

18.软件安装

一、linux中软件安装包分类

1、软件包分类

源码包

二进制包

2、源码包

2.1 源码包的概述

什么是源码包？其实就开发完是什么样子就是什么样子，大多数源码包都是c或汇编语言写的。

举例：

我们也可以自己写一个c语言的helloworld程序，然后自己编译，自己执行一下，玩玩儿，注意：c语言的程序后缀是.c

```
#include <stdio.h>

int main (void)
{
    printf ("hello world\n");
}
```

c语言需要c的编译器，就像java语言的jdk，所以要在linux上运行c语言的程序，先要安装工具gcc（这个工具是多语言的集成编译器，里边包含了c语言的编译器），c++语言编写的程序的编译器gcc-c++

```
yum insall -y gcc gcc-c++
```

gcc -c hello.c #-c 生成“.o”头文件。这里会生成 hello.o 头文件，但是不会生成执行文件

gcc -o hello hello.o #-o 生成执行文件，并制定执行文件名。这里生成的 hello 就是可执行文件

./hello #执行 hello 文件

2.2 源码包特点

源码包的优点：

开源，如果有足够的能力，可以修改源代码

软件是本地编译，安装，所以更加适合自己的系统，更加稳定也效率更高，普遍认为源码包安装比二进制包安装效率高5%左右

卸载方便，删除安装目录，伦理上认为没什么残留

源码包有缺点：

安装过程步骤较多，尤其安装较大的软件集合时，容易出现拼写错误

编译过程时间较长，安装比二进制安装时间长

因为是编译安装，安装过程中一旦报错新手很难解决，因为我们安装这些软件绝大多数都不是我们自己开发的

3、二进制包

3.1 二进制包分类，不同发行版本的linux系统中，二进制软件包的格式是不一样的
在 Debian 和 ubuntu 中，二进制软件包叫 DPKG 包。

在Red Hat系列的系统中，讲RPM包。

我们学习的是 CentOS 7.6，所以我们这里学习的是 RPM 包管理系统

RPM 包的优点：

包管理系统简单，只通过几个命令就可以实现包的安装、升级、查询和卸载
安装速度比源码包安装快的多

RPM 包的缺点：

软件包已经经过编译，看不到源代码
依赖性。

RPM简单易用，但是它的依赖关系是最头疼的！常见的依赖关系：

- 1) 树形依赖 a---->b---->c
- 2) 环形依赖 a---->b---->c---->a
- 3) 函数库依赖

安装某个软件，需要某个具体的文件，这个文件没有单独成包，是包含在某一个别的软件包中的。

二、rpm包的安装

1、rpm 包命名规则

/mnt/Packages/zsh-5.0.2-28.el7.x86_64.rpm

zsh -5. 0. 2- 28.
el7. x86_64.rpm

软件名 主版本号 次版本号 修订 release(第几次发布版本) 操作系统版本/软件发行商 软件包是64位包

#修订指的是第几次修改bug。

发布指的是：第几次发布。发布时，可能只是对软件安装的默认参数做了修改，而没有其它改动。

2、rpm 包手工安装命令

2.1 RPM 包具体的安装命令

1) 安装命令

rpm -ivh 包全名（带绝对路径）

选项：

- i install 安装 (install)
- v 显示更详细的信息 (verbose)
- h 显示安装进度 (hash)

