**Git分支 - 分支切换的几种情况**

这个案例我们要对切换分支命令进行详细说明。

在我们切换分支，执行git checkout master这条命令的时候，

Git做了如下两件事情：

1. 是使 HEAD 指回 master 分支。
2. 将工作目录中文件，恢复成 master 分支所指向提交的快照内容。

也就是说：**分支切换会改变你工作目录中的文件**。

**说明：**在切换分支时，一定要注意你工作目录里的文件会被改变。如果是切换到一个较旧的分支，你的工作目录会恢复到该分支最后一次提交时的样子。如果Git不能干净利落地完成这个任务，它将禁止切换分支。

下面我们就来详细的进行说明。

**1、演示分支切换会改变工作目录中的文件**

**1）查看工作目录中的状态**

|  |
| --- |
| Bash # 1.1 干净的目录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git status On branch master nothing to commit, working tree clean  # 1.2 只有一个分支 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git branch \* master  # 1.3 有三个提交历史 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git log --oneline b97ccfd (HEAD -> master) 第3次提交，新增内容：branch test v3 f72a9fe 第2次提交，新增内容：branch test v2 fa2439a 第1次提交，新增readme.txt文件 |

**2）新建一个testing分支，并提交一个a.txt文件到版本库**

|  |
| --- |
| Bash # 2.1 创建并切换到testing分支 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git checkout -b testing Switched to a new branch 'testing'  # 2.2 创建文件a.txt，并提交到版本库 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ echo "a,txt v1" > a.txt  L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git add a.txt warning: LF will be replaced by CRLF in a.txt. The file will have its original line endings in your working directory  L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git commit -m '新增a.txt文件' [testing 4e9f4d3] 新增a.txt文件  1 file changed, 1 insertion(+)  create mode 100644 a.txt |

**3）查看testing分支的文件**

|  |
| --- |
| Bash # 3.1 查看testing分支提交历史 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git log --oneline 4e9f4d3 (HEAD -> testing) 新增a.txt文件 b97ccfd (master) 第3次提交，新增内容：branch test v3 f72a9fe 第2次提交，新增内容：branch test v2 fa2439a 第1次提交，新增readme.txt文件  # 3.2 查看工作区状态，干净 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git status On branch testing nothing to commit, working tree clean  # 3.3 查看工作目录中的文件，有两个文件 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ ll total 2 -rw-r--r-- 1 L 197121 9 4月 17 12:55 a.txt -rw-r--r-- 1 L 197121 48 4月 17 12:36 readme.txt  # 3.4 查看暂存区中的文件，也有两个文件 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git ls-files a.txt readme.txt  # 版本库中的文件（也就是.git/objects目录中的文件）只会增加不会减少，所以用不看。 # 当然你可以通过命令`git ls-files --with-tree=HEAD`： # 查看本地版本库中，当前分支的文件列表。 |

**4）切换到master分支**

|  |
| --- |
| Bash # 4.1 查看工作目录中的文件状态，是干净的 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git status On branch testing nothing to commit, working tree clean  # 4.2 切换到master分支 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git checkout master Switched to branch 'master'  # 4.3 查看master分支提交历史，改变了 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git log --oneline b97ccfd (HEAD -> master) 第3次提交，新增内容：branch test v3 f72a9fe 第2次提交，新增内容：branch test v2 fa2439a 第1次提交，新增readme.txt文件  # 4.4 查看工作目录中的文件，只有一个文件了 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ ll total 1 -rw-r--r-- 1 L 197121 48 4月 17 12:36 readme.txt  # 4.5 查看暂存区中的文件，也只有一个文件了 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git ls-files readme.txt |

如上就可以证明了，切换分支时，工作目录中的文件会发生改变。

**2、分支切换工作目录不干净的**

如果切换分支的时候，该分支的工作目录不是已提交状态，那么会出现如下两种情况。

**（1）工作区不干净**

也就是有以修改或新增，未被追踪的文件。

**1）切换到testing分支**

|  |
| --- |
| Bash # 1.1 查看当前的工作分支 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git branch \* master  testing  # 1.2 工作目录是干净的 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git status On branch master nothing to commit, working tree clean  # 1.3 切换分支 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git checkout testing Switched to branch 'testing' |

**2）在testing分支新增一个文件b.txt，但不添加到暂存区**

|  |
| --- |
| Bash # 2.1 新增b.txt文件 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ echo 'b.txt v1' > b.txt  # 2.2 查看testing分支上，工作目录中的文件 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ ll total 3 -rw-r--r-- 1 L 197121 10 4月 17 13:58 a.txt -rw-r--r-- 1 L 197121 9 4月 17 14:00 b.txt -rw-r--r-- 1 L 197121 48 4月 17 12:36 readme.txt  # 2.3 查看testing分支上，暂存区中的文件，只有两个文件 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git ls-files a.txt readme.txt  # 2.4 查看testing分支工作目录中的文件状态 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git status On branch testing Untracked files:  (use "git add <file>..." to include in what will be committed)  b.txt  nothing added to commit but untracked files present (use "git add" to track) |

显示b.txt文件未被追踪。

**3）此时切换到master分支上**

|  |
| --- |
| Bash # 3.1 切换到master分支 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git checkout master Switched to branch 'master'  # 3.2 查看master分支工作目录中的文件 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git status On branch master Untracked files:  (use "git add <file>..." to include in what will be committed)  b.txt  nothing added to commit but untracked files present (use "git add" to track)  L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ ll total 3 -rw-r--r-- 1 L 197121 9 4月 17 12:55 a.txt -rw-r--r-- 1 L 197121 9 4月 17 14:00 b.txt -rw-r--r-- 1 L 197121 48 4月 17 12:36 readme.txt  # 3.3 查看master分支，暂存区中的文件 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git ls-files a.txt readme.txt |

