

任务背景

今天运维接到2张ITSM工单，具体内容为：

1. 测试人员需要运维人员协助安装一台Linux6.5操作系统，配置好基础环境，安装MySQL5.6的最新版本数据库用于测试。
2. 开发人员需要运维人员协助将原有的测试环境中的samba软件升级到最新版本并且安装一个ftp服务的软件。

任务要求

任务1：

1. 最小化安装Centos6.5操作系统，配置好基本的网络环境（静态IP）、主机名为testdb.itcast.cc
2. 安装MySQL数据库，版本为5.6.42
3. 配置好本地yum

任务2：

1. 开发环境的操作系统版本为Centos6.5，最小化安装。
2. 开发环境中原有samba服务软件包：samba-3.6.9。现需要将服该务的软件升级为：samba-3.6.23版本，并且安装vsftpd软件。

课程目标

- 能够根据需求获取到适合当前系统平台的软件包（二进制和源码包）
- 能够使用rpm工具安装rpm包
- 能够根据需求安装源码包
- 理解yum源如何解决依赖关系
- 掌握本地和网络yum源的配置（重点）
- 熟练使用yum工具安装软件包

涉及知识点

- 操作系统最小化安装（旧知识点）
- rpm工具使用（旧知识点）
- yum源配置（本地和网络源）**新知识点**
- yum工具使用 **新知识点**

具体实施

任务1：

1. 安装Linux6.5系统
2. 静态IP和主机名配置

```
NAT模式：自动获取IP，可以访问互联网
[root@node2 network-scripts]# cat ifcfg-eth1
DEVICE=eth1
TYPE=Ethernet
```

```
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=dhcp
```

host-only模式：配置静态IP，不可以访问互联网

```
[root@node2 network-scripts]# cat ifcfg-eth0
DEVICE=eth0
TYPE=Ethernet
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=none
IPADDR=10.1.1.2
NETMASK=255.255.255.0
```

重启网络：

```
[root@node2 network-scripts]# service network restart
```

```
root@node2 network-scripts]# route -n
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.159.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1
10.1.1.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
169.254.0.0	0.0.0.0	255.255.0.0	U	1002	0	0	eth0
169.254.0.0	0.0.0.0	255.255.0.0	U	1003	0	0	eth1
0.0.0.0	192.168.159.2	0.0.0.0	UG	0	0	0	eth1

验证：

```
[root@node2 network-scripts]# ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16384 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:41:a2:fc brd ff:ff:ff:ff:ff:ff
    inet 10.1.1.2/24 brd 10.1.1.255 scope global eth0
    inet6 fe80::20c:29ff:fe41:a2fc/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:41:a2:06 brd ff:ff:ff:ff:ff:ff
    inet 192.168.159.135/24 brd 192.168.159.255 scope global eth1
    inet6 fe80::20c:29ff:fe41:a206/64 scope link
        valid_lft forever preferred_lft forever
```

```
[root@node2 network-scripts]# ping www.baidu.com
```

```
PING www.a.shifen.com (180.149.131.98) 56(84) bytes of data.
64 bytes from 180.149.131.98: icmp_seq=1 ttl=128 time=6.00 ms
64 bytes from 180.149.131.98: icmp_seq=2 ttl=128 time=4.22 ms
64 bytes from 180.149.131.98: icmp_seq=3 ttl=128 time=4.72 ms
```

注意：确保当前虚拟机可以访问外网

配置主机名：

```
[root@node2 ~]# hostname
node2.itcast.cc
[root@node2 ~]# hostname testdb.itcast.cc
[root@node2 ~]# vi /etc/sysconfig/network
[root@node2 ~]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=testdb.itcast.cc
```

将IP和主机名——对应起来保存到/etc/hosts文件中：

```
[root@testdb ~]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
[root@testdb ~]# vi /etc/hosts
[root@testdb ~]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.1.1.2 testdb.itcast.cc testdb
```

注意：

前两行内容不要动！！

补充：

1. 关闭防火墙

临时关闭：

```
service iptables stop
```

开机不要自启动：

```
chkconfig iptables off
```

```
# chkconfig --list|grep iptables    查看是否关闭
```

```
iptables          0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

2. 关闭selinux

三种模式：

enforcing：强制模式

permissive：警告模式

disabled：关闭

临时放到警告模式：

```
[root@testdb ~]# setenforce 0
```

查看状态：

```
[root@testdb ~]# getenforce
```

Permissive

永久关闭：

```
[root@testdb ~]# vi /etc/selinux/config
```

```
# 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=disabled    改为disabled
```

...

