第9章 软件包管理

本章内容

9.1 RPM软件包管理

9.2 使用yum管理RPM软件包

9.3 tar包管理

- 在Linux系统中，最常用的软件包是RPM包和tar包。要管理RPM软件包可以使用rpm和yum命令，yum命令自动化地收集RPM软件包的相关信息，检查依赖性，并且一次安装所有依赖的软件包，无需繁琐地一次次安装。

9.1 RPM软件包管理

- 目前在众多的Linux系统上都采用RPM软件包，这种软件包格式在安装、升级、删除以及查询方面非常方便，不需要进行编译即可安装软件包。本节主要讲述RPM软件包的使用和管理。

RPM软件包概念

- RPM（Red Hat Package Manager，Red Hat软件包管理器）是一种开放的软件包管理系统，按照GPL条款发行，可以运行于各种Linux系统上。
- RPM简化了Linux系统安装、卸装、更新和升级的过程，只需要使用简短的命令就可完成。RPM维护一个已经安装软件包和它们的文件的数据库，因此，可以在系统上使用查询和校验软件包功能。
- RPM允许把软件编码包装成源码包和程序包，然后提供给终端用户，这个过程非常简单，这种对用户的纯净源码、补丁和建构指令的清晰描述减轻了发行软件新版本所带来的维护负担。Linux系统上的所有软件都被分成可被安装、升级或卸载的RPM软件包。

RPM软件包管理用途

- 可以安装、删除、升级、刷新和管理RPM软件包；
- 通过RPM软件包管理能知道软件包包含哪些文件，也能知道系统中的某个文件属于哪个RPM软件包；
- 可以查询系统中的RPM软件包是否安装并查询其安装的版本；
- 开发者可以把自己的程序打包为RPM软件包并发布；
- 软件包签名GPG和MD5的导入、验证和签名发布；
- 依赖性的检查，查看是否有RPM软件包由于不兼容而扰乱系统。

RPM软件包管理基本操作模式

- RPM软件包管理主要有安装（添加）、删除（卸载）、刷新、升级、查询这五种基本操作模式。

安装RPM软件包

- 使用rpm命令可以在Linux系统中安装、删除、刷新、升级、查询RPM软件包。

命令语法：

rpm -ivh [RPM软件包文件名称]

【例9.1】 安装bind-9.9.4-29.el7.x86_64.rpm软件包。

```
[root@rhel Packages]# rpm -ivh bind-9.9.4-29.el7.x86_64.rpm
准备中...          ################################# [100%]
正在升级/安装...
1:bind-32:9.9.4-29-el7      ########################## [100%]
```

【例9.3】在软件包bind-9.9.4-29.el7.x86_64.rpm已经安装的情况下仍旧安装该软件包。

```
[root@rhel Packages]# rpm -ivh --replacepkgs
  bind-9.9.4-29.el7.x86_64.rpm
```

删除RPM软件包

- 使用rpm -e命令可以在Linux系统中删除RPM软件包。

命令语法：

rpm -e [RPM包名称]

【例9.5】 删除bind-chroot软件包。

[root@rhel ~]# rpm -e bind-chroot

升级RPM软件包

- 使用rpm -Uvh命令可以在Linux系统中升级RPM软件包，升级软件包实际上是删除和安装的组合。不管该软件包的早期版本是否已被安装，升级选项都会安装该软件包。

命令语法：

rpm -Uvh [RPM软件包文件名称]

【例9.7】升级bind-9.9.4-29.el7.x86_64.rpm软件包，升级软件包实际上是删除和安装的组合。

[root@rhel Packages]# rpm -Uvh bind-9.9.4-29.el7.x86_64.rpm

刷新软件包

- 使用rpm -Fvh命令可以在Linux系统中刷新RPM软件包。使用RPM刷新软件包时，系统会比较指定的软件包的版本和系统上已安装的版本。当RPM的刷新选项处理的版本比已安装的版本更新，它就会升级到更新的版本。如果软件包先前没有安装，RPM的刷新选项将不会安装该软件包，这和RPM的升级选项不同。

命令语法：

rpm -Fvh [RPM软件包文件名称]

【例9.8】 刷新bind-9.9.4-29.el7.x86_64.rpm软件包。

```
[root@rhel Packages]# rpm -Fvh bind-9.9.4-
  29.el7.x86_64.rpm
```

（1）查询指定RPM软件包是否已经安装

命令语法：

rpm -q [RPM包名称]

