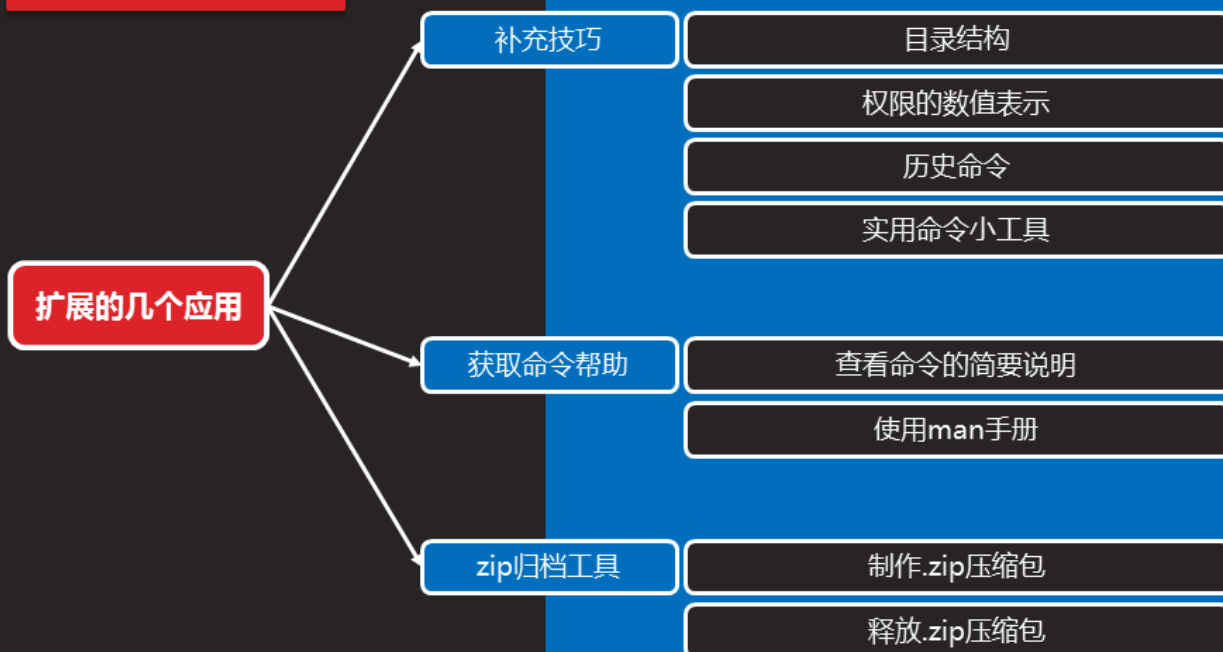


内容

上午	09:00 ~ 09:30	扩展的几个应用
	09:30 ~ 10:20	
	10:30 ~ 11:20	
	11:30 ~ 12:00	发布网络YUM源
下午	14:00 ~ 14:50	vim编辑技巧
	15:00 ~ 15:50	源码编译安装
	16:10 ~ 17:00	systemctl控制
	17:10 ~ 18:00	总结和答疑



扩展的几个应用



补充应用

目录结构

- 认识Linux的目录层次：
 - [man hier](#)

知识讲解

DESCRIPTION

A typical Linux system has, among others, the following directories:

/ This is the root directory. This is where the whole tree starts.

/bin This directory contains executable programs which are needed in single user mode and to bring the system up or repair it.

/boot Contains static files for the boot loader. This directory only holds the files which are needed during the boot process. The map installer and configuration files should go to /sbin and /etc.

/dev Special or device files, which refer to physi-
:



目录结构（续1）

- 常见一级目录的用途

知识讲解

目录路径	主要用途
/boot	存放系统引导必需的文件，包括内核、启动配置
/bin、/sbin	存放各种命令程序
/dev	存放硬盘、键盘、鼠标、光驱等各种设备文件
/etc	存放Linux系统及各种程序的配置文件
/root、/home/用户名	分别是管理员root、普通用户的默认家目录
/var	存放日志文件、邮箱目录等经常变化的文件
/proc	存放内存中的映射数据，不占用磁盘
/tmp	存放系统运行过程中使用的一些临时文件



权限的数值表示

- 权限的数值化
 - 基本权限： $r = 4$ ， $w = 2$ ， $x = 1$
 - 附加权限： $SUID = 4$ ， $SGID = 2$ ，Sticky Bit = 1
- 采用数值形式设置权限
 - `chmod [-R] nnn 文档...`
 - `chmod [-R] xnnn 文档...`

知识讲解

分组	User权限			Group权限			Other权限		
字符	r	w	x	r	-	x	r	-	x
数字	4	2	1	4	0	1	4	0	1
求和	7			5			5		



