

# Linux服务器构建与运维管理

## 第1章：Linux简介与安装

阮晓龙

13938213680 / rxl@hactcm.edu.cn

<http://linux.xg.hactcm.edu.cn>

河南中医药大学信息管理与信息系统教研室  
信息技术学院网络与信息系统科研工作室

2020.2

# 提纲

## □ Linux概述

Linux简介、Linux内核、Linux家族

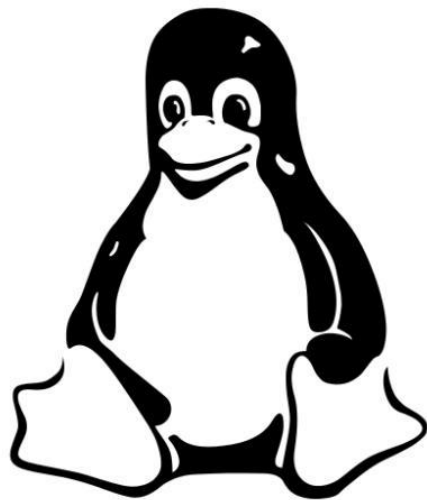
## □ Linux的安装

安装前准备、Linux安装的方式

- 基于VMware vSphere安装CentOS
- 基于VirtualBox安装Ubuntu Server

## □ 操作系统维护

- CentOS系统维护 (AI、RPM、YUM)
- 使用命令行管理方式 (Command、Shell)
- Ubuntu Server系统维护 (APT)



# 1.Linux概述

## 1.1 Linux简介

- Linux是一套可免费使用和自由传播的类Unix操作系统，是一个基于POSIX和UNIX的多用户、多任务、支持多线程和多CPU的操作系统。
- Linux应用广泛，可安装在各种计算机硬件设备中，比如手机、平板电脑、路由器、视频游戏控制台、台式计算机、大型机和超级计算机等。



# 1.Linux概述

## 1.1 Linux简介

- 全球的公有云中，90%的虚拟主机是Linux操作系统，嵌入式市场Linux的占有率是60%以上，超级计算机市场Linux占有率达99%，世界上超过80%智能手机上运行Linux。



# 1.Linux概述

## 1.1 Linux简介

- ❑ 1969年 UNIX是AT&T公司的贝尔实验室的一个实验项目。
- ❑ 1973年 UNIX被无偿提供给众多大学和实验室研究项目，UNIX开始与学术界合作开发，诞生了BSD（Berkeley Software Distribution，伯克利软件发行版）。
- ❑ 1979年 AT&T公司出于商业考虑决定将UNIX的版权收回，于是荷兰Andrew S. Tanenbaum教授开发了Minix系统用于教学。
- ❑ 1991年 芬兰的大学生Linus Torvalds对Minix进行研究，并基于此开发了自己的操作系统：Linux（Linus的UNIX）。
  - 同时Linus Torvalds还将系统的源代码上传到互联网，引起了很多人的兴趣，
  - 从此Linux正式诞生。



# 1.Linux概述



表 1-1 Linux 发展大事记

| 时间            | 事件                                                  |
|---------------|-----------------------------------------------------|
| 1991 年 9 月    | Linux 0.0.1 诞生                                      |
| 1991 年 10 月   | Linux 0.0.2 诞生，并且正是被命名为 Linux                       |
| 1992 年 4 月    | 第一个 Linux 新闻组 comp.os.linux 诞生                      |
| 1992 年 10 月   | 第一个可以安装的 Linux 版本 SLS (Softlanding Linux System) 诞生 |
| 1992 年-1994 年 | Slackware、RedHat、Debian 诞生                          |
| 1994 年 3 月    | Linux 1.0.0 推出，Linux 转向 GPL 版权协议                    |
| 1995 年 3 月    | Linux 1.2 推出                                        |
| 1995 年 11 月   | Alpha 架构的 Linux 推出                                  |
| 1996 年 6 月    | Linux 2.0 推出，Debian GNU/Linux 1.1 推出                |
| 1999 年 1 月    | Linux 2.2 推出                                        |
| 2000 年        | Linux 基金会成立                                         |
| 2001 年 1 月    | Linux 2.4 推出                                        |
| 2003 年 12 月   | Linux 2.6 推出                                        |
| 2004 年        | Ubuntu 第一次发布，版本为 4.10                               |
| 2009 年 9 月    | 移动端商用操作系统 Android 发布                                |
| 2011 年        | 基于 Linux 的 IBM 的 Watson 计算机赢得了 Jeopardy 智力挑战        |



# 1.Linux概述

## 1.1 Linux简介

- GPL是GNU的公共许可证。
  - GNU来源于20世纪80年代初期，其创始人Richard Stallman坚持认为软件应该是“自由”的，软件业应该发扬开放、团结、互助的精神，而这些想法就催生了后来的GNU计划。
  - GNU计划，目标是创建一套完全自由的操作系统。Richard Stallman最早在net.UNIX-wizards新闻组上公布该消息，并附带一份《GNU宣言》解释为何发起该计划，其中一个理由就是要“重现当年软件界合作互助的团结精神”，旨在发展一个类似UNIX，且为自由软件的完整操作系统GNU系统。
    - 在GNU计划下诞生了文字编辑器Emacs、C语言编译器gcc以及一系列UNIX程序库和工具，1991年Linux加入GNU计划，这让GNU实现了最初的目标。
    - 完整的GPL协议可以在互联网上通过各种途径获取，如GNU的官网(<http://www.gnu.org>)。



# 1. Linux概

## 1.1 Linux简介

GNU 操作系统 - GNU 工程 - X

www.gnu.org/gnu/gnu.html

跳转到正文 设置语言 自由软件 搜索

English [en] العربية [ar] български [bg] català [ca] Deutsch [de] ελληνικά [el] español [es] français [fr] hrvatski [hr] italiano [it] 日本語 [ja] ಕನ್ನಡ [kn] 한국어 [ko] македонски [mk] മലയാളം [ml] norsk (bokmål) [nb] Nederlands [nl] polski [pl] português do Brasil [pt-br] русский [ru] Shqip [sq] српски [sr] українська [uk] 简体中文 [zh-cn] 繁體中文 [zh-tw]

GNU操作系统  
由自由软件基金会资助

加入 FSF  
自由软件支持者  
email address 注册

关于 GNU 哲学 许可证 教育 软件 文档 帮助 GNU

这是针对英文原版页面的中文翻译。

### GNU 操作系统

- 关于GNU操作系统
- GNU的历史概览
- 更详尽的GNU历史
- GNU工程的初始的声明
- GNU宣言
- BYTE杂志对 Richard Stallman 的采访 (1986)
- 我的Lisp经历和GNU Emacs的开发历程 (Richard Stallman 著)
- 一个人自由软件而战, 关于 Richard Stallman 和GNU的早期开发经历的文章, 由纽约时报 (The New York Times) 在1989年1月11日发表。该文章有一个问题, 就是使用了“知识产权”这个宣传词语, 好像在用一个有道理的词。其实, 该词非常令人困惑, 所以使用它没有意义。该文章谈及 Symbolics公司的时候也有些混乱。Stallman先生还在MIT工作时, 是在独立编写系统改进, 这些改进可以替代Symbolics公司在MIT Lisp Machine系统上做的类似改进。
- 自由软件15年 (1999)

这是 Stallman 在1983年5月访问斯坦福时写的两则海报。它们体现了 Stallman 当时已经在酝酿开始开发 GNU系统。里面没有“自由软件”的字眼; 显然, 他还没有开始把这两个词放在一起使用。

- 为什么应该分享软件 (1983)
- 是的, 把它贡献出来 (1983)

### GNU 和 Linux

- GNU 和 Linux的关系
- 为什么“Linux系统”应该叫做GNU/Linux
- 从未听说过GNU的GNU用户
- GNU/Linux问答

### 其他GNU相关的资源

- GNU/Linux、GNU/Hurd和自由软件用户组

下载发行版



如果你需要安装一个完整的系统, 请参看我们的完全自由的GNU/Linux发行版列表。

地址: <http://www.gnu.org>

返回顶部



# 1.Linux概述

## 1.2 Linux内核

- Linux由三部分组成：内核（Kernel）、外壳（Shell）、实用工具。
  - 内核是Linux的核心部分，从本质上说是一个软件，它提供了硬件抽象层、磁盘及文件系统控制、多任务等操作系统基本功能。
    - 单独的一个内核不是一个完整的操作系统，还需要配置相关的Shell和实用工具才能运行。
    - 网站<https://www.kernel.org>列出Linux内核的详细信息，可查看内核各个版本的修改记录，并可下载查阅内核源代码。
  - Linux内核是由C语言写成的，主要模块有进程调度、存储管理，虚拟文件系统、网络接口、进程通信等。



# 1.Linux概

## 1.2 Linux内核

The Linux Kernel Archives

About Contact us FAQ Releases Signatures Site news

Protocol Location

HTTP <https://www.kernel.org/pub/>

GIT <https://git.kernel.org/>

RSYNC <rsync://rsync.kernel.org/pub/>

Latest Stable Kernel:

