

软件管理2

rpm包与格式

rpm包是红帽公司的一种软件包打包方式，可向RedHat公司购买订阅来获得官方软件包支持，也可以找系统镜像或网上的开源仓库来获得软件包。

优点为: 安装,更新, 卸载等管理方便

缺点为: 无法自定义软件的安装功能(但打包好的rpm包适合大部分的应用 场景)

软件包名称格式

包名-版本-架构.rpm

如: `zsh-html-5.5.1-6.el8.noarch.rpm`, `zenity-3.28.1-1.el8.x86_64.rpm`

包名分别为 `zsh-html` 与 `zenity`

版本信息分别为 `5.5.1-6.el8` 与 `3.28.1-1.el8` `noarch` 架构表示所有架构可用,

`x86_64` 表示x86体系的64位系统可用

一、封装包 (RPM/DEB) 管理工具

1. RPM包管理 (Red Hat系)

- 安装: `rpm -ivh package.rpm`
- 查询: `rpm -q package` (查看是否安装)。
- 卸载: `rpm -e package`。
- 依赖管理: `yum install package` (自动解决依赖)。

2. DEB包管理 (Debian系)

- 安装: `dpkg -i package.deb`
- 查询: `dpkg -l | grep package`。
- 卸载: `dpkg -r package`。
- 依赖管理: `apt install package` (自动解决依赖)。

二、iso镜像挂载与内容介绍

安装系统的镜像文件里有自带的rpm包，需要先挂载到系统的目录

```
[root@xnha httpd-2.2.15]# mount /dev/sr0 /mnt
mount: /mnt: WARNING: device write-protected, mounted read-only. #
/dev/sr0为光驱设备
[root@xnha httpd-2.2.15]# ls /mnt
AppStream BaseOS EFI images isolinux media.repo TRANS.TBL
(提前将iso镜像放到虚拟光驱中)
```

iso镜像中:

`AppStream` 与 `BaseOS` 里面的 `Packages` 子目录中存放的大量的rpm软件包

`EFI`, `images`, `isolinux` 里都是与系统启动有关的文件

```
[root@xnha httpd-2.2.15]# ls /mnt/AppStream/Packages/ | wc -l
5114
[root@xnha httpd-2.2.15]#
[root@xnha httpd-2.2.15]# ls /mnt/BaseOS/Packages/ | wc -l
1662
```

补充: 如果把iso镜像文件拷贝到linux系统中, 挂载的命令为

```
mount iso镜像文件路径 挂载目录 -o loop
```

三、需求:

1.我是否安装过该软件

```
[root@xnha Packages]# rpm -qa | wc -l    #wc 统计命令
1364                                     #安装过的软件数量
```

2.该软件的作用是什么

```
[root@xnha Packages]# rpm -qpi lftp-4.8.4-1.el8.x86_64.rpm

警告: lftp-4.8.4-1.el8.x86_64.rpm: 头V3 RSA/SHA256 Signature, 密钥 ID 8483c65d:
NOKEY
Name           : lftp
Version        : 4.8.4
Release        : 1.el8
Architecture   : x86_64
Install Date: (not installed)
Group          : Applications/Internet
Size           : 3352597
License        : GPLv3+
Signature      : RSA/SHA256, 2019年07月01日 星期一 19时04分41秒, Key ID
05b555b38483c65d
Source RPM     : lftp-4.8.4-1.el8.src.rpm
Build Date     : 2019年05月13日 星期一 21时08分24秒
Build Host     : x86-02.mbox.centos.org
Relocations    : (not relocatable)
Packager       : CentOS Buildsys <bugs@centos.org>
Vendor         : CentOS
URL            : http://lftp.yar.ru/
Summary        : A sophisticated file transfer program
Description    :
LFTP is a sophisticated ftp/http file transfer program. Like bash, it has job
control and uses the readline library for input. It has bookmarks, built-in
mirroring, and can transfer several files in parallel. It is designed with
reliability in mind.
```