历史命令

知识讲解

- 管理/调用曾经执行过的命令
 - history : 查看历史命令列表
 - history -c : 清空历史命令
 - !n : 执行命令历史中的第n条命令
 - !str : 执行最近一次以str开头的历史命令
- 调整历史命令的数量

```
[root@svr7 ~]# vim /etc/profile  
HISTSIZE=1000
```

//默认记录1000条



实用小命令工具

知识讲解

- du , 统计文件的占用空间
 - du [选项]... [目录或文件]...
 - -s : 只统计每个参数所占用的总空间大小
 - -h : 提供易读容量单位 (K、M等)

```
[root@svr7 ~]# du -sh /boot/ /etc/pki/  
130M  /boot/  
1.5M  /etc/pki/
```



实用小命令工具（续1）

知识讲解

- date , 查看/调整系统日期时间
 - date +%F、date +%R
 - date +"%Y-%m-%d %H:%M:%S"
 - date -s "yyyy-mm-dd HH:MM:SS"

```
[root@svr7 ~]# date -s "2001-12-31 09:30"  
2001年 12月 31日 星期一 09:30:00 CST
```

```
[root@svr7 ~]# date +%F %R  
2001-12-31 09:30
```

```
[root@svr7 ~]# hwclock -s //重设系统日期时间
```



案例1：补充应用技巧

系统管理小技巧操作

课堂练习

- 1) 采用数值形式将目录/root的权限调整为 rwx-----
- 2) 将记录的历史命令条数更改为 200 条
- 3) 统计 /boot、/etc/pki 目录占用的空间大小
- 4) 以格式 "yyyy-mm-dd HH:MM" 显示系统时间



实用小命令工具（续2）

知识讲解

- ln , 创建软连接
 - 软连接 --> 原始文档 --> i节点 --> 文档数据
 - ln -s 原始文件或目录 软连接文件

若原始文件或目录被删除，连接文件将失效
软连接可存放在不同分区/文件系统
- ln , 创建硬连接
 - 硬连接 --> i节点 --> 文档数据
 - ln 原始文件 硬连接文件

若原始文件被删除，连接文件仍可用
硬连接与原始文件必须在同一分区/文件系统



案例2：软连接与硬连接

新建文件 file1，内容为 AAAA，并完成下列操作

- 1) 为 file1 建立软连接 file1-s，对比两文件内容
- 2) 为 file1 建立硬连接 file1-h，对比两文件内容
- 3) 对比上述 3 个文件的 i 节点编号
- 4) 删除文件 file1，再查看文件 file1-s、file1-h 内容

课堂练习



获取命令帮助

查看命令的简要说明

- 对于内部命令，使用 “**help** 命令名” 查看

```
[root@svr7 ~]# help help
help: help [-dms] [pattern ...]
      Display information about builtin commands.
... ..
```

- 对于外部命令程序，通常会提供 “**--help**” 帮助选项

```
[root@svr7 ~]# ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
... ..
```

使用man手册

知识讲解

- 一种集中的手册页文档机制
 - 大多数程序/配置文件/库函数都提供man手册页
- **man** , 格式化手册阅读工具
 - 按 ↑ 、 ↓ 、 PgUp 、 PgDn 键滚动及翻页
 - 按 q 键退出
 - 按 / 键向后查找关键词 (n、N切换)

```
[root@svr7 ~]# man mkdir
```

```
.. ..
```

```
[root@svr7 ~]# man 5 passwd
```

```
.. ..
```



zip归档工具

制作.zip压缩包

- 归档+压缩操作

– zip [-r] 备份文件.zip 被归档的文档...

```
[root@svr7 ~]# zip -r /opt/etc_selinux.zip /etc/selinux/  
..  
[root@svr7 ~]# du -sh /opt/etc_selinux.zip /etc/selinux/  
9.8M    /etc/selinux/  
5.5M    /opt/etc_selinux.zip
```

知识讲解



释放.zip压缩包

- 释放归档+解压操作

– unzip 备份文件.zip [-d 目标文件夹]

```
[root@svr7 ~]# mv /etc/selinux /etc/selinux.old  
[root@svr7 ~]# unzip /opt/etc_selinux.zip -d /  
[root@svr7 ~]# ls -d /etc/selinux //确认恢复结果  
/etc/selinux
```

