Guidelines for Moss

Plagiarism Check for Programs

http://theory.stanford.edu/~aiken/moss/

Jianlin Zhang
(jianlin1992@gmail.com)
Feb 13, 2016

*Note: this guide is Unix-based. Errors might happen under Windows environment.

Step 1 - Prepare Files Accordingly

(The working directory here is where all files and directories are located, including `moss.pl`, `solution_directory`, `base_directory`, `solutions`*, and potential archived result from Moss)*

- Download all students' codes from blackboard and unzip them under "`solutions`" folder.
  Note: under solutions, there must be *directly* zip files of submissions of each student with no archive folder.
- Execute copyFile.sh like:
  ./copyFile.sh solutions
  (Note: After execution, solutions folder should contain folders named like "Project1Task1AndrewID". However, there are students who don't name their Netbeans project zips as required. We need to manually check any outliers.)
- Create a folder named "`solutions_directory`".
- Copy students' submission folders to `solutions_directory` task by task. For example, sort folders by name and copy all folders named "Project1Task1AndrewID".
- A valid directory containing all students' solutions should look like this:

```
solution_directory
|- Project1Task1student1
   |- classA.java
   |- ...
|- Project1Task1student2
   |- ...
|_ Project1Task1student3
   |- ...
```

- Create another folder called "`base_directory`". If there are sample codes provided by the faculties, add source codes into this directory. (Note: base files need be served one by one as parameters in moss.pl).

Step 2 – Execute the Submission Program

Execute moss.pl given parameters: (Please read the instructions in moss.pl for more details)

```
./ moss [-l language] [-d] [-b basefile1] ... [-b basefilen] [-m #] [-c "string"] file1
file2 file3
```

For example:

```
./moss –l java –d –b base_directory/file1.java –b
base_directory/file2.java –c "Project1"
solutions_directory/*/*
```

(if you can't execute it, execute: `chmod 777 moss.pl`)

Wait until the program provides a url for results. And this might take a short while like one or two minutes.

Step 3 – Fetch Results from Moss

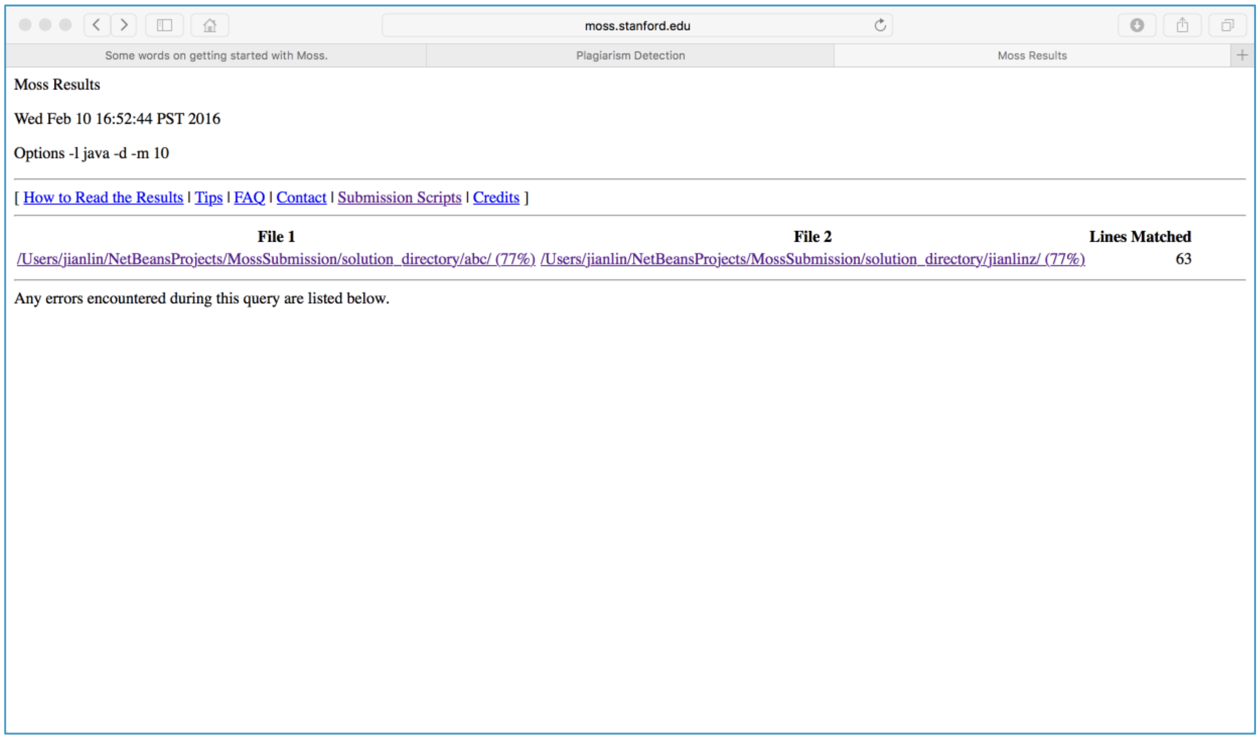
Moss platform is deployed as an online service, we don't have access directly to the system, but we can access the result given by Moss.

- Option 1: View on browser
  `http://moss.stanford.edu/results/result#`
- Option 2: download for local archives
  `wget –r –np http://moss.stanford.edu/results/result#`

Step 4 – Check Results

In the webpage, we will get results given by Moss. The results are *somewhat* sorted by percentages that two files have in common. We need to MANUALLY go into top-rated files before making any conclusion about academic plagiarism!

