

## Homework 3b

### Exercise 1 Recap Resolve Merge Conflicts

#### Part 1: Repository Setup and Basic Workflow

1. Use the repository of homework assignment 3. Change to its working directory.
2. Create a new branch feature-x, make a change to file.txt and commit.
3. Switch back to main, modify the same part of file.txt, and commit.

```
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git checkout -b feature-x
Switched to a new branch 'feature-x'
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ nano file.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git add file.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit file.txt
[feature-x 00ce6ea] created file.txt
1 file changed, 1 insertion(+)
create mode 100644 file.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git checkout main
Switched to branch 'main'
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ nano file.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git add file.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit file.txt
[main 1b62f2b] added file.txt
1 file changed, 1 insertion(+)
create mode 100644 file.txt
```

#### Part 2: Merging with Conflicts

1. Merge feature-x into main.
2. Git will produce a merge conflict. Show how you resolved the conflict.
3. Commit the merge resolution.
4. Push your changes to the remote repository

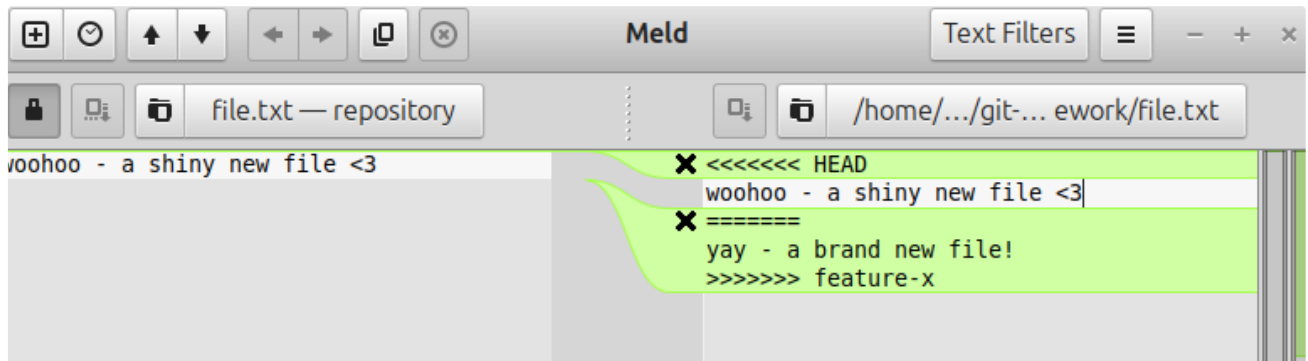
```
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git merge feature-x
Auto-merging file.txt
CONFLICT (add/add): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit the result.
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ meld file.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit
U file.txt
error: Committing is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git add file.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit
[main fc56a8b] Merge branch 'feature-x' all conflicts resolved
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ cat file.txt
yay - a brand new, shiny new file <3
woohoo!!!
```

```

ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ cat file.txt
yay - a brand new, shiny new file <3
woohoo!!!
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ nano file.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git checkout -p file.txt
diff --git a/file.txt b/file.txt
index 0094588..a523a8b 100644
--- a/file.txt
+++ b/file.txt
@@ -1,2 +1,4 @@
 yay - a brand new, shiny new file <3
 woohoo!!!
+
+this is not a good idea :-/
(1/1) Discard this hunk from worktree [y,n,q,a,d,e,?]? y

ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ cat file.txt
yay - a brand new, shiny new file <3
woohoo!!!
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$

```



```

ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git merge feature-x
Auto-merging file.txt
CONFLICT (add/add): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit the result.
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ meld file.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit
U      file.txt
error: Committing is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git add file.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit
[main fc56a8b] Merge branch 'feature-x' all conflicts resolved
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ cat file.txt
yay - a brand new, shiny new file <3
woohoo!!!
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$

```

## Exercise 2 Recovery Scenarios

Scenario A: git checkout

5. Make a wrong change to a file.txt on main and commit it.
6. Use git checkout to restore the file to its last correct state.
7. Show the command you used and the final file content.

```

ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ nano file.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git checkout -p file.txt
diff --git a/file.txt b/file.txt
index 0094588..a523a8b 100644
--- a/file.txt
+++ b/file.txt
@@ -1,2 +1,4 @@
 yay - a brand new, shiny new file <3
 woohoo!!!