4.15.7

| mainline:   | 4.16-rc4      | 2018-03-04 | [tarball] | [patch] | [inc. patch] | [view diff]  | [browse]                         |
|-------------|---------------|------------|-----------|---------|--------------|--------------|----------------------------------|
| stable:     | 4.15.7        | 2018-02-28 | [tarball] | [pgp]   | [patch]      | [inc. patch] | [view diff] [browse] [changelog] |
| longterm:   | 4.14.24       | 2018-03-03 | [tarball] | [pgp]   | [patch]      | [inc. patch] | [view diff] [browse] [changelog] |
| longterm:   | 4.9.86        | 2018-03-03 | [tarball] | [pgp]   | [patch]      | [inc. patch] | [view diff] [browse] [changelog] |
| longterm:   | 4.4.120       | 2018-03-03 | [tarball] | [pgp]   | [patch]      | [inc. patch] | [view diff] [browse] [changelog] |
| longterm:   | 4.1.49        | 2018-01-22 | [tarball] | [pgp]   | [patch]      | [inc. patch] | [view diff] [browse] [changelog] |
| longterm:   | 3.18.98 [EOL] | 2018-03-03 | [tarball] | [pgp]   | [patch]      | [inc. patch] | [view diff] [browse] [changelog] |
| longterm:   | 3.16.55       | 2018-03-03 | [tarball] | [pgp]   | [patch]      | [inc. patch] | [view diff] [browse] [changelog] |
| longterm:   | 3.2.100       | 2018-03-03 | [tarball] | [pgp]   | [patch]      | [inc. patch] | [view diff] [browse] [changelog] |
| linux-next: | next-20180306 | 2018-03-06 |           |         |              |              | [browse]                         |

Other resources

Cgit Bugzilla Mirrors Documentation Patchwork Linux.com Wikis Kernel Mailing Lists Linux Foundation

Social

Site Atom feed Releases Atom Feed Kernel Planet

This site is operated by the Linux Kernel Organization, Inc., a 501(c)3 nonprofit corporation, with support from the following sponsors.

redhat packet fastly Google

CONSTELLIX PURE STORAGE

THE LINUX FOUNDATION

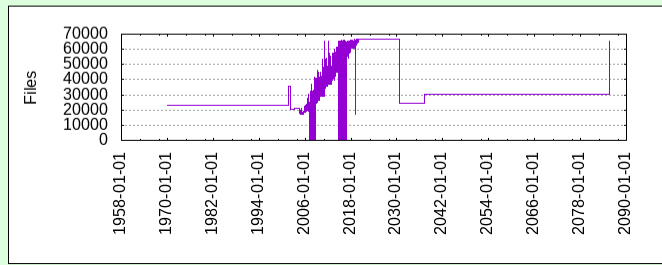
地址: <https://www.kernel.org>

## Files

General Activity Authors Files Lines Tags

**Total files:**  
66492  
**Total lines:**  
27852148  
**Average file size:**  
13448.10 bytes

### File count by date



### Extensions

| Extension  | Files (%)    | Lines (%)      | Lines/file |
|------------|--------------|----------------|------------|
|            | 5830 (8.77%) | 399628 (1.43%) | 68         |
| 0          | 6 (0.01%)    | 1679 (0.01%)   | 279        |
| 1          | 10 (0.02%)   | 1638 (0.01%)   | 163        |
| 1994-2004  | 1 (0.00%)    | 268 (0.00%)    | 268        |
| 1995-2002  | 1 (0.00%)    | 257 (0.00%)    | 257        |
| 1996-2002  | 1 (0.00%)    | 63 (0.00%)     | 63         |
| 8          | 8 (0.01%)    | 1506 (0.01%)   | 188        |
| ChangeLog  | 1 (0.00%)    | 44 (0.00%)     | 44         |
| DOC        | 1 (0.00%)    | 208 (0.00%)    | 208        |
| FlashPoint | 1 (0.00%)    | 60 (0.00%)     | 60         |
| Locking    | 1 (0.00%)    | 169 (0.00%)    | 169        |
| PL         | 1 (0.00%)    | 18 (0.00%)     | 18         |
| S          | 1326 (1.99%) | 375819 (1.35%) | 283        |
| SRC        | 1 (0.00%)    | 15 (0.00%)     | 15         |
| S_shipped  | 6 (0.01%)    | 9840 (0.04%)   | 1640       |
| WARNING    | 1 (0.00%)    | 4 (0.00%)      | 4          |
| ac         | 1 (0.00%)    | 111 (0.00%)    | 111        |
| aic79xx    | 1 (0.00%)    | 86 (0.00%)     | 86         |
| aic7xxx    | 1 (0.00%)    | 91 (0.00%)     | 91         |
| am         | 3 (0.00%)    | 31 (0.00%)     | 10         |

地址: <https://phoronix.com/misc/linux-eoy2019/files.html>

GitSats - linux

GitSats - linux

+

← → ↺ 🏠

🔒 https://phoronix.com/misc/linux-eoy2019/authors.html

🔍 📄 ⋮ ☆ ⬇ 📏 📄 👤 ↺ 🏠

Authors

General

Activity

Authors

Files

Lines

Tags

地址: https://phoronix.com/misc/linux-eoy2019/authors.html

List of Authors

| Author                | Commits (%)   | + lines | - lines | First commit | Last commit | Age                 | Active days | # by commits |
|-----------------------|---------------|---------|---------|--------------|-------------|---------------------|-------------|--------------|
| Linus Torvalds        | 28815 (3.25%) | 6775122 | 75895   | 2005-04-16   | 2019-12-31  | 5371 days, 20:54:22 | 4369        | 1            |
| David S. Miller       | 12212 (1.38%) | 187314  | 161370  | 2005-04-16   | 2019-12-29  | 5369 days, 22:05:13 | 3440        | 2            |
| Mark Brown            | 7373 (0.83%)  | 263833  | 121037  | 2006-02-01   | 2019-11-22  | 5041 days, 19:56:36 | 1744        | 3            |
| Takashi Iwai          | 7067 (0.80%)  | 224238  | 266693  | 2005-03-24   | 2019-12-17  | 5380 days, 20:30:02 | 2212        | 4            |
| Arnd Bergmann         | 7003 (0.79%)  | 76759   | 555085  | 2005-05-18   | 2019-12-16  | 5325 days, 0:35:50  | 1237        | 5            |
| Al Viro               | 6711 (0.76%)  | 121249  | 162783  | 2005-04-03   | 2019-11-12  | 5336 days, 12:57:14 | 1480        | 6            |
| Mauro Carvalho Chehab | 6259 (0.70%)  | 599401  | 545384  | 2005-06-24   | 2019-11-26  | 5268 days, 8:35:04  | 1472        | 7            |
| H Hartley Sweeten     | 5942 (0.67%)  | 127904  | 200103  | 2009-03-31   | 2017-09-03  | 3077 days, 19:19:59 | 405         | 8            |
| Ingo Molnar           | 5786 (0.65%)  | 129827  | 101816  | 2005-04-18   | 2019-12-17  | 5355 days, 17:29:26 | 1826        | 9            |
| Greg Kroah-Hartman    | 5786 (0.65%)  | 919955  | 1696150 | 2002-04-09   | 2019-12-16  | 6459 days, 13:58:09 | 1573        | 10           |
| Christoph Hellwig     | 5525 (0.62%)  | 207354  | 383629  | 2005-04-03   | 2019-12-04  | 5357 days, 21:41:08 | 1264        | 11           |
| Chris Wilson          | 5160 (0.58%)  | 185917  | 130318  | 2009-01-30   | 2019-12-18  | 3973 days, 13:30:21 | 1457        | 12           |
| Thomas Gleixner       | 5058 (0.57%)  | 134793  | 245232  | 2005-01-18   | 2019-11-20  | 5419 days, 7:25:24  | 1158        | 13           |
| Russell King          | 4803 (0.54%)  | 200337  | 166685  | 2005-01-24   | 2019-12-19  | 5441 days, 23:34:58 | 1537        | 14           |
| Johannes Berg         | 4458 (0.50%)  | 233450  | 161415  | 2005-12-10   | 2019-11-22  | 5094 days, 9:00:52  | 1510        | 15           |
| Tejun Heo             | 4209 (0.47%)  | 161635  | 124071  | 2005-04-03   | 2019-12-16  | 5370 days, 1:38:54  | 1070        | 16           |
| Geert Uytterhoeven    | 4199 (0.47%)  | 103295  | 76898   | 2005-04-18   | 2019-12-23  | 5361 days, 16:15:48 | 1172        | 17           |
| Hans Verkuil          | 4066 (0.46%)  | 251330  | 257702  | 2005-11-09   | 2019-12-02  | 5136 days, 12:31:28 | 1162        | 18           |
| Dan Carpenter         | 3741 (0.42%)  | 13045   | 14320   | 2006-11-20   | 2019-12-13  | 4771 days, 3:14:53  | 1436        | 19           |
| Eric Dumazet          | 3695 (0.42%)  | 68107   | 54203   | 2005-06-22   | 2019-12-23  | 5296 days, 21:40:33 | 1739        | 20           |

