

Top

NSD SERVICES DAY01

案例 1: 补充应用技巧

案例 2: 软连接与硬连接

案例 3: man 手册、zip 备份

案例 4: 自定义 yum 软件仓库

案例 5: 发布及测试 yum 仓库

案例 6: vim 效率操作

案例 7: 编译安装软件包

案例 8: 使用 systemctl 工具

1 案例 1: 补充应用技巧

1.1 问题

本例要求掌握在运维中比较常用的一些扩展命令技巧的使用，完成下列小技巧操作：

采用数值形式将目录/root 的权限调整为 `rwX-----`

将记录的历史命令条数更改为 200 条

统计 /boot、/etc/pki 目录占用的空间大小

以格式 “yyyy-mm-dd HH:MM” 显示系统时间

1.2 方案

date 日期时间工具：

显示日期时间：`date`、`date +%Y-%m-%d %H:%M:%S`

调整日期时间：`date -s "yyyy-mm-dd HH:MM:SS"`

恢复为硬件时间：`hwclock -s`

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：采用数值形式将目录/root 的权限调整为 `rwX-----`

1) 查看原来的权限

```
[root@svr7 ~]# ls -ld /root/
```

```
dr-xr-x---. 22 root root 4096 3月 26 14:59 /root/
```

2) 修改为新权限

```
[root@svr7 ~]# chmod 700 /root/
```

3) 确认权限设置结果

```
[root@svr7 ~]# ls -ld /root/
```

```
drwx-----. 22 root root 4096 3月 26 14:59 /root/
```

步骤二：将记录的历史命令条数更改为 200 条

1) 调整记录条数

修改配置文件/etc/profile，找到 HISTSIZE 行，将此变量的值修改为 200：

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

.. ..

HISTSIZE = 200

2) 确认设置结果

所有用户重新登录以后即可生效:

```
[root@svr7 ~]# su - root
[root@svr7 ~]# echo $HISTSIZE
200
```

步骤三: 统计 /boot、/etc/pki 目录占用的空间大小

1) 分别统计结果

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

2) 比较 du 与 ls 查看文件大小的差异 (默认块大小 4096 字节):

```
[root@svr7 ~]# ls -lh /etc/inittab           //数据大小 511 字节
-rw-r--r--. 1 root root 511 Sep 16  2015 /etc/inittab
[root@svr7 ~]# du -sh /etc/inittab           //实际占用 4KB 磁盘空间
4.0K    /etc/inittab
```

步骤四: 以格式 “yyyy-mm-dd HH:MM” 显示系统时间

```
[root@svr7 ~]# date +"%F %R"
2016-12-26 16:23
```

2 案例 2: 软连接与硬连接

2.1 问题

本例要求理解软连接与硬连接的基本差异, 完成下列操作:

新建文件 file1, 内容为 AAAA

为 file1 建立软连接 file1-s, 对比两文件内容

为 file1 建立硬连接 file1-h, 对比两文件内容

对比上述 3 个文件的 i 节点编号

删除文件 file1, 再查看文件 file1-s、file1-h 内容

2.2 方案

软连接与硬连接:

软连接: 指向原始文件的路径, 若原始文件被删除, 连接文件将失效; 原始文件可以是目录; 原始文件与连接文件可以在不同的分区/文件系统

硬连接: 指向原始文件的 i 节点档案, 若原始文件被删除, 连接文件仍然有效; 原始文件不能是目录; 原始文件与连接文件必须在同一个分区/文件系统

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一: 使用 ln 命令为文档/目录建立连接

1) 新建一个测试文件

```
[root@svr7 ~]# vim file1
```

AAAA

2) 为文件 file1 建立软连接 file1-s 并测试

```
[root@svr7 ~]# ln -s file1 file1-s
```

```
[root@svr7 ~]# cat file1-s
```

linux.tedu.cn

3) 为文件 file1 建立硬连接 file1-h 并测试

```
[root@svr7 ~]# ln file1 file1-h
```

```
[root@svr7 ~]# cat file1-h
```

linux.tedu.cn

4) 对比原始文件、软连接、硬连接的属性

可以发现软连接只是一个快捷方式，而硬连接与原始文件的 i 节点编号相同，其实对应同一块磁盘存储：

```
[root@svr7 ~]# ls -li /root/f0*.txt
```

```
204645793 -rw-r--r--. 2 root root 14 Jan  6 12:14 file1-h
```

```
201628464 lrwxrwxrwx. 1 root root 12 Jan  6 12:16 file1-s -> file1
```

```
204645793 -rw-r--r--. 2 root root 14 Jan  6 12:14 file1
```

步骤二：原始文件删除测试

1) 当原始文件被删除时，软连接将会失效，而硬连接仍然可访问文件数据

```
[root@svr7 ~]# rm -rf file1
```

```
[root@svr7 ~]# cat file1-s
```

cat: file1-s: No such file or directory

```
[root@svr7 ~]# cat file1-h
```

linux.tedu.cn

2) 如果已知原始文件和硬连接的路径，当原始文件丢失时，可以快速重建

```
[root@svr7 ~]# ln file1-h file1
```

```
[root@svr7 ~]# ls -li /root/f0*.txt
```

```
204645793 -rw-r--r--. 2 root root 14 Jan  6 12:14 file1-h
```

```
201628464 lrwxrwxrwx. 1 root root 12 Jan  6 12:16 file1-s -> file1
```

```
204645793 -rw-r--r--. 2 root root 14 Jan  6 12:14 file1
```

3) 不支持为目录创建硬连接，但可以为目录建立软连接

```
[root@svr7 ~]# ln /etc/sysconfig/network-scripts/ /etc/network
```

ln: '/etc/sysconfig/network-scripts/': hard link not allowed for directory

```
[root@svr7 ~]# ln -s /etc/sysconfig/network-scripts/ /etc/interface
```

```
[root@svr7 ~]# ls -l /etc/interface
```

```
lrwxrwxrwx. 1 root root 31 Jan  6 12:28 /etc/interface -> /etc/sysconfig/network-scripts/
```

3 案例 3: man 手册、zip 备份

3.1 问题

本例要求掌握 man 帮助手册的使用，以及 zip 压缩/解压缩的操作，完成下列任务：

查阅 passwd 命令、/etc/passwd 配置文件的手册页

使用 zip 打包/usr/share/doc/qemu-kvm/目录

3.2 方案

zip/unzip 压缩与解压缩:

制作 zip 压缩包: `zip [-r] 备份文件.zip 被归档的文档...`

释放 zip 压缩包: `unzip 备份文件.zip [-d 目标文件夹]`

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一: 使用 `man` 手册页获取帮助

1) 查看 `passwd` 命令的手册页

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

```
PASSWD(1) User utilities PASSWD(1)
```

NAME

`passwd` - update user's authentication tokens

SYNOPSIS

```
passwd [-k] [-l] [-u [-f]] [-d] [-e] [-n mindays] [-x maxdays] [-w  
warndays] [-i inactivedays] [-S] [--stdin] [username]
```

DESCRIPTION

The `passwd` utility is used to update user's authentication token(s). This task is achieved through calls to the Linux-PAM and Libuser API. Essentially, it initializes itself as a "passwd" service with Linux-PAM and utilizes configured password modules to authenticate and then update a user's password.

.. ..

2) 查看 `/etc/passwd` 配置文件的手册页

```
[root@svr7 ~]# man 2 passwd
```

```
PASSWD(5) Linux Programmer's Manual
```

PASSWD(5)

NAME

`passwd` - password file

DESCRIPTION

The `/etc/passwd` file is a text file that describes user login accounts for the system. It should have read permission allowed for all users (many utilities, like `ls(1)` use it to map user IDs to user - names), but write access only for the superuser.

In the good old days there was no great problem with this general read permission. Everybody could read the encrypted passwords,

but

the hardware was too slow to crack a well-chosen password, and more

-

over the basic assumption used to be that of a friendly user-commu

-

nity. These days many people run some version of the shadow password

suite, where /etc/passwd has an 'x' character in the password

.. ..

步骤二：使用 zip 命令制作压缩包

1) 将目录/usr/share/doc/qemu-kvm/备份为/root/qemu-kvm.zip

```
[root@svr7 ~]# zip -r /root/qemu-kvm.zip /usr/share/doc/qemu-kvm/
adding: usr/share/doc/qemu-kvm/ (stored 0%)
adding: usr/share/doc/qemu-kvm/COPYING (deflated 62%)
adding: usr/share/doc/qemu-kvm/COPYING.LIB (deflated 65%)
adding: usr/share/doc/qemu-kvm/Changelog (deflated 61%)
adding: usr/share/doc/qemu-kvm/LICENSE (deflated 45%)
adding: usr/share/doc/qemu-kvm/README (deflated 4%)
```

2) 恢复测试

删除目标文件夹并确认结果：

```
[root@svr7 ~]# rm -rf /usr/share/doc/qemu-kvm/
[root@svr7 ~]# ls /usr/share/doc/qemu-kvm/
ls: cannot access /usr/share/doc/qemu-kvm/: No such file or directory
```

恢复目标文件夹并确认结果：

```
[root@svr7 ~]# unzip /root/qemu-kvm.zip -d /
Archive: /root/qemu-kvm.zip
  creating: /usr/share/doc/qemu-kvm/
  inflating: /usr/share/doc/qemu-kvm/COPYING
  inflating: /usr/share/doc/qemu-kvm/COPYING.LIB
  inflating: /usr/share/doc/qemu-kvm/Changelog
```

.. ..

```
[root@svr7 ~]# ls /usr/share/doc/qemu-kvm/
COPYING      README      qemu-tech.html
COPYING.LIB  README.rhel6-gpxe-source  qmp-commands.txt
Changelog    README.systemtap  qmp-events.txt
LICENSE      qemu-doc.html    qmp-spec.txt
```

4 案例 4：自定义 yum 软件仓库

4.1 问题

本例要求在 CentOS 真机上利用 RHEL7 的光盘镜像文件准备一个软件仓库目录，完成下列任务：

创建目录 /var/www/html/rh7dvd

挂载 rhel-server-7.2-x86_64-dvd.iso 到上述目录

另外，利用收集的一些第三方 RPM 软件包文件，配置为可发布的 yum 仓库目录，相关任务如下：

挂载 RHEL7 光盘镜像文件到 /var/www/html/ 目录

下载 LibreOffice 办公软件的 rpm 集合版文件

将其中的内容释放到 /opt/libreoffice/ 目录

为 /opt/libreoffice/ 目录建立仓库档案

4.2 方案

作为 yum 软件源的目录需要准备的内容：

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/ 仓库档案提供的数据：

filelists.xml.gz：提供所有软件包的文件安装清单

primary.xml.gz：提供所有软件包的基本/主要信息

other.xml.gz：提供所有软件包的其他信息

repomd.xml：提供上述档案数据文件.xml.gz 的下载和校验信息

4.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：准备 /var/www/html/rh7dvd 仓库目录

1) 创建目录 /var/www/html/rh7dvd

```
[root@room9pc13 ~]# mkdir /var/www/html/rh7dvd
```

2) 挂载 rhel-server-7.2-x86_64-dvd.iso 到上述目录

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

.. ..

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

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

3) 确认部署结果

```
[root@room9pc13 ~]# ls /var/www/html/rh7dvd/
```

```
addons  images      Packages      RPM-GPG-KEY-redhat-release
```

```
EFI     isolinux    release-notes TRANS.TBL
```

```
EULA    LiveOS      repodata
```

```
GPL     media.repo RPM-GPG-KEY-redhat-beta
```

步骤二：准备 /opt/libreoffice/ 仓库目录

1) 将获取的 LibreOffice 软件集合包释放到指定目录

```
[root@room9pc13 ~]# ls LibreOffice_5.1.6.2_Linux_x86-64_rpm.zip
```

```
LibreOffice_5.1.6.2_Linux_x86-64_rpm.zip
```

```
[root@room9pc13 ~]# unzip LibreOffice_5.1*.zip -d /opt/libreoffice
```

```
Archive:  LibreOffice_5.1.6.2_Linux_x86-64_rpm.zip
```

```
  inflating: /opt/libreoffice/install
```

```
    creating: /opt/libreoffice/langpack_zh-CN/
```

```
    creating: /opt/libreoffice/langpack_zh-CN/RPMS/
```

.. ..