--nodeps 不检测依赖性安装，在实际的工作中不准用。不管依赖性，想强行安装。注意：这样不检测依赖性安装的软件基本

是不能使用的，所以不建议这样做，实验中可以玩玩儿，卸载软件的时候可以用一下。

--replacefiles 替换文件安装。

--replacepkgs 替换软件包安装。

--force 强制安装。不管是否已经安装，都重新安装。就是**--replacefiles**和**--replacepkgs**的综合。

作用：找回丢失的重要系统文件，一会儿例子中演示。

--test 测试安装。不会实际安装，只是检测一下依赖性。

--prefix 指定安装路径，这个选项基本不用。如果指定了安装路径，软件没有安装到系统默认路径中的话，系统会找不到这些

安装的软件，rpm包的安装的卸载命令就不能卸载掉软件，卸载就需要你一个文件一个文件的删除。所以 rpm 包我们一般都采用默认路径安装。

2) 依赖的解决办法

A、针对a---->b---->c 这种依赖，先安装c，再安装b，最后安装a

B、针对a---->b---->c---->a 这种依赖

1、强制安装 ----暴力型

使用rpm -i **--force --nodeps** 强制安装。忽略依赖关系。这种方法你可以先装A包，再装B包，再装C包。这样还是有点隐患的，感觉不是很踏实（虽然其实目前没发现什么不好），**安装后使用成功的前提是：你要搞清楚依赖关系，并且把这些包都装好。好处是：不用管它们的具体依赖关系先后顺序。**

2、还可以一次性全装上--- 一网打尽型

可以把依赖的几个包拷出来放在同一个文件夹里 然后 rpm -ivh *.rpm 这样也可以前提也是一个都不能少。

C、遇到函数库依赖，需要知道所缺文件在哪个软件包中，这需要在网站www.rpmfind.net上查询一下在安装。**搞定！**

上面A解决办法都是说的轻松但做起来，要命！太繁琐！B必须

事先知道安装的软件所需要的所有依赖，但是不同的系统，不同的安装模式下的系统，这个依赖都是不同的，是一个不确定的东西，实践起来，也不好使！

D、 使用yum技术安装 --使用服务器方式

yum是一个服务器资源技术。通过在线下载服务器资源的方式，也可以用光盘做为本地的软件包的来源。 缺点:太繁琐。要设置一堆的东西。优点: 设置以后，很方便，需要的大多数资源都可以从服务器/本地yum源上找到。这点感觉很像maven！

用apache为例子，安装一下，演示一下上面的讲的知识点！

apache: 静态的web站点服务器程序，html，不支持jsp动态的页面，服务阶段的时候，apache和tomcat做均衡负载

apache的rpm的名字httpd

apache安装完后，启动安装的软件（服务），以Apache 服务为例：

```
systemctl start httpd      启动
systemctl stop httpd       关闭
systemctl restart httpd    重启
systemctl status httpd     查看状态
systemctl enable httpd.service #开机启动
systemctl disable httpd.service #开机不启动
```

apache的网页存放位置：/var/www/html

apache的配置文件位置：/etc/httpd/conf/httpd.conf

上是rpm安装方式的这些文件的位置，源码安装的位置讲到的时候在说

2.2前面讲了，rpm包安装都是安装到默认安装位置，这些位置大多数在这些地方：

```
/etc/    配置文件安装目录
/usr/bin/ 可执行的命令安装目录
/usr/lib/ 程序所使用的函数库保存位置
/usr/share/doc/ 基本的软件使用手册保存位置
/usr/share/man/ 帮助文件保存位置
```

--force 强制安装来解决重要配置文件丢失的问题示例：

2.3 RPM包升级命令

rpm -Uvh 包全名

选项:

-U (大写) 升级安装, 如果没有安装过, 系统直接安装。如果安装过的版本较旧, 则升级到新版本 (upgrade)

rpm -Fvh 包全名

选项:

-F (大写) 升级安装, 如果没有安装过, 则不会安装。必须安装有较旧版本, 才能升级 (freshen)

2.4 卸载命令 (卸载软件, 不推荐使用yum命令的卸载)

rpm -e 包名 (只需要跟上软件的名称就ok)

选项:

--nodeps 不检查依赖性

-e 卸载

2.5 查询

1) 查询软件包是否安装 (只能查询出rpm包安装的软件, 源码安装的软件是查不了的)

rpm -q 包名

选项:

-q: 查询 (query)

2) 查询系统中的所有安装软件包

rpm -qa

选项:

-a: 所有 (all)

当然, 可以用管道符来查看所需的内容, 比如:

rpm -qa | grep httpd

3) 查询软件包的详细信息

rpm -qi 包名

选项:

