VIDHI ROHIRA

SY B.TECH

COMPUTER ENGINEERING

OST LAB-3
231071052

BATCH C

EXPERIMENT NO:- 3

AIM:- Version control systems, introduction to git, basic git commands (clone, commit, pull, push, branch, merge) working with GItHub or GitLab.

THEORY:-

Q1] What is git and why is it called as version control ?

- **Git-** Git is a distributed version control system used to track changes in source code during software development. It's widely adopted because it helps manage and coordinate work on projects among multiple developers and teams.
- Version control refers to the management of changes to documents, programs, and other information stored as computer files. It allows you to track and manage changes, keeping a history of revisions and enabling collaborative work.The term "version control" captures the essence of its functionality: controlling and managing different versions of files and tracking changes throughout the development lifecycle.

Q2] Explain the purpose of git clone command.

- The `git clone` command is used to create a copy of an existing Git repository. This command serves several purposes in the context of Git version control:
  1.] Initial Copy of a Repository
  2.] Getting Started with a Project
  3.] Setting Up a Working Environment
- `git clone` is a fundamental command in Git for creating a full local copy of a remote repository, setting up your working environment, and enabling collaboration.

Q3] What is the difference between git add and git commit ?

Feature	`git add`	`git commit`
Purpose	Stages changes for the next commit.	Records staged changes as a new commit in the repository history.
Function	Adds changes in files (modified, new, or deleted) to the staging area (index).	Creates a commit from the changes staged in the index and records it in the repository history.
Scope	Affects the staging area, not the repository.	Affects the repository by creating a new commit.

Q4] How does push command work ?

The `git push` command uploads your local repository changes to a remote repository. Here's a concise overview of how it works:

1. Update Remote: It updates the remote repository with your local commits, making the changes available to others.
2. Transmit Changes: Commits from your local branch are sent to the corresponding branch on the remote server.
3. Branch-Specific: By default, it pushes changes from the current local branch to the remote branch with the same name.
4. Conflict Handling: If there are changes on the remote branch not present locally, Git will prevent the push until you resolve conflicts by pulling and merging those changes.
5. Authentication: Pushing requires authentication (e.g., SSH keys or tokens) to verify your access to the remote repository.
6. Common Commands:
   - `git push origin main`: Pushes local `main` branch to `origin`.
   - `git push --force`: Overwrites remote branch with local branch (use cautiously).
7. Initial Push: When pushing a branch for the first time, Git creates it on the remote repository.

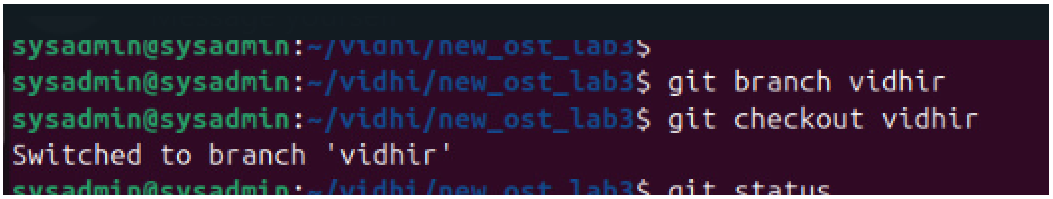
Q5] How do you create a new branch in git using the command line ?

1.] Create the Branch:

git branch <branch-name>

2.] Switch to the New Branch:

git checkout <branch-name>



```
sysadmin@sysadmin:~/vidhi/new_ost_lab3$
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git branch vidhir
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git checkout vidhir
Switched to branch 'vidhir'
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git status
```

Here we have created a branch and entered the branch too.

