第一章　pxe+kickstart 无人值守安装 RHEL7

在大规模的 Linux 应用环境中，如 Web 群集、分布式计算等，服务器往往并不配备光驱设备，这种情况下如何为数十甚至上百台服务器裸机快速安装系统？传统的 USB 光驱、移动硬盘等安装方法显然已经力所难及。

本单元将学习基于 PXE(Pre-boot Execution Environment,预启动执行环境)技术的网络装机方法并结合 Kickstart 配置实现无人值守自动安装。

1.1 部署 PXE 远程安装服务

PXE 是由 Intel 公司开发的网络引导技术,工作在 Client/Server 模式，允许客户机通过网络从远程服务器下载引导镜像，并加载安装文件或者整个操作系统，若要搭建 PXE 网络体系，必须满足以下几个前提条件。

- 客户机的网卡支持 PXE 协议（集成 BOOTROM 芯片），且主板支持网络引导
- 网络中有一台 DHCP 服务器以便为客户机自动分配地址，指定引导文件位置。
- 服务器通过 TFTP（Trivial File Protocol，简单文字传输协议）提供引导镜像文件的下载。

1.1.1　搭建 PXE 远程安装服务器(10.11.21.236/23)

本例的 PXE 远程安装服务器集成了 RHEL7 安装源，TFTP 服务、DHCP 服务，能够向客户裸机发送 PXE 引导程序，Linux 内核，启动菜单等数据，以及提供安装文件。

1、准备 RHEL7 安装源

RHEL7 的网络安装源一般通过 HTTP、FTP 协议发布，下面以 FTP 为例。

```
[root@localhost ~]# cat /etc/yum.repos.d/base.repo
[base]
name=base
baseurl=file:///mnt
enabled=1
gpgckeck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[root@localhost ~]# yum -y install vsftpd
[root@localhost ~]# systemctl start vsftpd.service          //启动 ftp 服务
[root@localhost ~]# systemctl enable vsftpd.service         //设置开机自动启动
[root@localhost ~]# systemctl stop iptables.service         //关闭防火墙
[root@localhost ~]# systemctl stop firewalld.service
[root@localhost ~]# setenforce 0
```

```
[root@localhost ~]# mkdir /var/ftp/rhel7
[root@localhost ~]# cp -rf /mnt/*  /var/ftp/rhel7/
```

2、安装并启用 TFTP 服务

TFTP 服务是由 tftp-server 软件包提供，默认由 xinetd 超级服务进行管理，因此配置文件位于/etc/xinetd.d/tftp。配置时只要将“disable= yes”改为“disable= no”，然后启动 xinetd 服务即可。

```
[root@localhost ~]# yum -y install tftp-server
[root@localhost ~]# vim /etc/xinetd.d/tftp
        ……
        server_args             = -s /var/lib/tftpboot
        disable                 = no
        ……
[root@localhost ~]#  systemctl start xinetd.service
[root@localhost ~]# systemctl enable xinetd.service
```

3、准备 Linux 内核、初始化镜像文件。

用于 PXE 网络安装的 Linux 内核、初始化镜像文件可以从 RHEL7 系统光盘获得，分别为 vmlinuz 和 initrd.img，位于文件夹 images/pxeboot/中。找到这两个文件并将其复制到 tftp 服务的根目录下。

```
[root@localhost ~]# cd /mnt/images/pxeboot/
[root@localhost pxeboot]# cp vmlinuz  initrd.img  /var/lib/tftpboot/
```

4、准备 PXE 引导程序、启动菜单文件

用于 PXE 网络安装的引导程序为 pxelinux.0 由软件包 syslinux 提供，安装好软件包 syslinux,然后将文件 pxelinux.0 也复制到 tftp 服务的根目录下。

```
[root@localhost ~]# yum -y install syslinux
[root@localhost ~]# cp /usr/share/syslinux/pxelinux.0  /var/lib/tftpboot/
```

启用菜单用来指导客户机的引导过程，包括如何调用内核，如何加载初始化镜像。默认的启动菜单文件为 default，应放置在 tftp 根目录的 pxelinux.cfg 子目录中，典型的启动菜单配置可参考以下操作手动建立。

```
[root@localhost ~]# mkdir /var/lib/tftpboot/pxelinux.cfg
[root@localhost ~]# vi /var/lib/tftpboot/pxelinux.cfg/default
default auto                                          //指定默认入口名称
prompt 1                                              //1 表示等待用户的控制
label auto
   kernel vmlinuz
  append initrd=initrd.img  inst.repo=ftp://192.168.2.3/rhel7
```

5、安装并启用 DHCP 服务

由于 PXE 客户机通常是尚未安装系统的裸机，因此为了与服务器取得联系并正确下载相关引导文件，需要预先配置好 DHCP 服务来自动分配地址并告知引导文件位置。若 PXE 服务器的 IP 地址为 10.11.21.236，DHCP 地址池为 10.11.21.10~10.11.21.100 则可以参考以下操作来搭建 DHCP 服务器。

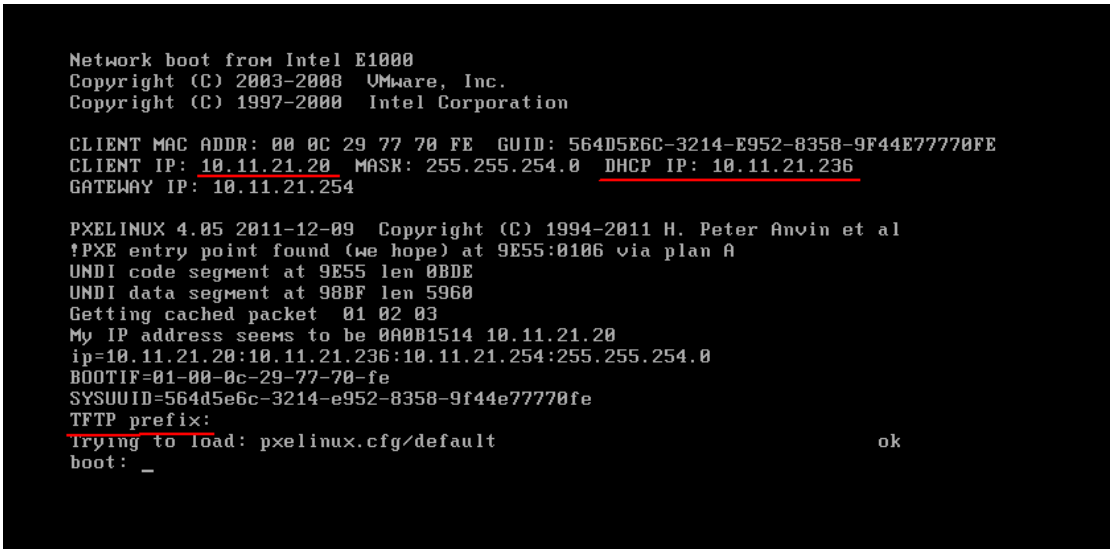
```
[root@localhost ~]# yum -y install dhcp
[root@localhost ~]# vi /etc/dhcp/dhcpd.conf
```

```
……
option domain-name-servers 8.8.8.8;
subnet 10.11.20.0netmask 255.255.254.0 {
      range 10.11.21.10 10.11.21.100;
      option routers 10.11.21.254;
          next-server 10.11.21.236;                    //指定 TFTP 服务器的地址
    filename "pxelinux.0";                            //指定 PXE 引导的程序的文件名
}
[root@localhost ~]# dhcpd -t
[root@localhost ~]#  systemctl enable dhcpd.service
[root@localhost ~]# systemctl start dhcpd.service
```

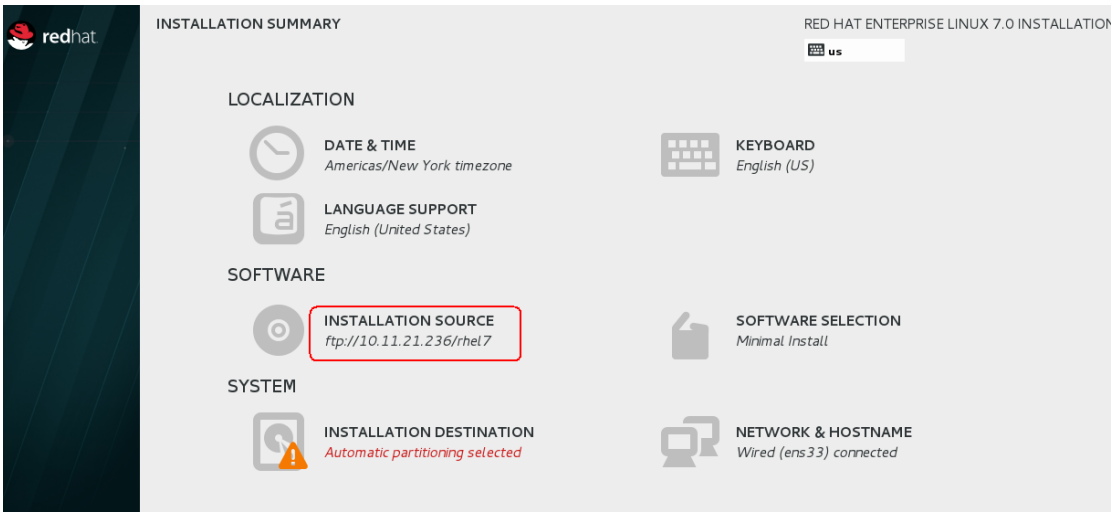
1.1.2 开客户机验证 PXE 网络安装

搭建好 PXE 远程安装服务器以后，就可以使用客户机进行安装测试了。对于新购买的服务器或 PC 裸机，一般不需要额外设置；若要为已有系统的主机重装系统，则可能需要修改 BIOS 设置，将“Boot First”设为“NETWORK”或“LAN”，然后重启主机。

新机客户机出现这个界面，回车即可安装。



出现安装摘要，可以看到安装源在 ftp://10.11.21.236/rhel7/中，设置分区后按常规方法安装。


redhat
INSTALLATION SUMMARY
RED HAT ENTERPRISE LINUX 7.0 INSTALLATION
us
LOCALIZATION
DATE & TIME
Americas/New York timezone
KEYBOARD
English (US)
LANGUAGE SUPPORT
English (United States)
SOFTWARE
INSTALLATION SOURCE
ftp://10.11.21.236/rhel7
SOFTWARE SELECTION
Minimal Install
SYSTEM
INSTALLATION DESTINATION
Automatic partitioning selected
NETWORK & HOSTNAME
Wired (ens33) connected

1.2 实现 Kickstart 无人值守安装

上一节学习了通过 PXE 技术远程安装 RHEL7 系统的方法，安装介质不再受限于光盘、移动硬盘等设备，大大提高了系统安装的灵活性。然而安装其间仍需要手动选择语言、键盘类型、指定安装源等一系列交互操作，当需要批量安装时显得非常不便。

本节将进一步学习如何实现无人值守自动安装，通过使用 Kickstart 工具配置安装应答文件，自动完成安装过程中的各种设置，从而无需手动干预、提高网络装机效率。

1.2.1 准备安装应答文件

在 RHEL7 系统中安装 system-config-kickstart 工具以后，即可通过图形法向导工具来配置安装应答文件。如果用户对自动应答文件的配置比较熟悉，也可以直接编辑 RHEL6 安装后自动创建的应答文件（/root/anaconda-ks.cfg），根据需要适当修订后使用。

```
[root@localhost ~]# yum -y install system-config-kickstart
```

1、配置安装应答参数

```
[root@localhost ~]# system-config-kickstart
```

1）基本信息及安装方法

“基本信息”可参考图来指定，例如将默认语言设为“中文（简体）”、时区设为“Asia/Shanghai”，将根口令设为“redhat”，并勾选“安装后重新引导系统”。

在“安装方法”对话框中，应正确指定 RHEL7 安装源，如图所示，若有用户验证信息

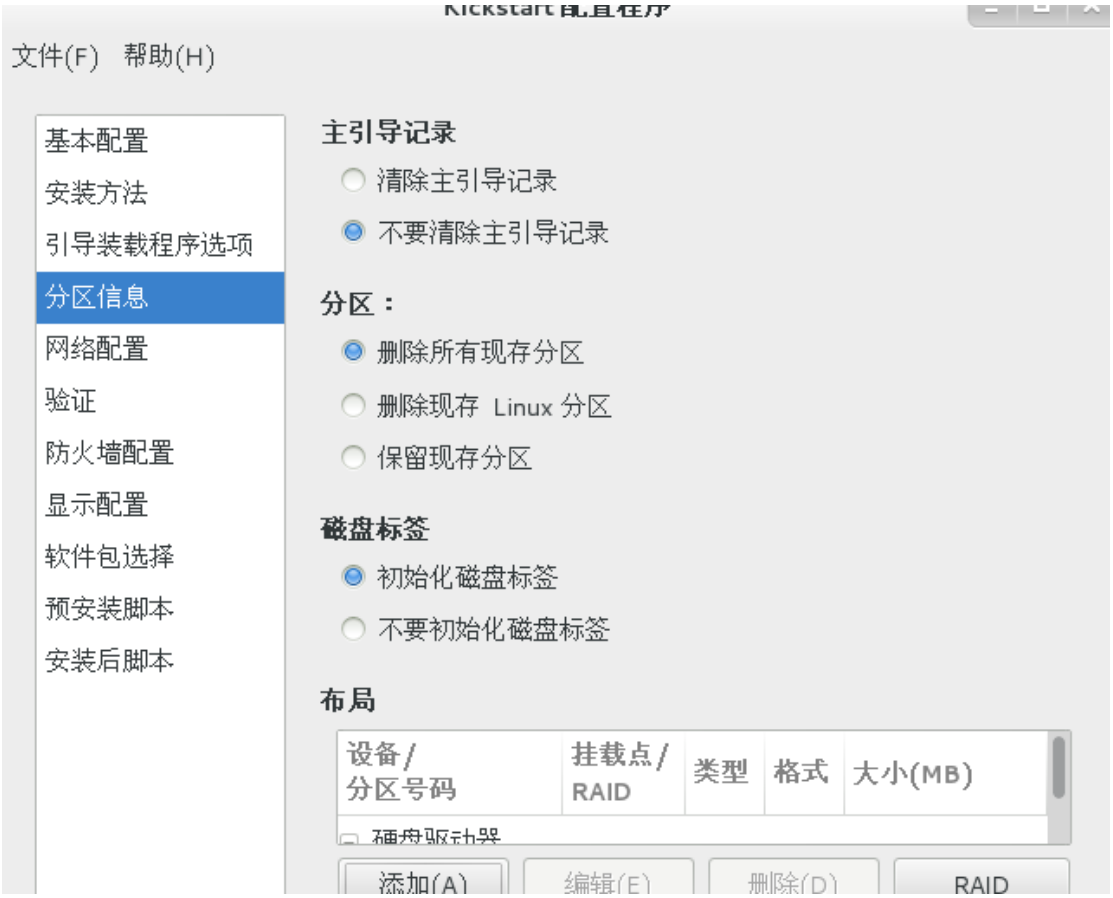
也需一并指定。


Kickstart 配置程序
文件(F) 帮助(H)
基本配置
安装方法
引导装载程序选项
分区信息
网络配置
验证
防火墙配置
显示配置
软件包选择
预安装脚本
安装后脚本
安装方法
执行新安装
升级现有安装
安装方法
光盘驱动器
NFS
FTP
HTTP
硬盘驱动器
FTP 服务器：
ftp://10.11.21.236
FTP 目录：
rhel7
指定 FTP 用户名和密码
FTP 用户名：
FTP 密码：


Kickstart 配置程序
文件(F) 帮助(H)
基本配置
安装方法
引导装载程序选项
分区信息
网络配置
验证
防火墙配置
显示配置
软件包选择
预安装脚本
安装后脚本
安装类型
安装新引导装载程序
不安装引导装载程序
升级现存引导装载程序
GRUB 选项：
使用 GRUB 密码
密码：
确认密码：
给 GRUB 密码加密
安装选项
在主引导记录(MBR)上安装引导装载程序
在引导分区的第一扇区上安装引导装载程序

2) 分区信息

在“分区信息”对话框中，需正确规划分区方案，例如可划分一个 500MB 的/boot 分区 2GB 的交换分区，将剩余空间划分给根分区。



3) 网络配置及防火墙配置

4) 软件包选择

5) 安装脚本

在“预安装脚本”、“安装后脚本”对话框中，可以分别添加在安装前、安装后自动运行的可执行语句。此项设置使用服务器的自动化配置变得更加容易，例如可以使客户机在完成安装后自动设置 YUM 仓库，如图。需要注意的是，应确保编写的代码能够正确执行，以免安装失败。

6) 其他信息

若没有特殊要求，在“引导装载程序”、“验证”、“显示配置”、“预安装脚本”、“安装后脚本”等对话框中，只要保持默认设置就可以了。

2、保存自动应答文件

单击 Kickstart 配置程序的“文件” “保存”菜单，指定目标文件夹、文件名，将配置好的应答参数保存为文本文件，例如/root/ks.cfg。以后若要修改此应答配置，可以在 Kickstart 配置程序中打开进行调整，或直接用 vi 等文本编辑工具修改。参照 /usr/share/doc/pykickstart-1.99.43.10/kickstart-docs.txt 或/root/anaconda-ks.cfg

```
[root@localhost ~]# grep -v ^# /root/ks.cfg

#platform=x86, AMD64, 鎴?Intel EM64T
#version=DEVEL
# Install OS instead of upgrade
install
# Keyboard layouts
keyboard 'us'
# Halt after installation
halt
# Root password
rootpw --iscrypted $1$eXHfTpnx$uQ/nqj1jNdu2BUL46jIr80
# System timezone
timezone Asia/Shanghai
# Use network installation
url --url="ftp://10.11.21.236/rhel7"
# System language
lang zh_CN
# Firewall configuration
```

```
firewall --disabled
# Network information
network  --bootproto=dhcp --device=eth0
# System authorization information
auth  --useshadow  --passalgo=sha512
# Use graphical install
graphical
firstboot --disable
# SELinux configuration
selinux --enforcing

# System bootloader configuration
bootloader --location=mbr
# Partition clearing information
clearpart --all
# Disk partitioning information
part /boot --fstype="xfs" --size=500
part swap --fstype="swap" --size=2000
part / --fstype="xfs" --grow --maxsize=20000 --size=1
%packages
@base
@core
@desktop-debugging
@dial-up
@fonts
@gnome-desktop
@guest-agents
@guest-desktop-agents
@input-methods
@internet-browser
@multimedia
@print-client
@x11

%end

%post --interpreter=/bin/bash
rm -f /etc/yum.repos.d/*
echo '[base]
> name=rhel7
> baseurl=ftp://10.11.21.236/rhel7
> enabled=1
> gpgcheck=1
>         gpgkey=         /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release'          >
```

```
/etc/yum.repos.d/base.repo
%end
```

软件包安装参考/root/anaconda-ks.cfg。以下命令对编好 ks.cfg 文件进行语法检查

```
[root@localhost ~]# ksvalidator /root/ks.cfg
```

1.2.2 实现批量自动装机

有了自动安装的应答文件以后，只要放置到了 PXE 安装服务器的 FTP 目录下，并适当修改引导菜单，就可以实现基于网络的批量自动装机了。

1、启用自动应答文件

在 PXE 远程安装服务器中，将上一节建立的应答文件复制到/var/ftp/rhel6 目录下，使客户机能够通过 ftp://10.11.21.236/rhel7/ks.cfg 访问,然后编辑引导菜单文件 default，添加 ks 引导参数以指定 ks.cfg 应答文件的 URL 路径。

```
[root@localhost ~]# cp /root/ks.cfg /var/ftp/rhel7/ks.cfg

[root@localhost ~]# vi /var/lib/tftpboot/pxelinux.cfg/default
default auto
prompt 1
label auto
    kernel vmlinuz
   append          ks=ftp://10.11.21.236/rhel7/ks.cfg          initrd=initrd.img
inst.repo=ftp://10.11.21.236/rhel7
```

2、验证无人值守安装

启用自动应答安装以后，每次当客户机以 PXE 方式引导时，将自动下载 ks.cfg 应答配置文件，然后根据其中的设置安装 RHEL7 系统，而无需手工干预，如图所示。这样就可以同时为多台客户机安装系统了。

客户机安装完成以后，检查其 YUM 仓库配置，可以发现已经按照“安装后脚本”的设置自动建立了 etc/yum.repos.d/rhe17.repo。

第二章　正则表则式

1.1 正则表达式基础

正则表达式(regular expression)描述了一种字符串匹配的模式，可以用来检查一个串是否含有某种子串、将匹配的子串做替换或者从某个串中取出符合某个条件的子串等。

例如 grep, expr, sed , awk. 或 Vi 中经常会使用到正则表达式,为了充分发挥 shell 编程的威力,需要精通正则表达式。(以下有些需要用 egrep)

1.1.1　书写正则表达式

. 匹配任意单个字符(除换行符)

* 匹配重复零次或多次前一字符

.* 匹配 0 个或任意个字符

+ 匹配一个或多个前面的字符.它的作用和*很相似,区别是它不匹配零个字符的情况

? 匹配零或一个前面的字符。它一般用于匹配单个字符

^ 匹配一行的开头

$ 匹配一行的行尾

[...] 匹配集合中任意字符 如"[xyz]" 匹配字符 x, y, 或 z

[^...] 匹配不属集合 中 任意字符

\<, \> 用于表示单词的边界。\< 匹配词首，\>词尾，如"\<the\>" 匹配单词"the"

\{ \} 指示前面正则表达式匹配的次数.

