SWE20001 – Development Project 1: Tools and Practices

Credit Task <2.2>

Duy Phuong Nguyen (101204984)

Lab: Friday 2:30pm

Tutor: Huai Liu

Team number: 3

Henry Tran	102075482
Edward Knight	102700043
Duy Phuong Nguyen	101204984
Christian Ilott	102169762

Merge Conflict

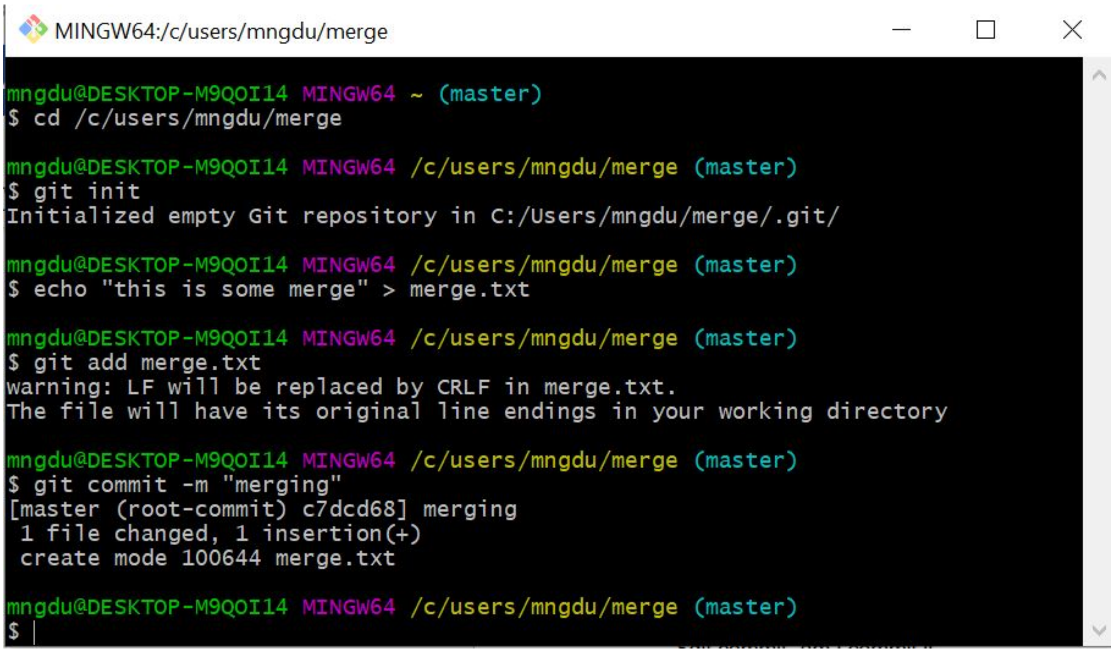
1. What is merge conflict? How can it occur??

Version control system is a great tool for project development because different people can work in the same project. The system will control changes and managing the changes from different contributors. However, conflict will happen when two people are trying to modify the same line in same file

Merge conflict causes fails to start the merge and Git fails during the merge. Git cannot recognise what is wrong what is correct in that situation and it will affect the people who are trying to conduct the merge. Git will mark the conflict and we need to solve it before merging the pull request.

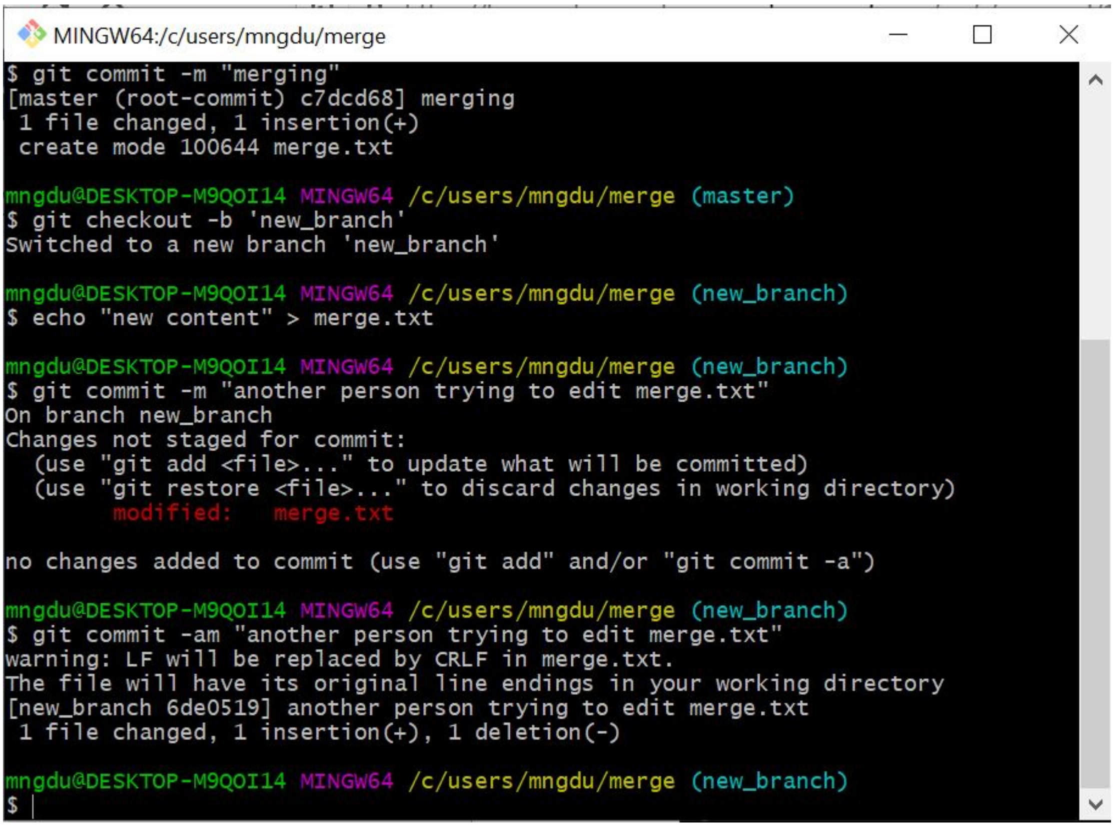
For example/ Demo:

