SENJUTI GHOSAL

RA2111030010096

COGNIZANT WEEK1 TASK

Exercise 1: Introduction to Version Control

Objective:

Initialize a new Git repository and commit your first file.

Instructions:

1. Create a new directory for your project.

```
C:\Users\SENJUTI>mkdir my_project
```

2. Navigate into the directory.

```
C:\Users\SENJUTI>cd my_project
```

3. Initialize a new Git repository in the directory.

```
C:\Users\SENJUTI\my_project>git init
Initialized empty Git repository in C:/Users/SENJUTI/my_project/.git/
```

4. Create a new file named file1.txt and add some content to it.

```
C:\Users\SENJUTI\my_project>echo "This is my first file." > file1.txt
```

5. Add the file to the staging area.

```
C:\Users\SENJUTI\my_project>git add file1.txt
```

6. Commit the file with a commit message.

```
C:\Users\SENJUTI\my_project>git commit -m "Initial commit with file1.txt"
[master (root-commit) fbe4219] Initial commit with file1.txt
 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt
```

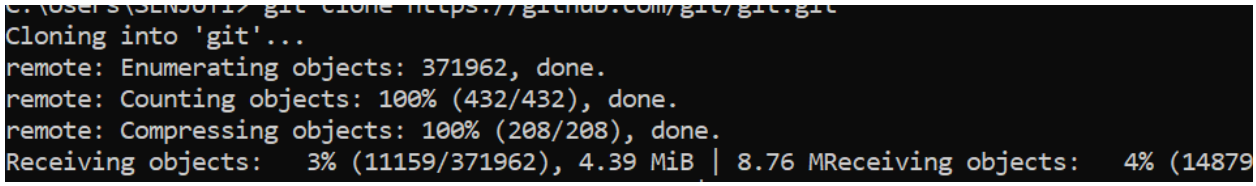
Exercise 2: Understanding Git

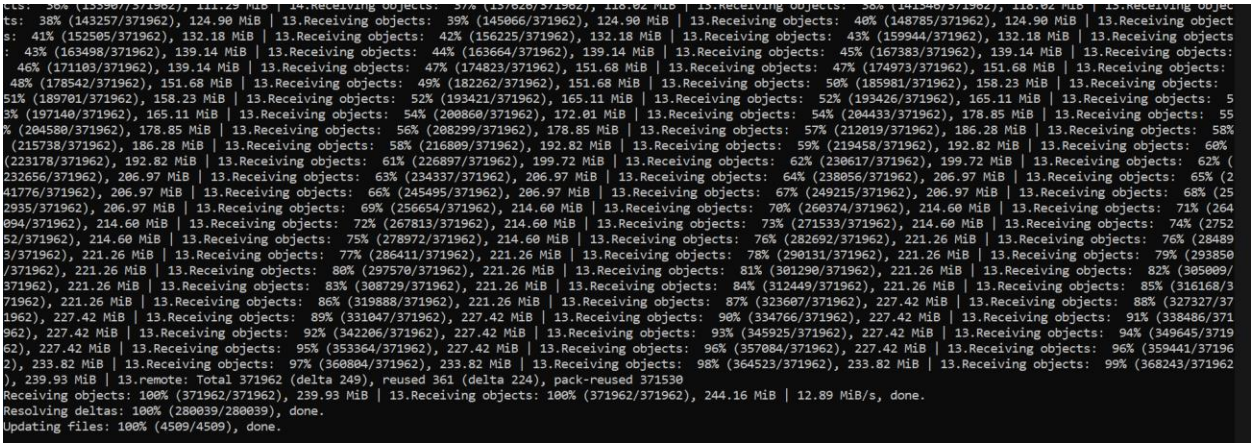
Objective:

Clone an existing repository and explore its history.