【例9.9】 查询bind和crontabs软件包是否已经安装

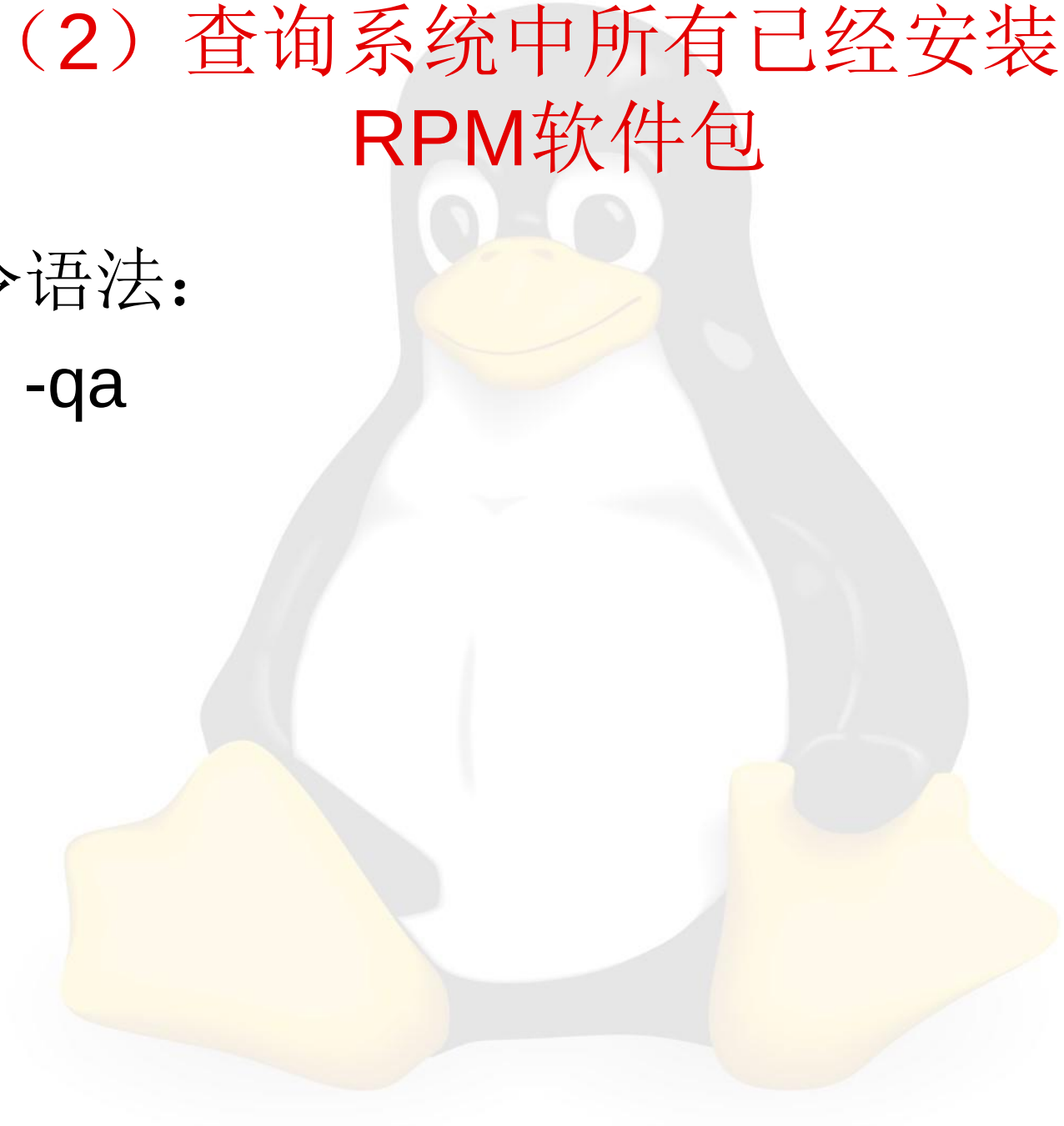
[root@rhel ~]# rpm -q bind
package bind is not installed
//查询到bind软件包没有安装

[root@rhel ~]# rpm -q crontabs
crontabs-1.11-6.20121102git.el7.noarch
//查询到crontabs软件包已经安装