These didn't make it to the top: Arnaldo Carvalho de Melo, Rafael J. Wysocki, Kuninori Morimoto, Dave Airlie, Olof Johansson, Alex Deucher, Linus Walleij, Daniel Vetter, Trond Myklebust, Laurent Pinchart, Joe Perches, Ville Syrjälä, Ben Skeggs, Axel Lin, Peter Zijlstra, Colin Ian King, Paul Mundt, David Howells, Paul E. McKenney, Bartlomiej Zolnierkiewicz, Vans de Goede, Bjorn Helgaas, Lars-Peter Clausen, Andy Shevchenko, Wolfgang Sang, Julia Lawall, Ralf Baechle, Fabio Estevam, Johan Hovold, NeilBrown, Randolph Dunlap, Tony Lindgren, Sachin Kamat, Alan Cox, Thierry Reding, Jens Axboe, Masahiro Yamada, Eric W. Biederman, Stephen Hemminger, Andi Kleen, Hresh Kumar, Wei Yongjun, Adrian Bunk, Jiri Olsa, Jean Delvare, Guenter Roeck, Fabio Fainelli, Andrew Morton, Herbert Xu, Benjamin Herrenschmidt, Dan Williams, Stephen Boyd, Christian König, Magnus Damm, Ben Dooks, Kees Cook, Chuck Lever, Jan Kara, Felipe Balbi, Michael Ellerman, Borislav Petkov, Bart Van Assche, Steven Rostedt, Maxime Ripard, Tomi Valkeinen, Joerg Roedel, Peter Ujfalusi, Patrick McHardy, Jeff Layton, Ian Abbott, Ben Hutchings, Oleg Nesterov, Krzysztof Kozlowski, Marc Zyngier, Gustavo A. R. Silva, Jiri Slaby, Will Deacon, James Smart, Jes Sorensen, Larry Finger, Uwe Kleine-König, Jeff Garzik, Alexandre Belloni, Adrian Hunter, Alexander Duyck, David Woodhouse, Len Brown, John W. Linville, Alan Stern, Rob Herring, J. Bruce Fields, YueHaibing, Yinghai Lu, Namhyung Kim, Felix Fietkau, Jiri Pirkko, Paul Mackerras, Alexey Dobriyan, Sergei Shtylyov, Rafal Milecki, Dave Chinner, Daniel Borkmann, Boris Brezillon, Jani Nikula, Antti Palosaari, Mike Frysinger, Emmanuel Grumbach, Malcolm Priestley, Johan Hedberg, Chris Mason, Paul Gortmaker, Markus Elfring, Jakub Kicinski, Alex Elder, Steve French, Marcel Holtmann, Lee Jones, Frederic Weisbecker, Florian Westphal, Guennadi Liakhovetski, Avi Kivity, Philipp Zabel, Sam Ravnborg, Josef Bacik, Rusty Russell, Matthew Wilcox, Michael Chan, Luis R. Rodriguez, Ard Biesheuvel, Javier Martinez Canillas, Darrick J. Wong, Jaegeuk Kim, Miklos Szeredi, Shawn Guo, Masami Hiramatsu, David Sterba, Simon Horman, Stephen Rothwell, David Ahern, Vinod Koul, Michael Krufky, Bob Moore, Jonathan Cameron, Theodore Ts'o, Sascha Hauer, K. Y. Srinivasan, Michal Simek, Roland Dreier, Arend van Spriel, Thomas Petazzoni, Chen-Yu Tsai, Martin Schwidefsky, Pablo Neira Ayuso, Aaro Koskinen, Thomas Winkler, Artem Bityutskiy, Stephen Warren, Anton Blanchard, Sebastian Andrezj Siewior, Jeremy Fitzhardinge, Paolo Bonzini, Andy Lutomiński, David Brownell, Aneesh Kumar K.V, Akinobu Mita, Christoph Lameter, Greg Ungerer, Chao Yu, Marek Szymkowski, Michael S. Tsirkin, Arvind Yadav, Paul Burton, Heiner Kallweit, Stanislaw Gruszka, Eric Biggers, Sylwester Nawrocki, H. Peter Anvin, Rex Zhu, Andrew Lunn, James Hogan, Bill Pemberton, Clemens Ladisch, Jiri Kosina, Stefan Richter, Kumar Gala, Nicolas Pitre, Fabian Frederick, Steven Whitehouse, Ulf Hansson, Takashi Sakamoto, Lorenzo Bianconi, Kalle Valo, Imre Deak, Maarten Lankhorst, FUJITA Tomonori, Catalin Marinas, Li Zefan, Brian Norris, Ido Schimmel, Lennert Buytenhek, Jesper Juhl, Sujith Manoharan, Grant Likely, Daniel Mack, Pavel Emelyanov, Hugh Dickins, Paul Zanolini, Gustavo Padovan, Ming Lei, Sage Weil, Sakari Ailius, James Bottomley, Charles Keepax, Ilya Dymov, Kirill A. Shutemov, Vineet Gupta, Jan Beulich, Mika Westerberg, Mike Rapoport, Mel Gorman, Mark Rutland, Kevin Hilman, Wey-Yi Guy, Paul Walmsley, Peter Hurley, Jesse Barnes, Hannes Reinecke, Sven Eckelmann, Harvey Harrison, Christophe Leroy, Jiang Liu, Nicholas Bellinger, Vivien Didelot, Neil Armstrong, Nicholas Piggini, Eric Paris, Alexander Aring, Andreas Gruenbacher, Anton Vorontsov, Ezequiel Garcia, Heiko Stuebner, Davidaney, stephen hemminger, Marek Vasut, Jacob Keller, Mikulas Patocka, Roel Kluijn, Sagi Grimmer, Alexei Starovoitov, Oliver Neukum, Lucas Stach, Dominik Brodowski, Archit Taneja, Xin Long, Dave Jones, Yoshihiro Shimoda, Masanari Iida, Yan, Zheng, Mike Christie, Johannes Weiner, Steven Rostedt (Red Hat), Srinivas Kandagatla, Bartosz Golaszewski, Helge Deller, Richard Weinberger, Bjorn Andersson, Alexander Shlyan, Konrad Rzeszutek, Wilk, Michal Kozior, Daniel Lezcano, Sudip Mukherjee, Jerome Brunet, Eric Sandeen, Sergio Paracuellos, Jeff Dike, Rob Clark, Shaohua Li, Chaehyun Lim, Nikolay Borisov, Michael Hennerich, Nicolas Ferre, Laxman Dewangan, Paul Bolle, Jarkko Nikula, Eliad Peller, Ben Widawsky, Nick Piggini, Eric Anholt, Jason Gunthorpe, Tero Kristo, Benjamin Tissoires, Sebastian Ott, Matt Carlson, John Stultz, Michal Hocko, Gregory CLEMENT, navin patidar, David Rientjes, Alexander Graf, Martin Blumenstingl, Jesse Brandeburg, Philipp Reisner, Christian Borntraeger, Jean-Francois Moine, Roger Quadros, Damien Lespiau, Dmitry Eremin-Solenikov, Atsushi Nemoto, Anson Huang, Tobias Klausner, Robert Richter, Christian Lamparter, Oleg Drokin, nand patidar, David Rientjes, Alexander Graf, Martin Blumenstingl, Jesse Brandeburg, Philipp Reisner, Christian Borntraeger, Jean-Francois Moine, Roger Quadros, Damien Lespiau, Dmitry Eremin-Solenikov, Chris Metcalf, Finn Thain, Kulkij Kim, Leo Kim, Gerrit Renker, Rodrigo Viji, Kishon Vijay Abraham I, Jan Engelhardt, Samuel Ortiz, Andrew Vasquez, Thomas Graf, Leon Romanovsky, Keith Busch, Ivo van Doorn, Or Gerlitz, Mateusz Kulikowski, Filippe Manana, Ajay Singh, Stefan Agner, Vasanthakumari Thiagarajan, Robert P. J. Day, Pekka Enberg, Xiu Guangrong, Maciej W. Rozycki, Peter Chen, Michael Buesch, Alexandre Courbot, David Hildenbrand, Jonathan Corbet, Antoine Tenart, John Fastabend, Davidlohr Bueso, Jan Kiszka, Rajkumar Manoharan, Gabor Juhas, Vitaly Kuznetsov, Pierre-Louis Bossart, Tomas Figa, Scott Wong, Benjamin Romer, Ido Bo, Josh Poimboeuf, Huang Ru, Julian Wiedmann, Arjan van de Ven, Cong Wang, Arik Nemtsov, Chanwoo Choi, Dave Hansen, Steven Rostedt (VMware), Shuah Khan, Robin Murphy, Brian Foster, Martin Schwidefsky, Jon Hunter, Bjju Das, YOSHIFUJI Hideaki, Sudeep Holia, Petr Machata, Jaswinder Singh Rajput, James Morris, Vlad Yasevich, Jeff Mahoney, David Herrmann, Krzysztof Helt, Matt Fleming, Cyrill Gorcunov, Sara Sharon, Ryusuke Konishi, Roland Vossen, Nishanth Menon, Zhang Ru, Stefan Wahren, Tom Herbert, Pavel Machek, Eric Miao, Andrey Sminov, Wu Fengguang, Sarah Sharp, Lai Jiangshan, Jouni Malinen, Alex Himamaron, John Chiffman, Gellang Tang, Santosh Shilimkar, Robert Jarzmik, Michael Neuling, Gleb Natapov, James Simmons, Evan Quan, Rabin Vincent, Jussi Kivilinna, Tetsuo Handa, Grygorii Strashko, Ondrej Zary, Nikolay Aleksandrov, Mika Kuoppala, Mark Kleine-Budde, Parav Pandit, Jason Wang, Bjorn Mork, Yuval Mintz, Vladimir Zapolskiy, Steven Toth, Allan Stephens, Ursula Braun, Alexander Shishkin, Sujith, Kay Sievers, Gerd Hoffmann, Steve Wise, Harry Wentland, Srinivas Pandrurava, Shannon Nelson, Sean Young, Suzuki K Poulose, Karl Mashev, Gu Wenruo, Nishka Durgut, Christopher Dall, Sonig Zhang, Michael Straube, Ian Campbell, Devin Heitsch, Roland McGrath, Cornelia Huck, Pavel Shilovsky, Ilpo Jarvinen, Mike Kirshner, Miao Xie, Gilad Ben-Yossef, Henrique de Moraes Holschuh, Francois Romieu, Marcelo Tosatti, Kirill Tkai, Matthew Garrett, Stephen M. Camper, Vivek Goyal, Jesper Nilsson, Sebastian Reichel, Wanpeng Li, Shawn Lin, Heiko Mueller, Brian King, Sean Paul, Jesper Dangaard Brouer, Wang Nan, Michael Lanel, Kent Overstreet, KAMEZAWA Hirokyu, Corey Minyard, Andrzej Pietrasiewicz, Mike Isely, Dave Martin, WANG Cong, Steffen Klassert, Jon Mason, John Whitmore, Sean Christopherson, Liam Girdwood, Dmitry Osipenko, Glauber Costa, Mitch Williams, Heikki Krogerus, Eduardo Valentin, Daniel Drake, Tao Ma, Timur Tabi, Christophe Ricard, Fabrizio Castro, Yoichi Yuasa, Noralf Trønnes, Christophe JAILLET,

