Todd Vorisek
CS 372
Homework 1
Feb 7, 2018

Git username: tmvorisek

Steps involved

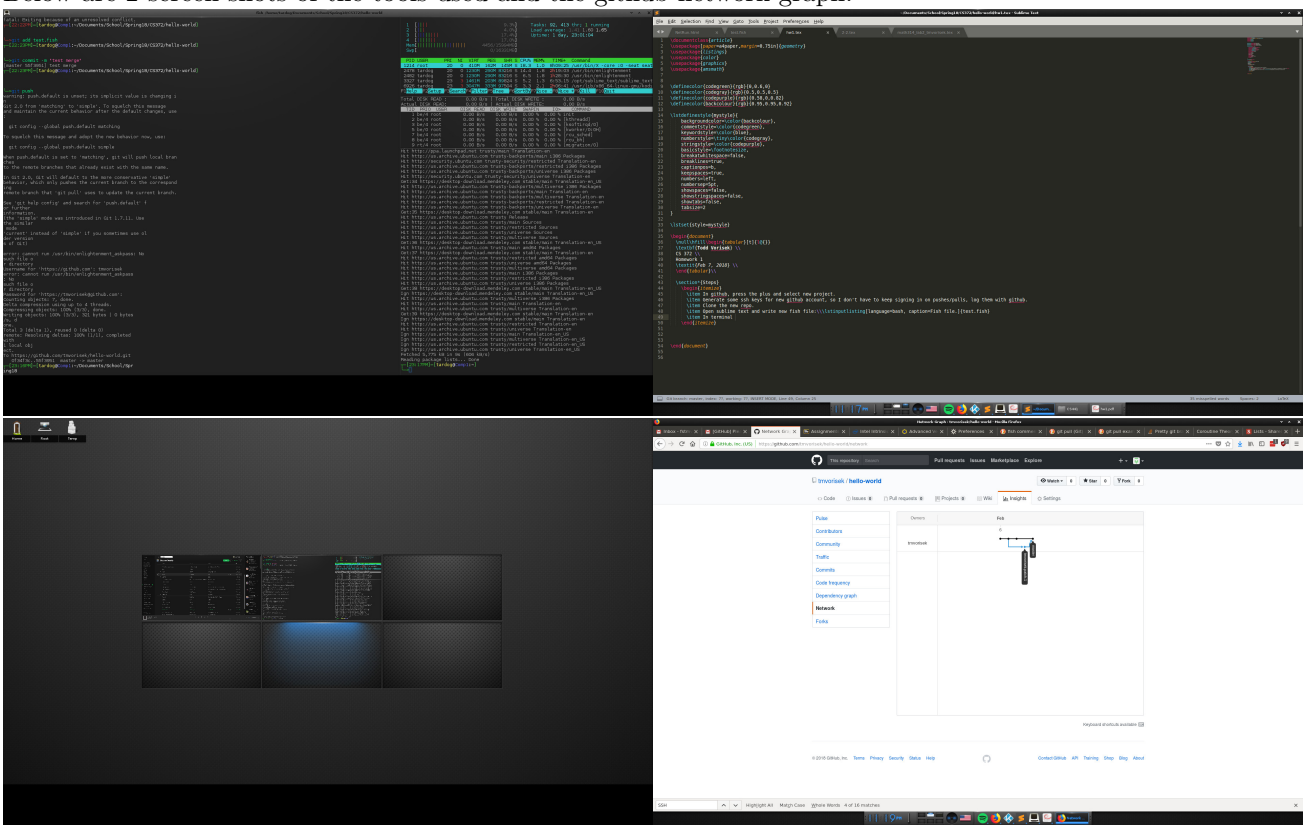
- In github, press the plus and select new project. Found at https://github.com/tmvorisek/hello-world
- Generate some ssh keys for new github account, so I don't have to keep signing in on pushes/pulls, log them with github.
- Clone the new repo.
- Open sublime text and write new fish file:

```
function hello_world #! Function to say hello world.
  echo "hello world"
end
```

Listing 1: Fish file.

- In a terminal add, commit and push the new file.
- In sublime text, make an edit to the new fish file, then add and commit the change.
- In github, make a different edit to the same line, then branch the change.
- In sublime text run merge, then do the grizzly business.
- Back in terminal push the merged file, and call it done.

Below are 2 screen shots of the tools used and the github network graph:



3 Git Commands

0.1 revert

```
1         git revert [options] <commit>
```

Get rid of one or more `<commit>`. `revert` effectively generates new commits that undo the changes present in the target commits. `revert` can also be used to alter an existing commits message, which I know I personally will probably find useful.

0.2 mv

```
1         git mv [-v] [-f] [-n] [-k] <source> <destination>
```

Moves a file or folder specified by `<source>` and stages those changes for commit, but the commits must still be performed my the user to add them to the current branch.

0.3 grep

```
1         git grep [options] [-e] <pattern> [<rev>...] [[--] <path>...]
```

Search for `<pattern>` in the tracked files of the current branch's work tree. Specify `-w` or `--word-regexp` to use regular expressions instead of simple pattern matching. Multiple text patterns can be specified by separating them with newline characters.