/logbackup.tar.bz2** of the **/var/log** directory on serverX:

```
[root@serverX ~]$ tar cjf /root/logbackup.tar.bz2 /var/log
```

请思考

- 将/usr/local中的所有文件归档到/var/tmp/backup.tar.bz2

使用scp实现远程文件传输

- scp命令

- ◆ 格式1： scp localpath [user@]deshost:/path

- ◆ 格式2： scp [user@]despath:/path localpath

This example shows how to copy the local files on desktopX, **/etc/yum.conf** and **/etc/hosts**, securely to the account student on the remote system serverX into the directory **/home/student/**:

```
[student@desktopX ~]$ scp /etc/yum.conf /etc/hosts serverX:/home/student
student@serverX's password: student
yum.conf                                100%   813      0.8KB/s  00:00
hosts                                    100%   227      0.2KB/s  00:00
```

A user can copy a file from a remote account on a remote machine to the local file system with **scp**. In this example, copy the file **/etc/hostname** from the account student on the serverX machine to the local directory **/home/student/**.

```
[student@desktopX ~]$ scp serverX:/etc/hostname /home/student/
student@serverX's password: student
hostname                               100%    22      0.0KB/s  00:00
```

使用sftp实现远程文件传输

- sftp工具

- ◆ 使用SSH传输，交互命令与ftp工具相同

```
[student@desktopX ~]$ sftp serverX
student@serverX's password: student
Connected to serverX.
sftp>
```

```
sftp> mkdir hostbackup
sftp> cd hostbackup
sftp> put /etc/hosts
Uploading /etc/hosts to /home/student/hostbackup/hosts
/etc/hosts                                         100%  227      0.2KB/s   00:00
sftp>
```

```
sftp> get /etc/yum.conf
Fetching /etc/yum.conf to yum.conf
/etc/yum.conf                                         100%  813      0.8KB/s   00:00
sftp> exit
[student@desktopX ~]$
```

使用rsync实现同步文件内容

- rsync命令

- ◆ 如果两个系统间的目录结构相似，rsync仅复制其差异部分

```
◀ [root@desktopX ~]# rsync -av /var/log/ /tmp  
...
```

```
◀ [root@desktopX ~]$ rsync -av /var/log serverX:/tmp  
root@serverX's password: redhat  
...
```

In the same way, the remote folder `/var/log` on serverX can be synchronized to the local directory `/tmp` on desktopX:

```
[root@desktopX ~]$ rsync -av serverX:/var/log /tmp  
root@serverX's password: redhat  
...
```

Q&A



第13章 安装与升级软件包



关注微信公众号
享终身免费培训

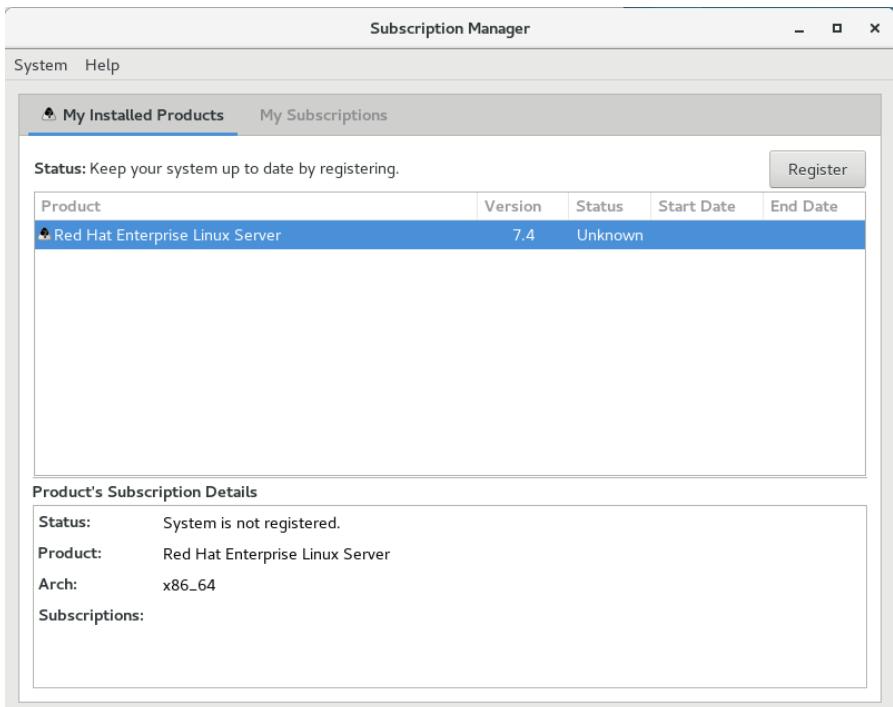
目标

- 在线注册升级
- 使用RPM工具管理软件包
- 使用YUM仓库管理软件包

红帽在线订阅

- 使用图形工具在线订阅

◆ **subscription-manager-gui**



RPM软件包管理

- RPM Package Manager
 - ◆ 由Red Hat公司提出，被众多Linux发行版所采用
- RPM软件包
 - ◆ 一般命令格式：

bash-4.2.46-28.el7.x86_64.rpm



该图展示了RPM软件包名称的组成部分：bash-4.2.46-28.el7.x86_64.rpm。图中用黄色气泡标注了各部分：'bash'标注为'软件名称'；'4.2.46-28'标注为'版本号'；'el7'标注为'硬件平台'；'28'标注为'修订版本号'；'.rpm'标注为'扩展名'" data-bbox="226 602 744 825"/>

