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.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~
$ mkdir cog_project
```

2. Navigate into the directory.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~
$ cd cog_project
```

3. Initialize a new Git repository in the directory.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project
$ git init
Initialized empty Git repository in C:/Users/SENJUTI/cog_project/.git/
```

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

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project (master)
$ echo "Hello">file1.txt
```

5. Add the file to the staging area.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project (master)
$ git add file1.txt
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the nex
t time Git touches it
```

6. Commit the file with a commit message.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project (master)
$ git commit -m "Initial commit with file1.txt"
[master (root-commit) 9c99df7] 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.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project (master)
$ git clone https://github.com/git/git.git
Cloning into 'git'...
remote: Enumerating objects: 372039, done.
remote: Counting objects: 100% (658/658), done.
remote: Compressing objects: 100% (323/323), done.
remote: Total 372039 (delta 379), reused 536 (delta 334), pack-reused 371381
Receiving objects: 100% (372039/372039), 244.37 MiB | 11.90 MiB/s, done.
Resolving deltas: 100% (280052/280052), done.
Updating files: 100% (4516/4516), done.
```

2. Navigate into the cloned repository.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project (master)
$ cd git
```

3. Check the commit history.

```
$ git log
commit 891ee3b9db5cbd35a9759896f347546c5edb7929 (HEAD -> master, origin/master,
origin/HEAD)
Author: Junio C Hamano <gitster@pobox.com>
Date:   Wed Jul 31 13:02:10 2024 -0700

    Start the 2.47 cycle

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

commit 3ff9ceca89aa187ae3874dbf013a1f1f356759db
Merge: d18eb5ba79 bb0498b1bb
Author: Junio C Hamano <gitster@pobox.com>
Date:   Wed Jul 31 13:34:21 2024 -0700
:...skipping...
commit 891ee3b9db5cbd35a9759896f347546c5edb7929 (HEAD -> master, origin/master,
origin/HEAD)
Author: Junio C Hamano <gitster@pobox.com>
```

4. Show changes introduced by a specific commit.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ 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
    of the form f<i>/f<j>/f<k> where i, j, and k are numbers from 1 to 4.
    The sparse-checkout cone is then selected as "f2/f4/". Creating "f1/f1/"
    will trigger the behavior and also lead to some interesting cases for
    the caching algorithm since "f1/f1/" exists but "f1/f2/" and "f3/" do
    not.
```

Exercise 3: Setting Up Git

Objective:

Set up Git configuration and verify it.

Instructions:

1. Set your username for Git.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ git config --global user.name "SENJUTI GHOSAL"
```

2. Set your email for Git.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ git config --global user.email "sj2656@srmist.edu.in"
```

3. Verify your configuration settings.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ 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
```

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.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ echo "This is the content for file2.txt">file2.txt
```

2. Add the file to the staging area.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ git add file2.txt
warning: in the working copy of 'file2.txt', LF will be replaced by CRLF the nex
t time Git touches it
```

3. Commit the new file with a commit message.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ git commit -m "Add file2.txt with initial content"
[master ff58a21630] 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.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ echo "Adding more content to the file1.txt">>file1.txt
```

5. Add the modified file to the staging area.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ git add file1.txt
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the nex
t time Git touches it
```

6. Commit the changes with a commit message.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ git commit -m "update file1.txt with additional content"
[master 658c4bdd06] update file1.txt with additional content
 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt
```

7. View the current status of your repository.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)
```

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

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ 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.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ git checkout -b new-feature
Switched to a new branch 'new-feature'
```

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

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ git checkout -b new-feature
Switched to a new branch 'new-feature'
```

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

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (new-feature)
$ echo "this is the content for feature.txt">feature.txt
```

4. Add the file to the staging area.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (new-feature)
$ git add feature.txt
warning: in the working copy of 'feature.txt', LF will be replaced by CRLF the n
ext time Git touches it
```

5. Commit the new file with a commit message.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (new-feature)
$ git commit -m "Add feature.txt with initial content"
[new-feature 1ece584073] Add feature.txt with initial content
 1 file changed, 1 insertion(+)
 create mode 100644 feature.txt
```

6. Switch back to the main branch.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (new-feature)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)
```

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

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ git merge new-feature
Updating 658c4bdd06..1ece584073
Fast-forward
 feature.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 feature.txt
```

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.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git (master)
$ git remote add cognizantweek1 https://github.com/senjuti09/cognizant_dn3.0_wee
k1.git
```

2. Push your local changes to the remote repository.