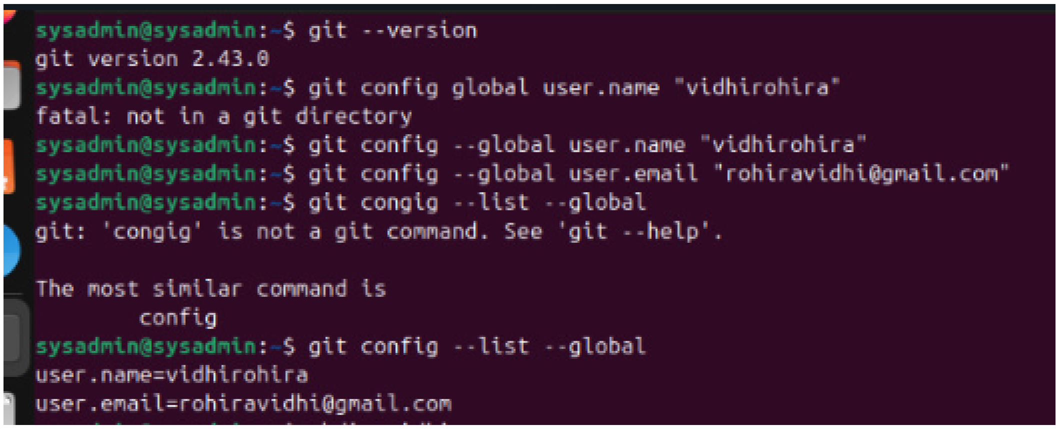
Q6] Explain how git merge command works.

The `git merge` command is used to integrate changes from one branch into another in a Git repository. When you run `git merge <branch-name>` from your current branch, Git performs the following steps:

1. Identifies Common Ancestor: Git finds the common ancestor (or merge base) of the current branch and the branch you're merging.
2. Compares Changes: It then compares the changes between the common ancestor and the latest commits of both branches.
3. Applies Changes: Git attempts to automatically apply changes from the branch being merged into the current branch.

4. Handles Conflicts: If there are conflicting changes that Git can't automatically resolve, it will mark the conflicts in the affected files and require manual resolution.
5. Creates a Merge Commit: If the merge is successful, Git creates a new commit called a merge commit that combines the changes from both branches, preserving their history**.**

