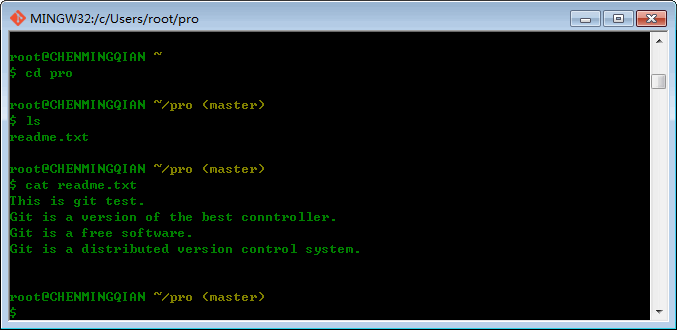
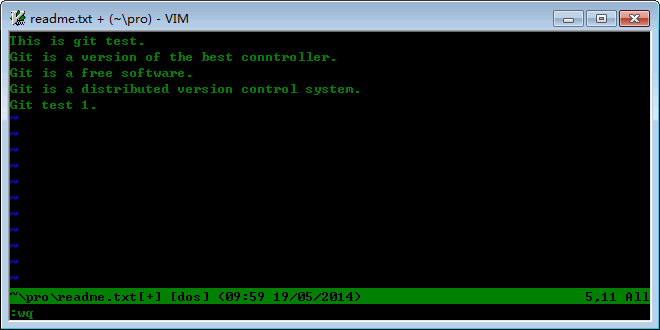
4.修改管理

1).关于修改

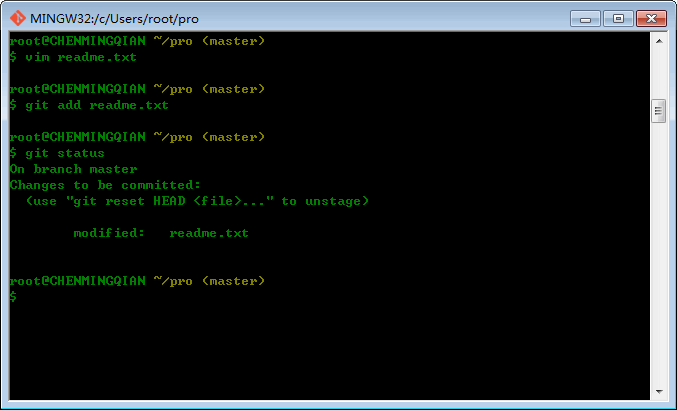
在上面的文章中我们讲解了工作区、暂存区和版本库的概念，有了这些概念有理解。下面我们说一说为什么Git比其它版本控制软件优秀？嘿嘿，是因为 Git跟踪管理的是我们每一次的修改（或操作），而不是文件。比如你新增了一行，这就是一个修改，删除了一行，也是一个修改，更改了某些字符，也是一个修 改，删了一些又加了一些，也是一个修改，甚至创建一个新文件，也算一个修改。下面我们来简单演示一下：我们先查看一下readme.txt文件，如下图：



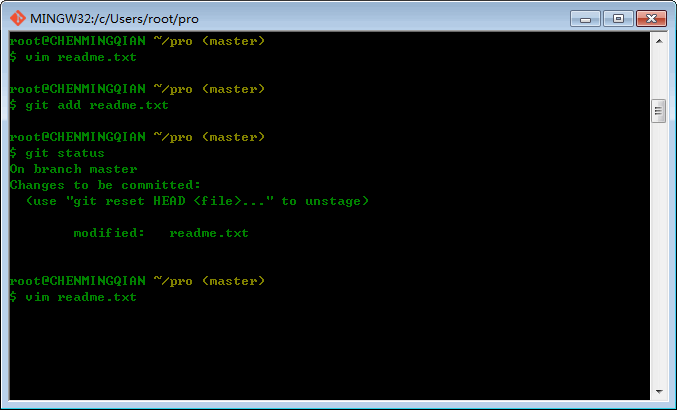
下面我们修改一下readme.txt内容，增加一行：Git rest 1.如下图：