RPM包管理命令

- rpm命令

- ◆ 主要功能

- 查询RPM软件、包文件的相关信息
 - 已安装格式: rpm -q [子选项] [软件名]
 - 常用选项: -a、 -i、 -l、 -c、 -f、 --scripts
 - 未安装格式: rpm -qp[子选项] RPM包文件
 - 安装、升级、卸载RPM软件包
 - 格式: rpm [选项] RPM包文件...
 - 常用选项: -i、 -U、 -e
 - 辅助选项: -h、 -v、 --force

请思考

- http://172.25.254.254/content/rhel7.0/x86_64/errata/Packages/
下有适合的 kernel, 请安装更新

解决软件包依赖关系

- 安装有依赖关系的多个软件时
 - ◆ 被依赖的软件包需要先安装
 - ◆ 同时指定多个.rpm包文件进行安装
- 卸载时，依赖其他程序的软件包需要先卸载
- 忽略依赖关系
 - ◆ 结合 “--nodeps” 选项，但可能导致软件异常

vim-enhanced



vim-common

YUM仓库管理软件包工具

- Yum (全称为 Yellow dog Updater, Modified)
- 是一个基于RPM包的软件包管理器
- 能够从指定的服务器自动下载RPM包并且安装，自动处理依赖性关系，并且一次安装所有依赖的软件包
- C/S模式
 - ◆ Server 服务端提供RPM软件包与数据库文件repodata
 - ◆ Client 客户端使用YUM仓库

YUM仓库客户端配置

- 配置文件： /etc/yum.repos.d/*.repo

◆ 格式：

- [*BASIC*] 仓库ID
- name= *BASIC* 仓库名
- baseurl= 仓库地址
 - 本地地址 file://本地软件包路径
 - FTP服务器地址 ftp://仓库位置
 - WEB服务器地址 http://仓库位置
- enabled = 0|1 是否开启仓库
- gpgcheck= 0|1 是否需要公钥验证
- gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release 公钥验证地址

使用命令行工具添加YUM仓库

- yum-config-manager
 - ◆ 选项 --add-repo=YUM仓库地址
 - 如: # yum-config-manager --add-repo=http://classroom.example.com/content/rhel7.0/x86_64/dvd
- 导入公钥
 - ◆ # rpm --import 仓库公钥地址
- 验证仓库是否配置正确
 - ◆ # yum clean all
 - ◆ # yum repolist

YUM相关命令

- yum help 查询帮助

| Task: | Command: |
|---|---|
| List installed and available packages by name | <code>yum list [NAME-PATTERN]</code> |
| List installed and available groups | <code>yum grouplist</code> |
| Search for a package by keyword | <code>yum search KEYWORD</code> |
| Show details of a package | <code>yum info PACKAGE NAME</code> |
| Install a package | <code>yum install PACKAGE NAME</code> |
| Install a package group | <code>yum groupinstall "GROUPNAME"</code> |
| Update all packages | <code>yum update</code> |
| Remove a package | <code>yum remove PACKAGE NAME</code> |
| Display transaction history | <code>yum history</code> |

YUM命令示例

- 使用yum命令列出与搜索软件包

```
[root@serverX ~]# yum list 'http*'
Loaded plugins: langpacks
Available Packages
httpcomponents-client.noarch          4.2.5-4.el7      rhel_dvd
httpcomponents-core.noarch            4.2.4-6.el7      rhel_dvd
httpd.x86_64                          2.4.6-17.el7    rhel_dvd
httpd-devel.x86_64                   2.4.6-17.el7    rhel_dvd
httpd-manual.noarch                  2.4.6-17.el7    rhel_dvd
httpd-tools.x86_64                   2.4.6-17.el7    rhel_dvd
```

```
[root@serverX ~]# yum search all 'web server'
Loaded plugins: langpacks
=====
Matched: web server =====
freeradius.x86_64 : High-performance and highly configurable free RADIUS server
hsqldb.noarch : HyperSQL Database Engine
httpd.x86_64 : Apache HTTP Server
libcurl.i686 : A library for getting files from web servers
libcurl.x86_64 : A library for getting files from web servers
mod_revocator.x86_64 : CRL retrieval module for the Apache HTTP server
mod_security.x86_64 : Security module for the Apache HTTP Server
python-paste.noarch : Tools for using a Web Server Gateway Interface stack
```