例：

.* 匹配 0 个或任意个字符

a* 匹配 空,a,aa,aaa .....

a+ 匹配 aa,aaa .....

? 匹配零或一个前面的字符。它一般用于匹配单个字符

^a 匹配以 a 开头的行

a$ 匹配以 a 结尾的行

[abc] 匹配当前位置是 a 或 b 或 c

[^abc] 匹配 abc 除外的字符

\<li 匹配以 li 开头的单词

sh\> 匹配以 sh 结尾的单词

\<the\> 匹配单词"the"

a\{n\} 表示重复 n 次 a

1.2 grep 匹配文本

grep 命令格式：

grep ［选项］ 查找条件 目标文件

常用选项如下：

选项	说 明：
-i	不区分大小写搜索
-n	返回包含行号
-v	返回不包含模式的行
-Ax	匹配关键字的后 x 行 如：# grep -A 5 ftp /etc/passwd
-Bx	匹配关键字的前 x 行
-r	递归式搜索，从当前目录开始
-c	匹配行的统计
--color	--color=auto 如：# date --help |grep --color year
-e	多个-e 选项表示用 or 连接多个表达式

应用例子：

1）搜索有 passwd 文件中有 liweijie 的行,并输出行号

```
$grep -n ‘liwiejie’ /etc/passwd
```

2)搜索有 passwd 文件中有 li 开头的行

```
$ grep ‘^li’ /etc/passwd
```

3）搜索 file1 文件中以小写字母开头的行

```
$ grep -n ’^[a-z]’ file1
```

4）搜索 file1 文件中不以#号开头的行，并去掉空行

```
$ grep -v ’^# ’ file1 | grep -v ’^$’
```

5）搜索 file1 文件不以#号和；开头的行：

```
$ grep -v ‘^[#;]’ file1
```

6)搜索 file1 文件中以 g 开头和结尾,中间是至少一个 o 的字符串,即 gog, goog....gooog...等

```
$ grep -n ’goo*g’ file1
```

7) 搜索 file1 文件中包 cat 或 dog 的行

```
$ grep -e ‘cat’ -e ‘dog’ file1
```

8) 搜索单词表字母超过 40 个的单词

```
$ grep ’.\{40,\}’ /usr/share/dict/words
```

1.3 vi 中使用正则表达式

一 VI 中如何使用正则表达式

使用正则表达式的命令最常见的就是 **/ （搜索）**命令。其格式如下：

/正则表达式

另一个很有用的命令就是 **:s（替换）**命令，将第一个//之间的正则表达式替换成第二个//之间的字符串。

:s/正则表达式/替换字符串/选项

二、元字符

元字符是具有特殊意义的字符。使用元字符可以表达**任意字符、行首、行 尾、某几个字符**等意义。

元字符一览

元字符	说明

元字符	说明
\d	匹配阿拉伯数字，等同于[0-9]。
\D	匹配阿拉伯数字之外的任意字符，等同于[^0-9]。
\x	匹配十六进制数字，等同于[0-9A-Fa-f]。
\X	匹配十六进制数字之外的任意字符，等同于[^0-9A-Fa-f]。
\w	匹配单词字母，等同于[0-9A-Za-z_]。
\W	匹配单词字母之外的任意字符，等同于[^0-9A-Za-z_]。
\t	匹配<TAB>字符。
\s	匹配空白字符，等同于[\t]，即空格和 Tab
\S	匹配非空白字符，等同于[^ \t]。

例子：

/char\s\+[A-Za-z_]\w*; ″ 查找所有以 char 开头，之后是一个以上的空白，″ 最后是一个标识符和分号

/\d\d:\d\d:\d\d ″ 查找如 17:37:01 格式的时间字符串

:g/^\s*$/d ″ 删除只有空白的行

:s/\<four\>/4/g ″ 将所有的 four 替换成 4，但是 fourteen 中的 four 不替换

三、替换变量

在正规表达式中使用 **\(** 和 **\)** 符号括起正规表达式，即可在后面使用**\1、\2** 等变量来访问 **\(** 和 **\)** 中的内容。

使用例

/\(a\+\)[^a]\+\1 //查找开头和结尾处 a 的个数相同的字符串，″ 如 aabbbaa，aaacccaaa，但是不匹配 abbbaa

:s/\(http:\/\/[-a-z\._~\+%\/]\+\)/<a href="\1">\1<\/a>/ //将 URL 替换为http://url的格式

:s/\(\w\+\)\s\+\(\w\+\)/\2\t\1
//将 data1 data2 修改为 data2 data1

四、函数式

在替换命令 **s///** 中可以使用函数表达式来书写替换内容，格式为

:s/替换字符串/\=函数式

在函数式中可以使用 submatch(1)、submatch(2) 等来引用 **\1、\2** 等的内容，而 submatch(0)可以引用匹配的整个内容。

使用例

:%s/\<id\>/\=line(".") ″ 将各行的 id 字符串替换为行号

:%s/^\<\w\+\>/\=(line(".")-10) ."." submatch(1) " 将每行开头的单词替换为 (行号-10).单词 的格式， 如第 11 行的 word 替换成 1. word

五、常用 vi 正则表达式

:%s/^/Head content/g "全局插入开头信息

:%s/$/Tail content/g "全局插入尾部信息

:%s= *$== "将所有行尾多余的空格删除

:g/^\s*$/d "将所有不包含字符(空格也不包含)的空行删除.

:%s/ */ /g 把一个或者多个空格替换为一个空格。

:%s/ *$// 去掉行尾的所有空格。

:%s/^/ / 在每一行头上加入一个空格。

:%s/^[0-9][0-9]* // 去掉行首的所有数字字符。

:%s/b[aeio]g/bug/g 将所有的 bag、beg、big 和 bog 改为 bug。

:%s/t([aou])g/h1t/g 将所有 tag、tog 和 tug 分别改为 hat、hot 和 hug（注意用 group 的用法和使用 1 引用前面被匹配的字符）。

第三章 使用 vim 编辑文本文件

3.1 Linux 的 VI 编辑器

1、 关于文本编辑器

文本编辑器有很多，比如图形模式的 gedit、kwrite、OpenOffice ... ... ，文本模式下的编辑器有 vi、vim（vi 的增强版本）和 nano ... ... vi 和 vim 是我们在 Linux 中最常用的编辑器。我们有必要介绍一下 vi（vim）最简单的用法，以让 Linux 入门级用户在最短的时间内学会使用它。

2、为什么要学会应用 vi

vi 或 vim 是 Linux 最基本的文本编辑工具，vi 或 vim 虽然没有图形界面编辑器那样点鼠标的简单操作，但 vi 编辑器在系统管理、服务器管理中，永远不是图形界面的编辑器能比的。当您没有安装 X-windows 桌面环境或桌面环境崩溃时，我们仍需要字符模式下的编辑器 vi；vi 或 vim 编辑器在创建和编辑简单文档最高效的工具；

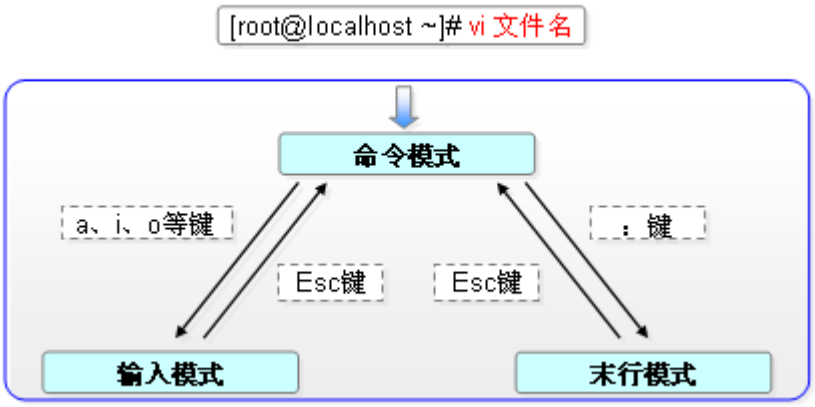
3.2 vi 编辑器的使用方法

3.2.1 vi 编辑器的工作模式

Vi 有三种基本的工作模式：命令模式；文本输入模式；末行模式。
vi 将命令模式和文本输入模式区分开来，这经常被认为是 vi 的一个大问题，但往往这也被认为是 vi 的优势所在。
理解其中的区别是掌握 vi 的关键，vi 启动时，开始处于命令模式。

- 命令模式：我们可以在文件中到处移动，改变文本的某个特定区域、剪切、复制和粘贴文本，还有更多。
- 输入模式：是指用户可以真正输入文本。
- 末行模式：设置 vi 编辑环境、保存文件、退出以及对文件内容进行查找、替换操作。
- 可视模式：便于选取文本

3.2.2 vi 编辑器模式的切换



1、从命令模式与文本输入模式切换：

i：光标在当前位置进入文本输入模式
I：光标跳到行首并进入文本输入模式
a：光标后退一格并进入文本输入模式
A：光标退到行尾并进入文本输入模式
o：在光标所在行下新起一行并进入文本输入模式
O：在光标所在行上新起一行并进入文本输入模式
s：删除光标所在字符并进入文本输入模式
S：删除光标所在行并进入文本输入模式
按 ESC 键可以返回命令模式。

2、命令行模式与末行模式的切换

在命令模式中，用户按住“：”键可以进入末行模式，等待用户输入命令。多数文件管理命令都是在此模式中之行的。命令执行完毕后，Vi 自动回到命令模式，按 ESC 键可以返回命令模式。

2、命令行模式与可视模式的切换

在命令模式中，用户按住“v”键可以进入可视模式

3.2.3 vi 编辑器命令行模式的基本操作

1、光标移动

- 当我们命令模式后，我们可以用键盘上方向键（h、j、k、l）来移动光标。
- 使用 Page Down 或 Ctrl+F 向下翻页。
- 使用 Page Up 或 Ctrl+B 向上翻页。
- 使用 Home 或“^”、数字 0 可以将光标移到本行的行首。
- 使用 End 或“$”可以将光标移到本行的行尾。
- 使用 1G 或 gg 可以跳转到文件内容的第一行。
- 使用 G 可以跳转到文件内容的最后一行。
- 使用#G 可以跳转到文件内容的第#行(其中#用具体的数字替换)。

2、文本内容的复制、粘贴和删除操作

Del 或 x	删除一个字符；
#x	删除几个字符，#表示数字，比如 3x；
dd	删除一行；
#dd	删除多个行，#代表数字，比如 3dd 表示删除光标行及光标的下两行；
dw	删除一个单词；
#dw	删除几个单词，#用数字表示，比如 3dw 表示删除三个单词；
d$	删除光标到行尾的内容；
yy	复制光标所在行
#yy	复制从光标开始的#行
yw	选定光标所在词复制
#yw	复制光标所在位置到之后 #个单词
y$	复制光标所在位置到行尾的部分
p	贴在光标所在位置之右
P	贴在光标所在位置之左

3、恢复修改及恢复删除操作

u	撤消最近一次的操作，可按多次；
U	撤销当前行所有操作。

4、文件内容查找

在命令模式中按下“/”后可输入要查找的字符串，从光标所在的位置向后查找，按 n 可以移到下一个查找结果。:set ic 可以忽略大小写，set noic 检查大小写。

3.2.4 vi 编辑器可视模式的基本操作

为了便于选取文本，VIM 引入了可视(Visual)模式。要选取一段文本，首先将光标移到段首，在普通模式下按 v 进入可视模式，然后把光标移到段末。需要注意，光标所在字符

是包含在选区中的。这时可以对所选的文本进行一些操作，常用的(可视模式)命令有：

x 或 d　　剪切(即删除，同时所选的文本进入剪贴板)

y　　复制

当输入了命令以后，VIM 将回到普通模式，这时可以按 p 或 P 进行粘贴。

3.2.4 vi 编辑器末行模式的基本操作

Vi 编好文件后，可以在末行模式下按自己的需求键入末行模式的参数。

先键入“ ： ”进入末行模式，

：q　　使用 q 指令退出 vi 编辑器。

：q!　　不保存退出 vi 编辑器。

：wq　　保存退出。

：w　　保存文件。

：w 文件名　　另存为。

：e 文件名　　打开新文件。

：r 文件名　　在当前文件中读入其他文件的内容。

：set nu　　显示行号

：set nonu　　取消显示行号

：sub /i/I　　将当前行中的第 1 个字母　　“i”替换为“I”。

：10,20s/default/DEFAULT/g　将 10~20 行中 default 全部替换为 DEFAULT.

：%s/default/DEFAULT/g　　将文档中 default 全部替换为 DEFAULT.

3.2.5 vi 编辑器高级操作

1、Vim 可以再多分隔窗口环境下编辑多个文件。有两种方法：

在启动 vi 时候使用-o 或者-O 选项，并加上需要同时编辑的文件名。

#vi -o file1.txt file2.txt 水平分割窗口编辑 file1.txt 和 file2.txt

#vi -O file1.txt file2.txt 垂直分割窗口编辑 file1.txt 和 file2.txt

在不同的窗口间移动，使用 ctrl+w 命令。

2、在 vim 中执行外部命令

输入 :! 然后紧随著输入一个外部命令可以执行该外部命令。

我们以 ls 命令为例。输入 !ls〈回车〉。该命令就会列举出您当前目录的内容。

例：把系统的日期时间导入光标所在位置

：r !date

3、在 vim 中定义快捷键

格式： map 快捷键　触发命令

例：定义 ctrl+p 在一行最前面加上#号，不管光标在行中什么位置。

：map ^p I#<ESC>　　----^p 为 ctrl+v+p 或 ctrl+v ctrl+p

例：定义 ctrl+x 在一中删除最前面的#号，不管光标在行中什么位置。

：map ^x 0x　　----^b 为 ctrl+v+x

例：定义 ctrl+e 在光标位置插入我的邮箱 liweijie@163.com。

：map ^e iliweijie@163.com　　----^e 为 ctrl+v+e

4、在 vim 中连续行注释

格式：　n1,n2s/^/#/g　　--^代表行首

例：在 10-20 行的行首加上#号注释掉

：10,20s/^/#/g

例：在 10-20 行的行首去掉#号

：10,20s/^#//g

5、替换

格式：　ab

例：输入 mymail 替换 liweijie@sina.com

:ab mymail　liweijie@sina.com

:unab mymail

5、保存 VIM 的一些设置，如自动设置行号、快捷键等

vim ~/.vimrc

set nu

ab mymail　liweijie@sina.com

第四章　计划任务

在很多时候为了自动化管理系统，我们都会用到计划任务，比如关机，管理，备份之类的操作，我们都可以使用计划任务来完成，这样可以是管理员的工作量大大降低，而且可靠性更好。linux 系统支持一些能够自动执行任务的服务，我们称为计划任务。LINUX 有二种计划任务：

- at：指定一个时间执行一个任务 （适用一个或多个任务，执行一次后就不用）。
- cron：根据一个时间表自动执行任务 （使用一个或多个任务，周期性执行）。

4.1　一次任务管理 at

使用 at 命令设置计划任务只在指定的时间执行一次，有一个前提条件，需要一个 atd 的系统后台进程。

1、具体使用方法：

#at [-q 队列] [-f 文档名] 时间

> 输入要执行的命令

> ctrl+d 结束输入

选项：

-q queue 使用指定的队列。队列名称是由单个字母组成，合法的队列名能够由 a-z 或 A-Z。a 队列是 linux at 命令的默认队列。

-f file 使用该选项将使命令从指定的 file 读取，而不是从标准输入读取。

时间：

16:10 2014-10-9

now +5min （now 现在）

teatime tomorrow （teatime is 16:00 下午茶时间）

noon +4 days (noon 中午）

5pm august 3 2016

例：使用 date 命令确认当前的系统时间，并设置 2014-9-30 22:40 自动执行以下任务：统计系统中的用户数，并该数量保存到/tmp/user.count 文件中。

```
[root@zx ~]# date
2012年 01月 30日 星期一 22:36:31 CST
[root@zx ~]# at 22:40 2012-01-30
at> wc -l /etc/passwd > /tmp/user.count
at> <EOT>
job 5 at 2012-01-30 22:40
[root@zx ~]#
```

<EOT>是 ctrl+d 中

例：设置在当天 17:30 发送一个广播“系统将在 15 分钟后重启!”并在 15 分钟后关闭系统。

```
[root@zx ~]# at 17:30
at> wall 系统将在15分钟后重启！
at> shutdown -r +15
at> <EOT>
job 6 at 2012-01-31 17:30
[root@zx ~]#
```

其他例子 ：

1) 三天后的下午 5 点执行 touch /home/a.txt

 at 5pm +3 days touch /home/a.txt

2) 三个星期后的下午 5 点钟执行 touch /home/a.txt

 at 5pm +3 weeks touch /home/a.txt

3) 明天的 17:20 执行 touch /home/a.txt

 at 17:20 tomorrow touch /home/a.txt

2、 atq 查询当前的等待任务

用 atq 来查询，已经运行的任务，就消失了。这就是 at 计划任务的重点，只运行一次。

```
[root@zx ~]# atq
6       2012-01-31 17:30 a root
[root@zx ~]#
```

3、atrm：删除等待任务

启动计划任务后，如果不想启动设定好的计划任务可以使用 atrm 命令删除。

格式：atrm 任务号

命令后面跟计划任务编号，如果不跟，就会删除这个用户所有的计划任务。

例：删除计划任务 6，并用 atq 查看。

```
[root@zx ~]# atrm 6
[root@zx ~]# atq
[root@zx ~]#
```

还可以进入到/var/spool/at 目录里把计划任务删除，计划任务的文件都保存在该目录里，可以用 rm -f 文件名来删除(以文件的形式删除计划任务，因为计划任务是以文件形式保存在该目录中) 。在通常情况下，超级用户都可以使用这个命令。对于其他用户来说，能否可以使用就取决于两个文件:/etc/at.allow 和/etc/at.deny。at 命令是可以基于用户来控制的，我们可以明确指定哪些用户可以使用 at 计划任务，哪些用户不可以使用 at 计划任务。

4.2 crontab 服务

相对与 at，crontab 的优点就是能够周期性的执行某个命令,at 却只能执行一次。crontab 的后台进程名字是 crond ,cron 也是系统服务。我们先简单介绍下 cron 的使用命令，然后来在结合实例讲解

制定个人计划任务

#crontab -e　　编辑当前用户的 cron 表

#crontab -l　　查看当前用户的 cron 表

#crontab -r　　删除当前用户的 cron 进程

#crontab -u 用户名 以某用户的身份来控制 cron 表

还有个重要的知识点，就是当用户的计划任务建立后是存放在 var/spool/cron 这个目录。

1、编辑用户的计划任务列表

当使用 crontab -e 编辑当前用户的 cron 表后，会出现一个类似 vi 界面，cron 的格式是这样的。分成两列，左边是时间，右边是运行的命令。时间是由 5 个部分组成。

例：* * * * *　　wall hello everyone

5 个星号分别代表:minute hour day-of-month month-of-year day-of-week ，而 wall hello everyone 这是命令内容。上面的意识是每分每小时每天每月每周广播 hello everyone。

分钟 小时 日 月 周 [用户名] 命令

这里的 5 个星号就代表的时间和日期：

第一个*星号代表个小时的第几分钟：minute 范围是从 0-59

第二个*星号代表每天的第几个小时：hour 范围是从 0-23

第三个*星号代表每月的第几个日：day-of-month 范围从 1-31

第四个*星号代表没年的第几个月：month-of-year 范围从 1-12

第五个*星号代表每周的星期几：day-of-week 范围从 0-6，其中 0 表示星期日

用户名：也就是执行程序要通过哪个用户来执行，这个一般可以省略；

命令：执行的命令和参数。

除了“*”以外，还可以用“-”、“,”、“/”与数字构成表达式来表示复杂的时间关系。

- -：可以表示一个连续的时间范围，如 1-4。
- ,：可以表示间隔不连续的时间范围，如 3，5，7。
- /：可以指定间隔的频率，如分钟字段中的 */4 ，表示每隔 4 分钟。

例如：

请写出下列任务时间是什么？		请写出下列要求的时间	
0 23 * * *	每天的 23 点整	周一到周五每天 17:00	0 17 * * 1-5
30 12,0 * * *	每天的 12：30 和 00:30	每周一、三、五的 8 点 30 分	30 8 * * 1,3,5
0 22 1 1-12 *	1-12 月每月 1 号 22 点整	8 点到 18 点之间每隔 2 小时	* 8-18/2 * * *
*/5 * * * *	每 5 分钟	每隔 3 天	* * */3 * *