知识讲解



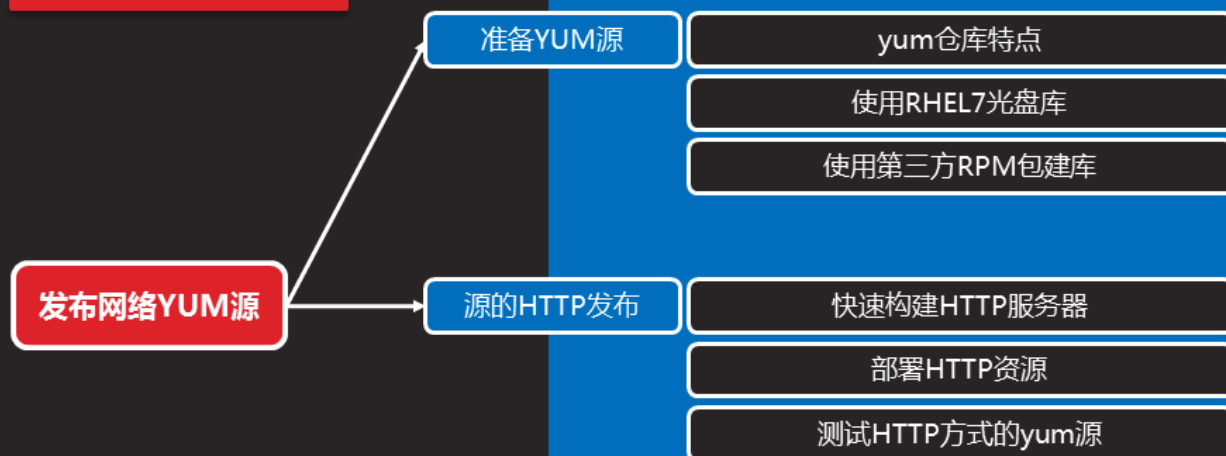
案例3：man手册、zip备份

1. 查阅passwd命令、/etc/passwd配置文件的手册页
2. 使用zip打包/usr/share/doc/qemu-kvm/目录

课堂练习



发布网络YUM源



准备YUM源

yum仓库特点

知识讲解

- 作为yum源需要准备的内容
 - 大量的 .rpm 软件安装包文件
 - 针对这些软件包的 **repodata/** 仓库档案
- repodata/ 仓库档案数据
 - filelists.xml.gz //软件包的文件安装清单
 - primary.xml.gz //软件包的基本/主要信息
 - other.xml.gz //软件包的其他信息
 - repomd.xml //提供.xml.gz下载和校验信息



使用RHEL7光盘库

知识讲解

- RHEL7光盘目录已经预先配置为yum源
 - Packages/ //存放 .rpm 软件包
 - repodata/ //仓库档案
 - isolinux/ //光盘引导数据
 - RPM-GPG-KEY-redhat-release //签名校验密钥

```
[root@room9pc13 ~]# vim /etc/fstab
```

```
.. ..
```

```
/ISO/rhel-server-7.2-x86_64-dvd.iso /var/www/html/rh7dvd  
iso9660 loop,ro 0 0 //可挂载ISO光盘镜像
```

```
[root@room9pc13 ~]# mount -a
```



使用第三方RPM包建库

知识讲解

- 需要额外创建 repodata/ 档案
 - 将搜集的 .rpm 包文件集中到指定目录
 - 针对此目录使用 createrepo 工具创建仓库档案

```
[root@room9pc13 ~]# ls /opt/libreoffice/ //原始目录  
install langpack_zh-CN readmes RPMS
```

```
[root@room9pc13 ~]# createrepo /opt/libreoffice/  
Spawning worker 0 with 53 pkgs
```

```
.. ..
```

```
[root@room9pc13 ~]# ls /opt/libreoffice/ //建立仓库档案  
install langpack_zh-CN readmes repodata RPMS  
//确认结果
```



案例4：自定义yum软件仓库

课堂练习

1. 准备 /var/www/html/rh7dvd/ 仓库目录
 - 1) 创建目录 /var/www/html/rh7dvd
 - 2) 挂载 rhel-server-7.2-x86_64-dvd.iso 到上述目录
2. 准备 /opt/libreoffice/ 仓库目录
 - 1) 下载 LibreOffice 办公软件的 rpm 集合版文件
 - 2) 将其中的内容释放到 /opt/libreoffice/ 目录
 - 3) 为 /opt/libreoffice/ 目录建立仓库档案



源的HTTP发布

快速构建HTTP服务器

知识讲解

- Hyper-Text Transfer Protocol , 超文本传输协议
 - 服务端 : httpd
 - 客户端 : elinks、firefox、

```
[root@room9pc13 ~]# yum -y install httpd
[root@room9pc13 ~]# systemctl restart httpd
[root@room9pc13 ~]# systemctl enable httpd
```

```
[root@room9pc13 ~]# netstat -anpt | grep httpd
tcp6      0      0 :::80      :::*        LISTEN    23672/httpd
```



部署HTTP资源

知识讲解

- 将yum仓库目录发布为Web下载资源
 - 建议部署为指定Web站点目录的子目录
 - 默认根目录位于 /var/www/html/

```
[root@room9pc13 ~]# du -sh /var/www/html/rh7dvd/
3.9G    /var/www/html/rh7dvd/           //确认RHEL7源数据
```

```
[root@room9pc13 ~]# mv /opt/libreoffice /var/www/html/
[root@room9pc13 ~]# du -sh /var/www/html/libreoffice/
234M    /var/www/html/libreoffice     //确认LibreOffice源数据
```