YUM命令示例

- 查询与指定路径名（通常含通配符）匹配的软件包

To find packages that provide the `/var/www/html` directory, use:

```
[root@serverX ~]# yum provides /var/www/html
Loaded plugins: langpacks
httpd-2.4.6-17.el7.x86_64 : Apache HTTP Server
Repo      : rhel_dvd
Matched from:
Filename  : /var/www/html

1:php-pear-1.9.4-21.el7.noarch : PHP Extension and Application Repository
                               : framework
Repo      : rhel_dvd
Matched from:
Filename  : /var/www/html
```

YUM事物历史记录

- 所有安装和删除事务的日志记录在/var/log/yum.log中
- 查看安装和删除事务的摘要
 - ◆ yum history
- 查看某个事务详细信息（N是事务编号）
 - ◆ yum history info N
- 历史中搜索某个软件包
 - ◆ yum history list 包名
- 撤销事务
 - ◆ yum history undo N
- 重做事务
 - ◆ yum history redo N

使用第三方软件仓库

- 第三方软件仓库是指非红帽来源提供的软件仓库
 - ◆ 如果知道url可以直接使用yum-config-manager 添加
 - ◆ 存储库的RPM配置软件包

```
[root@serverX ~]# rpm --import http://dl.fedoraproject.org/pub/epel/RPM-GPG-KEY-EPEL-7
[root@serverX ~]# yum install http://dl.fedoraproject.org/pub/epel/beta/7/x86_64/epel-
release-7-0.1.noarch.rpm
```

```
[EPEL]
name=EPEL 7 Beta
baseurl=http://dl.fedoraproject.org/pub/epel/beta/7/x86_64/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
```

请思考

- 按照要求建立 yum 软件仓库
- 这个 yum 仓库为默认仓库
- 地址为

http://classroom.example.com/content/rhel7.0/x86_64/dvd

Q&A



第14章 访问Linux文件系统



关注微信公众号
享终身免费培训

目标

- 识别文件系统层次结构
- 访问文件系统内容
- 使用硬链接及符号链接
- 搜索挂载的文件系统中的文件

文件系统

- 什么是文件系统?
 - ◆ 操作系统用于明确存储设备或分区上的文件的方法和数据结构
 - ◆ 是在存储设备上组织文件的方法
 - ◆ 包含保存数据的文件和目录的有序结构
- 将新的文件系统添加到现有目录树的过程称为挂载
- 挂载了新文件系统的目录称为挂载点