注意：关闭需要重启系统才生效

测试验证:

```
[root@testdb ~]# getenforce
Disabled
[root@testdb ~]# chkconfig --list|grep iptables
iptables          0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

3. 安装MySQL数据库

思路:

1. 优先到系统的光盘里找 (没有)
2. 软件的官方网站 (找到)
3. 安装
4. 确保成功安装

将下载好的软件包拷贝到虚拟机中的/soft目录

```
[root@testdb ~]# mkdir /soft
[root@testdb ~]# cd /soft/
[root@testdb soft]# ls
MySQL-5.6.41-1.el6.x86_64.rpm-bundle.tar
```

步骤:

1. 解压软件包

```
[root@testdb soft]# tar xf MySQL-5.6.41-1.el6.x86_64.rpm-bundle.tar
[root@testdb soft]# ls
MySQL-5.6.41-1.el6.x86_64.rpm-bundle.tar  MySQL-server-5.6.41-1.el6.x86_64.rpm
MySQL-client-5.6.41-1.el6.x86_64.rpm      MySQL-shared-5.6.41-1.el6.x86_64.rpm
MySQL-devel-5.6.41-1.el6.x86_64.rpm       MySQL-shared-compat-5.6.41-1.el6.x86_64.rpm
MySQL-embedded-5.6.41-1.el6.x86_64.rpm    MySQL-test-5.6.41-1.el6.x86_64.rpm
```

2. 安装软件包

```
[root@testdb soft]# rpm -ivh MySQL-server-5.6.41-1.el6.x86_64.rpm MySQL-client-5.6.41-1.el6.x86_64.rpm
warning: MySQL-server-5.6.41-1.el6.x86_64.rpm: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY
error: Failed dependencies:
    libaio.so.1()(64bit) is needed by MySQL-server-5.6.41-1.el6.x86_64
    libaio.so.1(LIBAIO_0.1)(64bit) is needed by MySQL-server-5.6.41-1.el6.x86_64
    libaio.so.1(LIBAIO_0.4)(64bit) is needed by MySQL-server-5.6.41-1.el6.x86_64
    libnuma.so.1()(64bit) is needed by MySQL-server-5.6.41-1.el6.x86_64
    libnuma.so.1(libnuma_1.1)(64bit) is needed by MySQL-server-5.6.41-1.el6.x86_64
    libnuma.so.1(libnuma_1.2)(64bit) is needed by MySQL-server-5.6.41-1.el6.x86_64
```

遇到问题:

有依赖

1. 如何知道依赖哪些软件包?

- 1) 通过网站去查找 rpm.pbone.net
- 2) 通过到光盘里去找 按照libaio*去查找

```
[root@testdb soft]# mount /dev/sr0 /mnt
mount: block device /dev/sr0 is write-protected, mounting read-only
[root@testdb soft]# df -h
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/vg01-lv_root 18G   1.8G   15G  11% /
tmpfs                   491M     0   491M   0% /dev/shm
```

```
/dev/sda1          291M   33M  244M  12% /boot
/dev/sr0            4.2G   4.2G    0 100% /mnt
```

```
[root@testdb soft]# ls /mnt/Packages/libaio-*
/mnt/Packages/libaio-0.3.107-10.el6.i686.rpm      /mnt/Packages/libaio-devel-0.3.107-
10.el6.i686.rpm
/mnt/Packages/libaio-0.3.107-10.el6.x86_64.rpm    /mnt/Packages/libaio-devel-0.3.107-
10.el6.x86_64.rpm
```

以上包可以在光盘里直接找到，直接安装即可

```
[root@testdb soft]# ls /mnt/Packages/libnuma-*
ls: cannot access /mnt/Packages/libnuma-*: No such file or directory
[root@testdb soft]# ls /mnt/Packages/numa-*
ls: cannot access /mnt/Packages/numa-*: No such file or directory
[root@testdb soft]# ls /mnt/Packages/numa*
/mnt/Packages/numactl-2.0.7-8.el6.i686.rpm      /mnt/Packages/numactl-devel-2.0.7-
8.el6.x86_64.rpm
/mnt/Packages/numactl-2.0.7-8.el6.x86_64.rpm    /mnt/Packages/numad-0.5-
9.20130814git.el6.x86_64.rpm
/mnt/Packages/numactl-devel-2.0.7-8.el6.i686.rpm
```

以上的包不确定是不是，可以先安装然后再去确认是不是想要的

2. 找到依赖关系后再去手动一个一个去解决

解决方案：

综上所述，选择一个比较好的解决方案，那就是配置yum源来安装mysql数据库。

3. 通过网络yum源解决问题

1) 下载rpm包

