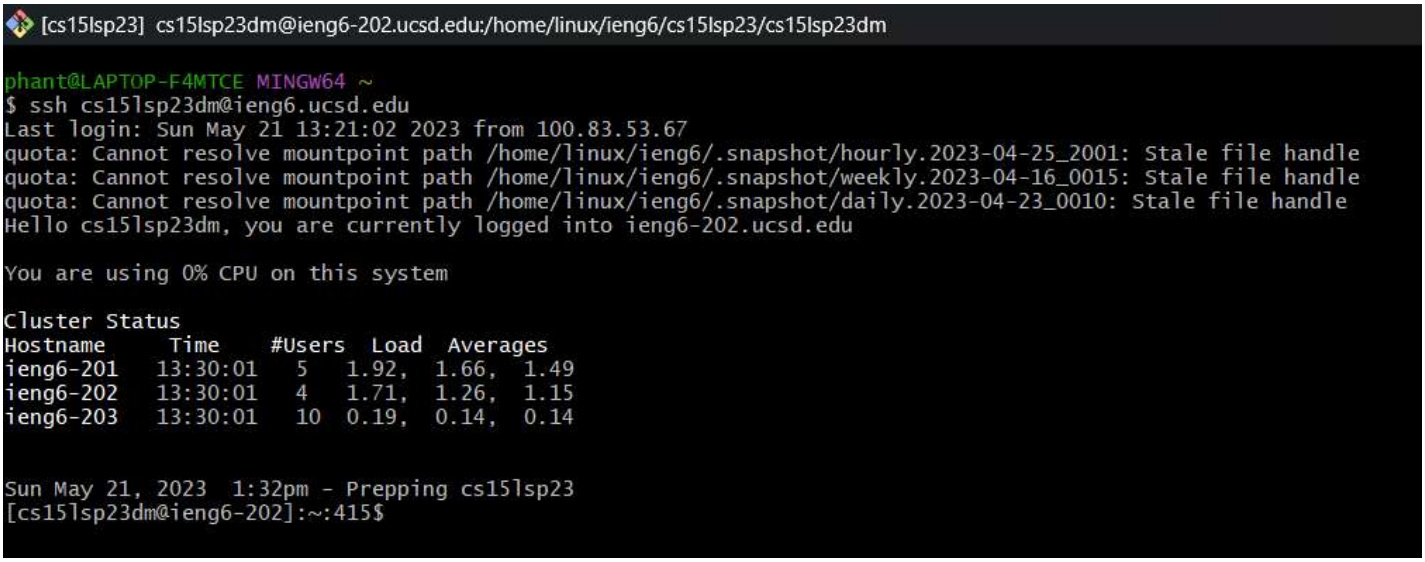
cse15l-lab-reports

Lab 4 - Doing it all in the terminal

Steps:

Step 4:

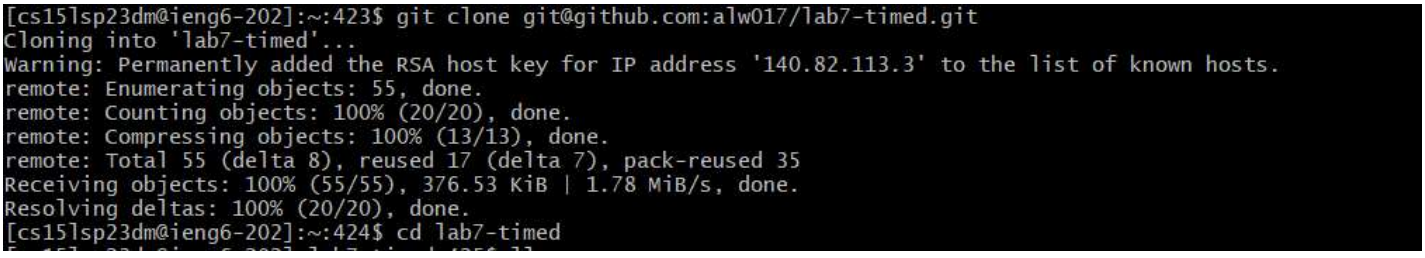


Keys Pressed:

`ssh <space> cs15lsp23dm@ieng6.ucsd.edu <enter>`

`ssh` logs me into my school-assigned terminal where I can remotely access the computer that I am connected to on the UCSD's servers. I didn't have to enter any password because I had already authenticated my personal PC's ssh key on the school's computer.

Step 5:



Keys Pressed:

`git clone <shift><insert> <enter>`

I had copied the link to the repository from github, and pasted it in the terminal to reduce the time I have to type.

The command `git clone` creates a copy of the repository and puts it in a folder with the name of the repository in the current working directory to edit.

