

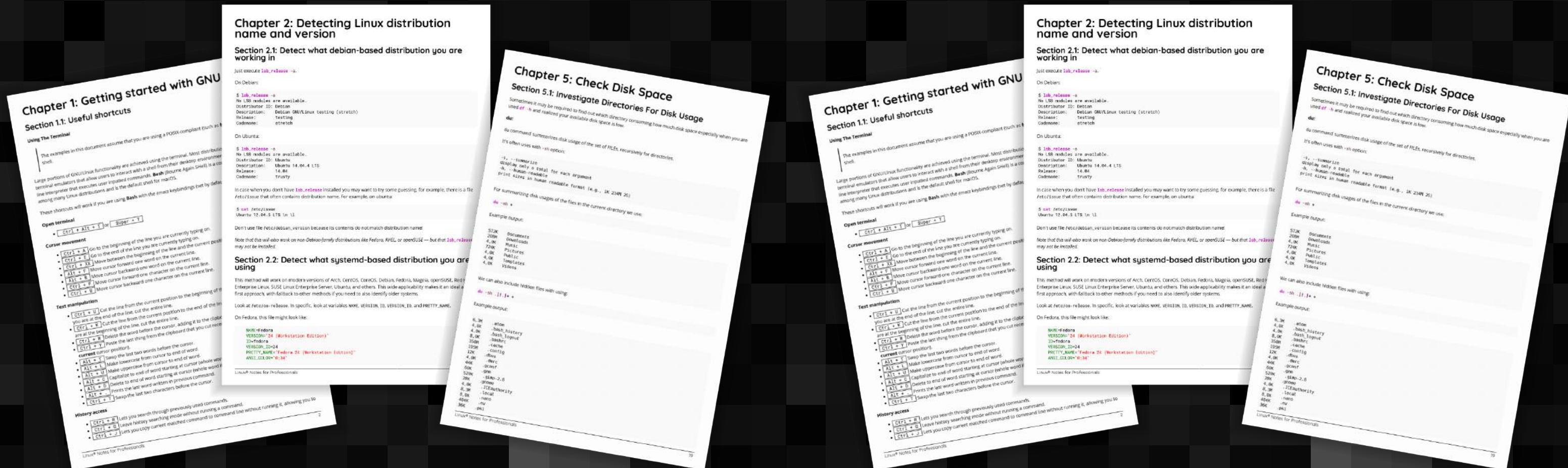
# Linux<sup>®</sup>

专业人士笔记

## 专业人士笔记

# Linux<sup>®</sup>

## Notes for Professionals



**50+ 页**  
专业提示和技巧

**50+ pages**  
of professional hints and tricks

# 目录

关于	1
第1章：GNU/Linux入门	2
第1.1节：实用快捷键	2
第1.2节：文件管理命令	3
第1.3节：你好，世界	5
第1.4节：基本的Linux工具	5
第1.5节：按名称/内容模式搜索文件	6
第1.6节：文件操作	7
第1.7节：文件/目录详情	8
第2章：检测Linux发行版名称和版本	11
第2.1节：检测你正在使用的基于Debian的发行版	11
第2.2节：检测您正在使用的基于systemd的发行版	11
第2.3节：检测您正在使用的RHEL / CentOS / Fedora发行版	12
第2.4节：Uname - 打印当前系统信息	13
第2.5节：检测您的发行版的基本信息	13
第2.6节：使用GNU coreutils	13
第2.7节：查找你的Linux操作系统（包括Debian和RPM）的名称和版本号	14
第3章：获取正在运行的Linux内核信息	15
第3.1节：获取Linux内核详细信息	15
第4章：Shell	16
第4.1节：更改默认Shell	16
第4.2节：基本Shell工具	17
第4.3节：创建自定义命令别名	18
第4.4节：在系统中定位文件	18
第5章：检查磁盘空间	19
第5.1节：调查目录的磁盘使用情况	19
第5.2节：检查磁盘空间	21
第6章：获取系统信息	23
第6.1节：关于CPU、内存、网络和磁盘（I/O操作）的统计信息	23
第6.2节：使用lscpu和lshw等工具	23
第6.3节：列出硬件	24
第6.4节：查找CPU型号/速度信息	25
第6.5节：进程监控和信息收集	26
第7章：ls命令	28
第7.1节：ls命令的选项	28
第7.2节：带有最常用选项的ls命令	28
第8章：使用'tar'命令进行文件压缩	30
第8.1节：压缩文件夹	30
第8.2节：从归档文件中提取文件夹	30
第8.3节：列出归档文件内容	30
第8.4节：列出归档内容	31
第8.5节：压缩并排除一个或多个文件夹	31
第8.6节：剥离前导组件	31
第9章：服务	32
第9.1节：列出Ubuntu上运行的服务	32
第9.2节：Systemd服务管理	32

# Contents

About	1
Chapter 1: Getting started with GNU/Linux	2
Section 1.1: Useful shortcuts	2
Section 1.2: File Management Commands	3
Section 1.3: Hello World	5
Section 1.4: Basic Linux Utilities	5
Section 1.5: Searching for files by patterns in name/contents	6
Section 1.6: File Manipulation	7
Section 1.7: File/Directory details	8
Chapter 2: Detecting Linux distribution name and version	11
Section 2.1: Detect what debian-based distribution you are working in	11
Section 2.2: Detect what systemd-based distribution you are using	11
Section 2.3: Detect what RHEL / CentOS / Fedora distribution you are working in	12
Section 2.4: Uname - Print information about the current system	13
Section 2.5: Detect basic information about your distro	13
Section 2.6: Using GNU coreutils	13
Section 2.7: Find your linux os (both debian & rpm) name and release number	14
Chapter 3: Getting information on a running Linux kernel	15
Section 3.1: Getting details of Linux kernel	15
Chapter 4: Shell	16
Section 4.1: Changing default shell	16
Section 4.2: Basic Shell Utilities	17
Section 4.3: Create Your Own Command Alias	18
Section 4.4: Locate a file on your system	18
Chapter 5: Check Disk Space	19
Section 5.1: Investigate Directories For Disk Usage	19
Section 5.2: Checking Disk Space	21
Chapter 6: Getting System Information	23
Section 6.1: Statistics about CPU, Memory, Network and Disk (I/O operations)	23
Section 6.2: Using tools like lscpu and lshw	23
Section 6.3: List Hardware	24
Section 6.4: Find CPU model/speed information	25
Section 6.5: Process monitoring and information gathering	26
Chapter 7: ls command	28
Section 7.1: Options for ls command	28
Section 7.2: ls command with most used options	28
Chapter 8: File Compression with 'tar' command	30
Section 8.1: Compress a folder	30
Section 8.2: Extract a folder from an archive	30
Section 8.3: List contents of an archive	30
Section 8.4: List archive content	31
Section 8.5: Compress and exclude one or multiple folder	31
Section 8.6: Strip leading components	31
Chapter 9: Services	32
Section 9.1: List running service on Ubuntu	32
Section 9.2: Systemd service management	32

<b>第10章：管理服务</b>	33
第10.1节：诊断服务问题	33
第10.2节：启动和停止服务	33
第10.3节：获取服务状态	34
<b>第11章：修改用户</b>	35
第11.1节：设置自己的密码	35
第11.2节：设置其他用户的密码	35
第11.3节：添加用户	35
第11.4节：删除用户	35
第11.5节：删除用户及其主目录	35
第11.6节：列出当前用户所属的组	35
第11.7节：列出某用户所属的组	35
<b>第12章：LAMP环境</b>	36
第12.1节：在Arch Linux上安装LAMP	36
第12.2节：在Ubuntu上安装LAMP	37
第12.3节：在CentOS上安装LAMP环境	38
<b>第13章：tee命令</b>	40
第13.1节：将输出写入标准输出，同时写入文件	40
第13.2节：将管道链中间的输出写入文件，并传回管道	40
第13.3节：将输出写入多个文件	40
第13.4节：指示tee命令追加到文件	40
<b>第14章：安全外壳（SSH）</b>	42
第14.1节：连接到远程服务器	42
第14.2节：安装OpenSSH套件	42
第14.3节：配置SSH服务器以接受连接	43
第14.4节：无密码连接（使用密钥对）	43
第14.5节：生成公钥和私钥	43
第14.6节：禁用ssh服务	43
<b>第15章：SCP</b>	45
第15.1节：安全复制	45
第15.2节：基本用法	45
<b>第16章：GnuPG（GPG）</b>	46
第16.1节：导出你的公钥	46
第16.2节：快速创建和使用GnuPG密钥	46
<b>第17章：网络配置</b>	47
第17.1节：本地DNS解析	47
第17.2节：配置DNS服务器进行域名解析	47
第17.3节：查看和操作路由	47
第17.4节：为网络中的其他系统配置主机名	48
第17.5节：接口详情	49
第17.6节：向接口添加IP	50
<b>第18章：午夜指挥官</b>	52
第18.1节：午夜指挥官浏览模式下的功能键	52
第18.2节：Midnight Commander文件编辑模式下的功能键	52
<b>第19章：更改根目录（chroot）</b>	54
第19.1节：要求	54
第19.2节：在目录中手动更改根目录	54
第19.3节：使用chroot的原因	55
<b>第20章：包管理器</b>	56

<b>Chapter 10: Managing Services</b>	33
Section 10.1: Diagnosing a problem with a service	33
Section 10.2: Starting and Stopping Services	33
Section 10.3: Getting the status of a service	34
<b>Chapter 11: Modifying Users</b>	35
Section 11.1: Setting your own password	35
Section 11.2: Setting another user's password	35
Section 11.3: Adding a user	35
Section 11.4: Removing a user	35
Section 11.5: Removing a user and its home folder	35
Section 11.6: Listing groups the current user is in	35
Section 11.7: Listing groups a user is in	35
<b>Chapter 12: LAMP Stack</b>	36
Section 12.1: Installing LAMP on Arch Linux	36
Section 12.2: Installing LAMP on Ubuntu	37
Section 12.3: Installing LAMP stack on CentoOS	38
<b>Chapter 13: tee command</b>	40
Section 13.1: Write output to stdout, and also to a file	40
Section 13.2: Write output from the middle of a pipe chain to a file and pass it back to the pipe	40
Section 13.3: write the output to multiple files	40
Section 13.4: Instruct tee command to append to the file	40
<b>Chapter 14: Secure Shell (SSH)</b>	42
Section 14.1: Connecting to a remote server	42
Section 14.2: Installing OpenSSH suite	42
Section 14.3: Configuring an SSH server to accept connections	43
Section 14.4: Passwordless connection (using a key pair)	43
Section 14.5: Generate public and private key	43
Section 14.6: Disable ssh service	43
<b>Chapter 15: SCP</b>	45
Section 15.1: Secure Copy	45
Section 15.2: Basic Usage	45
<b>Chapter 16: GnuPG (GPG)</b>	46
Section 16.1: Exporting your public key	46
Section 16.2: Create and use a GnuPG key quickly	46
<b>Chapter 17: Network Configuration</b>	47
Section 17.1: Local DNS resolution	47
Section 17.2: Configure DNS servers for domain name resolution	47
Section 17.3: See and manipulate routes	47
Section 17.4: Configure a hostname for some other system on your network	48
Section 17.5: Interface details	49
Section 17.6: Adding IP to an interface	50
<b>Chapter 18: Midnight Commander</b>	52
Section 18.1: Midnight Commander function keys in browsing mode	52
Section 18.2: Midnight Commander function keys in file editing mode	52
<b>Chapter 19: Change root (chroot)</b>	54
Section 19.1: Requirements	54
Section 19.2: Manually changing root in a directory	54
Section 19.3: Reasons to use chroot	55
<b>Chapter 20: Package Managers</b>	56

第20.1节：如何使用apt包管理器更新软件包	56
第20.2节：如何使用pacman包管理器安装软件包	56
第20.3节：如何使用pacman包管理器更新软件包	56
第20.4节：如何使用yum更新软件包	57
<b>第21章：编译Linux内核</b>	58
第21.1节：在Ubuntu上编译Linux内核	58
<b>鸣谢</b>	59
<b>你可能也喜欢</b>	61

Section 20.1: How to update packages with the apt package manager	56
Section 20.2: How to install a package with the pacman package manager	56
Section 20.3: How to update packages with the pacman package manager	56
Section 20.4: How to update packages with yum	57
<b>Chapter 21: Compiling the Linux kernel</b>	58
Section 21.1: Compilation of Linux Kernel on Ubuntu	58
<b>Credits</b>	59
<b>You may also like</b>	61

# 关于

请随意免费分享此PDF，  
本书最新版本可从以下网址下载：  
<https://goalkicker.com/LinuxBook>

本Linux® 专业人士笔记一书汇编自Stack Overflow  
文档，内容由Stack Overflow的优秀人士撰写。  
文本内容采用知识共享署名-相同方式共享许可协议发布，详见本书末尾对各章节贡  
献者的致谢。图片版权归各自所有者所有，除非另有说明。

这是一本非官方的免费书籍，旨在教育用途，与官方Linux®组织或公司及Sta  
ck Overflow无关。所有商标和注册商标均为其各自公司所有者所有。

本书所提供的信息不保证正确或准确，使用风险自负。

请将反馈和更正发送至 [web@petercv.com](mailto:web@petercv.com)

# About

Please feel free to share this PDF with anyone for free,  
latest version of this book can be downloaded from:  
<https://goalkicker.com/LinuxBook>

This *Linux® Notes for Professionals* book is compiled from [Stack Overflow Documentation](#), the content is written by the beautiful people at Stack Overflow. Text content is released under Creative Commons BY-SA, see credits at the end of this book whom contributed to the various chapters. Images may be copyright of their respective owners unless otherwise specified

This is an unofficial free book created for educational purposes and is not affiliated with official Linux® group(s) or company(s) nor Stack Overflow. All trademarks and registered trademarks are the property of their respective company owners

The information presented in this book is not guaranteed to be correct nor accurate, use at your own risk

Please send feedback and corrections to [web@petercv.com](mailto:web@petercv.com)

# 第1章：GNU/Linux入门

## 第1.1节：实用快捷键

### 使用终端

本文档中的示例假设您正在使用符合POSIX标准的（例如bash、sh、zsh、ksh）shell。

GNU/Linux的大部分功能都是通过终端实现的。大多数Linux发行版都包含终端模拟器，允许用户从桌面环境与shell交互。shell是一种命令行解释器，用于执行用户输入的命令。Bash（Bourne Again SHell）是许多Linux发行版中常见的默认shell，也是macOS的默认shell。

如果您使用带有emacs键绑定（默认设置）的Bash，这些快捷键将有效：

### 打开终端

- Ctrl + Alt + T 或 Super + T

### 光标移动

- Ctrl + A 移动到当前正在输入的行的开头。
- Ctrl + E 移动到当前正在输入的行尾。
- Ctrl + XX 在行首和光标当前位置之间切换移动。
- Alt + F 将光标向前移动一个单词（当前行内）。
- Alt + B 将光标向后移动一个单词（当前行内）。
- Ctrl + F 将光标向前移动一个字符（当前行内）。
- Ctrl + B 将光标向当前行的前一个字符移动。

### 文本操作

- Ctrl + U 从当前位置剪切到行首，并将其添加到剪贴板。如果你在行尾，则剪切整行。
- Ctrl + K 从当前位置剪切到行尾，并将其添加到剪贴板。如果你在行首，则剪切整行。
- Ctrl + W 删除光标前的单词，并将其添加到剪贴板。
- Ctrl + Y 粘贴最近剪切的内容（撤销上一次删除操作）  
当前 光标位置）。
- Alt + T 交换光标前的最后两个单词。
- Alt + L 将光标到单词末尾的字母转换为小写。
- Alt + U 将光标到单词末尾的字母转换为大写。
- Alt + C 从光标处开始将单词首字母大写（如果光标在单词开头，则对整个单词操作）。
- Alt + D 从光标处开始删除到单词末尾（如果光标在单词开头，则删除整个单词）。
- Alt + . 打印上一条命令中输入的最后一个单词。
- Ctrl + T 交换光标前的最后两个字符。

### 历史访问

- Ctrl + R 允许你搜索之前使用过的命令。
- Ctrl + G 退出历史搜索模式而不执行命令。
- Ctrl + J 允许您将当前匹配的命令复制到命令行而不执行它，从而使您能够

# Chapter 1: Getting started with GNU/Linux

## Section 1.1: Useful shortcuts

### Using The Terminal

The examples in this document assume that you are using a POSIX-compliant (such as **bash**, **sh**, **zsh**, **ksh**) shell.

Large portions of GNU/Linux functionality are achieved using the terminal. Most distributions of Linux include terminal emulators that allow users to interact with a shell from their desktop environment. A shell is a command-line interpreter that executes user inputted commands. **Bash** (Bourne Again SHell) is a common default shell among many Linux distributions and is the default shell for macOS.

These shortcuts will work if you are using **Bash** with the *emacs* keybindings (set by default):

### Open terminal

- Ctrl + Alt + T or Super + T

### Cursor movement

- Ctrl + A Go to the beginning of the line you are currently typing on.
- Ctrl + E Go to the end of the line you are currently typing on.
- Ctrl + XX Move between the beginning of the line and the current position of the cursor.
- Alt + F Move cursor forward one word on the current line.
- Alt + B Move cursor backward one word on the current line.
- Ctrl + F Move cursor forward one character on the current line.
- Ctrl + B Move cursor backward one character on the current line.

### Text manipulation

- Ctrl + U Cut the line from the current position to the beginning of the line, adding it to the clipboard. If you are at the end of the line, cut the entire line.
- Ctrl + K Cut the line from the current position to the end of the line, adding it to the clipboard. If you are at the beginning of the line, cut the entire line.
- Ctrl + W Delete the word before the cursor, adding it to the clipboard.
- Ctrl + Y Paste the last thing from the clipboard that you cut recently (undo the last delete at the **current** cursor position).
- Alt + T Swap the last two words before the cursor.
- Alt + L Make lowercase from cursor to end of word.
- Alt + U Make uppercase from cursor to end of word.
- Alt + C Capitalize to end of word starting at cursor (whole word if cursor is at the beginning of word).
- Alt + D Delete to end of word starting at cursor (whole word if cursor is at the beginning of word).
- Alt + . Prints the last word written in previous command.
- Ctrl + T Swap the last two characters before the cursor.

### History access

- Ctrl + R Lets you search through previously used commands.
- Ctrl + G Leave history searching mode without running a command.
- Ctrl + J Lets you copy current matched command to command line without running it, allowing you to

- 在运行命令之前进行修改。
- `Alt + R` 如果你编辑了从历史记录中调出的命令，撤销对该命令的任何更改。
- `Ctrl + P` 显示上一个执行的命令，即向后浏览命令历史（类似向上箭头）。
- `Ctrl + N` 显示下一个执行的命令，即向前浏览命令历史（类似向下箭头）。

终端控制

- `Ctrl + L` 清除屏幕，类似于 `clear` 命令。
- `Ctrl + S` 停止所有屏幕输出。当运行大量长输出的命令时，这非常有用。但这不会停止正在运行的命令。
- `Ctrl + Q` 在使用Ctrl+S停止输出后，恢复屏幕输出。
- `Ctrl + C` 结束当前运行的进程并返回提示符。
- `Ctrl + D` 注销当前的shell会话，类似于exit或logout命令。在某些命令中，作为文件结束（EOF）信号，表示已到达文件末尾。
- `Ctrl + Z` 挂起（暂停）当前正在运行的前台进程，返回 `shell` 提示符。然后你可以使用bg命令让该进程在后台运行。要再次将该进程带到前台，使用fg命令。要查看所有后台进程，使用jobs命令。
- `Tab键` 自动补全文件和目录名称。
- `Tab键 Tab键` 当输入的字符不能唯一匹配文件或目录名称时，显示所有可能的选项。

特殊字符

- `Ctrl + H` 与退格键相同。
- `Ctrl + J` 与回车键相同（历史上为换行符）。
- `Ctrl + M` 与回车键相同（历史上为回车符）。
- `Ctrl + I` 与Tab相同。
- `Ctrl + G` 响铃字符。
- `Ctrl + @` 空字符。
- `Esc` 等同于 `Alt` 修饰键。

关闭终端

- `Ctrl + Shift + W` 关闭终端标签页。
- `Ctrl + Shift + Q` 关闭整个终端。

或者，你可以使用 `set -o vi` 命令在 **bash** 中切换到 **vi** 键绑定。使用 `set -o emacs` 可以切换回 `emacs` 键绑定。

## 第1.2节：文件管理命令

Linux 对当前目录和父目录有一些约定。这对初学者来说可能有些困惑。

每当你在 Linux 终端中时，你所在的位置称为 *当前工作目录*。通常你的命令提示符会显示完整的工作目录，或者仅显示该目录的最后一部分。你的提示符可能看起来像以下几种之一：

```
user@host ~/somedir $
user@host somedir $
user@host /home/user/somedir $
```

这表示你当前的工作目录是 `/home/user/somedir`。

- make modifications before running the command.
- `Alt + R` Revert any changes to a command you've pulled from your history, if you've edited it.
- `Ctrl + P` Shows last executed command, i.e. walk back through the command history (Similar to up arrow).
- `Ctrl + N` Shows next executed command, i.e. walk forward through the command history (Similar to down arrow).

Terminal control

- `Ctrl + L` Clears the screen, similar to the clear command.
- `Ctrl + S` Stop all output to the screen. This is useful when running commands with lots of long output. But this doesn't stop the running command.
- `Ctrl + Q` Resume output to the screen after stopping it with Ctrl+S.
- `Ctrl + C` End currently running process and return the prompt.
- `Ctrl + D` Log out of the current shell session, similar to the exit or logout command. In some commands, acts as End of File signal to indicate that a file end has been reached.
- `Ctrl + Z` Suspends (pause) currently running foreground process, which returns shell prompt. You can then use bg command allowing that process to run in the background. To again bring that process to foreground, use fg command. To view all background processes, use **jobs** command.
- `Tab` Auto-complete files and directory names.
- `Tab Tab` Shows all possibilities, when typed characters doesn't uniquely match to a file or directory name.

Special characters

- `Ctrl + H` Same as Backspace.
- `Ctrl + J` Same as Return (historically Line Feed).
- `Ctrl + M` Same as Return (historically Carriage Return).
- `Ctrl + I` Same as Tab.
- `Ctrl + G` Bell Character.
- `Ctrl + @` Null Character.
- `Esc` `Deadkey` equivalent to the `Alt` modifier.

Close Terminal

- `Ctrl + Shift + W` To close terminal tab.
- `Ctrl + Shift + Q` To close entire terminal.

Alternatively, you can switch to the *vi* keybindings in **bash** using `set -o vi`. Use `set -o emacs` to switch back to the `emacs` keybindings.

## Section 1.2: File Management Commands

Linux uses some conventions for present and parent directories. This can be a little confusing for beginners.

Whenever you are in a terminal in Linux, you will be in what is called the *current working directory*. Often your command prompt will display either the full working directory, or just the last part of that directory. Your prompt could look like one of the following:

```
user@host ~/somedir $
user@host somedir $
user@host /home/user/somedir $
```

which says that your current working directory is `/home/user/somedir`.