//释放到 /opt/libreoffice 目录

2) 使用 createrepo 建立档案

```
[root@room9pc13 ~]# createrepo /opt/libreoffice/
```

```
Spawning worker 0 with 53 pkgs
```

```
Workers Finished
```

```
Saving Primary metadata
```

Saving file lists metadata

Saving other metadata

Generating sqlite DBs

Sqlite DBs complete

3) 确认 repodata/档案资料

```
[root@room9pc13 ~]# ls /opt/libreoffice/repodata
1a5d8311268f33ad2cbf91382110e1ef9875aeea366897253a5d27fd42f9e317-
other.xml.gz
2cd176f0b00492c0c13e0a659eda7dedeb1ab526dec5fd7c9bac7758558770d2
-filelists.xml.gz
6ecab3585a93a917202e177b9569046238332af449a6492fcace96ea7937466
8-filelists.sqlite.bz2
b09a1f838262e0b67a5ab0d6be516bd921a914afd89864e09650f95662a2037
1-primary.sqlite.bz2
b6fcf4a24de5dc08585bf52bd34be50d7df376d5fbcf50903bfd3c1dfdf160bf-oth
er.sqlite.bz2
f57724cf309cc102b2ee25596bf8fb39db3c23880404209ac1e379f7b9fd5c49-p
rimary.xml.gz
repomd.xml
```

5 案例 5：发布及测试 yum 仓库

5.1 问题

沿用案例 5，本例要求掌握发布及测试 yum 仓库的方法，方便在网络内提供集中的 yum 源服务器，主要完成下列任务：

在 CentOS 真机 上发布 yum 源，包括：rhel7 系统的光盘目录仓库、LibreOffice 的 rpm 软件包仓库

在主机 pc207 上使用上述 yum 源

5.2 方案

通过网络发布 yum 软件源时，只需要配置 HTTP 或 FTP 资源服务器，然后将提前准备好的 yum 仓库目录部署到可访问的资源位置即可。

在访问网络 yum 软件源时，注意客户端的 baseurl 地址必须与资源提供方式一致：

baseurl = http://服务器地址/目录名 ==》 /var/www/html/目录名

baseurl = ftp://服务器地址/目录位置 ==》 /var/ftp/目录名

5.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：在 CentOS 真机上发布 yum 仓库

1) 快速搭建 httpd 服务器（若已建好，此步可跳过）

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

//装包

```
[root@room9pc13 ~]# systemctl restart httpd
```

//起服务

```
[root@room9pc13 ~]# systemctl enable httpd
```

//设置开机自

启

2) 确认前一步已经部署到 Web 网站目录的 RHEL7 光盘数据:

```
[root@room9pc13 ~]# du -sh /var/www/html/rh7dvd/ //检查部署结果
```

```
3.9G    /var/www/html/rh7dvd/
```

3) 将准备好的 LibreOffice 仓库目录部署到 Web 网页目录

```
[root@room9pc13 ~]# mv /opt/libreoffice/ /var/www/html/
[root@room9pc13 ~]# du -sh /var/www/html/libreoffice/ //检查部署结果
```

```
234M    /var/www/html/libreoffice
```

步骤二: 在 pc207 上访问 yum 仓库

1) 添加新的 yum 仓库设置

```
[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 = LibreOffice 5
```

```
baseurl=http://192.168.4.254/libreoffice
```

```
gpgcheck = 0
```

2) 测试新的 yum 仓库

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

```
.. ..
```

repo id	repo name	status
libreoffice	LibreOffice 5	53
rh7dvd	RHEL 7.2 Server	4620

```
.. ..
```

6 案例 6: vim 效率操作

6.1 问题

本例要求掌握使用 vim 文本编辑器时能够提高操作效率的一些常用技巧和方法, 完成下列任务:

将文件 /etc/passwd 复制为 /opt/nsd.txt, 然后打开 /opt/nsd.txt 文件, 练习命令模式下的切换/复制/删除/查找操作

将文件 /etc/man_db.conf 复制到 /opt 目录下, 然后打开 /opt/man_db.conf 文件, 将第 50~100 行内的“man”替换为“MAN”, 在 vim 中设置显示行号查看效果

6.2 方案

命令模式常用操作:

1G 或 gg, 跳转到文件的首行

G, 跳转到文件的末尾行

yy、#yy, 复制光标处的一行、#行

p、P, 粘贴到光标处之后、之前

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^、d\$，从光标处之前删除至行首/行尾
/word 向后查找字符串“word”，再按 n/N 跳至后/前一个结果
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
ZZ 保存修改并退出
末行模式常用操作：
: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”
: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”
:w /root/newfile，另存为其它文件
:r /etc/filesystems，读入其他文件内容
:set nu|nonu，显示/不显示行号
:set ai|noai，启用/关闭自动缩进

6.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：vim 命令模式下的切换/复制/删除/查找

1) 建立练习文件

将文件 /etc/passwd 复制为 /opt/nsd.txt:

```
[root@svr7 ~]# cp /etc/passwd /opt/nsd.txt
```

2) 使用 vim 打开练习文件，默认处于命令模式

```
[root@svr7 ~]# vim /opt/nsd.txt
```

.. ..

3) 在命令模式下完成下列操作

切换操作：G 最后一行，5G 第 5 行，gg 第一行。

复制操作：按 2yy 复制 2 行，7G 移动到第 7 行，p 粘贴。

删除操作：25G 移动到第 25 行，200dd 从此行开始删除 200 行（不够就剩下全删）。

查找操作：gg 第一行，/adm 查找关键词 adm，n 跳转到下一个结果。

4) 保存并退出编辑器

ZZ 保存退出。

步骤二：vim 末行模式下的替换/设置操作

1) 建立练习文件

将文件 /etc/man_db.conf 复制到 /opt/ 目录下:

```
[root@svr7 ~]# cp /etc/man_db.conf /opt/
```

2) 使用 vim 打开练习文件，输入:切换到末行模式

```
[root@svr7 ~]# vim /opt/man_db.conf
```

.. ..

:

3) 在末行模式下完成下列操作

输入 `:set nu` , 确认后显示行号。

输入 `:50,100 s/man/MAN/g` , 确认将第 50~100 行内的 “man” 替换为 “MAN”。

4) 保存并退出编辑器

输入 `:wq` , 确认后保存并退出编辑器。

7 案例 7: 编译安装软件包

7.1 问题

本例要求掌握常规源代码应用的安装过程, 通过编译的方式安装 `inotify-tools` 软件工具, 完成下列任务:

释放 `inotify-tools-3.13.tar.gz` 源码包

配置 `./configure`

编译 `make`、安装 `make install`

测试 `inotifywait` 监控工具的用法及用途

7.2 方案

对于标准源码发布的 C/C++ 软件包, 编译安装一般包括以下过程:

解包: 使用 `tar` 命令, 将下载的源代码释放至指定目录

配置: 执行源码目录内的 `./configure` 脚本, 指定安装目录/功能模块等选项

编译: 在源码目录下执行 `make` 操作, 根据配置清单 `Makefile` 生成可执行的二进制程序文件

安装: 在源码目录下执行 `make install` 操作, 将编译好的程序及相关文件复制到安装目录

7.3 步骤

实现此案例需要按照如下步骤进行。

步骤一: 确认已配置好编译环境

```
[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 (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

步骤二: 编译安装 `inotify-tools` 软件包

1) 解包 `inotify-tools-3.13.tar.gz` 文件

```
[root@svr7 ~]# ls inotify-tools-3.13.tar.gz
```

```
inotify-tools-3.13.tar.gz
```

```
[root@svr7 ~]# tar xf inotify-tools-3.13.tar.gz -C /usr/src/
```

2) 配置 `./configure`, 安装目录默认 (`/usr/local/*`)

```
[root@svr7 ~]# cd /usr/src/inotify-tools-3.13/
```

//进入源码目录

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

//配置操作

```

checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
.. ..
configure: creating ./config.status
config.status: creating Makefile
.. ..
[root@svr7 inotify-tools-3.13]# ls    Makefile           //检查配置结果
Makefile
3) 编译 make
[root@svr7 inotify-tools-3.13]# make
.. ..
Making all in src
make[2]: Entering directory `/usr/src/inotify-tools-3.13/src'
make[3]: Entering directory `/usr/src/inotify-tools-3.13'
make[3]: Leaving directory `/usr/src/inotify-tools-3.13'
.. ..
4) 安装 make install
[root@svr7 inotify-tools-3.13]# make    install
.. ..
/usr/bin/install -c .libs/inotifywait /usr/local/bin/inotifywait
  /bin/sh  ../libtool  --mode=install  /usr/bin/install  -c  'inotifywatch'
'usr/local/bin/inotifywatch'
.. ..
[root@svr7 inotify-tools-3.13]# find  /usr/local/  -name  "inotify*"
/usr/local/bin/inotifywait           //确认安装结果
/usr/local/bin/inotifywatch
/usr/local/include/inotifytools
/usr/local/include/inotifytools/inotifytools.h
步骤三：测试 inotify-tools 软件程序

```

软件包 inotify-tools 提供了一个主要程序 inotifywait, 可以用来监控指定目录或文档的变化, 并及时给出通知。

1) 开启对/opt 目录的事件监控

```

[root@svr7 ~]# inotifywait -mrq /opt &           //开启监控
[1] 15568

```

2) 修改/opt/目录内容, 观察屏幕输出信息

```

[root@svr7 ~]# touch  /opt/a.txt                //新建文件 a.txt
/opt/ CREATE a.txt
/opt/ OPEN a.txt
/opt/ ATTRIB a.txt
/opt/ CLOSE_WRITE,CLOSE a.txt
[root@svr7 ~]# mv  /opt/a.txt  /opt/b.txt        //将文件改名
/opt/ MOVED_FROM a.txt

```

/opt/ MOVED_TO b.txt

3) 结束 inotifywait 监控

杀死当前用户的第一个后台任务:

```
[root@svr7 ~]# kill -9 %1
```

```
[1]+  Killed inotifywait -mrq /opt
```

8 案例 8: 使用 systemctl 工具

8.1 问题

本例要求掌握 systemctl 控制工具的基本操作, 完成下列任务:

重启 httpd、crond、bluetooth 服务, 查看状态

禁止 bluetooth 服务开机自启, 并停用此服务

设置默认级别为 multi-user.target 并确认

8.2 方案

systemd 是一个更高效的系统&服务管理器, 其相关特性如下:

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/

systemctl 是 systemd 的管理工具, 将相关资源组织为 unit 配置单元进行管理。

不同的 unit 决定了一组相关的启动任务, service 和 target 是最常用的配置单元:

service: 后台独立服务

target: 一套配置单元的组合, 类似于传统“运行级别”

8.3 步骤

实现此案例需要按照如下步骤进行。

步骤一: 重启 httpd、crond、bluetooth 服务, 查看状态

1) 重启系统服务 httpd、crond、bluetooth

```
[root@svr7 ~]# systemctl restart httpd crond bluetooth
```

2) 查看上述服务的状态

```
[root@svr7 ~]# systemctl status httpd crond bluetooth
```

```
* httpd.service - The Apache HTTP Server
```

```
Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor
preset: disabled)
```

```
Active: active (running) since Fri 2017-01-06 18:18:20 CST; 18s ago
```

```
.. ..
```

```
* crond.service - Command Scheduler
```

```
Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor
preset: enabled)
```

```
Active: active (running) since Fri 2017-01-06 18:18:19 CST; 19s ago
```

```
.. ..
```

```
* bluetooth.service - Bluetooth service
```

```
Loaded: loaded (/usr/lib/systemd/system/bluetooth.service; enabled; vendor
preset: enabled)
```

Active: active (running) since Fri 2017-01-06 18:18:19 CST; 19s ago

.. ..

步骤二：禁止 bluetooth 服务开机自启，并停用此服务

1) 停用 bluetooth 服务

```
[root@svr7 ~]# systemctl stop bluetooth
```

2) 禁止 bluetooth 服务开机自启

```
[root@svr7 ~]# systemctl disable bluetooth
```

```
Removed symlink /etc/systemd/system/dbus-org.bluez.service.
```

```
Removed
```

symlink

```
/etc/systemd/system/bluetooth.target.wants/bluetooth.service.
```

```
[root@svr7 ~]# systemctl is-enabled Bluetooth //检查结果
```

```
disabled
```

步骤三：设置默认级别为 multi-user.target 并确认

1) 查看默认运行级别

```
[root@svr7 ~]# systemctl get-default
```

```
graphical.target
```

2) 将默认运行级别设置为 multi-user.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.
```

3) 确认配置结果

```
[root@svr7 ~]# systemctl get-default
```

```
multi-user.target
```

根据此处的设置，重启此虚拟机后图形桌面将不再可用。

Top

NSD SERVICES DAY02

案例 1：搭建单区域 DNS 服务器

案例 2：特殊 DNS 解析

案例 3：配置 DNS 子域授权

案例 4：搭建并测试缓存 DNS

1 案例 1：搭建单区域 DNS 服务器

1.1 问题

本例要求为 DNS 区域 tedu.cn 搭建一台 DNS 服务器，以使用户能通过域名的方式访问网站。测试阶段主要提供以下正向记录：

svr7.tedu.cn ---> 192.168.4.7

pc207.tedu.cn ---> 192.168.4.207

www.tedu.cn ---> 192.168.4.100

配置完成后在客户机上验证查询结果。

1.2 方案

快速构建 DNS 服务器的基本过程：

安装 bind、bind-chroot 包

建立主配置文件 /etc/named.conf

建立地址库文件 /var/named/.. ..