下面举例来学习用户计划任务的编辑。

例：由 root 用户设置一份 crontab 计划任务列表，完成以下任务

- 每天早上 7:50 自动开启 sshd 服务，22:50 关闭 sshd 服务。
- 每隔 5 天清空一次 FTP 服务器的公共目录/var/ftp/pub 中的数据。
- 每周六的 7:30 重启 httpd 服务。
- 每周一、三、五的下午 17:30，使用 tar 命令自动备份 /etc/httpd 目录。

```
# crontab  -e
   50   7   *   *   *      /usr/bin/systemctl start  sshd
   50   22  *   *   *      /usr/bin/systemctl stop  sshd
   0    *   */5 *   *     /usr/bin/rm  -rf  /var/ftp/pub/*
   30   7   *   *   6     /usr/bin/systemctl restart   httpd
   30   17  *   *   1,3,5   /usr/bin/tar  jcvf  httpd.tar.bz2  /etc/httpd
```

例：为用户 zxuser 设置计任务，每周日周晚上 23:55 将/etc/passwd 文件内容复制到宿主目录中，保存为 pwd.txt。

```
  # crontab  -e  -U  zxuser
55  23  *   *   7      /usr/bin/cp  /etc/passwd  /home/zxuser/pwd.txt
```

2、查看用户的计划任务列表

例：查看 root 用户的计划任务列表的内容。

```
# crontab  -l
```

例：查看用户 zxuser 的计划任列表的内容。

```
# crontab  -l  -U  zxuser
```

3、删除用户的计划任务列表

例：删除 root 用户的计划任务列表的内容。

```
# crontab  -r
```

4、如何使要限制某个用户，不能用 cron 计划任务，如何办到？

```
vim /etc/cron.deny
```

写入用户即可。

例：配置 cron access。liweijie 不能使用 cron。这个限制不能影响其他任何用户

```
vim /etc/cron.deny
```

```
liweijie            #添加该用户
 su - liweijie
  crontab -e              #验证，是否可以做任务计划
```

在设置用户的 crontab 计划任务的过程中，由于每一条记录只能运行一行命令，难以完成更复杂的系统管理任务，因此在实际工作中，通常是先编写脚本文件，再用计划任务配置中加载脚本文件执行。

4.3 管理临时文件

现代的系统需要大量临时目录和文件，不仅仅存在/tmp 目录高级用户和普通用户都可使用和滥用，还有/run 给一些特殊任务如守护进程使用。/run 下的文件存在内存里，所以系统重启所有信息丢失。

为了保持系统运行的干静，当守护进程或脚本需要文件或目录不存在时需要建立，旧的文件需要净化，防止填满磁盘产生故障。

在这一部分中，系统管理员依靠一个 systemV init-scripts 的 rpm 包建立目录，用 tmpwatch 工具删除旧的没用的目录。

在 rhel7 中 systemd 提供了 systemd-tmpfiles 管理临时文件。

当 systemd 启动，首先加载一个 systemd-tmpfiles-setup 服务单元。这个服务运行 systemd-tmpfiles --create --remove 命令读 /etc/tmpfiles.d/*.conf 、/usr/lib/tmpfiles/*.conf、/run/tmpfiles/*.conf 文件建立或删除临时文件目录。

定期清理：为了确保长时间运行的系统磁盘不被存旧数据填满，systemd 定时器单元能调用 systemd-tmpfiles --clean 进行清理。Systemd-tmpfiles-clean.service 在 systemd 启动后 15 分钟启动，并每 24 小时运行一次。

清理旧文件比较三个时间戳：atime：最后一次访问时间，mtime：最后一次修改的时间，ctime：最后一次修改文件状态的时间。三个时间都老于设定时间才会清理。

用 ls -l 看到的 mtime，可用 stat 命令看三个时间戳。

Systemd-tmpfiles 配置文件可参考 man 5 tmpfiles.d

/etc/tmpfiles.d/*.conf 管理员配置自定义的临时位置配置

/usr/lib/tmpfiles.d/*.conf 提供给 RPM 包相关临时文件配置，不能修改

/run/tmpfiles.d/*.conf 守护进程产生的临时文件配置

如果三个目录中有相同的配置文件，优先级为 /etc > /run > /usr

实验一：

现在磁盘剩余空间较少， /tmp 目录占用空间比较大，默认配置为清理 10 天没用的文件，现在要求改为把 5 天没有使用文件清理掉。

1、复制/usr/lib/tmpfiles.d/tmp.conf 到 /etc/tmpfiles.d/ 目录下。/etc 目录下配置优先级高。

```
# cp /usr/lib/tmpfiles.d/tmp.conf  /etc/tmpfiles.d/
```

2、修改/etc/tmpfiles.d/tmp.conf ，把 10 天改成 5 天。

3、Systmed-tmpfiles --clean 接受新的配置文件。

```
# systemd-tmpfiles --clean   tmp.conf
```

实验二：

自定义一临时目录/run/mytmp，配置只有 root 有读写的权限，30 秒没使用的文件会被清理。

1、/etc/tmpfiles.d/ 创建一个 mytmp.conf 配置文件，设置如下。

```
# vim /etc/tmpfiles.d/mytmp.conf
  d  /run/mytmp  0700 root  root  30s
```

2、Systmed-tmpfiles --create 创建目录。

```
# systemd-tmpfiles --create   mytmp.conf
# ls -ld /run/mytmp
```

3、在/run/mytmp 中建一个测试文件，并待 30 秒。

```
#  touch  /run/mytmp/a.txt
#  sleep 30s
```

4、 运行 systemd-tmpfiles --clean，查看 a.txt 是否被删除。

```
# systemd-tmpfiles --clean  mytmp.conf
#  ls -l  /run/mytmp
```

第五章　管理进程优先级

5.1 进程优先级和 nice 概念

1、 linux 进程调度和多任务

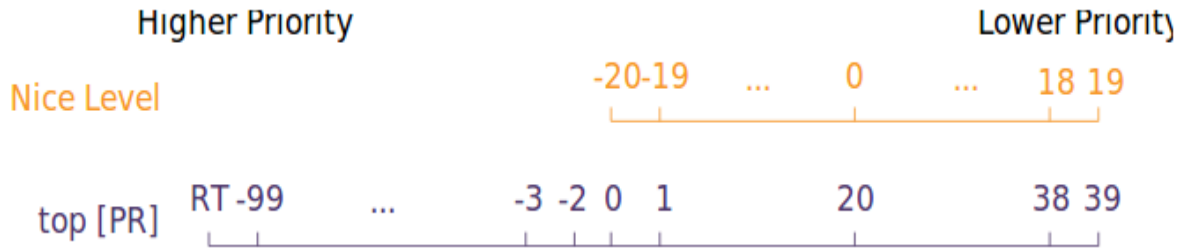
现代计算机从一次只能处理一条指令的低端处理器到有上百多个核处理器的超级计算机，都有运行进程数超过核心数的趋势。

操作系统实现多任务，也就是运行的进程和线程数超可用处理单元数，用的一种技术叫时间分片。在一个核心切换多个进程通过进程调度来实现。

2 、 进程优先级

因为不是每一个进程都同等重要，不同进程可以采用不同策略，通过 nice 值可以调整不同进程的优先级，共有 40 个不同级别值。

nice 值为-20----19，默认情况下，进程从父进程那里继承 nice 级别，一般为 0,高 nice 级别表示低优先级，低 nice 值表示高优先级。



5.2 用 nice 和 renice 影响进程优先级

1、 查看进程的 nice 级别

1） gnome-system-monitor

2） top 命令

Top 能交互式查看和管理进程，默认情况下有两列和 nice 值相关，NI 和 PR。NI 是实际 nice 值，PR 是 nice 映射到优先级队列。Nice 值 -20 映射优先级为 0。

按 r 可以交互式调整 nice 值。

3） ps 命令

ps axo pid,comm,nice --sort=nice

2、 用不同 nice 值加载进程

```
[root@localhost ~]# nice -n  -10  命令
```

注意：非 root 用户只能设置正数优先级 0-19

3、 改变正在运行的进程 nice 值

```
[root@localhost ~]# renice -n  -10  进程 ID
```

实验：进程优先级

1、查看 CPU 内核数

```
[root@server10 ~]# grep -c '^processor' /proc/cpuinfo
1
```

2、后台启动超过 CPU 内核数 sha1sum 进程。

```
[root@server10 ~]# sha1sum /dev/zero &
[1] 3588
[root@server10 ~]# sha1sum /dev/zero &
[2] 3589
```

3、查看进程占用 CPU 资源比较。

```
[root@server10 ~]# ps -o pid,pcpu,time,nice,comm
  PID %CPU     TIME  NI COMMAND
 2565  0.0 00:00:00   0 bash
 3588 50.2 00:00:36   0 sha1sum
 3589 48.9 00:00:34   0 sha1sum
```

4、终止上面启动 sha1sum 进程

```
[root@server10 ~]# killall sha1sum
```

5、后台按上面方法再次启次 sha1sum 进程，用 nice 启动另一个 sha1sum，并设置 nice 值为-10.

```
[root@server10 ~]# sha1sum /dev/zero &
[1] 3635
[root@server10 ~]# nice -n -10 sha1sum /dev/zero &
[2] 3647
```

6、再次查看进程占用 CPU 资源比较。

```
[root@server10 ~]# ps -o pid,pcpu,time,nice,comm
  PID %CPU     TIME  NI COMMAND
 2565  0.0 00:00:00   0 bash
 3635 14.0 00:00:43   0 sha1sum
```

```
 3647 89.2 00:01:23 -10 sha1sum
```

7、把用 nice 启动的进程 nice 重设为 10，再次查看进程占用 CPU 资源比较。

```
[root@server10 ~]# renice -n 10 3647
3647 (进程 ID) 旧优先级为 -10，新优先级为 10

[root@server10 ~]# ps -o pid,pcpu,time,nice,comm
  PID %CPU     TIME  NI COMMAND
 2565  0.0 00:00:00   0 bash
 3635 47.9 00:04:32   0 sha1sum
 3647 53.9 00:04:47  10 sha1sum

[root@server10 ~]# top
   PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 3635 root      20   0  116096   1048    784 R 91.0  0.1   6:49.82 sha1sum
 3647 root      30  10  116096   1052    784 R 12.1  0.1   5:01.97 sha1sum
 3803 root      20   0  123656   1532   1072 R  6.1  0.2   0:00.01 top
按 r 可以在 top 调整优先级。
```

8、终止进程。

```
[root@server10 ~]# killall sha1sum
```

第六章　文件访问控制 ACLs

6.1 POSIX 访问控制列表(ACLs)

1、 访问控制列表的概念

标准 Linux 文件权限，大多数情况是令人满意的，但有局限性。限制访问一个文件的权限是有限的文件的所有者，一个组的成员，或者其他人。它可能不适合的进程（运行程序）是文件的所有组成员，甚至为每个授予权限也令不满意。

ACL 允许细化文件的权限分配。除了文件所有者，所属组和其他用户三种标准权限以外，还可为命名的用户或组分配权限。相同的权限标志应用：R -读，W-写和 X -执行。

文件的所有者可以设置 ACL 在单个文件或目录。如果设置 ACLS，新的文件和子目录，可以自动从父目录的默认 ACL 继承 ACL 设置。类似于正常的文件访问规则，父目录的层次结构将至少需要执行权限设置，用户和组才能访问。

文件系统改必须支持 ACL，XFS 和 EXT4 文件系统内置支持 ACL。

2 、 查看解释 ACL 权限

1） 用 ls -l 命令显示最小 ACL 设置。

The **ls -l** command only outputs minimal ACL setting details:

```
[student@serverX steamies]$ ls -l roster.txt
-rwxrw----+ 1 student controller 130 Mar 19 23:56 roster.txt
```

The "**+**" at the end of the 10-character permission string indicates that there are ACL settings associated with this file. Interpret the *user*, *group*, and *other* "**rwx**" flags as:

- **user:** Shows the *user* ACL settings, which are the same as the standard *user* file settings; **rwx**.
- **group:** Shows the current ACL *mask* settings, not the *group-owner* settings; **rw**.
- **other:** Shows the *other* ACL settings, which are the same as the standard *other* file settings; no access.

2） 用 getfacl 显示文件 ACL 配置

To display ACL settings on a file, use **getfacl *file***:

```
[student@serverX steamies]$ getfacl roster.txt
# file: roster.txt
# owner: student
# group: controller
user::rwx
user:james:---
user:1005:rwx       #effective:rw-
group::rwx          #effective:rw-
group:sodor:r--
group:2210:rwx      #effective:rw-
mask::rw-
other::---
```

To display ACL settings on a file, use **getfacl *file***:

```
[student@serverX steamies]$ getfacl roster.txt
# file: roster.txt
# owner: student          所有者
# group: controller       所属组
user::rwx   所有者权限
user:james:---      james用户权限
user:1005:rwx       #effective:rw-     1005号ID用户的权限rwx，有效权限为rw.
group::rwx          #effective:rw-     属组的权限rwx，有效权限为rw.
group:sodor:r--                       sodor组户的权限r-
group:2210:rwx      #effective:rw-    组ID为2210的组的权限rwx，有效权限为
mask::rw-
other::---
```

- Mask 设置为所有命名用户，所属组和命名组的最大权限，不限制所有者。
- Other ： 上面没有列出的用户和组的权限，---表示没有任何权限。

3） 查看目录 ACLs

To display ACL settings on a directory, use **getfacl /directory**:

```
[student@serverX steamies]$ getfacl .
# file: .
# owner: student
# group: controller
# flags: -s-
user::rwx
user:james:---
user:1005:rwx
group::rwx
group:sodor:r-x
group:2210:rwx
mask::rwx
other::---
default:user::rwx
default:user:james:---
default:group::rwx
default:group:sodor:r-x
default:mask::rwx
default:other::---
```

To display ACL settings on a directory, use **getfacl /directory**:

```
[student@serverX steamies]$ getfacl .
# file: .
# owner: student
# group: controller
# flags: -s-      设置有setgid权限
user::rwx
user:james:---
user:1005:rwx
group::rwx
group:sodor:r-x
group:2210:rwx
mask::rwx
other::---
default:user::rwx      新建文件或子目录所有者的默认权限
default:user:james:---      新建文件或子目录jamas用户的默认权限
default:group::rwx
default:group:sodor:r-x
default:mask::rwx      新建文件或子目录，命名用户、属组和命名给的默认权限。
default:other::---
```

注意：

- Getfacl -R 可以显示目录和目录包含的子目录和文件的 ACL 权限。
- Getfacl 结果保存或输出另一个文件中，setfacl --set-file=file。

5） ACL mask

- Mask 设置为所有命名用户，所属组和命名组的最大权限，不限制所有者。
- Mask 可以用 getfacl 查看，用 setfacl 明确设置，如果没有明确设置能自动添加和设置，默认情况下每当有效 ACLs 添加修改或删除时重新计算。

6） ACL 权限优先级

当确定一个进程是否能访问一个文件，文件权限 ACL 应用如下。

- 如果进程以文件所有者身份运行，那么应用文件所有者的权限。
- 如果进程以命名用户的身份运行，那么应用命名用户的权限（只要 mask 权限许可）
- 如果进程以组的身份运行，相应应用组的权限（只要 mask 权限许可）
- 否则，应用 other 权限。

6.2 设置文件的 ACLs

1、 修改 ACLs

用 setfacl 添加、修改和删除文件和目录的 ACLs。

- r 读。
- w 写
- x 执行
- - 表示缺少相应的权限
- X 大写 X 表示只设置执行权限给目录，普通文件不设置。

Setfacl 选项

- -m ：添加或修改权限
- -x ： 删除权限

例：

1) 添加或修改所有者或命令用户的 ACL:

To add or modify a *user* or *named user* ACL:

```
[student@serverX ~]$ setfacl -m u:name:rX file
```

- 如果 name 为空，那么应用到文件所有者，否则命名用户能用用户名或 UID。
- ACL 文件所有者和标准文件所有者权限是等效的，因此，用 chmod 和 setfacl 是等效的。命名用户使用 chomd 无效。

2) 添加或修改属组或命名组的 ACL

To add or modify a *group* or *named group* ACL:

```
[student@serverX ~]$ setfacl -m g:name:rw file
```

- Chmod 对任何组设置都无效。

3) 添加或修改 other ACL

To add or modify the *other* ACL:

```
[student@serverX ~]$ setfacl -m o::- file
```

- \- 表示没有 ACL 权限，但标准的权限能用。
- ACL other 和标准文件 other 权限是等效的，因此，用 chmod 和 setfacl 是等效的。

4) 同一命令添加多个项，用逗号隔开

```
[student@serverX ~]$ setfacl -m u::rwx,g:sodor:rX,o::- file
```

5) 用 getfacl 输出做为 setfacl 的输入

```
[root@localhost ~]# getfacl file-A | setfacl - -set-file=- file-B
```

6) 设置一个明确的掩码

```
[student@serverX ~]$ setfacl -m m::r file
```

7) 递归设置

```
[root@localhost ~]# setfacl -R -m u:name:rX  directory
```

8) 删除 ACL

```
[root@localhost ~]# setfacl -x u:name,g:name file
```

2、控制默认的 ACLs 文件权限

```
[student@serverX ~]$ setfacl -m d:u:name:rx directory
```

Deleting a *default* ACL is also the same as deleting a standard ACL; again, preface with **d:**, or use the **-d** option.

```
[student@serverX ~]$ setfacl -x d:u:name directory
```

小结：

1、文件所有者属组权限，setfacl 和 chmod 设置等效。

2、命名用户（除文件所有者和 root）和命名组,只能 setfalc 设置权限。

4、other 用户权限，setfacl 和 chmod 设置等效。

第七章　管理 SELinux 安全

7.1 启用和监测 SELinux

1、 基本 SELinux 安全概念

计算机安全等级：a、b、c、d。开启 selinux，linux 系统升为 b 级，没有开启 selinux，只是在 c 级；当然 c 级也再分为 c1 c2...；windows NT 也只不过 C2 级。

Linux 文件权限分为 2 种：

DAC=Discretionary Access Control 传统直接访问控制。

► 以用户为出发点，来管理权限----root。DAC 缺点：因为 root 具有最高权限。

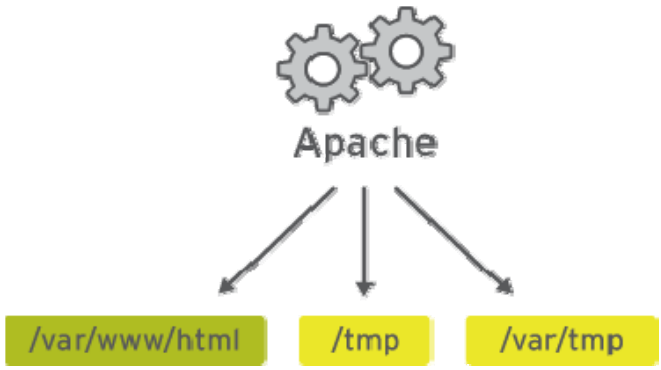
MAC=Mandatory Access Control 强制访问控制。

► 以程序为出发点----http 能访问文件的权限，由策略决定的。

SELinux(Secure Enhanced Linux)安全增强的 Linux 是由美国国家安全局 NSA 针对计算机基础结构安全开发的一个全新的 Linux 安全策略机制。SELinux 可以允许系统管理员更加灵活的来定义安全策略。SELinux 是一个内核级别的安全机制，从 Linux2.6 内核之后就将 SELinux 集成在了内核当中，因为 SELinux 是内核级别的，所以我们对于其配置文件的修改都是需要重新启动操作系统才能生效的。现在主流发现的 Linux 版本里面都集成了 SELinux 机制，CentOS/RHEL 都会默认开启 SELinux 机制。

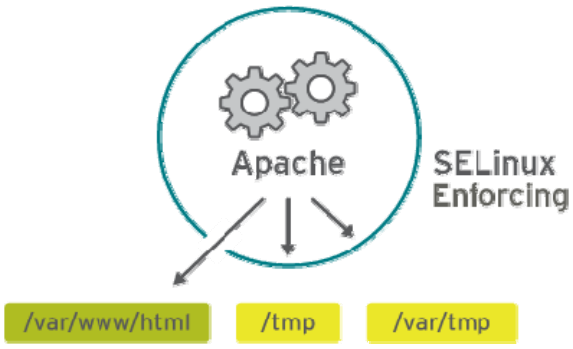
安全增强 Linux （SELinux） 是系统安全的一个额外层。SELinux 的一个主要目标是从被破坏的系统服务保护用户的数据。大多数的 Linux 管理员熟悉标准用户/组其他权限的安全模型。这是一个用户和基于组的模型称为自由访问控制。SELinux 提供额外的安全是基于对象和更复杂的规则，称为强制访问控制控制层。