在 Linux 中， `..` 表示父目录， `.` 表示当前目录。

因此，如果当前目录是/home/user/somedir，那么cd ../somedir不会改变工作目录。

下表列出了一些最常用的文件管理命令

目录导航	命令	功能
	<code>pwd</code>	获取当前工作目录的完整路径。
	<code>cd -</code>	切换到上一次工作的目录。
	<code>cd ~</code> 或者直接 <code>cd</code>	切换到当前用户的主目录。
	<code>cd ..</code>	进入当前目录的父目录（注意cd和..之间的空格）

列出目录中的文件	命令	功能
	<code>ls -l</code>	以长格式（表格形式）列出当前目录中的文件和目录（建议与ls一起使用-l以提高可读性）。
	<code>ls -ld</code> 目录名	列出目录目录名的信息，而不是其内容。
	<code>ls -a</code>	列出所有文件，包括隐藏文件（以.开头的文件名是Linux中的隐藏文件）。
	<code>ls -F</code>	在文件名末尾添加符号以指示其类型（*表示可执行文件，/表示目录，@表示符号链接，=表示套接字， 表示命名管道，>表示门）。
	<code>ls -lt</code>	列出按最后修改时间排序的文件，最近修改的文件显示在顶部（记住 -l 选项提供长格式，具有更好的可读性）。
	<code>ls -lh</code>	以人类可读的格式列出文件大小。
	<code>ls -lR</code>	递归显示所有子目录。
	<code>tree</code>	将从当前目录开始生成文件系统的树状表示。

文件/目录创建、复制和删除	命令	功能
	<code>cp -p</code> 源文件 目标文件	将文件从源文件复制到目标文件。-p 表示保留。它在复制时保留文件的原始属性，如文件所有者、时间戳、组、权限等。
	<code>cp -R</code> source_dir destination_dir	将源目录递归复制到指定目标。
	<code>mv</code> file1 file2	在 Linux 中没有专门的 <b>rename</b> 命令。因此 <code>mv</code> 会将 file1 移动/重命名为 file2。
	<code>rm -i</code> filename	在删除每个文件前都会询问确认。如果你是 Linux 命令行的新用户，应该始终使用 <code>rm -i</code> 。你可以指定多个文件。
	<code>rm -R</code> dir-name	将递归删除目录 dir-name 。
	<code>rm -rf</code> dir-name	将递归删除目录 dir ，忽略不存在的文件且永不提示任何信息。使用此命令时请小心！你可以指定多个目录。
	<code>rmdir</code> dir-name	如果目录 dir-name为空，则将其删除。此命令只能删除空目录。
	<code>mkdir</code> dir-name	创建目录 dir-name。
	<code>mkdir -p</code> dir-name/dir-name	创建目录层级。如果父目录不存在，则按需创建。可以指定多个目录。
	<code>touch</code> filename	如果文件 filename不存在，则创建该文件；否则将文件的时间戳更改为当前时间。

文件/目录权限和组	命令	功能
-----------	----	----

In Linux `..` represents the parent directory and `.` represents the current directory.

Therefore, if the current directory is /home/user/somedir, then `cd ../somedir` will not change the working directory.

The table below lists some of the most used file management commands

Directory navigation	Command	Utility
	<code>pwd</code>	Get the full path of the current working directory.
	<code>cd -</code>	Navigate to the last directory you were working in.
	<code>cd ~</code> or just <code>cd</code>	Navigate to the current user's home directory.
	<code>cd ..</code>	Go to the parent directory of current directory (mind the space between cd and ..)

Listing files inside a directory	Command	Utility
	<code>ls -l</code>	List the files and directories in the current directory in long (table) format (It is recommended to use -l with ls for better readability).
	<code>ls -ld</code> dir-name	List information about the directory dir-name instead of its contents.
	<code>ls -a</code>	List all the files including the hidden ones (File names starting with a . are hidden files in Linux).
	<code>ls -F</code>	Appends a symbol at the end of a file name to indicate its type (* means executable, / means directory, @ means symbolic link, = means socket,   means named pipe, > means door).
	<code>ls -lt</code>	List the files sorted by last modified time with most recently modified files showing at the top (remember -l option provides the long format which has better readability).
	<code>ls -lh</code>	List the file sizes in human readable format.
	<code>ls -lR</code>	Shows all subdirectories recursively.
	<code>tree</code>	Will generate a tree representation of the file system starting from the current directory.

File/directory create, copy and remove	Command	Utility
	<code>cp -p</code> source destination	Will copy the file from <b>source</b> to <i>destination</i> . -p stands for preservation. It preserves the original attributes of file while copying like file owner, timestamp, group, permissions etc.
	<code>cp -R</code> source_dir destination_dir	Will copy source directory to specified destination recursively.
	<code>mv</code> file1 file2	In Linux there is no <b>rename</b> command as such. Hence mv moves/renames the file1 to file2.
	<code>rm -i</code> filename	Asks you before every file removal for confirmation. <b>IF YOU ARE A NEW USER TO LINUX COMMAND LINE, YOU SHOULD ALWAYS USE <code>rm -i</code></b> . You can specify multiple files.
	<code>rm -R</code> dir-name	Will remove the directory dir-name recursively.
	<code>rm -rf</code> dir-name	Will remove the directory <b>dir</b> recursively, ignoring non-existent files and will never prompt for anything. <b>BE CAREFUL USING THIS COMMAND!</b> You can specify multiple directories.
	<code>rmdir</code> dir-name	Will remove the directory dir-name, if it's empty. This command can only remove empty directories.
	<code>mkdir</code> dir-name	Create a directory dir-name.
	<code>mkdir -p</code> dir-name/dir-name	Create a directory hierarchy. Create parent directories as needed, if they don't exist. You can specify multiple directories.
	<code>touch</code> filename	Create a file filename, if it doesn't exist, otherwise change the timestamp of the file to current time.

File/directory permissions and groups	Command	Utility
---------------------------------------	---------	---------



<code>chmod &lt;specification&gt; filename</code>	更改文件权限。规格 = u 用户，g 组，o 其他，+ 添加权限，- 移除权限，r 读，w 写，x 执行。
<code>chmod -R &lt;specification&gt; 目录名</code>	递归更改目录的权限。要更改目录及其内部所有内容的权限，请使用此命令。
<code>chmod go+=r myfile</code>	为所有者和组添加读取权限。
<code>chmod a +rwx myfile</code>	允许所有用户读取、写入或执行myfile。
<code>chmod go -r myfile</code>	从组和其他用户移除读取权限。
<code>chown owner1 filename</code>	将文件所有权更改为用户owner1。
<code>chgrp grp_owner filename</code>	将文件filename的主组所有权更改为组grp_owner。
<code>chgrp -R grp_owner dir-name</code>	递归地将目录dir-name的主组所有权更改为组grp_owner要更改目录及其所有内容的组所有权，请使用此命令。

## 第1.3节：Hello World

在终端中输入以下代码，然后按 `回车`：

```
echo "Hello World"
```

这将产生以下输出：

```
Hello World
```

## 第1.4节：基本的Linux实用工具

Linux 几乎有适用于任何任务的命令，而且大多数命令直观且易于理解。

### 在 Linux 中获取帮助

命令	可用性
<code>man &lt;name&gt;</code>	阅读 <name> 的手册页。
<code>man &lt;section&gt; &lt;name&gt;</code>	阅读与指定章节相关的 <name> 手册页。
<code>man -k &lt;editor&gt;</code>	输出所有手册页中包含 <editor> 关键字的软件。
<code>man -K &lt;keyword&gt;</code>	输出所有包含<keyword>的man手册页。
<code>apropos &lt;editor&gt;</code>	输出所有一行描述中包含单词editor的应用程序。当无法回忆应用程序名称时，使用此命令。
<code>help</code>	在Bash shell中，这将显示所有可用bash命令的列表。
<code>help &lt;name&gt;</code>	在Bash shell中，这将显示关于<name> bash命令的信息。
<code>info &lt;name&gt;</code>	查看关于<name>的所有信息。
<code>dpkg -l</code>	输出基于Debian系统上所有已安装软件包的列表。
<code>dpkg -L packageName</code>	列出Debian系统中指定软件包安装的文件及路径详情。
<code>dpkg -l   grep -i &lt;edit&gt;</code>	返回所有包含<edit>（不区分大小写）的.deb已安装软件包。
<code>less /var/lib/dpkg/available</code>	返回所有可用软件包的描述。
<code>whatis vim</code>	列出vim的一行描述。
<code>&lt;command-name&gt; --help</code>	显示关于 <tool-name> 的使用信息。有时 <code>command -h</code> 也有效，但并非所有命令都适用。

### Linux 世界中的用户身份识别及人员信息

命令	可用性
<code>hostname</code>	显示系统的主机名。

<code>chmod &lt;specification&gt; filename</code>	Change the file permissions. Specifications = u user, g group, o other, + add permission, - remove, r read, w write,x execute.
<code>chmod -R &lt;specification&gt; dir-name</code>	Change the permissions of a directory recursively. To change permission of a directory and everything within that directory, use this command.
<code>chmod go+=r myfile</code>	Add read permission for the owner and the group.
<code>chmod a +rwx myfile</code>	Allow all users to read, write or execute myfile.
<code>chmod go -r myfile</code>	Remove read permission from the group and others.
<code>chown owner1 filename</code>	Change ownership of a file to user owner1.
<code>chgrp grp_owner filename</code>	Change primary group ownership of file filename to group grp_owner.
<code>chgrp -R grp_owner dir-name</code>	Change primary group ownership of directory dir-name to group grp_owner recursively. To change group ownership of a directory and everything within that directory, use this command.

## Section 1.3: Hello World

Type the following code into your terminal, then press `Enter`：

```
echo "Hello World"
```

This will produce the following output:

```
Hello World
```

## Section 1.4: Basic Linux Utilities

Linux has a command for almost any tasks and most of them are intuitive and easily interpreted.

### Getting Help in Linux

Command	Usability
<code>man &lt;name&gt;</code>	Read the manual page of <name>.
<code>man &lt;section&gt; &lt;name&gt;</code>	Read the manual page of <name>, related to the given section.
<code>man -k &lt;editor&gt;</code>	Output all the software whose man pages contain <editor> keyword.
<code>man -K &lt;keyword&gt;</code>	Outputs all man pages containing <keyword> within them.
<code>apropos &lt;editor&gt;</code>	Output all the applications whose one line description matches the word <i>editor</i> . When <b>not able to recall</b> the name of the application, use this command.
<code>help</code>	In Bash shell, this will display the list of all available bash commands.
<code>help &lt;name&gt;</code>	In Bash shell, this will display the info about the <name> bash command.
<code>info &lt;name&gt;</code>	View all the information about <name>.
<code>dpkg -l</code>	Output a list of all installed packages on a Debian-based system.
<code>dpkg -L packageName</code>	Will list out the files installed and path details for a given package on Debian.
<code>dpkg -l   grep -i &lt;edit&gt;</code>	Return all .deb installed packages with <edit> irrespective of cases.
<code>less /var/lib/dpkg/available</code>	Return descriptions of all available packages.
<code>whatis vim</code>	List a one-line description of vim.
<code>&lt;command-name&gt; --help</code>	Display usage information about the <tool-name>. Sometimes <code>command -h</code> also works, but not for all commands.

### User identification and who is who in Linux world

Command	Usability
<code>hostname</code>	Display hostname of the system.

hostname -f	显示系统的完全限定域名（FQDN）。
密码	更改当前用户的密码。
当前用户名	终端登录用户的用户名。
who	当前以用户身份登录的所有用户列表。
w	显示当前系统状态、时间、持续时间、当前登录系统的用户列表及其他用户信息。
last	最近使用系统的用户。
last root	root 用户上次登录系统的时间。
lasttb	显示所有系统中的错误登录尝试。
chmod	更改文件或目录的权限——读、写、执行。

进程相关信息

命令	可用性
top	列出所有进程，按当前系统资源使用情况排序。显示持续更新的进程列表（默认每3秒刷新一次）。使用q键退出top。
ps	列出当前shell会话中正在运行的进程
ps -u root	列出root用户正在运行的所有进程和命令
ps aux	列出当前系统上所有用户的所有进程

第1.5节：按名称/内容模式搜索文件

使用Linux命令行（shell）时，一个常见的任务是搜索具有特定名称或包含特定文本的文件/目录。为完成此任务，你应熟悉以下两个命令：

按名称查找文件

```
find /var/www -name '*.css'
```

这将打印出/var/www/目录下所有以.css结尾的文件的完整路径/文件名。示例输出：

```
/var/www/html/text-cursor.css
/var/www/html/style.css
```

更多信息：

```
man find
```

查找包含文本的文件

```
grep font /var/www/html/style.css
```

这将打印指定文件中所有包含模式font的行。示例输出：

```
font-weight: bold;
font-family: monospace;
```

另一个示例：

```
grep font /var/www/html/
```

hostname -f	Displays Fully Qualified Domain Name (FQDN) of the system.
passwd	Change password of current user.
whoami	Username of the users logged in at the terminal.
who	List of all the users currently logged in as a user.
w	Display current system status, time, duration, list of users currently logged in on system and other user information.
last	Who recently used the system.
last root	When was the last time <b>root</b> logged in as user.
lasttb	Shows all bad login attempts into the system.
chmod	Changing permissions - read,write,execute of a file or directory.

Process related information

Command	Usability
top	List all processes sorted by their current system resource usage. Displays a continually updated display of processes (By default 3 seconds). Use q key to exit top.
ps	List processes currently running on current shell session
ps -u root	List all of the processes and commands root is running
ps aux	List all the processes by all users on the current system

Section 1.5: Searching for files by patterns in name/contents

A common and task of someone using the Linux Command Line (shell) is to search for files/directories with a certain name or containing certain text. There are 2 commands you should familiarise yourself with in order to accomplish this:

Find files by name

```
find /var/www -name '*.css'
```

This will print out the full path/filename to all files under /var/www that end in .css. Example output:

```
/var/www/html/text-cursor.css
/var/www/html/style.css
```

For more info:

```
man find
```

Find files containing text

```
grep font /var/www/html/style.css
```

This will print all lines containing the pattern font in the specified file. Example output:

```
font-weight: bold;
font-family: monospace;
```

Another example:

```
grep font /var/www/html/
```

这并不像你希望的那样工作。你会得到：

```
grep: /var/www/html/: 是一个目录
```

你需要递归地使用grep才能使其工作，使用-R选项：

```
grep -R font /var/www/html/
```

嘿，太棒了！看看这个命令的输出：

```
/var/www/html/admin/index.php:  echo '<font color=red><b>错误：没有结果</b></font><br/>';
/var/www/html/admin/index.php:  echo '<font color=red><b>错误：请再试一次</b></font><br/>';
/var/www/html/style.css:  font-weight: bold;
/var/www/html/style.css:  font-family: monospace;
```

注意，当**grep**匹配多个文件时，它会在匹配的行前加上文件名。如果你想，可以使用-h选项去掉它。

更多信息：

```
man grep
```

## 第1.6节：文件操作

文件和目录（文件夹的另一种称呼）是 Linux 的核心，因此能够从命令行创建、查看、移动和删除它们非常重要且功能强大。这些文件操作命令允许你执行与图形文件管理器相同的任务。

创建一个名为myFile的空文本文件：

```
touch myFile
```

将myFile重命名为myFirstFile：

```
mv myFile myFirstFile
```

查看文件内容：

```
cat myFirstFile
```

使用分页器查看文件内容（一次一屏）：

```
less myFirstFile
```

查看文件的前几行：

```
head myFirstFile
```

查看文件的最后几行：

```
tail myFirstFile
```

编辑文件：

This doesn't work as you'd hoped. You get:

```
grep: /var/www/html/: Is a directory
```

You need to **grep** recursively to make it work, using the -R option:

```
grep -R font /var/www/html/
```

Hey nice! Check out the output of this one:

```
/var/www/html/admin/index.php:  echo '<font color=red><b>Error: no dice</b></font><br/>';
/var/www/html/admin/index.php:  echo '<font color=red><b>Error: try again</b></font><br/>';
/var/www/html/style.css:  font-weight: bold;
/var/www/html/style.css:  font-family: monospace;
```

Notice that when **grep** is matching multiple files, it prefixes the matched lines with the filenames. You can use the -h option to get rid of that, if you want.

For more info:

```
man grep
```

## Section 1.6: File Manipulation

Files and directories (another name for folders) are at the heart of Linux, so being able to create, view, move, and delete them from the command line is very important and quite powerful. These file manipulation commands allow you to perform the same tasks that a graphical file explorer would perform.

Create an empty text file called myFile:

```
touch myFile
```

Rename myFile to myFirstFile:

```
mv myFile myFirstFile
```

View the contents of a file:

```
cat myFirstFile
```

View the content of a file with pager (one screenful at a time):

```
less myFirstFile
```

View the first several lines of a file:

```
head myFirstFile
```

View the last several lines of a file:

```
tail myFirstFile
```

Edit a file:

```
vi myFirstFile
```

查看当前工作目录中的文件：

```
ls
```

创建一个名为myFirstDirectory的空目录：

```
mkdir myFirstDirectory
```

创建多路径目录：（创建两个目录，src 和 myFirstDirectory）

```
mkdir -p src/myFirstDirectory
```

将文件移动到目录中：

```
mv myFirstFile myFirstDirectory/
```

你也可以重命名文件：

```
user@linux-computer:~$ mv myFirstFile secondFileName
```

将当前工作目录更改为myFirstDirectory：

```
cd myFirstDirectory
```

删除文件：

```
rm myFirstFile
```

进入父目录（用..表示）：

```
cd ..
```

删除空目录：

```
rmdir myFirstDirectory
```

删除非空目录（即包含文件和/或其他目录）：

```
rm -rf myFirstDirectory
```

请注意，删除目录时，应删除./而不是/，否则会清空整个文件系统。

## 第1.7节：文件/目录详情

ls命令有多个选项，可以组合使用以显示更多信息。

### 详情/权限

l选项显示文件权限、大小和最后修改日期。因此，如果根目录包含一个名为test的目录和一个文件someFile，命令：

```
vi myFirstFile
```

See what files are in your current working directory:

```
ls
```

Create an empty directory called myFirstDirectory:

```
mkdir myFirstDirectory
```

Create multi path directory: (creates two directories, src and myFirstDirectory)

```
mkdir -p src/myFirstDirectory
```

Move the file into the directory:

```
mv myFirstFile myFirstDirectory/
```

You can also rename the file:

```
user@linux-computer:~$ mv myFirstFile secondFileName
```

Change the current working directory to myFirstDirectory:

```
cd myFirstDirectory
```

Delete a file:

```
rm myFirstFile
```

Move into the parent directory (which is represented as ..):

```
cd ..
```

Delete an empty directory:

```
rmdir myFirstDirectory
```

Delete a non-empty directory (i.e. contains files and/or other directories):

```
rm -rf myFirstDirectory
```

Make note that when deleting directories, that you delete ./ not / that will wipe your whole filesystem.

## Section 1.7: File/Directory details

The ls command has several options that can be used together to show more information.

### Details/Rights

The l option shows the file permissions, size, and last modified date. So if the root directory contained a dir called test and a file someFile the command:

```
user@linux-computer:~$ ls -l
```

将输出类似如下内容

```
-rw-r--r-- 1 user users 70 7月 22 13:36 someFile.txt
drwxrwxrwx 2 user users 4096 7月 21 07:18 test
```

权限的格式是 drwxrwxrwx。第一个字符表示文件类型， d 表示目录，-表示其他类型。接下来的三个 rwx 是用户对文件的权限，接下来的三个是组对文件的权限，最后三个是其他所有人对文件的权限。

r 在 rwx 中表示文件是否可读，w 表示文件是否可修改，x 表示文件是否可执行。如果某个权限未被授予，则用 - 替代 r、w 或 x。

所以从上面可以看出，user 可以读取和修改 someFile.txt，但组只有只读权限。

要更改权限，如果你有sudo权限，可以使用 chmod ### fileName 命令。 r 用值4表示，w 用值2表示，x 用值1表示。所以如果你只想修改 test 目录的内容

```
所有者 rwx = 4+2+1 = 7
组 r-x = 4+0+1 = 5
其他 r-x = 4+0+1 = 5
```

所以完整的命令是

```
chmod 755 test
```

现在执行 ls -l 会显示类似如下内容

```
drwxr-xr-x 2 user users 4096 7月 21 07:20 test
```

可读大小

与l选项结合使用时，h选项显示人类可读的文件大小。运行

```
user@linux-computer:~$ ls -lh
```

将输出：

```
total 4166
-rw-r--r-- 1 user users 70 7月 22 13:36 someFile.txt
drwxrwxrwx 2 user users 4.0K 7月 21 07:18 test
```

隐藏文件

要查看隐藏文件，请使用 a选项。例如

```
user@linux-computer:~$ ls -a
```

可能列出

```
user@linux-computer:~$ ls -l
```

Would output something like

```
-rw-r--r-- 1 user users 70 Jul 22 13:36 someFile.txt
drwxrwxrwx 2 user users 4096 Jul 21 07:18 test
```

The permissions are in format of drwxrwxrwx. The first character represents the file type d if it's a directory - otherwise. The next three rwx are the permissions the user has over the file, the next three are the permissions the group has over the file, and the last three are the permissions everyone else has over the file.

The r of rwx stands for if a file can be read, the w represents if the file can be modified, and the x stands for if the file can be executed. If any permission isn't granted a - will be in place of r, w, or x.

So from above user can read and modify someFile.txt but the group has only read-only rights.

To change rights you can use the chmod ### fileName command if you have sudo rights. r is represented by a value of 4, w is represented by 2, and x is represented by a 1. So if only you want to be able to modify the contents to the test directory

```
Owner rwx = 4+2+1 = 7
Group r-x = 4+0+1 = 5
Other r-x = 4+0+1 = 5
```

So the whole command is

```
chmod 755 test
```

Now doing a ls -l would show something like

```
drwxr-xr-x 2 user users 4096 Jul 21 07:20 test
```

Readable Size

Used in conjunction with the l option the h option shows file sizes that are human readable. Running

```
user@linux-computer:~$ ls -lh
```

Would output:

```
total 4166
-rw-r--r-- 1 user users 70 Jul 22 13:36 someFile.txt
drwxrwxrwx 2 user users 4.0K Jul 21 07:18 test
```

Hidden

To view hidden files use the a option. For example

```
user@linux-computer:~$ ls -a
```

Might list

```
.profile
someFile.txt
测试
```

### 目录总大小

要查看当前目录的大小，请使用-s选项（也可以使用-h选项使大小更易读）。

```
user@linux-computer:~$ ls -s
```

输出

```
total 4166
someFile.txt      测试
```

### 递归查看

假设测试目录中有一个文件anotherFile，且你想从根文件夹查看它，可以使用R选项，该选项会列出递归树。

```
user@linux-computer:~$ ls -R
```

输出

```
.:
someFile.txt      测试

./测试:
anotherFile
```

```
.profile
someFile.txt
test
```

### Total Directory Size

To view the size of the current directory use the s option (the h option can also be used to make the size more readable).

```
user@linux-computer:~$ ls -s
```

Outputs

```
total 4166
someFile.txt      test
```

### Recursive View

Lets say test directory had a file anotherFile and you wanted to see it from the root folder, you could use the R option which would list the recursive tree.

```
user@linux-computer:~$ ls -R
```

Outputs

```
.:
someFile.txt      test

./test:
anotherFile
```



## 第2章：检测Linux发行版名称和版本

### 第2.1节：检测你正在使用的基于Debian的发行版

只需执行 `lsb_release -a`。

在Debian上：

```
$ lsb_release -a
没有可用的LSB模块。
发行商ID：Debian
描述：    Debian GNU/Linux 测试版 (stretch)
版本：    测试版
代号：    stretch
```

在 Ubuntu 上：

```
$ lsb_release -a
没有可用的LSB模块。
发行版 ID：Ubuntu
描述：    Ubuntu 14.04.4 LTS
版本：    14.04
代号：    trusty
```

如果你没有安装`lsb_release`，可能需要尝试一些猜测，例如，有一个文件 `/etc/issue`，通常包含发行版名称。例如，在 Ubuntu 上：

```
$ cat /etc/issue
Ubuntu 12.04.5 LTS \l
```

不要使用文件 `/etc/debian_version`，因为其内容与发行版名称不匹配！

注意，这也适用于非 Debian 系发行版，如 Fedora、RHEL 或 openSUSE—but `lsb_release` 可能未安装。

### 第 2.2 节：检测你正在使用的基于 systemd 的发行版

此方法适用于 Arch、CentOS、CoreOS、Debian、Fedora、Mageia、openSUSE、红帽企业 Linux、SUSE Linux 企业服务器、Ubuntu 等现代版本。其广泛的适用性使其成为首选方法，如果需要识别较旧系统，可以备选其他方法。

查看 `/etc/os-release`。具体查看变量 `NAME`、`VERSION`、`ID`、`VERSION_ID` 和 `PRETTY_NAME`。

在 Fedora 上，此文件可能如下所示：

```
名称=Fedora
版本="24 (工作站版) "
ID=fedora
版本号=24
完整名称="Fedora 24 (工作站版) "
ANSI颜色="0;34"
```

## Chapter 2: Detecting Linux distribution name and version

### Section 2.1: Detect what debian-based distribution you are working in

Just execute `lsb_release -a`.