Master create a new file called: merge.txt



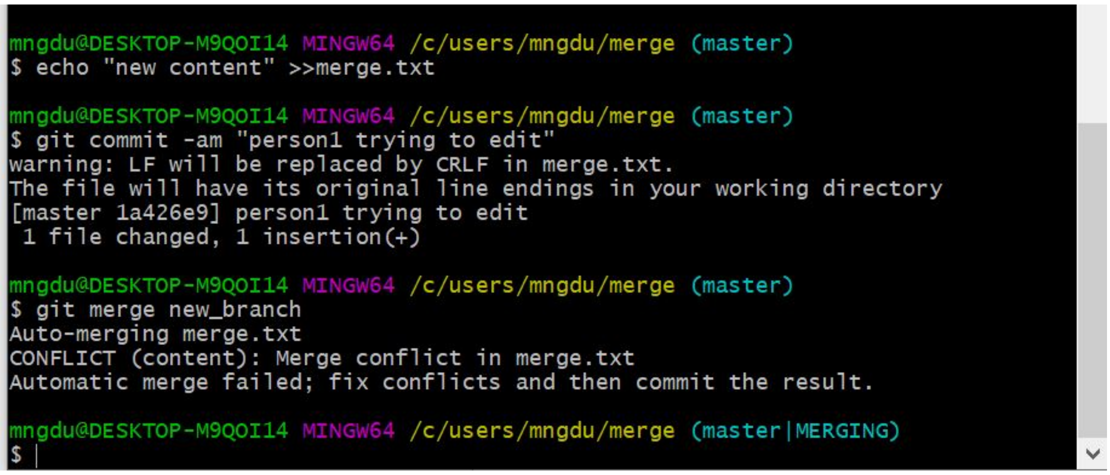
```
mngdu@DESKTOP-M9QOI14 MINGW64 ~ (master)
$ cd /c/users/mngdu/merge

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (master)
$ git init
Initialized empty Git repository in C:/Users/mngdu/merge/.git/

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (master)
$ echo "this is some merge" > merge.txt

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (master)
$ git add merge.txt
warning: LF will be replaced by CRLF in merge.txt.
The file will have its original line endings in your working directory

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (master)
$ git commit -m "merging"
[master (root-commit) c7dcd68] merging
 1 file changed, 1 insertion(+)
 create mode 100644 merge.txt

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (master)
$ |
```

Then another developer trying to edit something in this file



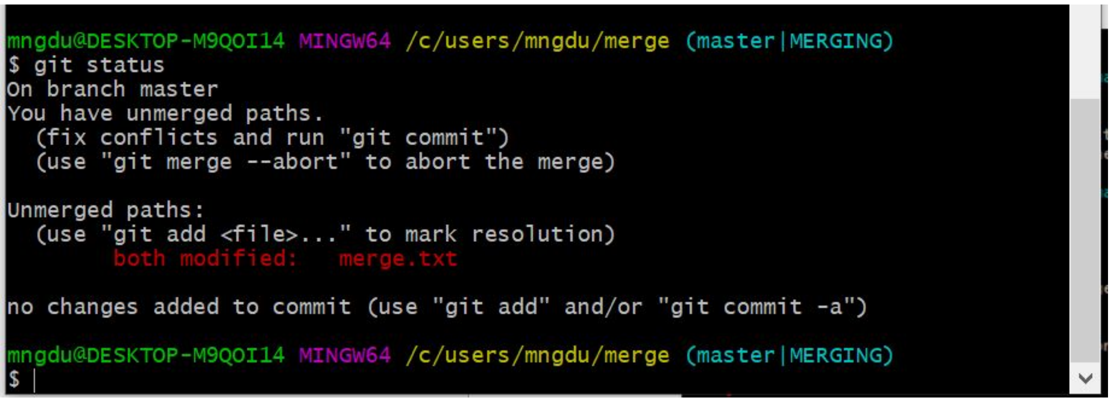
```
$ git commit -m "merging"
[master (root-commit) c7dcd68] merging
 1 file changed, 1 insertion(+)
 create mode 100644 merge.txt

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (master)
$ git checkout -b 'new_branch'
Switched to a new branch 'new_branch'

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (new_branch)
$ echo "new content" > merge.txt

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (new_branch)
$ git commit -m "another person trying to edit merge.txt"
On branch new_branch
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:   merge.txt

no changes added to commit (use "git add" and/or "git commit -a")

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (new_branch)
$ git commit -am "another person trying to edit merge.txt"
warning: LF will be replaced by CRLF in merge.txt.
The file will have its original line endings in your working directory
[new_branch 6de0519] another person trying to edit merge.txt
 1 file changed, 1 insertion(+), 1 deletion(-)

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (new_branch)
$ |
```