启动 named 服务

配置及使用 DNS 客户端的基本过程：

修改配置文件/etc/resolv.conf，添加 nameserver=DNS 服务器地址

使用 host 命令查询，提供目标域名作为参数

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置 DNS 服务器 svr7

1) 安装 bind、bind-chroot 包

```
[root@svr7 ~]# yum -y install bind bind-chroot
```

.. ..

2) 建立主配置文件 /etc/named.conf

```
[root@svr7 ~]# mv /etc/named.conf /etc/named.conf.origin //备份
```

默认配置

```
[root@svr7 ~]# vim /etc/named.conf //建立新配置
```

```
options {
```

```
    directory "/var/named"; //地址库默认存放位置
```

```
};
```

```
zone "tedu.cn" { //定义正向 DNS 区域
```

```
    type master; //主区域
```

```
    file "tedu.cn.zone"; //自定义地址库文件名
```

```
};
```

3) 建立地址库文件 /var/named/tedu.cn.zone

```
[root@svr7 ~]# cd /var/named/ //进地址库目录
```

```
[root@svr7 named]# cp -p named.localhost tedu.cn.zone //参考范本建
```

地址库文件

```
[root@svr7 named]# vim tedu.cn.zone //修订地址库
```

记录

```
$TTL 1D //文件开头部分可保持不改
```

```
@ IN SOA @ rname.invalid. (
```

```
    0 ; serial
```

```
    1D ; refresh
```

```
    1H ; retry
```

```
    1W ; expire
```

```
    3H ) ; minimum
```

```
@ NS svr7.tedu.cn.
```

//本区域 DNS 服务器的 FQDN

```

svr7    A    192.168.4.7           //为 NS 主机提供 A 记录
pc207   A    192.168.4.207        //其他正向地址记录.. ..
www     A    192.168.4.100
4) 启动 named 服务，并设置开机自启
[root@svr7 named]# systemctl restart named
[root@svr7 named]# systemctl enable named
Created                               symlink                                from
/etc/systemd/system/multi-user.target.wants/named.service          to
/usr/lib/systemd/system/named.service.
步骤二：配置 DNS 客户机 pc207 并测试

```

1) 修改配置文件/etc/resolv.conf，指定默认使用哪一台 DNS 服务器

```

[root@pc207 ~]# vim /etc/resolv.conf
nameserver 192.168.4.7

```

.. ..

2) 使用 host 命令查询，提供目标域名作为参数

```

[root@pc207 ~]# host svr7.tedu.cn
svr7.tedu.cn has address 192.168.4.7
[root@pc207 ~]# host pc207.tedu.cn
pc207.tedu.cn has address 192.168.4.207
[root@pc207 ~]# host www.tedu.cn
www.tedu.cn has address 192.168.4.100

```

使用 host 测试 DNS 查询结果时，如果不方便修改/etc/resolv.conf 文件，也可以采用“host 目标域名 DNS 服务器地址”形式临时指定使用哪一台 DNS 服务器。

```

[root@pc207 ~]# host pc207.tedu.cn 192.168.4.7
Using domain server:
Name: 192.168.4.7
Address: 192.168.4.7#53
Aliases:
pc207.tedu.cn has address 192.168.4.207

```

2 案例 2：特殊 DNS 解析

2.1 问题

沿用案例 1，本例要求掌握 DNS 轮询、泛域名解析的配置，实现的目标如下：

为站点 www.tedu.cn 提供 DNS 轮询解析，三台 Web 服务器节点的 IP 地址分别为：
192.168.4.100、192.168.4.110、192.168.4.120

配置泛域名解析实现以下解析记录：任意名称.tedu.cn ---> 119.75.217.56

2.2 方案

DNS 轮询：FQDN ---> IP 地址 1、IP 地址 2、.. ..

泛域名解析（站点名不确定）：多个 FQDN ---> 一个 IP 地址

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置 DNS 轮询

1) 修改 DNS 服务器上 tedu.cn 区域的地址库文件，在末尾添加轮询地址记录

```
[root@svr7 ~]# vim /var/named/tedu.cn.zone
```

```
.. ..
```

```
www      A    192.168.4.100
```

```
www      A    192.168.4.110
```

```
www      A    192.168.4.120
```

2) 重启系统服务 named

```
[root@svr7 named]# systemctl restart named
```

3) 在客户机 pc207 上测试轮询记录

针对目标 www.tedu.cn 执行多次查询，观察第 1 条结果的变化：

```
[root@pc207 ~]# host www.tedu.cn
```

```
www.tedu.cn has address 192.168.4.100           //第 1 个结果为 192.168.4.100
```

```
www.tedu.cn has address 192.168.4.110
```

```
www.tedu.cn has address 192.168.4.120
```

```
[root@pc207 ~]# host www.tedu.cn
```

```
www.tedu.cn has address 192.168.4.120           //第 1 个结果为 192.168.4.120
```

```
www.tedu.cn has address 192.168.4.110
```

```
www.tedu.cn has address 192.168.4.100
```

```
[root@pc207 ~]# host www.tedu.cn
```

```
www.tedu.cn has address 192.168.4.110           //第 1 个结果为 192.168.4.110
```

```
www.tedu.cn has address 192.168.4.120
```

```
www.tedu.cn has address 192.168.4.100
```

步骤二：配置多对一的泛域名解析

1) 修改 DNS 服务器上指定区域的地址库文件，在末尾添加*通配地址记录

```
[root@svr7 ~]# vim /var/named/tedu.cn.zone
```

```
.. ..
```

```
*        A    119.75.217.56
```

2) 重启系统服务 named

```
[root@svr7 named]# systemctl restart named
```

3) 在客户机 pc207 上测试多对一的泛域名解析记录

当查询未知站点（地址库中没有明确记录）时，以 * 对应的 IP 地址反馈：

```
[root@pc207 ~]# host station123.tedu.cn
```

```
station123.tedu.cn has address 119.75.217.56
```

```
[root@pc207 ~]# host movie.tedu.cn
```

```
movie.tedu.cn has address 119.75.217.56
```

```
[root@pc207 ~]# host tts8.tedu.cn
```

```
tts8.tedu.cn has address 119.75.217.56
```

3 案例 3：配置 DNS 子域授权

3.1 问题

沿用案例 1，本例要求为上下级两个 DNS 区域建立父子关联，实现客户机向父 DNS 也可

以查询到子域内的 FQDN，基本要求如下：

构建父 DNS（tedu.cn）服务器

构建子 DNS（bj.tedu.cn）服务器

在父 DNS 上配置子域授权

测试子域授权查询

3.2 方案

为一个 DNS 区域添加授权子域时，需要修改此区域的地址库，添加以下记录：

子域域名. IN NS 子 DNS 的 FQDN.

子 DNS 的 FQDN. IN A 子 DNS 的 IP 地址

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：构建父 DNS（tedu.cn）服务器

1) 将 svr7 配置为父 DNS 服务器，确认配置

主配置文件/etc/named.conf:

```
[root@svr7 ~]# viim /etc/named.conf
```

```
options {
    directory "/var/named";
};
zone "tedu.cn" {
    type master;
    file "tedu.cn.zone";
};
... ..
```

正向地址库文件:

```
[root@svr7 ~]# vim /var/named/tedu.cn.zone
```

```
$TTL 1D
```

```
@     IN SOA     @ name.invalid. (
                                0     ; serial
                                1D     ; refresh
                                1H     ; retry
                                1W     ; expire
                                3H )     ; minimum
```

```
@           NS     svr7.tedu.cn.
svr7          A     192.168.4.7
pc207        A     192.168.4.207
www          A     192.168.4.100
... ..
```

确保服务已启用:

```
[root@svr7 ~]# systemctl restart named
```

2) 测试 —— 向父 DNS 可成功查询到父区域中的站点

```
[root@pc207 ~]# host www.tedu.cn 192.168.4.7
```

Using domain server:
Name: 192.168.4.7
Address: 192.168.4.7#53
Aliases:
www.tedu.cn has address 192.168.4.100

.. ..

步骤二：构建子 DNS（bj.tedu.cn）服务器

1) 将 pc207 配置为子 DNS 服务器，确认配置

安装软件包 bind、bind-chroot:

```
[root@pc207 ~]# yum -y install bind bind-chroot
```

.. ..

建立主配置文件/etc/named.conf:

```
[root@pc207 ~]# mv /etc/named.conf /etc/named.conf.origin //备份默认配置
```

```
[root@pc207 ~]# vim /etc/named.conf //建立新配置
```

```
options {  
    directory "/var/named";  
};
```

```
zone "bj.tedu.cn" { //定义子 DNS 的正向区域  
    type master;  
    file "bj.tedu.cn.zone";  
};
```

建立地址库配置文件:

```
[root@pc207 ~]# cd /var/named/ //进地址库目录
```

```
[root@pc207 named]# cp -p named.localhost tedu.cn.zone //参考范本
```

建地址库文件

```
[root@pc207 named]# vim bj.tedu.cn.zone //修订地址库记录
```

```
$TTL 1D //文件开头部分可保持不改
```

```
@ IN SOA @ rname.invalid. (  
    0 ; serial  
    1D ; refresh  
    1H ; retry  
    1W ; expire  
    3H ) ; minimum
```

```
@ NS pc207.bj.tedu.cn. //本区域 DNS 服务器的 FQDN
```

```
pc207 A 192.168.4.207 //为 NS 主机提供 A 记录
```

```
www A 1.2.3.4 //添加测试记录
```

```
www.bj.tedu.cn
```

2) 启动系统服务 named，并设置开机自启

```
[root@pc207 named]# systemctl restart named
```

```
[root@pc207 named]# systemctl enable named
Created symlink from
/etc/systemd/system/multi-user.target.wants/named.service
to
/usr/lib/systemd/system/named.service.
```

3) 测试 —— 向父 DNS 可成功查询到子区域中的站点

```
[root@pc207 ~]# host www.bj.tedu.cn 192.168.4.207
Using domain server:
Name: 192.168.4.207
Address: 192.168.4.207#53
Aliases:
www.bj.tedu.cn has address 1.2.3.4
```

步骤三：在父 DNS 上配置子域授权

1) 测试 —— 未配置子域授权时，向父 DNS 无法正确查询到子区域中的站点

若父 DNS 配置有 * 泛域名，则反馈的结果为对应的 IP 地址 119.75.217.56，而不是子 DNS 中记录的 1.2.3.4:

```
[root@pc207 ~]# host www.bj.tedu.cn 192.168.4.7
Using domain server:
Name: 192.168.4.7
Address: 192.168.4.7#53
Aliases:
www.bj.tedu.cn has address 119.75.217.56
```

若父 DNS 未配置有 * 泛域名，则找不到解析结果 (not found):

```
[root@pc207 ~]# host www.bj.tedu.cn 192.168.4.7
Using domain server:
Name: 192.168.4.7
Address: 192.168.4.7#53
Aliases:
Host www.bj.tedu.cn not found: 3(NXDOMAIN)
```

2) 修改父 DNS 区域 tedu.cn 的地址库，添加授权子域信息

```
[root@svr7 ~]# vim /var/named/tedu.cn.zone
```

```
.. ..
bj.tedu.cn.      NS      pc207.bj.tedu.cn.      //子区域及子 DNS 主
机名
pc207.bj.tedu.cn.  A      192.168.4.207      //子 DNS 的 IP 地址
[root@svr7 named]# systemctl restart named      //重启服务
```

步骤四：测试子域授权查询

测试 —— 成功配置子域授权以后，向父 DNS 可以正确查询到子区域中的站点:

```
[root@pc207 ~]# host www.bj.tedu.cn 192.168.4.7
Using domain server:
Name: 192.168.4.7
Address: 192.168.4.7#53
Aliases:
```

www.bj.tedu.cn has address 1.2.3.4

4 案例 4：搭建并测试缓存 DNS

4.1 问题

本例要求熟悉缓存 DNS 的工作过程，准备一台可上网的 RHEL7 虚拟机，并完成下列任务：

安装 bind、bind-chroot 包

搭建并测试基于全局转发器的缓存 DNS

注意：若所在机房不具备访问互联网 DNS 条件，此案例改由学员自行在家完成。

4.2 方案

权威/官方 DNS 服务器的特点：

至少管理一个 DNS 区域，需要 IANA 等官方机构授权

典型应用：根域 DNS、一级域 DNS、二级域 DNS、三级域 DNS、.. ..

