

Linux内核开发：使用Git管理源代码（3）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_Linux_E5_86_85_E6_A0_c103_144960.htm 您会被提示说要求提供一个提交消息，它是一个用户生成的注释，用来帮助其他开发人员（也可能是您自己以后）理解刚才的实现到底进行了哪些修改。在我们的例子中，提交消息是一个描述刚才对文档所做修改的短句。如果您希望检查一下到目前为止工作的状态，可以执行 `git-log` 来查看本地仓库的历史（它继承了所克隆的仓库的信息）。您的提交消息应该在日志的最上面。添加或删除文件 但是请等一下！我们还没有添加

`Documentation/Sandwiches` 文件呢，因此我们需要将其添加到工作目录中，并告诉 Git 何时这个文件已经准备好了。我使用 `echo` 命令创建了想要添加的文件，因为这只是一个简单的例子而已。同样，您也可以使用自己喜欢的工具。 `$ echo "Turkey is superior" > Documentation/Sandwiches` 现在我们已经添加了一个文件，接下来需要将这个文件添加到 Git 中，从而让 Git 了解这种变化，然后才能提交这个版本。我们可以通过执行下面的命令来完成这些任务：`$ git-add`

`Documentation/Sandwiches $ git-commit`

`Documentation/Sandwiches` 如果您添加了多个文件，可以在同一行中的 `git-add` 命令后面列出这些文件，不过您也不必一次将它们全部添加到仓库中去。如果要删除某个文件，并且没有 `git-add` 之类的特殊命令；您只需要删除这个文件，然后提交就可以了。现在应该查看一下 `git-log`，从而确保到现在为止所做的事情都是正确的。这一次，我们将使用 `-p` 选项来以

单独的补丁格式查看日志。 `$git-log -p` 创建 diff 文件 最后，我们希望生成一个包含您修改后的文件和原文件之间区别的文本文件。这个文件通常是使用 diff 工具创建的，因此就称为 diff 文件。diff 可以帮助我们创建补丁文件（patch file），后者是我们向很多开放源码软件项目发送代码提交时通常使用的方法。有关 diff 的更多内容，请参看下面参考资料部分中有关 Kernel.org 的链接。Git 还可以干什么？我们可以使用 Git 来管理本地仓库，而不用镜像其他人的工作。例如，如果我们喜欢使用 Git 来管理自己个人对某个开放源码项目贡献的文件，就可以从项目快照中生成一个 Git 仓库。假设我们已经有了一个名为 `release.tar.gz` 的标准 release tarball，可以执行下面的命令来创建一个本地的 Git 仓库：`$ tar -zxvf release.tar.gz $ cd release $ git init-db` 我们可以看到消息说 Git 是“默认于本地存储区域的”。这些消息都是正常的，说明我们有一个 Git 仓库。现在我们已经对工作目录进行了初始化，接下来在项目目录中应该会看到一个新目录 `.git`。为了告诉 Git 我们希望对这个项目中的每个文件都进行跟踪，请执行下面的命令：`$ git add .` 最后，使用下面的命令将所监视的文件提交到仓库中：`$ git commit -a` 同样，系统会提示我们输入提交消息。从现在开始，我们就可以在自己的 Git 仓库中使用 Git 所提供的完整功能了，例如对实验特性进行分支，为了追踪回归测试问题而将代码一分为二，并使用常见的版本历史功能。有关 Git 的分支管理和其他有趣特性的更多信息，请参看 Kernel.org 上给出的 Git 的优秀教程（参见参考资料中的链接）。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com