测试HTTP方式的yum源

知识讲解

- 调整客户机的 yum 仓库设置
 - baseurl = <http://服务器地址/目录位置>
- ```
[root@pc207 ~]# vim /etc/yum.repos.d/new.repo
[rh7dvd]
name = RHEL 7.2 Server
baseurl = http://192.168.4.254/rh7dvd
gpgcheck = 0

[libreoffice]
name = Libre office 5.1
baseurl = http://192.168.4.254/libreoffice
gpgcheck = 0

[root@pc207 ~]# yum repolist
.. ..
```



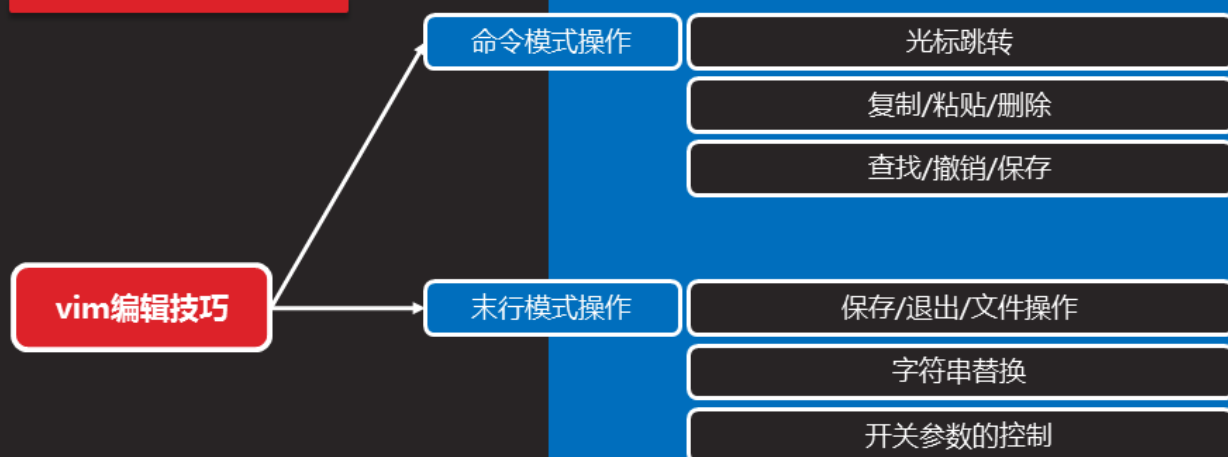
## 案例5：发布及测试yum仓库

课堂练习

1. 在 CentOS 真机上发布yum源
  - 1) 提供 rhel7 系统的光盘目录仓库
  - 2) 提供 LibreOffice 的rpm软件包仓库
2. 在 pc207 上使用上述yum源
  - 1) 创建仓库配置文件 /etc/yum.repos.d/new.repo
  - 2) 使用 CentOS 真机提供的 RHEL7、Libreoffice 源



## vim编辑技巧



## 命令模式操作



# 光标跳转

## 知识讲解

| 操作类型   | 按键指令            | 用途        |
|--------|-----------------|-----------|
| 移动光标   | ↑、↓、←、→         | 上、下、左、右   |
| 光标行内跳转 | Home 键 或 ^、数字 0 | 跳转到行首     |
|        | End 键 或 "\$" 键  | 跳转到行尾     |
| 全文翻页   | PgUp 键、PgDn 键   | 向上翻页、向下翻页 |
| 光标行间跳转 | 1G 或 gg         | 跳转到文件的首行  |
|        | G               | 跳转到文件的末尾行 |



# 复制/粘贴/删除

## 知识讲解

| 操作类型 | 按键指令         | 用途          |
|------|--------------|-------------|
| 复制   | yy、#yy       | 复制光标处的一行、#行 |
| 粘贴   | p、P          | 粘贴到光标处之后、之前 |
| 删除   | x 或 Delete 键 | 删除光标处的单个字符  |
|      | dd、#dd       | 删除光标处的一行、#行 |
|      | d^           | 从光标处之前删除至行首 |
|      | d\$          | 从光标处删除到行尾   |



## 查找/撤销/保存

知识讲解

| 操作类型 | 按键指令     | 用 途            |
|------|----------|----------------|
| 文本查找 | /word    | 向后查找字符串 “word” |
|      | n、N      | 跳至后/前一个结果      |
| 撤销编辑 | u        | 撤销最近的一次操作      |
|      | U        | 撤销对当前行的所有修改    |
|      | Ctrl + r | 取消前一次撤销操作      |
| 保存退出 | ZZ       | 保存修改并退出        |



## 末行模式操作

# 保存/退出/文件操作

## 知识讲解

| 操作类型  | 设置指令                | 用 途         |
|-------|---------------------|-------------|
| 存盘及退出 | :w                  | 保存当前文件      |
|       | :q!                 | 放弃已有更改后强制退出 |
|       | :wq 或 :x            | 保存已有修改后退出   |
| 文件操作  | :w /root/newfile    | 另存为其它文件     |
|       | :r /etc/filesystems | 读入其他文件内容    |