Step 6:

```
[cs15lsp23dm@ieng6-202]:~:424$ cd lab7-timed
[cs15lsp23dm@ieng6-202]:lab7-timed:425$ ll
total 16
-rw-r----- 1 cs15lsp23dm ieng6_cs15lsp23 1435 May 21 13:40 ListExamples.java
-rw-r----- 1 cs15lsp23dm ieng6_cs15lsp23  747 May 21 13:40 ListExamplesTests.java
drwxr-s--- 2 cs15lsp23dm ieng6_cs15lsp23 4096 May 21 13:40 lib
-rw-r----- 1 cs15lsp23dm ieng6_cs15lsp23  169 May 21 13:40 test.sh
[cs15lsp23dm@ieng6-202]:lab7-timed:426$ vim test.sh
[cs15lsp23dm@ieng6-202]:lab7-timed:427$ bash test.sh
JUnit version 4.13.2
..E
Time: 0.539
There was 1 failure:
1) testMerge2(ListExamplesTests)
org.junit.runners.model.TestTimedOutException: test timed out after 500 milliseconds
        at ListExamples.merge(ListExamples.java:44)
        at ListExamplesTests.testMerge2(ListExamplesTests.java:19)

FAILURES!!!
Tests run: 2,  Failures: 1

[cs15lsp23dm@ieng6-202]:lab7-timed:428$
```

Keys Pressed:
`cd <space> lab7-timed <enter> ll <enter> vim <space> test.sh <enter> :q! <enter> bash <space> test.sh <enter>`

I changed directory into the repository I just cloned, and then ran `ll` which is like ls but automatically puts the files as if you ran ls with the `-l` option.

Then, to check the contents of `test.sh`, I use vim to read the file and see what it does. Afterwards, I quit without making changes using `:q!` and run `bash test.sh` which compiles the files and runs the tests on `ListExamples`.

Step 7:

```java
import java.util.ArrayList;
import java.util.List;

interface StringChecker { boolean checkString(String s); }

class ListExamples {

  // Returns a new list that has all the elements of the input list for which
  // the StringChecker returns true, and not the elements that return false, in
  // the same order they appeared in the input list;
  static List<String> filter(List<String> list, StringChecker sc) {
    List<String> result = new ArrayList<>();
    for(String s: list) {
      if(sc.checkString(s)) {
        result.add(0, s);
      }
    }
    return result;
  }


  // Takes two sorted list of strings (so "a" appears before "b" and so on),
  // and return a new list that has all the strings in both list in sorted order.
  static List<String> merge(List<String> list1, List<String> list2) {
    List<String> result = new ArrayList<>();
    int index1 = 0, index2 = 0;
    while(index1 < list1.size() && index2 < list2.size()) {
      if(list1.get(index1).compareTo(list2.get(index2)) < 0) {
        result.add(list1.get(index1));
        index1 += 1;
      }
      else {
        result.add(list2.get(index2));
        index2 += 1;
      }
    }
    while(index1 < list1.size()) {
      result.add(list1.get(index1));
      index1 += 1;
    }
    while(index2 < list2.size()) {
      result.add(list2.get(index2));
      // change index1 below to index2 to fix test
      index2 += 1;
    }
    return result;
  }


}
```

Keys Pressed:

`vim <space> List<tab>.java <enter> <shift>G ?index<enter> cw index2 <esc> :wq <enter>`

I run vim on the `ListExamples.java` file to edit it. Once inside vim, I used `<shift>G` to go to the end of the file, then `?index<enter>` to look backwards for the first occurence of index. From there, I use `cw` to delete `index1` and immediately enter insert mode to enter `index2` which is the current replacement. Afterwards, I exit insert mode with `<esc>` and save and quit with `:wq<enter>`. This fixes the bug causing the test failure.

Step 8:

```
[cs15lsp23dm@ieng6-202]:lab7-timed:429$ bash test.sh
JUnit version 4.13.2
..
Time: 0.02

OK (2 tests)
```

Keys Pressed: `<up> <up> <enter>`

This runs the `bash test.sh` command again, which compiles both java files and runs the tests `ListExamples` with the newly compiled classes.

Step 9:

```
[cs15lsp23dm@ieng6-202]:lab7-timed:430$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   ListExamples.java

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ListExamples.class
        ListExamplesTests.class
        StringChecker.class

no changes added to commit (use "git add" and/or "git commit -a")
[cs15lsp23dm@ieng6-202]:lab7-timed:431$ git add ListExamples.java
[cs15lsp23dm@ieng6-202]:lab7-timed:432$ git commit -m "fixed ListExamples bug"
[main be7e10d] fixed ListExamples bug
 Committer: Alexander Wang <cs15lsp23dm@ieng6-202.ucsd.edu>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+), 1 deletion(-)
[cs15lsp23dm@ieng6-202]:lab7-timed:433$ git push origin main
Warning: Permanently added the RSA host key for IP address '140.82.114.3' to the list of known hosts.
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 312 bytes | 156.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:alw017/lab7-timed.git
   503ed81..be7e10d  main -> main
[cs15lsp23dm@ieng6-202]:lab7-timed:434$
```

Keys Pressed: `git <space> status <enter> git <space> add <space> List<tab>.java <enter> git <space> commit <space> -m <space> "fixed ListExamples bug" <enter> git <space> push <space> origin <space> main <enter>`

These git status lists all of the changes I have made to the directory, listing any new files I've created and any modifications I have made to existing files in the directory.

From there, I use that information to add the `ListExamples.java` file, staging it before commiting. Afterwards, I use `git commit -m "fixed ListExamples bug"` to commit the added changes, with the `-m` option specifying a message of fixing the bug in ListExamples.java.

Finally, I use `git push origin main` to push the committed changes I made to the main branch, which now should appear as a new change made on GitHub.