（2）查询系统中所有已经安装的RPM软件包

命令语法：

rpm -qa



【例9.10】 查询系统内所有已安装的RPM软件包。

[root@rhel ~]# rpm –qa

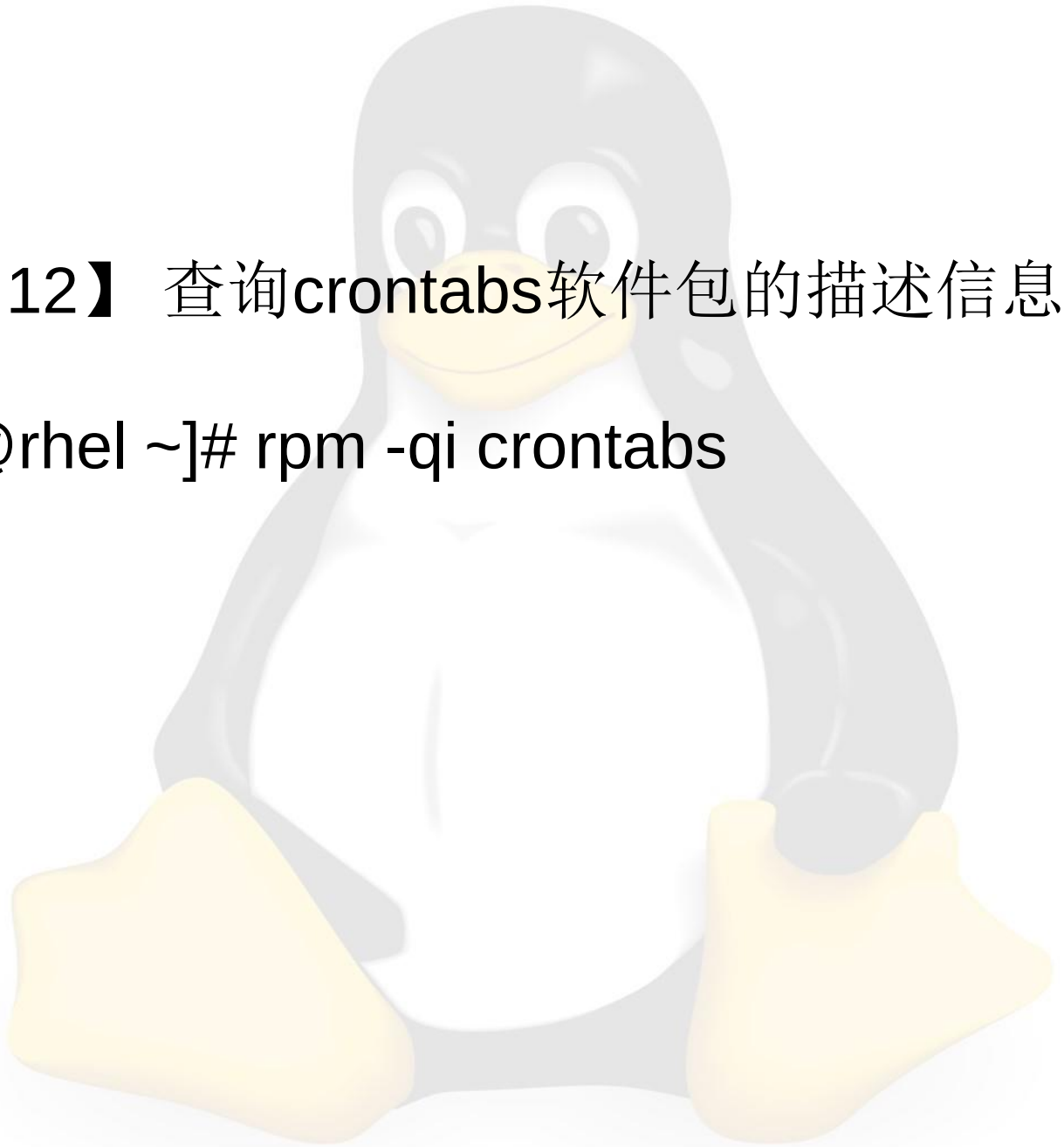
【例9.11】 查询以cront开头的RPM软件包是否已经安装。

[root@rhel ~]# rpm –qa |grep cront
crontabs-1.11-6.20121102git.el7.noarch
//结合管道方式查询

（3）查询已安装RPM软件包的描述信息

命令语法：

rpm -qi [RPM包名称]



