VTU 2025 Scheme

Project Management (with Git)

1BCSL307A – 3rd Semester

EXPERIMENT – 04

Collaboration and Remote Repositories

Cloning a Remote Git Repository

Aim

To clone a remote Git repository to the local machine using the `git clone` command.

Objectives

After completing this experiment, the student will be able to:

- Understand the concept of remote repositories.
- Understand the purpose of repository cloning.
- Copy a remote Git repository to the local machine.
- Understand the use of the `git clone` command.
- Verify the files downloaded from a remote repository.
- View the remote repository configuration.
- Understand the relationship between local and remote repositories.

Software Requirements

- Git
- Internet Connection
- Windows Command Prompt / PowerShell / Git Bash
- Access to a remote Git repository

Theory

A **remote repository** is a Git repository stored on another computer, server, or remote repository hosting service.

Remote repositories allow developers to share project files and collaborate with other developers.

A developer can create a local copy of an existing remote repository using the `git clone` command.

What is Git Clone?

`git clone` is a Git command used to create a complete local copy of an existing remote repository.

When a repository is cloned, Git downloads:

- Project files
- Directory structure
- Commit history
- Branch information
- Git repository metadata
- Remote repository configuration

The cloned repository can then be used for local development.

Why is Repository Cloning Used?

- To obtain a local copy of an existing project.
- To start working on a shared software project.
- To collaborate with other developers.
- To access the complete Git history of a project.
- To create a local working environment.
- To obtain project files without manually downloading them.

Remote and Local Repository Concept

The basic cloning operation is shown below.

Diagram Explanation:

- The original Git repository is available remotely.
- The user executes the `git clone` command.
- Git downloads the repository to the local machine.
- The local repository contains project files and Git history.
- The developer can now work on the project locally.

Important Commands

Command	Purpose
`git clone <URL>`	Creates a local copy of a remote Git repository.
`cd repository-name`	Moves into the cloned repository directory.
`git status`	Displays the current repository status.
`git remote -v`	Displays configured remote repository URLs.
`git branch`	Displays the local branches.
`git log --oneline`	Displays the commit history in compact form.

Procedure

Step 1: Install Git on the local machine.

Step 2: Open Command Prompt, PowerShell, or Git Bash.

Step 3: Select the location where the repository should be cloned.

Step 4: Obtain the URL of the required remote Git repository.

Step 5: Execute the `git clone` command with the repository URL.

Step 6: Wait until Git downloads the repository.

Step 7: Move into the cloned repository directory.

Step 8: Display the files available in the repository.

Step 9: Check the Git repository status.

Step 10: Display the configured remote repository.

Step 11: Display the available branches.

Step 12: View the commit history.

Step 13: Verify that the repository was cloned successfully.

Execution

Step 1: Open Terminal

Open:

- Command Prompt, or
- PowerShell, or
- Git Bash

Check whether Git is installed:

```
git --version
```

Sample Output:

```
git version 2.x.x
```

The exact version depends on the Git installation.

Step 2: Select a Working Location

For example:

```
cd Desktop
```

This changes the current working location to the Desktop.

Step 3: Obtain Remote Repository URL

A remote repository URL generally has the following form:

```
https://github.com/username/repository-name.git
```

Replace:

```
username
```

with the repository owner's username, and replace:

```
repository-name
```

with the actual repository name.

Step 4: Clone the Remote Repository

Execute:

```
git clone https://github.com/username/repository-name.git
```

Sample Output:

```
Cloning into 'repository-name'...
remote: Enumerating objects: ...
remote: Counting objects: 100% (...)
Receiving objects: 100% (...)
Resolving deltas: 100% (...)
```

Explanation:

The `git clone` command downloads the remote repository and creates a new local directory.

Step 5: Move into the Cloned Repository

Execute:

```
cd repository-name
```

The current working directory is now the cloned repository.

Step 6: Display Repository Files

On Windows, execute:

```
dir
```

On Git Bash, Linux, or macOS, execute:

```
ls
```

The command displays the project files downloaded from the remote repository.

Step 7: Check Repository Status

Execute:

```
git status
```

Sample Output:

```
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Note:

The default branch may be `main`, `master`, or another branch depending on the remote repository.

Step 8: Display Remote Repository

Execute:

```
git remote -v
```

Sample Output:

```
origin  https://github.com/username/repository-name.git (fetch)
origin  https://github.com/username/repository-name.git (push)
```

Explanation:

The name `origin` is normally assigned automatically to the remote repository from which the project was cloned.

Step 9: Display Available Branches

Execute:

```
git branch
```

Sample Output:

```
* main
```

The asterisk (*) indicates the currently active local branch.
To display both local and remote-tracking branches:

```
git branch -a
```

Step 10: Display Commit History

Execute:

```
git log --oneline
```

Sample Output:

```
a1b2c3d Update README
d4e5f6g Add project files
h7i8j9k Initial commit
```

Note:
The actual commit IDs and messages depend on the cloned repository.

Complete Command Sequence

The main command sequence for this experiment is:

```
git --version

cd Desktop

git clone https://github.com/username/repository-name.git

cd repository-name

dir

git status

git remote -v

git branch
```

```
git branch -a

git log --oneline
```

Understanding the Clone Process

When the following command is executed:

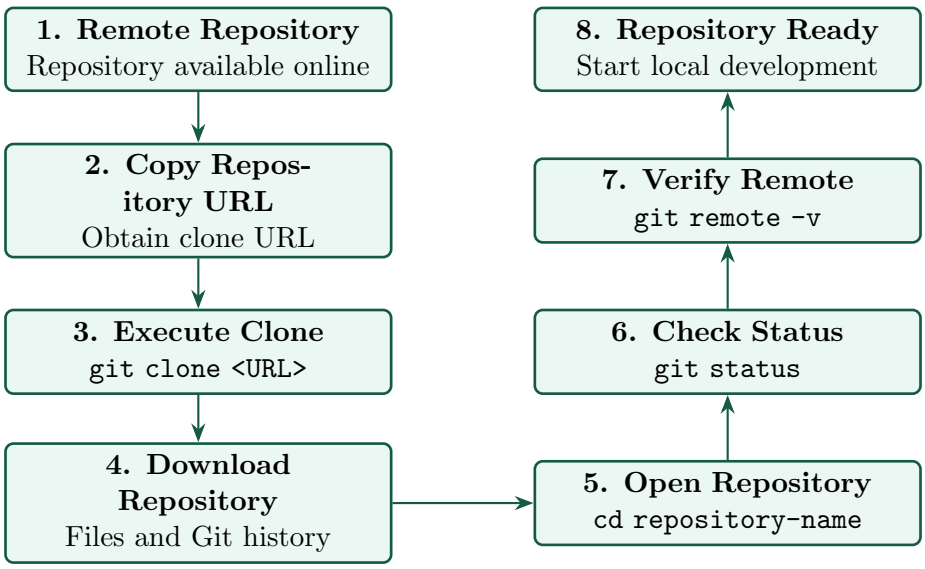
```
git clone https://github.com/username/repository-name.git
```

Git performs the following operations:

1. Connects to the remote repository.
2. Creates a new local directory.
3. Initializes a local Git repository.
4. Downloads repository objects.
5. Downloads the commit history.
6. Creates remote-tracking information.
7. Checks out the remote repository's default branch.
8. Configures the remote repository as `origin`.

Experiment Workflow

The complete workflow of Experiment 04 is shown below.



Workflow Explanation:

1. Identify the remote Git repository.
2. Obtain its clone URL.
3. Execute the `git clone` command.
4. Git downloads the repository files and history.
5. Move into the cloned repository.
6. Verify the repository status.
7. Check the configured remote using `git remote -v`.
8. The repository is now ready for local development.

Local Repository vs Remote Repository

Feature	Local Repository	Remote Repository
Location	Stored on the developer's computer	Stored on a remote server or hosting service
Main Purpose	Local development and version control	Sharing and collaboration
Internet Required	Not required for normal local operations	Required to communicate with the remote
Example Operation	`git commit`	`git clone`, fetch, pull, or push

Result

Thus, a remote Git repository was successfully cloned to the local machine using the `git clone` command. The downloaded project files, branches, commit history, and remote repository configuration were verified successfully.

Viva Questions and Answers

1. **What is a remote repository?**

   A remote repository is a Git repository stored on another computer, server, or repository hosting service.

2. **What is repository cloning?**

   Repository cloning is the process of creating a local copy of an existing remote Git repository.

3. **Which command is used to clone a Git repository?**

```
git clone <repository-URL>
```

4. **What does git clone download?**

   It downloads the project files, Git history, branches, and repository information required for the local copy.

5. **What is origin in Git?**

   `origin` is the conventional default name Git assigns to the remote repository from which a repository is cloned.

6. **How can we view configured remote repositories?**

```
git remote -v
```

7. **What does the -v option mean?**

   The `-v` option means verbose. It displays the remote names along with their URLs.

8. **Does git clone create a Git repository locally?**

   Yes. It creates a local Git repository containing the downloaded repository data.

9. **What is the difference between git clone and git init?**

   `git init` initializes a Git repository in an existing or new local directory.

   `git clone` creates a local copy of an existing remote repository.

10. **Which command checks the current repository status?**

```
git status
```

11. **How do you display local branches?**

```
git branch
```

12. **How do you display local and remote-tracking branches?**

```
git branch -a
```

13. **How do you view the commit history in a short format?**

```
git log --oneline
```

14. **Is an Internet connection required after cloning?**

    Most local Git operations such as editing files, staging, committing, and viewing history can be performed without an Internet connection. Communication with a remote repository requires network access.

Conclusion

In this experiment, the process of cloning a remote Git repository was demonstrated. The remote repository was copied to the local machine using `git clone`. The cloned repository was then verified using Git status, branch, remote, and log commands.