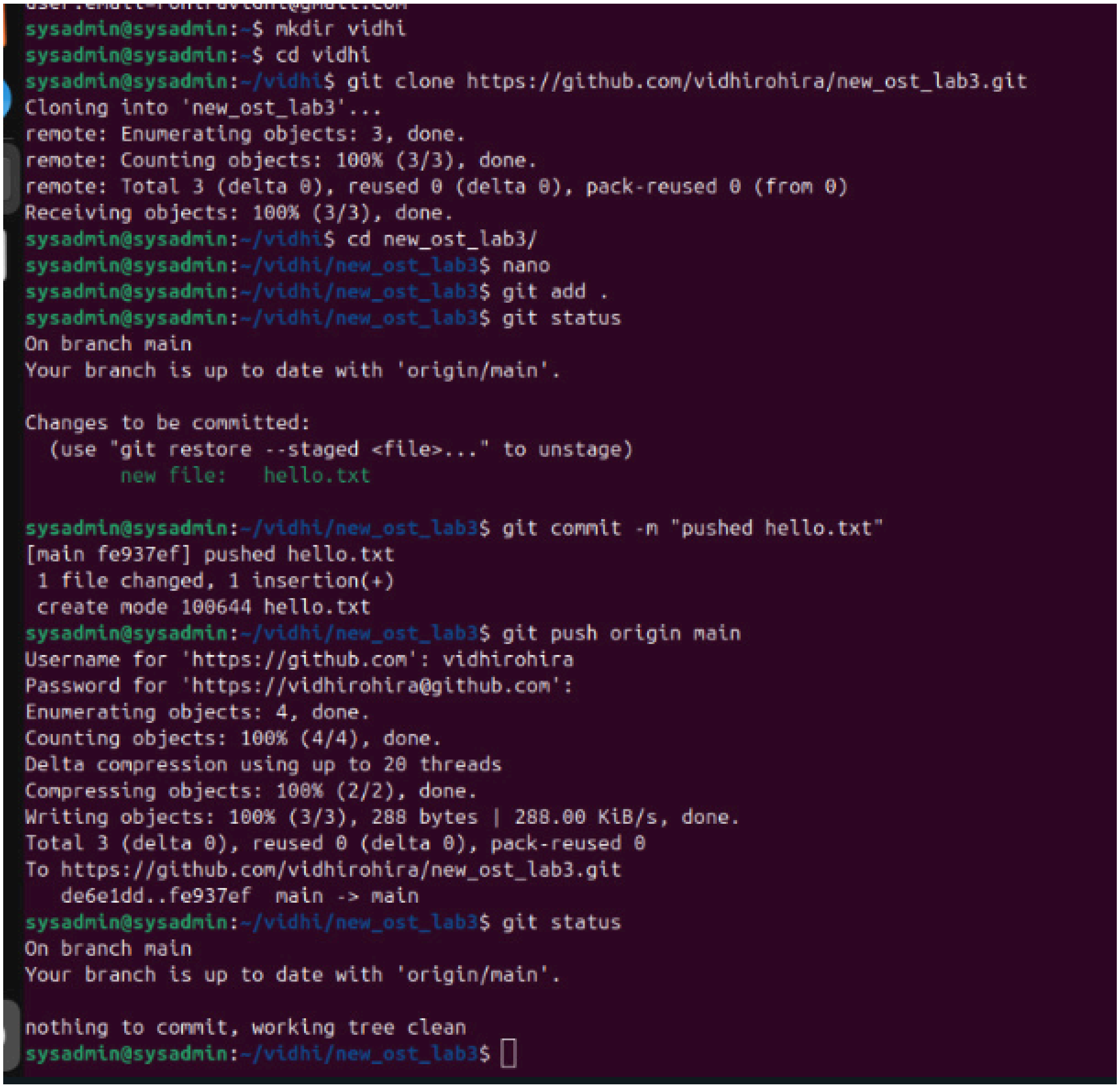
SCREENSHOTS OF LAB AND IT’S EXPLANATION:-



```
sysadmin@sysadmin:~$ git --version
git version 2.43.0
sysadmin@sysadmin:~$ git config global user.name "vidhirohira"
fatal: not in a git directory
sysadmin@sysadmin:~$ git config --global user.name "vidhirohira"
sysadmin@sysadmin:~$ git config --global user.email "rohiravidhi@gmail.com"
sysadmin@sysadmin:~$ git congig --list --global
git: 'congig' is not a git command. See 'git --help'.

The most similar command is
        config
sysadmin@sysadmin:~$ git config --list --global
user.name=vidhirohira
user.email=rohiravidhi@gmail.com
```

Checked whether our pc has git installed and what version of it is installed. Used commands to configure into our GitHub account and used git config --list --global to list username and email.



