(TR-102) MASTERING THE SEMANTIC WEB –

Training Day 17 Report :

5 July 2024

Introduction to Git and Git Bash:

Git:

Git is a distributed version control system designed to handle everything from small to very large projects with speed and efficiency. It allows multiple developers to work on the same project simultaneously without interfering with each other's work. Git tracks changes to files and allows to revert to previous versions, compare changes, and collaborate with others.

Key Features of Git:

- Version Control: Tracks changes to files and allows you to revert to previous versions.
- Branching and Merging: Allows you to create branches to work on different features or fixes and merge them back into the main branch.

By: Jaspinder Kaur Walia URN:2203843 CRN:2215199

- Distributed System: Every developer has a full copy of the repository, including its history, on their local machine.
- Collaboration: Facilitates collaboration between multiple developers, ensuring everyone has the latest version of the project.

Basic Git Commands:

- git init: Initializes a new Git repository.
- git clone <repository>: Clones an existing repository to your local machine.
- git status: Shows the status of changes in the working directory.
- git add <file>: Adds a file to the staging area.
- git commit -m "message": Commits the staged changes with a descriptive message.
- git push: Pushes the committed changes to a remote repository.
- git pull: Fetches and merges changes from a remote repository to your local repository.
- git branch: Lists, creates, or deletes branches.
- git checkout <branch>: Switches to a different branch.
- git merge <branch>: Merges another branch into the current branch.

Git Bash:

Git Bash is a command-line interface that provides a Unix-like environment for Windows users to use Git. It combines the Git command-line tool with a Bash shell, enabling users to run Git commands as well as Unix-based commands. This makes it

easier for developers accustomed to Unix-based systems to work on Windows.

Key Features of Git Bash:

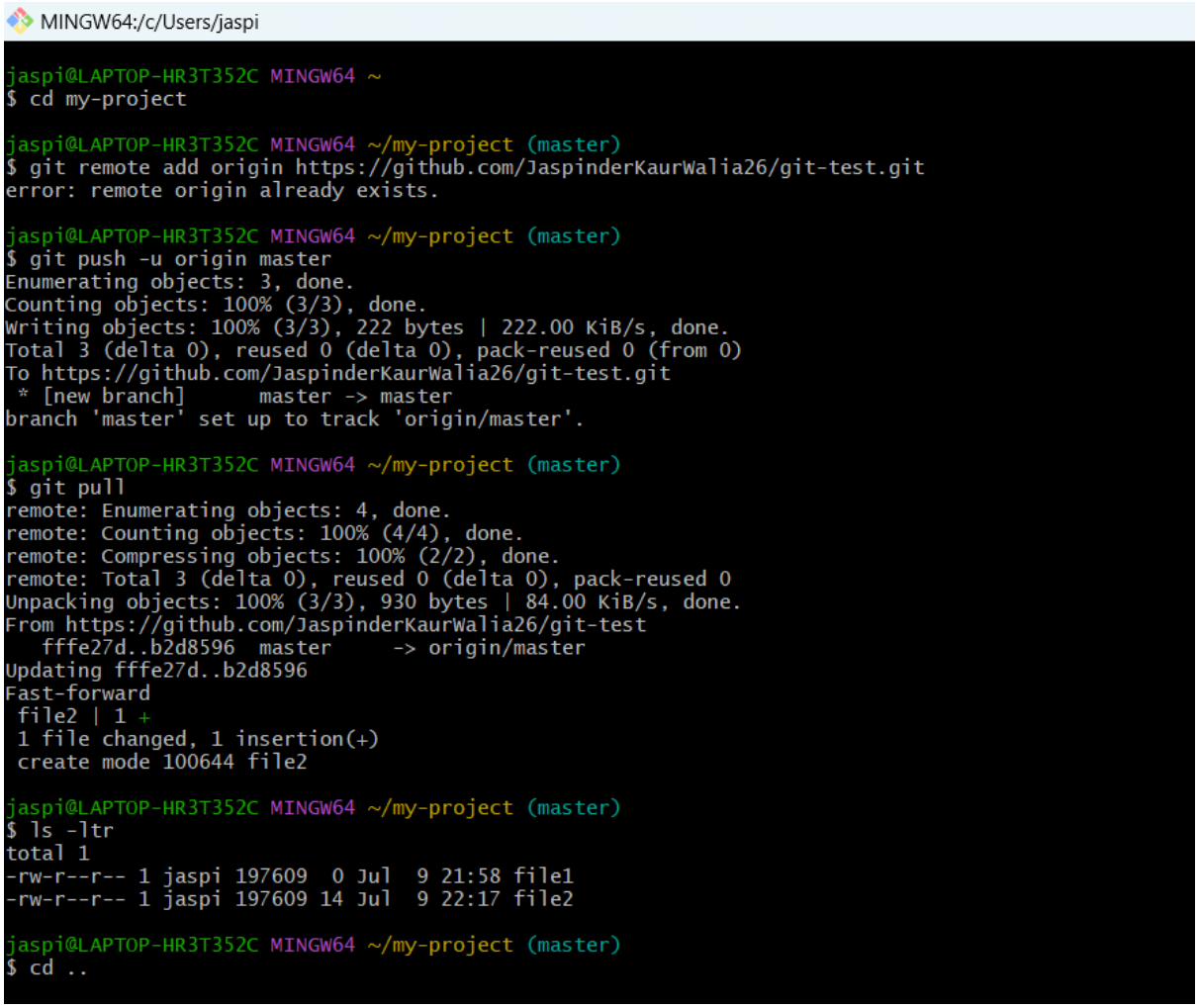
- Bash Shell: A powerful scripting environment commonly found on Unix-like operating systems.
- Git Commands: Full access to Git commands for version control.
- Unix Utilities: A collection of Unix tools like ssh, scp, cat, find, etc., enabling a consistent development environment across different operating systems.