```
mysql-community-release-el6.rpm
```

2) 安装

```
[root@testdb soft]# rpm -ivh mysql-community-release-el6.rpm
warning: mysql-community-release-el6.rpm: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY
Preparing... ##### [100%]
1:mysql-community-release##### [100%]
```

```
[root@testdb soft]# cd /etc/yum.repos.d/
```

```
[root@testdb yum.repos.d]# ls
```

```
local.repo  mysql-community.repo  mysql-community-source.repo
```

3) 删除没用的文件

```
[root@testdb yum.repos.d]# mv mysql-community-source.repo /opt
```

4) 查看需要的软件包

```
yum list|grep mysql-community-server
```

5) 安装相应的软件包

```
yum -y install mysql-community-server
```

yum源安装软件遇到其他问题：

```
[root@testdb yum.repos.d]# yum -y install vim
Loaded plugins: fastestmirror
```

```
Repository mysql-connectors-community is listed more than once in the configuration
Repository mysql-tools-community is listed more than once in the configuration
Repository mysql55-community is listed more than once in the configuration
Repository mysql56-community is listed more than once in the configuration
Repository mysql57-community is listed more than once in the configuration
Existing lock /var/run/yum.pid: another copy is running as pid 2027.
Another app is currently holding the yum lock; waiting for it to exit...
```

The other application is: yum

Memory : 58 M RSS (350 MB VSZ)

Started: Sun Nov 4 15:22:13 2018 - 03:34 ago

State : Sleeping, pid: 2027

原因：其他进程正在占用

解决：

```
[root@testdb yum.repos.d]# rm -f /var/run/yum.pid
```

第二种配置mysql网络源的方式

```
[root@testdb yum.repos.d]# vim mysql.repo
```

```
[mysql56-community]
```

```
name=MySQL 5.6 Community Server
```

```
baseurl=http://repo.mysql.com/yum/mysql-5.6-community/el/6/$basearch/
```

```
enabled=1
```

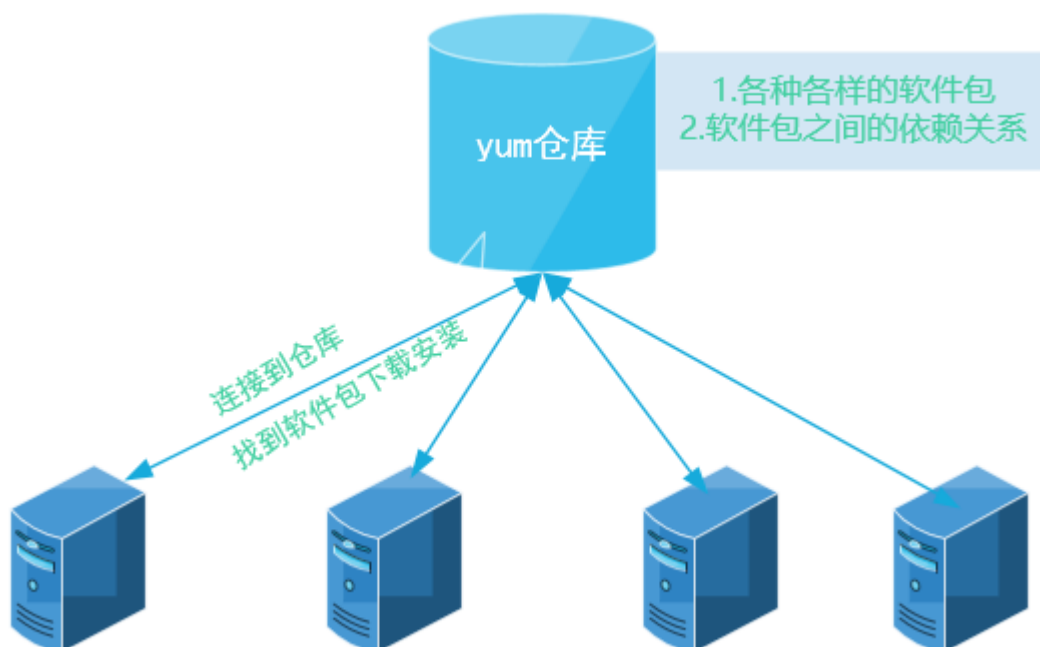
```
gpgcheck=0
```

注意：前提是可以上外网

4. yum源配置(重点)

理论储备

yum源的作用：软件包管理器，类似360的软件管家。



yum源的好处：解决软件包之间的依赖关系，提高运维人员的工作效率。

常见的yum源分类：

- **本地yum源**—yum仓库在本地（系统光盘/镜像文件）
- **网络yum源**—yum仓库不在本地（远程）
 - 国内较知名的网络源（aliyun源，163源，sohu源，知名大学开源镜像等）
 - 国外较知名的网络源（centos源、redhat源、扩展¹源等）
 - 特定软件相关的网络源（Nginx、MySQL、Zabbix等）

本地yum源配置

思路：

1. 本机需要挂载光盘或者镜像文件（仓库在本机）
2. 通过修改配置文件告诉yum工具去哪个仓库里找软件包

步骤：

1. 挂载光盘到本机

注意：如果前面已经挂载成功，则省略