【例9.12】 查询crontabs软件包的描述信息。

```
[root@rhel ~]# rpm -qi crontabs
```

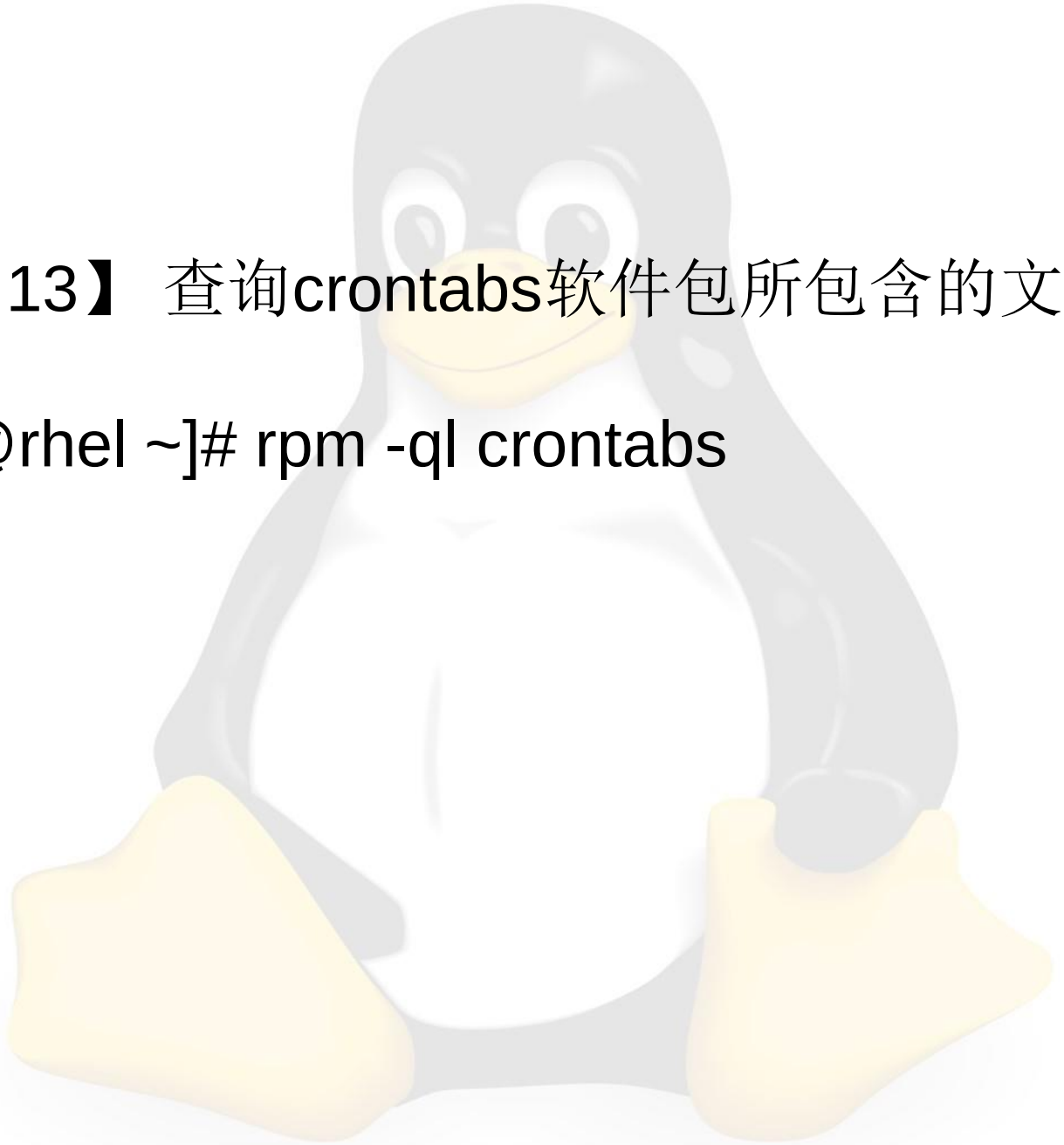
（4）查询指定已安装RPM软件包所包含的文件列表

命令语法：

rpm -ql [RPM包名称]