# 1.Linux概述

## 1.2 Linux内核

- Linux内核源代码是公开的，任何人都可以对内核加以修改并发布给其他人使用，这需要对内核版本进行规范化的管理。
- Linux内核版本号有两种体系，一个是内核版本号，一个是发行版本号。
  - Linux内核版本号由r、x、y三组数字组成，目前Linux内核版本有稳定版和开发版。
    - ①第一个组数字r：当前发布的内核主版本号。
    - ②第二个组数字x：偶数表示稳定版，奇数表示开发版。
    - ③第三个组数字y：错误修补的次数。
  - 例如：3.10.0 (r.x.y)：主版本号r=3；次版本号x=10，表示稳定版；错误修补的次数y=0；前两个数字（主、次版本号）组合可以描述内核系列，如稳定版的3.10.0，它是3.10版内核系列。



```
root@CentOS7Teach:/  
[root@CentOS7Teach /]# uname  
Linux  
[root@CentOS7Teach /]# uname -r  
3.10.0-693.17.1.el7.x86_64  
[root@CentOS7Teach /]# uname -a  
Linux CentOS7Teach 3.10.0-693.17.1.el7.x86_64 #1 SMP Thu Jan 25 20:13:58 UTC 201  
8 x86_64 x86_64 x86_64 GNU/Linux  
[root@CentOS7Teach /]# uname -m  
x86_64  
[root@CentOS7Teach /]#
```

```
administrator@UbuntuServer1710: ~  
administrator@UbuntuServer1710:~$ uname  
Linux  
administrator@UbuntuServer1710:~$ uname -r  
4.13.0-21-generic  
administrator@UbuntuServer1710:~$ uname -a  
Linux UbuntuServer1710 4.13.0-21-generic #24-Ubuntu SMP Mon Dec 18 17:29:16 UTC  
2017 x86_64 x86_64 x86_64 GNU/Linux  
administrator@UbuntuServer1710:~$ uname -m  
x86_64  
administrator@UbuntuServer1710:~$
```



# 1.Linux概述

## 1.2 Linux内核

- 内核是操作系统最基本的部分，它有多种不同的实现模式，其中主要有单内核和微内核，以及主要在科研领域使用的外内核。
- **单内核**
  - Linux大部分都是单内核的，单内核有一个较大的进程管理系统，属于集中式操作系统，它的内部又被分为用户程序、系统调用以及硬件控制三个层次和若干个模块。运行时每一个模块都是一个独立的二进制映像或者进程，并且可通过直接调用其他模块中的函数来实现通信。
  - 单内核的所有模块都在同一块寻址空间内运行，倘若某个模块有错误，运行时就会损及整个操作系统运行。如果单内核架构的操作系统在开发设计时相当完善，并经测试验证后具有高度可靠性，则操作系统内的各软件组件因具有高度紧密性，系统运行效率也会大幅提高。



# 1.Linux概述

## 1.2 Linux内核

- 内核是操作系统的最基本部分，它有多种不同的实现模式，其中主要有单内核和微内核，还有主要在科研系统里使用的外内核。
- **微内核**
  - 微内核出现于单内核之后，它仅将操作系统中最核心的功能写入内核，如线程管理、内存管理、地址空间、进程间通信等。微内核模式是面向对象理论在操作系统设计中的产物，其通过对系统逻辑功能的划分，将操作系统结构中的处理管理、文件管理、存储管理和设备管理等高级功能服务模块从内核中分离出来，成为在一定的特权方式下运行并且具有独立运行空间的独立进程，同时各个进程之间以消息机制来完成信息交换。
  - 微内核以模块化的方式完成操作系统中比较高级的操作，这样的设计使内核中最核心部分的设计更简单，增加了内核的灵活性，使之易于维护和移植。在微内核操作系统中，一个服务组件失效并不会导致整个系统崩溃，内核需要做的，仅仅是重新启动这个组件，而不影响其它的部分。





# 1.Linux概述

[ KB级内核 ]

[ 10-3毫瓦级功耗 ]

[ 微秒级响应 ]

[ 传感智能化 ]

降低时延

提升精确度

[ 联接智能化 ]

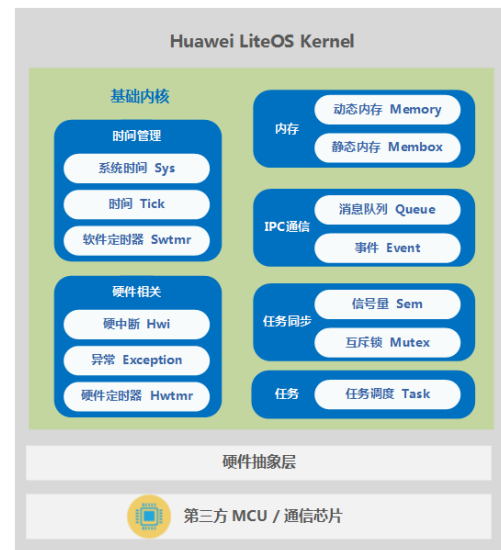
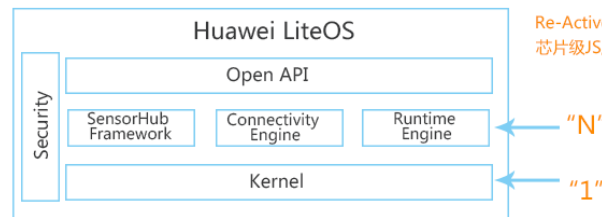
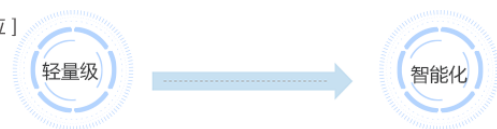
支持短距+长距多协议

自动组网

[ 应用智能化 ]

Re-Active 编程模型

芯片级JS虚拟机



<http://developer.huawei.com/ict/cn/site-iot/product/liteos>

# 1.Linux概述

## 1.2 Linux内核

- ❑ Linux Kernel是开源项目，经过数十年的发展，已经成为了全世界的财富，目前Linux Kernel主要由Linux基金会来负责维护。
- ❑ Linux基金会（Linux Foundation，简称：LF），是一家非营利性技术贸易协会，致力于促进、保护和推进Linux的协同开发，并支持“历史上最大的共享技术资源”。
  - 开始于2000年的开放源代码开发实验室（OSDL），并与自由标准组织（FSF）合并后从而成为现在的组织。
  - Linux基金会的会员越来越壮大，其中较著名的公司有AT&T、CISCO、FUJITSU、HITACHI、HUAWEI、IBM、Intel、Microsoft、NEC、Oracle、Qualcomm、SAMSUNG、VMWARE。
    - ❑ 关于Linux基金会的更多信息，可以访问官方网站  
<http://www.linuxfoundation.org>。



华为 物联网 操作系统\_百度搜... Home - The Linux Foundation X

www.linuxfoundation.org

HACTCM 棒棒堂 犀牛花 鲲鹏 MSDN CSDN 优酷 爱奇艺 虾米 果壳 译言 站酷 豆瓣 开源中国 教学演示 常用网址

THE LINUX FOUNDATION

Projects Membership Events Training Resources Newsroom About

The Linux Foundation Helps Hyperledger Build the Most Vibrant Open Source Ecosystem for Blockchain

LEARN MORE ABOUT HYPERLEDGER

Largest Shared Technology Investment

The Linux Foundation supports the creation of sustainable open source ecosystems by providing financial and intellectual resources, infrastructure, services, events, and training. Working together, The Linux Foundation and its projects form the most ambitious and successful investment in the creation of shared technology.