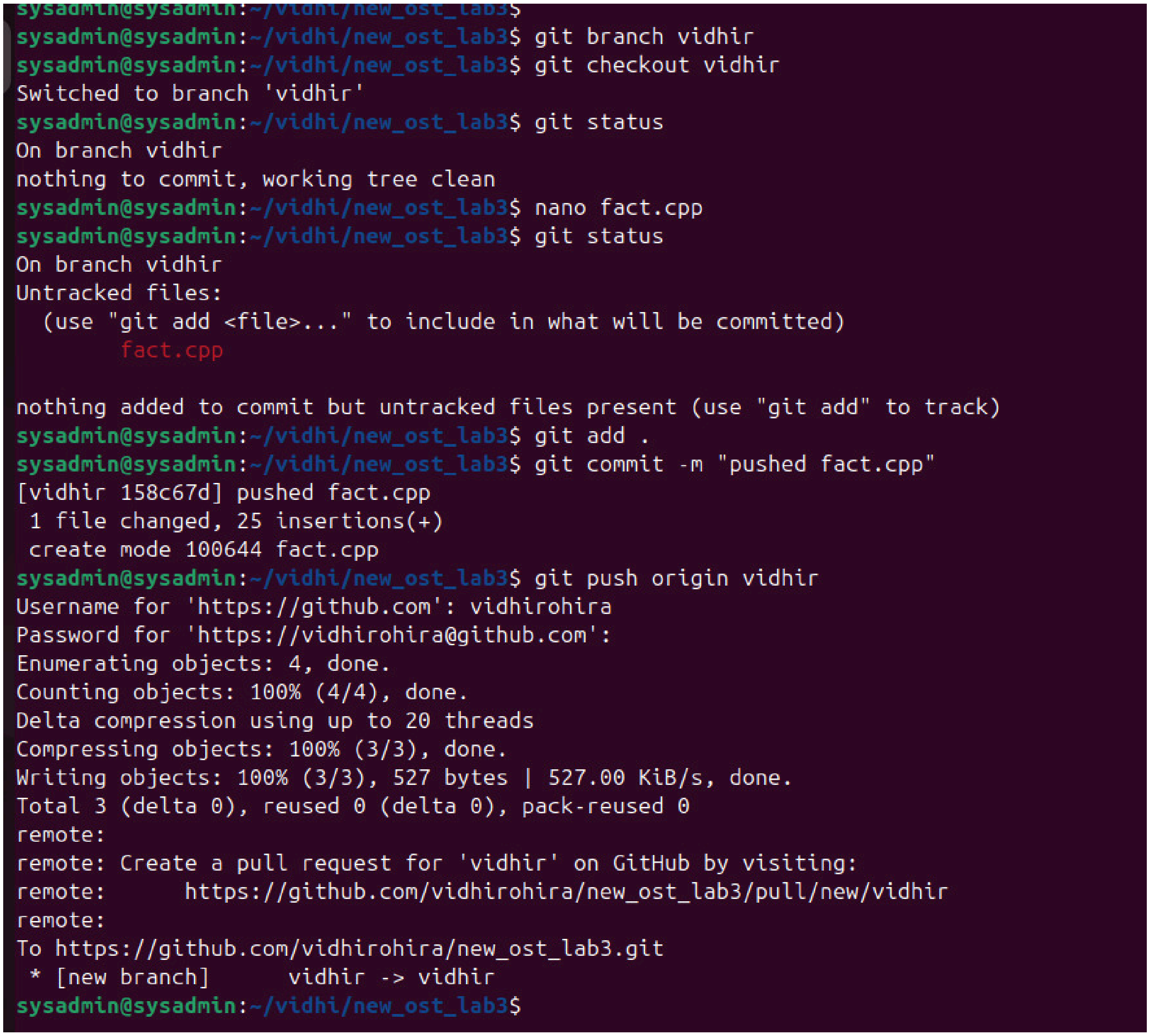
```
sysadmin@sysadmin:~$ mkdir vidhi
sysadmin@sysadmin:~$ cd vidhi
sysadmin@sysadmin:~/vidhi$ git clone https://github.com/vidhirohira/new_ost_lab3.git
Cloning into 'new_ost_lab3'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
sysadmin@sysadmin:~/vidhi$ cd new_ost_lab3/
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ nano
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git add .
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   hello.txt

sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git commit -m "pushed hello.txt"
[main fe937ef] pushed hello.txt
 1 file changed, 1 insertion(+)
 create mode 100644 hello.txt
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git push origin main
Username for 'https://github.com': vidhirohira
Password for 'https://vidhirohira@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 288 bytes | 288.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/vidhirohira/new_ost_lab3.git
   de6e1dd..fe937ef  main -> main
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ ▯
```

