

# Debian 系统管理

如有必要，请转到 Vagrantfile 终端中带有 的文件夹，然后输入 来启动 Debian 盒子 `vagrant up`。使用 登录到您的 Debian 盒子 `vagrant ssh`。我们将总体了解 Linux，特别是 Debian。

## 文件系统

Linux（和其他类似 POSIX 的操作系统）使用带有根文件夹的单个文件层次结构 `/`，尽管根文件夹下的不同位置可能安装有不同的文件系统。[文件系统层次结构标准 \(FHS\)](#) 中记录了文件的组织方式。使用命令看看 `ls /`：

`/bin` 代表二进制文件，即可以运行的程序。看看 with `ls /bin`：这里会有很多命令，包括 `ls` 本身。事实上，您可以通过 找到程序的位置 `which`，因此 `which ls` 将向您展示 `/usr/bin/ls` 示例。

`/usr` 是一个历史意外，而且有点混乱。[关于这个 stackexchange 问题的简短历史](#)，但本质上，在最早的日子里，

- `/bin` 仅适用于启动系统所需的二进制文件 - 或者至少是需要多个磁盘驱动器中速度更快的驱动器上运行的最重要的二进制文件，例如您的 shell。
- `/usr/bin` 是大多数二进制文件所在的位置，这些二进制文件在全球范围内可用，例如在组织中的所有机器上。
- `/usr/local/bin` 用于由本地管理员安装的二进制文件，例如组织内的部门。

无论如何，`/usr` 其子文件夹通常用于存放只读数据，例如程序和配置文件，但不包括临时数据或日志文件。它包含类似 `/usr/bin` 或 `/usr/lib` 与根目录中重复的文件夹的子文件夹。Debian 清理这个混乱的方法是使其 `/bin` 只是一个链接 `/usr/bin` 并将所有内容都放在那里，但在某些发行版中，文件夹之间存在真正的差异。

如果您打开了颜色（这是默认设置），您会看到一些文件是绿色的，但其他文件是蓝色的 - 这表明文件类型，绿色是可执行程序，蓝色是到另一个文件的链接。看一下 `ls -l /bin`：每行的第一个字符表示文件类型，主要是 - 普通文件、`d` 目录和 `l` 所谓的软链接。您可以在此列表末尾看到每个链接链接到的位置。例如，`slogin` 链接到 `ssh`。其他链接指向存储在文件系统其他位置的文件——您将看到很多对 `/etc/alternatives/`。

`/etc` 存储系统范围的配置文件，通常只有 `root`（管理员帐户）可以更改此处内容。例如，系统范围的 SSH 配置位于 `/etc/ssh`。

`/lib` 包含动态库 - Windows 调用这些 `.dll` 文件，POSIX 使用 `.so`。例如，`/lib/x86_64-linux-gnu/libc.so.6` C 库，它允许 C 程序使用 `printf`。

`/home` 是包含用户主目录的文件夹，例如默认用户 `vagrant` gets `/home/vagrant`。`root` 是个例外，即管理员帐户，他获得 `/root`。

`/sbin`（系统二进制文件）是另一个程序集合，通常只有系统管理员才会使用。例如，`fdisk` 创建或删除磁盘上的分区，以及许多 `fs` 名称中涉及管理文件系统的程序。`/sbin/halt`，以 `root`（或您允许执行此操作的其他用户）身份运行，关闭系统；还有 `/sbin/reboot`。

`/tmp` 是一个临时文件系统，可以存储在 RAM 中而不是磁盘上（但必要时交换出去），并且不必在重新启动计算机后继续存在。

`/var` 保存随时间变化的文件，例如日志或缓存。

`/dev`、`/sys` 和 `/proc` 是虚拟文件系统。UNIX 的设计原则之一是，几乎每次与操作系统的交互都应该看成是一个程序，就像读写文件一样，或者简而言之，一切都是文件。例如，`/dev` 为硬盘（`/dev/sda` 系统中的第一个 SCSI 磁盘，也是 `/dev/sda1` 其上的第一个分区）、内存（`/dev/mem`）和一些我们稍后将讨论的伪终端或 `tty` 等设备提供接口。`/proc` 提供对正在运行的进程的访问；`/sys` 提供对系统功能的访问。例如，在某些笔记本电脑系统上，写入 `/sys/class/backlight/acpi_video0/brightness` 可更改屏幕亮度。

该 `/vagrant` 文件夹不是 FHS 的一部分，而是我们在 Vagrant 虚拟机上与主机共享文件夹的约定。

## 包管理器

Linux 自从在软盘上发布以来就拥有了包管理器和存储库。存储库是您可以安装的软件的集合，并且可以托管在任何地方 - 软盘、CD-ROM、DVD 或现在的互联网上。包管理器是从存储库安装包的软件 - 到目前为止，这听起来就像一个应用程序商店，但包管理器可以做更多的事情。一方面，如果需要，您可以要求安装特定软件包的不同版本。但包管理器的要点是包可以依赖于其他包，当您安装一个包时，它会自动安装依赖项。

为了说明这一点，我们将介绍一种您需要非常熟悉的软件：在控制台中工作的文本编辑器。文本编辑是许多系统管理任务和编程的基础，人们经常发现熟悉的编辑器成为最喜欢的工具。