On Debian:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux testing (stretch)
Release:        testing
Codename:       stretch
```

On Ubuntu:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 14.04.4 LTS
Release:        14.04
Codename:       trusty
```

In case when you don't have `lsb_release` installed you may want to try some guessing, for example, there is a file `/etc/issue` that often contains distribution name. For example, on ubuntu:

```
$ cat /etc/issue
Ubuntu 12.04.5 LTS \n \l
```

Don't use file `/etc/debian_version` because its contents do not match distribution name!

Note that this will also work on non-Debian-family distributions like Fedora, RHEL, or openSUSE — but that `lsb_release` may not be installed.

### Section 2.2: Detect what systemd-based distribution you are using

This method will work on modern versions of Arch, CentOS, CoreOS, Debian, Fedora, Mageia, openSUSE, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, Ubuntu, and others. This wide applicability makes it an ideal as a first approach, with fallback to other methods if you need to also identify older systems.

Look at `/etc/os-release`. In specific, look at variables `NAME`, `VERSION`, `ID`, `VERSION_ID`, and `PRETTY_NAME`.

On Fedora, this file might look like:

```
NAME=Fedora
VERSION="24 (Workstation Edition)"
ID=fedora
VERSION_ID=24
PRETTY_NAME="Fedora 24 (Workstation Edition)"
ANSI_COLOR="0;34"
```

```
CPE名称="cpe:/o:fedoraproject:fedora:24"
主页URL="https://fedoraproject.org/"
错误报告URL="https://bugzilla.redhat.com/"
红帽Bugzilla产品="Fedora"
红帽Bugzilla产品版本=24
红帽支持产品="Fedora"
红帽支持产品版本=24
隐私政策URL=https://fedoraproject.org/wiki/Legal:PrivacyPolicy
变体="工作站版"
VARIANT_ID=工作站
```

在 CentOS 上，此文件可能如下所示：

```
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="7"
PRETTY_NAME="CentOS Linux 7 (Core)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:7"
HOME_URL="https://www.centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"

CENTOS_MANTISBT_PROJECT="CentOS-7"
CENTOS_MANTISBT_PROJECT_VERSION="7"
REDHAT_SUPPORT_PRODUCT="centos"
REDHAT_SUPPORT_PRODUCT_VERSION="7"
```

此文件在 freedesktop 网站上有文档说明；原则上，它并非 systemd 特有，但它会存在于所有基于 systemd 的发行版中。

在 bash shell 中，可以 source /etc/os-release 文件，然后直接使用各种变量，如下所示：

```
$ ( source /etc/os-release && echo "$PRETTY_NAME" )
Fedora 24 (Workstation Edition)
```

## 第 2.3 节：检测你正在使用的 RHEL / CentOS / Fedora 发行版

查看 /etc/redhat-release 的内容

```
cat /etc/redhat-release
```

以下是 Fedora 24 机器的输出：Fedora release 24 (二十四)如同在基于 Debian 的

回答中提到的，你也可以使用 lsb\_release -a 命令，它会输出来自 Fedora 24 机器的信息：

```
LSB 版本：      :core-4.1-amd64:core-4.1-noarch:cxx-4.1-amd64:cxx-4.1-noarch:desktop-4.1-
amd64:desktop-4.1-noarch:languages-4.1-amd64:languages-4.1-noarch:printing-4.1-amd64:printing-4.1-
noarch
发行商 ID：     Fedora
描述：         Fedora release 24 (二十四)
版本：         24
代号：         TwentyFour
```

```
CPE_NAME="cpe:/o:fedoraproject:fedora:24"
HOME_URL="https://fedoraproject.org/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"
REDHAT_BUGZILLA_PRODUCT="Fedora"
REDHAT_BUGZILLA_PRODUCT_VERSION=24
REDHAT_SUPPORT_PRODUCT="Fedora"
REDHAT_SUPPORT_PRODUCT_VERSION=24
PRIVACY_POLICY_URL=https://fedoraproject.org/wiki/Legal:PrivacyPolicy
VARIANT="Workstation Edition"
VARIANT_ID=workstation
```

On CentOS, this file might look like this:

```
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="7"
PRETTY_NAME="CentOS Linux 7 (Core)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:7"
HOME_URL="https://www.centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"

CENTOS_MANTISBT_PROJECT="CentOS-7"
CENTOS_MANTISBT_PROJECT_VERSION="7"
REDHAT_SUPPORT_PRODUCT="centos"
REDHAT_SUPPORT_PRODUCT_VERSION="7"
```

This file is documented on the freedesktop web site; in principle, it is not systemd specific — but it will exist on all systemd-based distributions.

From the bash shell, one can source the /etc/os-release file and then use the various variables directly, like this:

```
$ ( source /etc/os-release && echo "$PRETTY_NAME" )
Fedora 24 (Workstation Edition)
```

## Section 2.3: Detect what RHEL / CentOS / Fedora distribution you are working in

Look at the contents of /etc/redhat-release

```
cat /etc/redhat-release
```

Here is the output from a Fedora 24 machine: Fedora release 24 (Twenty Four)

As mentioned in the debian-based response, you can also use the lsb\_release -a command, which outputs this from a Fedora 24 machine:

```
LSB Version:      :core-4.1-amd64:core-4.1-noarch:cxx-4.1-amd64:cxx-4.1-noarch:desktop-4.1-
amd64:desktop-4.1-noarch:languages-4.1-amd64:languages-4.1-noarch:printing-4.1-amd64:printing-4.1-
noarch
Distributor ID:   Fedora
Description:      Fedora release 24 (Twenty Four)
Release:         24
Codename:         TwentyFour
```

## 第 2.4 节：Uname - 打印当前系统的信息

Uname 是 unix name 的简称。只需在控制台输入 uname 即可获取操作系统的信息。

```
uname [选项]

如果未指定 选项，uname 默认使用 -s 选项。

-a 或 --all - 打印所有信息，如果信息未知，则省略 -p 和 -i。

示例：
> uname -a

SunOS hope 5.7 Generic_106541-08 sun4m sparcsunw,SPARCstation-10
```

所有选项：

-s, --kernel-name	打印内核名称。
-n, --nodename	打印网络节点主机名。
-r, --kernel-release	打印内核版本。
-v, --kernel-version	打印内核版本。
-m, --machine	打印机器硬件名称。
-p, --processor	打印处理器类型，或“unknown”。
-i, --hardware-platform	打印硬件平台，或“unknown”。
-o, --operating-system	打印操作系统。
--help	显示帮助信息并退出。
--version	显示版本信息，并退出。

## 第2.5节：检测发行版的基本信息

只需执行 uname -a。

在Arch上：

```
$ uname -a
Linux nokia 4.6.4-1-ARCH #1 SMP PREEMPT Mon Jul 11 19:12:32 CEST 2016 x86_64 GNU/Linux
enter code here
```

## 第2.6节：使用GNU coreutils

因此，GNU coreutils 应该在所有基于Linux的系统上都可用（如果我错了请指正）。

如果你不知道自己使用的是什么系统，可能无法直接跳转到上面的某个示例，因此这可能是你首先要查看的内容。

\$ uname -a

在我的系统上，这给出了以下结果.....

Linux Scibearspace 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt25-2+deb8u3 (2016-07-02) x86\_64

## Section 2.4: Uname - Print information about the current system

**Uname** is the short name for **unix name**. Just type **uname** in console to get information about your operating system.

```
uname [OPTION]

If no OPTION is specified, uname assumes the -s option.

-a or --all - Prints all information, omitting -p and -i if the information is unknown.

Example:
> uname -a

SunOS hope 5.7 Generic_106541-08 sun4m sparcsunw,SPARCstation-10
```

All the options:

-s, --kernel-name	Print the kernel name.
-n, --nodename	Print the network node hostname.
-r, --kernel-release	Print the kernel release.
-v, --kernel-version	Print the kernel version.
-m, --machine	Print the machine hardware name.
-p, --processor	Print the processor type, or " <b>unknown</b> ".
-i, --hardware-platform	Print the hardware platform, or " <b>unknown</b> ".
-o, --operating-system	Print the operating system.
--help	Display a help message, and exit.
--version	Display version information, and exit.

## Section 2.5: Detect basic information about your distro

just execute **uname -a**.

On Arch:

```
$ uname -a
Linux nokia 4.6.4-1-ARCH #1 SMP PREEMPT Mon Jul 11 19:12:32 CEST 2016 x86_64 GNU/Linux
enter code here
```

## Section 2.6: Using GNU coreutils

So the GNU coreutils should be available on all linux based systems (please correct me if I am wrong here).

If you do not know what system you are using you may not be able to directly jump to one of the examples above, hence this may be your first port of call.

\$ uname -a

On my system this gives me the following...

Linux Scibearspace 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt25-2+deb8u3 (2016-07-02) x86\_64

您可以看到以下内容：

Scibearspace：我的电脑名称

- Scibearspace：我的电脑名称
- 3.16.0-4-amd64：内核和架构
- SMP Debian 3.16.7-CKT25-2+deb8u3：表示我运行的是带有3.16内核的Debian系统
- 最后部分表示我运行的是Debian 8（更新3）。

欢迎其他人补充RHEL和SuSe系统的结果。

## 第2.7节：查找您的Linux操作系统（包括Debian和RPM）名称和版本号

大多数Linux发行版将其版本信息存储在/etc/lsb-release（Debian）或/etc/redhat-release（基于RPM）文件中。使用以下通用命令应该可以让您获取大多数Debian和RPM衍生版的信息，如Linux Mint和Cent-Os。

Ubuntu机器上的示例：

```
cat /etc/*release
```

```
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=14.04
DISTRIB_CODENAME=trusty
DISTRIB_DESCRIPTION="Ubuntu 14.04 LTS"
```

Here you can see the following：

Scibearspace：the name of my pc

- Scibearspace：the name of my pc
- 3.16.0-4-amd64：the kernel and architecture
- SMP Debian 3.16.7-CKT25-2+deb8u3：tells me I am running debian with the 3.16 kernel
- Finally the last part I am running debian 8 (update 3).

I would welcome any others to add in results for RHEL, and SuSe systems.

## Section 2.7: Find your linux os (both debian & rpm) name and release number

Most of linux distros stores its version info in the /etc/lsb-release (debian) or /etc/redhat-release (RPM based) file. Using below generic command should get you past most of the Debian and RPM derivatives as Linux Mint and Cent-Os.

Example on Ubuntu Machine:

```
cat /etc/*release
```

```
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=14.04
DISTRIB_CODENAME=trusty
DISTRIB_DESCRIPTION="Ubuntu 14.04 LTS"
```

# 第3章：获取正在运行的Linux内核信息

## 第3.1节：获取Linux内核详细信息

我们可以使用uname命令及其各种选项来获取正在运行的内核的完整详细信息。

uname -a

```
Linux df1-ws-5084 4.4.0-64-generic #85-Ubuntu SMP 2017年2月20日 星期一 11:50:30 UTC x86_64
x86_64 x86_64 GNU/Linux
```

根据手册页，这里还有更多选项

用法：uname [OPTION]...

打印某些系统信息。无选项时，效果同-s。

```
-a, --all          打印所有信息，按以下顺序，
如果未知则省略-p和-i：
-s, --kernel-name  打印内核名称
-n, --nodename     打印网络节点主机名
-r, --kernel-release 打印内核版本号
-v, --kernel-version 打印内核版本
-m, --machine      打印机器硬件名称
-p, --processor    打印处理器类型（非通用）
-i, --hardware-platform 打印硬件平台（非通用）
-o, --operating-system 打印操作系统
--help           显示此帮助信息并退出
--version        输出版本信息并退出
```

# Chapter 3: Getting information on a running Linux kernel

## Section 3.1: Getting details of Linux kernel

We can use command uname with various options to get complete details of running kernel.

uname -a

```
Linux df1-ws-5084 4.4.0-64-generic #85-Ubuntu SMP Mon Feb 20 11:50:30 UTC 2017 x86_64
x86_64 x86_64 GNU/Linux
```

As per man page here few more options

Usage: uname [OPTION]...

Print certain system information. With no OPTION, same as -s.

```
-a, --all          print all information, in the following order,
except omit -p and -i if unknown:
-s, --kernel-name  print the kernel name
-n, --nodename     print the network node hostname
-r, --kernel-release  print the kernel release
-v, --kernel-version  print the kernel version
-m, --machine      print the machine hardware name
-p, --processor    print the processor type (non-portable)
-i, --hardware-platform  print the hardware platform (non-portable)
-o, --operating-system  print the operating system
--help           display this help and exit
--version        output version information and exit
```



# 第4章：壳体

Shell 会在提示符响应时执行程序。当你输入命令时，shell 会搜索该程序，然后执行它。例如，当你输入命令 `ls` 时，shell 会搜索名为 `ls` 的实用程序/程序，然后在 `shell` 中运行它。你提供给实用程序的参数和选项会影响你得到的结果。shell 也被称为 `CLI`，或命令行界面。

## 第4.1节：更改默认shell

大多数现代发行版都会预装并配置 `BASH`（`Bourne Again Shell`）作为默认shell。

负责在 `Linux` 中更改 `shell` 的命令（实际上是一个可执行二进制文件，`ELF`）是 `chsh`（`change shell`）。

我们可以先使用 `chsh -l` 命令检查机器上已安装和配置的 `shell`，输出结果类似如下：

```
[user@localhost ~]$ chsh -l
/bin/sh
/bin/bash
/sbin/nologin
/usr/bin/sh
/usr/bin/bash
/usr/sbin/nologin
/usr/bin/fish
```

在某些 `Linux` 发行版中，`chsh -l` 命令无效。这种情况下，所有可用 `shell` 的列表可以在 `/etc/shells` 文件中找到。你可以用 `cat` 命令查看文件内容：

```
[user@localhost ~]$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/sbin/nologin
/usr/bin/sh
/usr/bin/bash
/usr/sbin/nologin
/usr/bin/fish
```

现在我们可以选择新的默认 `shell`，例如 `fish`，并使用 `chsh -s` 命令进行配置，

```
[user@localhost ~]$ chsh -s /usr/bin/fish
正在为用户更改 shell。
密码：
Shell 已更改。
```

现在只需注销并重新登录，即可享受新的默认 `shell`。

如果您希望更改其他用户的默认 `shell`，并且您在该机器上拥有管理员权限，您可以使用 `chsh` 以 `root` 身份完成此操作。假设我们想将 `user_2` 的默认 `shell` 更改为 `fish`，我们将使用之前相同的命令，但需添加其他用户的用户名，命令为 `chsh -s /usr/bin/fish user_2`。

为了检查当前默认 `shell` 是什么，我们可以查看 `$SHELL` 环境变量，该变量指向默认 `shell` 的路径，因此更改后，我们期望得到类似以下的结果，

# Chapter 4: Shell

The shell executes a program in response to its prompt. When you give a command, the shell searches for the program, and then executes it. For example, when you give the command `ls`, the shell searches for the utility/program named `ls`, and then runs it in the shell. The arguments and the options that you provide with the utilities can impact the result that you get. The shell is also known as a `CLI`, or command line interface.

## Section 4.1: Changing default shell

Most modern distributions will come with `BASH` (**B**ourne **A**gain **S**hell) pre-installed and configured as a default shell.

The command (actually an executable binary, an `ELF`) that is responsible for changing shells in `Linux` is `chsh` (**ch**ange **sh**ell).

We can first check which shells are already installed and configured on our machine by using the `chsh -l` command, which will output a result similar to this:

```
[user@localhost ~]$ chsh -l
/bin/sh
/bin/bash
/sbin/nologin
/usr/bin/sh
/usr/bin/bash
/usr/sbin/nologin
/usr/bin/fish
```

In some `Linux` distributions, `chsh -l` is invalid. In this case, the list of all available shells can be found at `/etc/shells` file. You can show the file contents with `cat`:

```
[user@localhost ~]$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/sbin/nologin
/usr/bin/sh
/usr/bin/bash
/usr/sbin/nologin
/usr/bin/fish
```

Now we can choose our new default shell, e.g. `fish`, and configure it by using `chsh -s`,

```
[user@localhost ~]$ chsh -s /usr/bin/fish
Changing shell for user.
Password:
Shell changed.
```

Now all that is left to do is preform a logoff-logon cycle, and enjoy our new default shell.

If you wish to change the default shell for a different user, and you have administrative privileges on the machine, you'll be able to accomplish this by using `chsh` as `root`. So assuming we want to change `user_2`'s default shell to `fish`, we will use the same command as before, but with the addition of the other user's username, `chsh -s /usr/bin/fish user_2`.

In order to check what the current default shell is, we can view the `$SHELL` environment variable, which points to the path to our default shell, so after our change, we would expect to get a result similar to this,



```
~  echo $SHELL
/usr/bin/fish
```

chsh 选项：

```
-s shell
```

将 shell 设置为登录 shell。

```
-l, --list-shells
```

打印 /etc/shells 中列出的 shell 列表并退出。

```
-h, --help
```

打印使用说明信息并退出。

```
-v, --version
```

打印版本信息并退出。

第4.2节：基本Shell工具

自定义Shell提示符

默认的命令提示符可以更改为不同且简短的样式。如果当前目录路径较长，默认的命令提示符会变得过长。在这种情况下，使用PS1非常有用。一个简短且自定义的命令提示符既美观又优雅。下表中使用了PS1及多个参数来展示不同形式的shell提示符。默认的命令提示符看起来类似于：user@host ~ \$，我这里显示为：bruce@gotham ~ \$。它可以根据下表进行更改：

命令	功能
PS1='\w \$ '	~ \$ shell 提示符为目录名。在此情况下，根目录为 Root。
PS1='\h \$ '	gotham \$ shell 提示符为主机名
PS1='\u \$ '	bruce \$ shell 提示符为用户名
PS1=' \$ '	22:37:31 \$ shell 提示符为24小时制时间格式
PS1='@ \$ '	10:37 PM shell 提示符为12小时制时间格式
PS1='! \$ '	732 将显示命令的历史编号，代替shell提示符
PS1='dude \$ 'dude \$	将以你喜欢的方式显示shell提示符

一些基本的shell命令

命令	功能
Ctrl-k	剪切/删除
Ctrl-y	粘贴/恢复
Ctrl-a	将光标移到行首
Ctrl-e	将光标移到行尾
Ctrl-d	将删除光标后/处的字符
Ctrl-l	将清除屏幕/终端
Ctrl-u	将清除提示符与光标之间的所有内容
Ctrl-_	将撤销命令行上最后输入的内容
Ctrl-c	将中断/停止前台运行的作业/进程

```
~  echo $SHELL
/usr/bin/fish
```

chsh options:

```
-s shell
```

Sets shell as the login shell.

```
-l, --list-shells
```

Print the list of shells listed in /etc/shells and exit.

```
-h, --help
```

Print a usage message and exit.

```
-v, --version
```

Print version information and exit.

Section 4.2: Basic Shell Utilities

Customizing the Shell prompt

Default command prompt can be changed to look different and short. In case the current directory is long default command prompt becomes too large. Using PS1 becomes useful in these cases. A short and customized command pretty and elegant. In the table below PS1 has been used with a number of arguments to show different forms of shell prompts. Default command prompt looks something like this: user@host ~ \$ in my case it looks like this: bruce@gotham ~ \$. It can changed as per the table below:

Command	Utility
PS1='\w \$ '	~ \$ shell prompt as directory name. In this case root directory is Root.
PS1='\h \$ '	gotham \$ shell prompt as hostname
PS1='\u \$ '	bruce \$ shell prompt as username
PS1='\t \$ '	22:37:31 \$ shell prompt in 24 hour format
PS1='@ \$ '	10:37 PM shell prompt in 12 hour time format
PS1='! \$ '	732 will show the history number of command in place of shell prompt
PS1='dude \$ 'dude \$	will show the shell prompt the way you like

Some basic shell commands

Command	Utility
Ctrl-k	cut/kill
Ctrl-y	yank/paste
Ctrl-a	will take cursor to the start of the line
Ctrl-e	will take cursor to the end of the line
Ctrl-d	will delete the character after/at the cursor
Ctrl-l	will clear the screen/terminal
Ctrl-u	will clear everything between prompt and the cursor
Ctrl-_	will undo the last thing typed on the command line
Ctrl-c	will interrupt/stop the job/process running in the foreground

Ctrl-r	历史中的反向搜索
~/.bash_history	存储shell中最后使用的500条命令/事件
history	将显示命令历史
history   grep <关键字>将显示历史中所有包含关键字<关键字>的命令（在你只记得过去使用过的部分命令时非常有用） 当你记得过去使用过的命令的一部分时	

### 第4.3节：创建你自己的命令别名

如果你厌倦了在bash中使用冗长的命令，可以创建你自己的命令别名。

最好的方法是修改（如果不存在则创建）你主目录下名为.bash\_aliases的文件。其通用语法是：

```
alias command_alias='actual_command'
```

其中 actual\_command 是你要重命名的命令，command\_alias 是你给它的新名称。  
例如

```
alias install='sudo apt-get -y install'
```

将新的命令别名 install 映射到实际命令 sudo apt-get -y install。这意味着当你在终端使用 install 时，bash 会将其解释为 sudo apt-get -y install。

### 第4.4节：在系统中定位文件

使用 bash 你可以轻松地用 locate 命令定位文件。例如，假设你正在寻找文件 mykey.pem：

```
locate mykey.pem
```

有时文件名很奇怪，例如你可能有一个文件名为 random7897\_mykey\_0fidw.pem。假设你正在寻找这个文件，但只记得 mykey 和 pem 部分。你可以将 locate命令与 grep 结合使用管道，如下所示：

```
locate pem | grep mykey
```

这将显示包含这两个部分的所有结果。

请注意，并非所有系统都安装了locate工具，且许多安装了该工具的系统也未启用它。 locate 之所以快速高效，是因为它会定期扫描系统并缓存每个文件的名称和位置，但如果未启用该数据收集功能，则无法提供任何信息。你可以使用updatedb手动启动文件系统扫描，以更新文件系统上文件的缓存信息。

如果你没有可用的locate工具，可以使用find工具作为替代：

```
find / -name mykey.pem -print
```

大致相当于locate mykey.pem，但每次运行时都必须扫描你的文件系统以查找指定文件，而不是使用缓存数据。显然，这种方式更慢且效率较低，但更实时。 find工具的功能远不止查找文件，完整功能描述超出本示例范围。

Ctrl-r	reverse search in history
~/.bash_history	stores last 500 commands/events used on the shell
history	will show the command history
history   grep <key-word> will show all the commands in history having keyword <key-word> (useful in cases when you remember part of the command used in the past)	

### Section 4.3: Create Your Own Command Alias

If you are tired of using long commands in bash you can create your own command alias.