设备文件

- Linux中将硬盘、分区等设备均表示为文件
- 存储设备由一个特殊文件类型表示，称其块设备

硬件设备文件所在的目录

hd 表示IDE设备
sd 表示SCSI/SATA/PATA设备

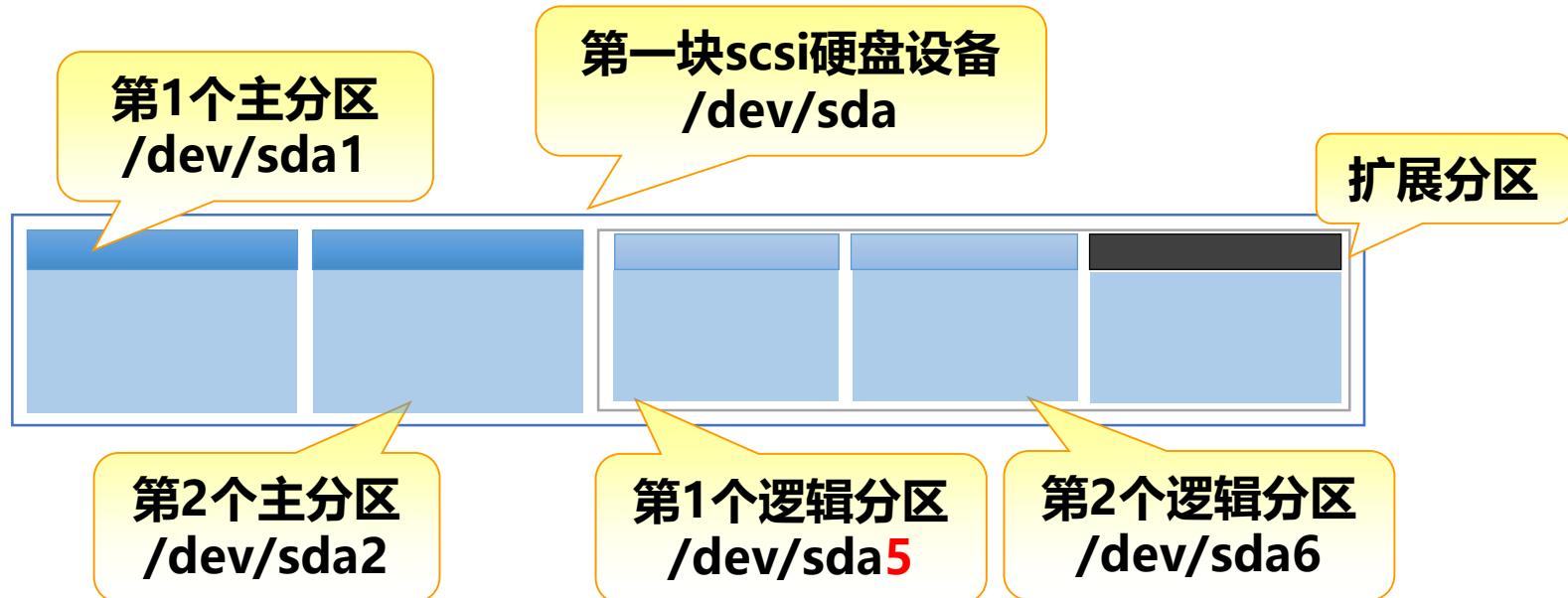
/dev/sda1

硬盘的顺序号，以字母a、b、c.....表示

分区的顺序号，以数字1、2、3.....15表示

磁盘和分区结构

- Linux分区的表示方法



使用文件系统

- 挂载文件系统：mount命令

- ◆ 用途：挂载文件系统到指定目录

- 命令格式
 - mount [-t type] 设备文件 挂载点
 - mount UUID="UUID" 挂载点
 - 可使用blkid查询块设备的唯一ID

Mount by device file of the partition that holds the file system.

```
[root@serverX ~]# mount /dev/vdb1 /mnt/mydata
```

Mount the file system by universal unique id, or the UUID, of the file system.

```
[root@serverX ~]# mount UUID="46f543fd-78c9-4526-a857-244811be2d88" /mnt/mydata
```

查看文件系统

- df命令

- ◆ 查看文件系统挂载点及可用空间

- 常用选项
 - -h: 单位为KiB(2^{10})、MiB(2^{20})或GiB(2^{30})...
 - -H: 单位为KiB(10^3)、MiB(10^6)或GiB(10^9)...

```
[student@serverX ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1        6.0G  3.9G  2.2G  65% /
devtmpfs         929M    0  929M   0% /dev
tmpfs            937M   80K  937M   1% /dev/shm
tmpfs            937M   2.2M  935M   1% /run
tmpfs            937M    0  937M   0% /sys/fs/cgroup
```

查看文件系统

- du命令

- ◆ 统计目录及文件的空间占用情况

- 常用选项
 - -s: 统计当前目录的大小

```
[root@serverX ~]# du -h /var/log
...
4.9M    /var/log/sa
68K     /var/log/prelink
0       /var/log/qemu-ga
14M    /var/log
```

使用文件系统

● 卸载文件系统：umount命令

- 用途：卸载已挂载的文件系统
- 命令格式
 - umount 设备文件
 - umount 挂载点

取消挂载示例

To unmount a file system, the **umount** command expects the mount point as an argument.

Change to the **/mnt/mydata** directory. Try to umount the device mounted on the **/mnt/mydata** mount point. It will fail.

```
[root@serverX ~]# cd /mnt/mydata
[root@serverX mydata]# umount /mnt/mydata
umount: /mnt/mydata: target is busy.
        (In some cases useful info about processes that use
         the device is found by lsof(8) or fuser(1))
```

Unmounting is not possible if the mount point is accessed by a process. For **umount** to be successful, the process needs to stop accessing the mount point.

The **lsof** command lists all open files and the process accessing them in the provided directory. It is useful to identify which processes currently prevent the file system from successful unmounting.

```
[root@serverX mydata]# lsof /mnt/mydata
COMMAND  PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
bash    1593 root cwd   DIR  253,2       6  128 /mnt/mydata
lsof    2532 root cwd   DIR  253,2      19  128 /mnt/mydata
lsof    2533 root cwd   DIR  253,2      19  128 /mnt/mydata
```

Once the processes are identified, an action can be taken, such as waiting for the process to complete or sending a SIGTERM or SIGKILL signal to the process. In this case, it is sufficient to change the current working directory to a directory outside the mount point.

```
[root@serverX mydata]# cd
[root@serverX ~]# umount /mnt/mydata
```

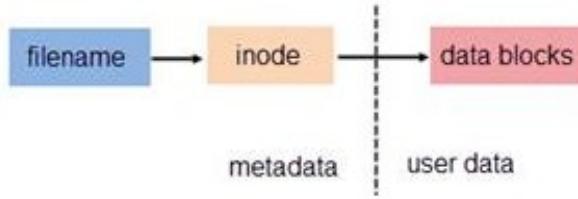
挂载练习

- 重置 server0 主机，使用 lab fs setup 设置为实验环境

链接文件

- Linux上的文件由两部分组成：
 - ◆ 文件数据块：记录文件真实内容
 - ◆ 元数据：文件附加属性 (`ls -l`)
 - 元数据中的inode号是文件的唯一标识 (`ls -i`)