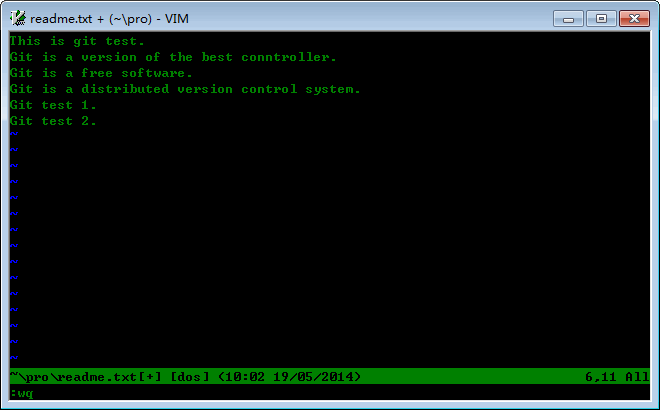
下面我们用 git add 命令提交一下，如下图：

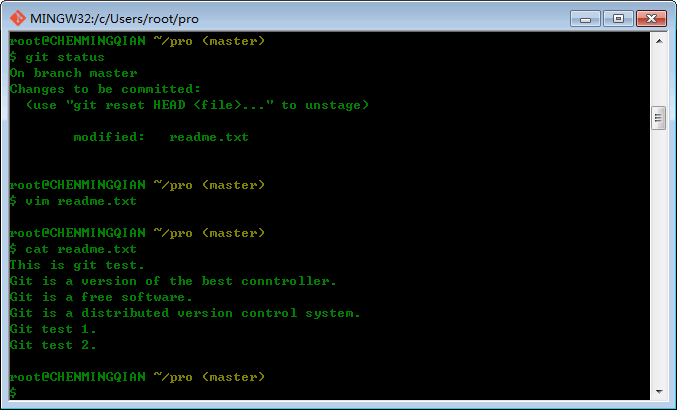


下面我们再修改一下readme.txt文件，如下图：

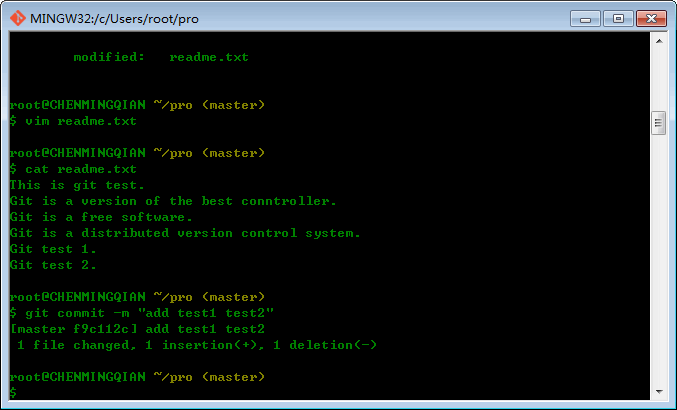


我们又增加一行 Git test 2. 如下图：

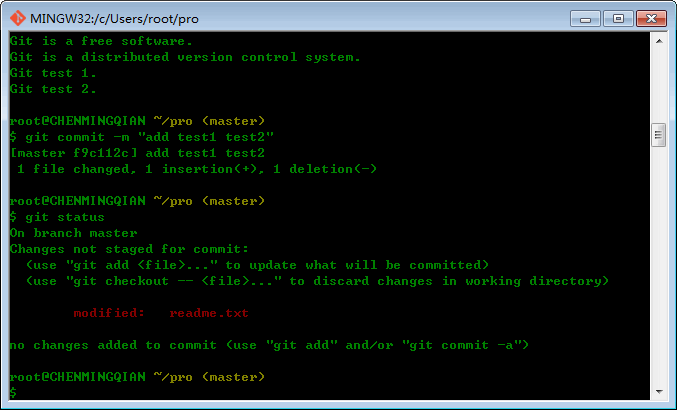


、

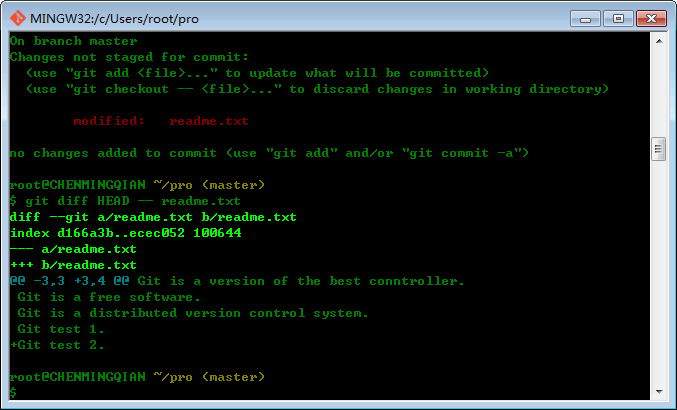
下面我们提交一下，如下图：



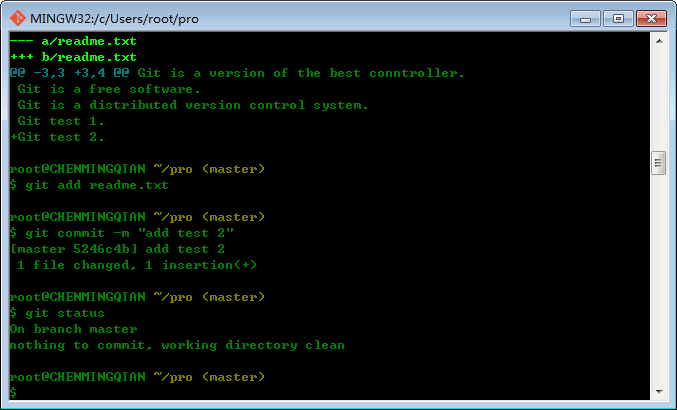
下面我们再查看一下状态，如下图：



大家可以看到我们第二次没有被提交，怎么会这样呢？细心的博友可以看到我们第二次修改后，直接执行 git commit了。没有执行 git add 命令，前面我们说了，当你用“git add”命令后，在工作区的第一次修改被放入暂存区，准备提交。但是，在工作区的第二次修改并没有放入暂存区，所以，“git commit”只负责把暂存区的修改提交了，也就是第一次的修改被提交了，第二次的修改不会被提交。下面我们用 git diff 命令查看一下工作区与版本库里面的区别：



大家可以从图上看到，我们增加 Git test 2. 还没有提交，嘿嘿！那么第二次修改怎么提交呢？其实我们再add再commit，就可以了。如下图：



好了，现在，你又理解了Git是如何跟踪修改的，每次修改，如果不add到暂存区，那就不会加入到commit中。那么我们想取消修改的版本，又该怎么取消呢？下面我们继续……