

Linux 软件安装

Linux 软件安装

源代码安装

源代码安装步骤

卸载步骤

MPlayer源码安装实验

二进制安装

RPM

YUM

yum 的作用

yum 仓库

依赖关系

安装软件 createrepo

配置仓库

Linux 软件安装课后作业

源代码安装

- 优势：可定制
- 缺点：操作复杂，编译时间长，极易出现错误

源代码安装步骤

- 下载解压,阅读软件包附带的 `install` 文件和 `readme` 文件,获取软件的相关信息。
- 进入解包之后的目录,执行 “`./configure`” 命令,为编译做好关于本地环境的配置。
- 配置成功后,执行 “`make`” 命令进行软件编译。
- 编译成功后,执行 “`make install`” 命令完成安装。
- 最后,执行 “`make clean`” 命令删除安装时产生的临时文件

卸载步骤

- 先进入软件的安装目录,然后执行卸载命令即可: `make uninstall`
- 如果有的软件包不提供 `uninstall` 功能,则必须进行手动删除。因此你需要阅读安装目录里面的`readme` 文件,或者在安装的过程中指定安装目录,即在 `./configure` 命令后面添加参数 `--prefix` ,例如: `./configure--prefix=/usr/local/dir`
- 该命令将把软件安装在 `/usr/local/` 路径的 `dir` 目录里。通常情况下,大多数软件都默认安装在 `/usr/local` 目录里。

MPlayer源码安装实验

下载软件包到系统某个目录下,本次演示中使用/tmp/mplayer/目录,软件包包括主程序、库函数、皮肤

```
wget ....  
all-20071007.tar.bz2    库函数  
Blue-1.7.tar.bz2       皮肤  
MPlayer-1.0rc2.tar.bz2  主程序
```

将压缩包解压

```
tar -jxf
```

创建目录用来存放库函数

```
mkdir /usr/local/lib/codes
```

将/tmp/mplayer/all/目录下所有文件复制到/usr/local/lib/codes

```
cp all-20071007/* /usr/local/lib/codes  
ll /usr/local/lib/codes
```

安装依赖包

```
yum install -y kernel-devel gcc zlib-devel gtk2-devel
```

检查安装环境

```
cd /tmp/mplayer/MPlayer-1.0rc2  
./configure --enable-gui --codecsdir=/usr/local/lib/codes --enable-x11 --enable-xshape --  
language=zh_CN --disable-ivtv --disable-png
```

编译make

```
make
```

安装

```
make install
```

装皮肤

这部分不是代码，而是程序装在了哪里

```
Install prefix: /usr/local  
Data directory: /usr/local/share/mplayer  
Config direct.: /usr/local/etc/mplayer
```

- 创建目录 `mkdir /usr/local/share/mplayer/skins/default/`
- 将皮肤文件复制到/usr/local/share/mplayer/default/目录中
`cp /tmp/mplayer/Blue/* /usr/local/share/mplayer/skins/default/`

从真机桌面用鼠标双击打开图形化界面的rhel6，去测试，是否安装成功。

```
applications--->sound&video---->mplayer
```

二进制安装

直接解压缩即可使用

例如 mycat 数据库代理服务器

安装 mycat ,直接解压缩即可使用

```
tar xf Mycat-server-1.5.1-RELEASE-20160328130228-linux.tar.gz -C /usr/local
cd /usr/local;ls
cd mycat;ll
chmod 755 * -R
vim conf/schema.xml
vim conf/server.xml
bin/mycat start
```