```

```
+
+this is not a good idea :-/
(1/1) Discard this hunk from worktree [y,n,q,a,d,e,?]? y
```

```
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ cat file.txt
yay - a brand new, shiny new file <3
woohoo!!!
```

```
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit
On branch main
Untracked files:
(use "git add <file>..." to include in what will be committed)
3b0.png
3b1.png
3b2.png
img/
notes_BACKUP_12932.txt
notes_BASE_12932.txt
notes_LOCAL_12932.txt
notes_REMOTE_12932.txt
```

```
nothing added to commit but untracked files present (use "git add" to track)
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git add file.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit
On branch main
Untracked files:
(use "git add <file>..." to include in what will be committed)
3b0.png
3b1.png
3b2.png
img/
notes_BACKUP_12932.txt
notes_BASE_12932.txt
notes_LOCAL_12932.txt
notes_REMOTE_12932.txt
```

```
nothing added to commit but untracked files present (use "git add" to track)
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit file.txt
On branch main
Untracked files:
(use "git add <file>..." to include in what will be committed)
3b0.png
3b1.png
3b2.png
img/
notes_BACKUP_12932.txt
notes_BASE_12932.txt
notes_LOCAL_12932.txt
notes_REMOTE_12932.txt
```

```
nothing added to commit but untracked files present (use "git add" to track)
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git status
On branch main
Untracked files:
(use "git add <file>..." to include in what will be committed)
3b0.png
3b1.png
3b2.png
img/
```

```
notes_BACKUP_12932.txt
notes_BASE_12932.txt
notes_LOCAL_12932.txt
notes_REMOTE_12932.txt
```

nothing added to commit but untracked files present (use "git add" to track)

```
>ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT$ git clone https://github.com/AlexandraIllner/IKT_homework-
assignments.git
Cloning into 'IKT_homework-assignments'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
```

## Scenario B: git reset

1. Add and commit a change to file2.txt.
2. Use git reset --soft HEAD~1 to undo the last commit but keep changes.
3. Use git reset --hard HEAD~1 to completely remove the commit and changes.
4. Document your terminal commands and file status before and

```
>ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$ nano file2.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$ git add file2.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$ git reset --soft HEAD~1
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$ cat file2.txt
git reset [<mode>] [<commit>]

This form resets the current branch head to <commit> and possibly updates the index (resetting it to
the tree of <commit>) and the working tree depending on <mode>. Before the operation, ORIG_HEAD is set
to the tip of the current branch. If <mode> is omitted, defaults to --mixed. The <mode> must be one of
the following:
```

### --soft

Does not touch the index file or the working tree at all (but resets the head to <commit>, just like all modes do). This leaves all your changed files "Changes to be committed", as git status would put it.

### --mixed

Resets the index but not the working tree (i.e., the changed files are preserved but not marked for commit) and reports what has not been updated. This is the default action.

If -N is specified, removed paths are marked as intent-to-add (see git-add[1]).

### --hard

Resets the index and working tree. Any changes to tracked files in the working tree since <commit> are discarded. Any untracked files or directories in the way of writing any tracked files are simply deleted.

### --merge

Resets the index and updates the files in the working tree that are different between <commit> and HEAD, but keeps those which are different between the index and working tree (i.e. which have changes which have not been added). If a file that is different between <commit> and the index has unstaged changes, reset is aborted.

In other words, --merge does something like a git read-tree -u -m <commit>, but carries forward unmerged index entries.

--keep

Resets index entries and updates files in the working tree that are different between <commit> and HEAD.  
If a file that is different between <commit> and HEAD has local changes, reset is aborted.