3.该软件安装后会安装到什么位置

```
[root@xnha Packages]# rpm -qp1 lftp-4.8.4-1.el8.x86_64.rpm
警告: lftp-4.8.4-1.el8.x86_64.rpm: 头V3 RSA/SHA256 Signature, 密钥 ID 8483c65d:
NOKEY
/etc/lftp.conf
/usr/bin/lftp
/usr/bin/lftpget
/usr/lib/.build-id
/usr/lib/.build-id/35
/usr/lib/.build-id/35/2f708a7467bd9eb44ebb86c034390a8313623e
/usr/lib/.build-id/3a
/usr/lib/.build-id/3a/545da28a74a9976490788e444a588adfa5e45e
....
```

4.该软件如何安装

```
[root@xnha Packages]# rpm -ivh lftp-4.8.4-1.el8.x86_64.rpm
警告: lftp-4.8.4-1.el8.x86_64.rpm: 头V3 RSA/SHA256 Signature, 密钥 ID 8483c65d:
NOKEY
Verifying... ##### [100%]
准备中... ##### [100%]
正在升级/安装...
  1:lftp-4.8.4-1.el8 ##### [100%]
[root@xnha Packages]#
```

5.安装后如何使用

```
[root@xnha Packages]# lftp -h
用法: lftp [OPTS] <site>
`lftp' 是在 rc 文件执行后 lftp 执行的第一个命令
-f <file>          执行文件中的命令后退出
-c <cmd>           执行命令后退出
--help            显示帮助信息后退出
--version         显示 lftp 版本后退出
其他的选项同 `open' 命令
-e <cmd>          在选择后执行命令
-u <user>[,<pass>] 使用指定的用户名/口令进行验证
-p <port>         连接指定的端口
<site>           主机名, URL 或书签的名字
[root@xnha Packages]# lftp -u admin,123456 ftp://192.168.62.142
lftp admin@192.168.62.142:~> dir
drwxr-xr-x 1 ftp ftp          0 Aug 27  2024 DVWA-master
drwxr-xr-x 1 ftp ftp          0 Sep 02  2024 efucms-master
drwxr-xr-x 1 ftp ftp          0 Jul 25  2024 error
-rw-r--r-- 1 ftp ftp      2307 Sep 03  2019 index.html
drwxr-xr-x 1 ftp ftp          0 Sep 02  2024 phpMyAdmin4.8.5
drwxr-xr-x 1 ftp ftp          0 Dec 17  2024 pikachu
-rw-r--r-- 1 ftp ftp    3688731 Jul 02  2023 pikachu.zip
drwxr-xr-x 1 ftp ftp          0 Dec 20  2024 sql
-rw-r--r-- 1 ftp ftp    3662098 Sep 04  2024 sqli-labs-master.zip
-rw-r--r-- 1 ftp ftp        75 Jan 06  2025 test.html
drwxr-xr-x 1 ftp ftp          0 Sep 03  2024 xss-labs-master
-rw-r--r-- 1 ftp ftp    1126110 May 12  2023 xss-labs-master.zip
lftp admin@192.168.62.142:/> get test.html
```

```
get: /mnt/AppStream/Packages/test.html: 只读文件系统 #现在在光盘里!
[root@xnha etc]# rpm -qf lftp.conf # 文件来自于哪一个软件包
lftp-4.8.4-1.el8.x86_64
lftp admin@192.168.62.142:/> lcd /root/桌面 #进入到本地目录中
lcd 成功, 本地目录=/root/桌面
lftp admin@192.168.62.142:/> get test.html
75 bytes transferred
lftp admin@192.168.62.142:/>
lftp admin@192.168.62.142:/> !ls #!后面跟的是linux命令
2024111 310.tar.gz test.html
```

6.如何卸载

```
[root@xnha Packages]# rpm -e lftp
```

四、RPM 包管理全解

1. 安装参数详解

- `-i`: 安装
- `-v`: 显示详细信息
- `-h`: 显示进度条
- **完整命令**: `rpm -ivh 包名.rpm`

2. 查询操作

- `rpm -qa`: 列出所有已安装包
- `rpm -qpi 包名`: 查看包详细信息 (版本、安装时间)
- `rpm -qpl 包名`: 列出包安装的文件路径

3. 依赖问题处理

- **典型错误**: `error: Failed dependencies: ...`
- **解决方法**: 按提示先安装依赖包, 或强制安装 (不推荐): `rpm -ivh --nodeps 包名`