RPM

RPM : redhat package management

rpm 命令是 RPM 软件包的管理工具。rpm 原本是 Red Hat Linux 发行版专门用来管理 Linux 各项套件的程序,由于它遵循 GPL 规则且功能强大方便,因而广受欢迎。逐渐受到其他发行版的采用。RPM 套件管理方式的出现,让 Linux 易于安装,升级,间接提升了 Linux 的适用度。

rpm 的命名规范: 软件名 - 版本号 - 操作系统平台

libreoffice4.1-calc-4.1.6.2-1.x86_64.rpm

zlib-1.2.3-29.el6.x86_64.rpm

语法 `rpm (选项)(参数)`

选项

- a :查询所有套件;
- b< 完成阶段 >< 套件档 >+ 或 -t < 完成阶段 >< 套件档 >+ :设置包装套件的完成阶段,并指定套件档的文件名称;
- c :只列出组态配置文件,本参数需配合 "-l" 参数使用;
- d :只列出文本文件,本参数需配合 "-l" 参数使用;
- e< 套件档 > 或 --erase< 套件档 > :删除指定的套件;
- f< 文件 >+ :查询拥有指定文件的套件;
- h 或 --hash :套件安装时列出标记;
- i :显示套件的相关信息;
- i< 套件档 > 或 --install< 套件档 > :安装指定的套件档;
- l :显示套件的文件列表;
- p< 套件档 >+ :查询指定的 RPM 套件档;
- q :使用询问模式,当遇到任何问题时, rpm 指令会先询问用户;
- R :显示套件的关联性信息;
- s :显示文件状态,本参数需配合 "-l" 参数使用;
- U< 套件档 > 或 --upgrade< 套件档 > :升级指定的套件档;
- v :显示指令执行过程;
- vv :详细显示指令执行过程,便于排错。

参数 软件包:指定要操纵的 rpm 软件包。

实例

install 安装

1)rpm -ivh [x.rpm] 安装 v\h 显示安装过程中的进度条 verbose\hash

query 查询

2)rpm -q [软件名称] 查看软件是否安装

3)rpm -qi [软件名称] 查看软件的详细信息

4)rpm -ql [软件名称] 查看软件在系统中安装过的文件

5)rpm -qf [文件名称] 查看文件是由哪个软件包安装出来的

6)rpm -qa 查看系统里所有已经安装过的软件包

卸载 remove

6)rpm -e [软件名称] 卸载软件

rpm -e --nodeps 不卸载依赖关系

update 升级

7)rpm -U [软件名称] 升级 , 若没有该软件则安装

8)rpm -F [软件名称] 升级 , 若没有该软件则不安装

YUM

yum 的作用

作用:为了解决包之间的依赖关系而存在的一种管理机制,基于 rpm 为前端的包管理机制。
为了解决依赖关系,引入了一种仓库的机制。

yum 仓库

仓库：用来存放软件和软件之间的依赖关系，当我们需要安装软件的时候，就可以通过该依赖关系，来将相应的依赖包都装上。`.repodata` 目录就是 `yum` 的仓库，存放软件和软件之间的依赖关系数据。

```
[root@rhel6 dvd]# ll repodata/ -d
dr-xr-xr-x. 2 root root 4096 Nov 12
[root@rhel6 dvd]# pwd
/mnt/rhel6.5/x86_64/dvd
```

依赖关系

安装系统的光盘中已经有建好的依赖关系了,即 `repodata/` 目录,如果要自己手动制作 `rpm` 包依赖关系目录,该怎么做呢?

安装软件 `createrepo`

rhel7 默认已经安装

```
[root@rhel7 ~]# yum install -y createrepo
Loaded plugins: langpacks, product-id, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-
manager
to register.
server
| 4.1 kB 00:00:00
(1/2): server/group_gz
| 134 kB 00:00:00
(2/2): server/primary_db
| 3.4 MB 00:00:00
Package createrepo-0.9.9-23.el7.noarch already installed and latest version
Nothing to do
```

rhel6 要自己安装

```
[root@rhel6 rc.d]# yum install -y createrepo
Loaded plugins: product-id, refresh-packagekit, security, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-
manager
to register.
server
Running Transaction
Installing : deltarpm-3.5-0.5.20090913git.el6.x86_64
1/3
Installing : python-deltarpm-3.5-0.5.20090913git.el6.x86_64
2/3
Installing : createrepo-0.9.9-18.el6.noarch
3/3
Verifying : createrepo-0.9.9-18.el6.noarch
1/3
Verifying : python-deltarpm-3.5-0.5.20090913git.el6.x86_64
2/3
Verifying : deltarpm-3.5-0.5.20090913git.el6.x86_64
3/3
Installed:
createrepo.noarch 0:0.9.9-18.el6
Dependency Installed:
deltarpm.x86_64 0:3.5-0.5.20090913git.el6
Complete!
python-deltarpm.x86_64 0:3.5-0.5.20090913git.el64.3.2 制作 rpm 包依赖关系目录
```