Screenshots:



Moss Results

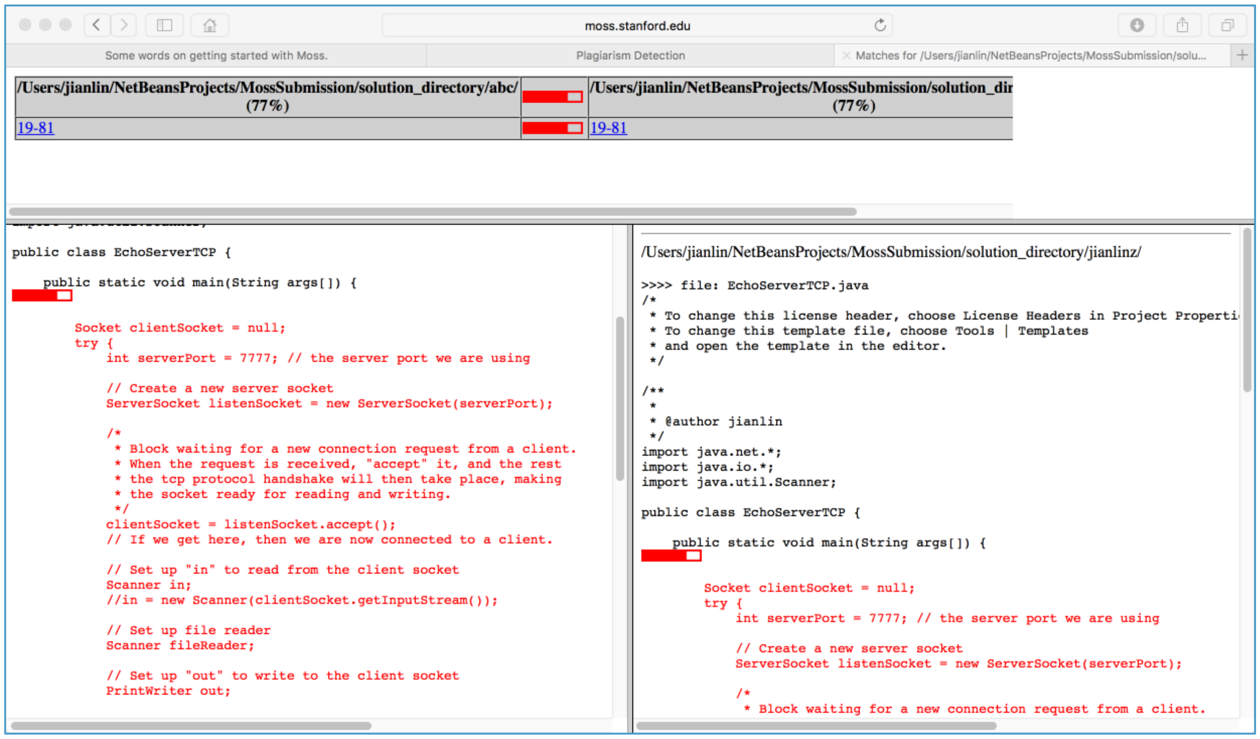
Wed Feb 10 16:52:44 PST 2016

Options -l java -d -m 10

[How to Read the Results | Tips | FAQ | Contact | Submission Scripts | Credits]

File 1	File 2	Lines Matched
/Users/jianlin/NetBeansProjects/MossSubmission/solution_directory/abc/ (77%)	/Users/jianlin/NetBeansProjects/MossSubmission/solution_directory/jianlinz/ (77%)	63

Any errors encountered during this query are listed below.


moss.stanford.edu
Some words on getting started with Moss.
Plagiarism Detection
Matches for /Users/jianlin/NetBeansProjects/MossSubmission/solu...

/Users/jianlin/NetBeansProjects/MossSubmission/solution_directory/abc/ (77%)		/Users/jianlin/NetBeansProjects/MossSubmission/solution_dir (77%)
19-81	🟥	19-81

```
public class EchoServerTCP {

    public static void main(String args[]) {


        Socket clientSocket = null;
        try {
            int serverPort = 7777; // the server port we are using

            // Create a new server socket
            ServerSocket listenSocket = new ServerSocket(serverPort);

            /*
             * Block waiting for a new connection request from a client.
             * When the request is received, "accept" it, and the rest
             * the tcp protocol handshake will then take place, making
             * the socket ready for reading and writing.
             */
            clientSocket = listenSocket.accept();
            // If we get here, then we are now connected to a client.

            // Set up "in" to read from the client socket
            Scanner in;
            //in = new Scanner(clientSocket.getInputStream());

            // Set up file reader
            Scanner fileReader;

            // Set up "out" to write to the client socket
            PrintWriter out;
```

/Users/jianlin/NetBeansProjects/MossSubmission/solution_directory/jianlinz/

```
>>>> file: EchoServerTCP.java
/*
 * To change this license header, choose License Headers in Project Properti
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author jianlin
 */
import java.net.*;
import java.io.*;
import java.util.Scanner;

public class EchoServerTCP {

    public static void main(String args[]) {


        Socket clientSocket = null;
        try {
            int serverPort = 7777; // the server port we are using

            // Create a new server socket
            ServerSocket listenSocket = new ServerSocket(serverPort);

            /*
             * Block waiting for a new connection request from a client.
```