 **16B USD**  
Estimated development cost of the 100+ world's leading projects hosted at The Linux Foundation

 **25,000**  
Technologists attend our events annually, from more than 4,500 companies and 85 countries

 **1 Million**  
Open source professionals have enrolled in our free open source training courses

 **10 / 10**  
Largest cloud service providers are Linux Foundation project contributors and members

LEARN WHY COMPANIES WORK WITH THE LINUX FOUNDATION

地址: <http://www.linuxfoundation.org>

Upcoming Event

Open Networking Summit · March 26 – 29 · Los Angeles, CA

REGISTER FOR OPEN NETWORKING SUMMIT

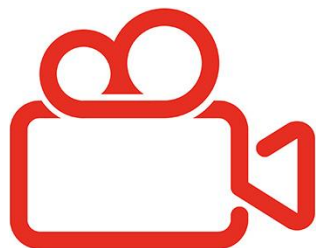
正在传输来自 www.linuxfoundation.org 的数据...

# 1.Linux概述

## 1.3 Linux家族

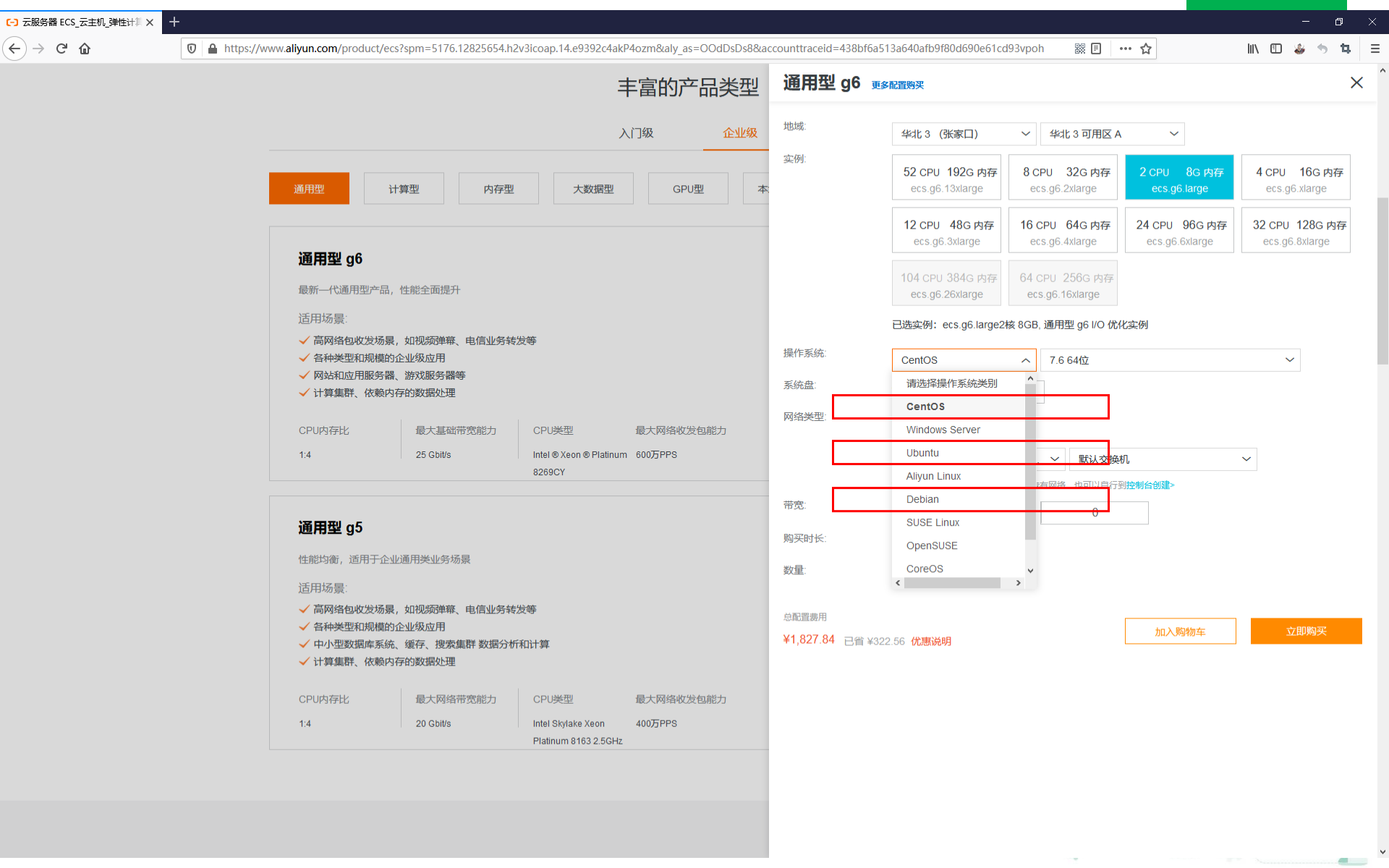
- Linux的开源特点让许多个人、组织和企业参与其中，他们将Linux系统的内核、应用软件和文档包装起来，并提供一些系统安装界面、系统配置设定管理工具，就构成了Linux发行版。
- 目前Linux发行版种类繁多，形成了一个庞大的Linux家族。





- ✓ 通过官网详细了解选用的操作系统







– The new CentOS Stream is a rolling-release distro that tracks just ahead of Red Hat Enterprise Linux (RHEL) development, positioned as a midstream between Fedora Linux and RHEL. For anyone interested in participating and collaborating in the RHEL ecosystem, CentOS Stream is your reliable platform for innovation.

## Get CentOS Now

## Around CentOS

- Friday Feb 7th : CentOS Blog: CentOS Community newsletter, February 2020 (#2002)
- Dear CentOS enthusiast, After a slowdown over the past few months, the year is off to a busy start. I'm getting the n
- Friday Feb 7th : CentOS Blog: Promo SIG: Quarterly report, Feb 2020
- Purpose The CentOS Promotion SIG exists to provide promotion, and consistent messaging, of CentOS, both at physical e

## News & Events



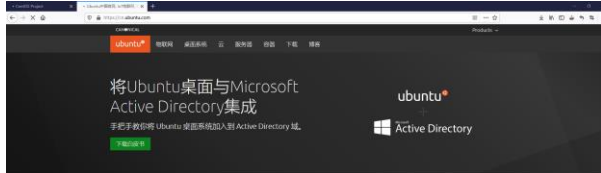
April 24: CentOS Dojo at Facebook

We have tentatively booked a CentOS Dojo at Facebook, in Menlo Park (San Francisco) California, on April 24th. Watch this space for more details.

## Sponsorship

CentOS would not be possible without the support of our sponsors. We would like to thank the following product/service for being a CentOS sponsor:





## Ubuntu桌面下载

### Ubuntu 18.04.3 LTS

下载专为桌面PC和笔记本精心打造的Ubuntu长期支持 (LTS) 版本。LTS意味着该版本将提供长期免费的安全更新维护支持至2023年4月。

[Ubuntu 18.04 LTS 发布公告](#)

#### 推荐系统配置

- 双核 2 GHz 处理器或更高
- 2 GB 系统内存
- 25 GB 磁盘存储空间
- 支持 DVD 光驱刻录或 USB 口安装
- 互联网接入

### Ubuntu 19.10

对于适用于台式机或笔记本电脑的Ubuntu系统的最新版本，19.10将具有9个月的安全维护更新支持期（至2020年7月）。

[Ubuntu 19.10 发布信息](#)

推荐系统配置与Ubuntu 18.04 LTS一致。

#### 快速下载

访问我们提供的Ubuntu 19.10下载、网络安装、ISO映像、请参见以下链接。

## 下载Ubuntu服务器版

### Ubuntu Server 18.04.3 LTS

Ubuntu服务器版的长期支持版本将包括OpenStack、容器和安全更新支持至2023年4月，仅面向64位平台。

此版本采用了全新的中间件——Ubuntu，它需要Ubuntu服务器版的系统库和依赖包。

[系统库和依赖包](#)

Ubuntu 18.04 LTS服务器版发行公告

如需将服务器认证或配置支持服务，请联系我们。

### Ubuntu Server 19.10

最新版本的Ubuntu Server，提供9个月的安全维护支持，支持期至2020年7月。

[Ubuntu Server 19.10 发布信息](#)

## 获取物联网版Ubuntu

Ubuntu Core是一个轻量级、严格受限、非桌面专用的操作系统。我们从头开始设计，专注于安全性和隐私保护，适用于边缘计算和嵌入式设备。Ubuntu Core使用了通用Linux内核和 systemd。

[了解更多关于Ubuntu Core的信息](#)

通过DVD或网络获取Ubuntu Core 18.04工作副本

### 支持的平台



# Ubuntu 18.04.3 LTS

下载专为桌面PC和笔记本精心打造的Ubuntu长期支持 (LTS) 版本，LTS意味着该版本将提供长期免费的安全更新维护支持至2023年4月。

[Ubuntu 18.04 LTS 发布公告](#)

推荐系统配置：

- ✓ 双核 2 GHz 处理器或更高
- ✓ 2 GB 系统内存
- ✓ 25 GB 磁盘存储空间
- ✓ 支持 DVD 光驱刻录或 USB 口安装
- ✓ 互联网接入

# Ubuntu 19.10

对于适用于台式机和笔记本电脑的Ubuntu系统的最新版本，19.10将具有9个月的安全维护更新支持期（至2020年7月）。

[Ubuntu 19.10 发布信息](#)