实验: 卸载再安装vim编辑器

1.查看安装了哪些vim软件包

```
[root@xnha ~]# rpm -qa | grep vim
vim-minimal-8.0.1763-19.el8.4.x86_64
vim-enhanced-8.0.1763-19.el8.4.x86_64
vim-filesystem-8.0.1763-19.el8.4.noarch
vim-common-8.0.1763-19.el8.4.x86_64
```

2.尝试卸载 vim-common-8.0.1763-19.el8.4.x86_64 , 发现卸载不掉

```
[root@xnha ~]# rpm -e vim-common
错误: 依赖检测失败:
    vim-common = 2:8.0.1763-19.el8.4 被 (已安装) vim-enhanced-2:8.0.1763-19.el8.4.x86_64 需要
```

3.根据错误提示, 先卸载 vim-enhanced-2:8.0.1763-19.el8.4.x86_64 , 再卸载 vim-common-8.0.1763-19.el8.4.x86_64

```
[root@xnha ~]# rpm -e vim-enhanced
[root@xnha ~]# rpm -e vim-common
[root@xnha ~]# vim
bash: vim: 未找到命令...
```

rpm安装/卸载都具有依赖性：因为站在软件开发的角度来看，开发一个A软件需要某个功能，但B软件提供了这个功能，那么直接调用B软件的功能就可以了。这样的话A软件就依赖了B软件，在安装时就要先安装B软件再安装A软件。

4.挂载镜像文件，安装vim，发现还是需要按顺序安装

```
[root@xnha ~]# mount /dev/sr0 /mnt
mount: /mnt: WARNING: device write-protected, mounted read-only.
[root@xnha ~]# cd /mnt
[root@xnha mnt]# ls
AppStream BaseOS EFI images isolinux media.repo TRANS.TBL
[root@xnha mnt]# cd AppStream/
[root@xnha AppStream]# cd Packages/
[root@xnha Packages]# rpm -ivh vim-enhanced-8.0.1763-13.el8.x86_64.rpm
警告: vim-enhanced-8.0.1763-13.el8.x86_64.rpm: 头V3 RSA/SHA256 Signature, 密钥 ID
8483c65d: NOKEY
错误: 依赖检测失败:
    vim-common = 2:8.0.1763-13.el8 被 vim-enhanced-2:8.0.1763-13.el8.x86_64 需要

-----按顺序安装-----
[root@xnha Packages]# rpm -ivh vim-common-8.0.1763-13.el8.x86_64.rpm
警告: vim-common-8.0.1763-13.el8.x86_64.rpm: 头V3 RSA/SHA256 Signature, 密钥 ID
8483c65d: NOKEY
Verifying... ##### [100%]
准备中... ##### [100%]
正在升级/安装...
    1:vim-common-2:8.0.1763-13.el8 ##### [100%]

[root@xnha Packages]# rpm -ivh vim-enhanced-8.0.1763-13.el8.x86_64.rpm
警告: vim-enhanced-8.0.1763-13.el8.x86_64.rpm: 头V3 RSA/SHA256 Signature, 密钥 ID
8483c65d: NOKEY
Verifying... ##### [100%]
准备中... ##### [100%]
正在升级/安装...
    1:vim-enhanced-2:8.0.1763-13.el8 ##### [100%]
[root@xnha Packages]# vim
```



```
root@xnha:/etc/yum.repos.d
文件(E) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
#CentOS-Stream-AppStream.repo
#
# The mirrorlist system uses the connecting IP address of the client and the
# update status of each mirror to pick current mirrors that are geographically
# close to the client. You should use this for CentOS updates unless you are
# manually picking other mirrors.
#
# If the mirrorlist does not work for you, you can try the commented out
# baseurl line instead.

[appstream]
name=CentOS Stream $releasever - AppStream
mirrorlist=http://mirrorlist.centos.org/?release=$stream&arch=$basearch&repo=AppStream&infra=$infra
#baseurl=http://mirror.centos.org/$contentdir/$stream/AppStream/$basearch/os/
gpgcheck=1
enabled=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial
~
~
~
~
-- 插入 -- 1,2 全部
```

○ 字段说明：

- `[local]`：标签/源标识（唯一）
- `name`：描述信息
- `baseurl`：仓库路径（支持 `file://`、`http://`、`ftp://`）
- `gpgcheck=1`：是否启用密钥对验证
- `enabled=1`：启用该源
- `gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial`：公钥
- 官方封装的rpm包，用官方的私钥对其进行签名（加密）

