Ann Liberman
CISC 3140

3/19/2021

Lab #6

First step: Create a new repo: RepoMeta

Task 1: Decentralized VCS (Git)

- Start logging the session: `script screen.log`

- Create a new repo, we'll call this Repo1:
  Click "Create a New Repo" in Github, then input these commands in terminal:

```
996  echo "#Repo1" >> README.md
997  git init
998  git add README.md
999  git commit -m "first commit"
1000  git branch -M main
1001  git remote add origin https://github.com/annielibs/Repo1.git
1002  git push -u origin main
```

- Add a new file to be tracked by Repo1, i.e. HelloWorld.java:

```
1018  git add HelloWorld.java
1019  git remote add repo1 https://github.com/annielibs/Repo1
1020  git commit -m "first commit"
1021  git push -u repo1 master
```

- Edit file (HelloWorld.java) slightly, perhaps printing today's date:

Original HelloWorld.java

```
public class HelloWorld {
    public static void main (String [] args){
        System.out.println ("Hello, World");
    }
}
```

Edited HelloWorld.java

```
import java.time.format.DateTimeFormatter;
import java.time.LocalDateTime;
public class HelloWorld {
    public static void main (String [] args){
        System.out.println ("Hello, World");

        DateTimeFormatter dtf =
DateTimeFormatter.ofPattern("yyyy/MM/dd
HH:mm:ss");
        LocalDateTime now =
LocalDateTime.now();
            System.out.println
(dtf.format(now));
    }
}
```

- Commit the changes:
    - This is what I tried doing, but it kept giving me these 2 errors:
      error: src refspec master does not match any.
      error: failed to push some refs to 'https://github.com/annielibs/RepoMeta.git'

```
1061  git add HelloWorld.java
1062  git commit -m "Changed class name & added printing of current date"
1063  git push -u repo1 master
1064  git push -u origin master
1065  git commit -m "Changed class name & added printing of current date"
1066  git push origin master
```

- Write a descriptive commit message for your log:
  `example: "Changed class name & added printing of current date"`

Task 2: Centralized VCS (SVN)

- Make sure session is logging: `script screen.log`
- Create a new subversion repository, let's call it Repo2:

```
bash-4.2$ svnadmin create /var/svn/repos
svnadmin: E000002: Repository creation failed
svnadmin: E000002: Could not create top-level directory
svnadmin: E000002: Can't create directory '/var/svn/repos': No such file
or directory
```

- Add a new file to be tracked by Repo2:

```
I tried uploading a new file to be tracked by Repo2, but I kept getting
the following error:
bash-4.2$ svn add HelloWorld.java
svn: E155007: '/users1/st/aliberma/D:SVNRepo2' is not a working copy
```

- Edit that file slightly:
- Commit the changes:
- Save the log into RepoMeta as a text file:
- Delete Repo2 (there's no use for it anymore):
- Push your RepoMeta up to where you've hosted it to share your work:

Task 3: Submit with your log to your RepoMeta **a quick breakdown of the comparison.**

- How did you accomplish each task?
- What did you like or dislike about each tool? What are the strengths and weaknesses?
- What is similar between the two tools? What is different?
- What tasks are each better suited for?
- Which one do you prefer? Why?

Synopsis:

In general, for me this assignment was very difficult. Once I created the RepoMeta and Repo1 stumbled into an issue where I kept getting a detached Head error from checking out a

branch between a commit and push. This took me more than an hour to figure out and was very frustrating. In general, where else I stumbled, was how to commit the changes I made to my HelloWorld.java file so that I could see those changes in Github. When I tried doing that I ran into the SAME errors I kept getting previously, but I cannot figure out where exactly I am making my mistakes. After spending many hours on git, I decided to switch over to subversion, but had even less success there. Even though I understand what version control is and how it works from the big picture, I think it's difficult to understand what the various actions and commands are hoping to achieve in the various workflows. A lot of the vocabulary being used in the explanations and manuals is highly technical, so I feel it is easy to get lost in the reading / trying to understand. After spending about 5 hours on Task 1 and another 2 hours on Task 2, and having lots of help from classmates, I decided to give myself a break and work on something else so as to refresh.