# 字符串替换

## 知识讲解

| 操作类型  | 设置指令             | 用 途              |
|-------|------------------|------------------|
| 行内替换  | :s/old/new       | 替换当前行第一个 "old"   |
|       | :s/old/new/g     | 替换当前行所有的 "old"   |
| 区域内替换 | :n,m s/old/new/g | 替换第n-m行所有的 "old" |
|       | :% s/old/new/g   | 替换文件内所有的 "old"   |



## 开关参数的控制

知识讲解

| 操作类型  | 设置指令         | 用途        |
|-------|--------------|-----------|
| 编辑器设置 | :set nu nonu | 显示/不显示行号  |
|       | :set ai noai | 启用/关闭自动缩进 |



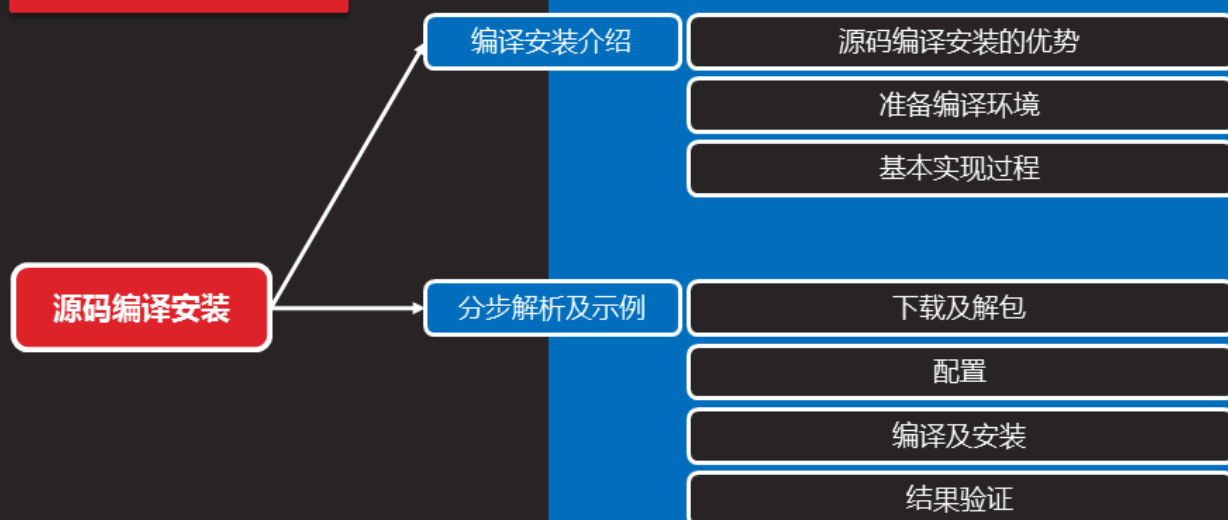
## 案例6：vim效率操作

课堂练习

- 将文件 /etc/passwd 复制为 /opt/nsd.txt
  - 打开 /opt/nsd.txt 文件
  - 练习命令模式下的光标切换/复制/删除/查找操作
- 将文件 /etc/man\_db.conf 复制到 /opt 目录下
  - 打开 /opt/man\_db.conf 文件
  - 将第50~100行内的 “man” 替换为 “MAN”
  - 在 vim 中设置显示行号



# 源码编译安装



## 编译安装介绍

# 源码编译安装的优势

知识讲解

- 主要优点
  - 获得软件的最新版，及时修复bug
  - 软件功能可按需选择/定制，有更多软件可供选择
  - 源码包适用各种平台
  - .....

<http://sourceforge.net/>  
<http://www.gnu.org/>、<http://www.apache.org/>  
.. ..



# 准备编译环境

知识讲解

- 开源软件多使用C/C++语言开发
  - 需要gcc、gcc-c++、make等编译工具

```
[root@svr7 ~]# yum -y install gcc gcc-c++ make
```

```
.. ..
```

```
[root@svr7 ~]# gcc --version
```

```
gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-4)
```