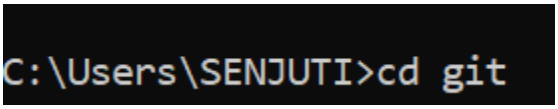
Instructions:

1. Clone a public repository from a platform like GitHub.





2. Navigate into the cloned repository.



3. Check the commit history.

```
C:\Users\SENJUTI\git>git log
commit 39bf06adf96da25b87c9aa7d35a32ef3683eb4a4 (HEAD -> master, tag: v2.46.0, origi
n/master, origin/HEAD)
Author: Junio C Hamano <gitster@pobox.com>
Date:   Mon Jul 29 07:14:09 2024 -0700

    Git 2.46

    Signed-off-by: Junio C Hamano <gitster@pobox.com>

commit 2ab3396b4e1357cc7f9d9d21888c28a1768f3a5d
Merge: ad57f148c6 de86879ace
Author: Junio C Hamano <gitster@pobox.com>
Date:   Mon Jul 29 07:11:16 2024 -0700

    Merge tag 'l10n-2.46.0-rnd2' of https://github.com/git-l10n/git-po

    l10n-2.46.0-rnd2

    * tag 'l10n-2.46.0-rnd2' of https://github.com/git-l10n/git-po:
      l10n: zh_CN: updated translation for 2.46
      l10n: sv.po: Update Swedish translation
      l10n: zh_TW: Git 2.46
      l10n: Update German translation
      l10n: vi: Updated translation for 2.46
      l10n: uk: v2.46 update
      l10n: bg.po: Updated Bulgarian translation (5734t)
      l10n: fr: v2.46.0
      l10n: tr: Update Turkish translations
      l10n: po-id for 2.46
```

4. Show changes introduced by a specific commit.

```
C:\Users\SENJUTI\git>git show 114bff72ac030b9e9c931a9efd2bd0af8137692b
commit 114bff72ac030b9e9c931a9efd2bd0af8137692b
Author: Derrick Stolee <stolee@gmail.com>
Date:   Fri Jun 28 12:43:25 2024 +0000

    sparse-index: improve lstat caching of sparse paths

    The clear_skip_worktree_from_present_files() method was first introduced
    in af6a51875a (repo_read_index: clear SKIP_WORKTREE bit from files
    present in worktree, 2022-01-14) to allow better interaction with the
    working directory in the presence of paths outside of the
    sparse-checkout. The initial implementation would lstat() every single
    SKIP_WORKTREE path to see if it existed; if it ran across a sparse
    directory that existed (when a sparse index was in use), then it would
    expand the index and then check every SKIP_WORKTREE path.

    Since these lstat() calls were very expensive, this was improved in
    d79d299352 (Accelerate clear_skip_worktree_from_present_files() by
    caching, 2022-01-14) by caching directories that do not exist so it
    could avoid lstat()ing any files under such directories. However, there
    are some inefficiencies in that caching mechanism.

    The caching mechanism stored only the parent directory as not existing,
    even if a higher parent directory also does not exist. This means that
    wasted lstat() calls would occur when the paths passed to path_found()
    change immediate parent directories but within the same parent directory
    that does not exist.

    To create an example repository that demonstrates this problem, it helps
    to have a directory outside of the sparse-checkout that contains many
    deep paths. In particular, the first paths (in lexicographic order)
    underneath the sparse directory should have deep directory structures,
    maximizing the difference between the old caching algorithm that looks
    to a single parent and the new caching algorithm that looks to the
    top-most missing directory.

    The performance test script p2000-sparse-operations.sh takes the sample
    repository and copies its HEAD to several copies nested in directories
```

Exercise 3: Setting Up Git

Objective:

Set up Git configuration and verify it.

Instructions:

1. Set your username for Git.

```
C:\Users\SENJUTI\git>git config --global user.name "SENJUTI GHOSAL"
```

2. Set your email for Git.

```
C:\Users\SENJUTI\git>git config --global user.email "sj2656@srmist.edu.in"
```

3. Verify your configuration settings.

```
C:\Users\SENJUTI\git>git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/projects/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=SENJUTI GHOSAL
user.email=sj2656@srmist.edu.in
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=https://github.com/git/git.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.master.remote=origin
branch.master.merge=refs/heads/master
```

Exercise 4: Basic Git Commands

Objective:

Practice basic Git commands by modifying files and tracking changes.

Instructions:

1. Create a new file named file2.txt and add some content to it.

```
C:\Users\SENJUTI\git>echo "This is the content for file2.txt">file2.txt
```

2. Add the file to the staging area.

```
C:\Users\SENJUTI\git>git add file2.txt
```

3. Commit the new file with a commit message.

```
C:\Users\SENJUTI\git>git commit -m "Add file2.txt with initial content"
[master ad91b28bd3] Add file2.txt with initial content
 1 file changed, 1 insertion(+)
 create mode 100644 file2.txt
```

4. Modify the existing file1.txt and add more content to it.

```
C:\Users\SENJUTI\git>echo "Adding more content to the file1.txt">>file1.txt
```

5. Add the modified file to the staging area.

```
C:\Users\SENJUTI\git>git add file1.txt
```

6. Commit the changes with a commit message.

```
C:\Users\SENJUTI\git>git commit -m "Update file1.txt with additional content"
[master 859d32ee65] Update file1.txt with additional content
 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt
```

7. View the current status of your repository.

```
C:\Users\SENJUTI\git>git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

8. View the differences between your working directory and the repository.

