1. 软件包安装
2. 软件包分类

Linux中软件包分为 源码包 和二进制包。

1. 源码包
2. 概述

源码包就是作者直接将源程序发布在网上，我们直接下载源文件，自己编译成二进制程序使用。

1. 优点

开源，可修改源代码；

可自由选择所需要的功能；

编译安装，更加适合自己的系统，更加稳定效率更高；

卸载方便也更加干净；

1. 缺点

安装步骤过多，容易出现错误；

编译过程较长，安装比二进制包安装时间长；

1. 二进制包
2. 分类

DPKG包：由 Debian Linux开发出的包管理机制，主要应用于Debian和Ubuntu中；

RPM包：由Ret Hat开发的包管理系统，安装，升级，卸载非常方便。主要应用于Centos, Fedora等等。

1. 优点

包管理系统简单，只需要几个命令即可实现安装，卸载等等；

安装速度比源码包安装快很多；

1. 缺点

不是开源的，看不到源代码；

功能选择不如源代码灵活；

安装软件包需要依赖于其他的软件包；

1. RPM包依赖

包依赖有三种类型：

树形依赖： a -> b -> c， 像树一样依序依赖不同的包

环形依赖： a -> b -> c -> a，最终依赖链会回到本身的包上；

模块依赖：即依赖不同的函数库.so文件，缺少的函数库文件通常没有单独成包，是包含在一个软件包中的，而如果要知道在哪个包中的话，需要查询网站[www.rpmfind.net](http://www.rpmfind.net)，然后安装对应的包，非常麻烦。

1. 软件包选择

如果安装的服务是给大量客户端使用的，建议使用源码包安装，效率更高；

如果安装的服务是给少量用户使用的，或者本地使用的，建议RPM包安装，因为包管理方便；

1. RPM安装
2. 概述

本机使用光盘安装的Centos6.8，光盘的Packages目录下已经存储了大量的RPM包，可以直接安装使用。在安装时，如果操作的是未安装的软件包，则需要使用包全名，以及使用包的绝对路径；如果操作的是已安装的软件包，则使用包名即可，系统会生成RPM包的数据库（/var/lib/rpm/）。

1. RPM包的命名规则

举例如下：

httpd-2.2.15-15.el6.centos.1.i686.rpm

httpd: 软件包名

2.2.15: 软件版本

15： 软件发布的次数

el6,centos: 支持RedHat6,Centos

i686: 适合的硬件平台。RPM可以在不同的硬件平台上安装，最大化地发挥CPUt性能。如x86\_64(64位CPU)，noarch(没有硬件限制)

rpm: 扩展名

1. RPM包手动安装

手动安装非常地麻烦，需要自己找依赖的包与库函数，艹。

rpm -ivh 包全名

-i: install安装

-v: 显示更详细的信息(verbose)

-h: 打印#显示安装进度(hash)

包全名后面还可以接设置参数，如下所示：

--nodeps: 不检测依赖性安装，这样安装的软件基本上无法使用，不建议；

--force: 直接强制安装，当该服务的某些文件找不到时，此命令可以重新安装找回；

--test: 测试安装，不会实际安装，仅仅检测依赖性；

1. RPM包默认安装位置

/etc/ 配置文件安装目录

/usr/bin: 可执行的命令安装目录

/usr/lib: 程序使用的函数库保存位置

/usr/share/doc: 软件使用手册位置

/usr/share/man/ 帮助文件位置

RPM包建议安装在默认路径中，因为系统中的其他服务会根据默认的路径查找安装的服务，且RPM包管理系统是由卸载命令的，可以根据数据库记录的安装位置来卸载。而源码包安装时必须指定安装路径，因为当要卸载源码包安装的目录时直接删除即可，如果不指定安装目录，服务会安装在系统的各个位置，没有卸载命令，需要一个个地手动删除，非常麻烦。

