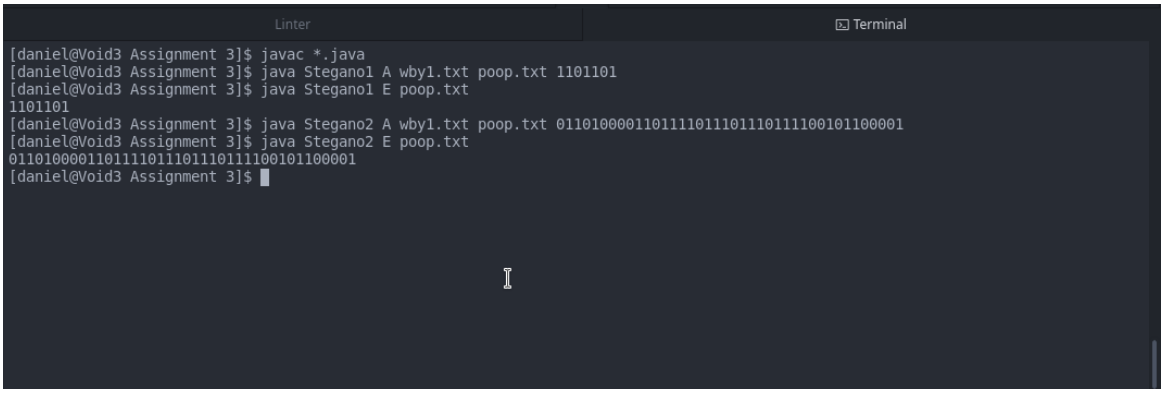
Assignment 4 - CT255 - Cybersecurity

Daniel Hannon (19484286)

December 2020

1 Terminal Output



2 Java Code

2.1 Stegano1

```java
/**
 * Skeleton code for Steganography assignment.
 *
 * @author Michael Schukat
 * @version 1.0
 */

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class Stegano1
{
   /**
    * Constructor for objects of class Stegano1
    */
   public Stegano1()
   {
   }

  public static void main(String[] args) {
    String arg1, arg2, arg3, arg4;
    Boolean err = false;

    if (args != null && args.length > 1) { // Check for minimum number of arguments
      arg1 = args[0];
      arg2 = args[1];
      if (arg2 == "") {
        err = true;
      }
      else if ((arg1.charAt(0) == 65) && (args.length > 3)) {
```

```
        // Get other arguments
        arg3 = args[2];
        arg4 = args[3];
        if (arg3 == "" || arg4 == "") {
          err = true;
        }
        else {
          // Hide bitstring
          hide(arg2, arg3, arg4);
        }
      }
      else if (arg1.charAt(0) == 69){
        // Extract bitstring from text
        retrieve(arg2);
      }
      else {
        err = true;
      }
    }
    else {
      err = true;
    }

    if (err == true) {
      System.out.println();
      System.out.println("Use: Stegano1 <A:E><Input File><OutputFile><Bitstring>");
      System.out.println("Example: Stegano1 A inp.txt out.txt 0010101");
      System.out.println("Example: Stegano1 E inp.txt");

    }
  }

  static void hide(String inpFile, String outFile, String binString) {
    //
    BufferedReader reader;
    BufferedWriter writer;
    int strLength = binString.length();
    int stringPos = 0;
    try {
      reader = new BufferedReader(new FileReader(inpFile));
      writer = new BufferedWriter(new FileWriter(outFile));
      String line = reader.readLine();
      while (line != null) {
        // Your code starts here
        if(stringPos < strLength) {
          /*Make Sure it does not try to access an illegal place in memory */
          if(binString.charAt(stringPos) == 48) { /*0*/
            line += " ";
          }
          if(binString.charAt(stringPos) == 49) { /*1*/
            line += "  ";
          }
        }
        stringPos++;
        // Store amended line in output file
        writer.write(line);
        writer.newLine();
        // read next line
        line = reader.readLine();
      }
```

```
      reader.close();
      writer.close();
    } catch (IOException e) {
  e.printStackTrace();
    }

  }

  static void retrieve(String inpFile) {
    BufferedReader reader;
    String output = "";
    try {
      reader = new BufferedReader(new FileReader(inpFile));
      String line = reader.readLine();
      while (line != null) {
        if(line.length() != 0) { /*Edge Case */
          if(line.charAt((line.length() - 1)) == 32) {/*Check for whitespace*/
            if(line.length() >= 2 && line.charAt((line.length() - 2)) == 32) {
            → //Another Edge Case and check for more whitespace
              output += "1";
            } else {
              output += "0";
            }
          }
        }
    // read next line
    line = reader.readLine();
      }
      System.out.println(output);
      reader.close();
    } catch (IOException e) {
      e.printStackTrace();
    }
  }
}
```