推荐系统配置与Ubuntu 18.04 LTS一致。



## 2.Linux的安装

### 2.1安装前准备

#### □ 了解操作系统所需的硬件需求

##### ■ Linux操作系统的硬件需求取决于所使用的发行版。

- Linux操作系统基本上兼容绝大多数计算机硬件，通常不需要考虑硬件兼容性问题。但由于硬件的发展速度很快，很难保证Linux与所有的硬件都能达到100%的兼容，因此在安装之前需要了解操作系统所需的硬件配置，相关发行版的硬件兼容性信息可通过官方网站获取。

##### ■ 例如：

##### □ Ubuntu官方硬件支持列表为：

- <https://wiki.ubuntu.com/HardwareSupport>

##### □ CentOS官方硬件支持列表为：

- <https://wiki.centos.org/AdditionalResources/HardwareList>



## 2.Linux的安装

### 2.1安装前准备

#### □ 了解操作系统的磁盘分区

- 在安装Linux操作系统之前，规划操作系统的磁盘分区很有必要。
  - 磁盘的分区主要有主分区（Primary partion）、扩展分区（Extension partion）和逻辑分区（Logical partion）。
  - Linux分区是通过将分区挂载到目录上实现对分区的访问，其分区方式根据系统的应用环境不同，各个分区的规划各不相同，好的分区规划可以有效提升系统的可用性。
- 对于初学者，建议使用默认分区方案以快速开始学习。



## 2.Linux的安装

### 2.1安装前准备

#### □ 了解操作系统的磁盘分区

表 1-2 Ubuntu Linux 文件系统的主要目录及其内容

| 目录     | 内容                     |
|--------|------------------------|
| /      | 根目录，主要用于存放系统           |
| /bin   | 存放系统的核心最常用的命令          |
| /boot  | 内核与启动文件                |
| /dev   | 设备文件                   |
| /home  | 用户主目录                  |
| /lib   | C 编译器的库                |
| /media | 加载各种的媒体，如光盘、U 盘等       |
| /mnt   | 用于加载各种文件系统             |
| /opt   | 存放第三方软件                |
| /root  | 超级用户 root 的目录          |
| /sbin  | 用于存放系统专用的二进制命令         |
| /proc  | 系统运行与进程信息，是内存映射而不是真实目录 |
| /tmp   | 存放临时文件                 |
| /usr   | 非系统的程序和命令              |
| /var   | 数据目录                   |

表 1-3 文件颜色及其含义

| 颜色   | 含义       |
|------|----------|
| 蓝色   | 目录       |
| 绿色   | 可执行文件    |
| 浅蓝色  | 链接文件     |
| 红色闪烁 | 链接的文件有问题 |
| 红色   | 压缩文件     |
| 黄色   | 设备文件     |
| 灰色   | 其他文件     |

```

root@CentOS7Teach/
login as: root
root@211.69.35.213's password:
Last failed login: Tue Mar  6 22:07:36 CST 2018 from 10.10.0.1 on ssh:notty
There were 148948 failed login attempts since the last successful login.
Last login: Thu Mar  1 15:41:58 2018 from 10.10.0.1
[root@CentOS7Teach ~]# cd /
[root@CentOS7Teach /]# ls
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var
boot  etc  lib  media  opt  root  sbin  sys  usr
[root@CentOS7Teach /]#
  
```



## 2.Linux的安装

### 2.1安装前准备

#### □ 获取Linux操作系统安装文件

##### ■ 获取渠道：

- 官网：<http://www.centos.org> <http://www.ubuntu.com>
- 镜像服务：<http://mirrors.aliyun.com> <http://mirrors.163.com>  
<http://mirrors.sohu.com> <http://mirrors.ustc.edu.cn>

##### ■ 获取版本：

- CentOS版本：DVD、Everything、LiveGNOME、LiveKDE、Minimal、NetInstall
- Ubuntu版本：desktop-amd64、server-amd64、desktop-i386、server-i386



## 2.Linux的安装

### 2.2Linux安装的三种方式

- Linux操作系统的安装方式灵活多样，可根据环境的不同而选择不同的安装方式，常见的安装方式有光盘安装、使用U盘安装、基于网络安装等。
  - 使用光盘安装
    - 通过光盘方式安装Linux操作系统是最常见的方式且简单易懂，但前提是所安装的PC和服务器的光驱或者外置光驱。
  - 使用U盘安装
    - 现在很多服务器和PC都不配备光驱，通过U盘安装是最为快速、廉价、高效的首选安装方式。
    - 推荐此种方式。
  - 基于网络安装

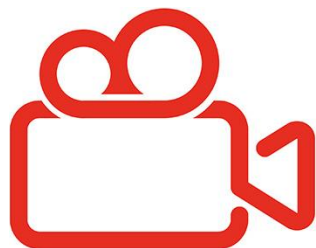


## 2.Linux的安装

### 2.3使用U盘安装CentOS

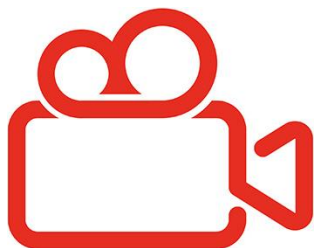
- 安装顺序：
  - 准备阶段
    - 下载CentOS 7操作系统
    - 下载Universal USB Installer软件
    - 制作U盘安装盘
  - 安装操作系统
    - 教学过程中：使用VirtualBox演示
  - 首次登录





- ✓ 基于VirtualBox安装CentOS 7 用于桌面应用
  - ✓ Universal USB Installer制作CentOS 7 U盘安装盘
  - ✓ 使用VirtualBox环境创建虚拟机
  - ✓ 在虚拟机上安装CentOS 7操作系统 (GNOME)
  - ✓ 首次登录CentOS 7操作系统





- ✓ 基于VirtualBox安装CentOS 7用于服务器应用
  - ✓ Universal USB Installer制作CentOS 7 U盘安装盘
  - ✓ 使用VirtualBox环境创建虚拟机
  - ✓ 在虚拟机上安装CentOS 7操作系统（最小化）
  - ✓ 首次登录CentOS 7操作系统



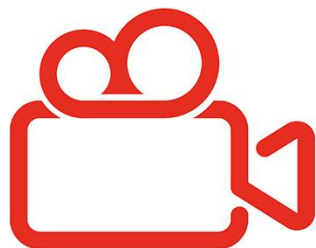


## 2.Linux的安装

### 2.4基于VirtualBox安装Ubuntu

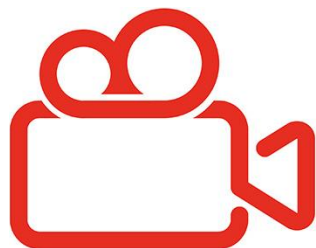
- 安装顺序：
  - 准备阶段
    - 下载Ubuntu Server 18.10操作系统
    - 下载VirtualBox软件并安装
    - 创建虚拟主机，并挂载ISO为光驱
  - 安装操作系统
    - 教学过程中：使用VirtualBox演示
  - 首次登录





- ✓ 基于VirtualBox安装Ubuntu Desktop
  - ✓ 使用VirtualBox环境创建虚拟机
  - ✓ 在虚拟机上安装Ubuntu Desktop 19.10操作系统
  - ✓ 首次登录Ubuntu Desktop 19.10操作系统





- ✓ 基于VirtualBox安装Ubuntu Server
  - ✓ 使用VirtualBox环境创建虚拟机
  - ✓ 在虚拟机上安装Ubuntu Server 19.10操作系统
  - ✓ 首次登录Ubuntu Server 19.10操作系统