【例9.13】 查询crontabs软件包所包含的文件列表。

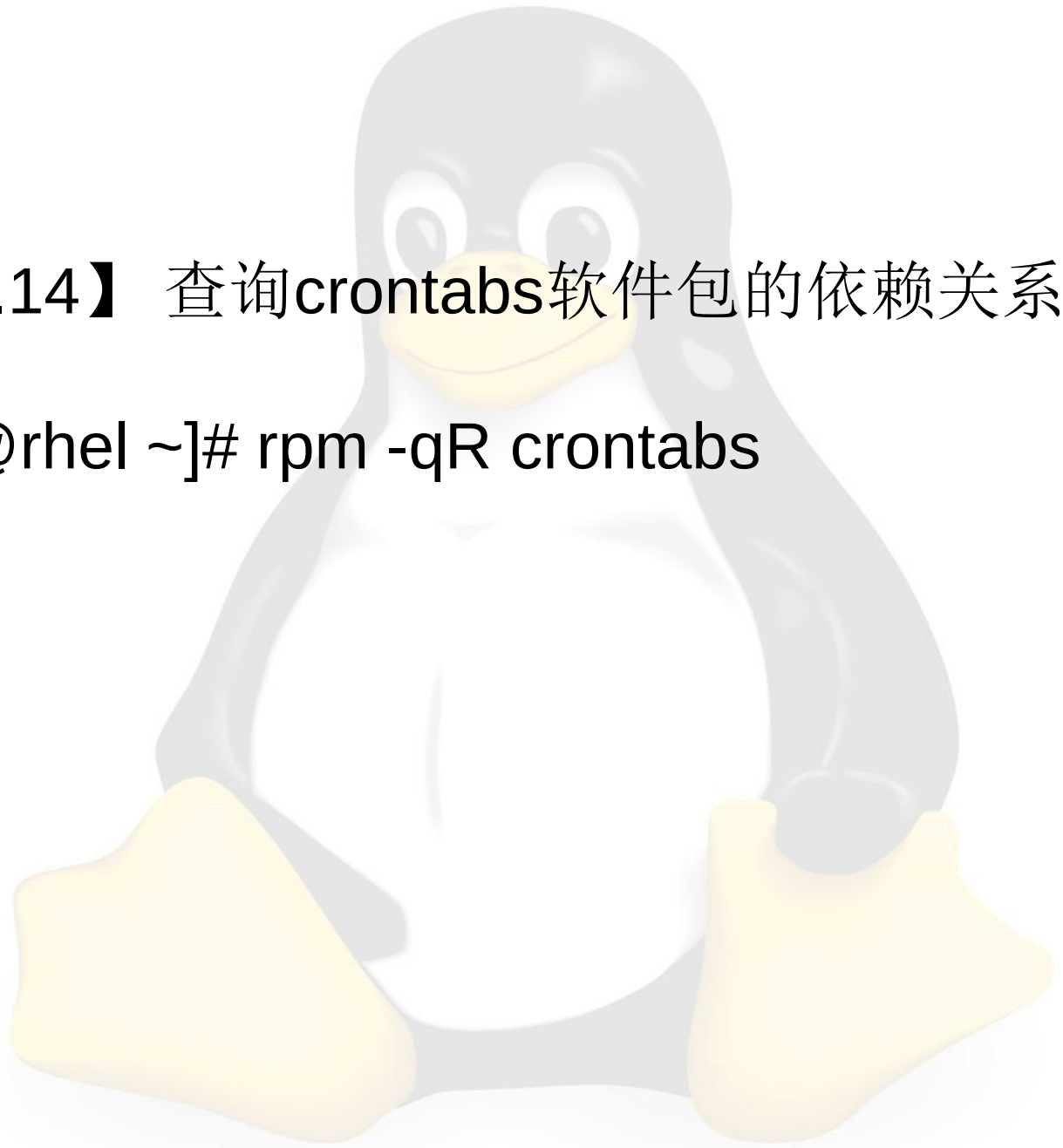
```
[root@rhel ~]# rpm -ql crontabs
```



（5）查询RPM软件包的依赖关系

命令语法：

rpm -qR [RPM包名称]



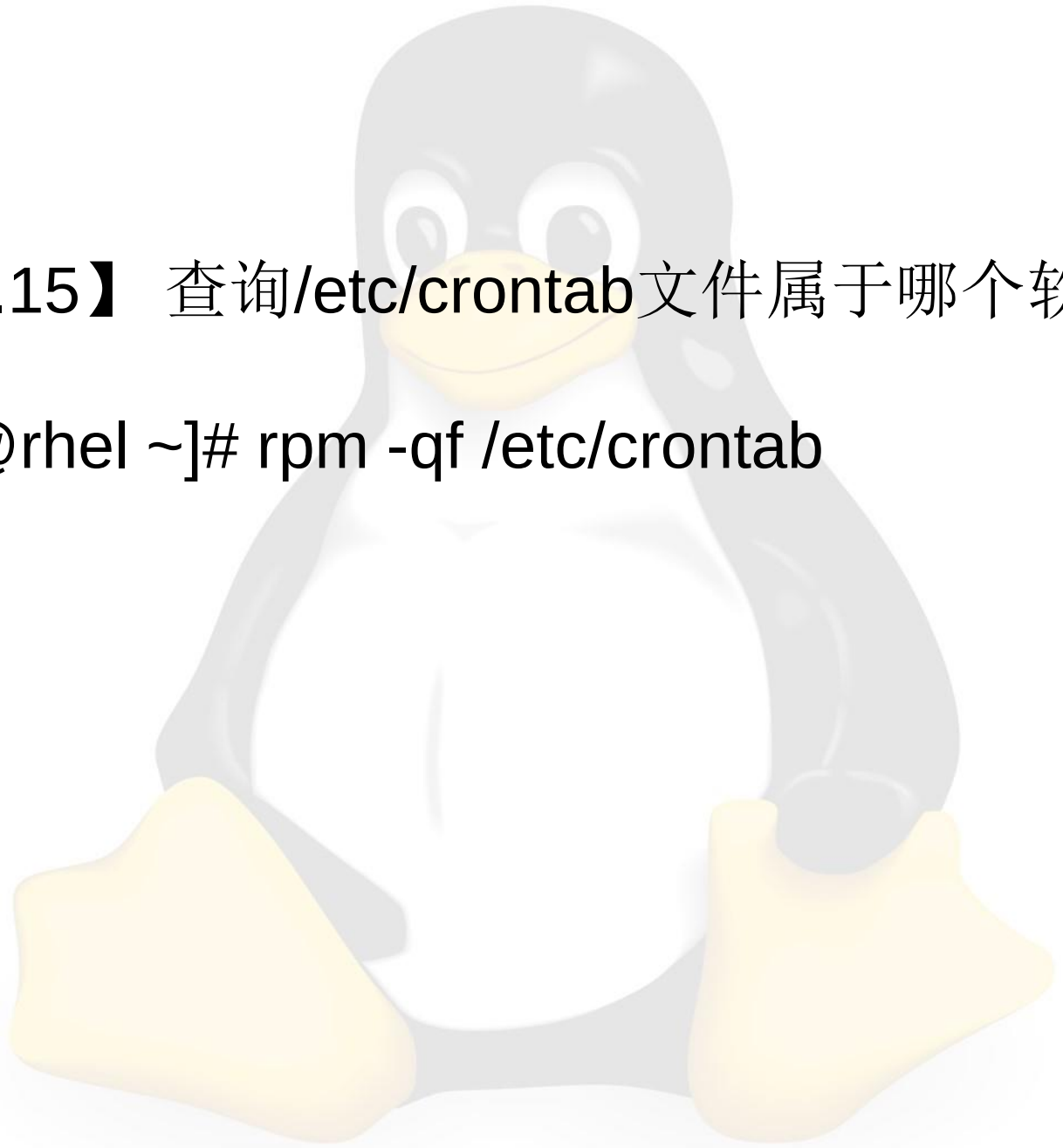
【例9.14】 查询crontabs软件包的依赖关系。

```
[root@rhel ~]# rpm -qR crontabs
```

