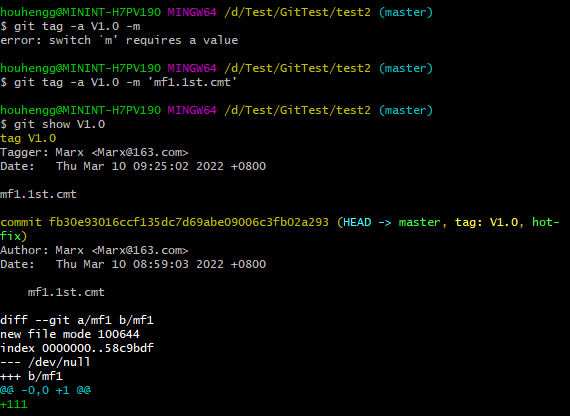
今天复习了之前学习的一些命令，然后学习了几个新的git命令。学习的内容如下

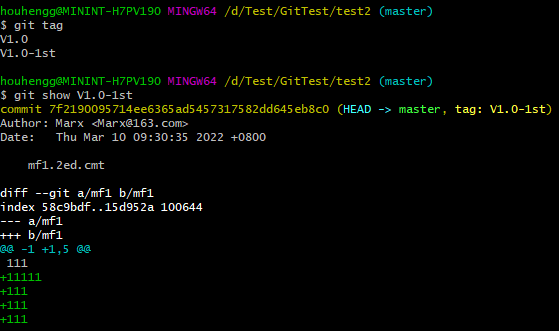
1. Create labels

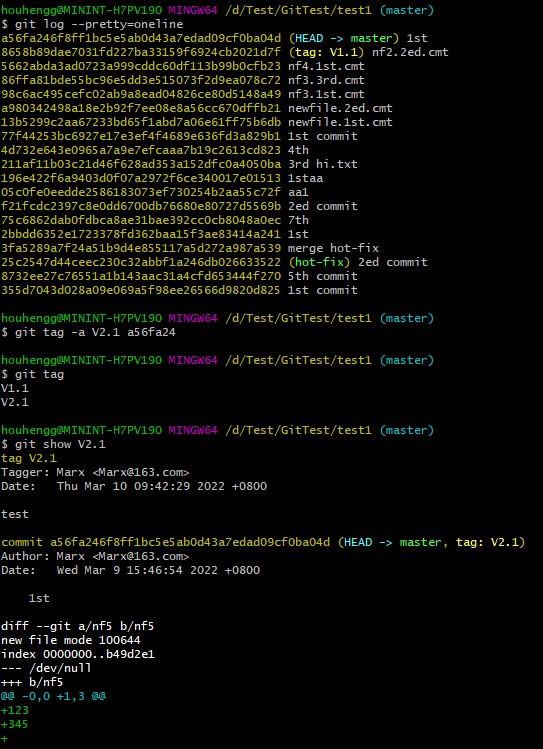
Creating a tag with an annotated type is very simple. Use -a (annotation: take the first letter of annotated) to specify the tag name; the -m option specifies the corresponding tag description, which Git will save together in the tag object



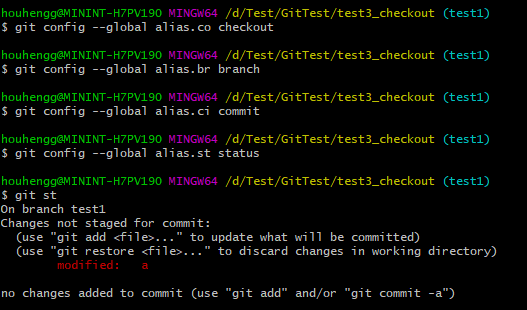
A lightweight tag is actually a file that holds the checksum information of the corresponding commit object. To create such a label, none of the -a, -s or -m options are used, just give the label name.

Now run git show to see this tag information, there is only a summary of the corresponding commit object:



If you forget to add tags in the early stage, you can add tags later. First use the git log command to query the checksum (or the first few characters) of the project that needs to be added, and then run: git tag -a [version] [checksum] 

2. Git Command Aliases: Replace complex commands with simple aliases, but the original commands still work.The following commands use the alias here.



3. git pull

This command is used to fetch the code from the remote and merge the local version.

Command format: git pull <remote host name> <remote branch name>:<local branch name>

To pull the master branch of the remote origin host and merge it with the current local branch, you can omit <local branch name>.

4. git push

This command is used to upload and merge from the local branch version to the remote. The command format is as follows:

git push <remote hostname> <local branch name>:<remote branch name>.

If the local branch name is the same as the remote branch name, you can omit the colon: git push <remote hostname> <local branch name>.

If there is a difference between the local version and the remote version, but you want to force a push, you can use: git push --force origin master.