-i: 查询软件信息 (information)

也可以查询还没有安装的软件包的详细信息

rpm -qip 包全名

选项:

-p: 查询没有安装的软件包 (package)

4) 查询软件包中的文件列表

可以查询已经安装的软件包中的文件列表和安装的完整目录

```
# rpm -ql 包名
```

选项:

-l: 列出软件包中所有的文件列表和软件所安装的目录 (list)

可以查询还没有安装的软件包中的文件列表和打算安装的位置吗? 答案是可以, 命令格式如下:

```
# rpm -qlp 包全名
```

选项:

-p: 查询没有安装的软件包, 默认会安装到那些目录

5) 查询系统文件属于哪个 RPM 包, 注意: 手工建立的文件是不能查询的, 因为这些文件不是通过 RPM 包安装的, 当然不能反向查询它属于哪个 RPM 包

```
# rpm -qf 系统文件名 (必须跟上绝对路径)
```

选项:

-f: 查询系统文件属于哪个软件包 (file)

6) 查询软件包所依赖的软件包 (想要的, 欲安装a软件包, 与a有依赖关系, 但是还没有安装的软件包)

查询系统中和已经安装的软件包有依赖关系的软件包

```
# rpm -qR 包名, 意义不大
```

选项:

-R: 查询软件包的依赖那些包和文件, 列出所有的, 不能看出那些安装了, 那些没安装

查询没有安装的软件包的依赖性, 如: 查看一下还没有安装的 bind软件包的依赖包

```
# rpm -qRp /mnt/Packages/bind-9.8.2-0.10.rc1.el6.i686.rpm
```

2.6 验证: 验证已经安装在系统中的软件里包含的系统文件, 有没的变化, 看看这些系统文件有没有被人动过, 如果不是自己改, 那就要注意, 是不是中毒了, 或者。。。

。

1) 基本命令

```
# rpm -Va
```

选项:

-Va 校验本机已经安装的所有软件包

```
# rpm -V 已安装的包名
```

选项:

-V 校验指定 RPM 包中的文件 (verify)

rpm -Vf 系统文件名

选项:

-Vf 校验某个系统文件是否被修改

2) 验证举例

rpm -V httpd

没什么显示, 如果改改/etc/httpd/conf/httpd.conf文件

在运行# rpm -V httpd

出现了提示信息:

S. 5. . . . T. c /etc/httpd/conf/httpd.conf

我们来解释下最前面共有 8 个信息内容, 是表示验证内容的。文件名前面的 c 是表示这是个配置文件 (configuration)。最后是文件名。那么验证内容中的 8 个信息的具体内容

如下:

S 文件大小是否改变

M 文件的类型或文件的权限 (rwx) 是否被改变

5 文件 MD5 校验和是否改变 (可以看成文件内容是否改变)

D 设备的主从代码是否改变, linux中设备文件都有一个主设备号和次设备号, 比

如sda

L 文件路径是否改变

U 文件的属主 (所有者) 是否改变

G 文件的属组是否改变

T 文件的修改时间是否改变

apache 配置文件的文件类型是 c, 那么还有哪些文件类型呢?

c 配置文件 (config file)

d 普通文档 (documentation)

g “鬼”文件 (ghost file), 很少见, 就是该文件不应该被这个 RPM 包包含

l 授权文件 (license file)

r 描述文件 (read me)

