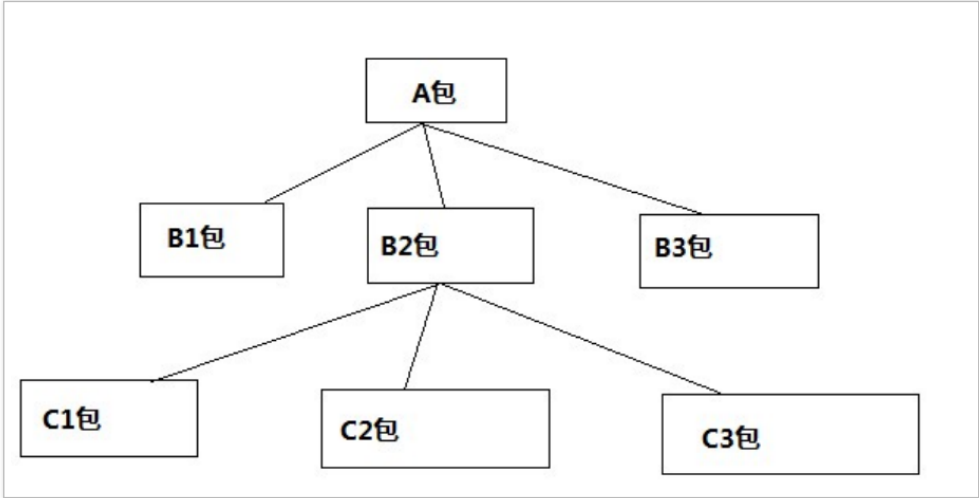


1.3 ubuntu 软件安装命令详解_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 1.3 ubuntu 软件安装命令详解涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

我们常常使用的软件包管理工具有两种，一种叫做 dpkg 软件包管理工具, 它是”debian package” 的简写，是 Debian 软件包管理器的基础。它是最早的 deb 软件包管理工具，他在 Debian Linux 一提出软件包管理模式后就诞生了。使用 dpkg 可以实现软件包的安装、编译、卸载、查询，以及应用程序打包等功能。但是由于当时 Linux 系统规模 和 Internet 网络条件的限制，我们使用 **dpkg 安装软件包的时候需要考虑软件包之前的依赖关系。也有人把它叫做本地安装工具。** 总之，dpkg 是一个底层的软件包管理系统，主要用于对已下载到本地和已安装的软件包进行管理。



基于这个特性，我们的 dpkg 安装一般来说有可能要安装多个软件包。而我们 linux 系统中的依赖关系相对复杂，因此就定义了一个依赖关系表以及优先级表。Ubuntu 中为每个软件包指定了一个优先级，作为软件包管理器选择安装和卸载的一个依据。

依赖性关系定义	
依 赖 关 系	关 系 描 述
依赖（depends）	要运行软件包A必须安装软件包B，甚至还依赖于B的特定版本，通常版本依赖有最低版本限制
推荐（recommends）	软件包维护者认为所有用户都不会喜欢缺少软件包A的某些功能，而这些功能需要B来提供
建议（suggests）	软件包B能够增强软件包A的功能
替换（replaces）	软件包B安装的文件被软件包A中的文件删除或覆盖了
冲突（conflicts）	如果系统中安装了软件包B，那么软件包A将无法运行。“Conflicts”常和“Replaces”同时出现
提供（provides）	软件包A中包含了软件包B中的所有文件和功能

软件包优先级描述			
级 别		含 义	补充说明
Required（必须）	高 低	该级别软件包是保证系统正常运行所必须的	包含所有必要的系统工具。尽管Require级别的软件不能满足整个系统的服务，但至少能够保证系统正常启动。如果删除其中一个软件包，系统将受到损坏而无法恢复。例如bash、mount、upstart
Important（重要）		若缺少该级别软件包，系统会运行困难或不好操作	该级别软件包是一些实现系统底层功能的程序。例如，aptitude、ubuntu-keyring、cpio
Standard（基本）		该级别软件包是任何Linux系统的标准件	该级别的软件包可以支撑命令行控制台系统运行，通常作为默认安装选项，例如，memtest86、telnet、pppconfig、ed
Optional（可选）		该级别软件包是否安装不影响系统的正常运行	该级别的软件包用于满足用户特定的需求或服务，它们不会影响系统的正常运行。例如，X11、mysql、openoffice.org
Extra（额外）		该级别软件包可能与其他高级别软件包存在冲突	

提取码：1c3z

1. 安装软件

- 1. sudo dpkg -i 软件包 [安装单个软件包]
- 2. 或
- 3. sudo apt-get install *.deb [多个软件包一起安装]
- 4. 例如：sudo dpkg -i nano_2.2.6-1_i386.deb

2. 移除已经安装的软件包

- 1. sudo dpkg -r 软件包名
- 2. 例如：sudo dpkg -r nano

3. 移除已安装的软件包及配置文件

```
sudo dpkg --get-selections | grep -v hold | xargs sudo dpkg --purge
```

```
sudo dpkg --get-selections | grep -v hold | xargs sudo dpkg --purge
```

```
sudo dpkg --get-selections | grep -v hold | xargs sudo dpkg --purge
```

APT 系列工具可能是 Deb 软件包管理工具中功能最强大的。它会检测软件包之间的依赖关系。因为它采用了集中式的软件仓库的机制，将各式各样的软件包分门别类地存放在软件仓库中，进行有效地组织和管理。然后，将软件仓库置于许许多多的镜像服务器中，并保持基本一致。（可理解为我们上网的时候，有很多的网站，网站中有一些下载的连接，这里下载的连接中就有很多的软件包，这就是软件包仓库，这里的镜像站点服务器可理解为下载的网站）。这些镜像服务器就是它们的软件源（下载网站的集合）。