（6）查询系统中指定文件属于哪个RPM软件包

命令语法：

rpm -qf [文件名]



【例9.15】 查询/etc/crontab文件属于哪个软件包。

```
[root@rhel ~]# rpm -qf /etc/crontab
```

9.2 使用yum管理RPM软件包

- 在Linux系统中安装软件包使用rpm命令，但是使用rpm命令安装软件包特别的麻烦，原因在于需要手动寻找安装该软件包所需要的一系列依赖关系。当软件包不用时需要卸载的话，由于卸载掉了某个依赖关系而导致其它的软件包不能用。在Linux系统中使用yum命令，令Linux的软件安装变得简单。

什么是yum

- yum（Yellow dog Updater Modified）起初是由Terra Soft研发，其宗旨是自动化地升级、安装和删除RPM软件包，收集RPM软件包的相关信息，检查依赖性并且一次安装所有依赖的软件包，无须繁琐地一次次安装。
- yum的关键之处是要有可靠的软件仓库，软件仓库可以是HTTP站点、FTP站点或者是本地软件池，但必须包含rpm的header，header包括了RPM软件包的各种信息，包括描述、功能、提供的文件以及依赖性等。正是收集了这些header并加以分析，才能自动化地完成余下的任务。

yum特点

- 可以同时配置多个软件仓库；
- 简洁的配置文件/etc/yum.conf；
- 自动解决安装或者删除RPM软件包时遇到的依赖性问题；
- 使用yum非常方便；
- 保持与RPM数据库的一致性。

yum软件仓库配置文件

- repo文件是Linux系统中yum源（软件仓库）的配置文件，通常一个repo文件定义了一个或者多个软件仓库的细节内容，比如从哪里下载需要安装或者升级的软件包，repo文件中的设置内容将被yum读取和应用。软件仓库配置文件默认存储在/etc/yum.repos.d目录中。

以下是yum软件仓库配置文件的格式内容。

[rhel-source]

//方括号里面是软件源的名称，会被yum识别。

name=Red Hat $releasever-$basearch-Source

//定义软件仓库名称，$releasever变量定义了发行版本，$basearch变量定义了系统架构，方便yum升级时选择适合当前系统的软件包。

baseurl= ftp://ftp.redhat.com/pub/redhat/linux/enterprise/$releasever/en/os/SRPM/

//指定RPM软件包来源，支持的协议有http://、ftp://、file:///（本地源）

enabled=1

//软件仓库中的源是否启用，0禁用，1启用。

gpgcheck=1

//表示对软件仓库中下载的RPM软件包进行GPG校验，确定来源是否有效和安全。

gpgkey=file:///etc/pki/rpm-gpg/the-file //用于校验的GPG密钥

创建本地软件仓库

1．安装软件包

安装deltarpm、python-deltarpm和createrepo软件包。

2．复制软件包

复制Linux系统安装光盘中的软件包。

3．创建软件仓库配置文件

创建软件仓库配置文件。

4．创建软件仓库

使用createrepo命令创建软件仓库。

yum命令使用

- 使用yum命令可以安装、更新、删除、显示软件包。yum可以自动进行系统更新，基于软件仓库的元数据分析，解决软件包依赖性关系。

命令语法：

yum [选项] [命令]

- 使用yum命令进行软件管理

  1.yum help 显示使用信息

  2.yum list　列出软件包

  3.yum search keyword　搜索关键字

  4.yum info packagename 列出软件包详细信息

  5.yum install packagename 安装软件包

  6.yum remove packagename 删除软件包

  7.yum update packagename　升级软件包

- 使用yum命令安装软件包组

  1.yum grouplist　列出所有可用组

  2.yum groupinfo　提供特定组的信息

  3.yum groupinstall　安装软件包组

  4.yum groupupdate　更新软件包组

  5.yum grouperase　　删除软件包组

- 使用yum命令安装本地rpm包

  yum　localinstall　*.rpm

【例9.16】无需确认、直接安装bind软件包。
[root@rhel ~]# yum -y bind
【例9.17】显示bind软件包的详细信息。
[root@rhel ~]# yum info bind
【例9.18】显示所有已经安装的软件包信息。
[root@rhel ~]# yum info installed
【例9.19】列出bind软件包。
[root@rhel ~]# yum list bind
【例9.20】列出bind软件包的依赖关系。
[root@rhel ~]# yum deplist bind
【例9.21】显示软件仓库的配置。
[root@rhel ~]# yum repolist
【例9.22】查看/etc/named.conf文件是属于哪个软件包的。
[root@rhel ~]# yum provides /etc/named.conf
【例9.23】删除bind软件包。
[root@rhel ~]# yum remove bind