We switch back to master and edit the same file. The system will tell you there is a conflict in content.



```
mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (master)
$ echo "new content" >>merge.txt

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (master)
$ git commit -am "person1 trying to edit"
warning: LF will be replaced by CRLF in merge.txt.
The file will have its original line endings in your working directory
[master 1a426e9] person1 trying to edit
 1 file changed, 1 insertion(+)

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (master)
$ git merge new_branch
Auto-merging merge.txt
CONFLICT (content): Merge conflict in merge.txt
Automatic merge failed; fix conflicts and then commit the result.

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (master|MERGING)
$ |
```

When we trying to merge the file, system will pop-up message and can't merge until we solve the problem.

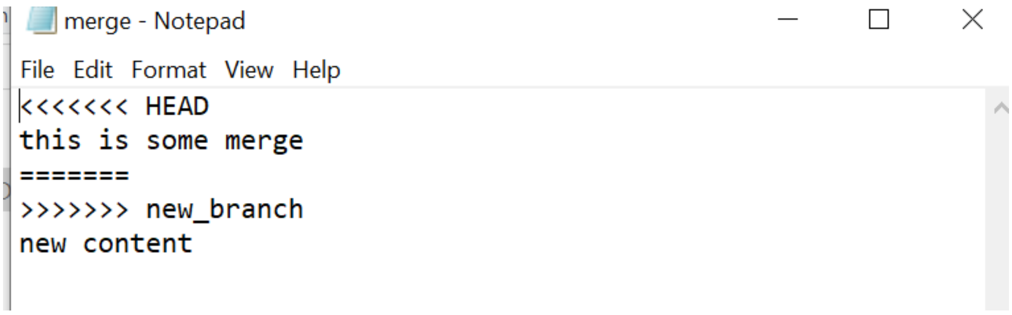


```
mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   merge.txt

no changes added to commit (use "git add" and/or "git commit -a")

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (master|MERGING)
$ |
```

2. How to solve merge conflict

There are two way you can solve the conflict: edit the file or using the command line to delete the one branch with few lines of coding.

a. Editing the file which causes the conflict

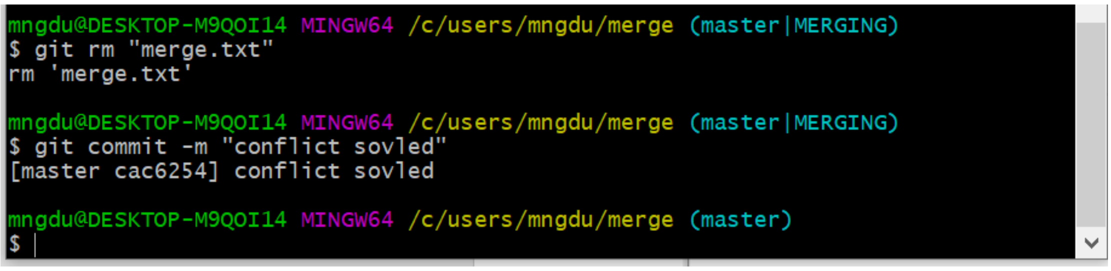


```
<<<<<<< HEAD
this is some merge
=======
>>>>>>> new_branch
new content
```

When you delete new_branch content, Git will allow you to merge

b. Deleting the file that another person is working on

This method maybe causes some problem to another person because we have to decide which one to delete.



```
mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (master|MERGING)
$ git rm "merge.txt"
rm 'merge.txt'

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (master|MERGING)
$ git commit -m "conflict sovled"
[master cac6254] conflict sovled

mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/merge (master)
$ |
```

3. What should not be done and why

We should not let another person working on the same file while they don't know where their part is. It causes conflict and make loss in the file.

Each team member should not work without reminding others because it make the situation becomes complicated when they have to delete a file which others just uploaded