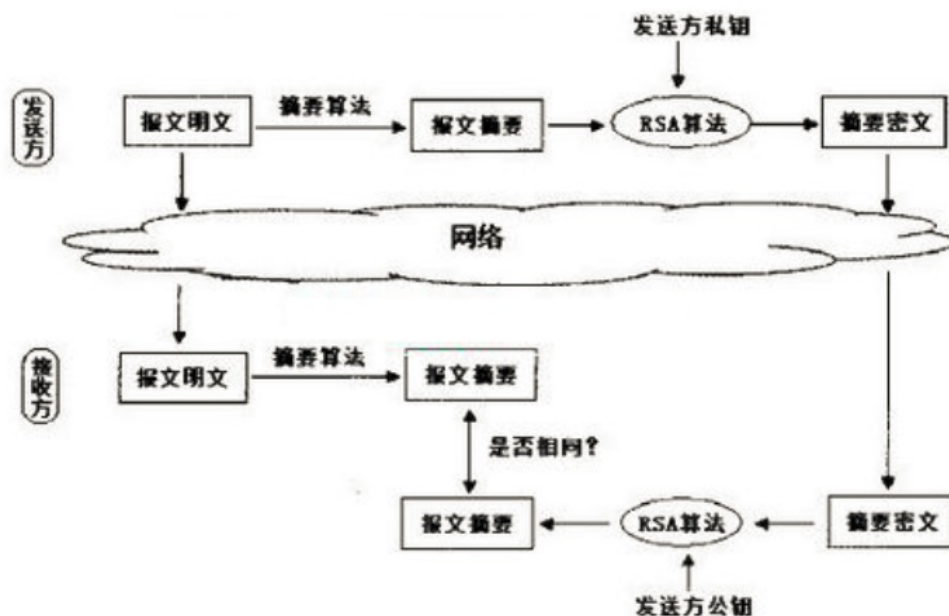
2. 7 数字证书

刚刚的校验方法只能对已经安装的 RPM 包中的文件进行校验, 但是如果 RPM 包本身就被动过手脚,

那么校验就不能解决问题了。我们就必须使用数字证书验证了。数字证书说白了就是一个电子签名, 有签名

表示它是centos官方开发的rpm软件包, 是安全的, 不是别人动过手脚的软件包, 没的, 就有可能是别人动

过手脚的软件包, 至少不是官方发布的。



数字证书的使用，重要在新的服务器器上一定做这几步操作：

首先必须找到原厂的公钥文件，然后进行安装

再安装 RPM 包时， 会去提取 RPM 包中的证书信息， 然后和本机安装的原厂证书进行验证

如果验证通过，则允许安装；如果验证不通过，则不允许安装并警告。

1) 数字证书（原厂的公钥文件）位置

数字证书在安装光盘中就有，安装系统后，我们应该导入这个证书：**RPM-GPG-KEY-CentOS-7**

另外在linux系统中也有，位置在：**/etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7**

2) 数字证书导入 ， 上面两个位置的， 导入那个都是一样的

```
# rpm --import /mnt/RPM-GPG-KEY-CentOS-7
```

选项：

```
--import 导入数字证书
```

导入后，查看有没有导入数字证书命令如下：

```
# rpm -qa | grep gpg-pubkey
```

显示：gpg-pubkey-f4a80eb5-53a7ff4b

2.8 RPM 包中文件的提取

为什么要学习rpm包文件中提出文件：为了还原修改某些重要配置文件后，改错了，改不回来了的解决办法

1) cpio命令

用到的选项：

```
-i: copy-in 模式，还原
```


-v: 显示还原过程

-d: 还原时自动新建目录

2) 提取 RPM 包中文件

rpm2cpio 包全名 | cpio -idv . 要提出的文件（包含绝对路径）

rpm2cpio : 将 rpm 包转换为 cpio 格式的命令

cpio : 是一个标准工具，它用于创建软件档案文件和从档案文件中提取文件

“.” : 这个点，不能省略，而且点后面直接跟上后面的目录，中间不要有空格，表示把提取出的文件放到当前的目录位置，没有点，提出的文件会直接覆盖系统中的文件，不建议的

要提出的文件（包含绝对路径） : 这部分是指rpm包中的要提取的文件的位置和文件名，它和系统中存在的文件位置是一致的

演示一下，提取httpd中的/etc/httpd/conf/httpd.conf文件

```
# rpm2cpio /mnt/Packages/httpd-2.4.6-88.el7.centos.x86_64.rpm | cpio -idv ./etc/httpd/conf/httpd.conf
```

