

REPORT NO. 03: MERGE CONFLICTS IN GIT

COURSE CODE: CSE 404
COURSE TITLE: SOFTWARE ENGINEERING AND ISD
LABORATORY

Submitted by

SHANJIDA ALAM (ID: 353)

Submitted to

Dr. Md. MUSFIQUE ANWAR, Professor

Dr. Md. HUMAYUN KABIR, Professor



Computer Science and Engineering
Jahangirnagar University

Dhaka, Bangladesh

September 12, 2024

Contents

| | | |
|----------|----------------------------------|----------|
| 1 | INTRODUCTION | 1 |
| 2 | OBJECTIVE | 1 |
| 3 | WHAT IS MERGE CONFLICT? | 2 |
| 4 | HOW MERGE CONFLICT OCCURS | 2 |
| 5 | CREATING A MERGE CONFLICT | 3 |

1. INTRODUCTION

If we have worked on a team project in Git, we have probably faced a problem called **"merge conflicts"**. It's like an unexpected problem for us when we think everything is going smoothly. If something changes in our code and some else's changes don't adjust together, then the merge conflict situation should arise.

But don't fear! Merge conflicts are a normal part of working with others on code. With the right approach, we can fix them quickly and get back to what matters—coding.

2. OBJECTIVE

The objectives are:

- Learn what merge conflicts are, and how they can create a problem in our code.
- Figure out why merge conflicts occur in code.
- Learn how to identify the problem and how to resolve the merge conflicts.
- Know what not to do.

3. WHAT IS MERGE CONFLICT?

A merge conflict happens when Git can't automatically figure out how to combine changes in the code from different commits. This usually happens when multiple developers are working on the same file, and Git isn't sure which changes to keep. This typically occurs during a merge operation, where changes from different branches are combined.

4. HOW MERGE CONFLICT OCCURS

Root causes of merge conflicts, such as:

- When multiple developers make changes to the same file at the same time.
- When different branches have changes that don't agree with each other.
- Renaming or deleting files.

5. CREATING A MERGE CONFLICT

To show a simple example of how a merge conflict can happen, we can manually trigger a merge conflict from the following set of commands in GIT bash terminal :

- **Step 1:** At first, create a repository in my GitHub account. Here I attach my repository link: <https://github.com/shanjida-alam/Merge-conflicts>
- **Step 2:** Clone the repository in my local machine.

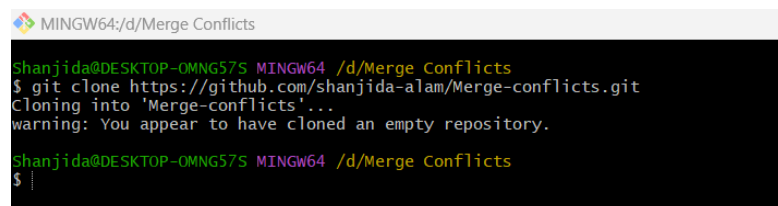
A screenshot of a terminal window with a black background and green text. The window title is 'MINGW64:/d/Merge Conflicts'. The prompt is 'Shanjida@DESKTOP-OMNG57S MINGW64 /d/Merge Conflicts'. The command entered is '\$ git clone https://github.com/shanjida-alam/Merge-conflicts.git'. The output is 'Cloning into 'Merge-conflicts'...' followed by 'warning: You appear to have cloned an empty repository.' The prompt is then '\$ |'.

Figure 5.1: Screenshots of git clone

- **Step 3:** Create a text file and some content add in it and push this file in my repository.
- **Step 4:** Now creating a new branch in my repository.

```

Shanjida@DESKTOP-OMNG57S MINGW64 /d/Merge Conflicts/Merge-conflicts (main)
$ cd First_File.txt
bash: cd: First_File.txt: Not a directory
Shanjida@DESKTOP-OMNG57S MINGW64 /d/Merge Conflicts/Merge-conflicts (main)
$ ls
First_File.txt
Shanjida@DESKTOP-OMNG57S MINGW64 /d/Merge Conflicts/Merge-conflicts (main)
$ git init
Reinitialized existing Git repository in D:/Merge Conflicts/Merge-conflicts/.git/
Shanjida@DESKTOP-OMNG57S MINGW64 /d/Merge Conflicts/Merge-conflicts (main)
$ git add .
Shanjida@DESKTOP-OMNG57S MINGW64 /d/Merge Conflicts/Merge-conflicts (main)
$ git commit -m "added the first text file"
[main (root-commit) b32ed61] added the first text file
1 file changed, 2 insertions(+)
create mode 100644 First_File.txt
Shanjida@DESKTOP-OMNG57S MINGW64 /d/Merge Conflicts/Merge-conflicts (main)
$ git remote add origin https://github.com/shanjida-alam/Merge-conflicts.git
error: remote origin already exists.
Shanjida@DESKTOP-OMNG57S MINGW64 /d/Merge Conflicts/Merge-conflicts (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 596 bytes | 296.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/shanjida-alam/Merge-conflicts.git
 * (new branch) main -> main
branch 'main' set up to track 'origin/main'.

```

Figure 5.2: Screenshots of push the code in my git hub repository

```

Shanjida@DESKTOP-OMNG57S MINGW64 /d/Merge Conflicts/Merge-conflicts (main)
$ git checkout -b new-branch-merge-conflict
Switched to a new branch 'new-branch-merge-conflict'
Shanjida@DESKTOP-OMNG57S MINGW64 /d/Merge Conflicts/Merge-conflicts (new-branch-merge-conflict)
$

```

Figure 5.3: Screenshots of creating new branch

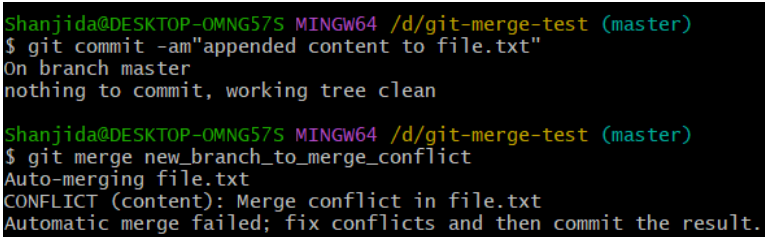
- Step 5:

```

Shanjida@DESKTOP-OMNG57S MINGW64 /d/Merge Conflicts/Merge-conflicts (new-branch-merge-conflict)
$ echo "Changing the context of text file from new branch" > test_file.txt
Shanjida@DESKTOP-OMNG57S MINGW64 /d/Merge Conflicts/Merge-conflicts (new-branch-merge-conflict)
$ git add .
warning: in the working copy of 'test_file.txt', LF will be replaced by CRLF the next time Git touches it
Shanjida@DESKTOP-OMNG57S MINGW64 /d/Merge Conflicts/Merge-conflicts (new-branch-merge-conflict)
$

```

Figure 5.4: Screenshots of add text in the same file in the new creating branch



```
Shanjida@DESKTOP-OMNG57S MINGW64 /d/git-merge-test (master)
$ git commit -am"appended content to file.txt"
On branch master
nothing to commit, working tree clean

Shanjida@DESKTOP-OMNG57S MINGW64 /d/git-merge-test (master)
$ git merge new_branch_to_merge_conflict
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Figure 5.5: Screenshots of trying to merge the branch and creating a conflict