cse15l-lab-reports

Lab 5 - Debugging Scenario

Original EdStem Post:

What environment are you using (computer, operating system, web browser, terminal/editor, and so on)?

UCSD Servers, ieng6-202, linux, using git bash.

Detail the symptom you're seeing. Be specific; include both what you're seeing and what you expected to see instead. Screenshots are great, copy-pasted terminal output is also great. Avoid saying "it doesn't work".

I'm expecting to see a message saying 1/2 tests passed, but instead I see an error message saying command javac not found, followed by a message that there was a compile error, and that the score was zero.

Related Output:

```
[cs15lsp23dm@ieng6-203]:cse15l-list-examples-grader:518$ bash grade.sh https://github.com/ucs
Cloning into 'student-submission'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
Finished cloning
grade.sh: line 30: javac: command not found
Compile Error! grade is 0
```

Detail the failure-inducing input and context. That might mean any or all of the command you're running, a test case, command-line arguments, working directory, even the last few commands you ran. Do your best to provide as much context as you can.

Command being run:

```
bash grade.sh https://github.com/ucsd-cse15l-f22/list-methods-lab3
```

Working Directory:

```
~/bash-scripts/cse15l-list-examples-grader
```

Code in grade.sh:

```
CPATH='.:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar'

rm -rf student-submission
rm -rf grading-area

mkdir grading-area

git clone $1 student-submission
echo 'Finished cloning'


# Draw a picture/take notes on the directory structure that's set up after
# getting to this point

# Then, add here code to compile and run, and do any post-processing of the
# tests

if [[ ! -f ./student-submission/ListExamples.java ]] # looks for ListExamples.java in student
then
        echo "file not found, grade is 0"
        exit
fi

cp -r student-submission grading-area
cp TestListExamples.java grading-area/student-submission

# path to folder ~/bash-scripts/cse15l-list-examples-grader/grading-area
PATH='grading-area/student-submission/*.java';

javac -cp $CPATH $PATH

if [[ $? -ne 0 ]]
then
        echo "Compile Error! grade is 0"
        exit
fi

java -cp '.:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar:grading-area/student-submission' c

if [[ $? -eq 0 ]]
then
        echo "grader passed, grade is 100"
else
```

```
        testspassed=$(grep -Po "Tests run: 2,  Failures: \K[0-9]+" output.txt)
        echo $testspassed "/2 tests passed"
fi
```

Response From TA

Try entering this command: `env` and look for a variable name which you might have defined in your script. `grep` might also be helpful here.

Information the student got:

I used `env > envoutput.txt` , and then used `grep -E "CPATH"` as well as `grep -E "PATH"` .

There was no output for `grep -E "CPATH"` .

Here is the output for `grep -E "PATH"`

```
MANPATH=/software/CSE/openjdk-19.0.1/man:/software/common/man:/software/common/mutt/share/mar
LD_LIBRARY_PATH=/software/CSE/openjdk-19.0.1/lib
PATH=/software/CSE/visualstudio/VSCode-linux-x64/bin:/software/CSE/openjdk-19.0.1/bin:/home/l
PULSE_CONFIG_PATH=/tmp/cs15lsp23dm-pulse
MODULEPATH=/home/linux/ieng6/cs15lsp23/public/modulefiles:/public/modulefiles:/public/Modules
PULSE_RUNTIME_PATH=/tmp/cs15lsp23dm-pulse
QT_PLUGIN_PATH=/usr/lib64/kde4/plugins:/usr/lib/kde4/plugins
PULSE_STATE_PATH=/tmp/cs15lsp23dm-pulse
```

I noticed that there was a line that says

```
PATH=/software/CSE/visualstudio/VSCode-linux-x64/bin:/software/CSE/openjdk-
19.0.1/bin:/home/linux/ieng6/cs15lsp23/public/bin:/usr/lib64/qt-
3.3/bin:/home/linux/ieng6/cs15lsp23/cs15lsp23dm/perl5/bin:/software/nonrdist64:/software/commo
n/bin:/software/common64/bin:/software/common/mutt/bin:/software/common/TeXLive/bin/i386-
linux:/software/common/AcroRead/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/bin:/so
ftware/common/pine/bin:/software/MATLAB/bin
```

This is the same variable name as the one I defined in my script.

Here is what's happening:

When running commands in the bash terminal, one critical variable is the `PATH` variable. This tells bash where to look for scripts and executable files, which allow you to run commands like `javac` and `echo`, etc.

When I defined the `$PATH` variable in the script `grade.sh`, it set the `PATH` environment variable to a different value, and now bash looks in the wrong directories for executable files. This is why you get the error message: `grade.sh: line 30: javac: command not found`

The message for a compile error naturally follows because bash exits the execution of a command with an error code since the command wasn't found.

File and Directory Structure Required:

- grade.sh
- TestListExamples.java
- grading-area/

Contents of Each File Required:

grade.sh

```
CPATH='.:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar'

rm -rf student-submission
rm -rf grading-area

mkdir grading-area

git clone $1 student-submission
echo 'Finished cloning'


# Draw a picture/take notes on the directory structure that's set up after
# getting to this point

# Then, add here code to compile and run, and do any post-processing of the
# tests

if [[ ! -f ./student-submission/ListExamples.java ]] # looks for ListExamples.java in student
then
        echo "file not found, grade is 0"
        exit
fi
```

```
cp -r student-submission grading-area
cp TestListExamples.java grading-area/student-submission

# path to folder ~/bash-scripts/cse15l-list-examples-grader/grading-area
PATH='grading-area/student-submission/*.java';

javac -cp $CPATH $PATH

if [[ $? -ne 0 ]]
then
        echo "Compile Error! grade is 0"
        exit
fi

java -cp '.:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar:grading-area/student-submission' c

if [[ $? -eq 0 ]]
then
        echo "grader passed, grade is 100"
else
        testspassed=$(grep -Po "Tests run: 2,  Failures: \K[0-9]+" output.txt)
        echo $testspassed "/2 tests passed"
fi
```

TestListExamples.java

```
import static org.junit.Assert.*;
import org.junit.*;
import java.util.Arrays;
import java.util.List;

class IsMoon implements StringChecker {
  public boolean checkString(String s) {
    return s.equalsIgnoreCase("moon");
  }
}

public class TestListExamples {
  @Test(timeout = 500)
  public void testMergeRightEnd() {
    List<String> left = Arrays.asList("a", "b", "c");
    List<String> right = Arrays.asList("a", "d");
    List<String> merged = ListExamples.merge(left, right);
    List<String> expected = Arrays.asList("a", "a", "b", "c", "d");
    assertEquals(expected, merged);
  }
  @Test(timeout = 500)
```

```
    public void testFilter() {
            List<String> left = Arrays.asList("a", "b", "d", "moon");
            List<String> list2 = Arrays.asList("a", "d");
            List<String> expected1 = Arrays.asList("moon");
            List<String> expected2 = Arrays.asList();
            StringChecker a = new IsMoon();
            assertEquals(expected1, ListExamples.filter(left, a));
            assertEquals(expected2, ListExamples.filter(list2, a));
    }
  }
```

Lines Ran to Trigger Bug:

```
bash grade.sh https://github.com/ucsd-cse15l-f22/list-methods-lab3
```

What to Edit to Fix the Bug:

Rename the `PATH` variable in `grade.sh` to anything other than `PATH`. In my case, I changed it to `SUBMISSION_PATH`. This will fix the issue of getting the `command javac not found` error.