软件包管理

软件包的组成：

二进制程序(二进制可执行)

配置文件

组织方式：

单文件

将主文件切割并放置到一个目录中

/etc/profile /etc/profile.d/\*

单文件但是内部分为多个片段

库文件(二进制可执行)

静态库：整个库文件都包含进去

动态库：

帮助文件

手册页

文档(/usr/share/doc)

程序的存放位置：

二进制程序：

/bin /sbin 系统自身启动需要的 这就是为什么这几个目录不能放在根外的分区

/usr/bin /usr/sbin 系统运行需要的程序

/usr/local/bin /usr/local/sbin 第三方程序 这个可以放一个独立分区

库文件：

/lib /usr/lib

/usr/local/lib一般找不到

这个文件/etc/ld.so.conf可以配置软件的库位置

/etc.ld.so.conf.d/\*.conf 软件的相应位置写在这里就可以了

配置文件：

/etc

/etc/httpd

/usr/local/etc 这个一般在安装的时候指定一下就可以了

帮助文档：

/usr/share/man

/usr/share/doc

debian:

每个程序的二进制库文件 配置文件 打包成了一个能被管理器管理的格式 deb 软件包管理工具:dpkg管理deb格式的软件

红帽根据这个启发开发了RPM：红帽包管理器

之后被工业标准协会接受：suse

rpm包命名：

name-version-release.arch.rpm

version:2.1.3

major

minor

release

release:第几次制作的rpm包

arch

一个rpm包可以被制作成几个功能相对独立的rpm包

还有子名字命名：

name-subname-version

name-devel-version

自动判断依赖关系，并且自动解决依赖关系关键包管理工具“

yum：yellowdog update modifier

rpm命令：

安装: -i PACKAGE1 PK2 PK3

卸载: -e pkage-name 擦除的意思

查询: -q

升级 -U|-F f:old到new 没有老的不安装 u:update：没有old直接安装

内核一般不应该升级安装 内核允许多版本并存

校验 -V

安装：

-v详细的过程 –iv安装的详细过程 v越多越详细

-h 显示安装进度

--force 强行安装，主要是为了无视冲突

--replacepkgs 重装软件包

依赖关系：**依赖的不一定是包**有可能还是别的文件

-nodeps 忽略依赖关系

**-qa** 全部安装的软件 **|grep name** 过滤下

-qi pkg-name 软件包自身的信息

rpm -ql PKG\_NAME: 用于列出此rpmp安装完成之后在系统上生成哪些文件；

rpm -qf FILE: 用户查询某文件是由哪个rpm包安装生成的；



rpm -qc PKG\_NAME: 查询此rpm包生成的配置文件；

rpm -qd PKG\_NAME: 查询此rpm包生成的文档；

rpm -q --scripts PKG\_NAME： 查询此rpm包内置的脚本

rpm -qpi PKG

rpm -qpl PKG

检验：

检查某rpm包安装生成的文件有没有被非法改变过；



-V PKG\_NAME

说明什么改变了：S.5....T

检验签名：

rpm --import 密钥文件(/etc/pki/rpm-gpg)

rpm -K PKG

--nodigest: 不检验单向的相关内容, sha1, md5

--nosignature: 不检验签名，gpg, dsa

rpm包的数据库(/var/lib/rpm)：

rpm --initdb: 重新初始化丢失了的库

--rebuilddb： 重建，无论事先是否存在；