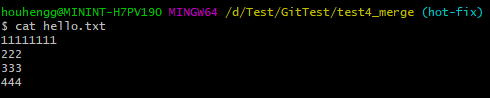
To delete the branch of the host, you can use the --delete parameter, such as deleting the master branch of the origin host: git push origin --delete master.

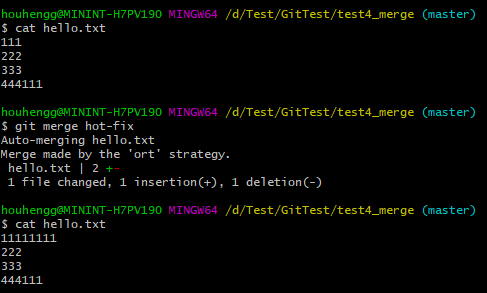
5. git merge

This command is used to merge two branches.

1. If the two branches do not have a fork, a fast merge will be performed, which does not generate a new commitId, just changed the direction of the pointer.

Use the command: git merge <branch\_name> to merge the specified branch into the current branch.





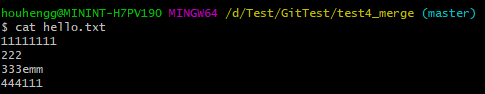
1. Conflict encountered in branch merge

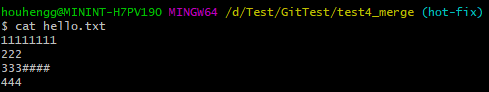
For example, branches dev and master modify and commit a file respectively. When master merges dev, it will conflict.

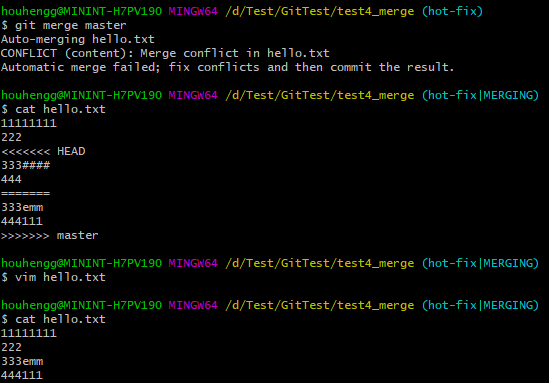
On the branch as the trunk, view the conflicting files and choose how to choose.

The upper part of == represents the content of the current branch, and the lower part of == represents the content of the merged branch.

Remove the conflict mark, and then modify the content appropriately according to the requirements, and then perform a submission to complete the conflict resolution.





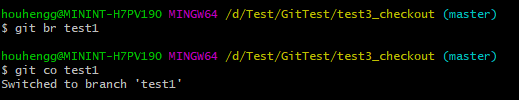




6.git checkout

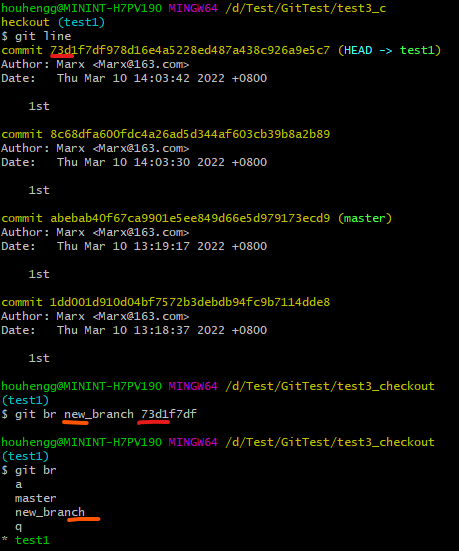
1. git branch <new\_branch>

Create a new branch based on the current branch.



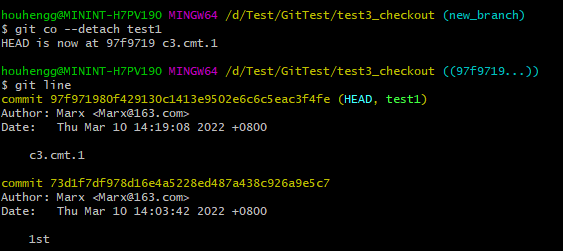
2.git branch <branch> <start point>

Create a branch based on a commit of the current branch.



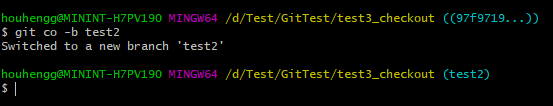
3.git checkout --datch <branch>

Switch to the free state of the branch, the default is the last commit ID under the branch.