```
$ git push cognizantweek1 master
Enumerating objects: 498, done.
Counting objects: 100% (357/357), done.
Delta compression using up to 12 threads
Compressing objects: 100% (120/120), done.
Writing objects: 100% (215/215), 110.28 KiB | 3.24 MiB/s, done.
Total 215 (delta 154), reused 139 (delta 91), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (154/154), completed with 54 local objects.
To https://github.com/senjuti09/cognizant_dn3.0_week1.git
   5bd70f89a7..5fe60fd086  master -> master
```

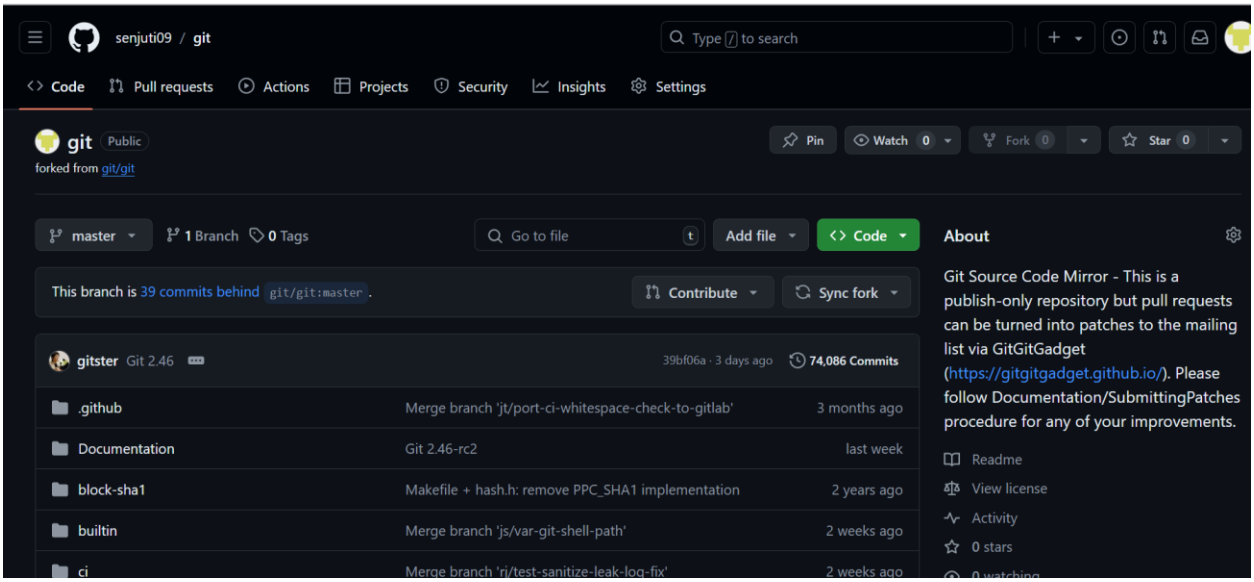
Exercise 7: Collaborating with Git

Objective:

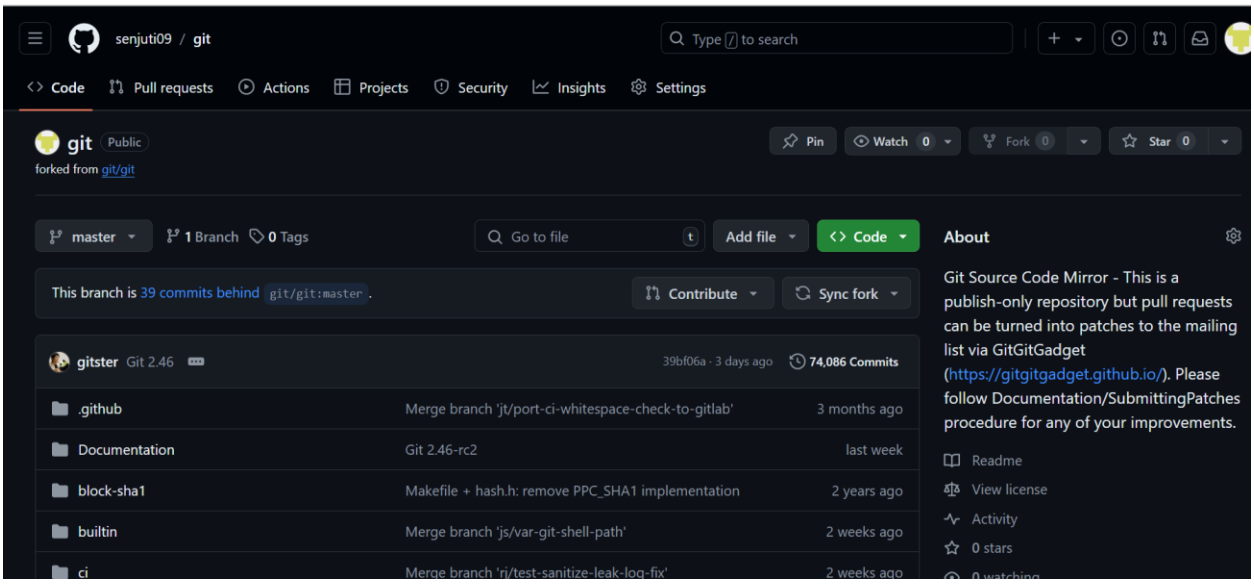
Collaborate on a repository by creating a pull request.

Instructions:

1. Fork a repository on GitHub.



2. Clone your forked repository to your local machine.