```
C:\Users\SENJUTI\git>git diff
```

Exercise 5: Branching and Merging

Objective:

Create a new branch, make changes, and merge it back to the main branch.

Instructions:

1. Create a new branch named new-feature.

```
C:\Users\SENJUTI\git>git checkout -b new-feature
Switched to a new branch 'new-feature'
```

2. Switch to the new branch (if not already switched).

```
C:\Users\SENJUTI\git>git checkout -b new-feture
Switched to a new branch 'new-feture'
```

3. Create a new file named feature.txt and add some content to it.

```
C:\Users\SENJUTI\git>echo "This is the content for feature.txt">feature.txt
```

4. Add the file to the staging area.

```
C:\Users\SENJUTI\git>git add feature.txt
```

5. Commit the new file with a commit message.

```
C:\Users\SENJUTI\git>git commit -m "Add feature.txt with initial content"
[new-feture 5af6f6fa51] Add feature.txt with initial content
 1 file changed, 1 insertion(+)
 create mode 100644 feature.txt
```

6. Switch back to the main branch.

```
C:\Users\SENJUTI\git>git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 3 commits.
  (use "git push" to publish your local commits)
```

7. Merge the new-feature branch into the main branch.

```
C:\Users\SENJUTI\git>git merge new-feature
Already up to date.
```

8. Resolve any conflicts if they arise and commit the merge.
   No conflicts raised during the commit the merge.

Exercise 6: Remote Repositories

Objective:

Add a remote repository and push your local changes.

Instructions:

1. Add a remote repository URL to your local Git repository.

```
C:\Users\SENJUTI\git>git remote add cognizantweek1 https://github.com/senjuti09/cogn
izant_dn3.0_week1.git
```

2. Push your local changes to the remote repository.