缓存 DNS 服务器的特点：

不需要管理任何 DNS 区域，但是能够替客户机查询，而且通过缓存、复用查询结果来加快响应速度

典型应用：ISP 服务商、企业局域网

缓存 DNS 服务器的解析记录来源：

方式 1：全局转发：将请求转发给指定的公共 DNS（其他缓存 DNS），请求递归服务

方式 2：根域迭代：依次向根、一级、二级……域的 DNS 服务器迭代

4.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：为虚拟机 pc207 提供上网条件

1) 为虚拟机添加一块新的网卡，选择 NAT 或 Bridge 模式

若选择 NAT 模式（地址转换），则新加网卡的上网参数由虚拟化平台自动设置。

若选择 Bridge 模式（桥接），则新加网卡的上网参数需要参考真实网络的主机，必要时请网络管理员提供支持。

此处所列地址信息可帮助大家理解上网条件，但不作为练习的配置依据：

```
[root@pc207 ~]# ifconfig eth1 //检查新增网卡的 IP 地址
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.70.129 netmask 255.255.255.0 broadcast
    192.168.70.255
.. ..
[root@pc207 ~]# route -n //确认已配好默认网关
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use
Iface
0.0.0.0 192.168.70.2 0.0.0.0 UG 100 0 0
eth1
192.168.70.0 0.0.0.0 255.255.255.0 U 100 0 0
eth1
.. ..
```

```
[root@pc207 ~]# cat /etc/resolv.conf //确认第一 DNS 为外部可用
DNS 地址
nameserver 192.168.70.2
```

.. ..

2) 确保从主机 pc207 可访问到外部 DNS

访问默认 DNS 可用 (本机正常连网需要):

```
[root@pc207 ~]# host www.qq.com
www.qq.com has address 111.30.132.101
www.qq.com has IPv6 address 240e:e1:8100:28::2:16
```

访问指定 DNS 可用 (全局转发的前提条件):

```
[root@pc207 ~]# host www.qq.com 202.106.0.20 //国内公共 DNS 服务器之一
```

Using domain server:

Name: 202.106.0.20

Address: 202.106.0.20#53

Aliases:

```
www.qq.com has address 111.30.132.101
www.qq.com is an alias for qq.com.edgesuite.net.
qq.com.edgesuite.net is an alias for a1574.b.akamai.net.
www.qq.com is an alias for qq.com.edgesuite.net.
qq.com.edgesuite.net is an alias for a1574.b.akamai.net.
```

步骤二: 将 pc207 配置为缓存 DNS (全局转发式)

1) 安装 bind、bind-chroot 软件包

```
[root@pc207 ~]# yum -y install bind bind-chroot
```

.. ..

2) 建立主配置文件/etc/named.conf

当收到来自客户机的 DNS 查询请求时, 转发到外网的其他 DNS 服务器

```
[root@pc207 ~]# vim /etc/named.conf
```

```
options {
    forwarders { 202.106.0.20; };
};
```

3) 启动系统服务 named, 并设置开机自启

```
[root@pc207 ~]# systemctl restart named
```

```
[root@pc207 ~]# systemctl enable named
```

4) 可向缓存 DNS 服务器 pc207 查询到公共域名 (百度、网易等站点)

```
[root@pc207 ~]# host www.baidu.com 192.168.4.207 //查百度的
站点 IP
```

Using domain server:

Name: 192.168.4.207

Address: 192.168.4.207#53

Aliases:

```
www.baidu.com is an alias for www.a.shifen.com.
www.a.shifen.com has address 111.13.100.92
```

```
www.a.shifen.com has address 111.13.100.91
[root@pc207 ~]# host www.163.com 192.168.4.207           //查网易
的站点 IP
Using domain server:
Name: 192.168.4.207
Address: 192.168.4.207#53
Aliases:
www.163.com is an alias for www.163.com.lxdns.com.
www.163.com.lxdns.com is an alias for 163.xdwscache.ourglb0.com.
163.xdwscache.ourglb0.com has address 111.11.31.104
163.xdwscache.ourglb0.com has address 111.11.31.114
```

Top

NSD SERVICES DAY03

案例 1：配置并验证 Split 分离解析

案例 2：查看进程信息

案例 3：进程调度及终止

案例 4：系统日志分析

1 案例 1：配置并验证 Split 分离解析

1.1 问题

本例要求配置一台智能 DNS 服务器，针对同一个 FQDN，当不同的客户机来查询时能够给出不同的答案。需要完成下列任务：

从主机 192.168.4.207 查询时，结果为：www.tedu.cn ---> 192.168.4.100

从其他客户端查询时，www.tedu.cn ---> 1.2.3.4

1.2 方案

在配置 DNS 服务器时，通过 view 视图设置来区分不同客户机、不同地址库：

```
view "视图 1" {
    match-clients { 客户机地址 1; .. .. };           //匹配第 1 类客户机地址
    zone "目标域名" IN {                               //同一个 DNS 区域
        type master;
        file "地址库 1";                               //第 1 份地址库
    };
};
view "视图 2" {
    match-clients { 客户机地址 2; .. .. };           //匹配第 2 类客户机地址
    match-clients { any; };                           //匹配任意地址
    zone "目标域名" IN {                               //同一个 DNS 区域
        type master;
        file "地址库 2";                               //第 2 份地址库
    };
};
```

```
};
...
view "视图 n" {
    match-clients { any; };           //匹配任意地址
    zone "目标域名" IN {              //同一个 DNS 区域
        type master;
        file "地址库 n";             //第 n 份地址库
    };
};
```

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置 Split 分离解析

1) 为 **tedu.cn** 区域建立两份解析记录文件

第一份解析记录文件提供给客户机 **192.168.4.207**、网段 **192.168.7.0/24**，对应目标域名 **www.tedu.cn** 的 A 记录地址为 **192.168.4.100**。相关操作及配置如下：

```
[root@svr7 ~]# cd /var/named/
[root@svr7 named]# cp -p tedu.cn.zone tedu.cn.zone.lan
[root@svr7 named]# vim tedu.cn.zone.lan
$TTL 1D
@ IN SOA @ rname.invalid. (
                        0 ; serial
                        1D ; refresh
                        1H ; retry
                        1W ; expire
                        3H ) ; minimum

@ NS svr7.tedu.cn.
svr7 A 192.168.4.7
pc207 A 192.168.4.207
www A 192.168.4.100
```

第二份解析记录文件提供给其他客户机，对应目标域名 **www.tedu.cn** 的 A 记录地址为 **1.2.3.4**。相关操作及配置如下：

```
[root@svr7 named]# cp -p tedu.cn.zone tedu.cn.zone.other
[root@svr7 named]# vim tedu.cn.zone.other
$TTL 1D
@ IN SOA @ rname.invalid. (
                        0 ; serial
                        1D ; refresh
                        1H ; retry
                        1W ; expire
                        3H ) ; minimum

@ NS svr7.tedu.cn.
svr7 A 192.168.4.7
```

```
pc207 A 192.168.4.207
```

```
www A 1.2.3.4
```

2) 修改 named.conf 配置文件, 定义两个 view, 分别调用不同解析记录文件

```
[root@svr7 ~]# vim /etc/named.conf
```

```
options {  
    directory "/var/named";  
};  
acl "mylan" {                                //名为 mylan 的列表  
    192.168.4.207; 192.168.7.0/24;  
};  
..  
view "mylan" {  
    match-clients { mylan; };                //检查客户机地址是否匹配此列表  
    zone "tedu.cn" IN {  
        type master;  
        file "tedu.cn.zone.lan";  
    };  
};  
view "other" {  
    match-clients { any; };                  //匹配任意客户机地址  
    zone "tedu.cn" IN {  
        type master;  
        file "tedu.cn.zone.other";  
    };  
};
```

3) 重启 named 服务

```
[root@svr7 ~]# systemctl restart named
```

步骤二: 测试分离解析效果

1) 从 mylan 地址列表中的客户机查询

在客户机 192.168.4.207 (或网段 192.168.7.0/24 内的任意客户机) 上查询 www.tedu.cn, 结果是 192.168.4.100:

```
[root@pc207 ~]# host www.tedu.cn 192.168.4.7
```

Using domain server:

Name: 192.168.4.7

Address: 192.168.4.7#53

Aliases:

www.tedu.cn has address 192.168.4.100

2) 从其他客户机查询

在 DNS 服务器本机或 CentOS 真机上查询 www.tedu.cn 时, 结果为 1.2.3.4:

```
[root@svr7 ~]# host www.tedu.cn 192.168.4.7
```

Using domain server:

Name: 192.168.4.7

Address: 192.168.4.7#53

Aliases:

www.tedu.cn has address 1.2.3.4

2 案例 2：查看进程信息

2.1 问题

本例要求掌握查看进程信息的操作，使用必要的命令工具完成下列任务：

找出进程 **gdm** 的 **PID** 编号值

列出由进程 **gdm** 开始的子进程树结构信息

找出进程 **sshd** 的父进程的 **PID** 编号/进程名称

查看当前系统的 **CPU** 负载/进程总量信息

2.2 方案

查看进程的主要命令工具：

ps aux、**ps -elf**：查看进程静态快照

top：查看进程动态排名

pstree：查看进程与进程之间的树型关系结构

pgrep：根据指定的名称或条件检索进程

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：找出进程 **gdm** 的 **PID** 编号值

使用 **pgrep** 命令查询指定名称的进程，选项 **-l** 显示 **PID** 号、**-x** 精确匹配进程名：

```
[root@svr7 ~]# pgrep -lx gdm
```

```
1584 gdm
```

步骤二：列出由进程 **gdm** 开始的子进程树结构信息

使用 **pstree** 命令，可以提供用户名或 **PID** 值作为参数。通过前一步已知进程 **gdm** 的 **PID** 为 **1584**，因此以下操作可列出进程 **gdm** 的进程树结构：

```
[root@svr7 ~]# pstree -p 1584
```

```
gdm(1584)-+-Xorg(1703)
            |-gdm-session-wor(2670)-+-gnome-session(2779)-+-gnom+
            |                                     |-gnom+
            |                                     |-{gno+
            |                                     |-{gno+
            |                                     `-{gno+
            |                                     |-{gdm-session-wor}(2678)
            |                                     `-{gdm-session-wor}(2682)
            |-{gdm}(1668)
            |-{gdm}(1671)
            `-{gdm}(1702)
```

步骤三：找出进程 **sshd** 的父进程的 **PID** 编号/进程名称

要查看进程的父进程 PID，可以使用 `ps -elf` 命令，简单 `grep` 过滤即可。找到进程 `sshd` 所在行对应到的 PPID 值即为其父进程的 PID 编号。为了方便直观查看，建议先列出 `ps` 表头行，以分号隔开再执行过滤操作。

```
[root@svr7 ~]# ps -elf | head -1 ; ps -elf | grep sshd
F S UID          PID    PPID  C PRI  NI ADDR SZ WCHAN  STIME TTY
TIME CMD
4 S root         1362      1  0  80   0 - 20636 poll_s Jan05 ?      00:00:00
/usr/sbin/sshd -D
```

.. .. //可获知进程 `sshd` 的父进程 PID 为 1

然后再根据 `pstree -p` 的结果过滤，可获知 PID 为 1 的进程名称为 `systemd`：

```
[root@svr7 ~]# pstree -p | grep '(1)'
systemd(1)-+-ModemManager(995)-+-{ModemManager}(1018)
```

步骤四：查看当前系统的 CPU 负载/进程总量信息

使用 `top` 命令，直接看开头部分即可；或者 `top -n 次数`：

```
[root@svr7 ~]# top
top - 15:45:25 up 23:55, 2 users, load average: 0.02, 0.03, 0.05
Tasks: 485 total, 2 running, 483 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.0 sy, 0.0 ni, 97.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 1001332 total, 76120 free, 419028 used, 506184
buff/cache
KiB Swap: 2097148 total, 2096012 free, 1136 used. 372288 avail
Mem
.. ..
```

观察 `Tasks: 485 total` 部分，表示进程总量信息。

观察 `load average: 0.02, 0.03, 0.05` 部分，表示 CPU 处理器在最近 1 分钟、5 分钟、15 分钟内的平均处理请求数（对于多核 CPU，此数量应除以核心数）。

对于多核 CPU 主机，如果要分别显示每颗 CPU 核心的占用情况，可以在 `top` 界面按数字键 1 进行切换：

```
[root@svr7 ~]# top
top - 15:47:45 up 23:57, 2 users, load average: 0.02, 0.03, 0.05
Tasks: 485 total, 2 running, 269 sleeping, 0 stopped, 1 zombie
Cpu0  : 0.6%us, 7.8%sy, 0.0%ni, 91.6%id, 0.0%wa, 0.0%hi, 0.0%si,
0.0%st
Cpu1  : 0.7%us, 3.7%sy, 0.0%ni, 95.6%id, 0.0%wa, 0.0%hi, 0.0%si,
0.0%st
Cpu2  : 0.7%us, 1.7%sy, 0.0%ni, 97.6%id, 0.0%wa, 0.0%hi, 0.0%si,
0.0%st
Cpu3  : 0.3%us, 1.0%sy, 0.0%ni, 98.3%id, 0.3%wa, 0.0%hi, 0.0%si,
0.0%st
Mem: 16230564k total, 15716576k used, 513988k free, 326124k buffers
Swap: 8388604k total, 220656k used, 8167948k free, 11275304k cached
.. ..
```