Apache 服务在没有启用 SELinux 时：



允许远程用户访问 WEB 服务器，防火墙端口必须打开。然而，这给恶意用户通过安全漏洞攻击系统提供机会。危险的进程，获取 apache 用户和组的权限，就能读/var/www/html 目文档，也读写/tmp、/var/tmp 和任何可读写的文件和目录。

Apache 服务在启用 SELinux 时：



SELinux 设置安全规则，决定哪些进程可以访问哪些文件、目录和端口。每个文件、目录、进程和端口都有一个特定的安全标签叫做 SELinux 安全上下文。上下文用做 SELinux 策略决定一个进程是否可以访问文件、目录或端口。默认情况下，这个策略不允许交互式访问，除非有一个明确规则授予访问。如果没有允许规，没有访问被允许。

我们知道，操作系统的安全机制其实就是对两样东西做出限制：进程和系统资源(文件、网络套接字、系统调用等)。

在之前学过的知识当中，Linux 操作系统是通过用户和组的概念来对我们的系统资源进行限制，我们知道每个进程都需要一个用户才能执行。

在 SELinux 当中针对这两样东西定义了两个基本概念：域(domin)和上下文(context)。

域就是用来对进程进行限制，而上下文就是对系统资源进行限制。

SELinux 就是限制哪些域可以访问哪些系资源。换句说：限制进程可以访问哪些文件等资源。

在 RHEL 中默认策略为 targeted 策略。目标策略定义了只有目标进程受到 SELinux 限制，非目标进程就不会受到 SELinux 限制，通常我们的网络应用程序都是目标进程，比如 httpd、mysqld，dhcpd 等等这些网络应用程序。

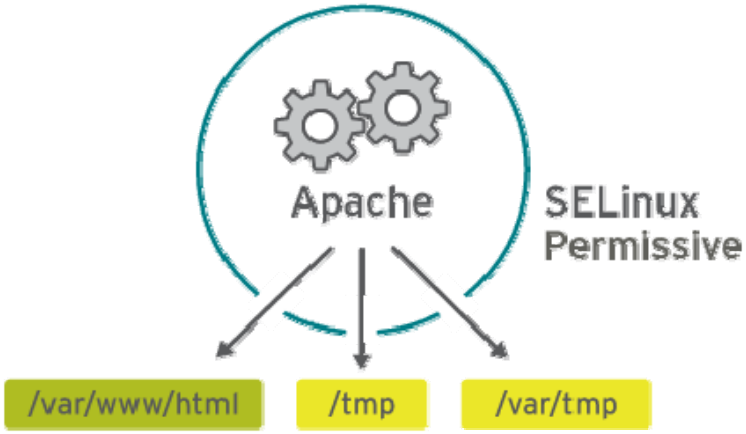
1） ps、ls 等命令带-Z 选项查看 SELinux 设置。

```
[root@serverX ~]# ps axZ
LABEL                               PID TTY      STAT   TIME COMMAND
system_u:system_r:init_t:s0           1 ?        Ss     0:09 /usr/lib/systemd/...
system_u:system_r:kernel_t:s0         2 ?        S      0:00 [kthreadd]
system_u:system_r:kernel_t:s0         3 ?        S      0:00 [ksoftirqd/0]
[... Output omitted ...]
[root@serverX ~]# systemctl start httpd
[root@serverX ~]# ps -ZC httpd
LABEL                               PID TTY          TIME CMD
system_u:system_r:httpd_t:s0       1608 ?        00:00:05 httpd
system_u:system_r:httpd_t:s0       1609 ?        00:00:00 httpd
[... Output omitted ...]
[root@serverX ~]# ls -Z /home
drwx------. root    root    system_u:object_r:lost_found_t:s0 lost+found
drwx------. student student unconfined_u:object_r:user_home_dir_t:s0 student
drwx------. visitor visitor unconfined_u:object_r:user_home_dir_t:s0 visitor
[root@serverX ~]# ls -Z /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 error
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 icons
```

2 、 SELinux 的模式

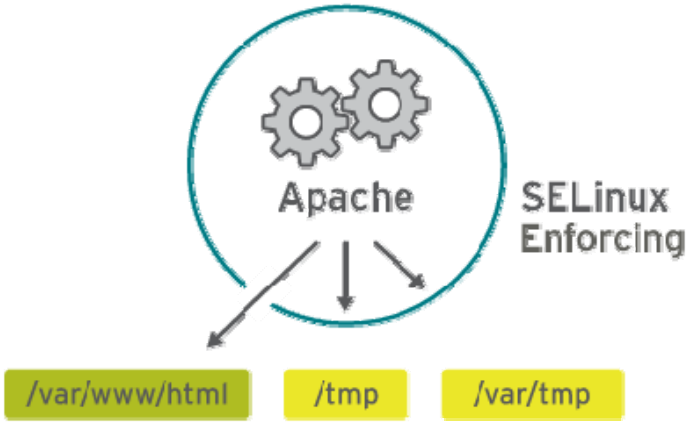
1）SELinux Permissive（允许模式）

允许模式 SELinux 保护被临时禁用，主要用于排错目的。在这种模式下，所有交互都会允许，不需要重启就可以在允许模式和强制模式间切换。



2）SELinux Enforcing（强制模式）

强制模式只要是违反策略的行动都被禁止，并作为内核信息记录。如：SELinux 拒绝用 tmp_t 上下文访问 WEB 服务器。SELinux 提供保护和日志。



3）SELinux Disables（禁用模式）

完全禁用 SELinux，和其他模式之间切换需要重启系统。

重要：

用允许模式比完全关闭 SELinux 要好,原因是在允许模式，系统内核自动维护 SELinux 系

统必需的标签，避免当系统重启需要启用 SELinux 时，重新标签文件系统代价很大。

查询 SELinux 模式状态

```
[root@serverX ~]# getenforce
Enforcing
```

3 、 SELinux 的 Booleans（布尔值）

SELinux 布尔值是改变 SELinux 策略行为的一个开关，SELinux 布尔值是能启用或禁用的规则，他能用做安全管理员选择性调整安全策略。

Getsebool 命令显示 SELinux 布尔值，-a 选项列出所有。

The **getsebool** command is used to display SELinux Booleans and their current value. The **-a** option causes this command to list all of the Booleans.

```
[root@serverX ~]# getsebool -a
abrt_anon_write --> off
allow_console_login --> on
allow_corosync_rw_tmpfs --> off
[... Output omitted ...]
```

7.2 更改 SELinux 模式

1、 更改当前 SELinux 模式

The **setenforce** command modifies the current SELinux mode:

```
[root@serverX ~]# getenforce
Enforcing
[root@serverX ~]# setenforce
usage:  setenforce [ Enforcing | Permissive | 1 | 0 ]
[root@serverX ~]# setenforce 0
[root@serverX ~]# getenforce
Permissive
[root@serverX ~]# setenforce Enforcing
[root@serverX ~]# getenforce
Enforcing
```

2、 更改默认 SELinux 模式

The configuration file that determines what the SELinux mode is at boot time is **/etc/selinux/config**. Notice that it contains some useful comments:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes
#               are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Use **/etc/selinux/config** to change the default SELinux mode at boot time. In the example shown, it is set to enforcing mode.

7.2 更改 SELinux 上下文

1、查看 SELinux 上下文

典型的 SELinux 上下文由文件的父目录初始决定的，用命令 cp、vim、touch 建文件时，父目录分配上下文到新建的文件。然而，如果这个文件在其他地方创建并设置权限保留（如：mv 或 cp -a），那么文件的上下文不变。

```
[root@serverX ~]# ls -Zd /var/www/html/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
[root@serverX ~]# touch /var/www/html/index.html
[root@serverX ~]# ls -Z /var/www/html/index.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0
/var/www/html/index.html
```

2、修改 SELinux 上下文

有两个命令可以修改 SELinux 上下文，chcon 和 restorecon。

- chcon 用 -t 选项修改 SELinux 上下文。不能用来修改文件的上下文，因为文件在文件系统重新标签会恢复的默认的上下文。
- restorecon 是参考一个文件或目录的上下文来修改，-v 显示更改，-R 递归。

```
[root@serverX ~]# mkdir /virtual
[root@serverX ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual
[root@serverX ~]# chcon -t httpd_sys_content_t /virtual
[root@serverX ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 /virtual
[root@serverX ~]# restorecon -v /virtual
restorecon reset /virtual context unconfined_u:object_r:httpd_sys_content_t:s0->
unconfined_u:object_r:default_t:s0
[root@serverX ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual
```

3、定义 SELinux 默认文件的上下文规则

Semanage fcontext 命令可以查看或修改用 restorecon 设置的默认上下文，用扩展的正则

表达式指定路径和文件名，在 fcontext 中最常用的正则表达式是 (/.*)? 。

```
[root@serverX ~]# touch /tmp/file1 /tmp/file2
[root@serverX ~]# ls -Z /tmp/file*
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /tmp/file1
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /tmp/file2
[root@serverX ~]# mv /tmp/file1 /var/www/html/
[root@serverX ~]# cp /tmp/file2 /var/www/html/
[root@serverX ~]# ls -Z /var/www/html/file*
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /var/www/html/file1
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/
html/file2
[root@serverX ~]# semanage fcontext -l
...
/var/www(/.*)?       all files    system_u:object_r:httpd_sys_content_t:s0
...
[root@serverX ~]# restorecon -Rv /var/www/
restorecon reset /var/www/html/file1 context unconfined_u:object_r:user_tmp_t:s0
-> system_u:object_r:httpd_sys_content_t:s0
[root@serverX ~]# ls -Z /var/www/html/file*
-rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0
   /var/www/html/file1
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0
   /var/www/html/file2
```

下面的例子显示怎样用 semanage 为一个新目录添加一个上下文。

The following example shows how to use **semanage** to add a context for a new directory.

```
[root@serverX ~]# mkdir /virtual
[root@serverX ~]# touch /virtual/index.html
[root@serverX ~]# ls -Zd /virtual/
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual/
[root@serverX ~]# ls -Z /virtual/
-rw-r--r--. root root unconfined_u:object_r:default_t:s0 index.html
[root@serverX ~]# semanage fcontext -a -t httpd_sys_content_t '/virtual(/.*)?'
[root@serverX ~]# restorecon -RFvv /virtual
[root@serverX ~]# ls -Zd /virtual/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /virtual/
[root@serverX ~]# ls -Z /virtual/
-rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0 index.html
```

7.3 更改 SELinux 布尔值

1、SELinux 布尔值

SELinux 布尔值是改变 SELinux 策略行为的一个开关，SELinux 布尔值是能启用或禁用的规则，他能用做安全管理员选择性调整安全策略。

getsebool 命令显示 SELinux 布尔值，-a 选项列出所有。

The **getsebool** command is used to display SELinux Booleans and their current value. The **-a** option causes this command to list all of the Booleans.

```
[root@serverX ~]# getsebool -a
abrt_anon_write --> off
allow_console_login --> on
allow_corosync_rw_tmpfs --> off
[... Output omitted ...]
```

setsebool 命令用于修改布尔值，setsebool -P 可能修改并持久修改的策略。Setsebool -l 显示简短的描述信息和是否持久。

The **getsebool** command is used to display SELinux Booleans and **setsebool** is used to modify them. **setsebool -P** modifies the SELinux policy to make the modification persistent. **semanage boolean -l** will show whether or not a Boolean is persistent, along with a short description of the Boolean.

```
[root@serverX ~]# getsebool -a
abrt_anon_write --> off
abrt_handle_event --> off
abrt_upload_watch_anon_write --> on
antivirus_can_scan_system --> off
antivirus_use_jit --> off
...
[root@serverX ~]# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off
[root@serverX ~]# setsebool httpd_enable_homedirs on
[root@serverX ~]# semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs          (on   ,  off)  Allow httpd to enable homedirs
[root@serverX ~]# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on
[root@serverX ~]# setsebool -P httpd_enable_homedirs on
[root@serverX ~]# semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs          (on   ,   on)  Allow httpd to enable homedirs
```

To only list local modifications to the state of the SELinux booleans (any setting that differs from the default in the policy), the command **semanage boolean -l -C** can be used.

```
[root@serverX ~]# semanage boolean -l -C
SELinux boolean                State  Default Description

cron_can_relabel               (off   ,   on)  Allow cron to can relabel
```

7.4 SELinux 排错

1、排除 SELinux 问题

排除 SELinux 问题的步骤：

1）在做任何调整之前，先考虑正在工作的的 SELinux 可能禁止试图的访问。如果一个 WEB 服务器试图访问/home 中文件，如果这个文件没有被用户发布，预示一个个危险信号。如果访问已被授权，看另外步骤去解决这个问题。

2）最常见的问题是不正确的文件上下文。这个发生在一个文件在一个地方建立，移动到其他地方而上下文不是预期的。大部分情况下，运行 restorecon 能改决这个问题。这种纠正对系统其他部分的安全影响非常小。

3) 另一种太过严格的限制需要调整布尔值。例如：ftpd_anon_write 布尔值控制匿名用户是否能上载文件，必须打开这个布尔值 ON 才能让匿名用户能上载文件。调整布尔值要非常小心因为他对系统安全有大的影响。

4）有可能是 SELinux 策略有一个 bug 阻止合法的访问，这个是比较罕见的因为 SELinux 技术已成熟。当很明显是一个策略 bug，请联系红帽技术支持。

2、监视 SELinux 违规

必须安装 setroubleshoot-server 包，发送 SELinxu 消息到 /var/log/messages 。setroubleshoot-server 侦听/var/log/audit/audit.log 并发短摘要消息到/var/log/messages。这个摘要消息包含 UUID 可以用来收集更深层的消息。Sealert -l uuid 可以用做为特定的事件生成一个报告。

Consider the following sample sequence of commands on a standard Apache web server:

```
[root@serverX ~]# touch /root/file3
[root@serverX ~]# mv /root/file3 /var/www/html
[root@serverX ~]# systemctl start httpd
[root@serverX ~]# curl http://localhost/file3
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /file3
on this server.</p>
</body></html>
```

Even though the contents of **file3** are expected, the web server returns a **permission denied** error. Inspecting both **/var/log/audit/audit.log** and **/var/log/messages** can reveal some extra information about this error.

```
[root@serverX ~]# tail /var/log/audit/audit.log
...
type=AVC msg=audit(1392944135.482:429): avc:  denied  { getattr } for
  pid=1609 comm="httpd" path="/var/www/html/file3" dev="vda1" ino=8980981
  scontext=system_u:system_r:httpd_t:s0
  tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file
...
[root@serverX ~]# tail /var/log/messages
...
Feb 20 19:55:42 serverX setroubleshoot: SELinux is preventing /usr/sbin/httpd
  from getattr access on the file . For complete SELinux messages. run
  sealert -l 613ca624-248d-48a2-a7d9-d28f5bbe2763
```

第八章　配置 网络用户连接

8.1 使用身份管理服务

1、 用户信息和身份验证

一个集中的身份管理系统将需要提供至少两个服务：

帐户信息： 这包括信息，如用户名、 家里的目录位置，UID 和 GID，组成员身份，等等。最受欢迎的解决方案包括 LDAP（轻量级目录访问协议），在 Active Directory 和 IPA 服务器和网络信息服务 (NIS) 等多个产品中使用。

身份验证信息：

可以通过系统来验证用户是谁。这可以通过提供加密的密码哈希值到客户端系统，或者通过将 （加密的） 密码发送到服务器，并接收响应。

LDAP 服务器可以提供身份验证信息和帐户信息。Kerberos 只提供 SSO 身份验证服务，和旁边 LDAP 用户信息，通常使用。使用 Kerberos 的使用在 IPA 服务器和 Active Directory 中。

在 RHEL 7 系统上，本地用户信息是由提供 /etc/passwd ，认证信息（在哈希密码的形式）存在/etc/shadow。

8.2 将一个系统附加到集中的 LDAP 和 Kerberos 服务器

1、 认证配置

与验证有关的配置文件

/etc/ldap.conf： LDAP 服务器设置信息

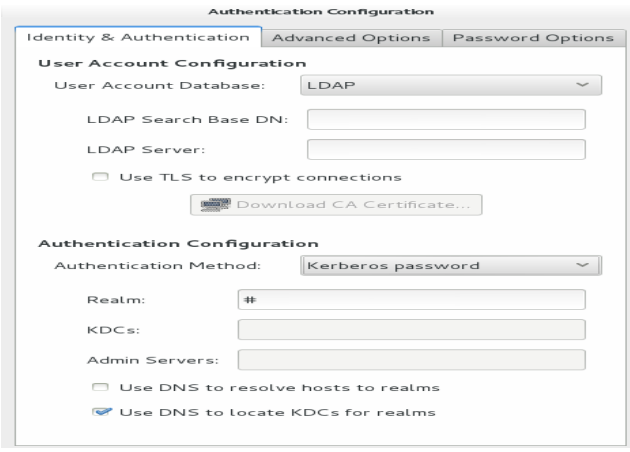
/etc/krb5.conf：Kerberos 基础设施有关的信息

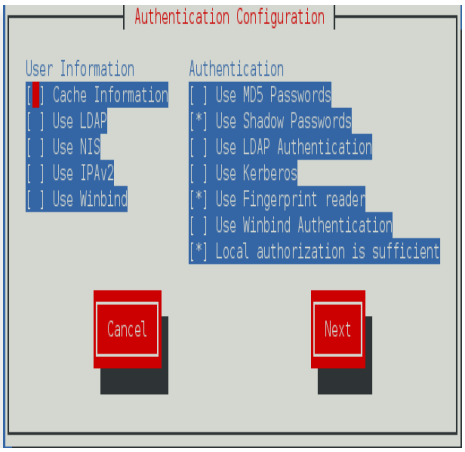
/etc/sssd/sssd.conf： 要配置系统安全服务守护进程 （sssd），守护进程负责检索和缓存用户信息和身份验证信息

/etc/nsswitch.conf： 指示到系统应使用哪些用户信息和身份验证服务

/etc/pam.d/*： 配置身份验证应如何处理各项服务

/etc/openldap/cacerts： 用于存储可以验证用于标识 LDAP 服务器的 SSL 证书的根证书颁发机构 (CA)





2、 必要的 LDAP 参数

要连接到一个中央的 LDAP 服务器的用户信息，authconfig 需要大量的设置：

（1） LDAP 服务器的主机名

（2.） 基本 DN （可分辨名称） 的系统应在其中查找用户的 LDAP 树的一部分。这通常看起来像 dc = example，dc = com 或 ou = 人，o = PonyCorp。此信息将会由 LDAP 服务器管理员提供。

（3）如果使用 SSL/TLS 加密与 LDAP 服务器的通信，可以验证证书的根 CA 证书是由 LDAP 服务器提供的。

重要提示： 系统还将需要一些额外的软件包安装提供 LDAP 客户端的功能。

3、 必要的 Kerberos 参数

要将系统配置为使用一个集中的 Kerberos 系统进行用户身份验证，authconfig 将需要以下设置：

（1） .Kerberos 领域使用的名称。Kerberos 境界是域中的机器都使用一套共同的 Kerberos 服务器和用户进行身份验证。

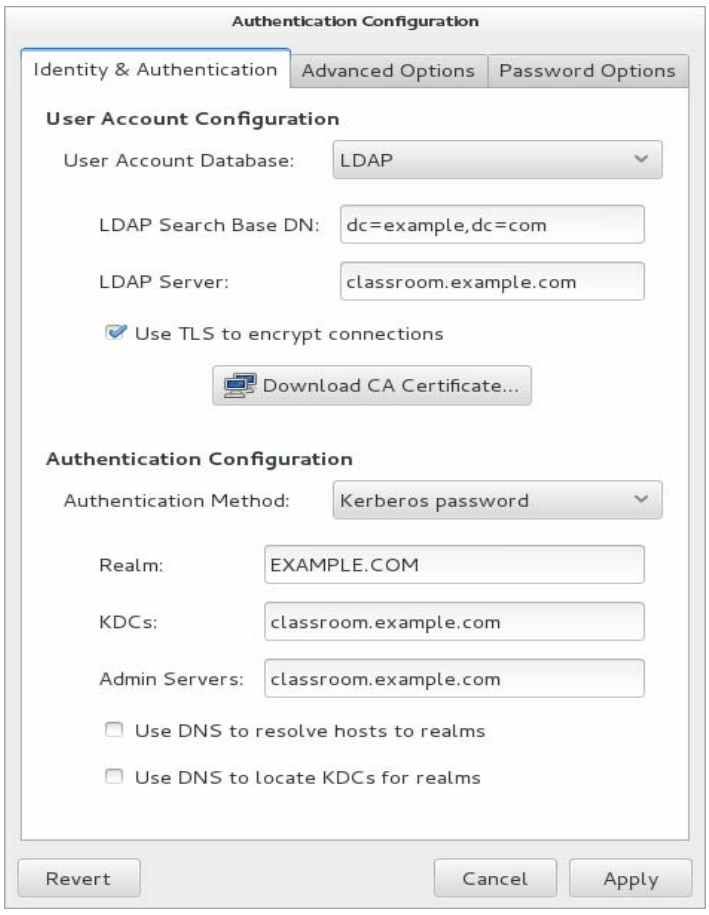
（2）.一个或多个密钥分发中心 (KDC)。这是您的 Kerberos 服务器的主机名。

（3.）一个或多个管理服务器的主机名。这是当他们想要更改其密码，或执行其他用户修改客户端会说话的机器。这通常是主 KDC，相同，但它可以是不同的机器。

（4.）此外，管理员可以指定是否应使用 DNS 来查找要用于特定的主机名称，并自动找到 Kdc 和 admin 服务器的境界。可以安装额外的软件包，来帮助调试 Kerberos 问题，并与 Kerberos 票证从命令行： krb5 工作站。

4、 配置 LDAP + Kerberos 验证

4. If the LDAP server supports TLS, check the **Use TLS to encrypt connections** box, and use the **Download CA Certificate** button to download the CA certificate.
5. From the **Authentication Method** dropdown, select **Kerberos password**, and fill out the **Realm**, **KDCs**, and **Admin Servers** fields. The last two fields are not available if the **Use DNS to locate KDCs for realms** option is checked.
6. If central home directories are not available, users can create directories on first login by checking the **Create home directories on the first login** box on the **Advanced Options** tab.
7. Click the **Apply** button to save and activate the changes. This will write all relevant configuration files and (re)start the **sssd** service.