--[no-]recurse-submodules

When the working tree is updated, using --recurse-submodules will also recursively reset the working tree of all active submodules according to the commit recorded in the superproject, also setting the submodules' HEAD to be detached at that commit.

```
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$ nano file2.txt
```

```
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$ git reset --hard HEAD~1
```

HEAD is now at 9b9f7aa Initial commit

```
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$ cat file2.txt
```

cat: file2.txt: No such file or directory

```
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$ git status
```

On branch main

Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.

(use "git pull" to update your local branch)

Untracked files:

(use "git add <file>..." to include in what will be committed)

./

nothing added to commit but untracked files present (use "git add" to track)

## Scenario C: git revert

1. Add a "bad" commit to main (e.g., delete a line by mistake).
2. Use git revert to undo it with a new commit.
3. Show git log before and after the revert.

```
>ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$ cat notes.txt
```

Profil des Open Source-Projekts Git

## Überblick

~~~~~

Git ist ein freies und quelloffenes, verteiltes Versionskontrollsystem. Es wurde entwickelt, um alles von kleinen bis zu sehr großen Projekten mit hoher Geschwindigkeit und Effizienz zu verwalten. Git ist heute das am weitesten verbreitete Versionskontrollsystem und bildet das Rückgrat vieler Open Source-Projekte und der modernen Softwareentwicklung.

## Hauptmerkmale

~~~~~

Verteiltes System: Jeder Entwickler besitzt eine vollständige Kopie des Repositories inklusive Historie.

Effizient und schnell: Git ist für hohe Performance optimiert, sowohl bei kleinen als auch bei großen Projekten.

Branching und Merging: Lokale Branches sind einfach, schnell und günstig zu erstellen. Das Zusammenführen (Merging) von Branches ist ein zentrales Feature.

Staging Area: Änderungen können gezielt für den nächsten Commit vorbereitet werden.

Unterstützung für verschiedene Workflows: Git eignet sich für Einzelentwickler, kleine Teams und große Open Source-Projekte gleichermaßen

## Geschichte

Entstehung: Git wurde 2005 von Linus Torvalds entwickelt, um die Entwicklung des Linux-Kernels zu unterstützen, nachdem das vorherige System (BitKeeper) nicht mehr genutzt werden konnte.

Eigenschaft	Beschreibung
-------------	--------------

Programmiersprache	C, Perl, Shell
--------------------	----------------

Plattformen	Windows, macOS, Linux, Unix
-------------	-----------------------------

Lizenz	GNU General Public License v2 (GPLv2)
--------	---------------------------------------

Repository	https://git-scm.com
------------	---------------------

Erste Veröffentlichung	589
------------------------	-----

## Geschichte

Entstehung: Git wurde 2005 von Linus Torvalds entwickelt, um die Entwicklung des Linux-Kernels zu unterstützen, nachdem das vorherige System (BitKeeper) nicht mehr genutzt werden konnte.

Eigenschaft	Beschreibung
-------------	--------------

Programmiersprache	C, Perl, Shell
--------------------	----------------

Plattformen	Windows, macOS, Linux, Unix
-------------	-----------------------------

Lizenz	GNU General Public License v2 (GPLv2)
--------	---------------------------------------

Repository	https://git-scm.com
------------	---------------------

Erste Veröffentlichung	589
------------------------	-----

```
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$ git revert --skip
```

```
error: no revert in progress
```

```
fatal: revert failed
```

```
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$ git revert elfaa8a76ed439bca90e6c6bac98ed914fc741a5
```

```
[main 995c3e3] Revert "edited notes.txt"
```

```
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
>ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$ git revert
```

```
elfaa8a76ed439bca90e6c6bac98ed914fc741a5
```

```
[main 995c3e3] Revert "edited notes.txt"
```

```
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$ cat notes.txt
```

Profil des Open Source-Projekts Git

## Überblick

Git ist ein freies und quelloffenes, verteiltes Versionskontrollsystem. Es wurde entwickelt, um alles von kleinen bis zu sehr großen

Projekten mit hoher Geschwindigkeit und Effizienz zu verwalten. Git ist heute das am weitesten verbreitete Versionskontrollsystem und bildet das Rückgrat

vieler Open Source-Projekte und der modernen Softwareentwicklung.