```
[root@testdb soft]# mount -o ro /dev/sr0 /mnt
```

```
[root@testdb soft]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/VolGroup-lv_root	18G	3.2G	14G	20%	/
tmpfs	491M	0	491M	0%	/dev/shm
/dev/sda1	485M	33M	427M	8%	/boot
/dev/sr0	4.2G	4.2G	0	100%	/mnt

2. 修改配置文件告诉yum工具到哪个仓库里找

```
[root@testdb ~]# cd /etc/yum.repos.d/
```

```
[root@testdb yum.repos.d]# ll
```

```
total 16
```

```
-rw-r--r--. 1 root root 1926 Nov 27 2013 CentOS-Base.repo
-rw-r--r--. 1 root root 638 Nov 27 2013 CentOS-Debuginfo.repo
-rw-r--r--. 1 root root 630 Nov 27 2013 CentOS-Media.repo
-rw-r--r--. 1 root root 3664 Nov 27 2013 CentOS-Vault.repo
```

//备份默认的yum源文件

```
[root@testdb yum.repos.d]# mkdir backup
```

```
[root@testdb yum.repos.d]# mv CentOS-* backup/
```

```
[root@testdb yum.repos.d]# ls
```

```
backup
```

```
[root@testdb yum.repos.d]# pwd
```

```
/etc/yum.repos.d
```

//创建一个自定义文件，文件名必须以.repo结尾

```
[root@testdb yum.repos.d]# vi local.repo
```

```
[local]                //仓库的名字，不要有特殊符号（自定义）
```

```
name=local yum         //仓库描述
```

```
baseurl=file:///mnt     //指定yum仓库的路径（重要）；file:///表示本地仓库；ftp:///或者http://
```

```
baseurl=file:///dev/sr0 //错误的写法
```

```
enabled=1              //启用仓库，1表示启用；0表示不启用
```

```
gpgcheck=0             //不用校验软件包的签名，0表示不校验；1表示校验
```

3. 清空yum缓存并创建缓存

```
[root@testdb yum.repos.d]# yum clean all
```

```
Loaded plugins: fastestmirror
```

```
Cleaning repos: local
Cleaning up Everything
[root@testdb yum.repos.d]# yum makecache
Loaded plugins: fastestmirror
Determining fastest mirrors
local                | 4.0 kB    00:00 ...
local/group_gz       | 220 kB    00:00 ...
local/filelists_db    | 5.8 MB    00:00 ...
local/primary_db     | 4.4 MB    00:00 ...
local/other_db        | 2.7 MB    00:00 ...
Metadata Cache Created
```

测试验证:

```
[root@testdb yum.repos.d]# yum repolist
[root@testdb yum.repos.d]# yum list
```

网络yum配置

```
[root@testdb yum.repos.d]# pwd
/etc/yum.repos.d
[root@testdb yum.repos.d]# cat 163.repo
[163]
name=163 network yum
baseurl=http://mirrors.163.com/centos/6/os/x86_64/
enabled=1
gpgcheck=1
gpgkey=http://mirrors.163.com/centos/6/os/x86_64/RPM-GPG-KEY-CentOS-6

[root@testdb yum.repos.d]# cat aliyun.repo
[aliyun]
name=aliyun
baseurl=https://mirrors.aliyun.com/centos/6/os/x86_64/
enabled=1
gpgcheck=1
gpgkey=https://mirrors.aliyun.com/centos/6/os/x86_64/RPM-GPG-KEY-CentOS-6
```