8.3 将一个系统连接到 IPA 服务器

Red Hat 提供一个集成的解决方案来配置 LDAP 和 Kerberos： IPA （身份、 政策和审计） 服务器。IPA 服务器提供 LDAP 和 Kerberos，结合这两个命令行和基于 web 的管理工具套件。除了用户和身份验证信息，IPA 服务器可以集中 sudo 规则，SSH 公共密钥、SSH 主机密钥，TLS 证书、 自动挂载的地图，以及更多。

1、 使用 ipa-客户端

红色帽子企业 Linux 7 系统可以配置为使用 IPA 服务器使用 authconfig 套件的工具，但也存在着一个专门的工具： ipa-client-install。此命令可以从 ipa-client 客户端软件包，拉取所有依赖项的安装。

使用 ipa-client-install 的好处之一是它可以从 DNS （配置时由 IPA 服务器或由管理员手动），检索几乎所有必要的信息，以及创建主机条目和更多的 IPA 服务器上。允许 IPA 服务器管理员设置该主机的访问策略，创建服务 （例如 NFSv4 出口），和更多。

8.4 加入活动目录

The following is an example of using **realmd** to join an Active Directory domain, and allow Active Directory users to log into the local system. This example assumes that the Active Directory domain is called **domain.example.com**.

1. Install the necessary packages: *realmd*.

```
[student@demo ~]$ yum -y install realmd
```

2. Discover the settings for the **domain.example.com** domain.

```
[student@demo ~]$ sudo realm discover domain.example.com
```

3. Join the Active Directory domain; this will install all necessary packages, and configure **sssd**, **pam**, **/etc/nsswitch.conf**, etc.

```
[student@demo ~]$ sudo realm join domain.example.com
```

This will attempt to join the local system to Active Directory using the **Administrator** account; enter the password for this account when prompted. To use a different account, use the **--user** argument.

4. Active Directory accounts are now usable on the local system, but logins using Active Directory are still disabled. To enable logins, use the following command:

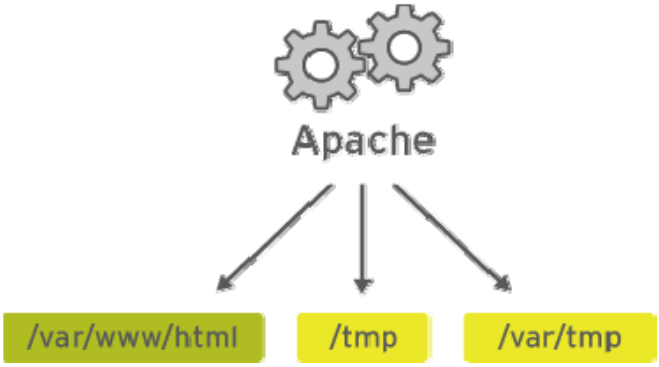
```
[student@demo ~]$ sudo realm permit --realm domain.example.com --all
```

To only allow certain users to log in, replace **--all** with a list of those users. For example:

```
[student@demo ~]$ sudo realm permit --realm domain.example.com DOMAIN\\Itchy
DOMAIN\\Scratchy
```

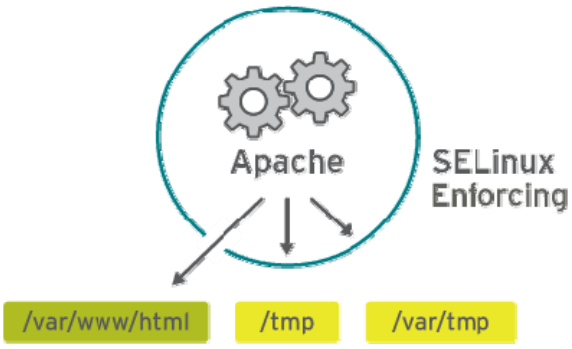
安全增强 Linux （SELinux） 是系统安全的一个额外层。SELinux 的一个主要目标是从被破坏的系统服务保护用户的数据。大多数的 Linux 管理员熟悉标准用户/组其他权限的安全模型。这是一个用户和基于组的模型称为自由访问控制。SELinux 提供额外的安全是基于对象和更复杂的规则，称为强制访问控制控制层。

Apache 服务在没有启用 SELinux 时：



允许远程用户访问 WEB 服务器，防火墙端口必须打开。然而，这给恶意用户通过安全漏洞攻击系统提供机会。危险的进程，获取 apache 用户和组的权限，就能读/var/www/html 目文档，也读写/tmp、/var/tmp 和任何可读写的文件和目录。

Apache 服务在启用 SELinux 时：



SELinux 设置安全规则，决定哪些进程可以访问哪些文件、目录和端口。每个文件、目录、进程和端口都有一个特定的安全标签叫做 SELinux 安全上下文。上下文用做 SELinux 策略决定一个进程是否可以访问文件、目录或端口。默认情况下，这个策略不允许交互式访问，除非有一个明确规则授予访问。如果没有允许规，没有访问被允许。

在 EHEL 中默认策略为 targeted 策略。目标策略定义了只有目标进程受到 SELinux 限制，非目标进程就不会受到 SELinux 限制，通常我们的网络应用程序都是目标进程，比如 httpd、mysqld，dhcpd 等等这些网络应用程序。

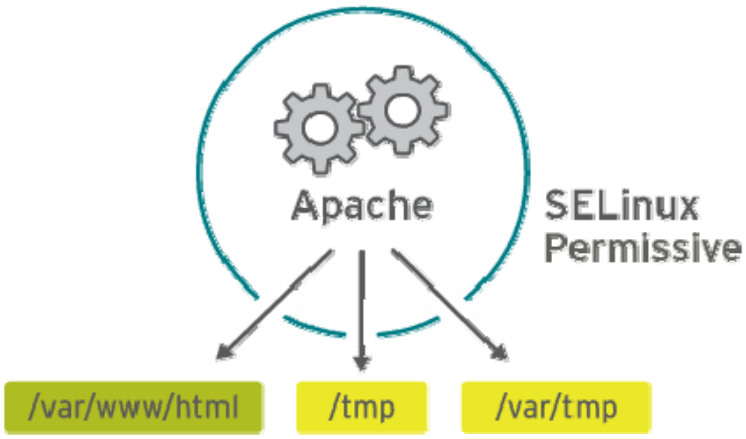
1） ps、ls 等命令带-Z 选项查看 SELinux 设置。

```
[root@serverX ~]# ps axZ
LABEL                               PID TTY      STAT   TIME COMMAND
system_u:system_r:init_t:s0           1 ?        Ss     0:09 /usr/lib/systemd/...
system_u:system_r:kernel_t:s0         2 ?        S      0:00 [kthreadd]
system_u:system_r:kernel_t:s0         3 ?        S      0:00 [ksoftirqd/0]
[... Output omitted ...]
[root@serverX ~]# systemctl start httpd
[root@serverX ~]# ps -ZC httpd
LABEL                               PID TTY          TIME CMD
system_u:system_r:httpd_t:s0       1608 ?        00:00:05 httpd
system_u:system_r:httpd_t:s0       1609 ?        00:00:00 httpd
[... Output omitted ...]
[root@serverX ~]# ls -Z /home
drwx------. root    root    system_u:object_r:lost_found_t:s0 lost+found
drwx------. student student unconfined_u:object_r:user_home_dir_t:s0 student
drwx------. visitor visitor unconfined_u:object_r:user_home_dir_t:s0 visitor
[root@serverX ~]# ls -Z /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 error
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 icons
```

2 、 SELinux 的模式

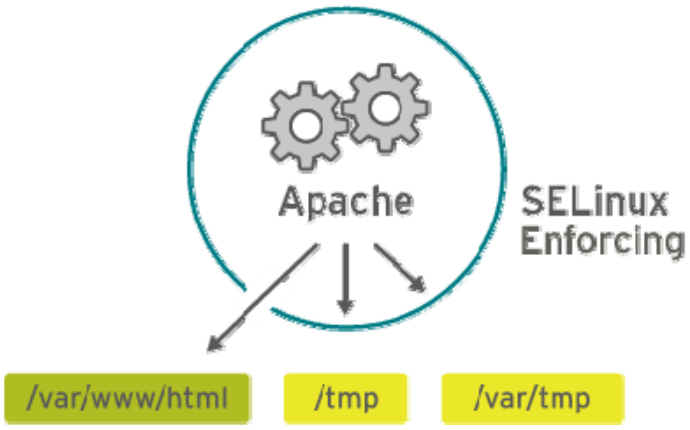
1）SELinux Permissive（允许模式）

允许模式 SELinux 保护被监时禁用，主要用于排错目的。在这种模式下，所有交互都会允许，不需要重启就可以在允许模式和强制模式间切换。



2）SELinux Enforcing（强制模式）

强制模式 SELinux 拒绝用 tmp_t 上下文访问 WEB 服务器。SELinux 提供保护和日志。



3）SELinux Disables（禁用模式）

完全禁用 SELinux，和其他模式之间切换需要重启系统。

重要：

用允许模式比完全关闭 SELinux 要好,原因是在允许模式，系统内核自动维护 SELinux 系统必需的标签，避免当系统重启需要启用 SELinux 时，重新标签文件系统代价很大。

查询 SELinux 模式状态

```
[root@serverX ~]# getenforce
Enforcing
```

3 、 SELinux 的 Booleans（布尔值）

SELinux 布尔值是改变 SELinux 策略行为的一个开关，SELinux 布尔值是能启用或禁用的规则，他能用做安全管理员选择性调整安全策略。

Getsebool 命令显示 SELinux 布尔值，-a 选项列出所有。

The **getsebool** command is used to display SELinux Booleans and their current value. The **-a** option causes this command to list all of the Booleans.

```
[root@serverX ~]# getsebool -a
abrt_anon_write --> off
allow_console_login --> on
allow_corosync_rw_tmpfs --> off
[... Output omitted ...]
```

7.2 更改 SELinux 模式

1、 更改当前 SELinux 模式

The **setenforce** command modifies the current SELinux mode:

```
[root@serverX ~]# getenforce
Enforcing
[root@serverX ~]# setenforce
usage:  setenforce [ Enforcing | Permissive | 1 | 0 ]
[root@serverX ~]# setenforce 0
[root@serverX ~]# getenforce
Permissive
[root@serverX ~]# setenforce Enforcing
[root@serverX ~]# getenforce
Enforcing
```

2、 更改默认 SELinux 模式

The configuration file that determines what the SELinux mode is at boot time is **/etc/selinux/config**. Notice that it contains some useful comments:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes
#               are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Use **/etc/selinux/config** to change the default SELinux mode at boot time. In the example shown, it is set to enforcing mode.

7.2 更改 SELinux 上下文

1、查看 SELinux 上下文

典型的 SELinux 上下文由文件的父目录初始决定的，用命令 cp、vim、touch 建文件时，父目录分配上下文到新建的文件。然而，如果这个文件在其他地方创建并设置权限保留（如：mv 或 cp -a），那么文件的上下文不变。

```
[root@serverX ~]# ls -Zd /var/www/html/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
[root@serverX ~]# touch /var/www/html/index.html
[root@serverX ~]# ls -Z /var/www/html/index.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0
/var/www/html/index.html
```

2、修改 SELinux 上下文

有两个命令可以修改 SELinux 上下文，chcon 和 restorecon。

- Chcon 用 -t 选项修改 SELinux 上下文。不能用来修改文件的上下文，因为文件在

文件系统重新标签会恢复的默认的上下文。

- Restorecon 是参考一个文件或目录的上下文来修改，-v 显示更改，-R 递归。

```
[root@serverX ~]# mkdir /virtual
[root@serverX ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual
[root@serverX ~]# chcon -t httpd_sys_content_t /virtual
[root@serverX ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 /virtual
[root@serverX ~]# restorecon -v /virtual
restorecon reset /virtual context unconfined_u:object_r:httpd_sys_content_t:s0->
unconfined_u:object_r:default_t:s0
[root@serverX ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual
```

3、定义 SELinux 默认文件的上下文规则

Semanage fcontext 命令可以查看或修改用 restorecon 设置的默认上下文，用扩展的正则表达式指定路径和文件名，在 fcontext 中最常用的正则表达式是 (/.*)? 。

```
[root@serverX ~]# touch /tmp/file1 /tmp/file2
[root@serverX ~]# ls -Z /tmp/file*
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /tmp/file1
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /tmp/file2
[root@serverX ~]# mv /tmp/file1 /var/www/html/
[root@serverX ~]# cp /tmp/file2 /var/www/html/
[root@serverX ~]# ls -Z /var/www/html/file*
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /var/www/html/file1
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/
html/file2
[root@serverX ~]# semanage fcontext -l
...
/var/www(/.*)?       all files    system_u:object_r:httpd_sys_content_t:s0
...
[root@serverX ~]# restorecon -Rv /var/www/
restorecon reset /var/www/html/file1 context unconfined_u:object_r:user_tmp_t:s0
-> system_u:object_r:httpd_sys_content_t:s0
[root@serverX ~]# ls -Z /var/www/html/file*
-rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0
   /var/www/html/file1
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0
   /var/www/html/file2
```

下面的例子显示怎样用 semanage 为一个新目录添加一个上下文。

The following example shows how to use **semanage** to add a context for a new directory.

```
[root@serverX ~]# mkdir /virtual
[root@serverX ~]# touch /virtual/index.html
[root@serverX ~]# ls -Zd /virtual/
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual/
[root@serverX ~]# ls -Z /virtual/
-rw-r--r--. root root unconfined_u:object_r:default_t:s0 index.html
[root@serverX ~]# semanage fcontext -a -t httpd_sys_content_t '/virtual(/.*)?'
[root@serverX ~]# restorecon -RFvv /virtual
[root@serverX ~]# ls -Zd /virtual/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /virtual/
[root@serverX ~]# ls -Z /virtual/
-rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0 index.html
```

7.3 更改 SELinux 布尔值

1、SELinux 布尔值

SELinux 布尔值是改变 SELinux 策略行为的一个开关，SELinux 布尔值是能启用或禁用的规则，他能用做安全管理员选择性调整安全策略。

getsebool 命令显示 SELinux 布尔值，-a 选项列出所有。

The **getsebool** command is used to display SELinux Booleans and their current value. The **-a** option causes this command to list all of the Booleans.

```
[root@serverX ~]# getsebool -a
abrt_anon_write --> off
allow_console_login --> on
allow_corosync_rw_tmpfs --> off
[... Output omitted ...]
```

setsebool 命令用于修改布尔值，setsebool -P 可能修改并持久修改的策略。Setsebool -l 显示简短的描述信息和是否持久。

The **getsebool** command is used to display SELinux Booleans and **setsebool** is used to modify them. **setsebool -P** modifies the SELinux policy to make the modification persistent. **semanage boolean -l** will show whether or not a Boolean is persistent, along with a short description of the Boolean.

```
[root@serverX ~]# getsebool -a
abrt_anon_write --> off
abrt_handle_event --> off
abrt_upload_watch_anon_write --> on
antivirus_can_scan_system --> off
antivirus_use_jit --> off
...
[root@serverX ~]# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off
[root@serverX ~]# setsebool httpd_enable_homedirs on
[root@serverX ~]# semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs          (on   ,  off)  Allow httpd to enable homedirs
[root@serverX ~]# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on
[root@serverX ~]# setsebool -P httpd_enable_homedirs on
[root@serverX ~]# semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs          (on   ,   on)  Allow httpd to enable homedirs
```

To only list local modifications to the state of the SELinux booleans (any setting that differs from the default in the policy), the command **semanage boolean -l -C** can be used.

```
[root@serverX ~]# semanage boolean -l -C
SELinux boolean                State  Default Description

cron_can_relabel               (off   ,   on)  Allow cron to can relabel
```

7.4 SELinux 排错

1、排除 SELinux 问题

排除 SELinux 问题的步骤：

1）在做任何调整之前，先考虑正在工作的的 SELinux 可能禁止试图的访问。如果一个 WEB 服务器试图访问/home 中文件，如果这个文件没有被用户发布，预示一个个危险信号。如果访问已被授权，看另外步骤去解决这个问题。

2）最常见的问题是不正确的文件上下文。这个发生在一个文件在一个地方建立，移动到其他地方而上下文不是预期的。大部分情况下，运行 restorecon 能改决这个问题。这种纠正对系统其他部分的安全影响非常小。

3) 另一种太过严格的限制需要调整布尔值。例如：ftpd_anon_write 布尔值控制匿名用户是否能上载文件，必须打开这个布尔值 ON 才能让匿名用户能上载文件。调整布尔值要非常小心因为他对系统安全有大的影响。

4）有可能是 SELinux 策略有一个 bug 阻止合法的访问，这个是比较罕见的因为 SELinux 技术已成熟。当很明显是一个策略 bug，请联系红帽技术支持。

3、监视 SELinux 违规

必须安装 setroubleshoot-server 包，发送 SELinxu 消息到 /var/log/messages 。setroubleshoot-server 侦听/var/log/audit/audit.log 并发短摘要消息到/var/log/messages。这个摘要消息包含 UUID 可以用来收集更深层的消息。Sealert -l uuid 可以用做为特定的事件生成一个报告。

Consider the following sample sequence of commands on a standard Apache web server:

```
[root@serverX ~]# touch /root/file3
[root@serverX ~]# mv /root/file3 /var/www/html
[root@serverX ~]# systemctl start httpd
[root@serverX ~]# curl http://localhost/file3
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /file3
on this server.</p>
</body></html>
```

Even though the contents of **file3** are expected, the web server returns a **permission denied** error. Inspecting both **/var/log/audit/audit.log** and **/var/log/messages** can reveal some extra information about this error.

```
[root@serverX ~]# tail /var/log/audit/audit.log
...
type=AVC msg=audit(1392944135.482:429): avc:  denied  { getattr } for
  pid=1609 comm="httpd" path="/var/www/html/file3" dev="vda1" ino=8980981
  scontext=system_u:system_r:httpd_t:s0
  tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file
...
[root@serverX ~]# tail /var/log/messages
...
Feb 20 19:55:42 serverX setroubleshoot: SELinux is preventing /usr/sbin/httpd
  from getattr access on the file . For complete SELinux messages. run
  sealert -l 613ca624-248d-48a2-a7d9-d28f5bbe2763
```

第九章　磁盘分区和文件系统

9.1 添加添加分区、文件系统和永久挂载点分区

9.1.1、 磁盘分区

- MBR 分区方案

自 1982 年以来的主引导扇区 (MBR) 分区方案取决于如何应进行磁盘分区运行 BIOS 固件系统上。这种方案可支持最多四个主分区。在 Linux 系统上，使用的扩展和逻辑分区，管理员可以创建最多 15 分区。磁盘分区的 MBR 方案有 最大 2 TiB 磁盘和分区大小的限制。

- GPT 分区方案

对于系统运行统一可扩展固件接口 (UEFI) 固件，GPT 是物理硬盘上的分区表的标准。GPT 是 UEFI 标准的一部分，解决了很多旧的基于 MBR 的所施加的限制。每 UEFI 规格，GPT 默认为支持达 128 个分区。 允许 GPT 以容纳分区和磁盘的最多 8 ZiB。

注意：

GTP 8ZiB 是基于 512 字节的块大小，硬件厂商转换成 4096 位。所以支持 64ZiB

位 bit(比特)(BinaryDigits)：存放一位二进制数，即 0 或 1，最小的存储单位。

字节 byte：8 个二进制位为一个字节(B)，最常用的单位。

1KB(Kilobyte 千字节)=1024B，

1MB(Megabyte 兆字节简称“兆”)=1024KB，

1GB(Gigabyte 吉字节又称“千兆”)=1024MB，

1TB(Trillionbyte 万亿字节太字节)=1024GB，其中 1024=2^10(2 的 10 次方)，

1PB（Petabyte 千万亿字节拍字节）=1024TB，

1EB（Exabyte 百亿亿字节艾字节）=1024PB，

1ZB(Zettabyte 十万亿亿字节泽字节)=1024EB,

1YB(Yottabyte 一亿亿亿字节尧字节)=1024ZB,

1BB(Brontobyte 一千亿亿亿字节)=1024YB.