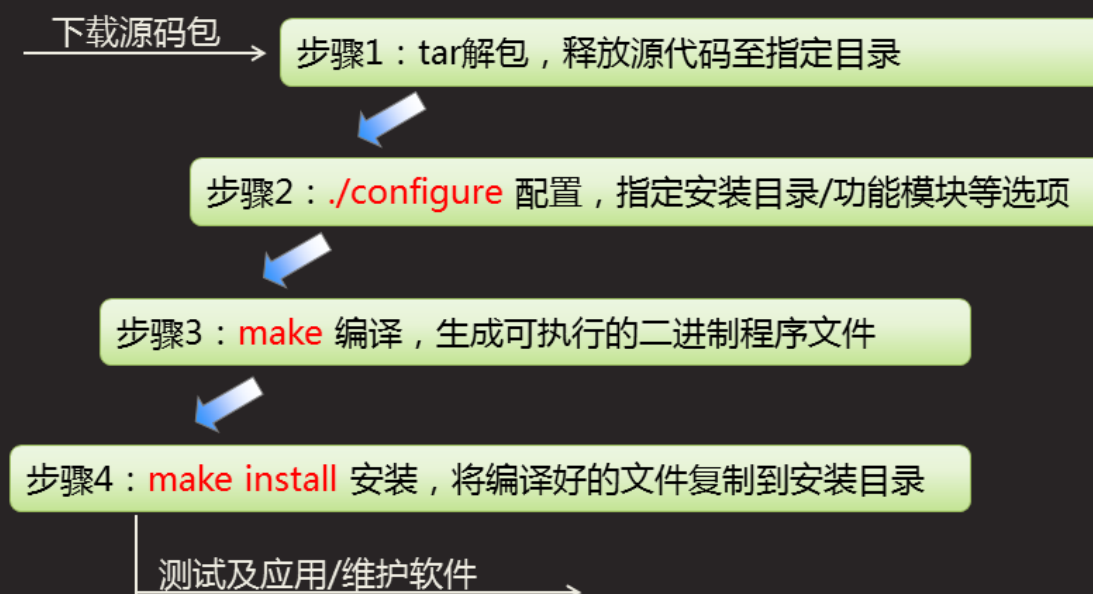
```
Copyright © 2015 Free Software Foundation, Inc.
```

本程序是自由软件；请参看源代码的版权声明。本软件没有任何担保；  
包括没有适销性和某一专用目的下的适用性担保。



# 基本实现过程

知识讲解



## 分步解析及示例

## 下载及解包

- 1. 使用tar解包，建议释放到 /usr/src/ 目录
  - 标准位置：/usr/src/软件名-版本号/

知识讲解

```
[root@svr7 ~]# tar xf inotify-tools-3.13.tar.gz -C /usr/src/
[root@svr7 ~]# ls /usr/src/inotify-tools-3.13/
aclocal.m4 config.sub INSTALL Makefile.in src
AUTHORS configure install-sh man
ChangeLog configure.ac libinotifytools missing
config.guess COPYING ltmain.sh NEWS
config.h.in depcomp Makefile.am README
```

软件包 inotify-tools 可提供文件/目录监控工具 inotifywait，  
当出现内容变更时，会给出提示  
下载地址：<https://sf.net/projects/inotify-tools/>



## 配置

- 2. 执行源码目录下的 configure 脚本
  - 通过 “./configure --help” 可以查看帮助
  - 典型配置参数：--prefix=安装目录

知识讲解

```
[root@svr7 ~]# cd /usr/src/inotify-tools-3.13/ //进目录
[root@svr7 inotify-tools-3.13]# ./configure //配置
...
[root@svr7 inotify-tools-3.13]# ls Makefile //确认结果
Makefile
```





## 编译及安装

- 3. make 编译
- 4. make install 安装

知识讲解

```
[root@svr5 httpd-2.2.25]# make
[root@svr5 httpd-2.2.25]# make install
```

} make && make install



## 结果验证

- 验证安装结果
  - 查看安装目录、了解程序的使用方法
  - 运行软件包提供的程序/服务

知识讲解

```
[root@svr7 ~]# ls /usr/local/*bin/inoti* //查找程序
/usr/local/bin/inotifywait /usr/local/bin/inotifywatch

[root@svr7 ~]# inotifywait --help //查看程序用法
.. ..
inotifywait 3.13
Wait for a particular event on a file or set of files.
Usage: inotifywait [options] file1 [file2] [file3] [...]
.. ..
```