任务2:

思路:

1. 配置好了本地yum源直接安装vsftpd软件
2. 查看samba软件是否安装, 没安装先安装, 然后再升级

具体操作:

```
1. 安装samba软件
1) 关闭aliyun或者其他网络yum源
[aliyun]
name=aliyun
baseurl=https://mirrors.aliyun.com/centos/6/os/x86_64/
```



```

enabled=0      //0表示关闭仓库
gpgcheck=1
gpgkey=https://mirrors.aliyun.com/centos/6/os/x86_64/RPM-GPG-KEY-CentOS-6

2) 安装samba软件
[root@testdb yum.repos.d]# yum -y install samba
3) 确认成功安装
[root@testdb yum.repos.d]# rpm -q samba
samba-3.6.9-164.el6.x86_64
或者
[root@testdb yum.repos.d]# yum list|grep samba
samba.x86_64                                3.6.9-164.el6                                @local
samba-common.x86_64                        3.6.9-164.el6                                @local
samba-winbind.x86_64                       3.6.9-164.el6                                @local
samba-winbind-clients.x86_64               3.6.9-164.el6                                @local

后面仓库名有@符号说明已经安装成功

升级samba软件：
1) 打开aliyun的网络源
enabled=1      //1表示开启仓库
2) 升级对应的软件包
yum -y update samba

3) 确认升级成功
[root@testdb yum.repos.d]# rpm -q samba
samba-3.6.23-51.el6.x86_64
4) 安装ftp软件
[root@testdb yum.repos.d]# yum -y install vsftpd
5) 确认成功安装
[root@testdb yum.repos.d]# rpm -q vsftpd
vsftpd-2.2.2-24.el6.x86_64
[root@testdb yum.repos.d]# yum list|grep vsftpd
vsftpd.x86_64                                2.2.2-24.el6                                @aliyun

```

扩展总结

1. 二进制包获取方式

1.1 RedHat/Centos光盘或官方网站 <ftp://ftp.redhat.com>

1.2 推荐网站

- www.rpmfind.net
- rpm.pbone.net (可搜索)

1.3 相应软件官方网站

- <http://www.mysql.com>
- <http://nginx.org/packages/>

2. 二进制包如何选择

2.1 选择适合当前系统的版本号：

- 找不到适合的，才去尝试别的系统版本号
- el6兼容el5；el5无法安装 el6

2.2 选择适合cpu的架构：

- x86_64包，只能安装在64位的系统上
- i386,i586,i686的软件包可以安装在32和64位系统上
- noarch表示这个软件包与硬件构架无关，可以通用
- 32位系统不能安装64位包

建议： 建议不要跨大版本号去安装软件包，尽量使用当前版本自带软件包安装

3. 源码包编译安装优缺点

xxx.tar.gz xxx.tar.bz2 xxx.tar.xz

• 优点：

1. 可以在任意平台上编译安装，编译出来的软件包非常适应所在机器。
2. 可以在编译的时候，通过配置，对某些功能进行定义，开启或关闭相应的功能。

• 缺点：

1. 安装麻烦
2. 卸载麻烦
3. 升级麻烦

• 源码安装三部曲：

- 根据需求配置：

```
--prefix=... Directories to put files in /usr/local 软件家目录
--bindir=... $prefix/bin 命令的目录
--etcdir=... $prefix/etc 配置文件的目录
--mandir=... $prefix/share/man man 文档路径
--locale=... $prefix/share/locale 语言编码
...
```

- 编译：

```
make (使用gcc编译器进行编译)
```

- 安装：

```
make install 类似 rpm -ivh
```

源码安装示例：

```
需求：将axel软件安装到/opt/axel目录下
默认的安装路径：/usr/local
[root@testdb soft]# ls
```

axel-1.0a.tar.gz

1. 解压

```
[root@testdb soft]# tar xf axel-1.0a.tar.gz
[root@testdb soft]# ls
axel-1.0a  axel-1.0a.tar.gz
```

