

CIS29 Take-Home Final: Design a Lab

Author: Wolfgang C. Strack

24 June 2015

Lab # (for Instructor):

This lab is intended to come after the lab that builds an XML parser. Using this quarter (Spring 2015) as a reference, this lab ideally would be an alternative Lab 3.

Purpose:

Use threading techniques

Background:

You have been tasked by the government to break into a corrupt private banking firm's user database. This bank caters to criminals and is infamous for secretly providing monetary oversight of illegal transactions. The reason why you are breaking into their user database is to track down their clients, some of whom have committed serious crimes. You are working with other government programmers to complete this mission. You, in particular, are in charge of designing a program that can process the bank's user data as quickly as possible so that your program will not be detected.

Description:

You will be designing a multi-threaded XML data parser, because the bank database stores parts of the encrypted user data in XML. You will also be in charge of taking this encrypted data and decrypting it to uncover the identities of this infamous bank's clients.

Fortunately, the bank's database designers were not very bright so data has leaked involving the bank's database security design. From these leaks, your employer tells you that the XML data is stored in

separate files and he also tells you how they are formatted. Finally, he also tells you the encryption/decryption algorithm, which was also somehow leaked...

Since you want your program to do its job as quickly as possible, you will use threading to parse multiple files at the same time. You will then store the parsed XML nodes into one list. Since you cannot copy or download this data (and risk getting detected), you will also design your program to decrypt the user data immediately after parsing and storing it into the node list.

Instructions:

Your employer has given you four encrypted sample files to work with that contain XML nodes, each node holding a username, a last name, and a first name. Design a class (or classes) that use(s) your XML parser from the previous lab, but adds multi-threading functionality to parse multiple files at the same time. Make sure your design can handle MORE THAN four files at once, even though you're only working with four sample files.

XML Data and Encryption Algorithm:

Here is what one **encrypted** user node looks like in an XML file:

```
<user>
  <username>aliquet@tellusfaucibusleo.com</username>
  <last>'ùEê</last>
  <first>%öDý@</first>
</user>
```

Notice that only the last/first names are encrypted.

Here is that same XML node but **decrypted**:

```
<user>
  <username>aliquet@tellusfaucibusleo.com</username>
  <last>Huff</last>
  <first>Freya</first>
</user>
```

The encryption/decryption algorithm is very simple and is used to encrypt/decrypt both the first and last names. Here is some pseudo code to **ENCRYPT** the first or last name:

```
string name = last or first name
for each character at index in name
    if index is even
        name[index] -= 33
    else if index is odd
        name[index] += 132
end-for
```

Again, this pseudo code is for ENCRYPTING a first/last name. Figure out the decryption algorithm using this pseudo code (Hint: Look up character ASCII codes).

Steps (summary):

1. Design a class (or classes) that use(s) your XML parser from the previous lab and adds multi-threading functionality to parse multiple files at the same time. Parse the sample encrypted XML files provided in the Lab-Data folder. The equivalent decrypted XML files are also provided if you want to check your work and/or test out the encryption algorithm pseudo code above.
2. After parsing the data and storing it all in a node list, decrypt the data stored in the nodes named <first> and <last>.
3. Print the decrypted XML nodes to the screen.

Instructor's Notes:

- The decrypted XML files were generated with random data here: <http://generatedata.com/>
- WARNING: some editors (Visual Studio included) can overwrite the encrypted XML data files when opening them. Visual Studio will give a warning that some characters in the file could not be displayed; it is okay to open these files in Visual Studio AS LONG AS YOU DO NOT PRESS SAVE after Visual Studio has given the warning.