3 案例 3：进程调度及终止

3.1 问题

本例要求掌握调度及终止进程的操作，使用必要的工具完成下列任务：

运行“sleep 600”命令，再另开一个终端，查出 sleep 程序的 PID 并杀死

运行多个 vim 程序并都放入后台，然后杀死所有 vim 进程

su 切换为 zhsan 用户，再另开一个终端，强制踢出 zhsan 用户

3.2 方案

进程调度及终止的主要命令工具：

命令行 &：将命令行在后台运行

Ctrl + z 组合键：挂起当前进程（暂停并转入后台）

jobs：列出当前用户当前终端的后台任务

bg 编号：启动指定编号的后台任务

fg 编号：将指定编号的后台任务调入前台运行

kill [-9] PID...：杀死指定 PID 值的进程

kill [-9] %n：杀死第 n 个后台任务

killall [-9] 进程名...：杀死指定名称的所有进程

pkill：根据指定的名称或条件杀死进程

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：根据 PID 杀死进程

1) 开启 sleep 测试进程

```
[root@svr7 ~]# sleep 600
```

//.. .. 进入 600 秒等待状态

2) 找出进程 sleep 的 PID

另开一个终端，ps aux 并过滤进程信息（第 2 列为 PID 值）：

```
[root@svr7 ~]# ps aux | grep sleep
```

```
root      32929  0.0  0.0  4312  360 pts/1    S+   17:25   0:00 sleep
600
```

3) 杀死指定 PID 的进程

```
[root@svr7 ~]# kill -9 32929
```

返回原终端会发现 sleep 进程已经被杀死：

```
[root@svr7 ~]# sleep 600
```

```
Killed
```

步骤二：根据进程名杀死多个进程

1) 在后台开启多个 vim 进程

```
[root@svr7 ~]# vim a.txt &
```

```
[1] 33152
```

```
[root@svr7 ~]# vim b.txt &
```

```
[2] 33154
```

```
[1]+  已停止                  vim a.txt
```

```
[root@svr7 ~]# vim c.txt &
[3] 33155
[2]+ 已停止                  vim b.txt
2) 确认 vim 进程信息
[root@svr7 ~]# jobs -l
[1] 33152 停止 (tty 输出)    vim a.txt
[2]- 33154 停止 (tty 输出)    vim b.txt
[3]+ 33155 停止 (tty 输出)    vim c.txt
3) 强制杀死所有名为 vim 的进程
[root@svr7 ~]# killall -9 vim
[1] 已杀死                  vim a.txt
[2]- 已杀死                  vim b.txt
[3]+ 已杀死                  vim c.txt
4) 确认杀进程结果
[root@svr7 ~]# jobs -l
[root@svr7 ~]#
```

步骤三：杀死属于指定用户的所有进程

1) 登入测试用户 zhsan

```
[root@svr7 ~]# useradd zhsan
[root@svr7 ~]# su - zhsan
[zhsan@svr7 ~]$
```

2) 另开一个终端，以 root 用户登入，查找属于用户 zhsan 的进程

```
[root@svr7 ~]# pgrep -u zhsan
33219
```

```
[root@svr7 ~]# pstree -up 33219
bash(33219,zhsan)
```

//检查进程树

3) 强制杀死属于用户 zhsan 的进程

```
[root@svr7 ~]# pkill -9 -u zhsan
[root@svr7 ~]#
```

4) 返回原来用户 zhsan 登录的终端，确认已经被终止

```
[zhsan@svr7 ~]$ 已杀死
[root@svr7 ~]#
```

4 案例 4：系统日志分析

4.1 问题

本例要求熟悉 Linux 系统中的常见日志文件，使用必要的命令工具完成下列任务：

列出所有包含关键词 **8909** 的系统日志消息

查看启动时识别的鼠标设备信息

列出最近 2 条成功/不成功的用户登录消息

列出最近 10 条重要程度在 **ERR** 及以上的日志消息

列出所有与服务 **httpd** 相关的消息

列出前 4 个小时内新记录的日志

4.2 方案

常见的系统日志及各自用途：

/var/log/messages，记录内核消息、各种服务的公共消息

/var/log/dmesg，记录系统启动过程的各种消息

/var/log/cron，记录与 cron 计划任务相关的消息

/var/log/maillog，记录邮件收发相关的消息

/var/log/secure，记录与访问限制相关的安全消息

日志消息的优先级（高-->低）：

EMERG（紧急）：级别 0，系统不可用的情况

ALERT（警报）：级别 1，必须马上采取措施的情况

CRIT（严重）：级别 2，严重情形

ERR（错误）：级别 3，出现错误

WARNING（警告）：级别 4，值得警告的情形

NOTICE（注意）：级别 5，普通但值得引起注意的事件

INFO（信息）：级别 6，一般信息

DEBUG（调试）：级别 7，程序/服务调试消息

RHEL7 提供的 journalctl 日志工具的常见用法：

journalctl | grep 关键词

journalctl -u 服务名 -p 优先级

journalctl -n 消息条数

journalctl --since="yyyy-mm-dd HH:MM:SS" --until="yyyy-mm-dd HH:MM:SS"

4.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：分析系统日志及用户日志

1) 列出所有包含关键词 8909 的系统日志消息

简单模拟一个故障（SELinux 阻止 Web 开放 8909 端口）：

```
[root@svr7 ~]# vim /etc/httpd/conf.d/8909.conf //添加开 8909 端口配置
```

Listen 8909

```
[root@svr7 ~]# setenforce 1 //开启强制模式
```

```
[root@svr7 ~]# systemctl restart httpd //起服务失败
```

Job for httpd.service failed because the control process exited with error code.
See "systemctl status httpd.service" and "journalctl -xe" for details.

从日志文件/var/log/messages 中检索信息：

```
[root@svr7 ~]# grep 8909 /var/log/messages
```

```
Jan 6 17:53:48 svr7 setroubleshoot: SELinux is preventing /usr/sbin/httpd from  
name_bind access on the tcp_socket port 8909. For complete SELinux  
messages. run sealert -l 6d37b8f0-ab8a-4082-9295-c784f4f57190
```

```
Jan 6 17:53:48 svr7 python: SELinux is preventing /usr/sbin/httpd from  
name_bind access on the tcp_socket port 8909.#012#012***** Plugin  
bind_ports (92.2 confidence) suggests *****#012#012If  
you want to allow /usr/sbin/httpd to bind to network port 8909#012Then you
```

need to modify the port type.#012Do#012# semanage port -a -t PORT_TYPE -p tcp 8909#012 where PORT_TYPE is one of the following: http_cache_port_t, http_port_t, jboss_management_port_t, jboss_messaging_port_t, ntop_port_t, puppet_port_t.#012#012***** Plugin catchall_boolean (7.83 confidence) suggests *****#012#012If you want to allow nis to enabled#012Then you must tell SELinux about this by enabling the 'nis_enabled' boolean.#012#012Do#012setsebool -P nis_enabled 1#012#012***** Plugin catchall (1.41 confidence) suggests *****#012#012If you believe that httpd should be allowed name_bind access on the port 8909 tcp_socket by default.#012Then you should report this as a bug.#012You can generate a local policy module to allow this access.#012Do#012allow this access for now by executing:#012# grep httpd /var/log/audit/audit.log | audit2allow -M mypol#012# semodule -i mypol.pp#012

.. ..

使用完毕记得删除测试配置文件:

```
[root@svr7 ~]# rm -rf /etc/httpd/conf.d/8909.conf
```

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

2) 查看启动时识别的鼠标设备信息

```
[root@svr7 ~]# dmesg | grep -i mouse
```

```
[ 1.020385] mousedev: PS/2 mouse device common for all mice
```

```
[ 1.249422] input: ImPS/2 Generic Wheel Mouse as /devices/platform/i8042/serio1/input/input2
```

```
[ 2.279665] usb 2-1: Product: VMware Virtual USB Mouse
```

```
[ 2.603999] input: VMware VMware Virtual USB Mouse as /devices/pci0000:00/0000:00:11.0/0000:02:00.0/usb2/2-1/2-1:1.0/input/input3
```

```
[ 2.604222] hid-generic 0003:0E0F:0003.0001: input,hidraw0: USB HID v1.10 Mouse [VMware VMware Virtual USB Mouse] on usb-0000:02:00.0-1/input0
```

3) 列出最近 2 条成功/不成功的用户登录消息

查看成功登录的事件消息:

```
[root@svr7 ~]# last -2
```

```
zhsan pts/2 192.168.4.207 Fri Jan 6 18:00 - 18:00 (00:00)
```

```
root pts/2 192.168.4.110 Fri Jan 6 17:26 - 17:59 (00:33)
```

```
wtmp begins Thu Aug 4 00:10:16 2016
```

查看失败登录的事件消息:

```
[root@svr7 ~]# lastb -2
```

```
anonymou ssh:notty 192.168.4.207 Fri Jan 6 18:00 - 18:00 (00:00)
```

```
anonymou ssh:notty 192.168.4.207 Fri Jan 6 18:00 - 18:00 (00:00)
```

```
btmtp begins Fri Jan 6 18:00:34 2017
```

步骤二: 使用 journalctl 日志提取工具

1) 列出最近 10 条重要程度在 ERR 及以上的日志消息

```
[root@svr7 ~]# journalctl -p err -n 10
```

-- Logs begin at Thu 2017-01-05 15:50:08 CST, end at Fri 2017-01-06 18:01:01 CST. --

Jan 06 14:56:57 svr7 setroubleshoot[23702]: SELinux is preventing /usr/sbin/vsftpd from getattr access on the file /rhel7/repodata/repomd.xml. For complete SELinux mes

Jan 06 14:56:57 svr7 setroubleshoot[23702]: SELinux is preventing /usr/sbin/vsftpd from read access on the file repomd.xml. For complete SELinux messages. run sealert

Jan 06 14:56:57 svr7 setroubleshoot[23702]: SELinux is preventing /usr/sbin/vsftpd from read access on the file repomd.xml. For complete SELinux messages. run sealert

Jan 06 14:56:57 svr7 setroubleshoot[23702]: SELinux is preventing /usr/sbin/vsftpd from lock access on the file /rhel7/repodata/repomd.xml. For complete SELinux messag

Jan 06 17:53:48 svr7 setroubleshoot[33743]: Plugin Exception restorecon_source

Jan 06 17:53:48 svr7 setroubleshoot[33743]: SELinux is preventing /usr/sbin/httpd from name_bind access on the tcp_socket port 8909. For complete SELinux messages. run

Jan 06 17:53:53 svr7 setroubleshoot[33743]: SELinux is preventing /usr/sbin/httpd from name_connect access on the tcp_socket port 8909. For complete SELinux messages.

Jan 06 17:53:54 svr7 systemd[1]: Failed to start The Apache HTTP Server.

.. ..

lines 1-11/11 (END)

2) 列出所有与服务 httpd 相关的消息

[root@svr7 ~]# journalctl -u httpd

-- Logs begin at Thu 2017-01-05 15:50:08 CST, end at Fri 2017-01-06 18:01:01 CST. --

Jan 06 14:57:16 svr7 systemd[1]: Starting The Apache HTTP Server...

Jan 06 14:57:16 svr7 httpd[23812]: AH00557: httpd: apr_sockaddr_info_get() failed for svr7

Jan 06 14:57:16 svr7 httpd[23812]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directi

Jan 06 14:57:16 svr7 systemd[1]: Started The Apache HTTP Server.

Jan 06 17:53:44 svr7 systemd[1]: Stopping The Apache HTTP Server...

Jan 06 17:53:46 svr7 systemd[1]: Starting The Apache HTTP Server...

Jan 06 17:53:46 svr7 httpd[33741]: AH00557: httpd: apr_sockaddr_info_get() failed for svr7

.. ..

3) 列出前 4 个小时内新记录的日志

根据当前日期时间往前推 4 个小时, 确定--since 起始和--until 结束时刻:

[root@svr7 ~]# journalctl --since "2017-01-06 14:11" --until "2017-01-06

18:11"

-- Logs begin at Thu 2017-01-05 15:50:08 CST, end at Fri 2017-01-06 18:10:01 CST. --

Jan 06 14:20:01 svr7 systemd[1]: Started Session 160 of user root.

Jan 06 14:20:01 svr7 CROND[22869]: (root) CMD (/usr/lib64/sa/sa1 1 1)

Jan 06 14:20:01 svr7 systemd[1]: Starting Session 160 of user root.

Jan 06 14:30:01 svr7 systemd[1]: Started Session 161 of user root.

Jan 06 14:30:01 svr7 CROND[23028]: (root) CMD (/usr/lib64/sa/sa1 1 1)

Jan 06 14:31:39 svr7 systemd[1]: Starting Session 162 of user root.

Jan 06 14:32:17 svr7 sshd[23046]: pam_unix(sshd:session): session closed for user root

Jan 06 14:31:39 svr7 systemd[1]: Started Session 162 of user root.