2. 进入到解压目录进行安装

```
[root@testdb soft]# cd axel-1.0a
[root@testdb axel-1.0a]# ls
API      axel.h      conf.c      conn.c      CREDITS     ftp.h      http.h      README      tcp.c
axel.1   axelrc.example  conf.h      conn.h      de.po      gui        Makefile    search.c    tcp.h
axel.c   CHANGES    configure   COPYING     ftp.c      http.c     nl.po       search.h    text.c
```

根据需求配置：

```
[root@testdb axel-1.0a]# ./configure --prefix=/opt/axel
```

编译：

```
[root@testdb axel-1.0a]# make
```

...

make -j 2 可以指定cpu的核数来提高效率

安装：

```
[root@testdb axel-1.0a]# make install
```

确认成功安装：

```
[root@testdb axel-1.0a]# ls /opt/axel/
bin  etc  share
[root@testdb axel-1.0a]# ls /opt/axel/bin/
axel
```

使用axel下载文件：

```
[root@testdb axel]# axel http://mirrors.163.com/centos/6/os/x86_64/RPM-GPG-KEY-CentOS-6
-bash: axel: command not found
```

分析可能原因：

1. 命令没有安装
2. 命令安装了但是找不到

原因：命令安装了但是不能找到，原因是不知道去/opt/axel/bin目录里找。

说明：系统中有PATH环境变量，该变量里保存的是命令的路径，只要PATH中有命令的所在路径，就可以找到。

3. 命令敲错了

解决：

```
[root@testdb axel-1.0a]# echo $PATH
/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
```

临时添加到环境变量中：

```
[root@testdb axel-1.0a]# export PATH=$PATH:/opt/axel/bin
[root@testdb axel-1.0a]# echo $PATH
/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin:/opt/axel/bin
```

永久添加到环境变量：

```
[root@testdb axel-1.0a]# vi /etc/profile
```

在文件最后增加如下内容：

```
export PATH=$PATH:/opt/axel/bin
```

重新读取配置文件：

```
[root@testdb axel-1.0a]# source /etc/profile
```

再次下载：成功

```
[root@testdb axel-1.0a]# axel http://mirrors.163.com/centos/6/os/x86_64/RPM-GPG-KEY-CentOS-6
Initializing download: http://mirrors.163.com/centos/6/os/x86_64/RPM-GPG-KEY-CentOS-6
File size: 1706 bytes
Opening output file RPM-GPG-KEY-CentOS-6
Starting download
```

查看axel更详细的帮助：

```
[root@testdb axel-1.0a]# man axel
-bash: man: command not found
```

原因：没有安装man程序

解决：

```
yum -y install man
```

```
[root@testdb axel]# man axel
No manual entry for axel
```

原因：找不到指定的man文档

解决：

```
[root@testdb axel]# vi /etc/man.config
```

...

增加如下内容：

```
MANPATH /opt/axel/share/man
```

源码卸载软件：

进入到解压包目录里：

```
[root@testdb axel-1.0a]# make uninstall //卸载安装
rm -f /opt/axel/bin/axel
rm -f /opt/axel/etc/axelrc
rm -f /opt/axel/share/man/man1/axel.1
[root@testdb axel-1.0a]# make distclean //卸载编译和配置过程
rm -f *.o axel search core *.mo
rm -f Makefile.settings config.h
```

4. 自建yum仓库

思考：什么情况下需要自建yum仓库？

具体步骤：

1. 创建一个目录来保存相应的软件

```
[root@testdb soft]# mkdir /soft
[root@testdb soft]# ls
```

```
esound-libs-0.2.41-3.1.el6.x86_64.rpm xlockmore-5.31-2.el6.x86_64.rpm
2. 使用createrepo扫描该目录下的所有软件
[root@testdb soft]# yum -y install createrepo
[root@testdb soft]# createrepo /soft/
[root@testdb soft]# ls
esound-libs-0.2.41-3.1.el6.x86_64.rpm repodata xlockmore-5.31-2.el6.x86_64.rpm
```

说明:

扫描后多出一个repodata目录, 存放的软件之间的依赖关系

3. 配置本地yum源

```
[root@testdb yum.repos.d]# cat local.repo
[local]
name=local yum
baseurl=file:///mnt
enabled=1
gpgcheck=0

[myself]
name=myself
baseurl=file:///soft
enabled=1
gpgcheck=0
```