```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git/git_cognizant (cog-changes)
$ git clone https://github.com/senjuti09/git
Cloning into 'git'...
remote: Enumerating objects: 360803, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 360803 (delta 21), reused 32 (delta 20), pack-reused 360765
Receiving objects: 100% (360803/360803), 236.45 MiB | 13.05 MiB/s, done.
Resolving deltas: 100% (273685/273685), done.
Updating files: 100% (4509/4509), done.
```

3. Navigate into the cloned repository.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git/git_cognizant (cog-changes)
$ cd git
```

4. Create a new branch for your changes.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git/git_cognizant/git (master)
$ git checkout -b new-branch
Switched to a new branch 'new-branch'
```

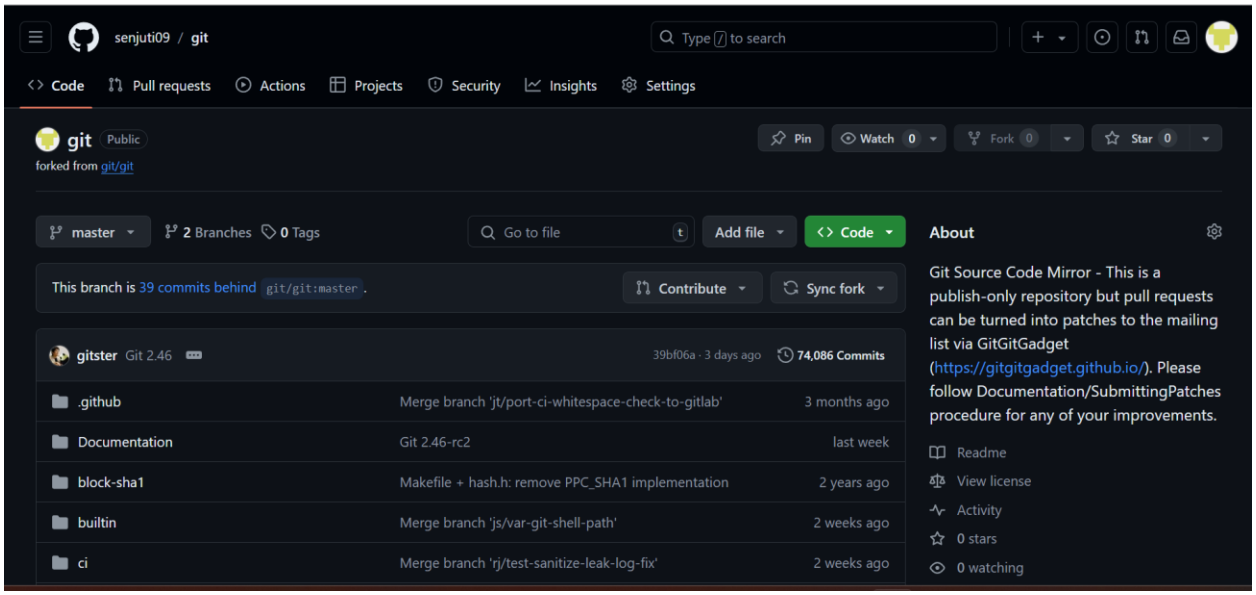
5. Make your changes and commit them.

```
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.

```
SENJUTI@LAPTOP-KI2DPS1M MINGW64 ~/cog_project/git/git_cognizant/git (new-branch)
$ git push -u origin new-branch
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'new-branch' on GitHub by visiting:
remote:      https://github.com/senjuti09/git/pull/new/new-branch
remote:
To https://github.com/senjuti09/git
 * [new branch]            new-branch -> new-branch
branch 'new-branch' set up to track 'origin/new-branch'.
```

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


senjuti09 / git
Type / to search
Code
Pull requests
Actions
Projects
Security
Insights
Settings
git
Public
forked from git/git
Pin
Watch 0
Fork 0
Star 0
master
2 Branches
0 Tags
Go to file
t
Add file
Code
This branch is 39 commits behind git/git:master .
Contribute
Sync fork
gitster Git 2.46
39bf06a · 3 days ago
74,086 Commits
.github
Merge branch 'jt/port-ci-whitespace-check-to-gitlab'
3 months ago
Documentation
Git 2.46-rc2
last week
block-sha1
Makefile + hash.h: remove PPC_SHA1 implementation
2 years ago
builtin
Merge branch 'js/var-git-shell-path'
2 weeks ago
ci
Merge branch 'rj/test-sanitize-leak-log-fix'
2 weeks ago
About
Git Source Code Mirror - This is a publish-only repository but pull requests can be turned into patches to the mailing list via GitGitGadget (https://gitgitgadget.github.io/). Please follow Documentation/SubmittingPatches procedure for any of your improvements.
Readme
View license
Activity
0 stars
0 watching