我们从上面可以看到，在testing分支上创建的，新增未被追踪的b.txt文件，被同步到了master分支上。

因为git checkout命令执行时，会将工作目录里的文件改变到指定的提交状态。

又因为你在testing分支上创建的b.txt文件，既没有被提交也没有被暂存，说明b.txt文件还未被Git管理。在这种情况下，你进行切换分支操作，你将丢失在testing分支对b.txt文件编写。Git认为这样是非常不安全的操作，会默认的把该状态的文件，一起带入所切换到的分支中。

所以git checkout命令还是很智能的，可以我们一般不会用该功能，这个功能用来会非常的混乱，可能会污染到其他分支，尽管Git是好意。

**（2）工作区和暂存区不干净**

也就是暂存区有新增的文件时，但还未提交到本地版本库。

（接上面示例，继续演示）

**1）切换到testing分支上，把b.txt文件添加到暂存区**

|  |
| --- |
| Bash # 1.1 切换分支 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git checkout testing Switched to branch 'testing'  # 1.2 把b.txt文件添加到暂存区中 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git add b.txt warning: LF will be replaced by CRLF in b.txt. The file will have its original line endings in your working directory  # 1.3 查看工作目录里中的文件，可以看到有三个文件 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ ll total 3 -rw-r--r-- 1 L 197121 10 4月 17 14:21 a.txt -rw-r--r-- 1 L 197121 9 4月 17 14:00 b.txt -rw-r--r-- 1 L 197121 48 4月 17 12:36 readme.txt  # 1.4 查看暂存区中的文件，可以看到有三个文件 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git ls-files a.txt b.txt readme.txt  # 1.5 查看工作目录中文件的状态 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git status On branch testing Changes to be committed:  (use "git restore --staged <file>..." to unstage)  new file: b.txt |

**2）此时切换到master分支上**

|  |
| --- |
| Bash # 2.1 切换分支 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git checkout master Switched to branch 'master' A b.txt # 提示你带有一个增加的文件 b.txt # 发现也是可以进行分支切换的。  # 2.2 查看工作目录中文件的状态 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git status On branch master Changes to be committed:  (use "git restore --staged <file>..." to unstage)  new file: b.txt  # 2.3 查看工作目录里中的文件，可以看到有b.txt文件 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ ll total 2 -rw-r--r-- 1 L 197121 10 4月 17 14:21 a.txt -rw-r--r-- 1 L 197121 9 4月 17 14:00 b.txt -rw-r--r-- 1 L 197121 48 4月 17 12:36 readme.txt  # 2.4 查看暂存区中的文件，可以看到有b.txt文件 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git ls-files a.txt b.txt readme.txt |

发现暂存区中存储的文件，只要未被提交，也会随着分支的切换，加入到其他分支中。

**3、不能切换分支的情况**

（接上面示例，继续演示）

**1）切换到testing分支上，把b.txt文件提交到版本库**

|  |
| --- |
| Bash # 1.1 切换分支 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git checkout testing Switched to branch 'testing' A b.txt  # 1.2 b.txt文件提交到版本库 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git commit -m '新增文件 b.txt' [testing 33f327c] 新增文件 b.txt  1 file changed, 1 insertion(+)  create mode 100644 b.txt   L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git log --oneline 33f327c (HEAD -> testing) 新增文件 b.txt 4e9f4d3 新增a.txt文件 b97ccfd (master) 第3次提交，新增内容：branch test v3 f72a9fe 第2次提交，新增内容：branch test v2 fa2439a 第1次提交，新增readme.txt文件 |

**2）继续修改b.txt文件，不提交到版本库中，然后切换到master分支**

|  |
| --- |
| Bash # 2.1 修改`b.txt`文件 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ echo 'b.txt v2' >> b.txt  # 2.2 查看testing文件工作目录中文件状态 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git status On branch testing Changes not staged for commit:  (use "git add <file>..." to update what will be committed)  (use "git restore <file>..." to discard changes in working directory)  modified: b.txt  no changes added to commit (use "git add" and/or "git commit -a")  # 2.3 切换到master分支 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (testing) $ git checkout master error: Your local changes to the following files would be overwritten by checkout:  b.txt Please commit your changes or stash them before you switch branches. Aborting |

说明：

error: Your local changes to the following files would be overwritten by checkout:b.txt

错误提示：b.txt文件在本地的更改，将随着切换分支的操作，被版本库所覆盖（也就是文件被还原）

Please commit your changes or stash them before you switch branches.

提示你，在切换分支之前，请先提交更改。（不提交不让你切换分支）

|  |
| --- |
| 注意：在暂存区，未提交，也是同理。 |

这是什么原因？

因为b.txt文件已经被Git管理过了，如果b.txt文件还处于修改未追踪状态，你就必须b.txt文件修改完成并提交后，再去其他分支工作。

**3、总结**

使用git checkout命令进行分支的切换，实际上有三个地方更改了

1. HEAD指针进行了移动，指向了新的分支上。
2. 暂存区恢复到新分支上，最新提交版本的状态。
3. 工作目录也恢复到新分支上，最新提交版本的状态。

**而切换分支的最佳实践：每次切换分支前，确保当前分支一定得是干净的（已提交状态）。**

因为要避免一下情况：

在切换分支时，如果当前分支上有新增且未暂存的修改（第一次），或者有新增，暂存，但未提交的修改（第一次）时，分支可以切换成功，但是这种操作可能会污染其地分叉。

作者：繁华似锦Fighting 链接：https://www.jianshu.com/p/b9d0b1e76449 来源：简书 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。