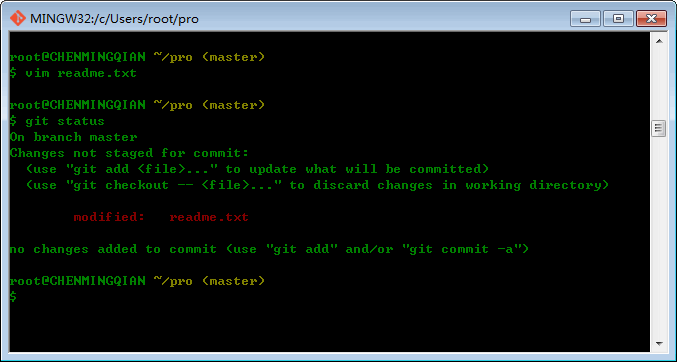
五、本地仓库管理详解

1.查看工作区的状态（git status）

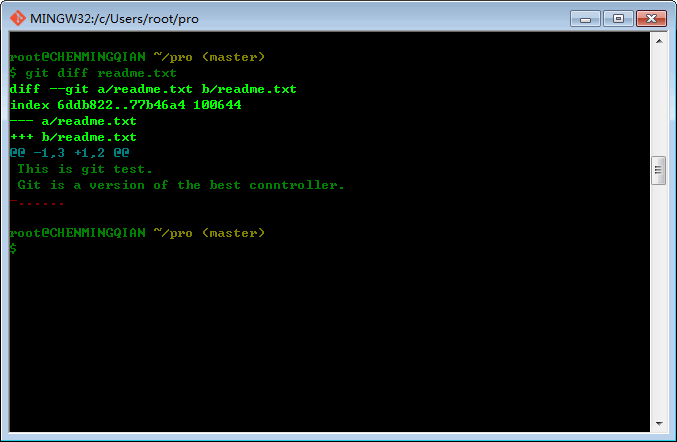
在上面的文章中我们成功的增加了一个readme.txt文件，并用git status命令查看了状态，在这节中我们就详细的说一下git status命令。我们修改一下readme.txt中的内容如下：

|  |  |
| --- | --- |
| 1  2 | This is git test.  Git is a version of the best conntroller. |

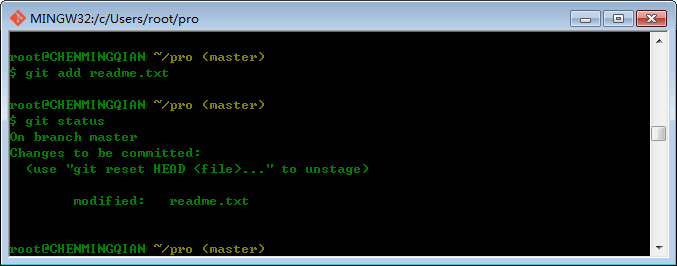
我们删除了最后一行，现在我们再用git status命令查看一下状态：



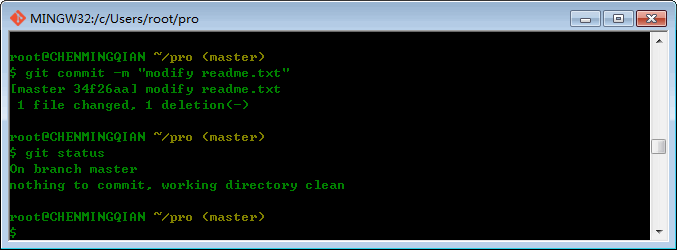
从上图中我们可以看到，我们用git status命令查看一下，我们得知readme.txt文件被修改过了，但是还没有提交这个修改。但是有博友会问了，git提示我们这个 readme.txt这个文件被修改了，但我们想看看具体修改了什么内容怎么办呢？这个时候我们需要git diff命令出场了，如下图：



大家可以从上面图片中可以看到，我们删除了最后一行的……省略号。嘿嘿，用红颜色标出来的。好了知道修改了什么，下面我们就提交修改吧，提交修改和提交新文件一致，也是两步，先add，再commit。如下图：



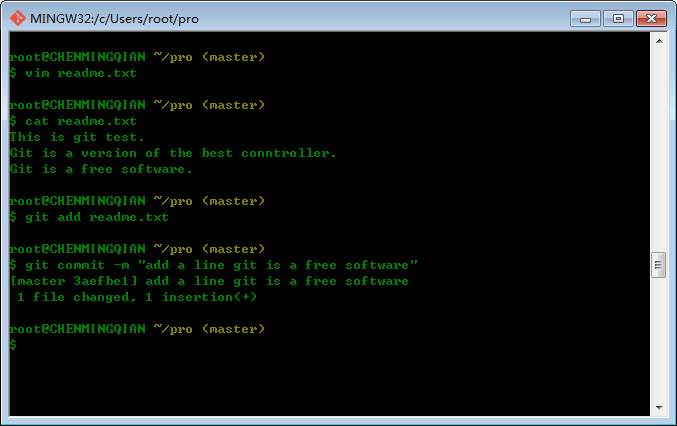
大家可以看到我们执行git add命令，同样没人输出。但是，我们可以用git status命令查看一下。下面我们用git commit命令提交到本地版本库，如下图：