4. 测试安装锁屏软件

```
[root@testdb yum.repos.d]# yum -y install xlockmore
```

5. 配置yum源核心思想

- 需要有个仓库 (仓库里有软件并且还有软件之间的依赖关系)
 - 如果仓库在本地——>本地仓库 (本地源)
 - 如果仓库在网络上 (外网|内网) ——>网络仓库 (网络源)
- 需要修改/etc/yum.repos.d/目录里以*.repo结尾的文件告诉yum工具去哪个仓库找软件包

```
[repositoryid]                                //仓库名或ID不要有特殊符号
name=Some name for this repository            //仓库描述
baseurl=url://path/to/repository/             //仓库路径
enabled=1                                     //启用仓库
gpgcheck=0                                    //不校验仓库里软件包的签名

[repositoryid]
name=Some name for this repository
baseurl=url://server1/path/to/repository/
        url://server2/path/to/repository/
        url://server3/path/to/repository/
enabled=1
gpgcheck=0
```

6. yum和rpm工具使用

6.1 rpm命令

```
rpm -ivh package
安装
rpm -e package
卸载
rpm -Uvh
升级, 如果已安装老版本, 则升级; 如果没安装, 则直接安装
rpm -Fvh
升级, 如果已安装老版本, 则升级; 如果没安装, 则不安装
rpm -ivh --force
强制安装
rpm --nodeps
忽略依赖关系
rpm -ql
查看已经安装的软件的文件列表
rpm -qlp package.rpm
查看未安装的rpm包里的文件列表
rpm -qa 查看已经安装的所有rpm包
rpm -qd 查看软件的文档列表
rpm -qc 查看软件的配置文件
rpm -qi 查看软件的详细信息
rpm -qf filename
查看文件来自哪个rpm包
rpm --import key_file
导入公钥用于检查rpm文件的签名
rpm -checksig package.rpm
检查rpm包的签名
```

6.2 yum命令

```
# yum install package -y
默认是安装来自仓库里的软件, 指定的是软件名字。多个包空格隔开; -y (取消交互)
# yum install ./xlockmore-5.31-2.el6.x86_64.rpm
或者
# yum localinstall ./xlockmore-5.31-2.el6.x86_64.rpm
安装来自本地指定路径下的rpm包, 而不是来自仓库
# yum remove 或者 erase package
卸载软件包
# yum update
更新仓库里所有比本机已经安装过的软件要的软件
# yum update package
指定升级的软件
# yum search mysql
搜索出所有软件名字或者软件描述包含“mysql”关键字的软件
# yum provides "*libmysqlclient.so*"
找出模块由哪些软件包提供
# yum provides "*xeye*"
搜索一个包含xeye关键字的软件包
# yum clean all
清空之前的yum列表缓存
# yum makecache
创建新的缓存
# yum list
```

列出仓库里的所有软件包

```
# yum repolist
```

列出已配置的软件仓库

```
# yum list|tail
```

查看未安装的软件包

```
# yum list |grep 关键字
```

@代表已经安装成功

```
# yum list installed
```

查看已安装的包

```
# yum grouplist
```

查看包组

```
# yum groupinstall "包组"
```

安装包组

```
# yum groupremove "包组"
```

```
# md5sum +包名
```

直接校验第三方提供的软件包

课堂练习：

安装xlockmore锁屏软件

课后实战

1. 需求1

Centos默认使用自带的国外源，由于网络问题请将其替换为国内的阿里云源，163源，sohu源其中之一

2. 需求2

开发人员需要安装某个软件包（epel源中有），发现现有yum源中没有，需要运维协助配置EPEL源（两种方式搭建EPEL源）

3. 需求3

给开发人员搭建能够提供指定软件包的安装，如Nginx和MySQL的yum源

puppet:

<http://yum.puppetlabs.com/puppetlabs-release-el-6.noarch.rpm>

<http://yum.puppetlabs.com/puppetlabs-release-el-7.noarch.rpm>

<http://yum.puppetlabs.com/puppetlabs-release-fedora-20.noarch.rpm>

Zabbix:

http://repo.zabbix.com/zabbix/2.4/rhel/6/x86_64/zabbix-release-2.4-1.el6.noarch.rpm

http://repo.zabbix.com/zabbix/2.2/rhel/6/x86_64/zabbix-release-2.2-1.el6.noarch.rpm

mysql:

<https://repo.mysql.com/mysql-community-release-el6.rpm>

<https://repo.mysql.com/mysql-community-release-el7.rpm>

1. “红帽系”的操作系统提供额外的软件包，适用于RHEL、CentOS等系统[↗](#)