- 链接文件：为了解决文件的共享使用
 - ◆ 软链接：用户数据块中存放内容是另一文件的路
 - 软链接又称符号链接
 - 若被指向的原文件被删除，则相关软连接被称为死链接
 - ◆ 硬链接：一个inode号对应多个文件名，即同一文件多个别名
 - 不能为目录建立硬链接文件
 - 硬链接与原始文件必须位于同一分区（文件系统）中
 - 删除一个硬链接不影响其它相同inode号的文件



文件链接

● ln命令

◆ 用途：为文件或目录建立链接（Link）

◆ 命令格式

- ln [-s] 源文件或目录... 链接文件或目标目录

- 默认建立硬链接

- -s 建立软链接文件

文件查找

- locate命令

- ◆ 由已生成的数据库中基于文件名及目录查找

- ◆ 命令格式： locate [选项] 文件名；

- -i 选项忽略大小写

- . Search for files with "passwd" in the name or path in directory trees readable by user student on serverX.

```
[student@serverX ~]$ locate passwd
/etc/passwd
/etc/passwd-
/etc/pam.d/passwd
/etc/security/opasswd
/usr/bin/gpasswd
/usr/bin/grub2-mkpasswd-pbkdf2
/usr/bin/lppasswd
/usr/bin/passwd
/usr/bin/userpasswd
/usr/bin/vino-passwd
/usr/bin/vncpasswd
```

文件查找

- **find命令**

- ◆ 用于实时查找文件或目录
- ◆ 命令语法

- find [查找范围] [查找条件及选项] [-type 查找类型] [动作]

```
[root@serverX ~]# find / -name sshd_config
/etc/ssh/sshd_config
```

```
[root@serverX ~]# find / -name '*.txt'
/etc/pki/nssdb/pkcs11.txt
/etc/brltty/brl-lt-all.txt
/etc/brltty/brl-mb-all.txt
/etc/brltty/brl-md-all.txt
/etc/brltty/brl-mn-all.txt
...
```

文件查找

● 常用find查找条件

- ◆ -name 按文件名称查找
- ◆ -iname 忽略大小写按文件名称查找
- ◆ -uid/ -gid 按所UID/GID查找
- ◆ -user/ -group 按文件拥有者/所属组查找
- ◆ -perm 按权限查找
- ◆ -size 按文件大小查找 (k , M , G , + 大于, - 小于)
- ◆ -exec command \;
 - 后接命令, 操作查找到的文件
 - 在command中使用{}代表查找到的文件

请思考

- 找出harry用户拥有的文件，拷贝到目录/opt/finddir

Q&A



第15章 使用虚拟系统



关注微信公众号
享终身免费培训

目标

- 了解虚拟化
- 认识KVM虚拟机
- 在RHEL系统上运行虚拟机

什么是虚拟化

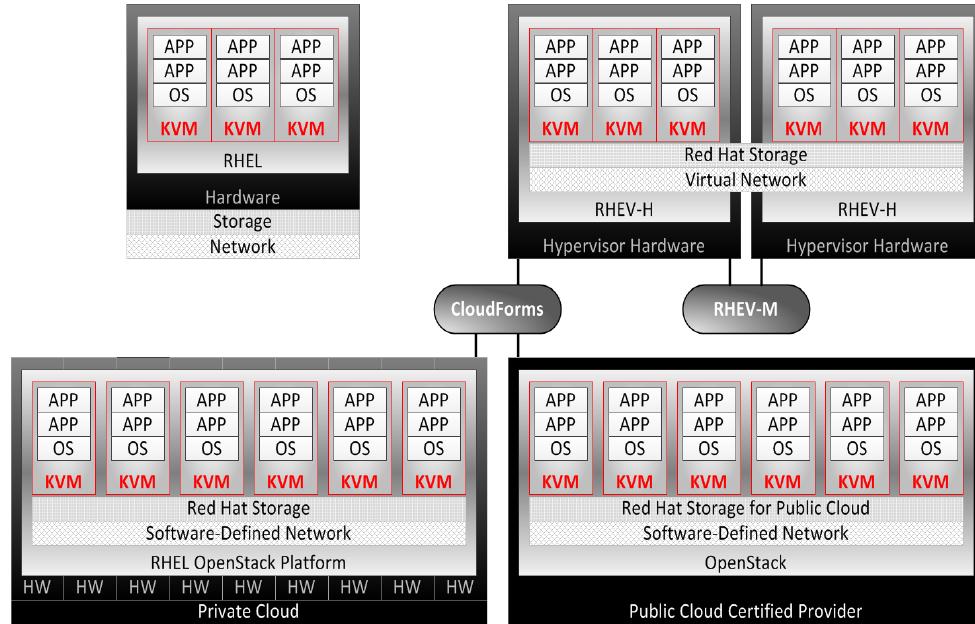
- 虚拟化 (Virtualization) 是一种资源管理技术
 - ◆ 将计算机的各种实体资源，如服务器、网络、内存及存储等，予以抽象、转换后呈现出来
 - ◆ 这些资源的新虚拟部份是不受现有资源的架设方式，地域或物理组态所限制