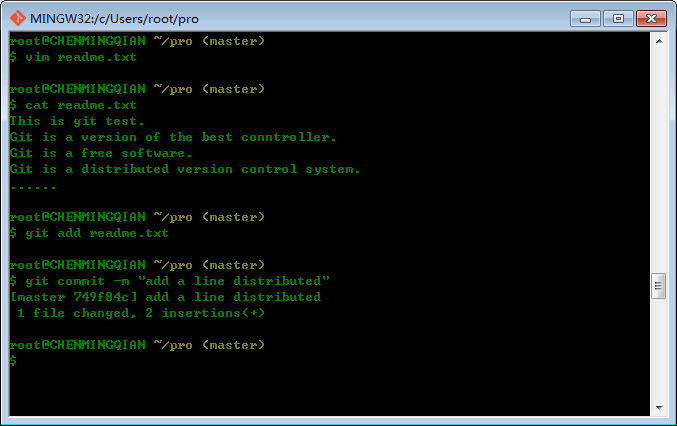


同样的我们使用git status命令查看一下状得知我们修改的readme.txt文件已提交。好了，到这里我们的git status命令就讲解完成了，同时我们也讲解一个git diff命令，可以查看我们修改了哪些内容。下面我们来讲解一下版本回滚。

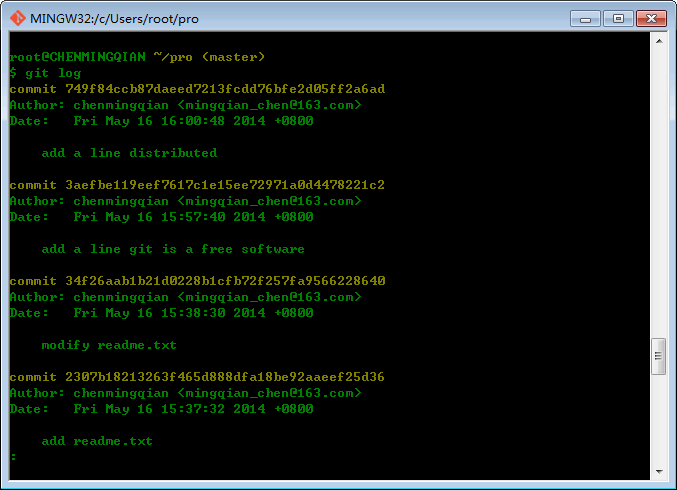
2.版本回滚

在上面一个章节中我们演示修改文件并提交文件，再用git status命令查看状态，下我们再修几个版本并提交：

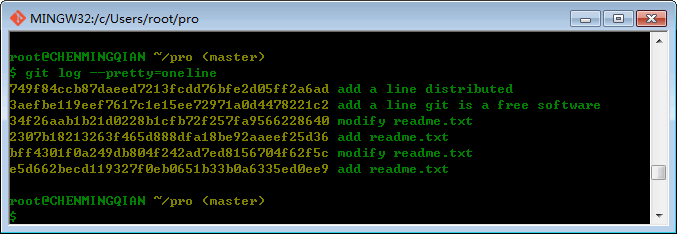




好了，就这样我们不断的修改并提交到本地的版本库中。一会你测试的时候发现前修改的版本有问题想回滚到前面的版本，或者你误删除了一些文件，想回到 上一个commit中去，然后继续工作。我们应该怎么办呢？在Git版本控制中，我们所有的操作都是有历史记录的，我们用git log命令查看一下（如下图）：