Installing Git Bash:

- Download Git Bash:
  - Visit the official Git website.
  - Download the installer for your operating system.
- Install Git Bash:
  - Run the downloaded installer.
  - Follow the installation prompts, selecting the default options unless you have specific preferences.

Advantages of Using Git Bash:

- Consistent Environment: Provides a consistent development environment across different operating systems.
- Powerful Scripting: Allows for complex scripting with Bash, enhancing automation and efficiency.
- Comprehensive Toolset: Combines Git with Unix utilities, offering a comprehensive toolset for development.


MINGW64:/c/Users/jaspi
jaspi@LAPTOP-HR3T352C MINGW64 ~
$ cd my-project
jaspi@LAPTOP-HR3T352C MINGW64 ~/my-project (master)
$ git remote add origin https://github.com/JaspinderKaurWalia26/git-test.git
error: remote origin already exists.
jaspi@LAPTOP-HR3T352C MINGW64 ~/my-project (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 222 bytes | 222.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/JaspinderKaurWalia26/git-test.git
* [new branch] master -> master
branch 'master' set up to track 'origin/master'.
jaspi@LAPTOP-HR3T352C MINGW64 ~/my-project (master)
$ git pull
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
Unpacking objects: 100% (3/3), 930 bytes | 84.00 KiB/s, done.
From https://github.com/JaspinderKaurWalia26/git-test
fffe27d..b2d8596 master -> origin/master
Updating fffe27d..b2d8596
Fast-forward
file2 | 1 +
1 file changed, 1 insertion(+)
create mode 100644 file2
jaspi@LAPTOP-HR3T352C MINGW64 ~/my-project (master)
$ ls -ltr
total 1
-rw-r--r-- 1 jaspi 197609 0 Jul 9 21:58 file1
-rw-r--r-- 1 jaspi 197609 14 Jul 9 22:17 file2
jaspi@LAPTOP-HR3T352C MINGW64 ~/my-project (master)
$ cd ..