系统虚拟化

- 表现在单一硬件上运行多个操作系统
- 虚拟操作系统可以同时运行
- 每个操作系统之间相互独立

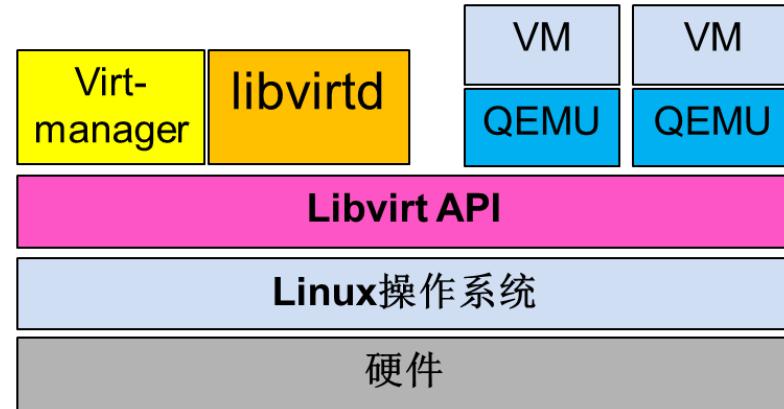
系统虚拟化

- KVM是红帽云基础架构多种产品的一致组件



什么是KVM

- KVM是基于虚拟化扩展（Intel VT 或AMD-V）的X86硬件，的Linux全虚拟化解决方案。
- virt-manager是用于管理KVM虚拟环境的主要工具
 - ◆ Libvirt是一个软件集合便于使用者管理虚拟机和其他虚拟化功能

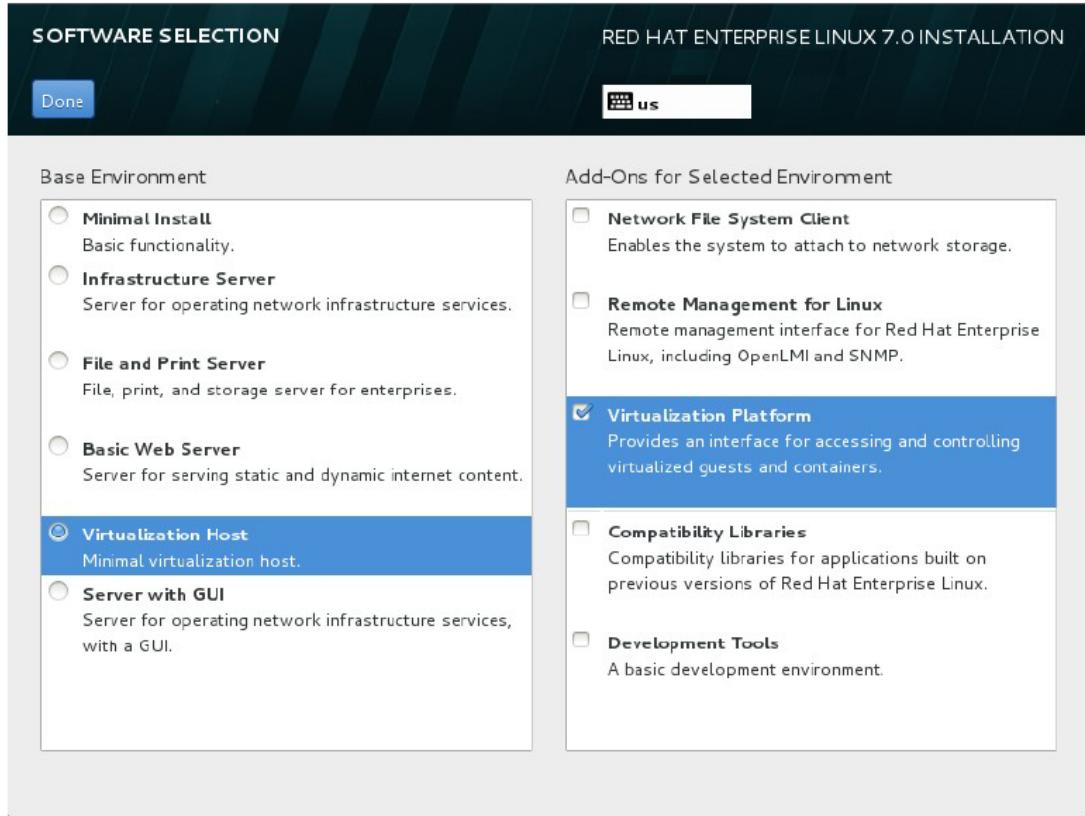


推荐安装系统要求

- 一个核心供主机使用，以及满足虚拟机中最大化CPU数量的需求
- 2GB 的 RAM，以及供虚拟机使用的额外 RAM
- 主机 6GB 磁盘空间，以及供每台虚拟机使用的必要磁盘空间。