2. 本地源优化技巧

- **挂载光盘**：通过 `autofs` 服务（自动挂载，适合挂载光盘）自动挂载到 `/misc/cd`，无需手动 `mount`。

```
[root@xnha ~]# yum install autofs
[root@xnha ~]# systemctl start autofs.service
[root@xnha ~]# systemctl enable autofs
```

- **访问路径**：`cd /misc/cd` 触发自动挂载，优化光盘访问路径

```
[root@xnha misc]# cd /misc/
[root@xnha misc]# ls
[root@xnha misc]# cd cd
[root@xnha cd]# ls
AppStream BaseOS EFI images isolinux media.repo TRANS.TBL
[root@xnha cd]#
```

- **配置本地源**：将原本的.repo打包至同目录下的bak文件夹中，编辑自己的本地源

```
[root@xnha yum.repos.d]# mkdir bak
[root@xnha yum.repos.d]# mv C* bak #备份默认源文件
[root@xnha yum.repos.d]# ls
bak
[root@xnha yum.repos.d]# touch my.repo
```

```
[root@xnha yum.repos.d]# vim my.repo
[root@xnha yum.repos.d]# cat my.repo
[local]
name=AppStream
baseurl=file:///misc/cd/AppStream
enable=1
gpgcheck=0

[local2]
name=Base
baseurl=file:///misc/cd/BaseOS
enable=1
gpgcheck=0

[root@xnha yum.repos.d]# yum clean all
24 文件已删除
```

- 使用yum 卸载 vim

```
[root@xnha yum.repos.d]# yum remove vim-common.x86_64
```

模块依赖问题

问题 1: conflicting requests

- nothing provides module(perl:5.26) needed by module perl-DBD-SQLite:1.58:8030020200716174729:3a70019f-0.x86_64

问题 2: conflicting requests

- nothing provides module(perl:5.26) needed by module perl-DBI:1.641:8030020200716150652:1e4bbb35-0.x86_64

依赖关系解决。

```
=====
=====
```

| 软件包 | 架构 | 版本 | 仓库 |
|-----|----|----|----|
| 大小 | | | |

```
=====
=====
```

移除:

| | | | |
|------------|--------|-------------------|------------|
| vim-common | x86_64 | 2:8.0.1763-13.e18 | @AppStream |
| 27 M | | | |

移除依赖的软件包:

| | | | |
|--------------|--------|-------------------|------------|
| vim-enhanced | x86_64 | 2:8.0.1763-13.e18 | @AppStream |
| 3.4 M | | | |

清除未被使用的依赖关系:

| | | | |
|----------|--------|---------------|------------|
| gpm-libs | x86_64 | 1.20.7-15.e18 | @AppStream |
| 30 k | | | |

事务概要

```
=====
=====
```

移除 3 软件包

将会释放空间: 30 M

确定吗? [y/N]: y

运行事务检查

事务检查成功。

运行事务测试


```
事务测试成功。
运行事务
 准备中   :
 1/1
删除      : vim-enhanced-2:8.0.1763-13.el8.x86_64
 1/3
删除      : gpm-libs-1.20.7-15.el8.x86_64
 2/3
运行脚本: gpm-libs-1.20.7-15.el8.x86_64
 2/3
删除      : vim-common-2:8.0.1763-13.el8.x86_64
 3/3
运行脚本: vim-common-2:8.0.1763-13.el8.x86_64
 3/3
验证      : gpm-libs-1.20.7-15.el8.x86_64
 1/3
验证      : vim-common-2:8.0.1763-13.el8.x86_64
 2/3
验证      : vim-enhanced-2:8.0.1763-13.el8.x86_64
 3/3

已移除:
  vim-common-2:8.0.1763-13.el8.x86_64   vim-enhanced-2:8.0.1763-
13.el8.x86_64
  gpm-libs-1.20.7-15.el8.x86_64

完毕!
[root@xnha yum.repos.d]# vim
bash: /usr/bin/vim: 没有那个文件或目录
```