9.3 tar包管理

- 使用tar命令可以将文件和目录进行打包或压缩以做备份用。本节主要讲述tar包的使用和管理。

tar包简介

- Linux系统中最常使用的归档程序是tar，使用tar程序归档的包称为tar包，tar包文件的名称通常都是以“.tar”结尾的。生成tar包以后，还可以使用其它程序来对tar包进行压缩。tar可以为文件和目录创建备份。利用tar命令，用户可以为某一特定文件创建备份，也可以在备份中改变文件，或者向备份中加入新的文件。
- 利用tar命令可以把一大堆的文件和目录打包成一个文件，这对于备份文件或是将几个文件组合成为一个文件进行网络传输是非常有用的。
- Linux系统中的很多压缩程序只能针对一个文件进行压缩，这样当需要压缩一大堆文件时，就得先借助其它的工具（比如tar）将这一大堆文件先打成一个包，然后再使用压缩程序进行压缩。

tar包使用和管理

- 使用tar命令可以将许多文件一起保存到一个单独的磁带或磁盘归档，并能从归档中单独还原所需文件。

命令语法：

tar [选项][文件|目录]

【例9.26】 备份/root/abc目录及其子目录下的全部文件，备份文件名为abc.tar。

```
[root@rhel ~]# tar cvf abc.tar /root/abc
tar: 从成员名中删除开头的“/”
/root/abc/
/root/abc/c
/root/abc/b
/root/abc/a
[root@rhel ~]# ls -l
总用量 284
drwxr-xr-x. 2 root root   4096 6月   3 05:38 abc
-rw-r--r--. 1 root root  10240 6月   3 05:39 abc.tar
-rw-------. 1 root root  10670 6月   3 01:17 anaconda-ks.cfg
-rw-r--r--. 1 root root 155641 6月   3 01:16 install.log
-rw-r--r--. 1 root root  65450 6月   3 01:15 install.log.syslog
//可以看到abc.tar就是/root/abc目录打包后生成的文件
```

【例9.27】 查看abc.tar备份文件的内容，并显示在显示器上。

```
[root@rhel ~]# tar tvf abc.tar
drwxr-xr-x root/root         0 2012-06-03 05:38 root/abc/
-rw-r--r-- root/root         0 2012-06-03 05:38 root/abc/c
-rw-r--r-- root/root         0 2012-06-03 05:38 root/abc/b
-rw-r--r-- root/root         0 2012-06-03 05:38 root/abc/a
//可以看到该打包文件由一个目录和该目录下的3个文件打包而成
```

【例9.28】 将打包文件abc.tar解包出来。

```
[root@rhel ~]# tar xvf abc.tar
root/abc/
root/abc/c
root/abc/b
root/abc/a
```

【例9.29】 将文件/root/abc/d添加到abc.tar包里面去。

```
 [root@rhel ~]# touch /root/abc/d
[root@rhel ~]# tar rvf abc.tar /root/abc/d
tar: 从成员名中删除开头的“/”
/root/abc/d
[root@rhel ~]# tar tvf abc.tar
drwxr-xr-x root/root         0 2012-06-03 05:38 root/abc/
-rw-r--r-- root/root         0 2012-06-03 05:38 root/abc/c
-rw-r--r-- root/root         0 2012-06-03 05:38 root/abc/b
-rw-r--r-- root/root         0 2012-06-03 05:38 root/abc/a
-rw-r--r-- root/root         0 2012-06-03 05:41 root/abc/d
//查看abc.tar包内容，可以看到文件/root/abc/d已经添加进去了
```

【例9.30】 更新原来tar包abc.tar中的文件/root/abc/d。

```
 [root@rhel ~]# tar uvf abc.tar /root/abc/d
tar: 从成员名中删除开头的 “/”
/root/abc/d
[root@rhel ~]# tar tvf abc.tar
drwxr-xr-x root/root         0 2012-06-03 05:38 root/abc/
-rw-r--r-- root/root         0 2012-06-03 05:38 root/abc/c
-rw-r--r-- root/root         0 2012-06-03 05:38 root/abc/b
-rw-r--r-- root/root         0 2012-06-03 05:38 root/abc/a
-rw-r--r-- root/root         0 2012-06-03 05:41 root/abc/d
-rw-r--r-- root/root         0 2012-06-03 05:41 root/abc/d
```

tar包的特殊使用

- 使用tar命令可以在打包或解包的同时调用其他的压缩程序，比如调用gzip、bzip2和xz等。

1．tar调用gzip

- 使用tar命令可以在归档或者是解包的同时调用gzip压缩程序。以“.gz”结尾的文件就是gzip压缩的结果。与gzip相对应的解压缩程序是gunzip，tar命令中使用-z选项来调用gzip。