## 案例7：编译安装软件包

### 编译安装 inotify-tools 软件工具

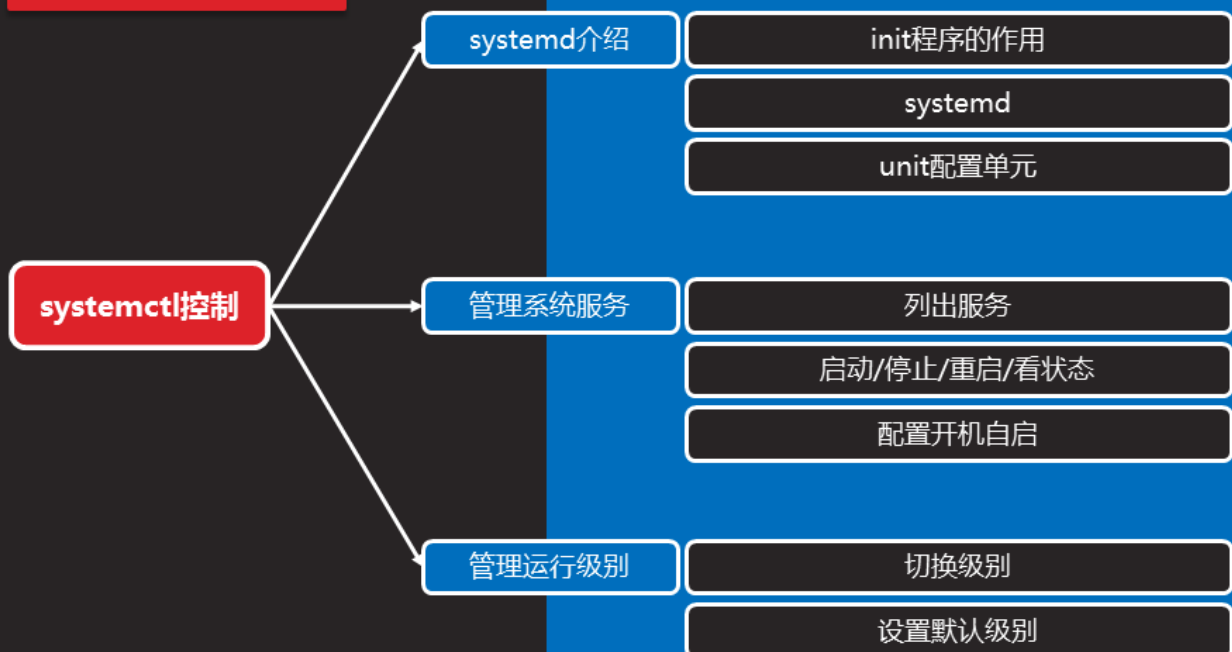
课堂练习

- 1) 释放 inotify-tools-3.13.tar.gz 源码包
- 2) 配置 ./configure
- 3) 编译 make、安装 make install
- 4) 测试inotifywait监控工具的用法及用途

```
[root@svr7 ~]# inotifywait -mrq /opt & //开启监控
[1] 15568
[root@svr7 ~]# touch /opt/a.txt //内容变更测试
..
[root@svr7 ~]# kill %1 //结束监控
```



### systemctl控制



# systemd介绍

## init程序的作用

- Linux系统和服务管理器
  - 是内核引导之后加载的第一个初始化进程（PID=1）
  - 负责掌控整个Linux的运行/服务资源组合
- 传统的 init 程序风格
  - system v：顺序加载，RHEL5系列采用
  - upstart：事件触发，RHEL6系列采用



# systemd

知识讲解

- 一个更高效的系统&服务管理器
  - 开机服务并行启动，各系统服务间的精确依赖
  - 配置目录：/etc/systemd/system/
  - 服务目录：/lib/systemd/system/
  - 主要管理工具：systemctl

```
[root@svr7 ~]# ls -l /sbin/init
lrwxrwxrwx. 1 root root 22 12月 7 09:34 /sbin/init ->
../lib/systemd/systemd
```



# unit配置单元

知识讲解

- 不同的unit决定了一组相关的启动任务
  - **service**：后台独立服务
  - **socket**：套接字，类似于xinetd管理的临时服务
  - **target**：一套配置单元的组合，类似于传统“运行级别”
  - **device**：对应udev规则标记的某个设备
  - **mount、automount**：挂载点、触发挂载点
  - .. ..



# 列出服务

## 知识讲解

- 列出活动的系统服务
  - `systemctl -t service`
- 列出所有系统服务（包括不活动的）
  - `systemctl -t service --all`

```
[root@svr7 ~]# systemctl -t service --all
UNIT LOAD ACTIVE SUB DESCRIPTION
abrt-ccpp.service loaded active exited Install ABRT
coredump hoo
...
abrt-vmcore.service loaded inactive dead Harvest vmcores
for ABRT
.. ..
```



# 管理运行级别

## 切换级别

知识讲解

- 列出可用运行级别
  - `systemctl -t target`
- 切换到文本/图形模式
  - `systemctl isolate multi-user.target`
  - `systemctl isolate graphical.target`

```
[root@svr7 ~]# systemctl isolate multi-user.target
```

```
.. ..
```

```
[root@svr7 ~]# runlevel
5 3
```

//确认结果

//旧级别 当前级别



## 设置默认级别

知识讲解

- 查看默认级别
  - `systemctl get-default`
- 设置默认级别
  - `systemctl set-default multi-user.target`
  - `systemctl set-default graphical.target`

```
[root@svr7 ~]# systemctl set-default multi-user.target
Removed symlink /etc/systemd/system/default.target.
Created symlink from /etc/systemd/system/default.target to
/usr/lib/systemd/system/multi-user.target.
```

```
[root@svr7 ~]# systemctl get-default
multi-user.target
```