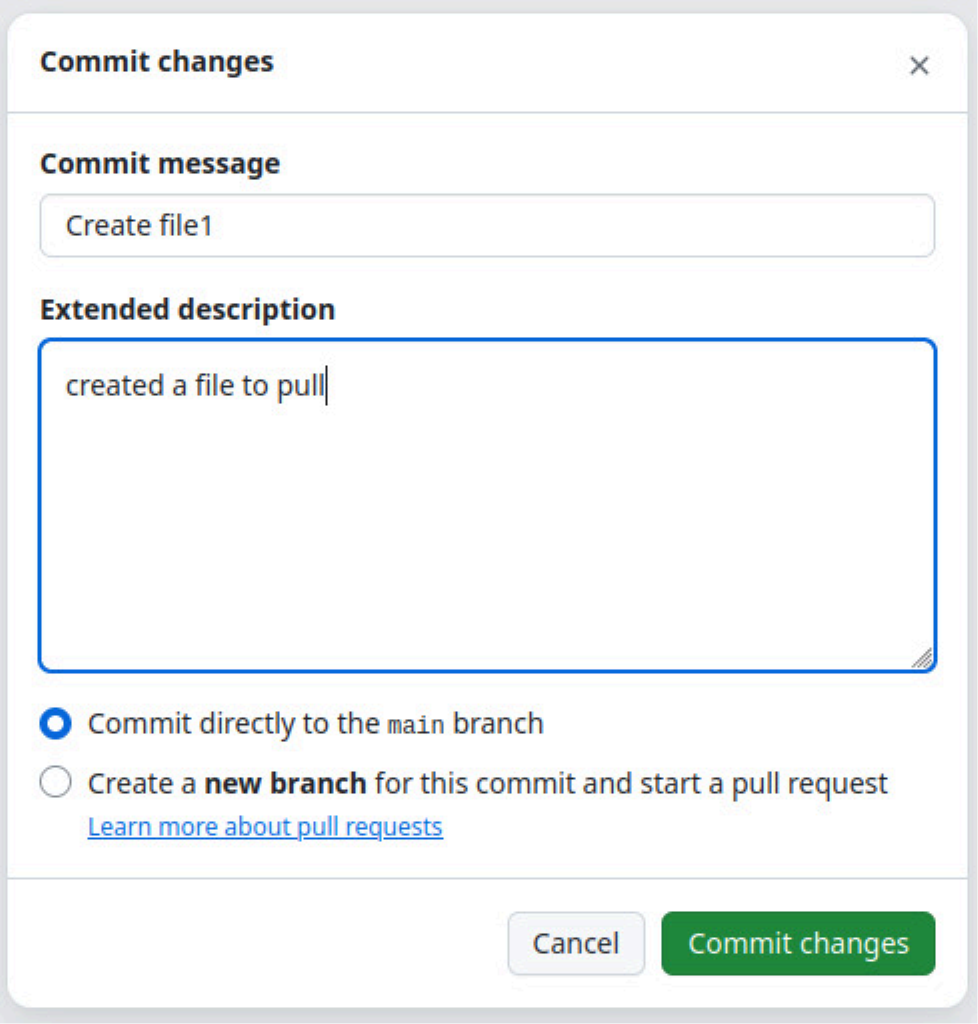
Created a new file named vidhi and entered it. Used git clone and cd command to clone into required GitHub repository. Created a new file using nano and pushed it onto our repository in main branch using git commands.



```
sysadmin@sysadmin:~/vidhi/new_ost_lab3$
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git branch vidhir
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git checkout vidhir
Switched to branch 'vidhir'
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git status
On branch vidhir
nothing to commit, working tree clean
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ nano fact.cpp
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git status
On branch vidhir
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        fact.cpp

nothing added to commit but untracked files present (use "git add" to track)
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git add .
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git commit -m "pushed fact.cpp"
[vidhir 158c67d] pushed fact.cpp
 1 file changed, 25 insertions(+)
 create mode 100644 fact.cpp
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git push origin vidhir
Username for 'https://github.com': vidhirohira
Password for 'https://vidhirohira@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 20 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 527 bytes | 527.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'vidhir' on GitHub by visiting:
remote:      https://github.com/vidhirohira/new_ost_lab3/pull/new/vidhir
remote:
To https://github.com/vidhirohira/new_ost_lab3.git
 * [new branch]      vidhir -> vidhir
sysadmin@sysadmin:~/vidhi/new_ost_lab3$
```

Created a branch named vidhir using git commands. Created a file using nano and pushed the file in our newly created branch using git commands.