The best way to do this is to modify (or create if it does not exist) a file called .bash\_aliases in your home folder. The general syntax is:

```
alias command_alias='actual_command'
```

where actual\_command is the command you are renaming and command\_alias is the new name you have given it. For example

```
alias install='sudo apt-get -y install'
```

maps the new command alias install to the actual command sudo apt-get -y install. This means that when you use install in a terminal this is interpreted by bash as sudo apt-get -y install.

### Section 4.4: Locate a file on your system

Using bash you can easily locate a file with the locate command. For example say you are looking for the file mykey.pem:

```
locate mykey.pem
```

Sometimes files have strange names for example you might have a file like random7897\_mykey\_0fidw.pem. Let's say you're looking for this file but you only remember the mykey and pem parts. You could combine the locate command with grep using a pipe like this:

```
locate pem | grep mykey
```

Which would bring up all results which contain both of these pieces.

Note that not all systems have the locate utility installed, and many that do have not enabled it. locate is fast and efficient because it periodically scans your system and caches the names and locations for every file on it, but if that data collection is not enabled then it cannot tell you anything. You can use updatedb to manually initiate the filesystem scan in order to update the cached info about files on your filesystem.

Should you not have a working locate, you can fall back on the find utility:

```
find / -name mykey.pem -print
```

is roughly equivalent to locate mykey.pem but has to scan your filesystem(s) each time you run it for the file in question, rather than using cached data. This is obviously slower and less efficient, but more real-time. The find utility can do much more than find files, but a full description of its capabilities is beyond the scope of this example.

# 第5章：检查磁盘空间

## 第5.1节：调查目录的磁盘使用情况

有时需要查明哪个目录占用了多少磁盘空间，尤其是在你使用了 `df -h`命令后，发现可用磁盘空间不足时。

**du：**

du命令汇总指定文件集的磁盘使用情况，目录则递归计算。

它通常与`-sh`选项一起使用：

```
-s, --summarize
仅显示每个参数的总计
-h, --human-readable
以人类可读的格式显示大小（例如，1K 234M 2G）
```

要汇总当前目录中文件的磁盘使用情况，我们使用：

```
du -sh *
```

示例输出：

```
572K  Documents
208M  Downloads
4,0K  Music
724K  Pictures
4,0K  Public
4,0K  Templates
4,0K  Videos
```

我们也可以使用以下命令包含隐藏文件：

```
du -sh .[*]* *
```

示例输出：

```
6,3M  .atom
4,0K  .bash_history
4,0K  .bash_logout
8,0K  .bashrc
350M  .cache
195M  .config
12K   .dbus
4,0K  .dmrc
44K   .gconf
60K   .gem
520K  .gimp-2.8
28K   .gnome
4,0K  .ICEauthority
8,3M  .local
8,0K  .nano
404K  .nv
36K   .pki
```

# Chapter 5: Check Disk Space

## Section 5.1: Investigate Directories For Disk Usage

Sometimes it may be required to find out which directory consuming how much disk space especially when you are used `df -h` and realized your available disk space is low.

**du:**

du command summarizes disk usage of the set of FILES, recursively for directories.

It's often uses with `-sh` option:

```
-s, --summarize
display only a total for each argument
-h, --human-readable
print sizes in human readable format (e.g., 1K 234M 2G)
```

For summarizing disk usages of the files in the current directory we use:

```
du -sh *
```

Example output:

```
572K  Documents
208M  Downloads
4,0K  Music
724K  Pictures
4,0K  Public
4,0K  Templates
4,0K  Videos
```

We can also include hidden files with using:

```
du -sh .[*.*]* *
```

Example output:

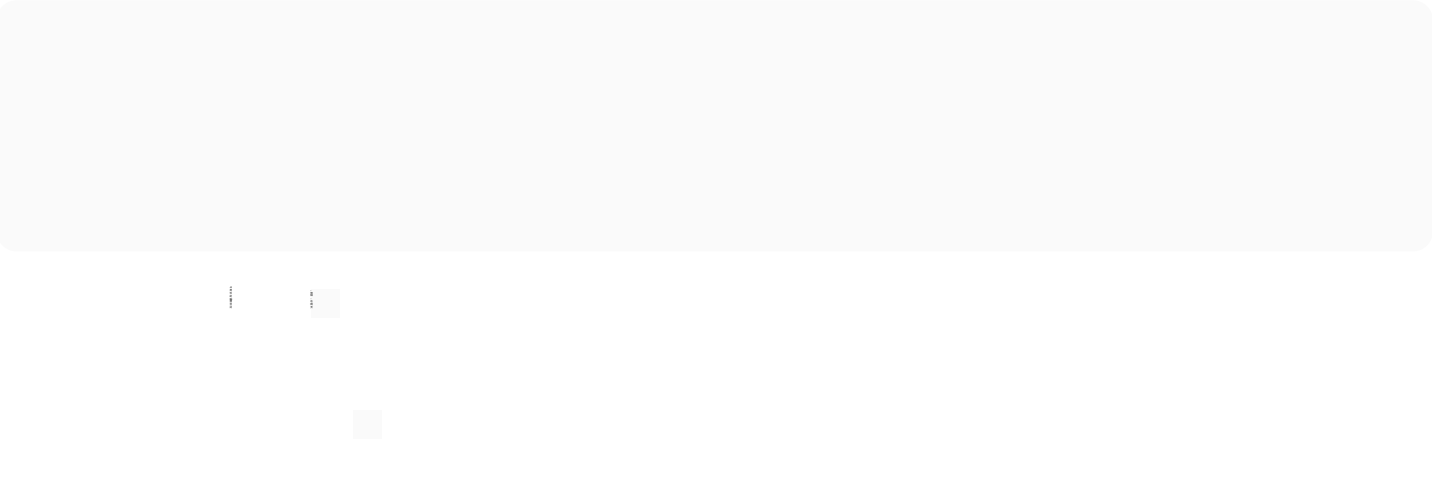
```
6,3M  .atom
4,0K  .bash_history
4,0K  .bash_logout
8,0K  .bashrc
350M  .cache
195M  .config
12K   .dbus
4,0K  .dmrc
44K   .gconf
60K   .gem
520K  .gimp-2.8
28K   .gnome
4,0K  .ICEauthority
8,3M  .local
8,0K  .nano
404K  .nv
36K   .pki
```

```
4,0K  .profile
8,0K  .ssh
0     .sudo_as_admin_successful
4,0K  .Xauthority
4,0K  .xsession-errors
4,0K  .xsession-errors.old
572K  Documents
208M  Downloads
4,0K  Music
724K  Pictures
4,0K  Public
4,0K  Templates
4,0K  Videos
```

第三，您可以通过添加 `,-c` 选项将总计添加到输出中：

```
du -sch .[!~]* *
```

结果：



```
./[!~]* /*
```

示例输出：

```
16K   /.VolumeIcon.icns
24K   /.VolumeIcon.png
13M   /bin
57M   /boot
4,0K  /cdrom
620K  /dev
13M   /etc
779M  /home
0     /initrd.img
406M  /lib
3,9M  /lib32
4,0K  /lib64
16K   /lost+found
4,0K  /media
4,0K  /mnt
367M  /opt
du: 无法访问 '/proc/18221/task/18221/fd/4': 没有那个文件或目录
du: 无法访问 '/proc/18221/task/18221/fdinfo/4': 没有那个文件或目录
du: 无法访问 '/proc/18221/fd/4': 没有那个文件或目录
```

```
4,0K  .profile
8,0K  .ssh
0     .sudo_as_admin_successful
4,0K  .Xauthority
4,0K  .xsession-errors
4,0K  .xsession-errors.old
572K  Documents
208M  Downloads
4,0K  Music
724K  Pictures
4,0K  Public
4,0K  Templates
4,0K  Videos
```

Thirdly, you can add total to the output by adding `,-c`, option:

```
du -sch .[!~]* *
```

Result:



**Most importantly** using `du` command properly on the root directory is a life saving action to find out what application/service or user is consuming your disk space wildly. For example, in case of a ridiculously low level of disk space availability for a web and mail server, the reason could be a spam attack to your mail service and you can diagnose it just by using `du` command.

Investigate root directory for disk usage:

```
sudo du -sch /[!~]* /*
```

Example output:

```
16K   /.VolumeIcon.icns
24K   /.VolumeIcon.png
13M   /bin
57M   /boot
4,0K  /cdrom
620K  /dev
13M   /etc
779M  /home
0     /initrd.img
406M  /lib
3,9M  /lib32
4,0K  /lib64
16K   /lost+found
4,0K  /media
4,0K  /mnt
367M  /opt
du: cannot access '/proc/18221/task/18221/fd/4': No such file or directory
du: cannot access '/proc/18221/task/18221/fdinfo/4': No such file or directory
du: cannot access '/proc/18221/fd/4': No such file or directory
```

```
du: 无法访问 '/proc/18221/fdinfo/4': 没有那个文件或目录
0      /proc
20K    /root
du: 无法访问 '/run/user/1000/gvfs': 权限被拒绝
9,4M   /run
13M    /sbin
4,0K   /srv
0      /sys
72K    /tmp
3,5G   /usr
639M   /var
0      /vmlinuz
5,8G   total
```

最后，最佳方法是在目录中添加一个阈值大小，以忽略较小的目录。该命令只会显示位于根目录下且大小超过1GB的文件夹，直到文件系统整个目录树的最远分支：

```
sudo du --threshold=1G -ch ./[!]* /*
```

示例输出：

```
1,4G   /usr/lib
1,8G   /usr/share
3,5G   /usr
5,8G   total
```

## 第5.2节：检查磁盘空间

通常我们会想检查服务器/计算机上各个分区/驱动器的状态，看看它们的使用情况。你需要运行以下命令：

```
df -h
```

这将产生类似以下的输出：

```
[root@mail ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup-lv_root
19G 1.6G  16G   9% /
tmpfs           245M    0  245M   0% /dev/shm
/dev/sda1       485M  47M  413M  11% /boot
```

在这个基本示例中，我们可以看到 / 分区只使用了 9%。

对于一个更复杂的示例，同时涵盖使用 df 查看各种挂载点，请见下文：

```
[root@mail ~]# df -h
文件系统      大小  已用  可用  使用%  挂载点
/dev/mapper/VG-root  1.9T  1.7T   89G   95% /
/dev/mapper/VG-var   431G  145G  264G   36% /var
devtmpfs       7.8G  204K  7.8G    1% /dev
tmpfs          7.8G  4.0K  7.8G    1% /dev/shm
/dev/md1        495M  126M  344M   27% /boot
ku.example.com:9421  2.5T  487G  2.0T   20% /mnt/test
tmpfs          500M   86M  415M   18% /var/ngx_pagespeed_cache
```

```
du: cannot access '/proc/18221/fdinfo/4': No such file or directory
0      /proc
20K    /root
du: cannot access '/run/user/1000/gvfs': Permission denied
9,4M   /run
13M    /sbin
4,0K   /srv
0      /sys
72K    /tmp
3,5G   /usr
639M   /var
0      /vmlinuz
5,8G   total
```

Lastly, the best method forms when you add a threshold size value for directories to ignore small ones. This command will only show folders with more than 1GB in size which located under root directory up to the farthestmost branch of the whole directory tree in your file system:

```
sudo du --threshold=1G -ch ./[!]* /*
```

Example output:

```
1,4G   /usr/lib
1,8G   /usr/share
3,5G   /usr
5,8G   total
```

## Section 5.2: Checking Disk Space

It's quite common to want to check the status of the various partitions/drives on your server/computer to see how full they are. The following command is the one you'll want to run:

```
df -h
```

This will produce output similar to the following:

```
[root@mail ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup-lv_root
19G 1.6G  16G   9% /
tmpfs           245M    0  245M   0% /dev/shm
/dev/sda1       485M  47M  413M  11% /boot
```

In this basic example, we can see that the / partition only has 9% used.

For a more complex example that also covers using df to see various mountpoints, see below:

```
[root@mail ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/VG-root  1.9T  1.7T   89G   95% /
/dev/mapper/VG-var   431G  145G  264G   36% /var
devtmpfs       7.8G  204K  7.8G    1% /dev
tmpfs          7.8G  4.0K  7.8G    1% /dev/shm
/dev/md1        495M  126M  344M   27% /boot
ku.example.com:9421  2.5T  487G  2.0T   20% /mnt/test
tmpfs          500M   86M  415M   18% /var/ngx_pagespeed_cache
```

在这个示例中，我们有一个 95% 满的 / 分区，以及一个仅 36% 满的 /var 分区。

它有一个挂载在 /mnt/test 的 2T 外部网络挂载，以及一个挂载在 /var/nginx\_pagespeed\_cache 的 500M 内存盘/tmpfs 挂载。

In this example, we have a / partition that's 95% full along with an additional /var partition that's only 36% full.

It's got an external network mount of 2T that's mounted on /mnt/**test** and a ramdisk/tmpfs mount of 500M mounted on /var/nginx\_pagespeed\_cache.



# 第6章：获取系统信息

收集获取系统相关信息的命令。

## 第6.1节：关于CPU、内存、网络 and 磁盘（I/O操作）的统计信息

要获取Linux系列主要组件的一般统计信息，stat命令非常有用

### CPU

要获取处理器相关的统计信息，可以使用mpstat命令，但加上一些选项会提供更好的可见性：

```
$ mpstat 2 10
```

### 内存

我们都知道命令free可以显示（剩余）RAM的数量，但要查看包括I/O操作在内的所有统计信息：

```
$ vmstat 2 10
```

### 磁盘

要实时获取有关磁盘操作的一般信息，您可以使用 iostat。

```
$ iostat -kx 2
```

### 网络

为了能够查看网络服务的运行情况，你可以使用 netstat

```
$ netstat -ntlp # 打开 TCP 套接字
$ netstat -nulp # 打开 UDP 套接字
$ netstat -nxlp # 打开 Unix 套接字
```

但你可以找到有用的监控工具来实时查看网络流量：

```
$ sudo iftop
```

### 可选

要实时生成与所有组件的 I/O 操作相关的统计数据，你可以使用 dstat。该工具是 vmstat、iostat 和 ifstat 的多功能替代品

## 第6.2节：使用 lscpu 和 lshw 等工具

使用像 lscpu 这样的工具，lscpu 是获取 CPU 信息的简便方法。

```
$ lscpu
架构：          x86_64
CPU 操作模式(s)： 32位，64位
字节序：        小端序
```

# Chapter 6: Getting System Information

Collection of commands to fetch system related information.

## Section 6.1: Statistics about CPU, Memory, Network and Disk (I/O operations)

To get general statistics about main components of Linux family of **stat** commands are extremely useful

### CPU

To get processors related statistics you can use mpstat command but with some options it will provide better visibility:

```
$ mpstat 2 10
```

### Memory

We all know command **free** to show amount of (remaining) RAM but to see all statistic including I/O operations:

```
$ vmstat 2 10
```

### Disk

To get general information about your disk operations in real time you can utilise iostat.

```
$ iostat -kx 2
```

### Network

To be able to see what is happening with your network services you can use **netstat**

```
$ netstat -ntlp # open TCP sockets
$ netstat -nulp # open UDP sockets
$ netstat -nxlp # open Unix sockets
```

But you can find useful monitoring to see network traffic in real time:

```
$ sudo iftop
```

### Optional

To generate statistics in real time related to I/O operations across all components you can use dstat. That tool that is a versatile replacement for **vmstat**, iostat and ifstat

## Section 6.2: Using tools like lscpu and lshw

By using tools like lscpu as lscpu is an easy way to get CPU information.

```
$ lscpu
Architecture:      x86_64
CPU op-mode(s):    32-bit, 64-bit
Byte Order:        Little Endian
```

```
CPU(s) : 4
在线 CPU(s) 列表 : 0-3
每核线程数(s) : 1
每插槽核心数(s) : 4
插槽数(s) : 1
NUMA 节点(s) : 1
供应商 ID : GenuineIntel
CPU 家族 : 6
型号 : 23
步进 : 10
CPU 频率 MHz : 1998.000
BogoMIPS : 5303.14
虚拟化 : VT-x
L1d 缓存 : 32K
L1i 缓存 : 32K
L2 缓存 : 2048K
NUMA 节点0 CPU(s) : 0-3
```

使用工具 lshw

```
$ lshw | grep cpu

df1-ws-5084
  描述 : 计算机
  宽度 : 64 位
功能 : vsyscall32
*-核心
描述 : 主板
  物理 ID : 0
*-内存
描述 : 系统内存
  物理ID : 0
大小 : 5881MiB
*-cpu
产品 : Intel(R) 奔腾(R) CPU G3220 @ 3.00GHz
  供应商 : Intel 公司
物理ID : 1
  总线信息 : cpu@0
  大小 : 3GHz
容量 : 3GHz
  宽度 : 64 位
```

### 第6.3节：列出硬件

Ubuntu：

**lshw** 是一个小工具，用于提取机器硬件配置的详细信息。它可以报告准确的内存配置、固件版本、主板配置、CPU版本和速度、缓存配置、总线速度等。

```
$ sudo lshw | less (or more)
$ sudo lshw -html > myhardware.html
$ sudo lshw -xml > myhardware.xml
```

显示 PCI 信息

```
$ lspci -tv
```

```
CPU(s) : 4
On-line CPU(s) list : 0-3
Thread(s) per core : 1
Core(s) per socket : 4
Socket(s) : 1
NUMA node(s) : 1
Vendor ID : GenuineIntel
CPU family : 6
Model : 23
Stepping : 10
CPU MHz : 1998.000
BogoMIPS : 5303.14
Virtualization : VT-x
L1d cache : 32K
L1i cache : 32K
L2 cache : 2048K
NUMA node0 CPU(s) : 0-3
```

By using tool lshw

```
$ lshw | grep cpu

df1-ws-5084
  description: Computer
  width: 64 bits
  capabilities: vsyscall32
*-core
  description: Motherboard
  physical id: 0
*-memory
  description: System memory
  physical id: 0
  size: 5881MiB
*-cpu
  product: Intel(R) Pentium(R) CPU G3220 @ 3.00GHz
  vendor: Intel Corp.
  physical id: 1
  bus info: cpu@0
  size: 3GHz
  capacity: 3GHz
  width: 64 bits
```

### Section 6.3: List Hardware

Ubuntu：

**lshw** is a small tool to extract detailed information on the hardware configuration of the machine. It can report exact memory configuration, firmware version, mainboard configuration, CPU version and speed, cache configuration, bus speed, etc.

```
$ sudo lshw | less (or more)
$ sudo lshw -html > myhardware.html
$ sudo lshw -xml > myhardware.xml
```

To show PCI info

```
$ lspci -tv
```

查看 USB 信息

```
$ lsusb -tv
```

显示 BIOS 信息

```
$ dmidecode -q | less
```

要查看有关磁盘的具体信息（示例中为磁盘 sda），可以使用：

```
$ hdparm -i /dev/sda
```

一些额外的实用工具/命令可以帮助收集更多信息：

```
$ smartctl -A /dev/sda | grep Power_On_Hours # 该磁盘（系统）累计通电时间

$ hdparm -tT /dev/sda # 对磁盘 sda 进行读取速度测试
$ badblocks -s /dev/sda # 测试磁盘 sda 上不可读的块
```

## 第6.4节：查找CPU型号/速度信息

Ubuntu：

```
$ cat /proc/cpuinfo
```

示例输出：

```
处理器      : 0
供应商ID    : GenuineIntel
CPU系列     : 6
型号        : 15
型号名称    : Intel(R) Core(TM)2 Quad CPU    Q6600  @ 2.40GHz
步进号      : 11
CPU MHz     : 1596.000
缓存大小    : 4096 KB
物理ID      : 0
线程数      : 4
核心ID      : 0
CPU核心数   : 4
APIC ID     : 0
初始APIC ID : 0
浮点单元    : 是
浮点异常    : 是
CUID级别    : 10
wp          : 是
flags       : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts
acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx lm constant_tsc arch_perfmon pebs bts rep_good pn
dtes64 monitor ds_cpl vmx est tm2 ssse3 cx16 xtpr pdcm lah_f_lm tpr_shadow vnmi flexpriority
bogomips    : 4800.18
clflush size : 64
cache_alignm ent : 64
address sizes : 36 位物理地址, 48 位虚拟地址
power managem ent:
....
..
processor    : 3
vendor_id    : GenuineIntel
cpu family   : 6
```

To see USB info

```
$ lsusb -tv
```

To display BIOS information

```
$ dmidecode -q | less
```

To see specific information about disk (disk sda in example) you can use:

```
$ hdparm -i /dev/sda
```

Few additional utilities/commands will help gather some extra information:

```
$ smartctl -A /dev/sda | grep Power_On_Hours # How long has this disk (system) been powered on in
total
$ hdparm -tT /dev/sda # Do a read speed test on disk sda
$ badblocks -s /dev/sda # Test for unreadable blocks on disk sda
```

## Section 6.4: Find CPU model/speed information

Ubuntu:

```
$ cat /proc/cpuinfo
```

Sample Output:

```
processor      : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 15
model name    : Intel(R) Core(TM)2 Quad CPU    Q6600  @ 2.40GHz
stepping      : 11
cpu MHz       : 1596.000
cache size    : 4096 KB
physical id   : 0
siblings      : 4
core id       : 0
cpu cores     : 4
apicid        : 0
initial apicid : 0
fpu           : yes
fpu_exception : yes
cpuid level   : 10
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts
acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx lm constant_tsc arch_perfmon pebs bts rep_good pn
dtes64 monitor ds_cpl vmx est tm2 ssse3 cx16 xtpr pdcm lah_f_lm tpr_shadow vnmi flexpriority
bogomips      : 4800.18
clflush size  : 64
cache_alignm ent : 64
address sizes  : 36 bits physical, 48 bits virtual
power managem ent:
....
..
processor      : 3
vendor_id      : GenuineIntel
cpu family     : 6
```

```
型号      : 15
型号名称   : Intel(R) Core(TM)2 Quad CPU    Q6600   @ 2.40GHz
步进号    : 11
CPU MHz    : 1596.000
缓存大小   : 4096 KB
物理ID     : 0
siblings   : 4
core id    : 3
cpu cores  : 4
apicid     : 3
initial apicid : 3
fpu        : 是
浮点异常   : 是
CUID级别   : 10
wp         : 是
flags      : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts
acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx lm constant_tsc arch_perfmon pebs bts rep_good pn
dtes64 monitor ds_cpl vmx est tm2 ssse3 cx16 xtpr pdcm lah_f_lm tpr_shadow vnmi flexpriority
bogomips   : 4800.30
clflush size : 64
cache_alignment : 64
address sizes : 36 位物理地址, 48 位虚拟地址
power management:
```

处理器计数（包括核心）：

```
$ grep -c processor /proc/cpuinfo
```

## 第6.5节：进程监控和信息收集

总体来说，你有两种方式在Linux主机上监控进程

### 静态监控

最常用的命令是ps（即进程状态）命令，用于提供当前运行进程的信息，包括它们的进程标识号（PID）。

这里有一些有用的选项来收集特定信息。

以层级结构列出进程

```
$ ps -e -o pid,args --forest
```

按CPU使用率%排序列出进程

```
$ ps -e -o pcpu,cpu,nice,state,cputime,args --sort pcpu | sed '/^ 0.0 /d'
```

按内存（KB）使用量排序列出进程。

```
$ ps -e -orss=,args= | sort -b -k1,1n | pr -TW$COLUMNS
```

列出特定进程的所有线程（示例中为“firefox-bin”进程）

```
$ ps -C firefox-bin -L -o pid,tid,pcpu,state
```

找到特定进程后，可以使用 lsof 获取与其相关的信息，列出该进程ID打开的路径

```
model      : 15
model name  : Intel(R) Core(TM)2 Quad CPU    Q6600   @ 2.40GHz
stepping    : 11
cpu MHz     : 1596.000
cache size  : 4096 KB
physical id  : 0
siblings    : 4
core id     : 3
cpu cores   : 4
apicid      : 3
initial apicid : 3
fpu         : yes
fpu_exception : yes
cpuid level : 10
wp          : yes
flags       : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts
acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx lm constant_tsc arch_perfmon pebs bts rep_good pn
dtes64 monitor ds_cpl vmx est tm2 ssse3 cx16 xtpr pdcm lah_f_lm tpr_shadow vnmi flexpriority
bogomips    : 4800.30
clflush size : 64
cache_alignment : 64
address sizes : 36 bits physical, 48 bits virtual
power management:
```

count processor (including cores):

```
$ grep -c processor /proc/cpuinfo
```

## Section 6.5: Process monitoring and information gathering

Overall you have two ways to monitor processes at linux host

### Static monitoring

Most widely used command is ps (i.e., process status) command is used to provide information about the currently running processes, including their process identification numbers (PIDs).