3、rpm 包在线安装（yum 安装）

3.1 yum 源的配置文件解析

yum 源配置文件保存在/etc/yum.repos.d/目录中，文件的扩展名一定是“*.repo”。也就是说，yum 源配置文件只要扩展名是“*.repo”就会生效。

```
# ls /etc/yum.repos.d/
```

```
CentOS-Base.repo  CentOS-Debuginfo.repo  CentOS-Media.repo  CentOS-Vault.repo
CentOS-CR.repo   CentOS-fasttrack.repo  CentOS-Sources.repo
```

这个目录中有 7 个 yum 源配置文件，默认情况下 CentOS-Base.repo 文件生效，这个文件中又有多个具体的yum源。

```
[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os&infra=$infra
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

在 CentOS-Base.repo 文件中有 4 个 yum 源，这里只列出了 base，其他和 base类似。我们解释一下 base 这个源。

[base]: 源的名称，一定要放在[]中，在一个配置文件中，这个名字不能重复。

name: 源说明，可以自己随便写。

mirrorlist: yum 源服务器的镜像站点地址。

baseurl: yum 源服务器的地址。

mirrorlist和baseurl两个配置，只能使用一个，用那个都可以，默认的地址都在国

外，这里可以使用国内的镜像163，aliyun都行

enabled: 此容器是否生效，如果不写或写成 enabled=1 则表示此容器生效，写成 enabled=0则表示此容器不生效。

gpgcheck: 如果为 1 则表示 RPM 的数字证书生效；如果为 0 则表示 RPM 的数字证书不生效。

gpgkey: 数字证书的公钥文件保存位置。不用修改。gpgcheck为1才有用，gpgcheck为0就可以不需要。

```
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

3.2搭建本地光盘 yum 源，前面已经讲过了，这里不重复

3.3 yum 命令

1) 查询

查询 yum 源服务器上/本地yum源上所有可安装的软件包列表。

```
# yum list
```

#查询所有可用软件包列表

查询 yum 源服务器中是否包含某个软件包。

```
# yum list 包名（这个的包名必须写完整）
```

#查询单个软件包

搜索 yum 源服务器上所有和关键字相关的软件包。

```
# yum search 关键字（不是完整的包名，是关键字）
```

#搜索服务器上所有和关键字相关的软件包

yum search 搜索可以用于确定某个命令在哪个相关包当中。比如可以确定“ifconfig”命令需要安装“net-tools”包。

查询指定软件包的信息。

```
# yum info samba
```

#查询 samba 软件包的信息

Available Packages	还没有安装
Name : samba	包名
Arch : i686	适合的硬件平台
Version : 3.5.10	版本
Release : 125.el6	发布版本
Size : 4.9 M	大小
Repo : c6-media	在光盘上

2) 安装

```
# yum -y install 包名      yum install -y httpd
```

选项:

install 安装

-y 自动回答 yes。如果不加-y, 那么每个安装的软件都需要手工回答 yes

3) 升级

```
# yum -y update 包名      #升级指定的软件包
```

选项:

update: 升级

-y: 自动回答 yes

注意: 在进行升级操作时, yum 源服务器中软件包的版本要比本机安装的软件包的版本高。

```
yum -y update
```

升级所有包同时也升级软件和系统内核

默认会升级内核。但有些服务器硬件(特别是组装的机器)在升级内核后,新的内核可能会认不出某些硬件,要重新安装驱动,很麻烦。所以在生产环境中不要輕易的升级内核,除非你确定升级内核后不会出现麻烦的问题。

代码如下:

```
yum -y upgrade
```

只升级所有包,不升级软件和系统内核

4) 卸载

强调一下, 除非你确定卸载的软件的依赖包不会对系统产生影响, 否则不要执行 yum 的卸载,

因为很有可能在卸载软件包的同时卸载的依赖包也是重要的系统文件,这就有可能导致系统崩溃。

卸载命令如下:

```
# yum remove 包名,      yum remove httpd*
```

3.4 yum 组管理命令, 针对某一个具体的应用场景所需要的很多个软件组成软件包, 可以一个命令帮我们安装完(了解)