大多数客户机操作系统需要至少 6GB 磁盘空间，但实际存储空间要求取决于各个虚拟客户机的镜像格式

安装系统时安装虚拟环境



虚拟化环境配置KVM虚拟机

● 虚拟化管理器 virt-manager

New VM

Create a new virtual machine
Step 1 of 5

Enter your virtual machine details

Name:

Connection: localhost (QEMU/KVM)

Choose how you would like to install the operating system

Local install media (ISO image or CDROM)
 Network Install (HTTP, FTP, or NFS)
 Network Boot (PXE)
 Import existing disk image

New VM

Create a new virtual machine
Step 2 of 5

Provide the operating system install URL

URL:

▼ URL Options

Kickstart URL:

Kernel options:

Automatically detect operating system based on install media

OS type: Linux
Version: Red Hat Enterprise Linux 7

New VM

Create a new virtual machine
Step 3 of 5

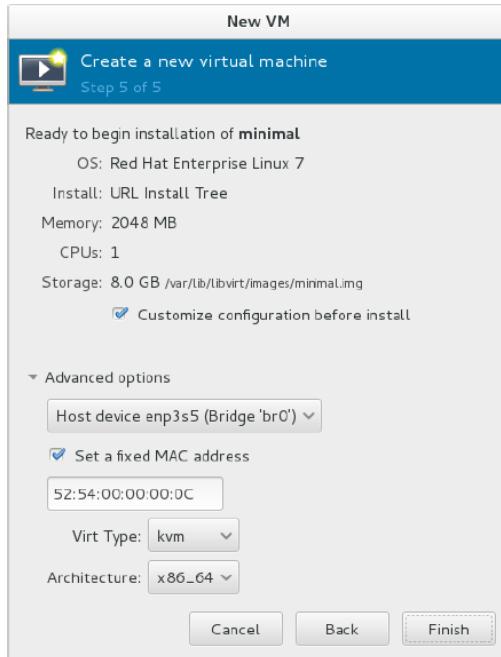
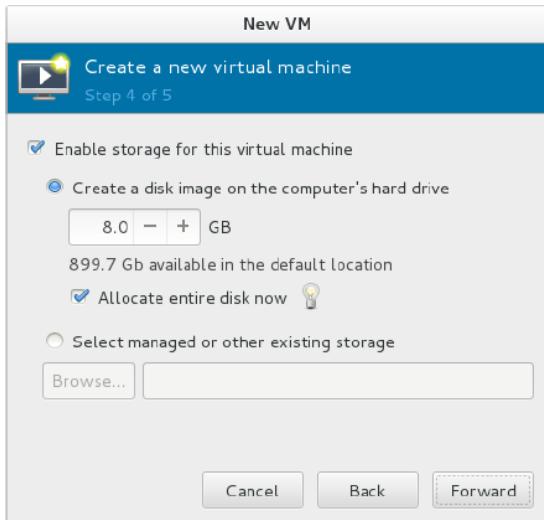
Choose Memory and CPU settings