Jan 06 14:31:39 svr7 sshd[23046]: pam_unix(sshd:session): session opened for user root by (uid=0)

Jan 06 14:31:39 svr7 systemd-logind[985]: New session 162 of user root.

.. .

Top

NSD SERVICES DAY04

案例 1: PXE 基础装机环境

案例 2: 配置并验证 DHCP 服务

案例 3: 配置 PXE 引导

案例 4: 验证 PXE 网络装机

案例 5: PXE+kickstart 自动装机

1 案例 1: PXE 基础装机环境

1.1 问题

本例要求为后续的 PXE 服务器构建提供 RHEL7 软件仓库，完成下列任务：

在 CentOS 真机部署 Web 目录/var/www/html/rh7dvd

挂载 RHEL7 光盘镜像文件到该目录

访问 <http://192.168.4.254/rhel7/> 测试，确保可用

1.2 方案

PXE 网络装机的整体思路 —— 装机条件准备：

准备 RHEL7 安装源（HTTP 方式 YUM 库）

启用 DHCP 服务

PXE 网络装机的整体思路 —— PXE 引导配置：

启用 TFTP 服务，提供装机用的内核、初始化文件

提供 PXE 引导程序、配置启动菜单

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：通过 HTTP 方式发布 RHEL7 软件源

此环节可以直接使用网络内共有的 YUM 软件源，比如 CentOS 真机。

1) 快速构建 httpd 服务器（若已构建，此步可跳过）

```
[root@room9pc13 ~]# yum -y install httpd           //装包
[root@room9pc13 ~]# systemctl restart httpd        //启动服务
[root@room9pc13 ~]# systemctl enable httpd          //设置开机自启
```

2) 准备 yum 仓库，部署到 Web 子目录

```
[root@room9pc13 ~]# mkdir /var/www/html/rh7dvd      //建挂载点
[root@room9pc13 ~]# vim /etc/fstab

.. ..
/ISO/rhel-server-7.2-x86_64-dvd.iso /var/www/html/rh7dvd iso9660 loop,ro
0 0
[root@room9pc13 ~]# mount -a                        //挂载 ISO 镜像文件
[root@room9pc13 ~]# ls /var/www/html/rh7dvd/        //确认部署位置
addons isolinux repodata
EFI LiveOS RPM-GPG-KEY-redhat-beta
EULA media.repo RPM-GPG-KEY-redhat-release
GPL Packages TRANS.TBL
images release-notes
```

步骤二：确保 yum 仓库 HTTP 资源可用

从浏览器访问 <http://192.168.4.254/rh7dvd/>，可看到仓库资源（如图-1 所示）。

图-1

2 案例 2：配置并验证 DHCP 服务

2.1 问题

本例要求为 PXE 客户机提供地址分配服务，在主机 svr7 上搭建支持 PXE 的 DHCP 服务器，提供的地址参数如下：

IP 地址范围 192.168.4.10~200/24

PXE 引导服务器位于 192.168.4.7、引导文件 pxelinux.0

然后在主机 pc207 上使用 dhclient 命令测试 DHCP 地址分配服务。

2.2 方案

DHCP 地址分配的四次会话：DISCOVERY --> OFFER --> REQUEST --> ACK。

DHCP 服务器基本概念：

DHCP 租期：允许客户机租用 IP 地址的时间期限，单位为秒

DHCP 作用域：分配给客户机的 IP 地址所在的网段

DHCP 地址池：用来动态分配的 IP 地址的范围

DHCP 服务端：软件包 dhcp、系统服务 dhcpd

DHCP 服务端配置文件：/etc/dhcp/dhcpd.conf

传输协议及端口：UDP 67（服务器）、UDP 68（客户端）

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置 DHCP 服务端

1) 安装 dhcp 软件包

```
[root@svr7 ~]# yum -y install dhcp
```

.. ..

2) 建立 dhcpd.conf 服务配置

```
[root@svr7 ~]# vim /etc/dhcp/dhcpd.conf
subnet 192.168.4.0 netmask 255.255.255.0 {
    range 192.168.4.10 192.168.4.200;
    next-server 192.168.4.7;
    filename "pxelinux.0";
}
```

3) 启动系统服务 dhcpd，并设置开机自启

```
[root@svr7 ~]# systemctl restart dhcpd
```

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

```
Created symlink from
/etc/systemd/system/multi-user.target.wants/dhcpd.service
to
/usr/lib/systemd/system/dhcpd.service.
```

4) 确认 dhcpd 服务状态

```
[root@svr7 ~]# netstat -anptu | grep dhcpd
udp        0          0 0.0.0.0:67          0.0.0.0:*
58693/dhcpd
```

.. ..

步骤二：在客户端测试 DHCP 服务

1) 使用 dhclient 命令测试，观察获取 IP 地址的过程

```
[root@pc207 ~]# dhclient -d eth0
```

Internet Systems Consortium DHCP Client 4.2.5

Copyright 2004-2013 Internet Systems Consortium.

All rights reserved.

For info, please visit <https://www.isc.org/software/dhcp/>

.. ..

```
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 7
(xid=0x6707682f)
```

```
DHCPREQUEST on eth0 to 255.255.255.255 port 67 (xid=0x6707682f)
```

```
DHCPOFFER from 192.168.4.7
```

```
DHCPACK from 192.168.4.7 (xid=0x6707682f)
```

```
bound to 192.168.4.10 -- renewal in 18008 seconds.
```

```
^C //按 Ctrl+c 键退出测试
```

2) 若因操作异常导致 IP 故障，可恢复客户机原有 IP 配置

```
[root@pc207 ~]# pkill -9 dhclient
```

//杀死 dhclient 进程

```
[root@pc207 ~]# nmcli connection up eth0
```

//激活原配置

.. ..

3 案例 3：配置 PXE 引导

3.1 问题

本例要求为 PXE 装机提供引导服务，并提供必要的素材，完成下列任务：

启用 TFTP 服务器，部署引导文件（内核 `vmlinuz`、初始文件 `initrd.img`、网卡启动程序 `pxelinux.0`）

创建 `pxelinux.cfg`/配置目录，在此目录下建立默认引导文件 `default`

3.2 方案

TFTP, Trivial File Transfer Protocol: 简单文件传输协议，通过 UDP 69 端口提供小文件的传输服务，默认应将资源部署到 `/var/lib/tftpboot` 目录下，不支持认证和目录访问等复杂 FTP 操作。

网卡启动程序 `pxelinux.0` 由软件包 `syslinux` 提供。

PXE 安装用的内核及初始化文件可从 RHEL7 的光盘目录 `/images/pxeboot/` 下提取。

PXE 启动配置相关资料可参考 RHEL7 的光盘目录 `/isolinux/`，其中包括图形支持模块 `vesamenu.c32`、背景图片 `spash.png`、菜单配置文件 `isolinux.cfg`（使用时改名为 `default`）。

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：快速构建 TFTP 服务器

1) 安装 `tftp-server` 软件包

```
[root@svr7 ~]# yum -y install tftp-server
```

.. ..

2) 启动系统服务 `tftp`，并设置开机自启

```
[root@svr7 ~]# systemctl restart tftp
```

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

Created symlink from `/etc/systemd/system/sockets.target.wants/tftp.socket` to `/usr/lib/systemd/system/tftp.socket`.

步骤二：部署启动文件

1) 拷贝 `pxelinux.0` 程序，部署到 TFTP 目录

在软件包 `syslinux` 提供的目录下找到 `pxelinux.0` 程序

```
[root@svr7 ~]# yum -y install syslinux
```

```
[root@svr7 ~]# rpm -ql syslinux | grep pxelinux.0
```

```
/usr/share/syslinux/gpxelinux.0
```

```
/usr/share/syslinux/pxelinux.0
```

将其拷贝到 `/var/lib/tftpboot/` 目录下，确认部署结果：

```
[root@svr7 ~]# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/
```

```
[root@svr7 ~]# ls /var/lib/tftpboot/
```

```
pxelinux.0
```

2) 拷贝引导装机的内核、初始镜像，部署到 TFTP 目录

当文件较多时，可以在 TFTP 目录下创建子目录：

```
[root@svr7 ~]# mkdir /var/lib/tftpboot/rhel7
```

再通过 RHEL7 光盘目录找到 PXE 版内核 vmlinuz、初始镜像 initrd.img，将其拷贝到上述子目录：

```
[root@svr7 ~]# cd /var/lib/tftpboot/rhel7/
```

```
[root@svr7 rhel7]# wget http://192.168.4.254/rh7dvd/isolinux/vmlinuz
```

```
[root@svr7 rhel7]# wget http://192.168.4.254/rh7dvd/isolinux/initrd.img
```

//下载内核、初始化文件

确认部署结果：

```
[root@svr7 pxeboot]# ls -R /var/lib/tftpboot/
```

```
/var/lib/tftpboot/:
```

```
pxelinux.0 rhel7
```

```
/var/lib/tftpboot/rhel7:
```

```
initrd.img vmlinuz
```

步骤三：配置启动菜单

1) 创建配置目录

```
[root@svr7 ~]# mkdir /var/lib/tftpboot/pxelinux.cfg
```

2) 以光盘中的 isolinux 目录为模板，拷贝必要的文件

```
[root@svr7 ~]# cd /var/lib/tftpboot/
```

```
[root@svr7 tftpboot]#
```

```
wget
```

```
http://192.168.4.254/rh7dvd/isolinux/vesamenu.c32
```

//提供图形支持

```
[root@svr7 tftpboot]# wget http://192.168.4.254/rh7dvd/isolinux/splash.png
```

//准备背景图片

```
[root@svr7 tftpboot]# wget -O
```

```
pxelinux.cfg/default
```

```
http://192.168.4.254/rh7dvd/isolinux/isolinux.cfg
```

//建立菜单配置

```
[root@svr7 isolinux]# ls -R /var/lib/tftpboot/
```

//确认部署结果

```
/var/lib/tftpboot/:
```

```
pxelinux.0 pxelinux.cfg rhel7 splash.png vesamenu.c32
```

```
/var/lib/tftpboot/pxelinux.cfg:
```

```
default
```

```
/var/lib/tftpboot/rhel7:
```

```
initrd.img vmlinuz
```

3) 调整启动参数

```
[root@svr7 ~]# vim /var/lib/tftpboot/pxelinux.cfg/default
```

```
default vesamenu.c32
```

//默认交给图形模块处理

```
timeout 600
```

//选择限时为 60 秒（单位 1/10

秒）

```
.. ..
```

```
menu title PXE Installation Server
```

//启动菜单标题信息

```
.. ..
```

```
label linux
```

//菜单项标签

```

    menu label ^Install Red Hat Enterprise Linux 7
    kernel rhel7/vmlinuz                //内核的位置
    append                               initrd=rhel7/initrd.img
inst.stage2=http://192.168.4.254/rh7dvd
                                     //初始镜像、安装源位置
label rescue
    menu label ^Rescue a Red Hat Enterprise Linux system
    kernel rhel7/vmlinuz
    append initrd=rhel7/initrd.img inst.stage2=http://192.168.4.254/rh7dvd
rescue
label local                            //从硬盘启动
    menu default                        //默认启动方式
    menu label Boot from ^local drive
    localboot 0xffff
menu end

```

步骤四：访问 TFTP 服务端确保可用

1) 在 pc207 上安装 tftp 命令工具

```
[root@pc207 ~]# yum -y install tftp
```

.. ..

2) 在 pc207 上访问 svr7 上的 TFTP 服务端，下载文件测试

```
[root@pc207 ~]# tftp 192.168.4.7 -c get pxelinux.0
```

```
[root@pc207 ~]# ls -lh pxelinux.0 //检查下载结果
```

```
-rw-r--r--. 1 root root 27K 1月 13 15:48 pxelinux.0
```

4 案例 4：验证 PXE 网络装机

4.1 问题

沿用案例三，本例要求新建一台虚拟机（内存 1G、硬盘 20G），完成 PXE 网络装机的过程测试：

选择采用 URL 源，自动获取 IP 地址

指定路径 `http://192.168.4.254/rh7dvd`

后续过程与光盘本地安装相同

4.2 方案

认识 PXE 客户机装机的引导环节：

通过 DHCP 配置网卡、获知 TFTP 地址及 PXE 启动文件

从 TFTP 服务器下载 PXE 启动文件

读取启动配置（`pxelinux.cfg/default`）

根据用户选择下载 `vmlinuz` 和 `initrd.img`

内核 `vmlinuz` 运行后，主导安装过程

4.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：新建虚拟机 `pxetest`

新建一台虚拟机，注意将安装方式选择“网络引导（PXE）”（如图-2 所示）。虚拟机的其他设置参考：内存 1G、硬盘 20G；网络类型要与 PXE 服务器相同，比如选用 `private1` 隔离网络。

图-2

注意：如果是生产环境中已经有系统的客户机，则需要调整 BIOS 设置，将网络引导作为第一启动设备。

步骤二：启动虚拟机 `pxetest`，验证 PXE 网络安装过程

1) 确认启动界面

客户机通过 PXE 引导以后，可以获得服务端提供的配置界面（如图-3 所示）。

图-3

2) 选择第一个菜单项开始安装 RHEL7 操作系统

成功出现 RHEL7 安装程序的欢迎界面，根据提示选择中文（如图-4 所示）后继续。

图-4

3) 下一步需手动选择安装源

指定位于 `http://192.168.4.254/rh7dvd` 的软件仓库（如图-5 所示），软件选择、安装位置等设置根据需要指定，确认后继续。

图-5

后续过程与正常的光盘安装类似，不再赘述。

顺利操作到这里说明 PXE 网络引导安装的目标已经实现了。

5 案例 5：PXE+kickstart 自动装机

5.1 问题

本例要求在 PXE 服务器上为 RHEL7 客户机准备 `ks` 应答文件，完成下列任务：

实现全自动的安装及配置

能够自动配好 YUM 仓库

然后在客户机上验证 PXE+kickstart 全自动装机过程：

再次将测试客户机从 PXE 启动并安装

完成后，重启客户机并验证结果

5.2 方案

使用图形配置工具 `system-config-kickstart` 来生成应答文件

5.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：准备应答文件

找一台 RHEL7 系统的模板虚拟机，使用配置工具生成应答文件。

1) 在模板机上安装 `system-config-kickstart` 软件包

```
[root@svr7 ~]# yum -y install system-config-kickstart
```

.. ..