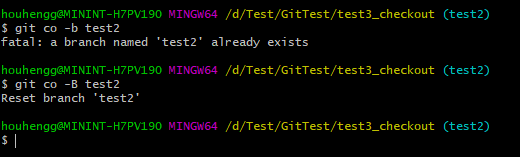
4.git checkout -b <branch\_name>

Create and switch to the specified branch.



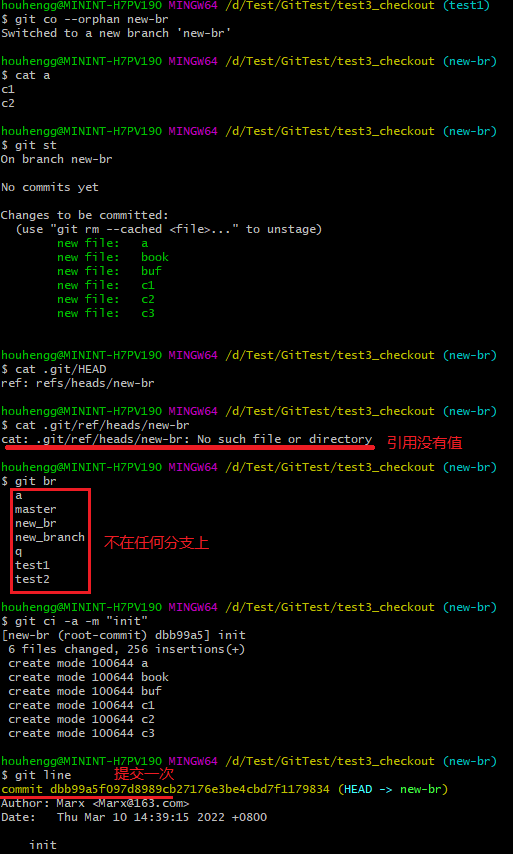
5.git checkout -B <branch>

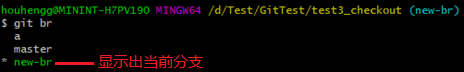
There is already a branch with the same name as your new branch, then using the normal git checkout -b <branch> command will report an error, and the branch with the same name cannot be created. If you use the -B parameter, you can force the creation of a new branch, which will overwrite the original branch.



6.git checkout --orphan <branch>

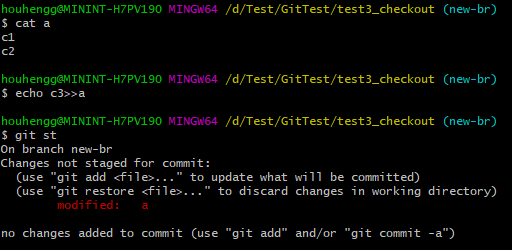
It will create a new bare branch based on the current branch, without any commit history, but the content of the current branch will be there.

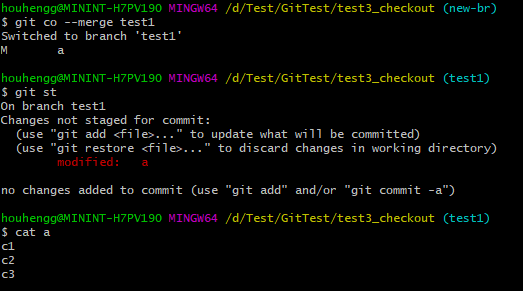




7.git checkout --merge <branch>

This command is applicable to package and take away the modified contents of the current branch when switching branches, and synchronize them to the switched branch.





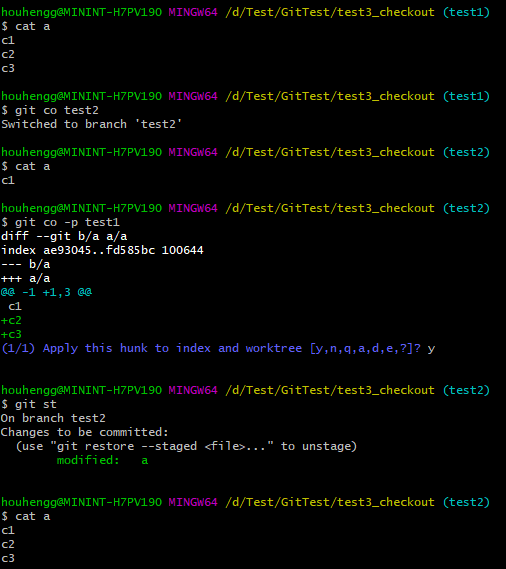
There are two issues to be aware of:

1. If the content between the current branch and the switch branch is different, it is easy to cause conflicts.

2. After switching to a new branch, the modified content of the current branch is lost

8.git checkout -p <branch>

This command is mainly used to compare the differences between two branches and provide an interactive interface to select further operations.



以上是我学习的主要内容，我觉得我还需要多练习和总结。