Commit changes

Commit message

Create file1

Extended description

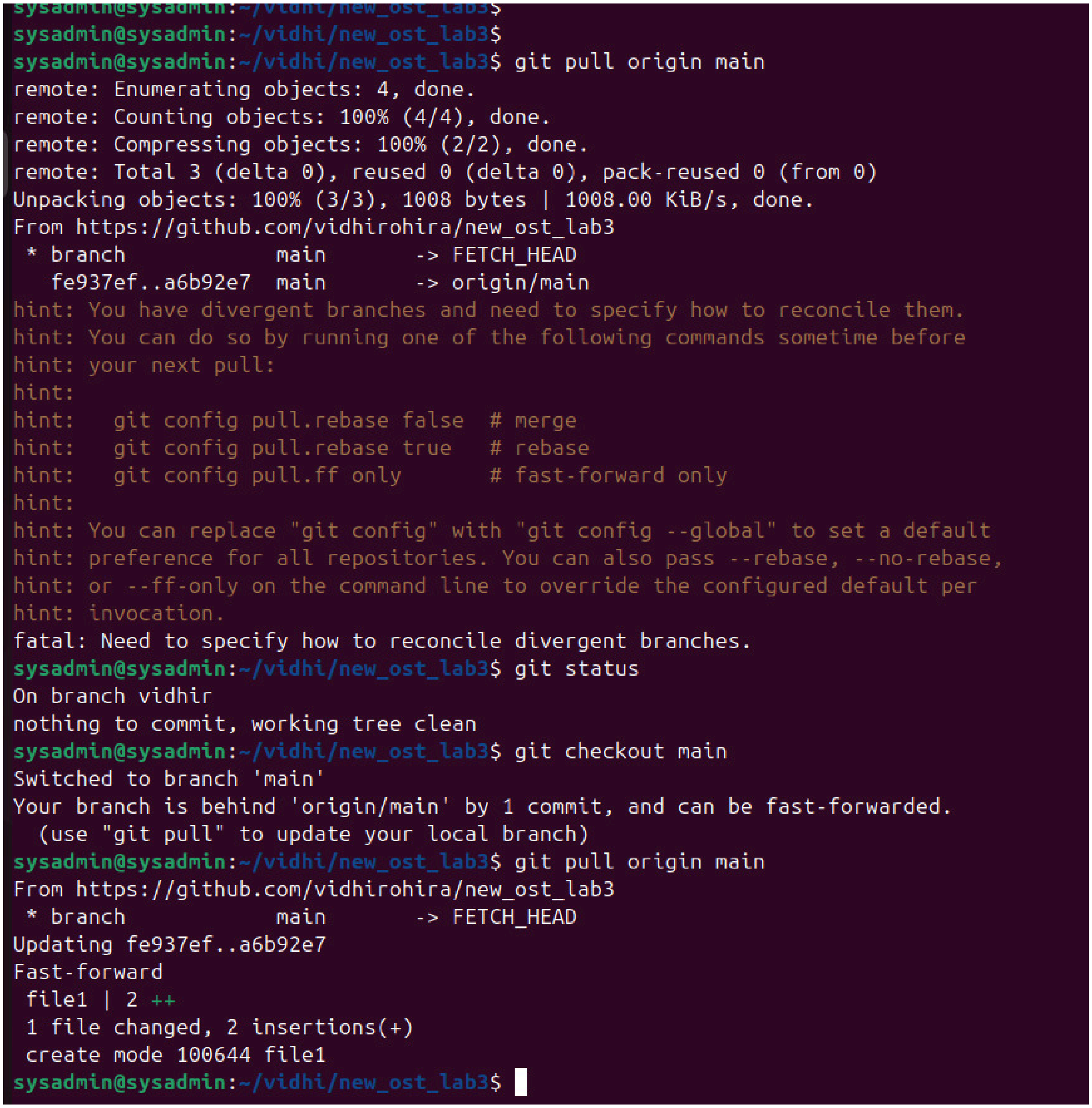
created a file to pull

● Commit directly to the `main` branch

○ Create a **new branch** for this commit and start a pull request

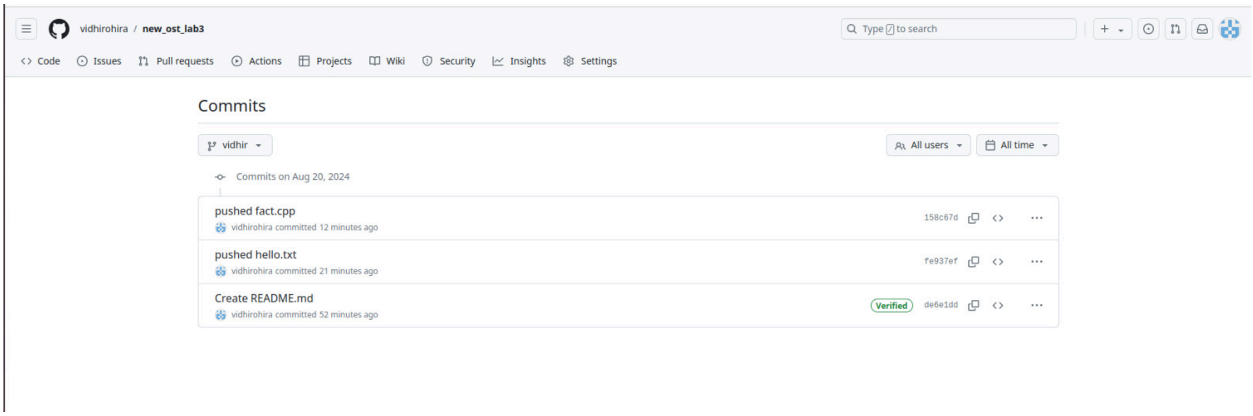
Learn more about pull requests

Created a file on the GitHub website to pull it using git commands.



```
sysadmin@sysadmin:~/vidhi/new_ost_lab3$
sysadmin@sysadmin:~/vidhi/new_ost_lab3$
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git pull origin main
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1008 bytes | 1008.00 KiB/s, done.
From https://github.com/vidhirohira/new_ost_lab3
 * branch            main       -> FETCH_HEAD
   fe937ef..a6b92e7  main       -> origin/main
hint: You have divergent branches and need to specify how to reconcile them.
hint: You can do so by running one of the following commands sometime before
hint: your next pull:
hint:
hint:   git config pull.rebase false  # merge
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only       # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
fatal: Need to specify how to reconcile divergent branches.
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git status
On branch vidhir
nothing to commit, working tree clean
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git checkout main
Switched to branch 'main'
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ git pull origin main
From https://github.com/vidhirohira/new_ost_lab3
 * branch            main       -> FETCH_HEAD
Updating fe937ef..a6b92e7
Fast-forward
 file1 | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 file1
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ 
```

Used git pull command to pull or obtain the file required from remote source to local source in our computer. Faced an error as shown because i tried to pull in the branch itself and not main where the file was.