Here few useful options to gather specific information.

List processes in a hierarchy

```
$ ps -e -o pid,args --forest
```

List processes sorted by % cpu usage

```
$ ps -e -o pcpu,cpu,nice,state,cputime,args --sort pcpu | sed '/^ 0.0 /d'
```

List processes sorted by mem (KB) usage.

```
$ ps -e -orss=,args= | sort -b -k1,1n | pr -TW$COLUMNS
```

List all threads for a particular process ("firefox-bin" process in example )

```
$ ps -C firefox-bin -L -o pid,tid,pcpu,state
```

After finding specific process you can gather information related to it using **lsof** to list paths that process id has open

```
$ lsof -p $$
```

或者根据路径查找打开指定路径的进程列表

```
$ lsof ~
```

交互式监控

最常用的动态监控工具是：

```
$ top
```

这通常是默认命令，具有大量选项，可实时过滤和显示信息（相比于ps命令）。

不过还有更高级的选项可以考虑并安装作为top的替代品。

```
$ htop -d 5
```

或者

```
$ atop
```

它能够将所有活动记录到日志文件中（默认情况下，atop每600秒记录一次所有活动）。此外，还有一些专用命令，如iotop或iftop。

```
$ sudo iotop
```

```
$ lsof -p $$
```

Or based on path find out list processes that have specified path open

```
$ lsof ~
```

Interactive monitoring

Most commonly known tool for dynamic monitoring is:

```
$ top
```

That mostly default command that have huge amount options to filter and represent information in real time (in comparison to ps command).

Still there are more advance options that can be considered and installed as top replacement

```
$ htop -d 5
```

or

```
$ atop
```

Which has ability to log all the activities into log file (default atop will log all the activity on every 600 seconds) To this list there are few specialised commands as iotop or iftop

```
$ sudo iotop
```

# 第7章：ls命令

## 第7.1节：ls命令的选项

完整选项列表：

ls -a 列出所有文件，包括以'.'开头的隐藏文件

ls --color 彩色列表 [=always/never/auto]

ls -d 列出目录 - 目录名后带 ' \*/'

ls -F 在条目后添加一个字符 \*/=>@|

ls -i 列出文件的inode索引号

ls -l 以长格式列出 - 显示权限

ls -la 以长格式列出包括隐藏文件

ls -lh 以长格式列出，文件大小可读

ls -ls 以长格式列出，带文件大小

ls -r 逆序列出

ls -R 递归列出目录树

ls -s 列出文件大小

ls -S 按文件大小排序

ls -t 按时间和日期排序

ls -X 按扩展名排序

## 第7.2节：带最常用选项的ls命令

ls显示当前工作目录中的文件和目录。（如果没有传递参数。）默认情况下不显示以.开头的隐藏文件。

```
user@ubuntu14:/usr$ ls
bin  games  include  lib  lib32  local  sbin  share  src
```

要查看所有文件（包括隐藏文件/文件夹）。使用 ls -a 或 ls --all

```
user@ubuntu14:/usr$ ls -a
.  ..  bin  games  include  lib  lib32  local  sbin  share  src
```

要区分文件、文件夹、符号链接及其他，使用 ls -F 或 ls --classify

```
user@ubuntu14:~$ ls -F
bash_profile_course  chat_apps/      Desktop/      Downloads/      foxitsoftware/
Public/              test/          bin/          ClionProjects/  Documents/      IDE/          Music/
Pictures/            Templates/      Videos/
```

# Chapter 7: ls command

## Section 7.1: Options for ls command

Full list of options:

ls -a list all files including hidden file starting with '.'

ls --color colored list [=always/never/auto]

ls -d list directories - with ' \*/'

ls -F add one char of \*/=>@| to enteries

ls -i list file's inode index number

ls -l list with long format - show permissions

ls -la list long format including hidden files

ls -lh list long format with readable file size

ls -ls list with long format with file size

ls -r list in reverse order

ls -R list recursively directory tree

ls -s list file size

ls -S sort by file size

ls -t sort by time & date

ls -X sort by extension name

## Section 7.2: ls command with most used options

ls shows files and directories in present working directory. (if no arguments are passed.) (It doesn't show hidden files which starts with . by default.)

```
user@ubuntu14:/usr$ ls
bin  games  include  lib  lib32  local  sbin  share  src
```

To see all files (hidden files/folders also). Use ls -a OR ls --all

```
user@ubuntu14:/usr$ ls -a
.  ..  bin  games  include  lib  lib32  local  sbin  share  src
```

To differentiate between files and folders and symbolic links and other, use ls -F OR ls --classify

```
user@ubuntu14:~$ ls -F
bash_profile_course  chat_apps/      Desktop/      Downloads/      foxitsoftware/
Public/              test/          bin/          ClionProjects/  Documents/      IDE/          Music/
Pictures/            Templates/      Videos/
```



这里，结尾字符用于区分文件和文件夹。

“/” 表示目录。

“\*”表示可执行文件。

“@”表示符号链接。

要获取有关文件和目录的更多详细信息，请使用 `ls -l`

```
user@ubuntu14:~/example$ ls -l
total 6464

-rw-r--r-- 1 dave dave      41 12月 24 12:19 Z.txt
drwxr-xr-x 2 user group   4096 12月 24 12:00 a_directory
-rw-r--r-- 1 user group     6 12月 24 12:01 a_file
lrwxrwxrwx 1 user group     6 12月 24 12:04 a_link -> a_file
-rw-r--r-- 1 user group     6 12月 24 12:03 a_newer_file
-rw-r----- 1 user group 6586816 12月 24 12:07 big.zip
```

在此示例中，内容的总大小为6460KB。

然后按字母顺序列出每个文件/目录的条目，大写字母排在小写字母之前。

第一个字符是类型（例如 d - 目录，l - 链接）。

接下来的9个字符显示用户、组和其他的权限。

接着是硬链接的数量，然后是所有者的名称和组名。

下一个字段是以字节为单位的大小。通过添加-h选项可以以更友好的形式显示，例如：  
6586816 显示为 6.3M

然后是一个时间戳（通常是修改时间）。

最后一个字段是名称。注意：链接还会显示链接的目标。

Here, ending characters are used to distinguish files and folders.

“/” suggest directory.

“\*”suggest executables.

“@” suggest symbolic links.

To get more details about the files and directories, use `ls -l`

```
user@ubuntu14:~/example$ ls -l
total 6464

-rw-r--r-- 1 dave dave      41 Dec 24 12:19 Z.txt
drwxr-xr-x 2 user group   4096 Dec 24 12:00 a_directory
-rw-r--r-- 1 user group     6 Dec 24 12:01 a_file
lrwxrwxrwx 1 user group     6 Dec 24 12:04 a_link -> a_file
-rw-r--r-- 1 user group     6 Dec 24 12:03 a_newer_file
-rw-r----- 1 user group 6586816 Dec 24 12:07 big.zip
```

In this example, the total size of the contents is 6460KB.

Then there is an entry for each file/directory in alphabetical order with upper case before lower case.

The first character is the type (e.g. d - directory, l - link).

The next 9 characters show the permissions for the user, group and other.

This is followed by the number of hard links, then the owner's name and group.

The next field is the size in bytes. This can be displayed in a human friendly form by adding the -h option e.g.  
6586816 is displayed as 6.3M

There then follows a timestamp (usually the modification time).

The final field is the name. Note: links also show the target of the link.

# 第8章：使用'tar'命令进行文件压缩

常用选项	
-c --create	创建一个新归档。
-x --extract	从归档中提取文件。
-t --list	列出归档内容。
-f --file=ARCHIVE	使用归档文件或目录ARCHIVE。
-v --verbose	详细列出处理的文件。
压缩选项 -	
-a --auto-compress	使用归档后缀来确定压缩程序。
-j --bzip2	通过 bzip2 过滤归档文件。
-J --xz --lzma	通过 xz 过滤归档文件。
-z --gzip	通过 gzip 过滤归档文件。

## 第8.1节：压缩文件夹

这将创建一个文件夹的简单归档：

```
tar -cf ./my-archive.tar ./my-folder/
```

详细输出显示哪些文件和目录被添加到归档中，使用 -v 选项：

```
tar -cvf ./my-archive.tar ./my-folder/
```

要归档并压缩为 'gzip' 格式，必须使用 -z 选项：

```
tar -czf ./my-archive.tar.gz ./my-folder/
```

你也可以使用 -j 选项用 'bzip2' 压缩归档：

```
tar -cjf ./my-archive.tar.bz2 ./my-folder/
```

或者使用 -J 选项用 'xz' 压缩：

```
tar -cJf ./my-archive.tar.xz ./my-folder/
```

## 第8.2节：从归档中提取文件夹

下面是一个在当前目录下从归档中提取文件夹的示例：

```
tar -xf archive-name.tar
```

如果您想将归档中的文件夹提取到指定目的地：

```
tar -xf archive-name.tar -C ./目录/目的地
```

## 第8.3节：列出归档内容

列出归档文件的内容而不提取：

# Chapter 8: File Compression with 'tar' command

Common Options	
-c --create	Create a new archive.
-x --extract	Extract files from an archive.
-t --list	List the contents of an archive.
-f --file=ARCHIVE	Use archive file or dir <i>ARCHIVE</i> .
-v --verbose	Verbosely list files processed.
Compression Options -	
-a --auto-compress	Use archive suffix to determine the compression program.
-j --bzip2	Filter the archive through bzip2.
-J --xz --lzma	Filter the archive through xz.
-z --gzip	Filter the archive through gzip.

## Section 8.1: Compress a folder

This creates a simple archive of a folder :

```
tar -cf ./my-archive.tar ./my-folder/
```

Verbose output shows which files and directories are added to the archive, use the -v option:

```
tar -cvf ./my-archive.tar ./my-folder/
```

For archiving a folder compressed 'gzip', you have to use the -z option :

```
tar -czf ./my-archive.tar.gz ./my-folder/
```

You can instead compress the archive with 'bzip2', by using the -j option:

```
tar -cjf ./my-archive.tar.bz2 ./my-folder/
```

Or compress with 'xz', by using the -J option:

```
tar -cJf ./my-archive.tar.xz ./my-folder/
```

## Section 8.2: Extract a folder from an archive

There is an example for extract a folder from an archive in the current location :

```
tar -xf archive-name.tar
```

If you want to extract a folder from an archive to a specific destination :

```
tar -xf archive-name.tar -C ./directory/destination
```

## Section 8.3: List contents of an archive

List the contents of an archive file without extracting it:

```
tar -tf archive.tar.gz
归档中的文件夹/
归档中的文件夹/file1
归档中的文件夹/另一个文件夹/
归档中的文件夹/另一个文件夹/file2
```

## 第8.4节：列出归档内容

这是一个列出内容的示例：

```
tar -tvf archive.tar
```

选项-t用于列出。要列出tar.gz归档的内容，您还必须使用-z选项：

```
tar -tzvf archive.tar.gz
```

## 第8.5节：压缩并排除一个或多个文件夹

如果你想解压一个文件夹，但想在解压过程中排除一个或多个文件夹，可以使用 `--exclude` 选项。

```
tar -cf archive.tar ./my-folder/ --exclude="my-folder/sub1" --exclude="my-folder/sub3"
```

文件夹结构如下：

```
my-folder/
sub1/
sub2/
sub3/
```

结果将是：

```
./archive.tar
my-folder/
sub2/
```

## 第8.6节：去除前导组件

要剥离任意数量的前导组件，请使用 `--strip-components` 选项：

```
--strip-components=NUMBER
从文件名中去除NUMBER个前导组件
```

例如，要去除最前面的文件夹，可以使用：

```
tar -xf --strip-components=1 archive-name.tar
```

```
tar -tf archive.tar.gz
Folder-In-Archive/
Folder-In-Archive/file1
Folder-In-Archive/Another-Folder/
Folder-In-Archive/Another-Folder/file2
```

## Section 8.4: List archive content

There is an example of listing content :

```
tar -tvf archive.tar
```

The option -t is used for the listing. For listing the content of a tar.gz archive, you have to use the -z option anymore :

```
tar -tzvf archive.tar.gz
```

## Section 8.5: Compress and exclude one or multiple folder

If you want to extract a folder, but you want to exclude one or several folders during the extraction, you can use the `--exclude` option.

```
tar -cf archive.tar ./my-folder/ --exclude="my-folder/sub1" --exclude="my-folder/sub3"
```

With this folder tree :

```
my-folder/
sub1/
sub2/
sub3/
```

The result will be :

```
./archive.tar
my-folder/
sub2/
```

## Section 8.6: Strip leading components

To strip any number of leading components, use the `--strip-components` option:

```
--strip-components=NUMBER
strip NUMBER leading components from file names on extraction
```

For example to strip the leading folder, use:

```
tar -xf --strip-components=1 archive-name.tar
```

# 第9章：服务

## 第9.1节：列出Ubuntu上正在运行的服务

要获取系统上的服务列表，可以运行：

```
service --status-all
```

命令 `service --status-all` 的输出列出了由 System V 控制的服务状态。

“+”表示服务正在运行，“-”表示服务已停止。你可以通过运行 `service SERVICENAME status` 来查看 + 和 - 服务的状态。

某些服务由**Upstart**管理。您可以使用 `sudo initctl list` 检查所有 Upstart 服务的状态。任何由 Upstart 管理的服务也会显示在 `service --status-all` 提供的列表中，但会标记为 ?。

ref: <https://askubuntu.com/questions/407075/how-to-read-service-status-all-results>

## 第9.2节：Systemd 服务管理

### 列出服务

- `systemctl` 列出正在运行的服务
- `systemctl --failed` 列出失败的服务

### 管理目标（类似于 SysV 中的运行级别）

- `systemctl get-default` 查找系统的默认目标
- `systemctl set-default <目标名称>` 设置系统的默认目标

### 运行时管理服务

- `systemctl start` [服务名称] 启动服务
- `systemctl stop` [服务名称] 停止服务
- `systemctl restart` [服务名称] 重启服务
- `systemctl reload` [服务名称] 请求服务重新加载配置
- `systemctl status` [服务名称] 显示服务的当前状态

### 管理服务的自动启动

- `systemctl is-enabled` [服务名称] 显示服务是否设置为开机启动
- `systemctl is-active` [服务名称] 显示服务当前是否处于活动（运行）状态
- `systemctl enable` [服务名称] 设置服务为开机启动
- `systemctl disable` [service-name] 用于禁用系统启动时的服务

### 屏蔽服务

- `systemctl mask` [service-name] 用于屏蔽服务（防止服务被误启动）
- `systemctl unmask` [service-name] 用于取消屏蔽服务

### 重启 systemd

```
systemctl daemon-reload
```

# Chapter 9: Services

## Section 9.1: List running service on Ubuntu

To get a list of the service on your system, you may run:

```
service --status-all
```

The output of `service --status-all` lists the state of services controlled by System V.

The + indicates the service is running, - indicates a stopped service. You can see this by running `service SERVICENAME status` for a + and - service.

Some services are managed by **Upstart**. You can check the status of all Upstart services with `sudo initctl list`. Any service managed by Upstart will also show in the list provided by `service --status-all` but will be marked with a ?.

ref: <https://askubuntu.com/questions/407075/how-to-read-service-status-all-results>

## Section 9.2: Systemd service management

### Listing services

- `systemctl` To list running services
- `systemctl --failed` To list failed services

### Managing Targets (Similar to Runlevels in SysV)

- `systemctl get-default` To find the default target for your system
- `systemctl set-default <target-name>` To set the default target for your system

### Managing services at runtime

- `systemctl start` [service-name] To start a service
- `systemctl stop` [service-name] To stop a service
- `systemctl restart` [service-name] To restart a service
- `systemctl reload` [service-name] To request service to reload its configuration
- `systemctl status` [service-name] To show current status of a service

### Managing autostart of services

- `systemctl is-enabled` [service-name] To show whether a service is enabled on system boot
- `systemctl is-active` [service-name] To show whether a service is currently active(running)
- `systemctl enable` [service-name] To enable a service on system boot
- `systemctl disable` [service-name] To disable a service on system boot

### Masking services

- `systemctl mask` [service-name] To mask a service (Makes it hard to start a service by mistake)
- `systemctl unmask` [service-name] To unmask a service

### Restarting systemd

```
systemctl daemon-reload
```

# 第10章：管理服务

## 第10.1节：诊断服务问题

在使用 systemd 的系统上，例如 Fedora 15及以上版本，Ubuntu（服务器和桌面版）15.04及以上版本，以及 RHEL/CentOS 7及以上版本：

```
systemctl status [servicename]
```

...其中 [servicename] 是相关服务；例如，systemctl status sshd。

这将显示基本状态信息和任何最近记录的错误。

你可以使用 **journalctl** 查看更多错误。例如，**journalctl -xe** 会加载最近的1000条日志并进入分页器（如 **less**），跳转到末尾。你也可以使用 **journalctl -f**，实时跟踪日志消息的输出。

要查看特定服务的日志，请使用-t标志，如下所示：

```
journalctl -f -t sshd
```

其他有用的选项包括-p用于优先级（-p warnings表示仅查看警告及以上级别），-b表示“自上次启动以来”，以及-s表示“自.....以来”——结合起来，我们可以这样做

```
journalctl -p err -S yesterday
```

以查看自昨天以来记录为错误的所有条目。

如果journalctl不可用，或者您正在查看不使用系统日志的应用程序错误日志，可以使用tail命令显示文件的最后几行。tail的一个有用标志是-f（表示“跟随”），它使tail在文件追加数据时继续显示。要查看系统上大多数服务的消息：

```
tail -f /var/log/messages
```

或者，如果该服务具有特权，可能会记录敏感数据：

```
tail -f /var/log/secure
```

某些服务有自己的日志文件，一个很好的例子是 auditd，Linux审计守护进程，它的日志存储在/var/log/audit/。如果你在/var/log/messages中没有看到你的服务的输出，试着在/var/log/中查找特定服务的日志。

## 第10.2节：启动和停止服务

在使用System-V风格init脚本的系统上，例如RHEL/CentOS 6：

```
service <service> start
```

```
service <service> stop
```

在使用systemd的系统上，例如Ubuntu（服务器和桌面版）>= 15.04，以及RHEL/CentOS >= 7：

```
systemctl <service> dnsmasq
```

```
systemctl <service> dnsmasq
```

# Chapter 10: Managing Services

## Section 10.1: Diagnosing a problem with a service

On systems using systemd, such as Fedora => 15, Ubuntu (Server and Desktop) >= 15.04, and RHEL/CentOS >= 7:

```
systemctl status [servicename]
```

...where [servicename] is the service in question; for example, **systemctl** status sshd.

This will show basic status information and any recent errors logged.

You can see further errors with **journalctl**. For example, **journalctl -xe** will load the last 1000 logged into a pager (like **less**), jumping to the end. You can also use **journalctl -f**, which will follow log messages as they come in.

To see logs for a particular service, use the -t flag, like this:

```
journalctl -f -t sshd
```

Other handy options include -p for priority (-p warnings to see only warnings and above), -b for "since last boot", and -S for "since" — putting that together, we might do

```
journalctl -p err -S yesterday
```

to see all items logged as errors since yesterday.

If journalctl is not available, or if you are following application error logs which do not use the system journal, the **tail** command can be used to show the last few lines of a file. A useful flag for tail is -f (for "follow"), which causes tail continue showing data as it gets appended to the file. To see messages from most services on the system:

```
tail -f /var/log/messages
```

Or, if the service is privileged, and may log sensitive data:

```
tail -f /var/log/secure
```

Some services have their own log files, a good example is auditd, the linux auditing daemon, which has its logs stored in /var/log/audit/. If you do not see output from your service in /var/log/messages try looking for service specific logs in /var/log/

## Section 10.2: Starting and Stopping Services

On systems that use the System-V style init scripts, such as RHEL/CentOS 6:

```
service <service> start
```

```
service <service> stop
```

On systems using systemd, such as Ubuntu (Server and Desktop) >= 15.04, and RHEL/CentOS >= 7:

```
systemctl <service> dnsmasq
```

```
systemctl <service> dnsmasq
```

## 第10.3节：获取服务状态

在使用System-V风格init脚本的系统上，例如RHEL/CentOS 6：

```
service <service> status
```

在使用systemd的系统上，例如Ubuntu（服务器和桌面版）>= 15.04，以及RHEL/CentOS >= 7.0：

```
systemctl status <service>
```

## Section 10.3: Getting the status of a service

On systems that use the System-V style init scripts, such as RHEL/CentOS 6:

```
service <service> status
```

On systems using systemd, such as Ubuntu (Server and Desktop) >= 15.04, and RHEL/CentOS >= 7.0:

```
systemctl status <service>
```



# 第11章：修改用户

参数	详情
用户名	用户的名称。不要使用大写字母，不要使用点，不要以连字符结尾，必须不包含冒号，不得有特殊字符。不能以数字开头。包含冒号，不得有特殊字符。不能以数字开头。

## 第11.1节：设置您自己的密码

密码
----

## 第11.2节：设置其他用户的密码

以root身份运行以下命令：

passwd 用户名
------------

## 第11.3节：添加用户

以root身份运行以下命令：

useradd 用户名
-------------

## 第11.4节：移除用户

以root身份运行以下命令：

userdel username
------------------

## 第11.5节：删除用户及其主目录

以root身份运行以下命令：

userdel -r username
---------------------

## 第11.6节：列出当前用户所属的组

groups
--------

有关用户和组数字ID的更详细信息，可以使用id命令查看。

## 第11.7节：列出某用户所属的组

groups username
-----------------

有关用户和组数字ID的更详细信息，可以使用id username命令查看。

# Chapter 11: Modifying Users

Parameter	Details
username	The name of the user. Do not use capital letters, do not use dots, do not end it in dash, it must not include colons, no special characters. Cannot start with a number.