9.1.2 使用 fdisk 管理 MBR 分区

服务器管理人员应该知道如何在服务器上添加硬盘，添加硬盘设备后要保证计算机系统能够识别该设备。若在 Vmware 虚拟机环境中，可以通过“VM”—“Settings”添加一块硬盘。

1、 检测新硬盘

要了解系统连接新磁盘的情况，可以运行 fdisk -l 命令。

```
[root@zx ~]# fdisk -l

Disk /dev/sda: 40.8 GB, 40802189312 bytes
255 heads, 63 sectors/track, 4960 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0002c467

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          26      204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2              26        1046     8192000   83  Linux
/dev/sda3            1046        1110      512000   82  Linux swap / Solaris
/dev/sda4            1110        4961    30936064    5  Extended
/dev/sda5            1110        4961    30935040   83  Linux

Disk /dev/sdb: 85.9 GB, 85899345920 bytes
255 heads, 63 sectors/track, 10443 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/sdb doesn't contain a valid partition table
[root@zx ~]#
```

在上图中，首先列出了系统上第一个磁盘/dev/sda 的基本信息，包括磁盘的总容量以及磁头、扇区、柱面等参数。接下来是分区表，可以看到该磁盘分为 3 个主分区，1 个扩展分区以及在扩展分区中的 1 个逻辑分区，其中/dev/sda1 为引导分区；此外，还可以看到每个分区的起始柱面数、结束柱面数和分区类型。

上述信息的最后一行表示磁盘/dev/sdb（新加的硬盘）上没有合法的磁盘分区表，即尚未进行分区。

以磁盘/dev/sdb 为例，启动 fdisk 分区工具进行分区操作。

2、 规划硬盘中的分区

可以在 Shell 提示符下以管理员身份输入命令：fdisk /dev/sdb

```
[root@zx ~]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0x4524493a.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help):
```

上面的命令表示已进入 fdisk 交互式操作，根据提示信息可以输入命令 m 以查看各个交互式命令的使用方法：

```
Command (m for help): m
Command action
   a   toggle a bootable flag          //引导标志开关
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition                        //删除一个分区
   l   list known partition types
   m   print this menu
   n   add a new partition                      //建立一个新分区
   o   create a new empty DOS partition table
   p   print the partition table                  //显示分区表
   q   quit without saving changes                不保存退出
   s   create a new empty Sun disklabel
   t   change a partition's system id      //改变分区的系统ID
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit           保存退出
   x   extra functionality (experts only)

Command (m for help):
```

(1)　“p”——显示硬盘中原有分区情况

```
Command (m for help): p

Disk /dev/sdb: 85.9 GB, 85899345920 bytes
255 heads, 63 sectors/track, 10443 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x4524493a

   Device Boot      Start         End      Blocks   Id  System

Command (m for help):
```

新加的硬盘尚未建立分区，所以上表分区显示为空。

（2）“n”——新建分区

根据提示选择“”为创建主分区，选择“”为创建扩展分区。之后依次选择分区序号、起始位置、结束位置或分区大小完成新分区创建。

选择分区号时，主分区和扩展分区的序号只能在 1-4 之间。分区的启始位置一般默认即可，结束位置或大小可以使用“+sizeM”的形式，如“+10000M”表示该分区的容量设置为 10GB。

下面以创建两个主分区和两个逻辑分区为例。

首先创建一个主分区（/dev/sdb1），容量指定为 20GB。

```
Command (m for help): n                    //开始创建分区
Command action
   e   extended
   p   primary partition (1-4)
p                                          //选择创建主分区
Partition number (1-4): 1                  //设置第一主分区编号
First cylinder (1-10443, default 1):       //启始位置，直接回车接受默认值
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-10443, default 10443): +20000M
                                                                   //分区容量大小
Command (m for help):
```

按照相同的步骤继续创建第二个主分区（/dev/sdb2），容量也为 20GB，完成后可以输入“p”查看分区情况，如下所示。

```
Command (m for help): p

Disk /dev/sdb: 85.9 GB, 85899345920 bytes
255 heads, 63 sectors/track, 10443 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x4524493a

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1        2551    20490876   83  Linux
/dev/sdb2            2552        5102    20490907+  83  Linux

Command (m for help):
```

接下来创建一个扩展分区和两个逻辑分区。

先建立扩展分区（/dev/sdb4），使用剩下的所有空间。

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
e                                          //创建扩展分区
Partition number (1-4): 4                  //分区编号
First cylinder (5103-10443, default 5103):    //启始位置，直接回车
Using default value 5103
Last cylinder, +cylinders or +size{K,M,G} (5103-10443, default 10443):
Using default value 10443                                  //结束位置，直接回车

Command (m for help):
```

在扩展分区中建立第一个逻辑分区（/dev/sdb5），容量为指定为 10GB。

```
Command (m for help): n
Command action
   l   logical (5 or over)
   p   primary partition (1-4)
l                                          //选择L创建逻辑分区
First cylinder (5103-10443, default 5103):
Using default value 5103
Last cylinder, +cylinders or +size{K,M,G} (5103-10443, default 10443): +10000M

Command (m for help):
```

按照相同的步骤继续创建第二个逻辑分区（/dev/sdb5），容量为所有剩余空间。完成后可以输入“p”查看分区情况，如下所示。

```
Command (m for help): p

Disk /dev/sdb: 85.9 GB, 85899345920 bytes
255 heads, 63 sectors/track, 10443 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x4524493a

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1        2551    20490876   83  Linux
/dev/sdb2            2552        5102    20490907+  83  Linux
/dev/sdb4            5103       10443    42901582+   5  Extended
/dev/sdb5            5103        6378    10249438+  83  Linux
/dev/sdb6            6379       10443    32652081   83  Linux

Command (m for help):
```

(3) “d”——删除分区

使用“d”可以删除分区，根据提法输入要删除的分区的序号即可。在删除分区时一定要慎重，首先使用 “p”查看分区的序号，确认后再删除。需要注意的是，如果是扩展分区被删除，则扩展分区之下的逻辑分区也同时被删除。因此建议从最后一个分区开始删除。

下面的操作过程将删除上一步创建的逻辑分区/dev/sdb6。

```
Command (m for help): d          //删除分区
Partition number (1-6): 6        //要删除的分区序号

Command (m for help): p

Disk /dev/sdb: 85.9 GB, 85899345920 bytes
255 heads, 63 sectors/track, 10443 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x4524493a

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1        2551    20490876   83  Linux
/dev/sdb2            2552        5102    20490907+  83  Linux
/dev/sdb4            5103       10443    42901582+   5  Extended
/dev/sdb5            5103        6378    10249438+  83  Linux

Command (m for help):
```

（4）“t”——改变分区类型

新建分区的类型默认为 Linux，id 为 83。如果要修改分区的类型，可以使用命令 t。使用命令 l 查看已知的分区类型及 id。

(5)“w”和“q”——退出 fdisk 工具。

最后要保存修改后的分区表，可以使用 w 命令。如果不想保存修改的结果，可以使用 q 命令。

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
[root@zx ~]#
```

变更硬盘（特别是正在使用的硬盘）的分区设置后，建议最好重启系统，或者执行“partprobe /dev/sdbv”命令使操作出了系统获知新的分区情况。

9.1.3 建立文件系统

我们使用 fdisk 创建分区后，还需要对分区进行格式化并挂载到系统中的指定的目录，才能用于存储数据。本节将学习如何格式化并挂载分区。

文件系统是操作系统在硬盘或者分区上保存、读取文件信息的方法和数据结构，也就是文件在硬盘或分区上的组织方式。

现在 Linux 系统默认使用 xfs 文件系统。

使用分区工具新建立的分区是没有文件系统的。要想在分区上存储数据，首先需要建

立文件系统。建立文件系统的过程类似于格式化。命令使用 mkfs，语法如下：

Mkfs -t 文件系统名 分区设备

例：要在分区/dev/sdb1 上建立 ext4 文件系统，可以使用如下命令：

```
[root@zx ~]# mkfs -t ext4 /dev/sdb1
mke2fs 1.41.12 (17-May-2010)
文件系统标签=
操作系统:Linux
块大小=4096 (log=2)
分块大小=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
1281120 inodes, 5122719 blocks
256135 blocks (5.00%) reserved for the super user
第一个数据块=0
Maximum filesystem blocks=0
157 block groups
32768 blocks per group, 32768 fragments per group
8160 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2
        4096000

正在写入inode表: 完成
Creating journal (32768 blocks): 完成
Writing superblocks and filesystem accounting information: 完成

This filesystem will be automatically checked every 30 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@zx ~]#
```

建立文件系统会将分区上的数据全部删除，所以执行这项操作要小心谨慎。

9.1.4 挂载、卸载文件系统

建立了文件系统后，还需要将文件系统安装到 Linux 目录树的某个位置上才能使用。这个过程称为挂载（mount），文件系统所挂载到的目录称为挂载点。除了磁盘分区之外，其他各种存储设备也想要进行挂载才能使用。

通常应该讲文件系统挂在到某个可以存取的空目录下，而且该目录应该是专门为挂载某个文件系统而建立的。Linux 系统提供了专门的挂载点/mnt，可以将临时使用的外部设备挂载到/mnt 目录上。

1、手动挂载文件系统

文件系统的挂载，可以在系统引导过程中自动挂载，也可以使用命令手工挂载，手工挂载文件系统的命令是 mount，该命令的语法是：

mount [-t 文件系统] 存储设备 挂载点

如果 mount 命令不加任何选项执行，则是现实当前系统中已经挂在的分区：

例：新建文件夹/soft，把上节中建立的/dev/sdb1 分区挂载到该目录中。

```
[root@zx /]# mkdir soft
[root@zx /]# mount /dev/sdb1 /soft
[root@zx /]#
```

2、卸载文件系统

要卸载已经挂载的设备，可使用命令 umount。

如要卸载前面挂载的分区/dev/sdb1，可使用如下两个命令中的任一个：

（1）通过挂载点目录位置来卸载。（建议方法）

```
[root@zx /]# umount /soft
[root@zx /]#
```

（2）通过对应设备名来卸载。

```
[root@zx /]# umount /dev/sdb1
[root@zx /]#
```

正在使用的文件系统不能卸载。如果正在访问文件系统的某个文件或者当前目录是要在卸载的文件系统上的目录，应该关闭文件或者退出当前目录，然后再执行卸载。

3、/etc/fstab 文件永久挂载

有时候用户可能需要一直使用某个文件系统或者需要挂载多个文件系统，一个一个手工挂载就比较麻烦了。这时可以使用配置文件/etc/fstab 来决定如何挂载设备。

该文件一共分为六列：

fs-spec　　fs-file　　fs-vfstype　　fs-mntops　　fs-freq　　fs-passno

各项含义如下：

- fs-spec：将要挂载的设备
- fs-file：文件系统的挂载点
- fs-vfstype：文件系统类型
- fs-mntops：挂载选项，传递给 mount 命令以决定如何挂载。各选项使用逗号分开
- fs-freq：由 dump 程序决定文件系统是否需要备份
- fs-passno：由 fsck 程序决定引导时是否检查磁盘以及检查的文件

/etc/fstab 文件的一个实例如下所示，不同的系统会有所不同。

用 blkid 命令可以看到所有设备的标签和文件系统类型

```
/etc/fstab
UUID=uuid /mountpoint ext4 default 1 2
```

```
# /etc/fstab
# Created by anaconda on Wed Jan  4 06:31:32 2012
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=1b3ea893-6f77-4e25-924f-1c0f7451fb40 /                       ext4    defaults
        1 1
UUID=611f0608-4487-44e3-b83e-49bb62383a95 /boot                   ext4    defaults
        1 2
UUID=98ab1a29-04bf-4946-8371-f84f105f323c /var                    ext4    defaults
        1 2
UUID=d4e51219-8eeb-4d3c-bbfa-ce7089a9612b swap                    swap    defaults
        0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
[root@zx /]#
```

例：修改/etc/fstab 文件，添加自动挂载/dev/sdb1 分区到 soft 目录的配置行。

```
[root@zx /]# tail -1 /etc/fstab
/dev/sdb1               /soft                   ext4    defaults        0 0
[root@zx /]#
```

9.1.5 查看磁盘命用情况

Df 命令可以查看系统中已挂载的各文件系统的磁盘使用情况。常用的选项为“-h”、“-T”，使用“-h”将显示更易读懂的容量单位，而“-T”可以显示对应的文件类型。

例：查看当前系统中挂载的文件系统的磁盘使用情况。

```
[root@zx /]# df -hT
文件系统     类型      容量  已用  可用 已用%% 挂载点
/dev/sda2      ext4    7.7G  5.6G  1.8G  76% /
tmpfs         tmpfs    330M  264K  330M   1% /dev/shm
/dev/sda1      ext4    194M   36M  148M  20% /boot
/dev/sda5      ext4     30G  349M   28G   2% /var
/dev/sr0    iso9660    2.8G  2.8G     0 100% /media/RHEL_6.
/dev/sdc1      vfat     3.9G   16M  3.9G   1% /media/usb
/dev/sr0    iso9660    2.8G  2.8G     0 100% /media/cdrom
/dev/loop0 iso9660     266K  266K     0 100% /mnt/iso
/dev/sdb1      ext4     20G  172M   19G   1% /soft
[root@zx /]#
```

9.1.6 用 gdisk 管理 GTP 分区

Creating GPT disk partitions

There are eight steps required to create a GPT-style partition.

1. Specify the disk device to create the partition on.
   Execute the **gdisk** command and specify the disk device name as an argument. This will start the **gdisk** command in interactive mode, and will present a **command** prompt.

```
[root@serverX ~]# gdisk /dev/vdb
GPT fdisk (gdisk) version 0.8.6

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries.

Command (? for help):
```

2. Request a new partition.
   Enter **n** to create a new partition.

```
Command (? for help): n
```

3. Specify the partition number.
   This partition number serves as the identification number of the partition on the disk for use in future partition operations. The default value is the lowest unused partition number.

```
Partition number (1-128, default 1): 1
```


李伟阶 QQ：17533203

4. Specify the disk location that the new partition will start from.

   gdisk allows for two different input types. The first input type is an absolute disk sector number representing the first sector of the new partition.

```
First sector (34-20971486, default = 2048) or {+-}size{KMGTP}: 2048
```

   The second input type indicates the partition's starting sector by its position relative to the first or last sector of the first contiguous block of free sectors on the disk. Using this relative sector position format, input is specified in units of KiB, MiB, GiB, TiB, or PiB.

   For example, a value of **+**512M signifies a sector position that is 512 MiB **after** the beginning of the next group of contiguous available sectors. On the other hand, a value of **-**512M denotes a sector positioned 512 MiB *before* the end of this group of contiguous available sectors.

5. Specify the last sector on the disk that the new partition will end on.

   The default value is the last of the available, unallocated sectors contiguous to the new partition's first sector.

```
Last sector (2048-20971486, default = 20971486) or {+-}size{KMGTP}: 1050623
```

   In addition to the absolute ending sector number, **gdisk** also offers the more user-friendly input option of specifying the end boundary of the new partition in units of KiB, MiB, GiB, TiB, or PiB from the beginning or end of the group of contiguous available sectors. A value of **+**512M signifies an ending partition position that is 512 MiB **after** the first sector.

```
Last sector (2048-20971486, default = 20971486) or {+-}size{KMGTP}: +512M
```

   A value of **-**512M indicates an ending partition position that is 512 MiB *before* the end of the contiguous available sectors.

```
Last sector (2048-20971486, default = 20971486) or {+-}size{KMGTP}: -512M
```

6. Define partition type.

   New partitions created by **gdisk** default to type Linux file system. If a different partition type is desired, enter the corresponding hex code. If needed, a table of the hex codes for all partition types can be displayed with the **L** command.

```
Current type is 'Linux filesystem'

Hex code or GUID (L to show codes, Enter = 8300): 8e00
Changed type of partition to 'Linux LVM'
```

7. Save partition table changes.

   Issue the **w** command to finalize the partition creation request by writing the changes to the disk's partition table. Enter **y** when **gdisk** prompts for a final confirmation.

```
Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/vdb.
The operation has completed successfully.
```

8. Initiate a kernel re-read of the new partition table.

   Run the **partprobe** command with the disk device name as an argument to force a re-read of its partition table.

```
[root@serverX ~]# partprobe /dev/vdb
```

Creating file systems

After a block device has been created, the next step is applying a file system format to it. A file system applies a structure to the block device so that data can be stored and retrieved from it. Red Hat Enterprise Linux supports many different file system types, but two common ones are **xfs** and **ext4**. **xfs** is used by default in **anaconda**, the installer for Red Hat Enterprise Linux.

The **mkfs** command can be used to apply a file system to a block device. If no type is specified, an extended type two (ext2) file system will be used, which for many uses is not desirable. To specify the file system type, a **-t** should be used.

```
[root@serverX ~]# mkfs -t xfs /dev/vdb1
meta-data=/dev/vdb1              isize=256    agcount=4, agsize=16384 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=0
data     =                       bsize=4096   blocks=65536, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=0
log      =internal log           bsize=4096   blocks=853, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

Manually mounting file systems

Administrators can use the **mount** command to manually attach the device onto a directory location, or *mount point*, by specifying the device and the mount point, as well as any options that may be desired, to customize the behavior of the device.

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

The **mount** can also be used to view currently mounted file systems, the mount points, and options.

```
[root@serverX ~]# mount | grep vdb1
/dev/vdb1 on /mnt type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

Manually mounting a file system is an excellent way to verify that a formatted device is accessible or working in the way desired. However, once the system is rebooted, the file system, while it still exists and has intact data, will not be mounted into the directory tree again. If an administrator wants the file system to be persistently mounted, a listing for the file system needs to be added to **/etc/fstab**.

Persistently mounting file systems

By adding a listing for a device into the **/etc/fstab** file, administrators can configure a device to be mounted to a mount point at system boot.

/etc/fstab is a white space-delimited file with six fields per line.

```
[root@serverX ~]# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Thu Mar 20 14:52:46 2014
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=7a20315d-ed8b-4e75-a5b6-24ff9e1f9838  /  xfs  defaults  1 1
```

The first field specifies the device to be used. In the previous example, the **UUID** is being used to specify the device. Alternatively, the device file could be used; for example, **/dev/vdb1**. The **UUID** is stored in the file system superblock and created when the file system is created.