1. 服务启动命令

service 服务名 start|stop|restart|status

systemctl restart 服务名

1. RPM包升级

rpm -Uvh 包全名

-U：升级安装，如果没有安装过，则直接安装；如果安装的版本太旧，则升级；

-F：升级安装，必须有旧版本，才能够升级；没有旧版本，则不会安装；

1. RPM包卸载

rpm -e 包名

--nodeps 不检查依赖性

-e 卸载

注意，RPM包在卸载时也是有依赖性的，安装时是从下往上依赖，卸载时就是从上往下依赖了。卸载某个包，那么其上的那些包都会被卸载掉。

1. RPM包查询
2. 查询软件包是否安装

rpm -q 包名

-q: query

1. 查询系统中所有已安装软件包

rpm -qa

-a: all

1. 查询软件包的详细信息

rpm -qi 包名

-i: information

还可以查询未安装的软件包：

rpm -qip 包全名

1. 查询软件包的文件列表

rpm -ql 包名

-l: 查询该软件包中所有文件列表和软件所安装的目录：

1. 查询系统文件属于哪个RPM包

只要系统文件才能查询，手工创建的文件不是通过RPM包安装的，无法查询。

rpm -qf 系统文件名

-f: file

1. 查询软件包所依赖的软件包

rpm -qR 包名

-R ： requires

1. 验证

验证RPM包中的文件是否被修改了。

1. 命令

rpm -Va

-Va: 校验本机已经安装的所有软件包

rpm -V 已安装的包名

-V: 校验指定RPM包中文件

rpm -Vf 系统文件名

-Vf: 校验某个系统文件是否被修改

1. 提示信息

当校验某个文件或包时会出现提示信息，通过这些信息告诉我们校验结果。

最前面的8个信息：

S： 文件大小已改变

M: 文件类型或权限已改变

5： 文件MD5校验已改变，即内容

D： 设备的主从代码已改变

L： 文件路径已改变

U： 文件所有者已改变

G： 文件所属组已改变

T： 文件修改时间已改变

文件名前面是文件的类型：

c： 配置文件 config

d: 普通文档 documentation

g: 鬼文件ghost，不应存在在这个RPM包下的文件

l: 授权文件 license

r： 描述文件 read me

1. 数字证书

检验方法根据原有的RPM文件校验，如果原有的RPM文件已经被修改了，仅仅校验是没用的，需要数字证书来校验。数字证书即原厂的检验信息，首先我们安装数字证书，在安装RPM包时，会提取RPM包中的证书信息，和本机已安装的原厂证书进行校验，如果通过则允许安装，否则不允许安装。

数字证书位置，默认在系统中：

/etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

安装数字证书：  
rpm –import /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

查询系统中安装好的数字证书：  
rpm -qa | grep gpg-pubkey

1. RPM包中文件的提取

当我们修改了某个包中的文件后，如果想要恢复该文件的原始内容，仅仅重新安装该RPM包是做不到的，重新安装仅仅适用于误删除文件的情况，可以恢复该文件。这里，我们需要使用到rpm2cpio命令来提取RPM包中的文件，然后覆盖修改的文件即可。

rpm2cpio 包全名 | cpio -idv . 文件绝对路径

rpm2cpio : 将RPM包转换为CPIO格式的命令

cpio: 是一个标准工具，用于创建档案文件和从档案文件中提取文件

注意，最后的文件绝对路径是指要提取的文件所在的路径，可以看到必须放到当前目录下，这是因为提取出的文件不能直接覆盖掉系统中的文件，我们需要对比后手动覆盖或者修改。如下所示：

rpm2cpio /mnt/cdrom/Packages/coreutils-8.4-18.el6.i686.rpm | cpio -idv ./bin/ls

1. RPM包在线安装

yum是从yum源地址的服务器中获取RPM包，首先需要配置yum源的文件。

1. yum源的文件解析

yum源配置文件保存在/etc/yum.repos.d/目录中，文件的扩展名是”\*.repo”，只要扩展名是\*.repo的配置文件都会生效。

该目录下共有5个yum源配置文件，默认情况下CentOS-Base.repo文件是生效的，这是联网的yum源，进入这个文件，可以看到有5个yum源容器，每个容器有不同的参数如下所示：

[base]: 容器的名称，放在[]中

name: 容器的说明

mirrorlist: 镜像站点，可注释

baseurl: yum源服务器的地址，默认是CentOS官方的yum源服务器，可换成国内的源

enabled: 1表示该容器生效，0表示该容器不生效

gpgcheck: 1表示RPM的数字证书生效，0表示不生效

gpgkey: 数字证书的公钥文件保存位置

1. 搭建本地光盘yum源
2. 放入CentOS光盘，挂载光盘到指定位置

mkdir /mnt/cdrom

mount /dev/cdrom /mnt/cdrom

1. 修改其他的yum源配置文件的扩展名，让它们失效，添加.bak后缀即可，对于这种重要的系统文件，不需要时修改后缀名是最好的解决方法。

cd /etc/yum.repos.d

mv CentOS-Base.repo CentOS-Base.repo.bak

1. 修改光盘yum源配置文件CentOS-Media.repo

修改baseurl为光盘挂载的地址：

baseurl=file:///mnt/cdrom

注意，这里的地址有三条斜杠，因为前面两条表示文件协议file，后面一条表示根目录。

注释掉其他的地址，修改enabled为1，使这个yum源配置文件生效。

修改完配置文件后，就可以直接使用yum了。

1. yum命令
2. 查询

查询yum源服务器上所有可安装的软件包列表

yum list

输出的每个RPM包的包名，版本号，以及所在的源容器。

查询yum源服务器中是否包含某个软件包，之前我们讲过在RPM的手动安装中，若这个RPM包未安装，则需要使用包全名，如果已安装，则使用包名。但是这种规则仅仅限于RPM的手动安装中，在yum命令中就直接使用包名即可：