2.2 Stegano2

```
/**
 * Skeleton code for Steganography assignment.
 *
 * @author Michael Schukat
 * @version 1.0
 */

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class Stegano2
{
  /**
   * Constructor for objects of class Stegano2
   */
   public Stegano2()
```

```
  {
  }

  public static void main(String[] args) {
    String arg1, arg2, arg3, arg4;
    Boolean err = false;

    if (args != null && args.length > 1) { // Check for minimum number of arguments
      arg1 = args[0];
      arg2 = args[1];
      if (arg2 == "") {
        err = true;
      }
      else if ((arg1.charAt(0) == 65) && (args.length > 3)) {
        // Get other arguments
        arg3 = args[2];
        arg4 = args[3];
        if (arg3 == "" || arg4 == "") {
          err = true;
        }
        else {
          // Hide bitstring
          hide(arg2, arg3, arg4);
        }
      }
      else if (arg1.charAt(0) == 69){
        // Extract bitstring from text
        retrieve(arg2);
      }
      else {
        err = true;
      }
    }
    else {
      err = true;
    }

    if (err == true) {
      System.out.println();
      System.out.println("Use: Stegano1 <A:E><Input File><OutputFile><Bitstring>");
      System.out.println("Example: Stegano1 A inp.txt out.txt 0010101");
      System.out.println("Example: Stegano1 E inp.txt");

    }
  }

  static void hide(String inpFile, String outFile, String binString) {
    //Perform this check immediately
    if(binString.length() % 2 != 0) {
      /*Check if the binary string has an odd length, appends an extra digit */
      binString+="0";
    }
    BufferedReader reader;
    BufferedWriter writer;
    int strLength = binString.length();
    int stringPos = 0;
    int i = 0;
    try {
      reader = new BufferedReader(new FileReader(inpFile));
      writer = new BufferedWriter(new FileWriter(outFile));
```

```
      String line = reader.readLine();
      while (line != null) {
        for(i = 0; i < 2; i++) {
          if(stringPos < strLength) {
            if(binString.charAt(stringPos) == 48) { /*0*/
              line += " ";
            }
            if(binString.charAt(stringPos) == 49) { /*1*/
              line += "\t";
            }
          }
          stringPos++;
        }
        // Store amended line in output file
        writer.write(line);
        writer.newLine();
        // read next line
        line = reader.readLine();
      }
      reader.close();
      writer.close();
    } catch (IOException e) {
  e.printStackTrace();
    }

  }

  static void retrieve(String inpFile) {
    BufferedReader reader;
    String output = "";
    int i = 2;
    try {
      reader = new BufferedReader(new FileReader(inpFile));
      String line = reader.readLine();
      while (line != null) {
        if(line.length() >= 2) { /*Make sure there's a point in reading the line*/
          for(i = 2; i > 0; i--) {
            if(line.charAt(line.length() - i) == 32) { /*Space*/
              output += "0";
            }
            if(line.charAt(line.length() - i) == 9) { /*Tab*/
              output += "1";
            }
          }
        }
    // read next line
    line = reader.readLine();
      }
      System.out.println(output);
      reader.close();
    } catch (IOException e) {
      e.printStackTrace();
    }
  }
}
```