- 使用yum 安装vim

```
◦ yum install vim-enhanced
AppStream                               6.0 MB/s | 5.4 MB
00:00
Base                                    88 MB/s | 2.2 MB
00:00
依赖关系解决。

=====
=====
 软件包              架构      版本              仓库
大小
=====
=====
安装:
  vim-enhanced      x86_64     2:8.0.1763-13.el8  local
1.4 M
安装依赖关系:
  gpm-libs          x86_64     1.20.7-15.el8     local
  39 k
  vim-common        x86_64     2:8.0.1763-13.el8  local
6.3 M

事务概要
```

```

=====
=====
安装 3 软件包

总计: 7.7 M
安装大小: 30 M
确定吗? [y/N]: y
下载软件包:
运行事务检查
事务检查成功。
运行事务测试
事务测试成功。
运行事务
 准备中   :
 1/1
 安装      : vim-common-2:8.0.1763-13.el8.x86_64
 1/3
 安装      : gpm-libs-1.20.7-15.el8.x86_64
 2/3
 运行脚本: gpm-libs-1.20.7-15.el8.x86_64
 2/3
 安装      : vim-enhanced-2:8.0.1763-13.el8.x86_64
 3/3
 运行脚本: vim-enhanced-2:8.0.1763-13.el8.x86_64
 3/3
 运行脚本: vim-common-2:8.0.1763-13.el8.x86_64
 3/3
 验证      : gpm-libs-1.20.7-15.el8.x86_64
 1/3
 验证      : vim-common-2:8.0.1763-13.el8.x86_64
 2/3
 验证      : vim-enhanced-2:8.0.1763-13.el8.x86_64
 3/3

已安装:
  vim-enhanced-2:8.0.1763-13.el8.x86_64      gpm-libs-1.20.7-
15.el8.x86_64
  vim-common-2:8.0.1763-13.el8.x86_64

完毕!

```

3. 网络源加速

- 华为镜像站: <https://mirrors.huaweicloud.com/home>
- 国内镜像站: 华为、阿里云 (替换 `baseurl` 为镜像站地址)。
- 示例 (华为源):

bash

```

wget -O /etc/yum.repos.d/CentOS-Base.repo
https://mirrors.huaweicloud.com/repository/conf/CentOS-8-reg.repo

```

4. yum常见命令

- 查看仓库（仓库是用来存放软件包的，默认有如下几个，都在公网上）

```
[root@xnha yum.repos.d]# yum repolist
仓库 id                仓库名称
appstream              CentOS Stream 8 - AppStream
baseos                 CentOS Stream 8 - BaseOS
extras                 CentOS Stream 8 - Extras
extras-common          CentOS Stream 8 - Extras common packages
```

- 列表软件包

```
[root@xnha yum.repos.d]# yum list
Repository AppStream is listed more than once in the configuration
Repository extras is listed more than once in the configuration
Repository PowerTools is listed more than once in the configuration
Repository centosplus is listed more than once in the configuration
```

- 清除缓存

```
[root@xnha network-scripts]# yum clean all
17 文件已删除
```

- 卸载软件

```
yum remove
```

- 安装软件

```
yum install
```

六、德班系 (Debian/Kali) 软件管理

1. dpkg 命令细节

- 安装: `dpkg -i 包名.deb`
- 卸载: `dpkg -r 包名` (保留配置文件) 或 `dpkg -P 包名` (彻底删除)
- 查询: `dpkg -l | grep 包名`

2. 依赖修复

- 自动修复: `apt-get install -f` (需联网下载依赖)。

3. APT 工具补充

- 更新源: `apt-get update`
 - 安装软件: `apt-get install 包名` (自动处理依赖)
-

七、故障排查与调试

1. 依赖冲突解决

- 查看冲突包: `yum history` + `yum history info` 事务ID
- 回滚操作: `yum history undo` 事务ID

2. 日志分析

- YUM 日志: `/var/log/yum.log`
- RPM 安装日志: `/var/log/rpmpkgs`

3. 网络源超时处理

- 超时设置: 在 `.repo` 文件中添加 `timeout=30` (单位: 秒)。

总结

- **核心:** 掌握包管理工具 (YUM/RPM/dpkg)、别名配置、用户权限管理。
- **细节:** 注意配置文件生效方式、危险命令防护、日志与调试技巧。
- **安全:** 始终验证软件包来源, 避免执行未知脚本。
- **高效操作:** 善用别名、管道符、自动挂载等提升工作效率。