## Hauptmerkmale

Verteiltes System: Jeder Entwickler besitzt eine vollständige Kopie des Repositories inklusive Historie.

Effizient und schnell: Git ist für hohe Performance optimiert, sowohl bei kleinen als auch bei großen Projekten.

Branching und Merging: Lokale Branches sind einfach, schnell und günstig zu erstellen. Das Zusammenführen (Merging) von Branches ist ein zentrales Feature.

Staging Area: Änderungen können gezielt für den nächsten Commit vorbereitet werden.

Unterstützung für verschiedene Workflows: Git eignet sich für Einzelentwickler, kleine Teams und große Open Source-Projekte gleichermaßen

## Geschichte

Entstehung: Git wurde 2005 von Linus Torvalds entwickelt, um die Entwicklung des Linux-Kernels zu unterstützen, nachdem das vorherige System (BitKeeper) nicht mehr genutzt werden konnte.

Eigenschaft	Beschreibung
-------------	--------------

Programmiersprache	C, Perl, Shell
--------------------	----------------

Plattformen	Windows, macOS, Linux, Unix
-------------	-----------------------------

Lizenz	GNU General Public License v2 (GPLv2)
--------	---------------------------------------

Repository	https://git-scm.com
------------	---------------------

Erste Veröffentlichung	2005
------------------------	------

```
-----
Eigenschaft |Beschreibung|
-----
Programmiersprache |C, Perl, Shell
Plattformen |Windows, macOS, Linux, Unix
Lizenz |GNU General Public License v2 (GPLv2)
Repository |https://git-scm.com
Erste Veröffentlichung |2005
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$ git log
commit 995c3e37b70743d49955e8a7749f6c230890f583 (HEAD -> main)
Author: Alexandra Weber <s0574421@htw-berlin.de>
Date: Mon Jul 7 05:02:41 2025 +0200

    Revert "edited notes.txt"

    This reverts commit elfaa8a76ed439bca90e6c6bac98ed914fc741a5.

commit elfaa8a76ed439bca90e6c6bac98ed914fc741a5
Author: Alexandra Weber <s0574421@htw-berlin.de>
Date: Mon Jul 7 04:58:04 2025 +0200

    edited notes.txt

commit f82a1475647d6cca38715cb4af4453e5306e47d2 (origin/main, origin/HEAD)
Author: Alexandra Weber <s0574421@htw-berlin.de>
Date: Mon Jul 7 04:26:24 2025 +0200

    initial commit assignments 3 & 3b

commit 9b9f7aa8b6f70cf164566925c0a2f0197e068eb1
Author: AlexandraWeber <82087641+AlexandraIllner@users.noreply.github.com>
Date: Sun May 25 07:46:56 2025 -0700

    Initial commit
```

```
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$ git log
commit 995c3e37b70743d49955e8a7749f6c230890f583 (HEAD -> main)
Author: Alexandra Weber <s0574421@htw-berlin.de>
Date: Mon Jul 7 05:02:41 2025 +0200
```

```
`Revert "edited notes.txt"`
```

```
`This reverts commit elfaa8a76ed439bca90e6c6bac98ed914fc741a5.`
```

```
commit elfaa8a76ed439bca90e6c6bac98ed914fc741a5
Author: Alexandra Weber <s0574421@htw-berlin.de>
Date: Mon Jul 7 04:58:04 2025 +0200
```

```
`edited notes.txt`
```

```
commit f82a1475647d6cca38715cb4af4453e5306e47d2 (origin/main, origin/HEAD)  
Author: Alexandra Weber <s0574421@htw-berlin.de>  
Date: Mon Jul 7 04:26:24 2025 +0200
```

```
`initial commit assignments 3 & 3b`
```

```
commit 9b9f7aa8b6f70cf164566925c0a2f0197e068eb1  
Author: AlexandraWeber <82087641+AlexandraIllner@users.noreply.github.com>  
Date: Sun May 25 07:46:56 2025 -0700
```

```
`Initial commit`
```

```
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/IKT_homework-assignments/3_3b$
```