我们可以在 `*/etc/apt/sources.list*` 中来查看我们的软件源。这个路径就叫做镜像站点服务器

```
linux@ubuntu:~$ vi /etc/apt/sources.list
```

内容如下：

```

1 # deb cdrom:[Ubuntu 18.04.5 LTS _Bionic Beaver_ - Release amd64 (20200806.1)]/ bion
2
3 # See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
4 # newer versions of the distribution.
5 deb http://mirrors.aliyun.com/ubuntu/ bionic main restricted
6 # deb-src http://cn.archive.ubuntu.com/ubuntu/ bionic main restricted
7
8 ## Major bug fix updates produced after the final release of the
9 ## distribution.
10 deb http://mirrors.aliyun.com/ubuntu/ bionic-updates main restricted
11 # deb-src http://cn.archive.ubuntu.com/ubuntu/ bionic-updates main restricted
12
13 ## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
14 ## team. Also, please note that software in universe WILL NOT receive any
15 ## review or updates from the Ubuntu security team.
16 deb http://mirrors.aliyun.com/ubuntu/ bionic universe
17 # deb-src http://cn.archive.ubuntu.com/ubuntu/ bionic universe
18 deb http://mirrors.aliyun.com/ubuntu/ bionic-updates universe
19 # deb-src http://cn.archive.ubuntu.com/ubuntu/ bionic-updates universe
20
21 ## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
22 ## team, and may not be under a free licence. Please satisfy yourself as to
23 ## your rights to use the software. Also, please note that software in
24 ## multiverse WILL NOT receive any review or updates from the Ubuntu
25 ## security team.
26 deb http://mirrors.aliyun.com/ubuntu/ bionic multiverse
27 # deb-src http://cn.archive.ubuntu.com/ubuntu/ bionic multiverse
28 deb http://mirrors.aliyun.com/ubuntu/ bionic-updates multiverse
29 # deb-src http://cn.archive.ubuntu.com/ubuntu/ bionic-updates multiverse
30

```

我们的镜像站点服务器只是告诉了我们。我们的软件包应该在哪里去下载。但是我们这些镜像站点具体拥有哪些资源，对我们来说，不是很清楚。如果，我们每安装一个软件包就到我们的服务器上去寻找一遍，这样的话效率就太低了。因此，我们提出了一个概念，叫做 ** 索引文件。** 它的本质就是我们为服务器上的软件资源在本地列了一个清单，以便于主机进行查询操作。（可理解为网站服务器在本地的缓存）。

我们可以在 **/var/lib/apt/lists/** 这个目录来查看

```

linux@ubuntu:~$ cd /var/lib/apt/lists/
linux@ubuntu:/var/lib/apt/lists$ ls
auxfiles
lock
mirrors.aliyun.com_ubuntu_dists_bionic-backports_InRelease
mirrors.aliyun.com_ubuntu_dists_bionic-backports_main_binary-amd64_Packag
es
mirrors.aliyun.com_ubuntu_dists_bionic-backports_main_binary-i386_Package
s
mirrors.aliyun.com_ubuntu_dists_bionic-backports_main_i18n_Translation-en
mirrors.aliyun.com_ubuntu_dists_bionic-backports_universe_binary-amd64_Pa
ckages
mirrors.aliyun.com_ubuntu_dists_bionic-backports_universe_binary-i386_Pac
kages
mirrors.aliyun.com_ubuntu_dists_bionic-backports_universe_dep11_Component
s-amd64.yml.gz
mirrors.aliyun.com_ubuntu_dists_bionic-backports_universe_dep11_icons-48x
48.tar.gz

```

我们在 windos 上下载软件的时候，一般是会把 .exe 的安装包下载到一个指定的路径，然后双击它进行安装。我们的 ubuntu 也是一样的。只不过这个下载路径是固定的。

我们可以到 **/var/cache/apt/archives** 这个目录下来看我们的软件包

```

linux@ubuntu:/var/lib/apt/lists$ cd /var/cache/apt/archives/
linux@ubuntu:/var/cache/apt/archives$ ls
lock nano_2.9.3-2_amd64.deb partial sl_3.03-17build2_amd64.deb
linux@ubuntu:/var/cache/apt/archives$

```

sudo apt-get install 软件包名

例如： sudo apt-get install sl 或 sudo apt-get install btanks

sudo apt-get remove 软件包名

例如: `sudo apt-get install sl`

`sudo apt-get update`

`sudo apt-get clean`

`/etc/apt/sources.list` [文件] -> 镜像站点服务器地址 (可以理解成软件包的下载网站)

`/var/lib/apt/lists` [目录] -> 每个镜像站点软件包的索引文件 (可以理解成我们软件包下载 网站具体内容的缓存, 例如: 软件版本, 软件名等等)

`/var/cache/apt/archives` [目录] -> 下载下来软件包存放路径

我们使用 `apt-get` 命令下载文件的时候, 默认是在镜像站点软件包的索引文件 [缓存]

`/var/lib/apt/lists` 中查找该软件的一些信息, 当我们找到索引文件中的软件信息之后, 就进入了我们的镜像站点服务器 [网站]/`etc/apt/sources.list` 地址来下载我们需要的软件, 下载的文件存放在 `**/var/cache/apt/archives**` 目录下。

1. 大家利用 `apt-get` 命令安装一下 `btanks`, 观察是什么样的现象。(把现象截图即可)

安装命令: `sudo apt-get install btanks`

全文完

本文由 简悦 SimpRead 优化, 用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta, 点击查看详细说明