2) 调整模板机的 yum 仓库设置

清理掉无关的 yum 源，只保留为客户机安装 RHEL7 系统所必要的 yum 源，并且将源的 ID 修改为 development。

```
[root@svr7 ~]# vim /etc/yum.repos.d/rh7dvd.repo
```

```
[development]
```

```
name = Red Hat Enterprise Linux 7
```

```
baseurl = http://192.168.4.254/rh7dvd
```

```
gpgcheck = 0
```

3) 运行 system-config-kickstart 工具，创建应答文件

在支持图形程序的环境运行 system-config-kickstart，即可打开该配置工具（如图-6 所示）。

图-6

通过“文件”菜单打开/root/anaconda-ks.cfg 文件，作为应答配置模板，这样可以节省很多时间（如图-7 所示）。根据需要确认默认语言、时区，设置根口令、磁盘分区规划等基本信息。

图-7

在安装方法部分，选择“执行新安装”，并正确设置 HTTP 安装源的访问地址信息（如图-8 所示）。

图-8

在分区信息部分，选择清除主引导记录、删除所有现存分区、初始化磁盘标签，并手动添加 /boot 200MB、SWAP 分区 2000MB、/分区 所有剩余空间（如图-9 所示）。

图-9

在网络配置部分，确认已添加第一块网卡，设为 DHCP 自动获取（如图-10 所示）。

图-10

在防火墙配置部分，禁用 SELinux、禁用防火墙（如图-11 所示）。

图-11

在软件包选择部分，根据客户机的实际需要定制。比如若要使用图形桌面环境，建议将 GNOME 相关的包勾选上（如图-12 所示）。

图-12

在安装后脚本部分，添加经验证的正确可执行语句，使客户机装好系统后能够自动配好 YUM 源（如图-13 所示）。

图-13

确认上述调整后，通过“文件”菜单将其保存为/root/ks-rhel7.cfg。

4) 应答文件微调整

删除掉随模板机配置残留的不可用仓库记录（比如以 `repo --name` 开头的 `Server-HighAvailability`、`Server-ResilientStorage`、`autopart` 等行）

```
[root@svr7 ~]# vim /root/ks-rhel7.cfg
install                                     //安装基本信息设置
xconfig --startxonboot
keyboard --vckeymap=cn --xlayouts='cn'
rootpw --iscrypted $1$.48kBNVL$e.Ym0L/RzkJonYwbg9Brq1
timezone Asia/Shanghai
url --url="http://192.168.4.254/rh7dvd"      //安装源设置
lang zh_CN
firewall --disabled
#repo                                     --name="Server-HighAvailability"
--baseurl=file:///run/install/repo/addons/HighAvailability
#repo                                     --name="Server-ResilientStorage"
--baseurl=file:///run/install/repo/addons/ResilientStorage
#autopart --type=lvm                       //分区设置
zerombr
clearpart --all --initlabel
part /boot --fstype="xfs" --size=200
part swap --fstype="swap" --size=2000
part / --fstype="xfs" --grow --size=1
.. ..
%post --interpreter=/bin/bash              //安装后脚本设置
echo '[rhel7]
name = Red Hat Enterprise Linux 7
baseurl = http://192.168.4.254/rh7dvd
gpgcheck = 0' > /etc/yum.repos.d/rhel7.repo
%end
%packages                                  //软件包设置
@^graphical-server-environment
@base
@core
@desktop-debugging
@development
.. ..
initial-setup
initial-setup-gui
-NetworkManager
-NetworkManager-team
.. ..
%end
```

步骤二：部署应答文件

1) 将应答文件部署在客户机可访问的位置

部署并确认文件:

```
[root@room9pc13 ~]# scp root@192.168.4.7:/root/ks-rhel7.cfg /var/www/html/
```

```
[root@room9pc13 ~]# ls -lh /var/www/html/ks-rhel7.cfg //检查部署的文件
```

```
-rw-r--r--. 1 root root 4.5K 1月 13 20:20 /var/www/html/ks-rhel7.cfg
```

在客户端下载应答文件, 确保可访问:

```
[root@pc207 ~]# wget http://192.168.4.254/ks-rhel7.cfg
```

```
.. ..
```

```
2017-01-13 20:22:19 (183 MB/s) - "ks-rhel7.cfg" 已保存 [4508]
```

```
[root@pc207 ~]# ls -lh ks-rhel7.cfg //检查下载的文件
```

```
-rw-r--r--. 1 root root 4.5K 1月 13 20:22 ks-rhel7.cfg
```

2) 在 PXE 服务器上修改 default 引导配置, 调用应答文件

找到相应的 label 启动项, 在 append 后添加 ks=应答文件地址, 去掉原有的 inst.stage2 参数设置:

```
[root@svr7 ~]# vim /var/lib/tftpboot/pxelinux.cfg/default
```

```
.. ..
```

```
label linux
```

```
    menu label ^Install Red Hat Enterprise Linux 7
```

```
    kernel rhel7/vmlinuz
```

```
    append initrd=rhel7/initrd.img ks=http://192.168.4.254/ks-rhel7.cfg
```

```
.. ..
```

步骤三: 验证 PXE+kickstart 自动应答

1) 新建一台虚拟机裸机, 确认支持 PXE 网卡启动

新建一台虚拟机裸机, 注意以下事项: 将内存设为 1G、硬盘设为 20G; 网络类型要与 pxe-svr 服务器的相同, 比如选择 private1。

2) 启动虚拟机裸机, 验证 PXE 网络安装过程

正常 PXE 引导, 选择第一个启动项回车确认即快速进入全自动安装, 后续过程基本无需人工干预 (如图-14 所示)。

图-14

Top

NSD SERVICES DAY05

案例 1: rsync 基本用法

案例 2: rsync+SSH 同步

案例 3: 使用 inotifywait 工具

案例 4: 配置 Web 镜像同步

1 案例 1: rsync 基本用法

1.1 问题

本例要求掌握远程同步的基本操作，使用 `rsync` 命令完成下列任务：

将目录 `/boot` 同步到目录 `/todir` 下

将目录 `/boot` 下的文档同步到目录 `/todir` 下

在目录 `/boot` 下新增文件 `a.txt`，删除 `/todir` 下的子目录 `grub2`，再次同步使 `/todir` 与 `/boot` 一致

验证 `-a`、`-n`、`-v`、`--delete` 选项的含义

1.2 方案

本地同步操作：

`rsync [选项...] 本地目录 1 本地目录 2`

`rsync [选项...] 本地目录 1/ 本地目录 2`

`rsync` 同步工具的常用选项：

`-n`：测试同步过程，不做实际修改

`--delete`：删除目标文件夹内多余的文档

`-a`：归档模式，相当于 `-rlptgoD`

`-v`：显示详细操作信息

`-z`：传输过程中启用压缩/解压

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：`rsync` 同步基本操作

1) 将目录 `/boot` 同步到目录 `/todir` 下

```
[root@svr7 ~]# ls -l /todir //同步前
```

`ls`：无法访问 `/todir`：没有那个文件或目录

```
[root@svr7 ~]# rsync -a /boot /todir //将目录 1 作为目录 2 的子目录
```

```
[root@svr7 ~]# ls -l /todir //检查同步结果
```

总用量 4

```
dr-xr-xr-x. 4 root root 4096 11 月 30 18:50 boot
```

2) 将目录 `/boot` 下的文档同步到目录 `/todir` 下

```
[root@svr7 ~]# rm -rf /todir //清理掉目录 2
```

```
[root@svr7 ~]# rsync -a /boot/ /todir //将目录 1 下的文档同步到目录
```

2 下

```
[root@svr7 ~]# ls -l /todir //检查同步结果
```

总用量 126708

```
-rw-r--r--. 1 root root 126426 10 月 30 2015 config-3.10.0-327.el7.x86_64
```

```
drwxr-xr-x. 2 root root 4096 11 月 30 18:50 extlinux
```

```
drwx-----. 6 root root 104 12 月 9 09:58 grub2
```

.. ..

3) 同步效果测试

在目录 `/boot` 下新增文件 `a.txt`，删除 `/todir` 下的子目录 `grub2`：

```
[root@svr7 ~]# touch /boot/a.txt
```

```
[root@svr7 ~]# rm -rf /todir/grub2/
```

现在目录 `/boot` 和 `/todir` 目录下的内容已经不一致了：

```
[root@svr7 ~]# ls -ld /boot/a.txt /tdir/a.txt
ls: 无法访问/tdir/a.txt: 没有那个文件或目录
-rw-r--r--. 1 root root 0 1 月 11 21:09 /boot/a.txt
[root@svr7 ~]# ls -ld /boot/grub2 /tdir/grub2
ls: 无法访问/tdir/grub2: 没有那个文件或目录
drwx-----. 6 root root 104 12 月 9 09:58 /boot/grub2
再次同步使/tdir 与/boot 一致:
[root@svr7 ~]# rsync -a /boot/ /tdir/
确认同步结果:
[root@svr7 ~]# ls -ld /boot/a.txt /tdir/a.txt
-rw-r--r--. 1 root root 0 1 月 11 21:09 /boot/a.txt
-rw-r--r--. 1 root root 0 1 月 11 21:09 /tdir/a.txt
[root@svr7 ~]# ls -ld /boot/grub2 /tdir/grub2
drwx-----. 6 root root 104 12 月 9 09:58 /boot/grub2
drwx-----. 6 root root 104 12 月 9 09:58 /tdir/grub2
步骤二: 验证 -a、-v、-n、--delete 选项的含义
```

1) 验证-a 选项

当目录 1 包含文件夹时，若缺少-a 或-r 选项则文件夹会被忽略:

```
[root@svr7 ~]# rsync /home /testa
skipping directory home
[root@svr7 ~]# ls -ld /testa
ls: 无法访问/testa: 没有那个文件或目录
添加-a 后才会执行同步:
[root@svr7 ~]# rsync -a /home/ /testa
[root@svr7 ~]# ls -ld /testa
drwxr-xr-x. 4 root root 31 1 月 6 17:33 /testa
```

类似的情况，当目录 1 中的数据出现权限、归属、修改时间等变化时，若文件内容不变默认不会同步，若希望目录 2 也同步这些变化，也需要-a 选项。

2) 验证-v 选项

创建测试目录及文档:

```
[root@svr7 ~]# mkdir /fdir
[root@svr7 ~]# touch /fdir/1.txt
添加-v 选项时，可以看到操作细节信息，比如第一次同步时:
[root@svr7 ~]# rsync -av /fdir/ /tdir
sending incremental file list
created directory /tdir
./
1.txt                                //传输文档列表
sent 82 bytes received 34 bytes 232.00 bytes/sec
total size is 0 speedup is 0.00
在目录/fdir/添加文件 2.txt，再次跟踪同步信息:
[root@svr7 ~]# touch /fdir/2.txt
sending incremental file list
```

```
./
2.txt //传输文档列表
sent 100 bytes received 34 bytes 268.00 bytes/sec
total size is 0 speedup is 0.00
```

确认目录 1 和目录 2 的内容已经一致：

```
[root@svr7 ~]# ls /fdir/ /tdir/
/fdir/:
1.txt 2.txt
/tdir/:
1.txt 2.txt
```

再次跟踪同步信息，已经无需传输文件：

```
[root@svr7 ~]# rsync -av /fdir/ /tdir/
sending incremental file list
sent 58 bytes received 12 bytes 140.00 bytes/sec
total size is 0 speedup is 0.00
```