9.2 管理交换分区

若要创建交换空间，管理员需要做三件事：

1、创建一个分区。

2、设置为 82 Linux 交换的分区的类型。

3、格式化激活交换分区。

一、制作一个交换分区，大小为 500MB ，并使其在系统重启后自动挂载（永久挂载）。

(1)创建分区作为 swap 分区：fdisk（注意要把文件系统的标签改为 82）

```
# fdisk /dev/sda
 p（查看分区表）
 n（新建分区，扩展分区按 e，主分区按 p） :p
 再按 3
 启始回车
   +500（定义大小）
 t（改变分区类型）
   3（输入分区号）
   l（查看分区类型说明）
   82（swap 分区类型是 82）
 w（保存分区表）
# reboot（重启，或用 partprobe ）
```

(2)格式 swap 分区：mkswap /dev/vdaX (X 是你分出来的分区的号)

(3)激活 swap 分区：swapon /dev/vdaX

(4)查看 swap 分区：swapon -s

(5)写入/etc/fstab

```
vim /etc/fstab
  /dev/vdaX swap swap defaults 0 0
```

二、制作一个交换空间，大小为 756MB，并且使其在系统重启后依然能自动被挂载（永久挂载）。（以大文件的方式制作交换空间）不要移除和更改你系统上现存的交换分区。

```
dd if=/dev/zero of=fileswap bs=1M count=756
                                  #生成一个大小为 756M 的文件
mkswap fileswap
swapon fileswap
vim /etc/fstab
  /root/fileswap  swap  swap  defaults   0  0
```

验证

cat /proc/swaps ，查看会发现新建的 file。

也可用 free -l 或 swapon -s 验证

第十章　管理逻辑卷 LVM

10.1 逻辑卷 LVM 概念

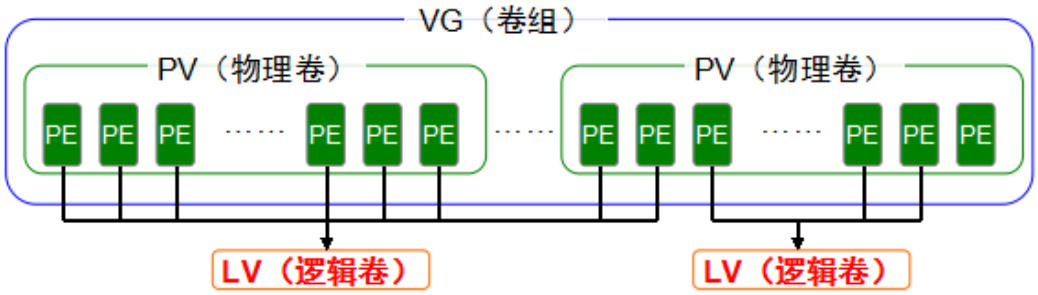
每个 Linux 使用者在安装 Linux 时都会遇到这样的困境：在为系统分区时，如何精确评估和分配各个硬盘分区的容量，因为系统管理员不但要考虑到当前某个分区需要的容量，还要预见该分区以后可能需要的容量的最大值。因为如果估计不准确，当遇到某个分区不够用时管理员可能甚至要备份整个系统、清除硬盘、重新对硬盘分区和重新安装操作系统，然后恢复数据到新分区。

Linux 提供的逻辑盘卷管理（LVM，LogicalVolumeManager）机制就是一个完美的解决方案。Linux 的 LVM 功能非常强大，可以在生产运行系统上面直接在线扩展硬盘分区，可以把分区 umount 以后收缩分区大小，还可以在系统运行过程中把一个分区从一块硬盘搬到另一块硬盘上面去等等，简直就像变魔术，而且这一切都可以在一个繁忙运行的系统上面直接操作，不会对你的系统运行产生任何影响，很安全。

LVM 是逻辑盘卷管理（LogicalVolumeManager）的简称，它是 Linux 环境下对磁盘分区进行管理的一种机制，LVM 是建立在硬盘和 分区之上的一个逻辑层，来提高磁盘分区管理的灵活性。通过 LVM 系统管理员可以轻松管理磁盘分区，如：将若干个磁盘分区连接为一个整块的卷组 （volumegroup），形成一个存储池。管理员可以在卷组上随意创建逻辑卷组（logicalvolumes），并进一步在逻辑卷组上创建文件系 统。管理员通过 LVM 可以方便的调整存储卷组的大小，并且可以对磁盘存储按照组的方式进行命名、管理和分配，例如按照使用用途进行定义： “development”和“sales”，而不是使用物理磁盘名“sda”和“sdb”。而且当系统添加了新的磁盘，通过 LVM 管理员就不必将磁盘的 文件移动到新的磁盘上以充分利用新的存储空间，而是直接扩展文件系统跨越磁盘即可。

在学习之前首先需要了解 LVM 基本术语。

- **PV（Physical Volume，物理卷）**
  - 整个硬盘，或使用 fdisk 等工具建立的普通分区
  - 包括许多默认 4MB 大小的 PE（Physical Extent，基本单元）。
  - 物理卷一般直接使用设备名称，如/dev/sdb1。
- **VG（Volume Group，卷组）**
  - 一个或多个物理卷组成一个整体，称为卷组，在卷组中可以动态添加或移除物理卷
- **LV（Logical Volume，逻辑卷）**
  - 从卷组中分割出的一块空间，用于建立文件系统



通过上述对物理卷、卷组、逻辑卷的解释可以看出，建立 LVM 分区管理机制的过程就是：将普通分区或整块硬盘建立物理卷；将各物理卷的存储空间组成一个逻辑整体即卷组；最后，基于卷组这个整体，分割出不同的数据存储空间，形成逻辑卷。而逻辑卷才是最终用户可以格式化并挂载使用的存储单位。

LVM 管理命令

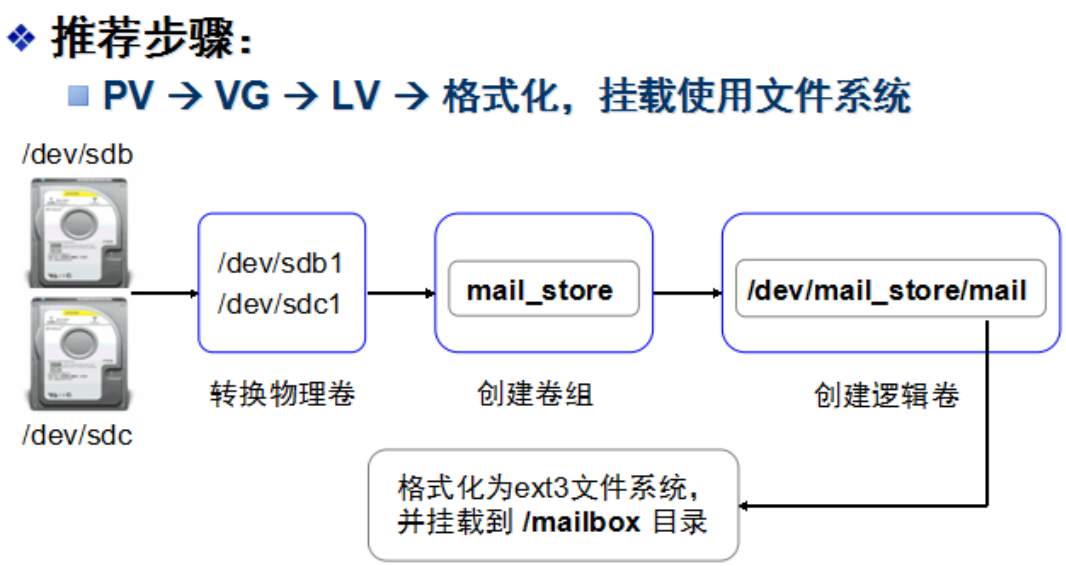
功能	物理卷管理	卷组管理	逻辑卷管理
Scan 扫描	pvscan	vgscan	lvscan
Create 建立	pvcreate	vgcreate	lvcreate
Display 显示	pvdisplay	vgdisplay	lvdisplay
Remove 删除	pvremove	vgremove	lvremove
Extend 扩展		vgextend	lvextend

主要命令的用法

```
pvcreate  设备名
vgcreate  卷组名  物理卷名 1 物理卷名 2
lvcreate   -L  大小  -n  逻辑卷名  卷组名
lvextend  -L  +大小  /dev/卷组名/逻辑卷名
```

10.2 管理 LVM

环境和需求描述如下：公司准备在 Internet 中搭建自己的邮件服务器，面向全国各地的员工提供电子邮箱空间。由于用户数量众多，邮件存储需要大量的存储空间，考虑到动态扩容的需要，计划增加两块 SCSI 硬盘并构建 LVM 逻辑卷（ 挂载到“/mailbox 目录下”）专门用于存放邮件数据。



为了完成公司需求，操作步骤如下：

一、基本操作

1、 正确安装两块 SCSI 硬盘。

2、 开启服务器主机，并执行 fdisk -l 命令确认已识别新增的硬盘。

3、 在新增的磁盘中进行分区，将每块硬盘的所有空间划分为一个独立的主分区。

4、 将/dev/sdc1 和/dev/sdd1 分区转为物理卷。

```
[root@zx ~]# pvcreate /dev/sdc1
  Physical volume "/dev/sdc1" successfully created
[root@zx ~]# pvcreate /dev/sdd1
  Physical volume "/dev/sdd1" successfully created
[root@zx ~]#
```

5、 将上述第一块物理卷整合，创建一个名为 vg_mail 的卷组。

```
[root@zx ~]# vgcreate vg_mail /dev/sdc1
  Volume group "vg_mail" successfully created
[root@zx ~]#
```

可用 vgdisplay 查看

6、 在 vg_mail 卷组中创建一个名为 lv_mail 的逻辑卷，容量大小设置为 60G.

```
[root@zx ~]# lvcreate -L 60G -n lv_mail vg_mail
  Logical volume "lv_mail" created
[root@zx ~]#
```

可用 lvdisplay 查看

7、 使用 mkfs 命令在 lv_mail 逻辑卷中创建 ext4 文件系统，并挂载到/mailbox 目录

下。

```
[root@zx ~]# mkfs -t ext4 /dev/vg_mail/lv_mail
```

```
[root@zx ~]# mkdir /mailbox
[root@zx ~]# mount /dev/vg_mail/lv_mail /mailbox
[root@zx ~]# df -hT /mailbox
文件系统     类型      容量  已用  可用 已用% 挂载点
/dev/mapper/vg_mail-lv_mail
               ext4     60G  180M   56G   1% /mailbox
[root@zx ~]#
```

8、在 vg_mail 卷组中创建一个 lv_share 逻辑卷，大小为 10G

```
# lvcreate -L 10G -n lv_share vg_mail
```

9、在 lv_share 逻辑卷中创建 xfs 文件系统,并挂载到/share。(开机自动挂载写/etc/fstab 文件)

```
# mkfs -t xfs /dev/vg_mail/lv_share
# mkdir /share
# mount /dev/vg_mail/lv_share  /share
# df -h
```

二、动态扩展操作

1、把上面第 4 步中创建的物理卷/dev/sdd1,扩展到卷组 vg_mail。

```
# vgextent vg_mail  /dev/sdd1
```

2、确认扩展成功。

```
# vgdisplay vg_mail
```

3、动态扩展 ext4 文件系 lv_mail 逻辑卷的容量(增加 10G)，并更新系统识别文件系统的大小。

```
[root@zx ~]# lvextend -L +10G /dev/vg_mail/lv_mail
  Extending logical volume lv_mail to 70.00 GiB
  Logical volume lv_mail successfully resized
[root@zx ~]# resize2fs /dev/vg_mail/lv_mail
```

```
[root@zx ~]# df -hT /mailbox
文件系统     类型      容量  已用  可用 已用% 挂载点
/dev/mapper/vg_mail-lv_mail
               ext4     69G  180M   66G   1% /mailbox
[root@zx ~]#
```

4、动态扩展 xfs 文件系 lv_share 逻辑卷的容量(增加 10G)，并更新系统识别文件系统的大小。

```
# lvextent -L +10G /dev/vg_mail/lv_share
# xfs_growfs /share
```

注意：ext4 和 xfs 重新识别文件系统命令不同。

三、缩小 ext4 文件系统逻辑卷 lv_mail 到 200M。

(1)缩小文件系统：必须先要卸载：umount /mailbox
(2)检查文件系统： e2fsck -f /dev/vg_mail/lv_mail
(3)先缩小是文件系统: resize2fs /dev/vg_mail/lv_mail 200M
(4)再缩小空间： lvreduce -L 220M /dev/vg_mail/lv_mail
(5)挂载起来: mount -a
(6)检查： df -h

三、从卷组中移除物理卷/dev/sdd1

1、把/dev/sdd1 上的数据移到上/dev/sdc1 上

```
# pvmove /dev/sdd1 /dev/sdc1
```

注意这个过程不能被打断，建议在转移数据之前先备份数据。

2、从卷组中移除物理卷/dev/sdd1

```
# vgreduce vgmail /dev/sdd1
```

第十一章 访问网络文件系统 (NFS)

1.1 手动挂载与卸载 NFS 共享

NFS 网络文件系统，是作为 Linux、UNIX 和类似操作系统网络文件系统使用的的网络标准协议。 它是一个开放的标准，根据活动的扩展，支持本机 Linux 权限和文件系统功能。

RHEL 7 支持 NFSv4，NFSv4 使用 TCP 协议与服务器进行通信，而旧版本的 NFS 可能使用 TCP 或 UDP。NFS 服务器共享 （目录） 和 NFS 客户端装载到一个本地挂载点 （目录） 的出口。 本地挂载点必须存在。NFS 共享可以安装多种方式：

- 手动挂载 NFS 共享使用 mount 命令。
- 在启动时使用 /etc/fstab 自动挂载 NFS 共享。
- 通过一个称为自动挂载程序按需挂载 NFS 共享。

```
[student@desktopX ~]$ sudo systemctl enable nfs-secure
ln -s '/usr/lib/systemd/system/nfs-secure.service'  ...
[student@desktopX ~]$ sudo systemctl start nfs-secure
```

Note

The **nfs-secure** service is part of the **nfs-utils** package, which should be installed by default. If it is not installed, use:

```
[student@desktopX ~]$ sudo yum -y install nfs-utils
```

There are three basic steps to mounting an NFS share:

1. **Identify:** The administrator for the NFS server can provide export details, including security requirements. Alternatively:
   NFSv4 shares can be identified by mounting the root folder of the NFS server and exploring the exported directories. Do this as **root**. Access to shares that are using Kerberos security will be denied, but the share (directory) name will be visible. Other share directories will be browsable.

   ```
   [student@desktopX ~]$ sudo mkdir /mountpoint
   [student@desktopX ~]$ sudo mount serverX:/ /mountpoint
   [student@desktopX ~]$ sudo ls /mountpoint
   ```

   NFSv2 and NFSv3 shares can be discovered using **showmount**.

   ```
   [student@desktopX ~]$ showmount -e serverX
   ```

2. **Mount point:** Use **mkdir** to create a mount point in a suitable location.

   ```
   [student@desktopX ~]$ mkdir -p /mountpoint
   ```

3. **Mount:** There are two choices here: manually or incorporated in the **/etc/fstab** file. Switch to *root* or use **sudo** for either operation.
   - *Manual:* Use the **mount** command.

     ```
     [student@desktopX ~]$ sudo mount -t nfs -o sync serverX:/share /mountpoint
     ```

     The **-t nfs** option is the file system type for NFS shares (not strictly required, shown for completeness). The **-o sync** option tells **mount** to immediately synchronize write operations with the NFS server (the default is asynchronous). The default security method (sec=sys) will be used to try mounting the NFS share, using standard Linux file permissions.
   - */etc/fstab:* Use **vim** to edit the **/etc/fstab** file and add the mount entry to the bottom of the file. The NFS share will be mounted at each system boot.

     ```
     [student@desktopX ~]$ sudo vim /etc/fstab
     ...
     serverX:/share  /mountpoint  nfs  sync  0 0
     ```

Use **umount**, using *root* privileges, to manually unmount the share.

```
[student@desktopX ~]$ sudo umount /mountpoint
```

1.2 自动挂载 NFS 共享

自动挂载程序是一个服务（autofs)，可以"按需"自动挂载 NFS 共享，他们不再被使用时，将自动卸载 NFS 共享。

挂载的好处：

1.用户不需要有根权限才能运行的装载/卸载命令。

2.配置在挂载 NFS 共享上的机器，根据访问权限的所有用户可用。

3.NFS 共享不永久连接像 /etc/fstab，释放网络和系统资源中的条目。
4.自在客户端动安装程序配置完全；不需要服务器端配置。
5.挂载使用 mount 命令，其中包括安全选项所使用的相同装载选项。
6.对这两个直接和间接的挂载点映射，在挂载点的位置提供弹性的支持。
7.间接装入点创建和删除的 autofs，缓解了需要手动对其进行管理。
8.NFS 是默认的文件系统的挂载，但它可以用来自动挂载一个范围的不同的文件系统。
9.autofs 是一种服务，管理的像其他系统服务。

1、创建自动挂载

Configuring an automount is a multistep process:

1. Install the **autofs** package.

```
[student@desktopX ~]$ sudo yum -y install autofs
```

   This package contains everything needed to use the automounter for NFS shares.

2. Add a *master-map* file to **/etc/auto.master.d**—this file identifies the base directory used for mount points and identifies the mapping file used for creating the automounts.

   Use **vim** to create and edit the master-map file:

```
[student@desktopX ~]$ sudo vim /etc/auto.master.d/demo.autofs
```

The name of the master-map file is not important, but it is normally something meaningful. The only requirement is it must have an extension of **.autofs**. The master-map file can hold multiple mapping entries, or use multiple files to separate configuration data.

Add the master-map entry, in this case, for indirectly mapped mounts:

```
/shares  /etc/auto.demo
```

This entry would use the **/shares** directory as the base of future indirect automounts. The **/etc/auto.demo** file contains the mount details; use an absolute filename. The**auto.demo** file needs to be created before starting the **autofs** service.

To use directly mapped mount points, add an entry to the same file (or in a separate file):

```
/-  /etc/auto.direct
```

All direct map entries use "**/-**" as the base directory. In this case, the mapping file that contains the mount details is **/etc/auto.direct**.

李伟阶 QQ：17533203

3. Create the mapping file(s). The mapping file identifies the mount point, mount options, and source location to mount.

   Use **vim** to create and edit the mapping file:

```
[student@desktopX ~]$ sudo vim /etc/auto.demo
```

   The file name is not important, but by convention is located in **/etc** and called **auto.*name***, where ***name*** is something meaningful to the included contents.

```
work  -rw,sync  serverX:/shares/work
```

   The format of an entry is *mount point*, *mount options*, and *source location*. This example is showing a basic indirect mapping entry. Direct maps and indirect maps using wildcards will be covered later in this section.

4. Start and enable the automount service.

   Use **systemctl** to both start and enable the **autofs** service.

```
[student@desktopX ~]$ sudo systemctl enable autofs
ln -s '/usr/lib/systemd/system/autofs.service'  ...
[student@desktopX ~]$ sudo systemctl start autofs
```

2、配置自动挂载格式分析

The mapping file—direct maps

As the name implies, direct maps are used to map an NFS share to an existing mount point. The automounter will not attempt to create the mount point automatically; it must exist prior to the **autofs** service being started.

Continuing from the previous example, the content for the **/etc/auto.direct** file might look like this:

```
/mnt/docs  -rw,sync  serverX:/shares/docs
```

The mount point (or key) is always an absolute path, starting with "/" (slash). The rest of the mapping file uses the same structure.

Only the right-most directory is put under automounter control. Thus, the directory structure above the mount point (**/mnt** in this example) is not obscured by **autofs**.

李伟阶 QQ：17533203

The mapping file—indirect wildcard maps

When an NFS server is exporting multiple subdirectories within a directory, then the automounter can be configured to access any one of those subdirectories using a single mapping entry. As an example, this can be really useful for automounting user *home*directories from an NFS server.

Continuing the previous example, if **serverX:/shares** is exporting two or more subdirectories and they are able to be accessed using the same mount options, then the content for the **/etc/auto.demo** file might look like this:

```
*  -rw,sync  serverX:/shares/&
```

The mount point (or key) is an "*****" (asterisk), and the subdirectory on the source location is an "**&**" (ampersand). Everything else in the entry is the same.

The mapping file—indirect wildcard maps

When an NFS server is exporting multiple subdirectories within a directory, then the automounter can be configured to access any one of those subdirectories using a single mapping entry. As an example, this can be really useful for automounting user *home*directories from an NFS server.

Continuing the previous example, if **serverX:/shares** is exporting two or more subdirectories and they are able to be accessed using the same mount options, then the content for the **/etc/auto.demo** file might look like this:

```
*  -rw,sync  serverX:/shares/&
```

The mount point (or key) is an "*****" (asterisk), and the subdirectory on the source location is an "**&**" (ampersand). Everything else in the entry is the same.

第十二章　挂载 SMB 文件系统

1.1 访问 SMB 网络存储

Red Hat 桌面机和服务器,通过提供使用 SMB 协议的共享,可以连接到任何服务器。访问 SMB 共享的三个基本步骤确定要访问的远程共享。

1.确定一个挂载点

2.安装和创建挂载点空的目录共享

3.使用相应的挂载命令或配置更改的网络文件系统

在开始之前，必须按顺序装入 SMB 共享安装： cifs-utils。mount 命令和 autofs 挂载依靠

此包为 CIFS 文件系统挂载。同时还需要 samba-client，通过 samba-client 命令可以连接或访问远端的 samba 服务器。

多组织需要提供网络存储和打印服务的范围的桌面操作系统。红帽企业 Linux 使用 Samba 服务器提供服务，微软 Windows 的客户端可以使用。Samba 实现了服务器消息块 (SMB) 协议，并共同互联网文件系统 (CIFS) 是一种方言的 SMB。这两个名字都常常可以互换。

1.2 配置手动与开机挂载 samba 共享

1、挂载 samba 共享

Mount SMB Share

1. **Identify:** The administrator for the SMB server host can provide share details, such as *username* and *password*, *share* names, etc. An alternative is to use a client that can browse the shares, such as **smbclient**.

```
[student@desktopX ~]$ smbclient -L //serverX
```

The **-L** option asks the **smbclient** to list the shares available on serverX.

2. **Mount point:** Use **mkdir** to create a mount point in a suitable location.

```
[student@desktopX ~]$ mkdir -p /mountpoint
```

3. **Mount:** There are two choices here: manually or incorporated in the **/etc/fstab** file. Switch to *root* or use **sudo** for either operation.
   - *Manual:* Use the **mount** command.

```
[student@desktopX ~]$ sudo mount -t cifs -o guest //serverX/share /mountpoint
```

The **-t cifs** option is the file system type for SMB shares and the **-o guest** option tells **mount** to try and authenticate as a *guest* account without a password.

   - */etc/fstab:* Use **vim** to edit the **/etc/fstab** file and add the mount entry to the bottom of the file. The SMB share will be mounted at each system boot.

```
[student@desktopX ~]$ sudo vim /etc/fstab
...
//serverX/share  /mountpoint  cifs  guest  0 0
```

Use **umount**, using *root* privileges, to manually unmount the share.

```
[student@desktopX ~]$ sudo umount /mountpoint
```

2、基于身份验证的 samba 的挂载

```
[student@desktopX ~]$ sudo mount -t cifs -o guest //serverX/docs /public/docs
```

Mount the SMB share **//serverX/docs** at **/public/docs** and attempt to authenticate as *guest*.

```
[student@desktopX ~]$ sudo mount -t cifs -o username=watson //serverX/cases
/bakerst/cases
```

Mount the SMB share **//serverX/cases** at **/bakerst/cases** and attempt to authenticate as *watson*. The **mount** command will prompt for the password in this example.