yum list 包名

搜索yum源服务器上所有和关键字有关的软件包，这个命令可以确定某个软件在哪个相关的包中：

yum search 关键字

查询指定软件包的信息：

yum info 包名

1. 安装

yum -y install 包名

-y: 自动回答yes，否则要手动输入

1. 升级

yum -y update 包名

yum -y update

这条命令会升级本机中的所有软件包

1. 卸载

yum的卸载命令会将依赖的包也卸载掉，可能会有风险，因此需要仔细检查依赖性后再删除。

yum remove 包名

1. yum组管理命令

yum组将属于一个类型的RPM包组合到一起，方便我们直接下载。

1. 查询可以安装的软件组

yum grouplist

1. 查询软件组内包含的软件

yum groupinfo 软件组名

1. 安装软件组

yum groupinstall 软件组名

1. 卸载软件组

yum groupremove 软件组名

1. 源码包安装
2. RPM包安装与源码包安装区别

RPM包安装不建议自己指定安装位置，建议安装在默认的位置，因为RPM包有标准的卸载命令，即使各个文件安装在系统的不同位置，也能够卸载干净：

如 安装httpd，各个文件的位置：

配置文件： /etc/httpd/conf/httpd.conf

网页位置： /var/www/html/

日志位置： /var/log/httpd/

启动方法： service httpd restart 或是 /etc/rc.d/init.d/httpd restart ，其实第一个命令就是到第二个命令的目录下找执行脚本的。

源码包安装必须指定安装位置，因为没有安装数据库，没有卸载命令，很难卸载干净。当指定了安装目录后，很容易找到安装的文件。

同样的安装httpd:

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

启动方法： /usr/local/apache2/bin/apache2ctl start

1. 同时使用RPM和源码安装

同一个软件可以使用RPM和源码同时安装，因为安装到不同的位置。但是这样做是没有必要，因为系统相关的端口只有一个，安装多个是浪费资源，建议安装源码的。

1. 安装过程
2. 下载源码包解压缩
3. 进入解压目录
4. 编译前准备 ./configure脚本

这个命令有三个作用：

1. 检测系统环境是否符合要求
2. 定义需要的功能选项
3. 将系统环境的检测结果和定义好的功能选项写入Makefile文件中，后续的编译和安装需要依赖这个文件的内容

一般通过如下参数来指定安装路径：  
./configure --prefix=安装路径

1. make 编译

make会调用gcc编译器，读取Makefile文件中的信息进行系统软件编译，编译就是为了将源码程序转变为能被Linux识别的可执行文件，保存在当前目录下。

1. make clean 清空编译内容

若在./configure 或 make 命令后报错，则需要清空生成的Makefile文件或编译生成的.o文件。

1. make install 编译安装

真正的安装过程，需要写清楚程序的安装位置，便于删除。如果在这一步报错，需要将整个文件夹删除，重新走一遍安装流程。因为编译安装这一步有写入操作，之前的./configure和 make 都只是生成临时文件，用make clean 删除即可。

1. 删除

源码包没有删除命令，直接删除安装目录即可。

1. 补丁
2. 补丁的生成

补丁是两个文件之间的差别，可以通过diff命令生成补丁。

diff 选项 old new

这里的old和new表示旧文件和新文件的绝对路径。

选项：

-a 当做文本处理

-b 忽略空格

-B 忽略空白行

-I 忽略大小写

-N 当比较的是两个目录时，若某个文件只在一个目录中，则该文件在另一个目录中视为空文件

-r 递归比较目录

-u 使用统一的输出格式

如：

diff -Naur /root/test/old.txt /root/test/new.txt > txt.patch

生成了补丁文件txt.patch，在这个文件中，+号表示内容更多的，-表示内容更少的，可以很清楚地看到哪个文件多了哪些内容。

1. 打入补丁

将旧文件按照补丁更新。

patch -pn < 补丁文件

-pn n为数字，表示按照补丁文件中的路径，指定更新文件的路径。因为补丁文件是要打入旧文件中，旧文件的路径为/root/test/old.txt，而补丁文件中记录的旧文件路径也是/root/test/old.txt。如果不做调整，如果当前在/root/test目录下，补丁文件就会打入/root/test/root/test/old.txx文件中。因此，我们需要调整更新文件的目录，-pn的作用就是取消补丁文件路径的n级目录，如-p1，取消一级目录“/”根目录，-p2取消二级目录/root目录，-p3取消/root/test目录，因此补丁文件打入/root/test/old.txt是正确的。

patch -p3 < txt.patch

1. 脚本安装程序

通过脚本安装十分方便，与Windows中类似，运行安装程序，进行简单的功能定制选择，即安装成功。常见的脚本程序以各类硬件的驱动居多。

1. Webmin
2. 介绍

Webmin是一个脚本程序，基于Web的Linux系统管理系统，可在Web上修改Linux的系统文件以及设置，使用Perl语言编写。

1. 安装

下载webmin压缩包，解压缩，进入目录下，执行./setup.sh脚本文件，自动安装后，可以在登录<http://localhost:10000>中查看Linux中的功能。