Commits

vidhir

All users

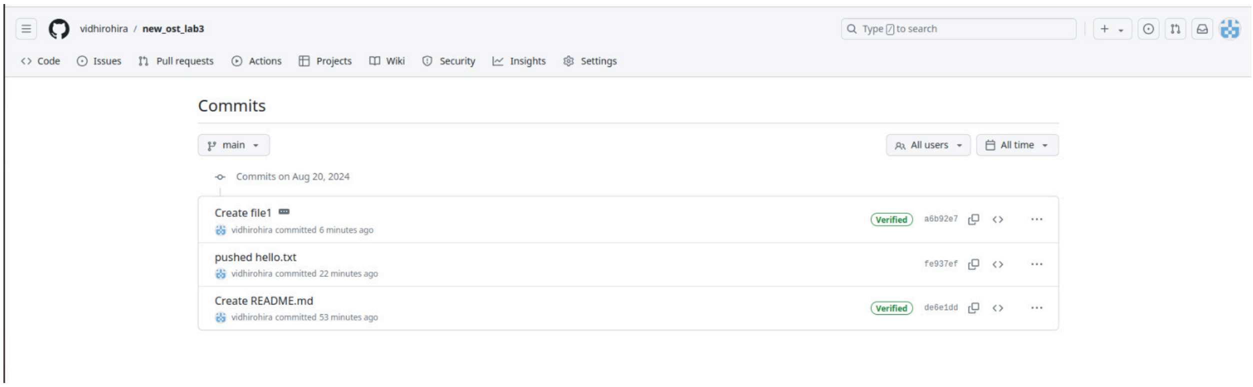
All time

Commits on Aug 20, 2024

pushed fact.cpp
vidhirohira committed 12 minutes ago
158c67d

pushed hello.txt
vidhirohira committed 21 minutes ago
fe937ef

Create README.md
vidhirohira committed 52 minutes ago
Verified de6e1dd



Commits

main

All users

All time

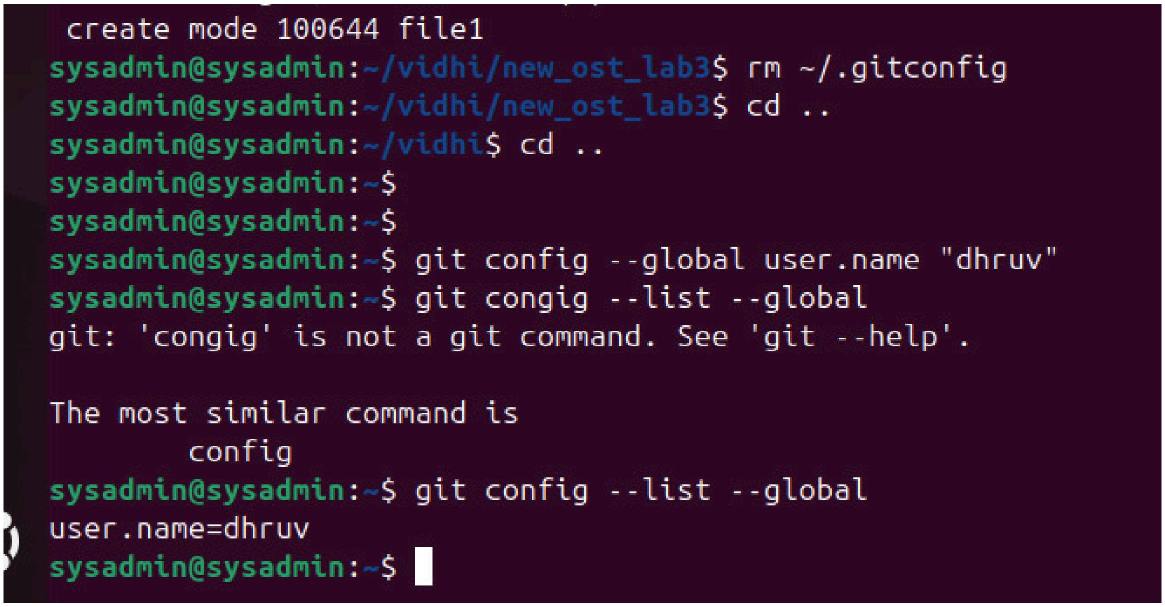
Commits on Aug 20, 2024

Create file1
vidhirohira committed 6 minutes ago
Verified a6b92e7

pushed hello.txt
vidhirohira committed 22 minutes ago
fe937ef

Create README.md
vidhirohira committed 53 minutes ago
Verified de6e1dd

Screenshots of commits in both main and newly created branch



```
 create mode 100644 file1
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ rm ~/.gitconfig
sysadmin@sysadmin:~/vidhi/new_ost_lab3$ cd ..
sysadmin@sysadmin:~/vidhi$ cd ..
sysadmin@sysadmin:~$
sysadmin@sysadmin:~$
sysadmin@sysadmin:~$ git config --global user.name "dhruv"
sysadmin@sysadmin:~$ git congig --list --global
git: 'congig' is not a git command. See 'git --help'.

The most similar command is
        config
sysadmin@sysadmin:~$ git config --list --global
user.name=dhruv
sysadmin@sysadmin:~$ 
```

Command to logout of repo from terminal.

CONCLUSION:- Hence we have learnt about Version control systems, introduction to git, basic git commands (clone, commit, pull, push, branch, merge) working with GItHub or GitLab. We have also implemented the basic essential commands on terminal to clone, commit, pull, push, create branch and merge in today's laboratory session.