Memory (RAM): MB
Up to 15635 MB available on the host

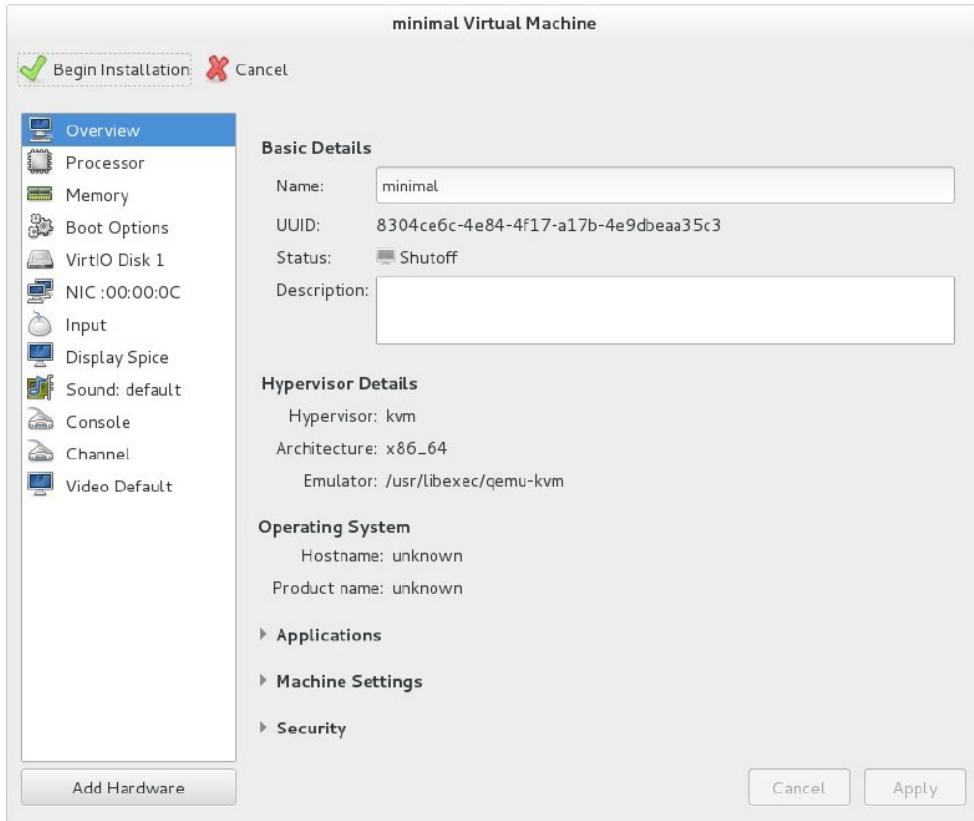
CPUs: Up to 6 available

虚拟化环境配置KVM虚拟机

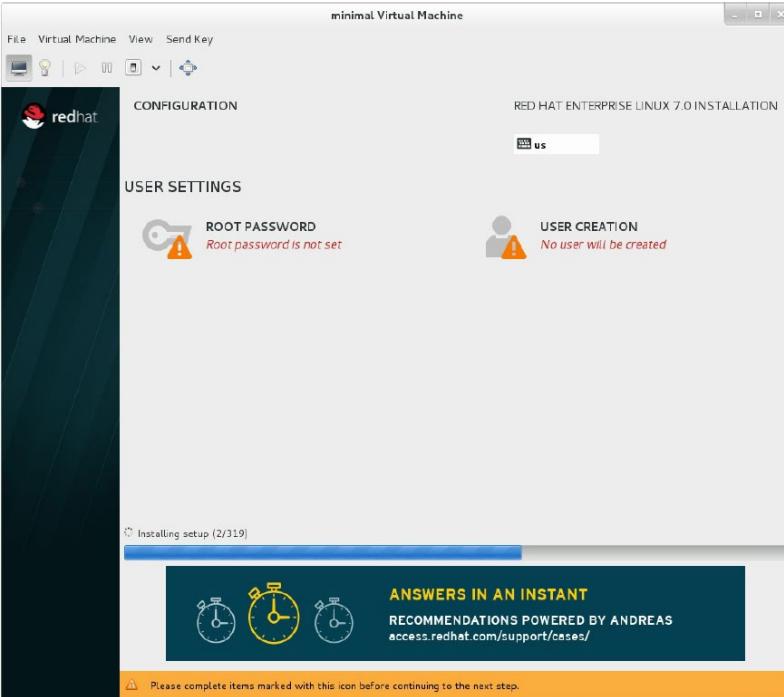
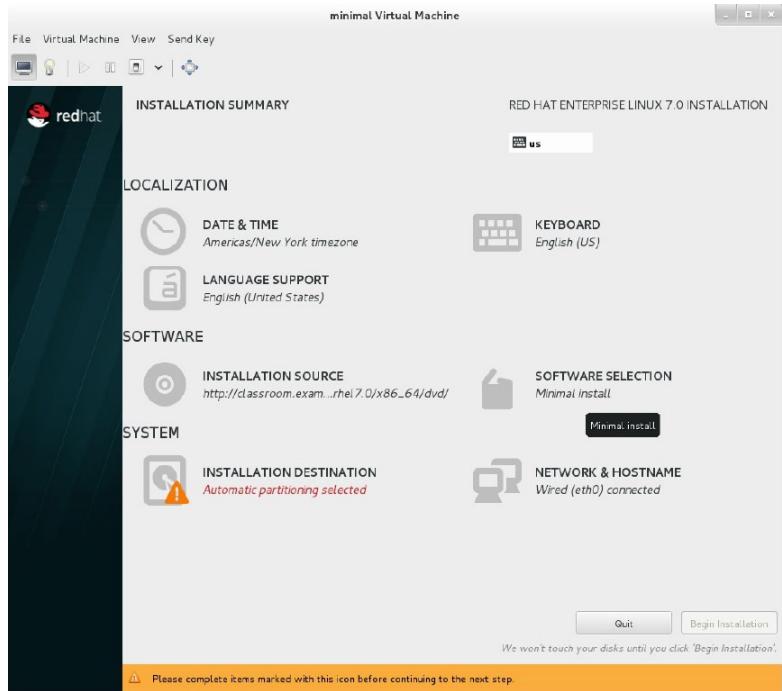
● 虚拟化管理器 virt-manager



管理KVM虚拟机硬件



安装操作系统



Q&A

Thank You

全国统一咨询热线：400-690-6115

北京|上海|广州|深圳|天津|成都|重庆|武汉|济南|青岛|杭州|西安

easthome.com