【例9.31】 把/root/abc目录包括其子目录全部做备份文件，并进行压缩，文件名为abc.tar.gz。

```
[root@rhel ~]# tar zcvf abc.tar.gz /root/abc
tar: 从成员名中删除开头的“/”
/root/abc/
/root/abc/c
/root/abc/b
/root/abc/a
[root@rhel ~]# ls -l
总用量 276
drwxr-xr-x. 2 root root   4096 6月   3 05:48 abc
-rw-r--r--. 1 root root    161 6月   3 05:48 abc.tar.gz
-rw-------. 1 root root  10670 6月   3 01:17 anaconda-ks.cfg
-rw-r--r--. 1 root root 155641 6月   3 01:16 install.log
-rw-r--r--. 1 root root  65450 6月   3 01:15 install.log.syslog
//可以看到abc.tar.gz文件就是/root/abc目录压缩后的文件
```

【例9.32】 查看压缩文件abc.tar.gz的内容，并显示在显示器上。

```
[root@rhel ~]# tar ztvf abc.tar.gz
drwxr-xr-x root/root         0 2012-06-03 05:48 root/abc/
-rw-r--r-- root/root         0 2012-06-03 05:44 root/abc/c
-rw-r--r-- root/root         0 2012-06-03 05:44 root/abc/b
-rw-r--r-- root/root         0 2012-06-03 05:44 root/abc/a
```

//可以看到该压缩文件由一个目录和该目录下的3个文件压缩而成

【例9.33】 将压缩文件abc.tar.gz解压缩出来。

```
[root@rhel ~]# tar zxvf abc.tar.gz
root/abc/
root/abc/c
root/abc/b
root/abc/a
```

2．tar调用bzip2

- 使用tar命令可以在归档或者是解包的同时调用bzip2压缩程序。以“.bz2”结尾的文件就是bzip2压缩的结果。与bzip2相对应的解压缩程序是bunzip2。tar命令中使用-j选项来调用bzip2。

【例9.34】 将目录/root/abc及该目录所有文件压缩成abc.tar.bz2文件。

```
 [root@rhel ~]# tar jcvf abc.tar.bz2 /root/abc
tar: 从成员名中删除开头的“/”
root/abc/
root/abc/c
root/abc/b
root/abc/a
[root@rhel ~]# ls -l
总用量 284
drwxr-xr-x. 2 root root   4096 6月   3 05:48 abc
-rw-r--r--. 1 root root    159 6月   3 05:50 abc.tar.bz2
-rw-------. 1 root root  10670 6月   3 01:17 anaconda-ks.cfg
-rw-r--r--. 1 root root 155641 6月   3 01:16 install.log
-rw-r--r--. 1 root root  65450 6月   3 01:15 install.log.syslog
```

【例9.35】 查看压缩文件abc.tar.bz2的内容，并显示在显示器上。

```
[root@rhel ~]# tar jtvf abc.tar.bz2
drwxr-xr-x root/root         0 2012-06-03 05:48 root/abc/
-rw-r--r-- root/root         0 2012-06-03 05:44 root/abc/c
-rw-r--r-- root/root         0 2012-06-03 05:44 root/abc/b
-rw-r--r-- root/root         0 2012-06-03 05:44 root/abc/a
```

【例9.36】 将abc.tar.bz2文件解压缩。

```
[root@rhel ~]# tar jxvf abc.tar.bz2
root/abc/
root/abc/c
root/abc/b
root/abc/a
```

3．tar调用xz

- 使用tar命令可以在归档或者是解包的同时调用xz压缩程序。以“.xz”结尾的文件就是xz压缩的结果。tar命令中使用-J选项来调用。

【例9.37】 将目录/root/abc及该目录所有文件压缩成abc.tar.xz文件。

```
[root@rhel ~]# tar Jcvf abc.tar.xz /root/abc
tar: 从成员名中删除开头的“/”
[root@rhel ~]# ls -l
总用量 284
drwxr-xr-x. 2 root root   4096 6月   3 05:48 abc
-rw-r--r--. 1 root root    159 6月   3 05:50 abc.tar.xz
-rw-------. 1 root root  10670 6月   3 01:17 anaconda-ks.cfg
-rw-r--r--. 1 root root 155641 6月   3 01:16 install.log
-rw-r--r--. 1 root root  65450 6月   3 01:15 install.log.syslog
```

【例9.38】 查看压缩文件abc.tar.xz的内容，并显示在显示器上。

```
[root@rhel ~]# tar Jtvf abc.tar.bz2
drwxr-xr-x root/root         0 2012-06-03 05:48 root/abc/
-rw-r--r-- root/root         0 2012-06-03 05:44 root/abc/c
-rw-r--r-- root/root         0 2012-06-03 05:44 root/abc/b
-rw-r--r-- root/root         0 2012-06-03 05:44 root/abc/a
```

【例9.39】 将abc.tar.xz文件解压缩。

```
[root@rhel ~]# tar Jxvf abc.tar.xz
root/abc/
root/abc/c
root/abc/b
root/abc/a
```