Debian 中已经安装了两个基于控制台的文本编辑器：`nano` 和 `vim`。

`nano` 是一个在控制台中工作的基本文本编辑器，并且安装在大多数 Linux 发行版中，包括 `seis` 和实验室计算机上的发行版，因此您可以使用它来远程编辑文件。您可以键入 `nano FILENAME` 来编辑文件。键盘快捷键位于屏幕底部，您需要的主要快捷键是 `Control+X` 退出（如果有未保存的更改，它会询问您是否要保存）。`Nano` 被认为是相当友好的控制台编辑器。

`vim` `vi` 是更旧的 (1976) 编辑器（也已安装）的 1991 改进版本。它是一个学习曲线陡峭的模式编辑器，但其极其强大的编辑语法和广泛的可用性意味着它经常出现在最喜欢的文本编辑器列表的顶部。如果您想开始使用 `vim`，我建议您首先 `vimtutor` 在命令行中键入 - 这将打开 `vim`，其中包含一个文件，指导您完成基本的 `vim` 使用。

另一种基于控制台的编辑器可以追溯到 70 年代中期 `emacs`，`vi` 它是基于 `Lisp` 的传统竞争对手。但是，默认情况下并未安装 `emacs`：在控制台中键入，您将得到 `emacs: command not found`。您可以使用命令安装它

```
sudo apt install emacs-nox
```

- `sudo` (superuser do) 允许您以 `root`（也称为管理员或超级用户）身份运行命令。根据您的系统配置方式，这可能根本不允许（您不能在实验室机器上执行此操作），或者需要密码，但在您使用的 `Vagrant` 盒子上，您可以执行此操作。使用 `sudo` 进行系统管理而不是直接以 `root` 身份登录是一种很好的做法，但是如果您确实需要 `root shell`，那么 `sudo bash` 可以使用一个 `root shell`，`#` 而不是 `$` 提示您警告您正在以 `root` 身份工作。
- `apt` 是 `Debian` 包管理器。
- `install PACKAGE` 添加一个包，这意味着下载并安装它及其所有依赖项（您将看到这些依赖项的列表以确认您想要安装它们 - 您确实这样做）。

我们进行安装是 `emacs-nox` 因为这是打包供控制台使用的 `emacs` 版本。如果你尝试安装，`emacs` 那么包管理器会识别出你需要一个图形显示管理器来运行 `emacs` 的 `GUI` 模式，并安装更多的依赖项来启用它，但我们不需要。

现在 `emacs` 已安装，您可以使用启动它 `emacs`，从那里您应该看到一些介绍性说明，包括如何访问 `emacs` 教程，该教程将教您如何使用它。（如果你想退出 `emacs`，按 `Control-X` 后跟 `Control-C` 即可）。

其他流行的编辑器包括

- `mcedit`，一个文件编辑器，作为“午夜指挥官”包的一部分提供，您可以使用 `sudo apt install mc`。使用启动编辑器并 `mcedit filename` 对其进行测试（`Alt-O` 退出）。
- `tilde`，具有通过 `Alt` 字母序列控制的 `GUI` 式菜单系统的编辑器。安装与 `sudo apt install tilde`。
- `micro`，一个简单的编辑器，有点像的高级版本 `nano`。安装与 `sudo apt install micro`。

您还可以通过以下方式查找有关软件包的信息 `apt info PACKAGE` -- 尝试使用上述方法之一。

我建议您尝试其中一些编辑器，并找出您喜欢如何编辑文件。其中一些工具可能需要投入时间来学习，但在您的计算机科学职业生涯早期这样做可能是一个不错的决定。

无论您最终决定使用哪种编辑器，您可能都不需要安装所有替代品。您可以离开 `nano` 并 `vim` 安装，但对于您在上面尝试过并决定不喜欢的其他编辑器，您可以（并且应该）使用 `sudo apt remove PACKAGE`。

## 更新升级

您正在使用的存储库记录在 中，请使用或 `/etc/apt/sources.list` 查看此文件以查看它们的位置，然后在浏览器中查找站点。有不同 `Debian` 版本的文件夹和不同架构的包索引文件。

`cat nano`

出于安全原因，系统管理员应定期运行两个命令：

- `sudo apt update` 从存储库中获取新的包列表。这样，`apt` 可以告诉您自上次检查以来是否有任何软件包已更新到新版本。
- `sudo apt upgrade` 将您已安装的每个软件包升级到本地软件包列表中的最新版本（在执行操作时下载 `apt update`）。

## 实验室机器

如果您在实验室机器上运行虚拟机，那么在实验室机器重新启动后，您的虚拟机可能不再存在，或者您注销并再次登录并最终在另一台机器上 - 正如您登录时的通知所告诉您的那样，虚拟机存储在 `/tmp`。

每次登录到不同的计算机时都必须重新安装您最喜欢的软件包（例如您选择的文本编辑器），这会很烦人，因此您应该将它们放入 `Vagrantfile` 中，然后系统会自动为您执行此操作 `vagrant up`。`Vagrantfile` 已经包含一行 `echo`，您可以 `apt install PACKAGE` 在其下面添加一行，并且可以根据需要列出任意数量的包。这里没有 `sudo`，因为 `Vagrant` 在安装系统时，会自动以 `root` 身份运行。

- 除非是 `nano` 或 `vim`，请将您最喜欢的文本编辑器的包添加到此行，以便下次您重建 `Vagrant` 机器时，它们会自动添加。
- 注销您的 `vagrant` 机器并执行 `vagrant destroy` 删除虚拟机的操作。然后重新加载，`vagrant up` 将再次下载并配置该框。
- 登录 `vagrant ssh` 并检查编辑器是否已安装。