## Section 11.1: Setting your own password

passwd
--------

## Section 11.2: Setting another user's password

Run the following as root:

passwd username
-----------------

## Section 11.3: Adding a user

Run the following as root:

useradd username
------------------

## Section 11.4: Removing a user

Run the following as root:

userdel username
------------------

## Section 11.5: Removing a user and its home folder

Run the following as root:

userdel -r username
---------------------

## Section 11.6: Listing groups the current user is in

groups
--------

More detailed information about user and group numerical IDs can be found with the id command.

## Section 11.7: Listing groups a user is in

groups username
-----------------

More detailed information about user and group numerical IDs can be found with **id** username.

# 第12章：LAMP 堆栈

LAMP（Linux Apache MySQL PHP）由Linux操作系统作为开发环境，Apache HTTP服务器作为网页服务器，MySQL关系型数据库管理系统（RDBMS）作为数据库（DB）系统，以及PHP编程语言作为服务器端（后端）编程语言组成。

LAMP作为开源技术栈解决方案被用于网页开发领域。该技术栈的Windows版本称为WAMP（Windows Apache MyS QL PHP）

## 第12.1节：在Arch Linux上安装LAMP

通过这条命令，我们将一步安装所有必要的软件包，并进行最后的更新：

```
pacman -Syu apache php php-apache mariadb
```

### HTTP

编辑

```
/etc/httpd/conf/httpd.conf
```

根据需要更改ServerAdmin的邮箱地址，例如you@example.com。

网页文件夹默认是ServerRoot `" /etc/httpd "`。目录必须设置为相同的文件夹，因此请更改该行

```
目录"/etc/httpd"
```

该文件夹必须具有读取和执行权限，因此

```
chmod o+x /etc/httpd
```

将 AllowOverride 从 none （默认）更改为 All，以便 .htaccess 能正常工作。

现在你需要为每个用户创建 ~/public\_html 文件夹。（以便通过 <http://localhost/~yourusername/> 访问每个用户的根页面。取消注释此行：

```
Include conf/extra/httpd-userdir.conf
```

现在以 root 身份为每个用户创建 ~/public\_html 文件夹，并将每个文件夹的权限更改为（755）。

```
chmod 755 /home
chmod 755 /home/username
chmod 755 /home/username/public_html
```

如果你想使用 SSL，可以注释掉此行：

```
LoadModule ssl_module modules/mod_ssl.so
```

如果需要使用虚拟域名，请取消注释以下行：

```
Include conf/extra/httpd-vhosts.conf
```

并且在/etc/httpd/conf/extra/httpd-vhosts.conf中必须添加所有虚拟域名。（如果想测试这些虚拟域名，还需在/etc/h osts中添加）

# Chapter 12: LAMP Stack

LAMP (**L**inux **A**pache **M**ySQL **P**HP) consists of the Linux operating system as development environment, the Apache HTTP Server as web server, the MySQL relational database management system (RDBMS) as DB (**D**ata **B**ase) system, and the PHP programming language as Server side (Back End) programming language.

LAMP is used as a Open Source stack of technologies solution to web development area. Windows version of this stack is called WAMP (**W**indows **A**pache **M**ySQL **P**HP)

## Section 12.1: Installing LAMP on Arch Linux

With this line we will install all the necessary packages in one step, and the last update:

```
pacman -Syu apache php php-apache mariadb
```

### HTTP

Edit

```
/etc/httpd/conf/httpd.conf
```

Change ServerAdmin you@example.com as you need.

The folder of the WEB Pages by default is ServerRoot `" /etc/httpd "`. Directory must be set to the same folder, so change the line

```
<Directory "/etc/httpd">
```

This folder must have read and execution access, so

```
chmod o+x /etc/httpd
```

Change AllowOverride from none (default) to All so .htaccess will works.

Now you need the ~/public\_html folder for each user. (to get the root page of each user as <http://localhost/~yourusername/>. Unremark this line:

```
Include conf/extra/httpd-userdir.conf
```

Now as root you need to create the ~/public\_html for each user and change the access to (755) of each one.

```
chmod 755 /home
chmod 755 /home/username
chmod 755 /home/username/public_html
```

You can comment out this line if you want to use SSL:

```
LoadModule ssl_module modules/mod_ssl.so
```

If you need to use virtual domains, uncomment the line:

```
Include conf/extra/httpd-vhosts.conf
```

and in /etc/httpd/conf/extra/httpd-vhosts.conf you must to add all the virtual domains. (plus into /etc/hosts if you want to test those virtuals domains)

编辑/etc/httpd/conf/extra/httpd-default.conf，将**ServerSignature**改为Off，**ServerToken**改为Prod以隐藏关键信息

PHP

编辑：/etc/httpd/conf/httpd.conf

注释掉：LoadModule mpm\_event\_module modules/mod\_mpm\_event.so

取消注释：LoadModule mpm\_prefork\_module modules/mod\_mpm\_prefork.so

在LoadModule列表的最后一项添加LoadModule php7\_module modules/libphp7.so

在include列表的最后一项添加Include conf/extra/php7\_module.conf

编辑/etc/php/php.ini

取消注释extension=mysqli.so和extension=pdo\_mysql.so

根据需要更改时区，例如：

date.timezone = America/Argentina/Buenos\_Aires, date.default\_latitude = 0.0, date.default\_longitude = 0.0

MySQL

以 root 身份运行：

mysql\_install\_db --user=mysql --basedir=/usr --datadir=/var/lib/mysql

现在你已经拥有了 MySQL 服务器的 root 权限。

启动 MySQL 守护进程：

```
systemctl enable mysqld
systemctl start mysqld
```

最后，运行：

sh /usr/bin/mysql\_secure\_installation

以上步骤完成后，您的 Web 服务器即可根据需要进行定制。

第12.2节：在 Ubuntu 上安装 LAMP

安装 Apache：

```
sudo apt-get install apache2
```

安装 MySql：

```
sudo apt-get install mysql-server
```

安装 PHP：

Edit /etc/httpd/conf/extra/httpd-default.conf and change **ServerSignature** to Off and **ServerToken** to Prod for hiding critical data

PHP

Edit: /etc/httpd/conf/httpd.conf

Comment out: LoadModule mpm\_event\_module modules/mod\_mpm\_event.so

Uncomment: LoadModule mpm\_prefork\_module modules/mod\_mpm\_prefork.so

As last item in the LoadModule list, add LoadModule php7\_module modules/libphp7.so

As last item in the include list, add Include conf/extra/php7\_module.conf

Edit /etc/php/php.ini

Uncomment extension=mysqli.so and extension=pdo\_mysql.so

Change the timezone as you need, for example:

date.timezone = America/Argentina/Buenos\_Aires, date.default\_latitude = 0.0, date.default\_longitude = 0.0

MySQL

Run as root:

mysql\_install\_db --user=mysql --basedir=/usr --datadir=/var/lib/mysql

Now you have the root of the MySQL Server.

Start MySQL daemon:

```
systemctl enable mysqld
systemctl start mysqld
```

At last, run:

sh /usr/bin/mysql\_secure\_installation

That all to get a web server ready to be customized as you need.

Section 12.2: Installing LAMP on Ubuntu

Install apache:

```
sudo apt-get install apache2
```

Install MySql:

```
sudo apt-get install mysql-server
```

Install PHP:

```
sudo apt-get install php5 libapache2-mod-php5
```

重启系统：

```
sudo systemctl restart apache2
```

检查 PHP 安装：

```
php -r 'echo "Your PHP installation is working fine.";'
```

## 第12.3节：在 CentoOS 上安装 LAMP 堆栈

### 安装 Apache 网络服务器

第一步是安装 Apache 网络服务器。

```
sudo yum -y install httpd
```

安装完成后，启用（设置为开机启动）并启动 Apache 网络服务器服务。

```
sudo systemctl enable --now httpd
```

在浏览器中访问：

<http://localhost>

你将看到默认的 Apache 网络服务器页面。

### 安装 MariaDB 服务器

第二步是安装 MariaDB：

```
sudo yum -y install mariadb-server
```

然后启动并设置 MariaDB 服务器开机自启：

```
sudo systemctl enable --now mariadb
```

根据需要，使用 `mysql_secure_installation` 来保护您的数据库。

该脚本将允许您执行以下操作：

- 更改 root 用户密码
- 删除测试数据库
- 禁用远程访问

### 安装 PHP

```
sudo yum -y install php php-common
```

然后重启 Apache 的 httpd 服务。

```
sudo systemctl restart httpd
```

要测试 PHP，请在 `/var/www/html` 目录下创建一个名为 `index.php` 的文件。

然后将以下行添加到文件中：

```
sudo apt-get install php5 libapache2-mod-php5
```

Restart system:

```
sudo systemctl restart apache2
```

Check PHP installation:

```
php -r 'echo "\n\nYour PHP installation is working fine.\n\n\n";'
```

## Section 12.3: Installing LAMP stack on CentoOS

### Install Apache Web Server

First step is to install web server Apache.

```
sudo yum -y install httpd
```

Once it is installed, enable (to run on startup) and start Apache web server service.

```
sudo systemctl enable --now httpd
```

Point your browser to:

<http://localhost>

You will see the default Apache web server page.

### Install MariaDB Server

Second step is to install MariaDB:

```
sudo yum -y install mariadb-server
```

Then start and enable (on startup) the MariaDB server:

```
sudo systemctl enable --now mariadb
```

As needed, use **mysql\_secure\_installation** to secure your database.

This script will allow you to do the following:

- Change the root user's password
- Remove test databases
- Disable remote access

### Install PHP

```
sudo yum -y install php php-common
```

Then restart Apache's httpd service.

```
sudo systemctl restart httpd
```

To test PHP, create a file called **index.php** in **/var/www/html**.

Then add the following line to the file:

然后将浏览器指向：

<http://localhost/index.php>

你应该能看到与你的服务器相关的信息。如果没有， 请通过运行以下命令确保 PHP 已正确安装：

```
php --version
```

如果你收到类似以下内容：

```
PHP 5.4.16 (cli) (built: Nov 6 2016 00:29:02) Copyright (c) 1997-2013 The PHP Group
```

那么 PHP 已正确安装。如果是这种情况，请确保你已重启了你的网络服务器。

Then point your browser to:

<http://localhost/index.php>

You should see information related to your server. If you do not, ensure that php is for sure installed correctly by running the following command:

```
php --version
```

If you receive something like:

```
PHP 5.4.16 (cli) (built: Nov 6 2016 00:29:02) Copyright (c) 1997-2013 The PHP Group
```

Then PHP is installed correctly. If this is the case, please ensure that you've restarted your web server.

# 第13章：tee命令

选项	描述
-a, --append	追加到指定的文件。不要覆盖。
-i, --ignore-interrupts	忽略中断信号。
--help	显示帮助信息并退出。
--version	显示版本信息，并退出。

tee - 从标准输入读取，并写入标准输出和文件。

tee 命令的名称来源于管道中的 T 型分流器，它将水分成两个方向，形状像大写字母 T。

tee 将数据从标准输入复制到每个文件，同时也复制到标准输出。实际上，tee 复制其输入，将其同时路由到多个输出。

## 第13.1节：将输出写入标准输出，同时写入文件

下面的命令仅在屏幕（标准输出）上显示输出。

```
$ ls
```

以下命令仅将输出写入文件，而不显示在屏幕上。

```
$ ls > file
```

以下命令（借助tee命令）将输出同时写入屏幕（标准输出）和文件。

```
$ ls | tee file
```

## 第13.2节：将管道链中间的输出写入文件并传回管道

你也可以使用tee命令将命令的输出存储到文件中，并将相同的输出重定向到另一个命令。

以下命令将当前的crontab条目写入文件crontab-backup.txt，并将crontab条目传递给sed命令进行替换。替换完成后，将作为新的cron任务添加。

```
$ crontab -l | tee crontab-backup.txt | sed 's/old/new/' | crontab -
```

## 第13.3节：将输出写入多个文件

你可以使用tee将输出通过管道传递到多个文件（包括终端），方法如下：

```
$ ls | tee file1 file2 file3
```

## 第13.4节：指示tee命令追加到文件

默认情况下，tee命令会覆盖文件。你可以使用-a选项指示tee追加到文件，如下所示

# Chapter 13: tee command

Options	Description
-a, --append	Append to the given FILEs. Do not overwrite.
-i, --ignore-interrupts	Ignore interrupt signals.
--help	Display a help message, and exit.
--version	Display version information, and exit.

tee - read from standard input and write to standard output and files.

The tee command is named after the T-splitter in plumbing, which splits water into two directions and is shaped like an uppercase T.

tee copies data from standard input to each FILE, and also to standard output. In effect, tee duplicates its input, routing it to multiple outputs at once.

## Section 13.1: Write output to stdout, and also to a file

The following command displays output only on the screen (stdout).

```
$ ls
```

The following command writes the output only to the file and not to the screen.

```
$ ls > file
```

The following command (with the help of tee command) writes the output both to the screen (stdout) and to the file.

```
$ ls | tee file
```

## Section 13.2: Write output from the middle of a pipe chain to a file and pass it back to the pipe

You can also use tee command to store the output of a command in a file and redirect the same output to another command.

The following command will write current crontab entries to a file crontab-backup.txt and pass the crontab entries to sed command, which will do the substitution. After the substitution, it will be added as a new cron job.

```
$ crontab -l | tee crontab-backup.txt | sed 's/old/new/' | crontab -
```

## Section 13.3: write the output to multiple files

You can pipe your output to multiple files (including your terminal) by using tee like this:

```
$ ls | tee file1 file2 file3
```

## Section 13.4: Instruct tee command to append to the file

By default tee command overwrites the file. You can instruct tee to append to the file using the -a option as shown



如下。

```
$ ls | tee -a file
```

below.

```
$ ls | tee -a file
```

# 第14章：安全外壳（SSH）

安全外壳用于通过加密连接从客户端远程访问服务器。OpenSSH作为Telnet连接的替代方案，后者实现远程shell访问但不加密。OpenSSH客户端默认安装在大多数GNU/Linux发行版中，用于连接服务器。这些示例展示了如何使用SSH套件接受SSH连接和连接到另一台主机。

## 第14.1节：连接到远程服务器

要连接到服务器，我们必须在客户端使用SSH，命令如下，

```
# ssh -p 端口 用户@服务器地址
```

- port - 服务器的监听SSH端口（默认端口22）。
- user - 必须是服务器上具有SSH权限的现有用户。
- server address - 服务器的IP地址或域名。

举个实际例子，假设你正在搭建一个网站。你选择的托管公司告诉你服务器位于web-servers.com，使用自定义端口2020，且你的账户名usr1被选中在服务器上创建一个具有SSH权限的用户。在这种情况下，使用的SSH命令如下

```
# ssh -p 2020 usr1@web-servers.com
```

如果远程系统上的账户名与你本地客户端上的相同，可以省略用户名。  
所以如果你在两个系统上都是usr1，那么你可以直接使用web-servers.com代替usr1@web-servers.com。

当你想连接的服务器无法直接访问时，可以尝试使用ProxyJump选项，通过另一个你可以访问且能连接到目标服务器的服务器进行连接。

```
# ssh -J usr1@10.0.0.1:2020 usr2@10.0.0.2 -p 2222
```

这将允许你通过位于10.0.0.1（SSH运行在端口2020）上的服务器连接到10.0.0.2服务器（SSH运行在端口2222）。当然，你需要在两个服务器上都有账户。另请注意，-J选项是在OpenSSH 7.3版本中引入的。

## 第14.2节：安装OpenSSH套件

连接远程SSH服务器和接受SSH连接都需要安装openssh

Debian :

```
# apt-get install openssh
```

Arch Linux :

```
# pacman -S openssh
```

Yum :

```
# yum install openssh
```

# Chapter 14: Secure Shell (SSH)

A secure shell is used to remotely access a server from a client over an encrypted connection. OpenSSH is used as an alternative to Telnet connections that achieve remote shell access but are unencrypted. The OpenSSH Client is installed on most GNU/Linux distributions by default and is used to connect to a server. These examples show use how to use the SSH suite to for accept SSH connections and connecting to another host.

## Section 14.1: Connecting to a remote server

To connect to a server we must use SSH on the client as follows,

```
# ssh -p port user@server-address
```

- **port** - The listening ssh port of the server (default port 22).
- **user** - Must be an existing user on the server with SSH privileges.
- **server address** - The IP/Domain of the server.

For a real world example lets pretend that you're making a website. The company you chose to host your site tells you that the server is located at **web-servers.com** on a custom port of **2020** and your account name **usr1** has been chosen to create a user on the server with SSH privileges. In this case the SSH command used would be as such

```
# ssh -p 2020 usr1@web-servers.com
```

If account name on the remote system is the same as the one one the local client you may leave the user name off. So if you are usr1 on both systems then you my simply use web-servers.com instead of usr1@web-servers.com.

When a server you want to connect to is not directly accessible to you, you can try using ProxyJump switch to connect to it through another server which is accessible to you and can connect to the desired server.

```
# ssh -J usr1@10.0.0.1:2020 usr2@10.0.0.2 -p 2222
```

This will let you connect to the server 10.0.0.2 (running ssh on port 2222) through server at 10.0.0.1 (running ssh on port 2020). You will need to have accounts on both servers of course. Also note that the -J switch is introduced in OpenSSH version 7.3.

## Section 14.2: Installing OpenSSH suite

Both connecting to a remove SSH server and accepting SSH connections require installation of openssh

Debian:

```
# apt-get install openssh
```

Arch Linux:

```
# pacman -S openssh
```

Yum:

```
# yum install openssh
```

## 第14.3节：配置SSH服务器以接受连接

首先我们必须编辑SSH守护进程配置文件。虽然在不同的Linux发行版中该文件可能位于不同目录，通常它存放在/etc/ssh/sshd\_config

使用您的文本编辑器更改此文件中设置的值，所有以#开头的行均为注释，必须删除此字符才能生效。以下是一些建议列表。

```
端口（选择一个0到65535之间的数字，通常大于四位数）
PasswordAuthentication yes
允许用户      user1 user2 ...等
```

请注意，最好完全禁用密码登录，改用SSH密钥以提高安全性，如本文件中所述。

## 第14.4节：无密码连接（使用密钥对）

首先，你需要有一对密钥。如果还没有，请查看“生成公钥和私钥”主题。

你的密钥对由私钥（id\_rsa）和公钥（id\_rsa.pub）组成。你需要做的就是将公钥复制到远程主机，并将其内容添加到~/.ssh/authorized\_keys文件中。

一个简单的方法是：

```
ssh <user>@<ssh-server> 'cat >> ~/.ssh/authorized_keys' < id_rsa.pub
```

一旦公钥正确放置在用户的主目录中，你只需使用相应的私钥登录：

```
ssh <user>@<ssh-server> -i id_rsa
```

## 第14.5节：生成公钥和私钥

生成SSH客户端密钥：

```
ssh-keygen [-t rsa | rsa1 | dsa ] [-C <注释>] [-b 位数]
```

例如：

```
ssh-keygen -t rsa -b 4096 - C myemail@email.com
```

默认位置为 ~/.ssh/id\_rsa（私钥）和 ~/.ssh/id\_rsa.pub（公钥）。

更多信息请访问 [man.openbsd.org](http://man.openbsd.org)

## 第14.6节：禁用ssh服务

这将禁用SSH服务器端服务，如有需要，可确保客户端无法通过ssh连接

Ubuntu

```
sudo service ssh stop
```

## Section 14.3: Configuring an SSH server to accept connections

First we must edit the SSH daemon config file. Though under different Linux distributions this may be located in different directories, usually it is stored under /etc/ssh/sshd\_config

Use your text editor to change the values set in this file, all lines starting with # are commented out and must have this character removed to take any effect. A list of recommendations follow as such.

```
Port (chose a number between 0 - 65535, normaly greater than four digits)
PasswordAuthentication yes
AllowUsers      user1 user2 ...etc
```

Note that it is preferable to disable password logins all together and use SSH Keys for improved security as explained in this document.

## Section 14.4: Passwordless connection (using a key pair)

First of all you'll need to have a key pair. If you don't have one yet, take a look at the 'Generate public and private key topic'.

Your key pair is composed by a private key (id\_rsa) and a public key (id\_rsa.pub). All you need to do is to copy the public key to the remote host and add its contents to the ~/.ssh/authorized\_keys file.

One simple way to do that is:

```
ssh <user>@<ssh-server> 'cat >> ~/.ssh/authorized_keys' < id_rsa.pub
```

Once the public key is properly placed in your user's home directory, you just need to login using the respective private key:

```
ssh <user>@<ssh-server> -i id_rsa
```

## Section 14.5: Generate public and private key

To generate keys for SSH client:

```
ssh-keygen [-t rsa | rsa1 | dsa ] [-C <comment>] [-b bits]
```

For example:

```
ssh-keygen -t rsa -b 4096 - C myemail@email.com
```

Default location is ~/.ssh/id\_rsa for private and ~/.ssh/id\_rsa.pub for public key.