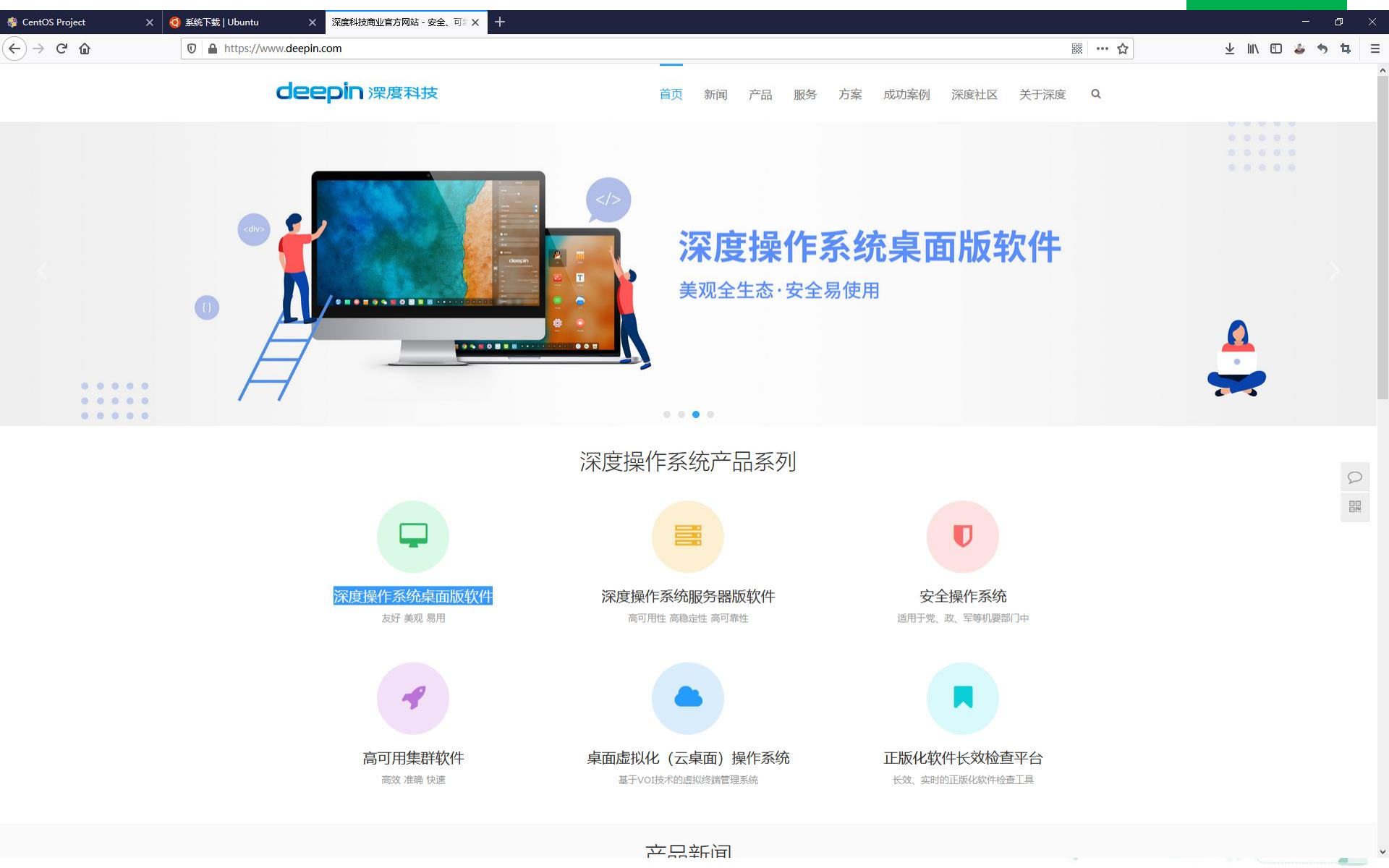


## 2.Linux的安装

### 2.5基于VirtualBox安装Deepin Linux

- 安装顺序：
  - 准备阶段
    - 下载深度操作系统
    - 下载VirtualBox软件并安装
    - 创建虚拟主机，并挂载ISO为光驱
  - 安装操作系统
    - 教学过程中：使用VirtualBox演示
  - 首次登录





## 深度操作系统桌面版软件

美观全生态·安全易使用



### 深度操作系统产品系列

  
**深度操作系统桌面版软件**  
友好 美观 易用

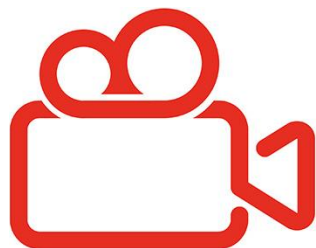
  
**深度操作系统服务器版软件**  
高可用性 高稳定性 高可靠性

  
**安全操作系统**  
适用于党、政、军等机要部门中

  
**高可用集群软件**  
高效 准确 快速

  
**桌面虚拟化（云桌面）操作系统**  
基于VOI技术的虚拟终端管理系统

  
**正版化软件长效检查平台**  
长效、实时的正版化软件检查工具



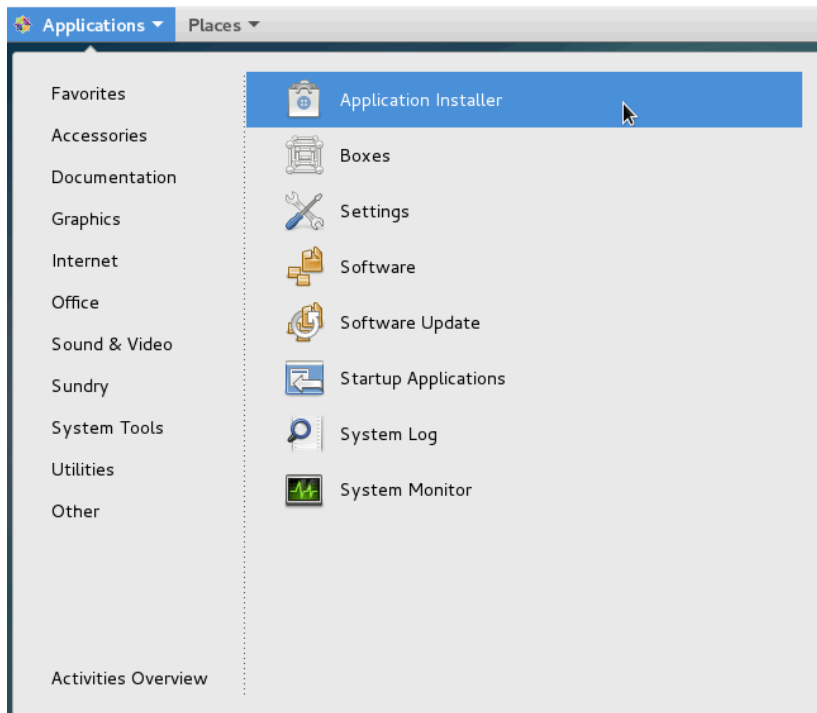
- ✓ 基于VirtualBox安装深度操作系统桌面版软件
  - ✓ 使用VirtualBox环境创建虚拟机
  - ✓ 在虚拟机上安装深度操作系统桌面版软件
  - ✓ 首次登录

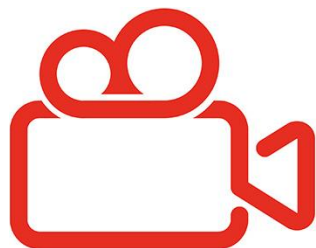


## 3.操作系统维护

### 3.1 CentOS系统维护

#### □ 通过Application Installer安装应用软件





✓ CentOS系统的基本维护

- CentOS Application Installer的基本功能
- CentOS Application Installer进行软件安装
- 安装QQ for Linux
- 安装办公软件OpenOffice





## 3.操作系统维护

### 3.1 CentOS系统维护

#### □ RPM

- RPM是RedHat Package Manager的缩写，是由RedHat公司开发的软件包安装和管理程序。用户可以使用RPM自行安装和管理Linux上的应用程序和系统工具。
- RPM可以让用户直接以二进制的方式安装软件包，并且可替用户查询是否已经安装了有关的库文件；如果使用RPM来升级软件，RPM会保留原有的配置文件，升级后的软件不用重新配置。
- RPM保留一个数据库，这个数据库中包含了所有的软件包的资料，通过这个数据库，用户可以进行软件包的查询。RPM共有多种模式，如安装、查询、验证、删除等。



## 3.操作系统维护

### □ RPM

#### 【语法】

```
rpm [选项] [参数]
```

#### 【选项说明】

rpm 命令的选项及说明，如表 1-6 所示。

表 1-6 rpm 命令的选项及其说明

| 选项            | 说明                      |
|---------------|-------------------------|
| -a            | 查询所有套件                  |
| -b<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               |
| -d            | 只列出文本文件                 |
| -e<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            | 套件安装时列出标记               |
| -i            | 显示套件的相关信息               |
| -l            | 显示套件的文件列表               |
| -p<套件档>       | 查询指定的 RPM 套件档           |
| -q            | 遇到问题询问用户                |
| -R            | 显示套件的关联性信息              |
| -s            | 显示文件状态                  |
| -U<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         |

#### 【参数说明】

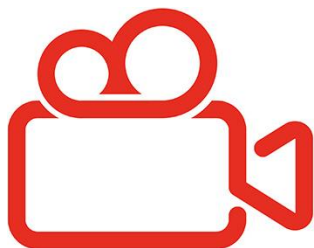
rpm 命令的选项及说明，如表 1-7 所示。

表 1-7 rpm 命令的参数及其说明

| 选项  | 说明             |
|-----|----------------|
| 软件包 | 指定要操纵的 rpm 软件包 |

IS系统维护





- ✓ CentOS上通过Shell安装MySQL
  - 第1步：下载MySQL Community RPM安装包
  - 第2步：安装
  - 第3步：测试MySQL服务



## 3.操作系统维护

### 3.1 CentOS系统维护

#### □ YUM

- YUM即Yellow dog Updater Modified，是一个Shell软件包管理器。yum基于RPM包管理，能够从指定的服务器自动下载RPM包进行安装，安装过程中可自动处理依赖关系，并且一次安装所有依赖的软件包。
- yum提供了查找、安装、删除某一个、一组甚至全部软件包的命令。
- yum可以同时配置多个资源库（Repository）；有着简洁的配置文件（/etc/yum.conf）；同时保持与RPM数据库的一致性，有一个比较详细的log，可以查看何时安装了什么软件包等。



## 3.操作系统维护

### 3.1 CentOS系统维护

#### □ YUM

##### 【语法】

```
yum [选项] [参数]
```

##### 【选项说明】

yum 命令的选项及说明，如表 1-8 所示。

表 1-8 yum 命令的选项及其说明

| 选项 | 说明                     |
|----|------------------------|
| -h | 显示帮助信息                 |
| -y | 对所有的询问均回答“yes”         |
| -c | 指定配置文件                 |
| -q | 静音模式                   |
| -v | 显示其详细信息                |
| -d | 设置调试等级（0~10）           |
| -e | 设置错误等级（0~10）           |
| -c | 配置文件位置                 |
| -C | 完全从缓存中运行，但不下载或者更新任何头文件 |
| -R | 设置命令的最大等待时间            |



