Git Lab

Madeline Stephens

January 30, 2017

1 Introduction

GitHub is a useful tool for remote collaboration and file back up. This lab provides a brief overview of the GitHub basics, including cloning repositories and setting git user information. Additionally, the process of using git bash to add files to the repository will be explained. This knowledge will be vital to ensure efficient and retrievable work throughout the semester.

2 Interface

In addition to the GitHub website, a local application called git bash was used to execute commands. This program facilitated communication between the computer and GitHub servers. When using Vivado to edit and simulate programs, git bash can be used to "send" the files to GitHub.

3 Implementation

The initial step taken to create a repository for lab work was to clone an existing repository. This was done by executing the **git clone [url]** command. In this case, the URL for the existing MIPS-lab repository was used. With this command, a copy of the existing repository was downloaded to the hard drive.

Next, it was necessary to set git user information to ensure that Git would be able to associate commits with a specific account or user. Using **git config –local user.name "first last"** the user name was set for the individual user. This allows users to verify who made changes to the repository, due to the fact that it adds a name tag to each upload. Also to that end, the user email was set using **git config –local user.email "your github email"**. This command establishes a email tag to changes for the same reasons.

When using Vivado to create or edit code, Git can become very handy. Git automatically tracks the changes made to the local repository. When working, it is always possible to view what changes have been made or which files have

been added by using the **git status** command. If desired, it is possible to add new files to the repository with the **git add "file name(s)"** or the **git add *** commands. The former allows the user to specify which individual files he or she would like committed, while the latter automatically adds all files that differ from the on-line repository.

Finally, the online repository can be updated to reflected the changes made. This can be done using the **git push -u origin "your branch name"** command. This pushes the changes made onto the user's branch in the on-line repository. This adds all of the files and changes to the GitHub website so that it can be accessed from anywhere.

4   Conclusions

With this knowledge, GitHub can be effectively used for efficient collaboration and file back up. By allowing users to access the repository from any device, Github provides flexibility. Additionally, GitHub makes it easy to track changes and who made them. By understanding how to make use of such a useful tool, users have the ability to work from anywhere.

Overview the main points you want to stick in peoples minds and answer key questions you want to stick in peoples minds. Did it work? How well? What would you have done differently? What did you learn?