## 案例8：使用systemctl工具

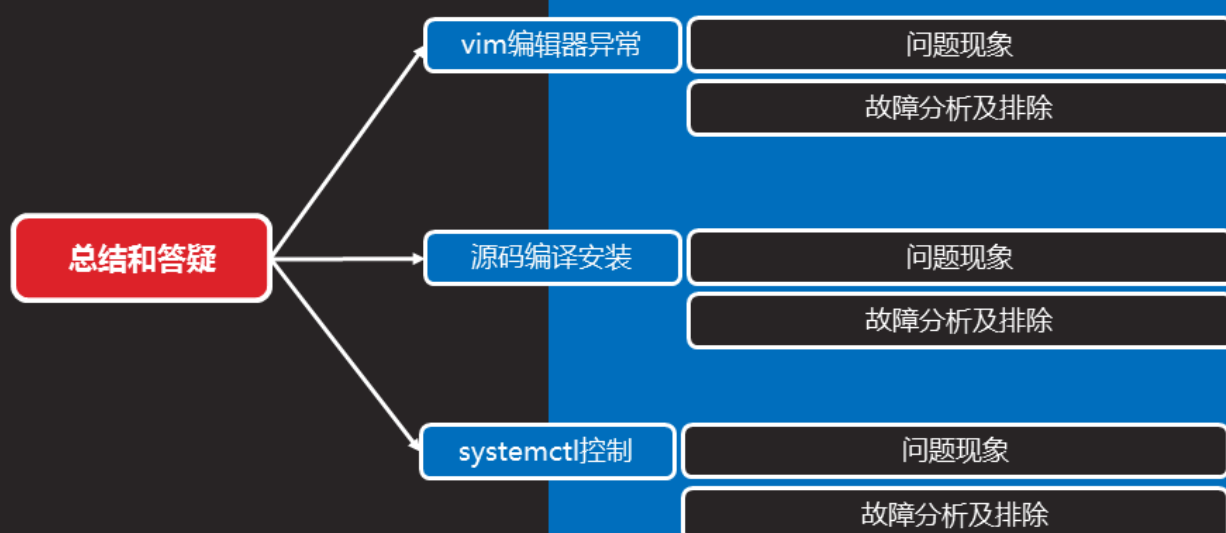
通过 systemctl 完成下列任务

- 1) 重启 httpd、crond、bluetooth 服务，查看状态
- 2) 禁止 bluetooth 服务开机自启，并停用此服务
- 3) 设置默认级别为 multi-user.target 并确认

课堂练习



### 总结和答疑



# vim编辑器异常

## 问题现象

- vim编辑一个新文件，执行 :wq 无法保存
  - 报错1：... 已设定选项 'readonly'
  - 报错2：... 无法打开并写入文件

知识讲解

defaults 0 0  
**E45: 已设定选项 'readonly' (请加 ! 强制执行)**

"/a/b/c.txt"  
**"/a/b/c.txt" E212: 无法打开并写入文件**  
请按 **ENTER** 或其它命令继续





# 故障分析及排除

知识讲解

- 原因分析
  - 报错1：当前用户对所修改的文件没有 w 权限
  - 报错2：所修改文件所在的目录不存在，或者当前用户对此目录没有 w 权限
- 解决办法
  - 报错1：确保用户对此文件有 w 写入权限
  - 报错2：确保此文件所在的目录已存在，而且当前用户对此目录有 w 权限



# 源码编译安装

## 问题现象

知识讲解

- 源代码编译软件包时，执行 `./configure` 步骤失败
  - 报错1：... 没有那个文件或目录
  - 报错2：... no acceptable C compiler found ...

```
[root@svr7 ~]# ./configure
```

```
-bash: ./configure: 没有那个文件或目录
```

```
[root@svr7 ~]# cd /usr/src/inotify-tools/
```

```
[root@svr7 inotify-tools-3.13]# ./configure
```

```
.. ..
```

```
configure: error: no acceptable C compiler found in $PATH
```



## 故障分析及排除

知识讲解

- 原因分析
  - 报错1：未进入到释放后的源码包目录
  - 报错2：未安装 gcc 编译环境
- 解决办法
  - 报错1：先 `cd` 切换到源码包目录，再执行配置操作
  - 报错2：需要安装好 gcc 等必要的软件包



# systemctl控制

## 问题现象

- 使用 systemctl 工具时，运行结果异常
  - 问题1：启动服务时报错 **No such file or directory.**
  - 问题2：将一个服务设置为自启时没有任何提示

知识讲解

```
[root@svr7 ~]# systemctl start http
Failed to start http.service: Unit http.service failed to load: No
such file or directory.
```

```
[root@svr7 ~]# systemctl enable httpd
```



# 故障分析及排除

## 知识讲解

- 原因分析
  - 报错1：没有这个服务，或服务名拼写错误
  - 报错2：此服务已经是开机自启了
- 解决办法
  - 报错1：指定正确的服务名，缺服务则需要装包
  - 报错2：若已经设置过，忽略即可