## 3.操作系统

### □ YUM

表 1-9 常用的功能类命令

| 命令                            | 说明          |
|-------------------------------|-------------|
| yum install yum-fastestmirror | 自动搜索最快镜像插件  |
| yum install                   | 安装软件        |
| yum groupinstall              | 查看可能批量安装的列表 |

表 1-10 安装类命令

| 命令                     | 说明               |
|------------------------|------------------|
| yum install            | 全部安装             |
| yum install package    | 安装指定的安装包 package |
| yum groupinstall group | 安装程序组 group      |

表 1-11 查找和显示类命令

| 命令                        | 说明                             |
|---------------------------|--------------------------------|
| yum search string         | 根据关键字 string 查找安装包             |
| yum deplist package       | 查看程序 package 以来关系              |
| yum info package          | 显示安装包 package 信息               |
| yum list                  | 显示所有已安装和可安装的程序包                |
| yum list installed        | 查看已安装的程序包                      |
| yum list package          | 显示指定程序包 package 安装情况           |
| yum list all              | 列出 yum 仓库中的所有软件                |
| yum groupinfo group       | 显示软件组 group 的信息                |
| yum provides nginx        | 查看软件的功能                        |
| yum whatprovides filename | 查找包含指定的目标文件 filename 的 rpm 文件包 |



## 3.操作系统

### □ YUM

表 1-12 更新类命令

| 命令                                 | 说明              |
|------------------------------------|-----------------|
| yum check-update                   | 检查可更新的程序        |
| yum update                         | 全部更新            |
| yum update package                 | 更新指定程序包 package |
| yum localupdate update-package.rpm | 升级本地包           |
| yum groupupdate group              | 升级程序组 group     |

表 1-13 卸载类命令

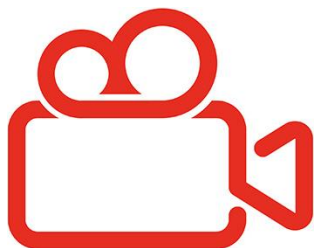
| 命令                    | 说明                |
|-----------------------|-------------------|
| yum remove package    | 删除程序包 package     |
| yum groupremove group | 删除程序组 group       |
| yum delist package    | 查看程序 package 依赖情况 |

表 1-14 清除与缓存类命令

| 命令                   | 说明                     |
|----------------------|------------------------|
| yum clean packages   | 清除缓存目录下的程序包 packages   |
| yum clean headers    | 清除缓存目录下的 headers       |
| yum clean oldheaders | 清除缓存目录下旧的 headers      |
| yun clean all        | 清除缓存目录下的软件包及旧的 headers |

1 CentOS系统维护





- ✓ CentOS YUM应用
  - 使用YUM安装软件：VLC Media Player
  - 通过YUM进行操作系统更新





## 3.操作系统维护

### 3.2 Ubuntu系统维护

#### □ APT

- apt即Advanced Package Tool，意为“高级软件包工具”。
- apt是一个命令行包管理器，同时也提供命令搜索和管理，以及查询有关软件包信息的功能。它能够在发现软件包依赖性的同时自动下载相应的软件包并加以安装。apt使得管理员能毫无故障地执行系统软件的升级。
- Ubuntu采用集中式的软件仓库机制，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。所有的Ubuntu用户可使用APT获得最新版本的安装软件包。



## 3.操作系统维护

### 3.2 Ubuntu系统维护

#### □ apt相关的文件

表 1-15 apt 相关的文件或目录及其内容

|                                 |                |
|---------------------------------|----------------|
| /etc/apt/sources.list           | 设置软件包的获取来源     |
| /etc/apt/apt.conf               | APT 的配置文件      |
| /etc/apt/apt.conf.d/            | APT 的零碎配置文件    |
| /etc/apt/preferences            | 版本参数           |
| /var/cache/apt/archives/        | 存放已经下载的软件包     |
| /var/cache/apt/archives/partial | 存放正在下载的软件包     |
| /var/lib/apt/lists/             | 存放已经下载的软件包详细信息 |
| /var/lib/apt/lists/partial/     | 存放正在下载的软件包详细信息 |



## 3.操作系统维护

### 3.2 Ubuntu系统维护

#### □ apt的软件仓库

- ubuntu官方公布的软件仓库源: <https://wiki.ubuntu.com/Mirrors>
- 国内知名的软件仓库: <http://mirrors.aliyun.com> <http://mirrors.163.com>  
<http://mirrors.sohu.com> <http://mirrors.ustc.edu.cn>

#### □ apt的软件仓库配置

- /etc/apt/sources.list文件定义了软件仓库
- 通过修改该文件, 可以为ubuntu指定更快速的软件仓库



apt-get [选项] [子命令] [参数]

### 3.操作系统维护

apt-get 命令的选项及说明，如表 1-16 所示。

表 1-16 apt-get 命令选项及其说明

| 选项            | 说明                                 |
|---------------|------------------------------------|
| -b            | 在源码包下载完成后进行编译                      |
| -d            | 仅下载，不安装                            |
| -f            | 修复依赖问题                             |
| -m            | 遇到无法下载的软件包，自动忽略                    |
| --no-download | 禁止下载软件包，其与-m 结合，可使 apt 只使用已经下载的软件包 |
| -q            | 静默模式，输出的信息适合做日志                    |
| -s            | 模拟测试，不做出实际操作，不改变系统                 |
| -y            | 在系统询问时，自动应答为 yes                   |
| -u            | 显示已升级的软件包                          |
| -V            | 显示已安装和升级的软件包的完整版本号                 |
| --ignore-hold | 忽略被保留的软件包                          |
| --no-upgrade  | 不要升级软件包                            |
| --force-yes   | 强制回答 yes                           |
| --print-uris  | 仅答应软件包地址，不安装                       |
| --purge       | 彻底删除，包括配置文件                        |
| --reinstall   | 重新安装软件包                            |

#### □ apt-get



## 3. 操作系统维护

### □ apt-get

#### 【子命令说明】

apt-get 命令的选项及说明，如表 1-17 所示。

表 1-17 apt-get 的子命令参数

| 命令              | 说明                      |
|-----------------|-------------------------|
| update          | 更新软件包列表                 |
| upgrade         | 升级系统中的所有软件包             |
| install         | 安装软件包                   |
| remove          | 卸载软件包                   |
| autoremove      | 自动删除所有未使用的包             |
| purge           | 彻底删除软件包及其配置文件           |
| source          | 下载源码包                   |
| build-dep       | 自动下载编译过程中需要用到的软件包并将其配置好 |
| dist-upgrade    | 升级整个发行版                 |
| dselect-upgrade | 使用 dselect 升级           |
| clean           | 删除本次缓存的所有软件包            |
| autoclean       | 删除本地缓存的无用的软件包           |
| check           | 检查是否存在有问题的依赖关系          |
| download        | 将二进制包下载到当前目录            |
| changelog       | 下载并显示给定包的更改日志           |

#### 【参数说明】

apt-get 命令的选项及说明，如表 1-18 所示。

表 1-18 apt-get 的参数说明

| 参数  | 说明         |
|-----|------------|
| 软件包 | 指定下载的软件包名称 |



## 3.操作系统维护

### 【语法】

```
apt-cache [选项] [参数]
```

### 【选项说明】

apt-cache 命令的选项说明，如表 1-19 所示。

表 1-19 apt-cache 的子命令及其说明

| 命令       | 说明                     |
|----------|------------------------|
| showsrc  | 显示软件包的信息，包括安装状态和包依赖关系等 |
| search   | 搜索软件包                  |
| depends  | 显示包的原始依赖关系信息           |
| rdepends | 显示包的反向依赖关系信息           |
| show     | 显示包的可读记录               |
| pkgname  | 列出系统中所有包的名称            |
| policy   | 显示策略设置                 |

### 【参数说明】

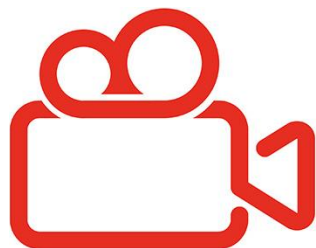
apt-cache 命令参数及说明，如表 1-20 所示。

表 1-20 apt-cache 命令参数说明

| 参数  | 说明          |
|-----|-------------|
| 软件名 | 指定所需下载的文件名称 |

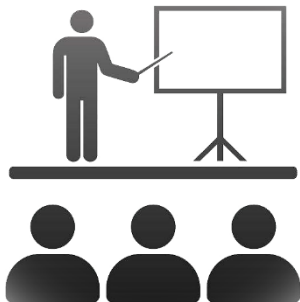
### □ apt-cache





- ✓ Ubuntu下的APT应用
  - 通过apt安装软件：sar
  - 通过apt进行操作系统的升级维护





- ✓ Ubuntu Desktop
  - 基于VirtualBox安装Ubuntu Desktop
  - Ubuntu One服务
  - Ubuntu Software Center进行软件管理
  - Ubuntu Desktop做开发：  
Netbeans、Virstual Studio Core  
MySQL Workbench、Chrome