The **credentials** file offers better security because the password is stored in a more secure file, whereas the **/etc/fstab** file is easily examined.

```
[student@desktopX ~]$ sudo mount -t cifs -o credentials=/secure/sherlock
//serverX/sherlock /home/sherlock/work
```

Mount the SMB share **//serverX/sherlock** at **/home/sherlock/work** and attempt to authenticate using the credentials stored in **/secure/sherlock**.

The format for the **credentials** file is:

```
username=username
password=password
domain=domain
```

3、自动挂载 samba 共享

The following creates an automount at **/bakerst/cases** for SMB share **//serverX/cases**, and authenticates against the credentials file **/secure/sherlock**.

- **/etc/auto.master.d/bakerst.autofs** content:

```
/bakerst   /etc/auto.bakerst
```

- **/etc/auto.bakerst** content:

```
cases  -fstype=cifs,credentials=/secure/sherlock  ://serverX/cases
```

- **/secure/sherlock** content (owned by **root**, perms **600**):

```
username=sherlock
password=violin221B
domain=BAKERST
```

- **autofs** enable and start:

```
[student@desktopX ~]$ sudo systemctl enable autofs
[student@desktopX ~]$ sudo systemctl start autofs
```

文件系统类型需要指定与-fstype = cifs 选项，然后以逗号分隔的装载选项，使用 mount 命令的同一装载选项列表。服务器 URI 地址需要前缀为冒号"："。

第十三章 RHEL 引导过程控制和故障排除

13.1 RHEL 7 引导过程

1、 RHEL 7 引导过程

1) 系统固件 （现代 UEFI 或更老式 BIOS） 运行开机自检 (POST)，并开始初始化的一些硬件。
2) 系统固件搜索一个可引导的设备，配置在 UEFI 引导固件或通过搜索在所有磁盘上的主启动记录 (MBR) 。
3) 系统固件从磁盘上读取一个引导加载程序，然后将系统的控制传递到引导加载程序。（RHEL 7 系统上，为 grub2）
   使用配置：grub2-install
4) 从磁盘引导加载程序，加载其配置并向用户呈现配置启动菜单。
   使用配置文件： /etc/grub.d/，/etc/default/grub，和 boot/grub2/grub.cfg。
5) 用户做出了一个选择 （或自动超时发生后），引导加载程序从磁盘加载已配置好的内核和 initramfs 到内存中。Initramfs （ram 文件系统）是 gzip 格式的文件，通过 cpio 将 gzip 包含所有启动必需的内核模块、硬件驱动，boot、init 脚本（initramfs 本身包含一个可用的系统）
   使用配置文件： /etc/dracut.conf
6) 引导加载程序传递在内核命令行上指定的参数。并指定 initramfs 在内存中的位置
   使用配置： /etc/grub.d/，/etc/default/grub，和 （不是手动的） /boot/grub2/grub.cfg。
7) 内核初始化所有在 initramfs 中硬件驱动程序，然后从 initramfs 执行/sbin/init PID 1 。RHEL 7 initramfs 包含 systemd
   使用配置： init = 命令行参数。
8) 从 initramfs 实例执行的 initrd.target 目标的所有单位。这包括在 /sysroot 上安装的实际根文件系统。
   使用配置： /etc/fstab
9) 内核的根文件系统从 initramfs 根文件系统切换到系统的根文件系统，以前安装在 /sysroot。然后 systemd 重新执行本身使用的 systemd 系统中安装的副本。
10) systemd 看起来为一个默认目标，或者通过在从内核命令行或配置在系统上，然后启动 （和停止） 相关单元目标，自动解决配置单元之间依赖关系。这些目标通常包括至少一个基于文本的登录或一个图形化的登录屏幕。
    使用配置： /etc/systemd/system/default.target、/etc/systemd/system/*

2、 启动，重启与关机

若要从命令行关机或重新启动运行的系统，管理员可以使用 systemctl 命令。

systemctl poweroff 将会停止正在运行的所有服务、 卸载所有文件系统 （或他们不能被卸载时重新以只读方法挂载），然后关闭系统。

systemctl reboot 将会停止正在运行的所有服务、 卸载所有的文件系统，然后重新启动系统。

为方便的向后兼容性，关机 power off 和重启 reboot 命令仍然存在，但在 RHEL7 中他们使用软链接到 systemctl 执行。（init0，6 仍支持）

重要：

Systemclt halt 和 halt 也可关闭系系，但不同 poweroff ，这个命令不关闭系统，只会使系统处在一点可以手动安全关机。

3、 选择 systemd target

1）可用 tartet

sytemd target 设置哪些目标单元会启动来达到一个想得到的状态。

A **systemd** target is a set of **systemd** units that should be started to reach a desired state. The most important of these targets are listed in the following table.

Target	Purpose
graphical.target	System supports multiple users, graphical and text-based logins.
multi-user.target	System supports multiple users, text-based logins only.
rescue.target	**sulogin** prompt, basic system initialization completed.
emergency.target	**sulogin** prompt, **initramfs** pivot complete and system root mounted on **/** read-only.

It is possible for a target to be a part of another target; for example, the **graphical.target** includes **multi-user.target**, which in turn depends on **basic.target** and others. These dependencies can be viewed from the command line with the following command:

```
[root@serverX ~]# systemctl list-dependencies graphical.target | grep target
```

An overview of all available targets can be viewed with:

```
[root@serverX ~]# systemctl list-units --type=target --all
```

An overview of all targets installed on disk can be viewed with:

```
[root@serverX ~]# systemctl list-unit-files --type=target --all
```

2）运行系统中切换 target。

在系统运行时间，系统管理员能用 systemctl isolate 命令切换到不同 target。停止所有不需要的服务启动需要的服务。

Selecting a target at runtime

On a running system, administrators can choose to switch to a different target using the **systemctl isolate** command; for example:

```
[root@serverX ~]# systemctl isolate multi-user.target
```

Isolating a target will stop all services not required by that target (and its dependencies), and start any required services that have not yet been started.

3）设置默认 target。

当系统启动后，控制权从 initramfs 传递给 systemd。Systemd 激活默认 target。通常默认 target 用符号链接(在/etc/systemd/system)到 graphical.target 或 multi-user.target。

Systemctl get-default 和 systemctl set-default 是管理默认 target 的两个命令。

Setting a default target

When the system starts, and control is passed over to **systemd** from the **initramfs**, **systemd** will try to activate the **default.target** target. Normally the **default.target** target will be a symbolic link (in **/etc/systemd/system/**) to either **graphical.target** or **multi-user.target**.

Instead of editing this symbolic link by hand, the **systemctl** tool comes with two commands to manage this link: **get-default** and **set-default**.

```
[root@serverX ~]# systemctl get-default
multi-user.target
[root@serverX ~]# systemctl set-default graphical.target
rm '/etc/systemd/system/default.target'
ln -s '/usr/lib/systemd/system/graphical.target' '/etc/systemd/system/default.target'
[root@serverX ~]# systemctl get-default
graphical.target
```

4）在启动过程中选择不同 target。

1、重新启动系统
2、按任何键打断引导加载程序菜单开始倒计时
3、将光标移动到要启动的项
4、按 e 键编辑当前条目
5、将光标移动到该以 linux16 开头的行，也就是内核参数行
6、追加 systemd.unit=desired.target （内核传导）
7、按 Ctrl + x 来启动这些更改

13.2 修复一般的启动问题

1、 恢复 root 口令

在 RHEL6 及以前版本恢复 root 口令，管理员可以启到系统运行级别 1 的单用户模式，得到 root 提示符可以修口令。

RHEL7 和级别 1 类似的是 rescue.target 和 emergency.target，两者都需要 root 口令登录。

RHEL 7 是在从 initramfs 点暂停运行，提供根 shell。虽然这主要用于调试，它也可以用来恢复丢失的根密码：

1、重新启动系统。
2、按任意键中断的引导加载程序倒计时。
3、将光标移动到需要引导的条目。
4、按 e 键编辑选定的条目。
5、将光标移动到内核的命令行 （以 linux16 开头的行）。
6、追加的 rd.break （这会打破之前控制从 initramfs 交给实际的系统）。
7、Ctrl+x 启动

To recover the **root** password from this point, use the following procedure:

1. Remount **/sysroot** as read-write.

```
switch_root:/# mount -oremount,rw /sysroot
```

2. Switch into a **chroot** jail, where **/sysroot** is treated as the root of the file system tree.

```
switch_root:/# chroot /sysroot
```

3. Set a new root password:

```
sh-4.2# passwd root
```

4. Make sure that all unlabeled files (including **/etc/shadow** at this point) get relabeled during boot.

```
sh-4.2# touch /.autorelabel
```

5. Type **exit** twice. The first will exit the **chroot** jail, and the second will exit the **initramfs** debug shell.

At this point, the system will continue booting, perform a full SELinux relabel, then reboot again.

2、 使用 journalctl

如果 journal log 配置永久保存，可以使用 journalctl 工具来查看启动失败的日志。

1）确保 journal log 永久保存

2）用 journalctl -b 选项查看启动相关的日志

First make sure that you have persistent **journald** logging enabled:

```
[root@serverX ~]# mkdir -p -m2775 /var/log/journal
[root@serverX ~]# chown :systemd-journal /var/log/journal
[root@serverX ~]# killall -USR1 systemd-journald
```

To inspect the log files for a previous boot, use the **-b** option to **journalctl**. Without any arguments, the **-b** option will filter output only to messages pertaining to this boot, but with a negative number as an argument, it will filter on previous boots. For example:

```
[root@serverX ~]# journalctl -b-1 -p err
```

This command will show all messages rated as an error or worse from the previous boot.

3、 诊断和修复 systemd 启动问题

如果在启动服务时有问题，有几个有用工具能帮系统管理员调试和排错。

1） debug shell

通过运行 systemctl enable debug-shell.service，能在早期启动中产生一个在 TTY9（Ctrl+Alt+F9）上的 root shell。这个 shell 能自动以 root 登录，管理员在系统正在启动时用来运行其他调试工具。

警告：

不要忘记当调试完后禁用 debug-shell.service，防止一个未授权的用户获得 root shell。

2）rescue 和 emergency targets

在启动加载程序内核命令行后添加

systemd.unit=rescue.target 或

sysytemd.unit=emergency.target

系统将启动救援或紧急模式 shell,这两个 shell 都需要 root 口令。紧急模式以只读方式挂载根文件系统。

这两个 shell 可以用来修复防碍系统正常启动的问题，比如不正确/etc/fstab。

3）Stuck jobs

在启动过程中，systemd 产生一系列作业，如果有一个作业不能完成，它将阻止其他作业运行。管理员能使用命令 systemctl list-jobs 检查当前的作业列表，必须 running 的作业完成，waiting 的作业才能继续。

13.3 修复引导加载程序问题

引导加载程序使用默认情况下，GRUB2 是 RHEL7 统一的引导加载器。可以使用 grub2 启动在 BIOS 和 UEFI 系统，以及支持引导几乎任何在现代的硬件运行的操作系统上。

Grub2 的主要配置文件是 /boot/grub2/grub.cfg，但管理员不应该直接编辑此文件。相反，一种称为 grub2 mkconfig 工具用于生成该配置使用一组不同的配置文件和列表中已安装的内核。

grub2-mkconfig 将 /etc/default/grub 的默认菜单超时和内核命令行选项来使用，然后在 /etc/grub.d/ 中使用一组脚本生成的配置文件。

要永久更改引导加载程序配置，管理员需要编辑以前，列出的配置文件，然后运行以下命令：

```
[root@serverX ~]# grub2-mkconfig > /boot/grub2/grub.cfg
```

第十四章 firewalld 限制网络通信

14.1 限制网络通信

1、 Netfilter 和 firewalld 的概念

Linux 内核包括功能强大网络过滤子系统，netfilter。Netfilter 子系统允许内核模块检查每个数据包。这意味着可以检查、 修改、 删除，或通过编程方式拒绝，在到达用户空间中的组件之前拒绝了任何传入、 传出，或转发的网络数据包。

1） 与 Netfilter 交互

虽然从理论上系统管理员可能要编写自己的内核模块与 netfilter 进行交互，但这通常不需要。相反，用于其他的程序与 netfilter 进行交互。其中一个最常见、 最知名的这些程序是 iptables。在以前的红帽企业 Linux 版本，iptables 是与内核 netfilter 子系统进行交互的主要方法。

Iptables 命令是一个低级的工具，和它可能很难正确地管理防火墙。此外，它只能调整

IPv4 防火墙规则。其他实用程序，可用于更完整的防火墙覆盖率， IPv6 的 ip6tables 和 ebtables 的桥梁软件。

2）介绍 firewalld

RHEL7 中介绍了一种新方法与 netfilter 交互的： firewalld。firewalld 是一个系统守护进程，可以配置和监视系统的防火墙规则。应用程序可以根据请求的端口，打开要使用传递系统消息通道，一个功能，可以禁用或锁定 DBus firewalld。它既包括 IPv4、 IPv6，和潜在 ebtables 设置。Firewalld 守护进程是从 firewalld 包安装的。

firewalld 通过区域分类所有的网络流量，从而简化了防火墙管理。基于传入的网络接口或数据包的源 IP 地址的标准，是然后转入适当的区域的防火墙规则。每个区域都可以具有自己的端口和服务，以打开或关闭的列表表示。

注意：

笔记本或其他电脑经常改变网络，NetworkManager 能自动为网络连接设置防火墙区域，为特定的连接的这个区域能定义合造的规则。

这个当你在家，公司和公共无线网之间移动时非常有用。

每个包进入系统首先检查源地址，如果源地址与某个特定区域关联则应用这个区域的规则。没果源地址没有与某个区域关联则应用入站网络接口区域。

如果网络接口没有关联到某个区域，则应用默认区域，public 区域用做默认的区域，管理员可以修改。

3）预定义的区域

Firewalld 定义许多预定义区域，默认区域是 public，接口分配到 public。下表列出安装的预定义区域，系统管理员也可以自定义区域。默认情况下，所有区域允许系统初始化一部分入站流量和所有出站流量。

Default configuration of `firewalld` zones

Zone name	Default configuration
trusted	Allow all incoming traffic.
home	Reject incoming traffic unless related to outgoing traffic or matching the **ssh**, **mdns**, **ipp-client**, **samba-client**, or **dhcpv6-client** pre-defined services.
internal	Reject incoming traffic unless related to outgoing traffic or matching the **ssh**, **mdns**, **ipp-client**, **samba-client**, or **dhcpv6-client** pre-defined services (same as the **home** zone to start with).
work	Reject incoming traffic unless related to outgoing traffic or matching the **ssh**, **ipp-client**, or **dhcpv6-client** pre-defined services.
public	Reject incoming traffic unless related to outgoing traffic or matching the **ssh** or **dhcpv6-client** pre-defined services. *The default zone for newly-added network interfaces.*
external	Reject incoming traffic unless related to outgoing traffic or matching the **ssh** pre-defined service. Outgoing IPv4 traffic forwarded through this zone is *masqueraded* to look like it originated from the IPv4 address of the outgoing network interface.
dmz	Reject incoming traffic unless related to outgoing traffic or matching the **ssh** pre-defined service.
block	Reject all incoming traffic unless related to outgoing traffic.
drop	Drop all incoming traffic unless related to outgoing traffic (do not even respond with ICMP errors).

3）预定义的服务

Selected pre-defined `firewalld` services

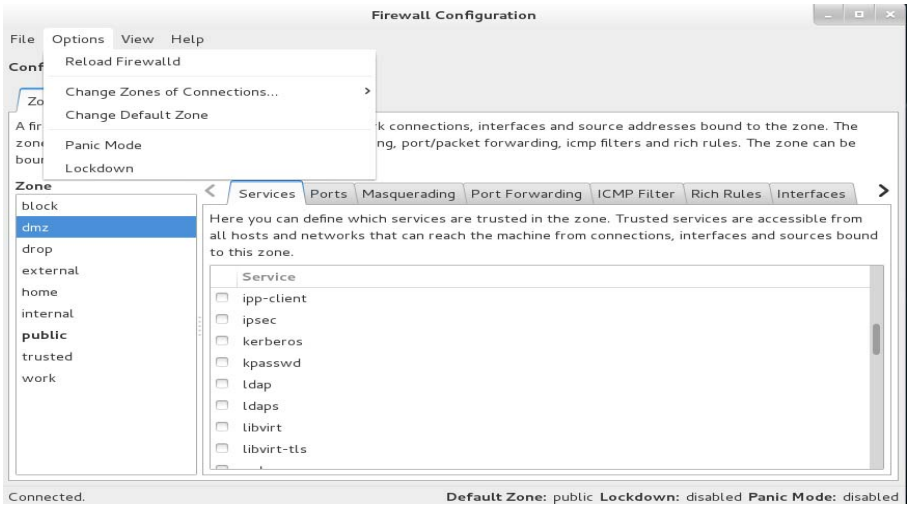
Service name	Configuration
ssh	Local SSH server. Traffic to 22/tcp
dhcpv6-client	Local DHCPv6 client. Traffic to 546/udp on the fe80::/64 IPv6 network
ipp-client	Local IPP printing. Traffic to 631/udp.
samba-client	Local Windows file and print sharing client. Traffic to 137/udp and 138/udp.
mdns	Multicast DNS (mDNS) local-link name resolution. Traffic to 5353/udp to the 224.0.0.251 (IPv4) or ff02::fb (IPv6) multicast addresses.

2、配置 Firewall

有三种主要方式为系统管理员与 firewalld 进行交互：

- 通过直接编辑配置文件/etc/firewalld/（不在这一章中讨论）
- 通过使用图形化的防火墙配置工具
- 通过使用防火墙 cmd 命令行

1）图形化防火墙配置工具 firewall-config



2）配置防火墙工具 firewall-cmd

李伟阶 QQ：17533203

firewall-cmd commands	Explanation
`--get-default-zone`	Query the current default zone.
`--set-default-zone=<ZONE>`	Set the default zone. This changes both the runtime and the permanent configuration.
`--get-zones`	List all available zones.
`--get-active-zones`	List all zones currently in use (have an interface or source tied to them), along with their interface and source information.
`--add-source=<CIDR> [--zone=<ZONE>]`	Route all traffic coming from the IP address or network/netmask **<CIDR>** to the specified zone. If no **--zone=** option is provided, the default zone will be used.
`--remove-source=<CIDR> [--zone=<ZONE>]`	Remove the rule routing all traffic coming from the IP address or network/netmask **<CIDR>** from the specified zone. If no **--zone=** option is provided, the default zone will be used.
`--add-interface=<INTERFACE> [--zone=<ZONE>]`	Route all traffic coming from **<INTERFACE>** to the specified zone. If no **--zone=** option is provided, the default zone will be used.
`--change-interface=<INTERFACE> [--zone=<ZONE>]`	Associate the interface with **<ZONE>** instead of its current zone. If no **--zone=** option is provided, the default zone will be used.
`--list-all [--zone=<ZONE>]`	List all configured interfaces, sources, services, and ports for **<ZONE>**. If no **--zone=** option is provided, the default zone will be used.
`--list-all-zones`	Retrieve all information for all zones. (Interfaces, sources, ports, services, etc.)
`--add-service=<SERVICE> [--zone=<ZONE>]`	Allow traffic to **<SERVICE>**. If no **--zone=** option is provided, the default zone will be used.
`--add-port=<PORT/PROTOCOL> [--zone=<ZONE>]`	Allow traffic to the **<PORT/PROTOCOL>** port(s). If no **--zone=** option is provided, the default zone will be used.
`--remove-service=<SERVICE> [--zone=<ZONE>]`	Remove **<SERVICE>** from the allowed list for the zone. If no **--zone=** option is provided, the default zone will be used.
`--remove-port=<PORT/PROTOCOL> [--zone=<ZONE>]`	Remove the **<PORT/PROTOCOL>** port(s) from the allowed list for the zone. If no **--zone=** option is provided, the default zone will be used.
`--reload`	Drop the runtime configuration and apply the persistent configuration.

The following examples show the default zone being set to **dmz**, all traffic coming from the **192.168.0.0/24** network being assigned to the **internal** zone, and the network ports for **mysql** being opened on the **internal** zone.

```
[root@serverX ~]# firewall-cmd --set-default-zone=dmz
[root@serverX ~]# firewall-cmd --permanent --zone=internal --add-source=192.168.0.0/24
[root@serverX ~]# firewall-cmd --permanent --zone=internal --add-service=mysql
[root@serverX ~]# firewall-cmd --reload
```

李 伟 阶
QQ：17533203
整理于 2014 年 9 月