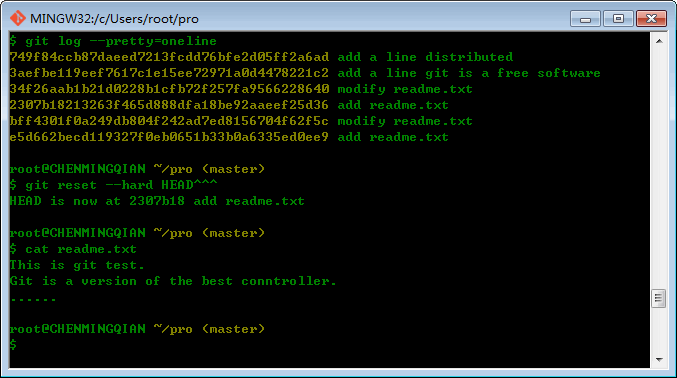


大家可以看到，用git log命令可以查看最近几次提交的版本。如果你发现内容太多，可以用这个命令查看（如下图）：

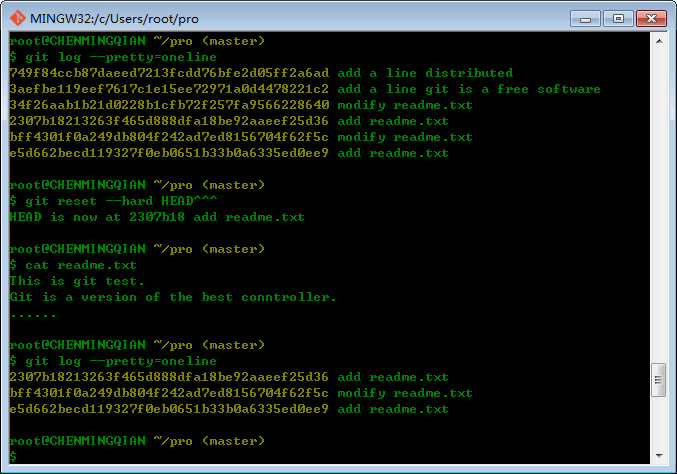


上面你看到的一大串数字，其实是commit id(即，版本号)，和svn不一样，Git的commit id不是1,2,3……这样的递数字，而是一相SHA1计算出来的哈希数，用16进制表示。有博友会问了，为什么Git的commit id不是1,2,3……这样的递增数字，而是用SHA1算出来的哈希数来表示呢？大家应该知道Git是分布式管理系统，我们会有多个人同时开发，如果大家 都用1,2,3……这样的递增数字，作为版本号，那不就冲突了嘛，嘿嘿。好了，下面我们来说版本回滚，准备把readme.txt回滚到“add readme.txt“的那个版本，我们应该怎么做？

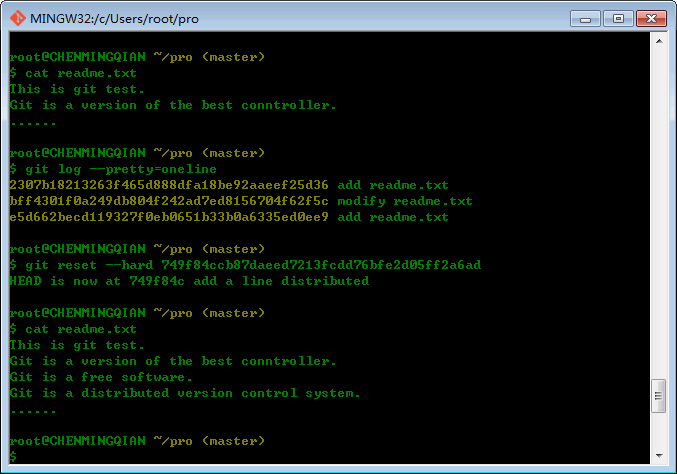
我们先来说一下，Git的当前是在哪个版本，在Git中我们用HEAD表示当前版本，也就是我们最新提交 的”749f84ccb87……ff2q6ad“，上一个版本就应该这么表示”HEAD^“，再上一个版本就这样表示”HEAD^^“，当然如果有几十个 版本或者1000个版本，我们还要写1000个^，这是不可能的，所以哟。我们写成这样HEAD~1000。好了，现在我们就用git reset命令，将我们的版本回退到”add readme.txt“中吧，如下图：



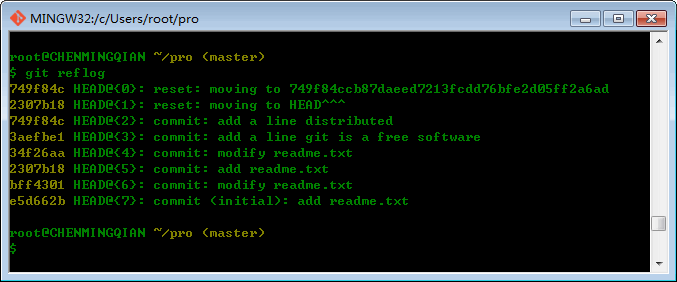
大家可以看到，我们回到了”add readme.txt“版本中了，下面我们再来查看一下git log，如下图：



有个问题哦，有没有发现我们最后提交的那个版本不见了，如上图。现在只有三个版本了，这说明什么问题呢？举个简单的例子说明一下，就好比你从现在回 到了18世界，想再回到21世界已经回不去了，那应该怎么办呢？只要我们找到最新版本的commit id，我这里新版本的commit id是”749f84ccb87……ff2q6ad“，于是就可以回到未来的某个版本，就好比我们知道现在的时间，就能回到现在。下面我们来演示一下（如 下图）：



好了，大家可以看到我们又回到最新版本，嘿嘿。现在你回退到某个版本中，已经关掉了Git bash。但是呢，你又后悔了，想恢复到新版本中，但是呢？你又找不到commit id了，那又该怎么办呢？在Git中，我们当时用$ git reset --hard HEAD^^^，回退到“add readme.txt”版本时，再想恢复到最新版的 “add a line distributed” 中，必须有commit id才行，这时我们已经关闭了Git bash。在Git中提供了一个git reflog命令，是用来记录你每一次执行的命令，下面我们来演示一下（如下图）：



嘿嘿，现在我们又可以回去最新版了。好了，到这里我们的回滚版本就讲解完成了。下面我们来说一下，工作区、提交区/暂存区（stage/index）、版本库。