```
C:\Users\SENJUTI\git>git push -u cognizantweek1 master
info: please complete authentication in your browser...
Enumerating objects: 360812, done.
Counting objects: 100% (360812/360812), done.
Delta compression using up to 12 threads
Compressing objects: 100% (84761/84761), done.
Writing objects: 100% (360812/360812), 236.27 MiB | 2.32 MiB/s, done.
Total 360812 (delta 273689), reused 360800 (delta 273684), pack-reused 0 (from 0)
```

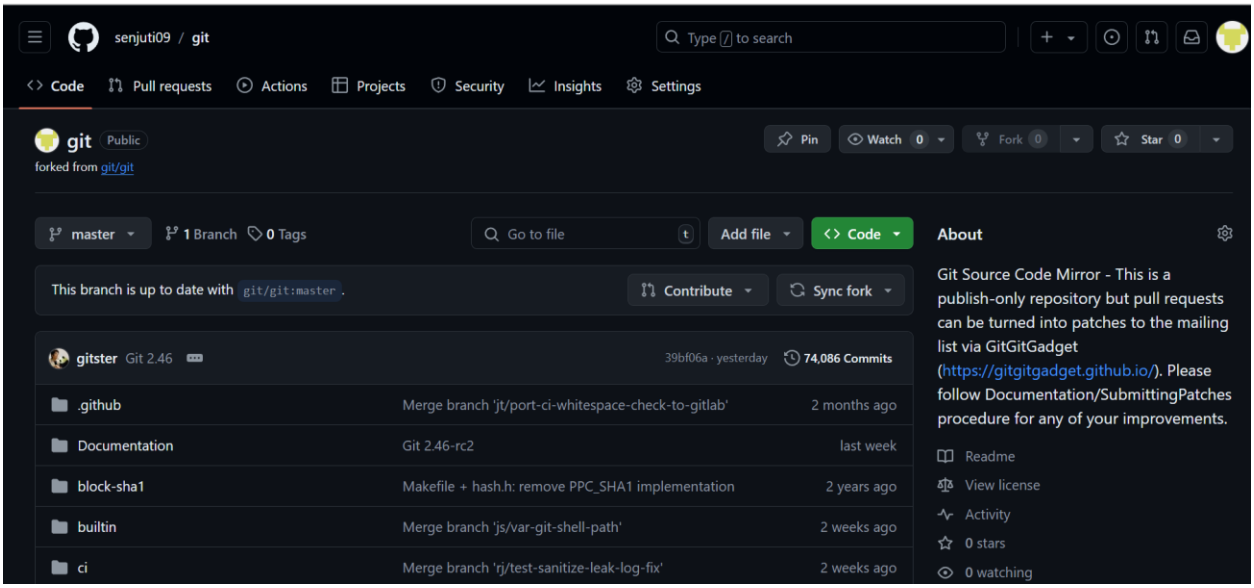
Exercise 7: Collaborating with Git

Objective:

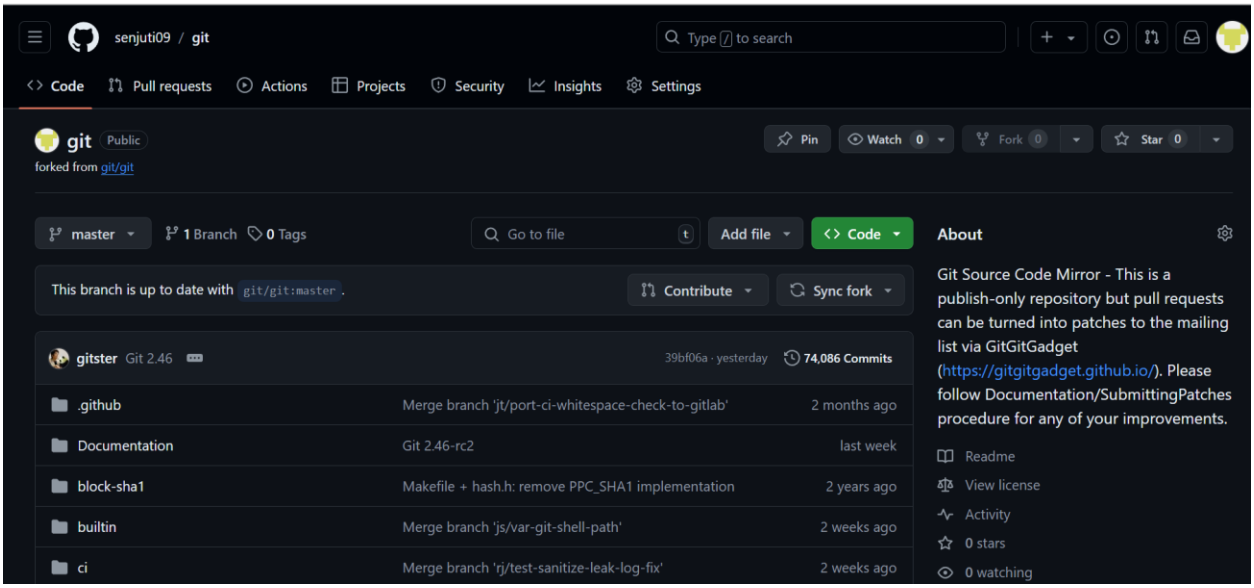
Collaborate on a repository by creating a pull request.

Instructions:

1. Fork a repository on GitHub.



2. Clone your forked repository to your local machine.



3. Navigate into the cloned repository.

```
:\Users\SENJUTI\git\git_cognizant>cd C:\Users\SENJUTI\git\git_cognizant
```

4. Create a new branch for your changes.

```
:\Users\SENJUTI\git\git_cognizant>git add changes.txt
```

5. Make your changes and commit them.

```
C:\Users\SENJUTI\git\git_cognizant>git commit -m "Add changes to changes.txt"
[my-changes (root-commit) 8224e9c] Add changes to changes.txt
 1 file changed, 1 insertion(+)
 create mode 100644 changes.txt
```

6. Push the branch to your forked repository.

```
C:\Users\SENJUTI\git\git_cognizant>git push -u origin my-changes
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 250 bytes | 41.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/senjuti09/git_cognizant
 * [new branch]      my-changes -> my-changes
branch 'my-changes' set up to track 'origin/my-changes'.
```

7. Create a pull request from your forked repository to the original repository.