查询可以安装的软件组

```
# yum grouplist
```

#列出所有可用的软件组列表

查询软件组内包含的软件

```
# yum groupinfo 软件组名
```

#列出软件组中包含的软件

例如：

```
[root@localhost ~]# yum groupinfo 基本网页服务器  
#查询软件组“Web Server”中包含的软件
```

安装软件组

```
[root@localhost ~]# yum groupinstall 软件组名  
#安装指定软件组，组名可以由 grouplist 查询出来
```

例如：

```
[root@localhost ~]# yum groupinstall 基本网页服务器  
#安装网页服务软件组
```

卸载软件组

```
[root@localhost ~]# yum groupremove 软件组名  
#卸载指定软件组
```

三. 源码包安装

1. 注意事项

1.1 应该选择哪种软件包？

如果软件包是给大量客户提供访问，**建议**使用源码包安装，如LAMP环境搭建，因为源码包效率更高。其实现在的集群环境下，大多数RPM包也不慢了，大部分软件包的安装也都是二级制RPM包安装了。

如果软件包是给Linux底层使用，或只给少量客户访问，建议使用rpm包安装，因为rpm包简单。

1.2 源码包是从哪里来的？

rpm包是光盘中直接包含的，所以不需要用户单独下载。而源码包是通过官方网站下载的。

1.3 源码包：大多数情况下须指定安装位置（源码包没有安装数据库，没有删除命令，最好指定安装到一个目录里，删除目录就是删除软件），极少数源码安装指定了位置会报错。

2. 安装过程（核心的过程就是三板斧，编译前的准备，编译，安装）

我们来解释一下源码包安装的具体步骤。**一般情况下源码包里都会有大写INSTALL一样的名字说明文件，里边详细有安装的步骤。**

- （1）下载软件包。
- （2）解压缩。
- （3）进入解压目录。

(4) ./configure 编译前准备

这一步主要有三个作用：

- ① 在安装之前需要检测系统环境是否符合安装要求。
- ② 定义需要的功能选项。“./configure”支持的功能选项较多，可以执行“./configure --help”命令查询其支持的功能。

一般都会通过“./configure --prefix=安装路径”来指定安装路径。

- ③ 把系统环境的检测结果和定义好的功能选项写入 Makefile 文件，后续的编译和安装需要依赖这个文件的内容。

需要注意的是，configure 不是系统命令，而是源码包软件自带的一个脚本程序，所以必须采用“./configure”方式执行（“./”代表在当前目录下）。

(5) make 编译

make 会调用 gcc 编译器，并读取 Makefile 文件中的信息进行系统软件编译。编译的目的就是把源码程序转变为能被 Linux 识别的可执行文件，这些可执行文件保存在当前目录下。编译过程较为耗时，需要有足够的耐心。

(6) make clean: 清空编译内容（非必需步骤，出错后，从新编译前，一般要执行这一步，清理一下）。

如果在“./configure”或“make”编译中报错，那么我们在重新执行命令前一定要记得执行 make clean 命令，它会清空 Makefile 文件或编译产生的“.o”头文件。

(7) make install: 编译安装

这才是真正的安装过程，一般会写清楚程序的安装位置。如果忘记指定安装目录，则可以把这个命令的执行过程保存下来，以备将来删除使用。

3. 删除

源码包没有删除命令，如果需要删除，直接删除安装目录即可。

4. 源码安装出先找不到库文件或软件包的错误后的一般解决办法总结：

yum search 找不到的包或库文件

yum search prce #全部名字查不到，可以查找部分关键字

<http://www.rpmfind.net/> #还可以在网上取查，遇到库文件的时候

no

error

安装Apach来举例演示：

Apache源码安装后的配置文件的位置：

配置文件： /usr/local/apache2/conf/httpd.conf

网页文件： /usr/local/apache2/htdocs/

日志位置： /usr/local/apache2/logs/