For more info, please visit [man.openbsd.org](http://man.openbsd.org)

## Section 14.6: Disable ssh service

This will disable the SSH server side service, as if needed this will insure that clients cannot connect via ssh

Ubuntu

```
sudo service ssh stop
```

```
sudo systemctl disable sshd.service
```

*Debian*

```
sudo /etc/init.d/ssh stop  
sudo systemctl disable sshd.service
```

*Arch Linux*

```
sudo killall sshd  
sudo systemctl disable sshd.service
```

```
sudo systemctl disable sshd.service
```

*Debian*

```
sudo /etc/init.d/ssh stop  
sudo systemctl disable sshd.service
```

*Arch Linux*

```
sudo killall sshd  
sudo systemctl disable sshd.service
```

# 第15章：SCP

## 第15.1节：安全复制

scp命令用于安全地将文件复制到远程目的地或从远程目的地复制文件。如果文件在当前工作目录中，只需文件名即可，否则需要包含远程主机名的完整路径，例如：  
remote\_user@some\_server.org:/path/to/file

将本地文件从当前工作目录复制到新目录

```
scp localfile.txt /home/friend/share/
```

将远程文件复制到您当前的工作目录

```
scp rocky@arena51.net:/home/rocky/game/data.txt ./
```

将文件从一个远程位置复制到另一个远程位置

```
scp mars@universe.org:/beacon/light/bitmap.conf jupiter@universe.org:/beacon/night/
```

要复制目录及其子目录，请使用 scp 的 '-r' 递归选项

```
scp -r user@192.168.0.4:~/project/* ./workspace/
```

## 第15.2节：基本用法

```
# 将远程文件复制到本地目录
scp user@remotehost.com:/remote/path/to/foobar.md /local/dest

# 将本地文件复制到远程目录
scp foobar.md 用户@remotehost.com:/remote/dest

# 密钥文件可以使用（就像 ssh 一样）
scp -i my_key.pem foobar.md 用户@remotehost.com:/remote/dest
```

# Chapter 15: SCP

## Section 15.1: Secure Copy

scp command is used to securely copy a file to or from a remote destination. If the file is in current working directly only filename is sufficient else full path is required which included the remote hostname e.g.  
remote\_user@some\_server.org:/path/to/file

Copy local file in your CWD to new directory

```
scp localfile.txt /home/friend/share/
```

Copy remote file to you current working directory

```
scp rocky@arena51.net:/home/rocky/game/data.txt ./
```

Copy file from one remote location to another remote location

```
scp mars@universe.org:/beacon/light/bitmap.conf jupiter@universe.org:/beacon/night/
```

To copy directory and sub-directories use '-r' recursive option to scp

```
scp -r user@192.168.0.4:~/project/* ./workspace/
```

## Section 15.2: Basic Usage

```
# Copy remote file to local dir
scp user@remotehost.com:/remote/path/to/foobar.md /local/dest

# Copy local file to remote dir
scp foobar.md user@remotehost.com:/remote/dest

# Key files can be used (just like ssh)
scp -i my_key.pem foobar.md user@remotehost.com:/remote/dest
```

# 第16章：GnuPG（GPG）

GnuPG 是一个复杂的密钥管理系统，允许安全地签名或加密数据。GPG 是一个命令行工具，用于创建和操作 GnuPG 密钥。

GnuPG 最广泛的用途是实现无密码或任何交互式认证方式的 SSH（安全外壳）连接，这显著提高了安全级别。

以下章节描述了创建、使用和维护 GnuPG 密钥安全的方法。

## 第16.1节：导出你的公钥

为了使你的公私钥对有用，你必须让他人自由获取你的公钥。请确保你这里操作的是公钥，因为你绝不应该分享你的私钥。你可以使用以下命令导出你的公钥：

```
gpg --armor --export EMAIL_ADDRESS > public_key.asc
```

其中 EMAIL\_ADDRESS 是与密钥关联的电子邮件地址

或者，你也可以将公钥上传到公共密钥服务器，如 keys.gnupg.net，以便他人使用。为此，请在终端输入以下内容：

```
gpg --list-keys
```

然后，搜索与您想导出的密钥相关联的8位字符串（主ID）。接着，执行命令：

```
gpg --send-keys PRIMARY_ID
```

其中 PRIMARY\_ID 是该密钥的实际ID。

现在，公钥已上传到密钥服务器并公开可用。

## 第16.2节：快速创建和使用GnuPG密钥

安装 haveged（例如 sudo apt-get install haveged）以加快随机字节生成过程。然后：

```
gpg --gen-key
gpg --list-keys
```

输出：

```
pub 2048R/NNNNNNNNN2016-01-01
uid          名称 <name@example.com>
sub 2048R/xxxxxxxxx2016-01-01
```

然后发布：

```
gpg --keyserver pgp.mit.edu --send-keys NNNNNNNNN
```

然后计划撤销：<https://www.hackdiary.com/2004/01/18/revoking-a-gpg-key/>\_\_\_\_\_

# Chapter 16: GnuPG (GPG)

GnuPG is a sophisticated key management system which allows for secure signing or encrypting data. GPG is a command-line tool used to create and manipulate GnuPG keys.

GnuPG is most widely used for having SSH (Secure Shell) connections without password or any means of interactive authentication, which improves security level significantly.

Following sections describe ways to create, use, and maintain security of GnuPG keys.

## Section 16.1: Exporting your public key

In order for your public-private keypair to be of use, you must make your public key freely available to others. Be sure that you are working with your public key here since you should *never* share your private key. You can export your public key with the following command:

```
gpg --armor --export EMAIL_ADDRESS > public_key.asc
```

where EMAIL\_ADDRESS is the email address associated with the key

Alternately, you can upload your public key to a public key server such as keys.gnupg.net so that others can use it. To do so, enter the following in a terminal:

```
gpg --list-keys
```

Then, search for the 8-digit string (the primary ID) associated with the key you want to export. Then, issue the command:

```
gpg --send-keys PRIMARY_ID
```

where PRIMARY\_ID is the actual ID of that key.

Now, the public key has been uploaded to the key server and is publicly available.

## Section 16.2: Create and use a GnuPG key quickly

Install haveged (example sudo apt-get install haveged) to speed up the random byte process. Then:

```
gpg --gen-key
gpg --list-keys
```

outputs:

```
pub 2048R/NNNNNNNNN 2016-01-01
uid          Name <name@example.com>
sub 2048R/xxxxxxxxx 2016-01-01
```

Then publish:

```
gpg --keyserver pgp.mit.edu --send-keys NNNNNNNNN
```

Then plan to revoke: <https://www.hackdiary.com/2004/01/18/revoking-a-gpg-key/>



# 第17章：网络配置

本文档涵盖TCP/IP网络、网络管理和系统配置基础。Linux可以支持多个网络设备。设备名称以数字编号，从零开始递增。例如，一台有两个网卡的计算机将有两个设备，分别标记为eth0和eth1。

## 第17.1节：本地DNS解析

文件：`/etc/hosts` 包含一个本地解析的主机列表（非通过DNS）

文件示例内容：

```
127.0.0.1      your-node-name.your-domain.com  localhost.localdomain  localhost
XXX.XXX.XXX.XXX  node-name
```

hosts文件的格式由RFC 952规范

## 第17.2节：配置域名解析的DNS服务器

文件：`/etc/resolv.conf` 包含用于域名解析的 DNS 服务器列表

文件示例内容：

```
nameserver 8.8.8.8 # 主名称服务器的 IP 地址
nameserver 8.8.4.4 # 备用名称服务器的 IP 地址
```

如果是内部 DNS 服务器，可以使用 `dig` 命令验证该服务器是否正确解析 DNS 名称：

```
$ dig google.com @your.dns.server.com +short
```

## 第 17.3 节：查看和操作路由

使用 `route` 操作 IP 路由表

显示路由表

```
$ route # 显示路由列表并解析主机名
$ route -n # 显示路由列表但不解析主机名以加快速度
```

添加/删除路由

选项	描述
add 或 del	添加或删除路由
-host x.x.x.x	添加到由IP地址标识的单个主机的路由
-net x.x.x.x	添加到由网络地址标识的网络的路由
gw x.x.x.x.x	指定网络网关
netmask x.x.x.x	指定网络子网掩码
default	添加默认路由

示例：

# Chapter 17: Network Configuration

This document covers TCP/IP networking, network administration and system configuration basics. Linux can support multiple network devices. The device names are numbered and begin at zero and count upwards. For example, a computer with two NICs will have two devices labeled eth0 and eth1.

## Section 17.1: Local DNS resolution

File: `/etc/hosts` contains a list of hosts that are to be resolved locally(not by DNS)

Sample contents of the file:

```
127.0.0.1      your-node-name.your-domain.com  localhost.localdomain  localhost
XXX.XXX.XXX.XXX  node-name
```

The file format for the hosts file is specified by [RFC 952](#)

## Section 17.2: Configure DNS servers for domain name resolution

File: `/etc/resolv.conf` contains a list of DNS servers for domain name resolution

Sample contents of the file:

```
nameserver 8.8.8.8 # IP address of the primary name server
nameserver 8.8.4.4 # IP address of the secondary name server
```

In case internal DNS server you can validate if this server resolve DNS names properly using `dig` command:

```
$ dig google.com @your.dns.server.com +short
```

## Section 17.3: See and manipulate routes

Manipulate the IP routing table using `route`

Display routing table

```
$ route # Displays list or routes and also resolves host names
$ route -n # Displays list of routes without resolving host names for faster results
```

Add/Delete route

Option	Description
add or del	Add or delete a route
-host x.x.x.x	Add route to a single host identified by the IP address
-net x.x.x.x	Add route to a network identified by the network address
gw x.x.x.x.x	Specify the network gateway
netmask x.x.x.x	Specify the network netmask
default	Add a default route

Examples:

- 添加到主机的路由 \$ **route** add **-host** x.x.x.x eth1
- 添加到网络的路由 \$ **route** add **-net** 2.2.2.0 netmask 255.255.255.0 eth0
- 或者，你也可以使用CIDR格式添加到网络的路由 **route** add **-net** 2.2.2.0/24 eth0
- 添加默认网关 \$ **route** add default gw 2.2.2.1 eth0
- 删除一条路由 \$ **路由** del **-net** 2.2.2.0/24

使用 ip 操作 IP 路由表

显示路由表

```
$ ip 路由 show # 列出路由表
```

添加/删除路由

选项	描述
添加 或 删除 或 更改 或 追加 或 替换	更改一条路由
显示 或 清空	该命令显示路由表内容或将其删除
恢复	从标准输入恢复路由表信息
获取	该命令获取到达目的地的单条路由，并按内核看到的内容精确打印

示例：

- 将默认网关设置为 1.2.3.254 \$ **ip route** add default via 1.2.3.254
- 通过本地网关 192.168.1.1 添加默认路由（适用于所有地址），该网关可通过设备 eth0 访问  
\$ ip route add default via 192.168.1.1 dev eth0

## 第17.4节：为网络上的其他系统配置主机名

您可以配置您的Linux（或macOS）系统，将标识符<hostname>绑定到网络中其他系统的IP地址。您可以这样配置：

- 系统范围内。您应修改/etc/hosts文件。只需在该文件中添加一行，内容包括：
  1. 远程系统的IP地址<ip\_rem>，一个或多个空格，以及
  3. 标识符<hostname>。
- 针对单个用户。您应修改 ~/.hosts 文件——您需要创建该文件。它不像系统范围内配置那么简单。这里有一个说明。

例如，您可以使用cat Unix工具添加此行。假设您想对本地网络中IP地址为192.168.1.44的PC进行ping，并且想用remote\_pc来代替该IP地址。那么您必须在shell中输入：

```
$ sudo cat 192.168.1.44 remote_pc
```

然后你可以通过以下方式进行ping操作：

```
$ ping remote_pc
```

- add route to a host \$ **route** add **-host** x.x.x.x eth1
- add route to a network \$ **route** add **-net** 2.2.2.0 netmask 255.255.255.0 eth0
- Alternatively, you could also use cidr format to add a route to network **route** add **-net** 2.2.2.0/24 eth0
- add default gateway \$ **route** add default gw 2.2.2.1 eth0
- delete a route \$ **route** del **-net** 2.2.2.0/24

Manipulate the IP routing table using ip

Display routing table

```
$ ip route show # List routing table
```

Add/Delete route

Option	Description
add or del or change or append or replace	Change a route
show or flush	the command displays the contents of the routing tables or remove it
restore	restore routing table information from stdin
get	this command gets a single route to a destination and prints its contents exactly as the kernel sees it

Examples:

- Set default gateway to 1.2.3.254 \$ **ip route** add default via 1.2.3.254
- Adds a default route (for all addresses) via the local gateway 192.168.1.1 that can be reached on device eth0  
\$ **ip route** add default via 192.168.1.1 dev eth0

## Section 17.4: Configure a hostname for some other system on your network

You can configure your Linux (or macOS) system in order to tie in an identifier **<hostname>** to some other system's IP address in your network. You can configure it:

- Systemwide. You should modify the */etc/hosts* file. You just have to add to that file a new line containing:
  1. the remote system's IP address **<ip\_rem>**,
  2. one or more blank spaces, and
  3. the identifier **<hostname>**.
- For a single user. You should modify the *~/.hosts* file --- you-d have to create it. It is not as simple as for systemwide. [Here](#) you can see an explanation.

For instance, you could add this line using the **cat** Unix tool. Suppose that you want to make a **ping** to a PC in yout local network whose IP address is 192.168.1.44 and you want to refer to that IP address just by remote\_pc. Then you must write on your shell:

```
$ sudo cat 192.168.1.44 remote_pc
```

Then you can make that ping just by:

```
$ ping remote_pc
```

第17.5节：接口详情

Ifconfig

列出机器上所有可用的接口

```
$ ifconfig -a
```

列出特定接口的详细信息

语法：\$ ifconfig <interface>

示例：

```
$ ifconfig eth0
eth0      链路封装：以太网  硬件地址  xx:xx:xx:xx:xx:xx
          inet  地址：x.x.x.x  广播地址：x.x.x.x  子网掩码：x.x.x.x
          inet6 地址： xxxx::xxx:xxxx:xxxx:xxxx/64 范围：链路
          状态：启动 广播 运行 多播  MTU:1500  指标:1
          接收数据包:4426618 错误:0 丢弃:1124 超限:0 帧错误:0
          发送数据包:189171 错误:0 丢弃:0 超限:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节数:382611580 (382.6 MB) 发送字节数:36923665 (36.9 MB)
          中断:16 内存:fb5e0000-fb600000
```

Ethtool - 查询网络驱动程序和硬件设置

语法：\$ ethtool <接口>

示例：

```
$ ethtool eth0

eth0 的设置：
支持的端口：[ TP ]
支持的链路模式：  10baseT/半双工 10baseT/全双工
100baseT/半双工 100baseT/全双工
1000baseT/全双工
支持暂停帧使用：否
支持自动协商：是
通告的链路模式：  10baseT/半双工 10baseT/全双工
100baseT/半双工 100baseT/全双工
1000baseT/全双工
通告的暂停帧使用：否
通告的自动协商：是
速度：1000Mb/s
双工模式：全双工
端口：双绞线
PHY地址：1
收发器：内部
自动协商：开启
MDI-X：开启（自动）
支持唤醒方式：pumbg
唤醒方式：g
当前消息级别：0x00000007 (7)
drv 探测链接
检测到链接：是
```

ip - 显示 / 操作路由、设备、策略路由和隧道

Section 17.5: Interface details

Ifconfig

List all the interfaces available on the machine

```
$ ifconfig -a
```

List the details of a specific interface

Syntax: \$ ifconfig <interface>

Example:

```
$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr xx:xx:xx:xx:xx:xx
          inet  addr:x.x.x.x  Bcast:x.x.x.x  Mask:x.x.x.x
          inet6  addr: xxxx::xxx:xxxx:xxxx:xxxx/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4426618 errors:0 dropped:1124 overruns:0 frame:0
          TX packets:189171 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:382611580 (382.6 MB)  TX bytes:36923665 (36.9 MB)
          Interrupt:16 Memory:fb5e0000-fb600000
```

Ethtool - query the network driver and hardware settings

Syntax: \$ ethtool <interface>

Example:

```
$ ethtool eth0

Settings for eth0:
Supported ports: [ TP ]
Supported link modes:  10baseT/Half 10baseT/Full
100baseT/Half 100baseT/Full
1000baseT/Full
Supported pause frame use: No
Supports auto-negotiation: Yes
Advertised link modes:  10baseT/Half 10baseT/Full
100baseT/Half 100baseT/Full
1000baseT/Full
Advertised pause frame use: No
Advertised auto-negotiation: Yes
Speed: 1000Mb/s
Duplex: Full
Port: Twisted Pair
PHYAD: 1
Transceiver: internal
Auto-negotiation: on
MDI-X: on (auto)
Supports Wake-on: pumbg
Wake-on: g
Current message level: 0x00000007 (7)
drv probe link
Link detected: yes
```

ip - show / manipulate routing, devices, policy routing and tunnels

语法：\$ **ip { link | ... | route | macsec }** （完整对象列表请参见 **man ip**）

示例

列出网络接口

```
$ ip link show
```

将接口 eth0 重命名为 wan

```
$ ip link set dev eth0 name wan
```

启用（或禁用）接口 eth0

```
$ ip link set dev eth0 up
```

列出接口地址

```
$ ip addr show
```

添加（或删除）IP和掩码（255.255.255.0）

```
$ ip addr add 1.2.3.4/24 brd + dev eth0
```

## 第17.6节：向接口添加IP

接口的IP地址可以通过DHCP或静态分配获得

DHCP 如果你连接到运行DHCP服务器的网络，dhclient 命令可以为你的接口获取IP地址

```
$ dhclient <interface>
```

或者，你也可以修改/etc/network/interfaces文件，使接口在启动时被激活并通过DHCP获取IP

```
auto eth0
iface eth0 inet dhcp
```

使用/etc/network/interfaces文件进行静态配置（永久更改）

如果您想永久配置接口设置，可以在  
/etc/network/interfaces文件中进行。

示例：

```
auto eth0 # 启动时激活接口
iface eth0 inet static
address 10.10.70.10
netmask 255.255.0.0
gateway 10.10.1.1
dns-nameservers 10.10.1.20
dns-nameservers 10.10.1.30
```

这些更改在系统重启后依然有效。

Syntax: \$ **ip { link | ... | route | macsec }** (please see **man ip** for full list of objects)

Examples

List network interfaces

```
$ ip link show
```

Rename interface eth0 to wan

```
$ ip link set dev eth0 name wan
```

Bring interface eth0 up (or down)

```
$ ip link set dev eth0 up
```

List addresses for interfaces

```
$ ip addr show
```

Add (or del) ip and mask (255.255.255.0)

```
$ ip addr add 1.2.3.4/24 brd + dev eth0
```

## Section 17.6: Adding IP to an interface

An IP address to an interface could be obtained via DHCP or Static assignment

**DHCP** If you are connected to a network with a DHCP server running, dhclient command can get an IP address for your interface

```
$ dhclient <interface>
```

or alternatively, you could make a change to the /etc/network/interfaces file for the interface to be brought up on boot and obtain DHCP IP

```
auto eth0
iface eth0 inet dhcp
```

Static configuration(Permanent Change) using /etc/network/interfaces file

If you want to statically configure the interface settings(permanent change), you could do so in the /etc/network/interfaces file.

Example:

```
auto eth0 # Bring up the interface on boot
iface eth0 inet static
address 10.10.70.10
netmask 255.255.0.0
gateway 10.10.1.1
dns-nameservers 10.10.1.20
dns-nameservers 10.10.1.30
```

These changes persist even after system reboot.

## 使用ifconfig工具进行静态配置（临时更改）

可以使用ifconfig工具为接口添加静态IP地址，方法如下

```
$ ifconfig <interface> <ip-address>/<mask> up
```

示例：

```
$ ifconfig eth0 10.10.50.100/16 up
```

## Static configuration(Temporary change) using ifconfig utility

A static IP address could be added to an interface using the ifconfig utility as follows

```
$ ifconfig <interface> <ip-address>/<mask> up
```

Example:

```
$ ifconfig eth0 10.10.50.100/16 up
```

# 第18章：午夜指挥官

午夜指挥官（Midnight Commander，简称mc）是一款控制台文件管理器。本章节包括其功能描述以及如何充分利用它的示例和技巧。

## 第18.1节：午夜指挥官浏览模式下的功能键

以下是在午夜指挥官文件系统浏览模式中，可以通过键盘上的功能键触发的一系列操作。

- F1 显示帮助
- F2 打开用户菜单
- F3 显示所选文件的内容
- F4 在内部文件编辑器中打开所选文件
- F5 将所选文件复制到第二面板打开的目录
- F6 将选定的文件移动到第二个面板中打开的目录
- F7 在当前面板中打开的目录中新建一个目录
- F8 删除选定的文件或目录
- F9 将焦点移至屏幕顶部的主菜单
- F10 退出 mc

## 第18.2节：午夜指挥官文件编辑模式中的功能键

Midnight Commander 内置了一个编辑器，当在浏览模式中选中文件时，按F4功能键即可启动。也可以通过执行以下命令以独立模式调用

```
mcedit <filename>
```

以下是编辑模式下可以触发的操作列表。

- F1 显示帮助
- F2 保存当前文件
- F3 标记文本选择的开始。移动光标任意方向进行选择。第二次按下标记选择的结束。
- F4 调出文本查找/替换对话框
- F5 将选中文本复制到光标位置（复制/粘贴）
- F6 将选中文本移动到光标位置（剪切/粘贴）
- F7 调出文本查找对话框

# Chapter 18: Midnight Commander

Midnight Commander or mc is a console file manager. This topic includes the descripton of it's functionalities and examples and tips of how to use it to it's full potential.

## Section 18.1: Midnight Commander function keys in browsing mode

Here is a list of actions which can be triggered in the Midnight Commander filesystem browsing mode by using function keys on your keyboard.

- F1 Displays help
- F2 Opens user menu
- F3 Displays the contents of the selected file
- F4 Opens the selected file in the internal file editor
- F5 Copies the selected file to the directory open in the second panel
- F6 Moves the selected file to the directory open in the second panel
- F7 Makes a new directory in the directory open in the current panel
- F8 Deletes the selected file or directory
- F9 Focuses to the main menu on the top of the screen
- F10 Exits mc

## Section 18.2: Midnight Commander function keys in file editing mode

Midnight Commander has a built in editor which is started by F4 function key when over the desired file in the browse mode. It can also be invoked in standalone mode by executing

```
mcedit <filename>
```

Here is a list of actions which can be triggered in the edit mode.

- F1 Displays help
- F2 Saves current file
- F3 Marks the start of the text selection. Move cursor any direction to select. Second hit marks the end of the selection.
- F4 Brings up the text search/replace dialog
- F5 Copies selected text to the cursor location (copy/paste)
- F6 Moves selected text to the cursor location (cut/paste)
- F7 Brings up the text search dialog



F8 删除选定文本

F9 将焦点移至屏幕顶部的主菜单

F10 退出编辑器

F8 Deletes selected text

F9 Focuses to the main menu on the top of the screen

F10 Exits the editor

# 第19章：更改根目录（chroot）

更改根目录（chroot）是一种操作，它更改当前运行进程及其子进程的表观根目录。在这样修改的环境中运行的程序无法访问该环境目录树之外的文件和命令。

## 第19.1节：要求

- root权限
- 另一个可用的Linux工作环境，例如Live CD启动或现有发行版匹配chroot源和目标环境
- 架构（使用uname -m检查当前环境架构）
- 可能需要在chroot环境中加载的内核模块（例如，使用modprobe）

## 第19.2节：在目录中手动更改根目录

- 1.确保满足所有要求，如“要求”所述
- 2.挂载临时API文件系统：

```
cd /location/new/root
mount -t proc proc proc/
mount --rbind /sys sys/
mount --rbind /dev dev/
mount --rbind /run run/ (optionally)
```
- 3.如果需要在chroot环境中使用网络连接，请复制DNS配置：

```
cp /etc/resolv.conf etc/resolv.conf
```
4. 切换根目录到 /location/of/new/root，指定shell（本例中为/bin/bash）：

```
chroot /location/of/new/root /bin/bash
```
- 5.进入chroot后，可能需要加载本地bash配置：

```
source /etc/profile
source ~/.bashrc
```
- 6.可选，创建一个独特的提示符以区分你的chroot环境：

```
export PS1="(chroot) $PS1"
```
- 7.完成chroot操作后，可以通过以下命令退出：

```
退出
```
8. 卸载临时文件系统：

```
cd /
umount --recursive /location/of/new/root
```

# Chapter 19: Change root (chroot)

Change root (chroot) is an operation that changes the apparent root directory for the current running process and their children. A program that is run in such a modified environment cannot access files and commands outside that environmental directory tree.