3) 验证-n 选项

将-n、-v 选项合用，可以模拟同步过程，显示需要做哪些操作（但并不真的同步）。

在目录/fdir 下新建文件 3.txt，测试同步操作：

```
[root@svr7 ~]# touch /fdir/3.txt
[root@svr7 ~]# rsync -avn /fdir/ /tdir/
sending incremental file list
```

```
./
3.txt //提示同步时会传输哪些文件
sent 78 bytes received 18 bytes 192.00 bytes/sec
total size is 0 speedup is 0.00 (DRY RUN)
```

```
[root@svr7 ~]# ls -l /tdir/3.txt //但实际并未真的同步
```

ls: 无法访问/tdir/3.txt: 没有那个文件或目录

去掉-n 选项才会真正同步：

```
[root@svr7 ~]# rsync -av /fdir/ /tdir/
sending incremental file list
```

```
./
3.txt
sent 114 bytes received 34 bytes 296.00 bytes/sec
total size is 0 speedup is 0.00
```

```
[root@svr7 ~]# ls -l /tdir/3.txt
-rw-r--r--. 1 root root 0 1 月 11 21:46 /tdir/3.txt
```

4) 验证--delete 选项

rsync 同步操作默认只是将目录 1 的数据同步到目录 2，但如果目录 2 存在多余的文件却并不会去除，除非添加一delete 选项。

在目录/fdir、/tdir 已经完成同步后，删除/tdir/2.txt 文件，再次同步：

```
[root@svr7 ~]# rm -rf /fdir/2.txt
[root@svr7 ~]# rsync -a /fdir/ /tdir/
检查发现目标文件夹/tdir 下的 2.txt 文件还在：
[root@svr7 ~]# ls /fdir/ /tdir/
```

```
/fdir/:  
1.txt 3.txt
```

```
/tdir/:  
1.txt 2.txt 3.txt
```

这种情况下添加--delete 选项再次执行同步，两个目录的内容就一致了：

```
[root@svr7 ~]# rsync -a --delete /fdir/ /tdir/
```

```
[root@svr7 ~]# ls /fdir/ /tdir/
```

```
/fdir/:  
1.txt 3.txt
```

```
/tdir/:  
1.txt 3.txt
```

2 案例 2: rsync+SSH 同步

2.1 问题

本例要求掌握 rsync 与远程 SSH 资源的同步操作，使用 rsync 命令访问远程主机 svr7，完成下列任务：

查看远程主机的 / 目录下有哪些子目录

从远程主机下载 /etc/passwd 文件到当前目录

将远程主机的 /boot/ 目录同步为本地的 /fromssh

将本机的 /etc 目录同步到远程主机的 /opt/下

2.2 方案

列出 SSH 服务端资源

rsync user@host:远程目录/

rsync+SSH 远程同步操作：

rsync [...] user@host:远程目录 本地目录

rsync [...] 本地目录 user@host:远程目录

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：列出远程主机的 SSH 资源

查看远程主机 svr7 的/目录下有哪些子目录：

```
[root@pc207 ~]# rsync root@192.168.4.7:/
```

```
root@192.168.4.7's password:
```

//验证对方的密码

```
dr-xr-xr-x      4096 2016/12/15 10:39:34 .  
lrwxrwxrwx       7 2016/12/07 09:21:50 bin  
lrwxrwxrwx       7 2016/12/07 09:21:50 lib  
lrwxrwxrwx       9 2016/12/07 09:21:50 lib64  
lrwxrwxrwx       8 2016/12/07 09:21:50/sbin  
dr-xr-xr-x      4096 2016/12/07 11:25:29 boot  
drwxr-xr-x       6 2016/12/07 09:21:14 data  
drwxr-xr-x     3200 2016/12/15 10:46:15 dev  
drwxr-xr-x     8192 2016/12/20 17:01:02 etc
```

步骤二：rsync+SSH 同步操作

1) 从远程主机 svr7 下载/etc/passwd 文件到当前目录

```
[root@pc207 ~]# rsync root@192.168.4.7:/etc/passwd ./
root@192.168.4.7's password: //验证对方的密码
[root@pc207 ~]# cat passwd //检查同步结果
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
.. ..
```

2) 将远程主机 svr7 的/boot/目录同步为本地的/fromssh

```
[root@pc207 ~]# rsync -a root@192.168.4.7:/boot/ /fromssh
root@192.168.4.7's password: //验证对方的密码
[root@pc207 ~]# ls /fromssh/ //检查同步结果
config-3.10.0-327.el7.x86_64
extlinux
grub2
initramfs-0-rescue-a19921505cc7e19d20dfcd5cea7d8aa2.img
initramfs-3.10.0-327.el7.x86_64.img
initramfs-3.10.0-327.el7.x86_64kdump.img
.. ..
```

3) 将本机的/etc 目录同步到远程主机 svr7 的/opt/下

确认目录大小:

```
[root@pc207 ~]# du -sh /etc
35M /etc
```

上行同步到远程主机 svr7 上:

```
[root@pc207 ~]# rsync -a /etc root@192.168.4.7:/opt/
root@192.168.4.7's password:
在远程主机上检查同步结果:
```

```
[root@svr7 ~]# du -sh /opt/etc
35M /opt/etc
```

3 案例 3：使用 inotifywait 工具

3.1 问题

本例要求安装 inotify-tools 工具，并针对文件夹 /opt 启用 inotifywait 监控，完成下列任务：

当此目录下出现新建、修改、更改权限、删除文件等事件时能给出提示
验证上述监控事件的效果

3.2 方案

inotifywait 监控操作：

inotifywait [选项] 目标文件夹

inotifywait 常用命令选项:

-m, 持续监控 (捕获一个事件后不退出)

-r, 递归监控、包括子目录及文件

-q, 减少屏幕输出信息

-e, 指定监视的 modify、move、create、delete、attrib 等事件类别

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一: 安装 inotify-tools 软件包

1) 解包

```
[root@svr7 ~]# tar xf inotify-tools-3.13.tar.gz -C /usr/src/
```

2) 配置

```
[root@svr7 ~]# cd /usr/src/inotify-tools-3.13/
```

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

3) 编译

```
[root@svr7 inotify-tools-3.13]# make
```

4) 安装

```
[root@svr7 inotify-tools-3.13]# make
```

5) 检查安装结果 (inotifywait 程序可用)

```
[root@svr7 ~]# inotifywait --help
```

```
inotifywait 3.13
```

```
Wait for a particular event on a file or set of files.
```

```
Usage: inotifywait [ options ] file1 [ file2 ] [ file3 ] [ ... ]
```

```
Options:
```

```
    -h|--help          Show this help text.
```

```
.. ..
```

步骤二: 测试 inotifywait 监控

1) 开启监控任务, 置入后台

```
[root@svr7 ~]# inotifywait -mrq -e create,modify,move,attrib,delete /opt &  
[1] 55564
```

2) 测试/opt/目录下的新建、修改、改名、更改权限、删除文件等事件的响应消息

观察新建文件时的监控信息:

```
[root@svr7 ~]# touch /opt/a.txt
```

```
/opt/ CREATE a.txt
```

```
/opt/ ATTRIB a.txt
```

观察修改文件内容时的监控信息:

```
[root@svr7 ~]# echo Hello > /opt/a.txt
```

```
[root@svr7 ~]# /opt/ MODIFY a.txt
```

```
/opt/ MODIFY a.txt
```

观察将文件改名时的监控信息:

```
[root@svr7 ~]# mv /opt/a.txt /opt/b.txt
```

```
/opt/ MOVED_FROM a.txt
```

```
/opt/ MOVED_TO b.txt
```

观察修改文件权限时的监控信息：

```
[root@svr7 ~]# chmod 600 /opt/b.txt
```

```
/opt/ ATTRIB b.txt
```

观察删除文件时的监控信息：

```
[root@svr7 ~]# rm -rf /opt/b.txt
```

```
/opt/ DELETE b.txt
```

3) 停止监控任务

```
[root@svr7 ~]# kill -9 %1
```

```
[1]+ 已杀死 inotifywait -mr -e create,modify,move,attrib,delete /opt
```

4 案例 4：配置 Web 镜像同步

4.1 问题

本例要求为两台 Web 服务器 svr7、pc207 的网页文档目录配置镜像同步，主要基于 inotifywait 监控技术实现实时触发操作，需要完成下列任务：

以 svr7 为发起方，原始目录为 /var/www/html/

以 pc207 为同步目标，基于 SSH 免密验证

编写 inotify+rsync 同步脚本，验证实时同步效果

4.2 方案

inotifywait 与 rsync 的结合，主要思路：

```
while inotifywait 监控操作
```

```
do
```

```
    需要执行的 rsync 同步操作
```

```
done
```

4.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：为主机 svr7、pc207 部署同步目录

双方的目录均为/var/www/html/，如果安装了 httpd，此目录会自动出现。

1) 确认 svr7 的目录内容

```
[root@svr7 ~]# yum -y install httpd
```

```
.. ..
```

```
[root@svr7 ~]# ls /var/www/html/
```

```
//向目录下提供一些测试
```

```
文件
```

```
libreoffice
```

2) 确认 pc207 的目录内容

```
[root@pc207 ~]# yum -y install httpd
```

```
.. ..
```

```
[root@pc207 ~]# ls /var/www/html
```

```
//初始目录无数据
```

```
[root@pc207 ~]#
```

步骤二：为 svr7 配置到 pc207 的 SSH 密钥对验证，实现免密码交互

1) 检查当前用户是否已经有可用的 SSH 密钥对文件

```
[root@svr7 ~]# ls ~/.ssh/id_*
```

```
/root/.ssh/id_rsa /root/.ssh/id_rsa.pub
```

如果找不到 id_rsa、id_rsa.pub 密钥对文件，则需要执行下列操作创建：

```
[root@svr7 ~]# ssh-keygen
```

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id_rsa): //按回车，确认存放位置

Enter passphrase (empty for no passphrase): //按回车，确认不要密码

Enter same passphrase again: //再次按回车，确认

Your identification has been saved in /root/.ssh/id_rsa.

Your public key has been saved in /root/.ssh/id_rsa.pub.

The key fingerprint is:

```
00:a7:cb:2d:9d:b8:8a:df:f5:ff:5b:ed:bd:04:10:fe root@svr7
```

The key's randomart image is:

```
+--[ RSA 2048]-----+
```

```
|  . . . . |  
|  + . . |  
|  . . o |  
|  . = o o |  
|  = + S E |  
|  o .. |  
|  . . ...|  
| . o . . ....|  
|..o . ....o. .+|  
+-----+
```

2) 将当前用户的 SSH 公钥部署到远程主机

```
[root@svr7 ~]# ssh-copy-id root@192.168.4.207
```

The authenticity of host '192.168.4.207 (192.168.4.207)' can't be established.

ECDSA key fingerprint is d3:16:2c:9a:9d:91:28:c8:74:9c:af:2d:04:82:c9:66.

Are you sure you want to continue connecting (yes/no)? yes //首次连 yes 确认

root@192.168.4.207's password: //验证对方的密码

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@192.168.4.207'"

and check to make sure that only the key(s) you wanted were added.

3) 验证免密码登录效果

```
[root@svr7 ~]# ssh root@192.168.4.207
```

```
Last login: Fri Jan 13 09:52:08 2017 from 192.168.4.110
```

```
[root@pc207 ~]# //确认已免密码连入远程主机
```

```
[root@pc207 ~]# exit //退出 SSH 登录环境
```

登出

```
Connection to 192.168.4.207 closed.
```

[root@svr7 ~]#

//已反对原客户机

步骤三：编写镜像同步脚本并测试效果

1) 编写脚本文件/root/isync.sh

[root@svr7 ~]# vim /root/isync.sh

```
#!/bin/bash
```

```
FROM_DIR="/var/www/html/"
```

```
RSYNC_CMD="rsync -az --delete $FROM_DIR
```

```
root@192.168.4.207:/var/www/html"
```

```
while inotifywait -rqq -e modify,move,create,delete,attrib $FROM_DIR
```

```
do
```

```
    $RSYNC_CMD
```

```
done &
```

[root@svr7 ~]# chmod +x /root/isync.sh

2) 运行脚本

[root@svr7 ~]# /root/isync.sh

[root@svr7 ~]# pgrep -l inotify

//确认任务在运行

56494 inotifywait

3) 测试同步效果

在 svr7 上向/var/www/html/目录下添加一个测试网页（触发同步）：

[root@svr7 ~]# touch /var/www/html/a.txt

[root@svr7 ~]# ls /var/www/html/

a.txt libreoffice

在 pc207 上检查/var/www/html/目录，内容应该已经与 svr7 上的同名目录一致：

[root@pc207 ~]# ls /var/www/html

a.txt libreoffice

4) 结束测试后，在 svr7 上停止监控任务

[root@svr7 ~]# pkill -9 inotify

[root@svr7 ~]# pgrep -l inotify

//确认已没有监控任务

[root@svr7 ~]#