以 rhel7 为例从学校服务器上拷贝一些软件到 /tmp/dvd7.1/ 目录下

```
[root@rhel7 tmp]# cp /mnt/rhel7.1/x86_64/dvd/Packages/* /tmp/dvd7.1
[root@rhel7 tmp]# cd /tmp/dvd7.1
```

制作 rpm 包依赖关系

```
[root@rhel7 dvd7.1]# createrepo /tmp/dvd7.1/
Spawning worker 0 with 4371 pkgs
Workers Finished
Saving Primary metadata
Saving file lists metadata
Saving other metadata
Generating sqlite DBs
Sqlite DBs complete
```

已经成功创建

```
[root@rhel7 tmp]# ll dvd7.1/ |grep repodata
drwxr-xr-x. 2 root root
4096 Mar 18 07:42 repodata
```

配置仓库

为此，我们需要优先去配置一个仓库指向文件。这个文件的位置在 /etc/yum.repos.d/ 目录下，以 .repo 结尾

```
[root@rhel7 tmp]# cd /etc/yum.repos.d/
[root@rhel7 yum.repos.d]# ls
server.repo
```

```
[ 仓库名 ]
name = 说明信息
baseurl = 指向 repodata 目录的上一级 .
enable = 是否启用该仓库 0 代表不启用 ,1 代表启用
gpgcheck = 是否需要检测
其中 baseurl
1. 本地路径 file:///content/rhel6.5..... 以下省略 第三个 / 代表的是根 .
2. 远程路径 协议 :// 位置

[server]
name = rhel7.1 repos
baseurl = http://classroom.example.com/content/rhel6.5/x86_64/dvd/
enable=1
gpgcheck=0
[test]
name = info
baseurl=file:///tmp/dvd7.1/
enable=1
gpgcheck=0
```

配置完仓库后，需要使用

- yum clean all 来清理缓存
- yum makecache 来重新生成缓存 .4.5 yum 安装

安装

- yum install 软件名 安装指定软件
- yum groupinstall 组名 用来安装一系列的软件包 ,
- yum reinstall 软件名 重新安装指定软件

-y 选项，忽略安装过程出现的 is this ok 的交互信息

- yum localinstall 软件名 本地安装指定软件

查询

- yum list 用来查询所有的软件包
- yum list installed 用来查看已经安装过的软件包
- yum search 字符串 能够将说明信息里含该字符串的相关软件包罗列出来
- yum info 软件包名 用来查看软件包的详细信息
- yum grouplist 组名 用来查询系统里所有的组包
- yum groupinfo 组名 用来查询指定组的相关信息

升级

- yum update 软件名
- yum upgrade 软件名

两种写法执行效果没有区别

卸载

- `yum remove` 软件名

不要使用 `yum` 去卸载，会将软件的依赖关系一并卸载掉。

Linux 软件安装课后作业

1. 配置一个 `yum` 仓库。熟悉一下配置的几个字段内容。
2. 安装 `vsftpd` 软件。
3. 查看 `vsftpd` 软件的安装文件有哪些
4. 查看一下 `touch` 命令是由哪个软件包安装出来的
5. 搜索一下含 `bind` 字符串的软件包。
6. 查询一下 `httpd` 软件有没有安装，没有则将该软件装上。
7. 查看一下 `httpd` 软件的版本号及试用平台。
8. 卸载 `vsftpd` 软件。
9. 卸载 `httpd` 软件，不要卸载与其有依赖关系的软件包