## Section 19.1: Requirements

- root privileges
- another working Linux environment,such as Live CD boot or an existing distribution
- matching environment architectures of **chroot** source and destination (check current environment architecture with **uname -m**)
- kernel modules which you may need in **chroot** environment must be loaded (for example, with modprobe)

## Section 19.2: Manually changing root in a directory

1. Ensure you met all requirements, as per Requirements
2. Mount the temporary API filesystems:

```
cd /location/of/new/root
mount -t proc proc proc/
mount --rbind /sys sys/
mount --rbind /dev dev/
mount --rbind /run run/ (optionally)
```
3. If you need to use an internet connection in the chroot environment, copy over the DNS details:

```
cp /etc/resolv.conf etc/resolv.conf
```
4. Change root into /location/of/new/root, specifying the shell (/bin/**bash** in this example):

```
chroot /location/of/new/root /bin/bash
```
5. After chrooting it may be necessary to load the local bash configuration:

```
source /etc/profile
source ~/.bashrc
```
6. Optionally, create a unique prompt to be able to differentiate your chroot environment:

```
export PS1="(chroot) $PS1"
```
7. When finished with the chroot, you can exit it via:

```
exit
```
8. Unmount the temporary file systems:

```
cd /
umount --recursive /location/of/new/root
```

## 第19.3节：使用chroot的原因

更改根目录通常用于在无法启动和/或登录的系统上执行系统维护。

常见示例有：

- 重新安装引导加载程序
- 重建initramfs镜像
- 升级或降级软件包
- 重置忘记密码
- 在干净的根环境中构建软件

## Section 19.3: Reasons to use chroot

Changing root is commonly done for performing system maintenance on systems where booting and/or logging in is no longer possible.

Common examples are:

- reinstalling the bootloader
- rebuilding the initramfs image
- upgrading or downgrading packages
- resetting a forgotten password
- building software in a clean root environment

# 第20章：软件包管理器

## 第20.1节：如何使用apt软件包管理器更新软件包

高级包管理工具，恰如其名的“apt”包管理器，可以处理Debian、Slackware及其他Linux发行版上的软件安装和卸载。以下是一些简单的使用示例：

**更新**  
此选项检索并扫描 Packages.gz 文件，以便获取有关新软件包和更新软件包的信息。要执行此操作，请输入以下命令：

```
sudo apt-get update
```

**升级**  
此选项用于安装系统上当前已安装的所有软件包的最新版本。对于当前已安装且有新版本的软件包，将检索并升级；在任何情况下，当前已安装的软件包不会被删除，未安装的软件包也不会被检索和安装。要升级，请输入以下命令：

```
sudo apt-get upgrade
```

**发行版升级**  
除了执行升级的功能外，发行版升级（dist-upgrade）还会智能处理软件包新版本中依赖关系的变化。如果必要，它会以牺牲较不重要的软件包为代价，尝试升级最重要的软件包。要执行此操作，请输入以下命令：

```
sudo apt-get dist-upgrade
```

## 第20.2节：如何使用 pacman 软件包管理器安装软件包

为了在数据库中搜索软件包，同时搜索软件包的名称和描述：

```
pacman -Ss string1 string2 ...
```

要安装单个软件包或软件包列表（包括依赖项），请执行以下命令：

```
sudo pacman -S package_name1 package_name2 ...
```

[source](#)

## 第20.3节：如何使用pacman软件包管理器更新软件包

要更新特定程序：

```
sudo pacman -S <programName>
```

要更新整个系统：

```
sudo pacman -Syu
```

# Chapter 20: Package Managers

## Section 20.1: How to update packages with the apt package manager

The **A**dvanced **P**ackage **T**ool, aptly named the 'apt' package manager can handle the installation and removal of software on the Debian, Slackware, and other Linux Distributions. Below are some simple examples of use:

**update**  
This option retrieves and scans the Packages.gz files, so that information about new and updated packages is available. To do so, enter the following command:

```
sudo apt-get update
```

**upgrade**  
This option is used to install the newest versions of all packages currently installed on the system. Packages currently installed with new versions available are retrieved and upgraded; under no circumstances are currently installed packages removed, or packages not already installed retrieved and installed. To upgrade, enter the following command:

```
sudo apt-get upgrade
```

**dist-upgrade**  
In addition to performing the function of upgrade, dist-upgrade also intelligently handles changing dependencies with new versions of packages. It will attempt to upgrade the most important packages at the expense of less important ones if necessary. To do so, enter the following command:

```
sudo apt-get dist-upgrade
```

## Section 20.2: How to install a package with the pacman package manager

In order to search for packages in the databse, searching both in packages' names and descriptions:

```
pacman -Ss string1 string2 ...
```

To install a single package or list of packages (including dependencies), issue the following command:

```
sudo pacman -S package_name1 package_name2 ...
```

[source](#)

## Section 20.3: How to update packages with the pacman package manager

To update a specific program:

```
sudo pacman -S <programName>
```

To update entire the system:

```
sudo pacman -Syu
```

## 第20.4节：如何使用yum更新软件包

Yellowdog 更新器（Yellowdog Updater，修改版），是 Yellow Dog Linux 剩余的最后遗迹之一，是 Red Hat、Fedora 和 CentOS 系统及其衍生版本使用的软件包管理器。它可以处理这些 Linux 发行版中以 rpm 格式打包的软件的安装和卸载。以下是一些简单的使用示例：

### 搜索

该命令将尝试在配置的软件仓库中查找符合给定搜索条件的软件包，并显示找到的匹配项的名称/版本/仓库位置。使用时，输入以下命令：

```
yum search <queryString>
```

### 安装

该命令将尝试从配置的软件仓库中查找并安装指定的软件，同时递归查找并安装任何所需的前置软件。使用时，输入以下命令：

```
sudo yum install <packageName>
```

### 更新

此选项用于安装系统上当前已安装的所有软件包的最新版本。对于当前已安装且有新版本的软件包，将下载并升级；必要时也会下载并安装新的前置软件包，同时移除被替换或废弃的软件包。要升级，请输入以下命令：

```
sudo yum update
```

与 apt 不同，大多数 yum 命令还会自动检查仓库元数据的更新（如果最近未检查过，或被强制检查），并下载和扫描更新的元数据，以便在执行请求的操作之前获取有关新软件包和更新软件包的信息。

## Section 20.4: How to update packages with yum

Yellowdog Updater, Modified, one of the last remaining vestiges of Yellow Dog Linux, is the package manager used by Red Hat, Fedora, and CentOS systems and their derivatives. It can handle the installation and removal of software packaged as **rpms** for these Linux distributions. Below are some simple examples of use:

### search

This command will attempt to locate software packages in the configured software repositories that match the given search criteria, and display the name / version / repository location of the matches it finds. To use it, enter the following command:

```
yum search <queryString>
```

### install

This command will attempt to locate and install the named software from the configured software repositories, recursively locating and installing any needed prerequisite software as well. To use it, enter the following command:

```
sudo yum install <packageName>
```

### update

This option is used to install the newest versions of all packages currently installed on the system. Packages currently installed with new versions available are retrieved and upgraded; new prerequisites are also retrieved and installed as necessary, and replaced or obsoleted packages are removed. To upgrade, enter the following command:

```
sudo yum update
```

Unlike **apt**, most **yum** commands will also automatically check for updates to repository metadata if a check has not been done recently (or if forced to do so) and will retrieve and scan updated metadata so that information about new and updated packages is available before the requested operation is performed.

# 第21章：编译Linux内核

## 第21.1节：在Ubuntu上编译Linux内核

警告： 请确保至少有15 GB的可用磁盘空间。

### 在Ubuntu >=13.04上编译

#### 选项A) 使用Git

如果您想保持与最新Ubuntu内核源码同步，请使用git。详细说明可参见内核Git指南。git仓库不包含必要的控制文件，因此您必须通过以下方式构建它们：

```
fakeroot debian/rules clean
```

#### 选项B) 下载源码归档

下载源码归档——适用于想要在标准Ubuntu包基础上添加补丁重新构建的用户。使用以下命令安装构建依赖并解压源码（到当前目录）：

- 1.安装以下软件包：

```
sudo apt-get build-dep linux-image-`uname -r`
```

#### 选项 C) 下载源代码包并构建

这是针对想要修改或玩转Ubuntu补丁内核源码的用户。

1. 从kernel.org获取最新的内核源码。 \_\_\_\_\_
2. 将压缩包解压到一个目录中，并cd进入该目录：

```
tar xf linux-*.tar.xz
cd linux-*
```

3. 构建ncurses配置界面：

```
make menuconfig
```

4. 要接受默认配置，按下  以高亮显示< 退出 >然后 .
5. 按  再次保存配置。
6. 使用make来构建内核：

```
make
```

请注意，您可以使用-jem>标志来并行编译文件，充分利用多核处理器。

压缩的内核映像可以在arch/[arch]/boot/bzImage找到，其中[arch]等于uname -a。

# Chapter 21: Compiling the Linux kernel

## Section 21.1: Compilation of Linux Kernel on Ubuntu

**Warning:** be sure you have at least 15 GB of free disk space.

### Compilation in Ubuntu >=13.04

#### Option A) Use Git

Use git if you want to stay in sync with the latest Ubuntu kernel source. Detailed instructions can be found in the Kernel Git Guide. The git repository does not include necessary control files, so you must build them by:

```
fakeroot debian/rules clean
```

#### Option B) Download the source archive

Download the source archive - This is for users who want to rebuild the standard Ubuntu packages with additional patches. Use a follow command to install the build dependencies and extract the source (to the current directory):

1. Install the following packages:

```
sudo apt-get build-dep linux-image-`uname -r`
```

#### Option C) Download the source package and build

This is for users who want to modify, or play around with, the Ubuntu-patched kernel source.

1. Retrieve the latest kernel source from [kernel.org](http://kernel.org).
2. Extract the archive to a directory and cd into it:

```
tar xf linux-*.tar.xz
cd linux-*
```

3. Build the ncurses configuration interface:

```
make menuconfig
```

4. To accept the default configuration, press  to highlight < Exit > and then .
5. Press  again to save the configuration.
6. Use **make** to build the kernel:

```
make
```

Note that you can use the -jem> flag to compile files in parallel and take advantage of multiple cores.

The compressed kernel image can be found at arch/[**arch**]/boot/bzImage, where [**arch**] is equal to **uname -a**.



# 致谢

非常感谢所有来自Stack Overflow Documentation的人员提供此内容，  
如需发布或更新新内容，可将[更多更改发送至web@petercv.com](mailto:web@petercv.com)

<a href="#">7theo.tk</a>	第1章
<a href="#">亚伦·斯科姆拉</a>	第16章
<a href="#">阿贾伊·桑加莱</a>	第1章和第9章
<a href="#">阿纳格·赫格德</a>	第4章和第14章
<a href="#">阿尼·梅农</a>	第2章和第4章
<a href="#">阿登·沙克尔福德</a>	第12章
<a href="#">阿马利</a>	第1章
<a href="#">巴尔德·科佩鲁德</a>	第8章
<a href="#">布莱特一号</a>	第9、13、14和19章
<a href="#">C.W.霍尔曼二世</a>	第14章
<a href="#">caped114</a>	第1章和第4章
<a href="#">colelemonz</a>	第1章
<a href="#">ctafur</a>	第17章
<a href="#">戴夫·M</a>	第2章
<a href="#">depperm</a>	第1章
<a href="#">e.dan</a>	第1章
<a href="#">嵌入式</a>	第8章
<a href="#">埃马纽埃尔·马蒂</a>	第4章
<a href="#">EsmaeelE</a>	第1章
<a href="#">fdeslaur</a>	第3章
<a href="#">费德里科·庞齐</a>	第9章
<a href="#">菲利普</a>	第14章
<a href="#">Flamewires</a>	第10章
<a href="#">FOP</a>	第12章
<a href="#">foxtrot9</a>	第7章
<a href="#">geek1011</a>	第11章
<a href="#">贾里德</a>	第1章
<a href="#">延斯德</a>	第1章和第4章
<a href="#">克达姆</a>	第1章
<a href="#">基兰·韦穆里</a>	第6章和第17章
<a href="#">库尔迪普·米什拉</a>	第13章
<a href="#">莱奥·莱奥波德·赫兹 준영</a>	第21章
<a href="#">lardenn</a>	第2章和第4章
<a href="#">leeor</a>	第9章
<a href="#">likewhoa</a>	第1章
<a href="#">manav m</a>	第15章
<a href="#">曼努埃尔</a>	第14章
<a href="#">Marsso</a>	第8章
<a href="#">马特乌什·皮奥特罗夫斯基</a>	第1章
<a href="#">mattdm</a>	第2、10和11章
<a href="#">mertyildiran</a>	第5章
<a href="#">迈克·P</a>	第1章
<a href="#">穆罕默德</a>	第1章
<a href="#">内森·奥斯曼</a>	第21章
<a href="#">纳文·查克拉瓦尔蒂</a>	第1章
<a href="#">尼基尔·拉杰</a>	第2章
<a href="#">Not22</a>	第8章

# Credits

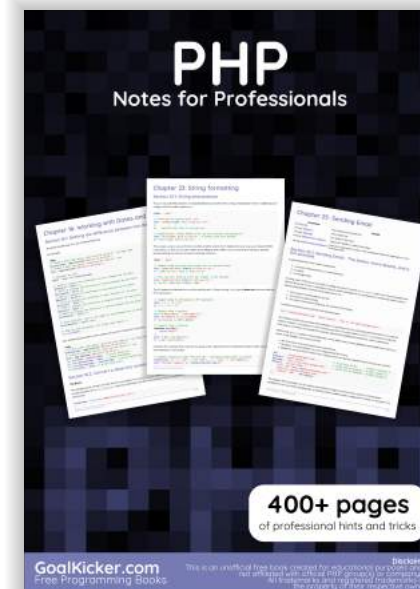
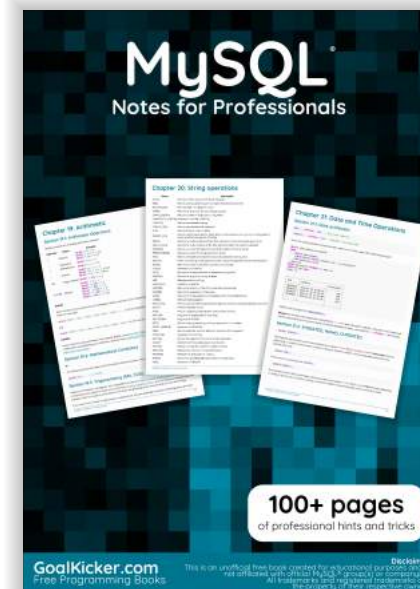
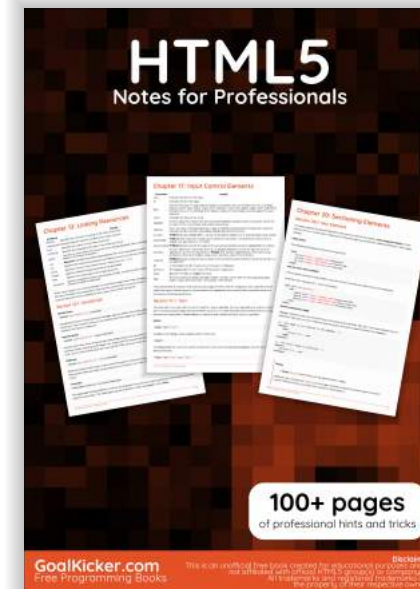
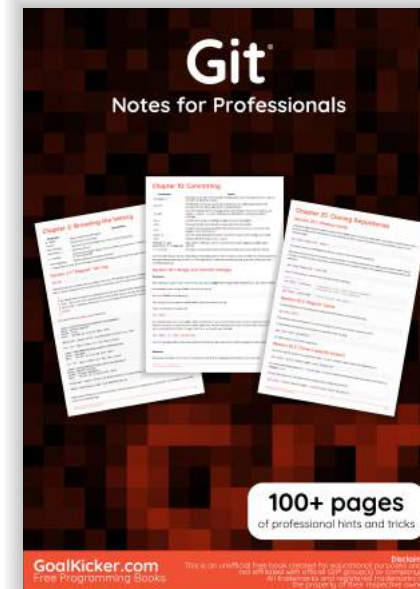
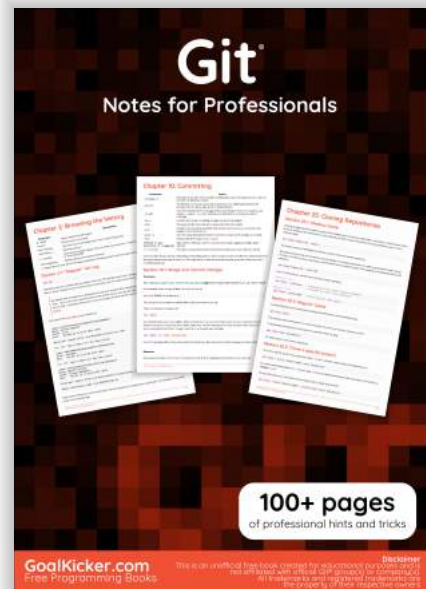
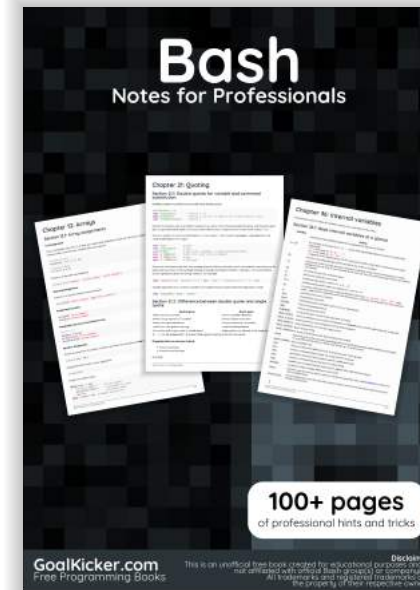
Thank you greatly to all the people from Stack Overflow Documentation who helped provide this content,  
more changes can be sent to [web@petercv.com](mailto:web@petercv.com) for new content to be published or updated

<a href="#">7theo.tk</a>	Chapter 1
<a href="#">Aaron Skomra</a>	Chapter 16
<a href="#">Ajay Sangale</a>	Chapters 1 and 9
<a href="#">Anagh Hegde</a>	Chapters 4 and 14
<a href="#">Ani Menon</a>	Chapters 2 and 4
<a href="#">Arden Shackelford</a>	Chapter 12
<a href="#">Armali</a>	Chapter 1
<a href="#">Baard Kopperud</a>	Chapter 8
<a href="#">BrightOne</a>	Chapters 9, 13, 14 and 19
<a href="#">C.W.Holeman II</a>	Chapter 14
<a href="#">caped114</a>	Chapters 1 and 4
<a href="#">colelemonz</a>	Chapter 1
<a href="#">ctafur</a>	Chapter 17
<a href="#">DaveM</a>	Chapter 2
<a href="#">depperm</a>	Chapter 1
<a href="#">e.dan</a>	Chapter 1
<a href="#">embedded</a>	Chapter 8
<a href="#">Emmanuel Mathi</a>	Chapter 4
<a href="#">EsmaeelE</a>	Chapter 1
<a href="#">fdeslaur</a>	Chapter 3
<a href="#">Federico Ponzi</a>	Chapter 9
<a href="#">Filipe</a>	Chapter 14
<a href="#">Flamewires</a>	Chapter 10
<a href="#">FOP</a>	Chapter 12
<a href="#">foxtrot9</a>	Chapter 7
<a href="#">geek1011</a>	Chapter 11
<a href="#">Jarryd</a>	Chapter 1
<a href="#">Jensd</a>	Chapters 1 and 4
<a href="#">KerDam</a>	Chapter 1
<a href="#">Kiran Vemuri</a>	Chapters 6 and 17
<a href="#">kuldeep mishra</a>	Chapter 13
<a href="#">Léo Léopold Hertz 준영</a>	Chapter 21
<a href="#">lardenn</a>	Chapters 2 and 4
<a href="#">leeor</a>	Chapter 9
<a href="#">likewhoa</a>	Chapter 1
<a href="#">manav m</a>	Chapter 15
<a href="#">Manuel</a>	Chapter 14
<a href="#">Marsso</a>	Chapter 8
<a href="#">Mateusz Piotrowski</a>	Chapter 1
<a href="#">mattdm</a>	Chapters 2, 10 and 11
<a href="#">mertyildiran</a>	Chapter 5
<a href="#">Mike P</a>	Chapter 1
<a href="#">Mohammad</a>	Chapter 1
<a href="#">Nathan Osman</a>	Chapter 21
<a href="#">Naveen Chakravarthy</a>	Chapter 1
<a href="#">Nikhil Raj</a>	Chapter 2
<a href="#">Not22</a>	Chapter 8

<a href="#">oznek</a>	第4章
<a href="#">悖论</a>	第16章和第20章
<a href="#">parkydr</a>	第7章
<a href="#">菲利普·柯克布赖德</a>	第4章、第7章、第12章和第20章
<a href="#">贵格会</a>	第4章
<a href="#">拉杰什·伦加拉吉</a>	第14章
<a href="#">赖利·格林</a>	第15章
<a href="#">鲁比奥</a>	第1章、第4章和第20章
<a href="#">S.罗希特</a>	第3、6和12章
<a href="#">萨瓦</a>	第1、4、14和18章
<a href="#">谢尔盖·斯托利亚罗夫</a>	第2章
<a href="#">苏迪普·班达里</a>	第1章
<a href="#">泰迪</a>	第2章
<a href="#">特朱斯·普拉萨德</a>	第1章
<a href="#">田硕</a>	第1章
<a href="#">托德</a>	第1和14章
<a href="#">vishram0709</a>	第1章
<a href="#">我是谁</a>	第1章
<a href="#">Y4Rv1K</a>	第6章和第17章
<a href="#">玻璃果汁</a>	第1章
<a href="#">zyio</a>	第5章

<a href="#">oznek</a>	Chapter 4
<a href="#">Paradox</a>	Chapters 16 and 20
<a href="#">parkydr</a>	Chapter 7
<a href="#">Philip Kirkbride</a>	Chapters 4, 7, 12 and 20
<a href="#">Quaker</a>	Chapter 4
<a href="#">Rajesh Rengaraj</a>	Chapter 14
<a href="#">Riley Guerin</a>	Chapter 15
<a href="#">Rubio</a>	Chapters 1, 4 and 20
<a href="#">S.Rohit</a>	Chapters 3, 6 and 12
<a href="#">Sava</a>	Chapters 1, 4, 14 and 18
<a href="#">Sergey Stolyarov</a>	Chapter 2
<a href="#">Sudip Bhandari</a>	Chapter 1
<a href="#">Teddy</a>	Chapter 2
<a href="#">Tejus Prasad</a>	Chapter 1
<a href="#">TiansHUo</a>	Chapter 1
<a href="#">Todd</a>	Chapters 1 and 14
<a href="#">vishram0709</a>	Chapter 1
<a href="#">Whoami</a>	Chapter 1
<a href="#">Y4Rv1K</a>	Chapters 6 and 17
<a href="#">ZumodeVidrio</a>	Chapter 1
<a href="#">zyio</a>	Chapter 5

## 你可能也喜欢



## You may also like