CS 240 Programming in C

Git and Version control

Dec 13, 2022

Content

- Introduction to version control and Git
- Setting up Git
- Git basics
  - Creating a new repository
  - Adding files to the repository
  - Committing changes
  - Viewing commit history
- Branching and merging
- Pull request

Version Control

version control is an essential tool for any software development team because it helps to ensure the integrity and reliability of the codebase. By keeping track of changes and allowing for easy recovery of previous versions, version control makes it possible for teams to collaborate and work on projects together without losing any important work.

- Easily revert files back to a previous state.
- Revert the entire project back to a previous state.
- Compare changes over time.
- See who last modified something that might be causing a problem.
- Protect and manage your work.
- Recover from mistakes or data loss.

Setting up Git

To set up Git on your Mac or Windows computer, you'll need to install the Git software on your computer. Here's how to do it:

- Open a web browser and go to the Git website at https://git-scm.com/.
- Download the latest version of Git for your operating system and start the installation process.
- Once the installation is complete, open a terminal window on your computer.
- In the terminal window, type the following command and press Enter: git –version
- This will display the version of Git that you have installed, which should be the latest version.

Congratulations, you have successfully installed Git on your computer! Now you can use Git to manage your code repositories and collaborate with other developers.

Git Basics

Git is a version control system that is commonly used for software development and other collaborative projects. A Git repository is a collection of files and directories that are tracked by Git, along with the version history of each file. Using Git, developers can work on the same codebase simultaneously, track changes made to the code, and roll back to previous versions if necessary. Git repositories can be hosted locally or on a remote server, and they can be accessed using the Git command line interface or a graphical user interface.

Branching and Merging

Branching and merging are two important concepts in Git that allow developers to work on multiple versions of a project at the same time and eventually merge those changes back into the main branch of the project.

- Branching in Git allows developers to create a new independent line of development, called a branch, based on an existing branch. This allows developers to work on new features or fixes without impacting the main branch of the project. Once the work on the new branch is complete, the changes can be merged back into the main branch.
- Merging in Git is the process of combining the changes from multiple branches into a single branch. This allows developers to integrate changes from different branches into the main branch, and ultimately create a new version of the project that includes all of the changes from the different branches.

Pull Request

- A pull request is a way to suggest changes to a project that is hosted on a code hosting platform, such as GitHub. To create a pull request, you will need to fork the project, make your changes on your local copy of the project, push those changes to your fork, and then submit a request to the original project maintainers asking them to review and merge your changes.
- Keep in mind that pull requests are typically reviewed by the project maintainers, who may suggest changes or improvements before merging the request. It's important to be responsive to feedback and willing to make any necessary changes.

Pull Request

```
## Description

Please include a summary of the change and which issue is fixed. Please
also include relevant motivation and context. List any dependencies that
are required for this change.

## Type of change

Please delete options that are not relevant.

- [ ] Bug fix (non-breaking change which fixes an issue)
- [ ] New feature (non-breaking change which adds functionality)
- [ ] Breaking change (fix or feature that would cause existing
functionality to not work as expected)
```

Pull Request

```
## How Has This Been Tested?

Please describe the tests that you ran to verify your changes. Provide
instructions so we can reproduce. Please also list any relevant details
for your test configuration

- [ ] Test A
- [ ] Test B

## Checklist:

- [ ] My code follows the style guidelines of this project
- [ ] I have performed a self-review of my own code
- [ ] I have commented my code, particularly in hard-to-understand areas
- [ ] I have made corresponding changes to the documentation
- [ ] My changes generate no new warnings
- [ ] I have added tests that prove my fix is effective or that my
feature works
- [ ] New and existing unit tests pass locally with my changes
```

Resource Links

- https://git-scm.com/
- https://docs.github.com/en/get-started/quickstart/hello-world