

Linux and Git Guide

Git - Keeping Your Code Updated

Git is a software used for tracking changes in files and collaborative code development. GitHub is a provider of Internet hosting for software development and version control using Git.

Below is the recommended Git flow you should follow for assignment development.

This guide assumes you have a GitHub account with token based authentication. If not, please follow this tutorial to create an ssh key and link it to your GitHub account to perform Git actions.

1. First start off by cloning a git repo to your local machine.

```
git clone git@github.com:scpd-proed/<XCS-ASSIGNMENT-REPO>.git
```

2. Create a branch to do your own development

```
git checkout -b <branch-name>
```

3. If CFs make updates to the assignment commit your unsaved changes

```
git commit -am "<some_message>"
```

4. Merge updates from the assignment to your local branch

```
git pull origin master
```

After performing step 4, you may experience conflicts. To see which files are in conflict, please run

```
git status
```

The files in conflict are those "Not staged for commit". Open these files in the editor of your choice and look for lines <<<< and >>>>.

Once you have resolved the conflicts, run

```
git add -A .
```

And lastly commit your updates to your branch

```
git commit -am "<some_message>"
```

Please DO NOT push your local branch changes to GitHub or create pull requests. This will expose your solution code from your development branch and is in violation of the honor code.

Linux - Commands You Should Know

Linux is an open-source Unix-like operating system. All our assignments assume you have some understanding on how to run commands on Linux machines.

Below we will go through popular Linux Commands you should know for assignment development.

1. Listing directory (ls) - see files in your current directory

```
ls
```

2. Changing directory (cd) - move to another directory

```
cd /path/to/directory
```

3. Path Working Directory (pwd) - display pathname of current working directory

```
pwd
```

4. Display file contents (cat)

```
cat file.txt
```

5. Get a clear console window (clear)

```
clear
```

6. Copying files (cp)

```
cp source destination
```

7. Remove a file (rm)

```
rm file.txt
```

8. Zipping up files (zip)

```
zip myfile.zip filename.txt
```

9. Remote log into another machine (ssh) (will be useful for using Azure GPUs)

```
ssh user@machine
```

10. Rename or move a file (mv)

```
mv source destination
```