INITIAL PLAN

COBE

Correction of Bad Extensions

Author: Borislav A. Ivanov
Supervisor: Michael Daley
Moderator: Paul Rosin
Module Code: CM3203
Module Name: One Semester Individual Project
Credits: 40
Year: 2015/2016

Project Description

The purpose of this project is to develop a software with a GUI that will allow a user to examine files that have bad extensions by extracting the "magic number" and comparing it to the current file extension.

Currently there are programs out there which can identify a file independent of its file extension. To name a few: “FTK – Forensic Tool Kit” [1], “file” [2] and “Foremost” [3].

In Foremost you have to specify what files you looking for, so it's a carving tool and you have to then chose the next type of file you want to carve out and so forth. It does have the ability to extract multiple file types in one command, but a person will have to know what they are looking for in advance and they will have to have some knowledge of Linux commands in order to be able to work with the program. Also foremost does not preserve the metadata of each file it carves out.

The situation is the same with “file”, in terms of needing to have Linux knowledge in order to use it. “file” recognizes the file type, but it does not extract the files to their own category nor does it extract them at all. Also “file” cannot distinguish between similar file types such as C and C++.

Finally, the FTK tool is a tool that has many options and not very intuitive and does not change the file extensions of files it has recognized to their original intended extensions. Also it is very expensive, making it not very accessible and it can only run on a Windows based machine, so the limitations are many.

The name of the software, I plan to develop, shall be COBE – Correction of Bad Extensions. COBE shall offer the user to extract each file with bad file extension and convert it to its original file extension, this will allow the user to view the file as it was meant to be, that is if he has the appropriate program that opens the specific file to be opened. COBE is different from other software on the market as it will have a very intuitive GUI, thus eliminating the need for the user to have Linux command skills for example. Also COBE shall run on any OS that has the corresponding JVM (Java Virtual Machine) [4] installed. The user will also have the opportunity to extract one file at a time or as many files as the user wants to select and COBE will output files to their own category (word files will go to a documents category folder, jpeg and other image types shall be extracted to an images folder, etc.).

Project Aims and Objectives

Compulsory

- Should be able to run on any computer OS that has JVM (Java Virtual Machine) installed
- A GUI that is intuitive and easy to use
- Ability to add more file extensions, through the interface, to be recognized by the software
- Each extracted file should be organized to folders according to file type for easy search and find
- Ability to load more than one file at a time and distinguish which ones have bad extensions
- Ability for the user to select the file types they are looking for, for example:
  - A user may want to extract only text files, or only pictures
  - COBE should allow the user to select which categories they want to deal with
- Java based - cross platform
- COBE will provide the user with information on each file type found such as:
  - What program can be used to open the file
  - What extension it had before being converted to its intended file extension
- MD5 hash [5] integrity verification for the corrected bad extension files, so they can be used as evidence in the court of law
- Initially it should support 20 file signatures
- Log in – which should track users' activity such as:
  - when they logged in, for how long, and who logged in
  - this also means that a master account shall be set up which will have more privileges and options available to the user than a normal account
    - the master account shall be the only account able to add more accounts to the system

Optional (if there is time left)

- Ability to distinguish between similar file types such as:
  - C
  - C++
  - Java
  - Etc.
- Ability to distinguish between files by not only looking at the magic number, but by reading the whole file
- In terms of tracking users' activity an optional, but may be very important feature shall be added if there is time. This feature shall track the users' actual activity. What is meant by this is that as soon as a user logs in, the system will start a log of the user interaction with the system, which means that every step the user takes whilst using the software shall be recorded in a file. The file shall be encrypted and only the main account can view it via their log in details. This will ensure that the user who used the software did not make any changes to the file in case they did something they should have not.

Work Plan

WEEK 1 (25TH JANUARY)

- Forming Initial Plan
  - ***MILESTONE: Submit Initial Report***

WEEK 2 (1ST FEBRUARY)

- Do more research to identify any gaps
- Identify what technologies specifically to use
- Conduct requirements analysis
- Database Design
- Model the system in UML
  - ***MEETING: Supervisor Meeting to discuss current progress***

WEEK 3 (8TH FEBRUARY)

- Begin the design of the User Interface on paper
- Begin implementing the part of the system that will look at the magic number of a file and correct it in Java
- Begin implementing the MD5 hashing for files that are to be corrected in Java
  - ***MEETING: Supervisor Meeting to discuss current progress***

WEEK 4 (15TH FEBRUARY)

- Continue implementing parts of the system which are compulsory for the correct operation of the software
  - ***MEETING: Supervisor Meeting to discuss current progress***

WEEK 5 (22ND FEBRUARY) – WEEK 8 (14TH MARCH)

- This time will be used to complete a prototype of the software including the implementation of the User Interface in Java
- User testing
  - ***MILESTONE: Have a working prototype***
  - ***MEETING: Supervisor Meeting to discuss current progress each week if possible***

EASTER RECESS (19th MARCH 10th APRIL)

- The Easter recess will be used to catch up with any part of the development process that might have fallen behind

WEEK 9 (11TH APRIL)

- Reflect on where the software is at this moment in time and set further targets based on current progress
- User testing and adjusting according to faults found whilst user testing
  - ***MEETING: Supervisor Meeting to discuss current progress***
  - ***MILESTONE: Have a final version of the software***

WEEK 10 (18TH APRIL) & WEEK 11 (25TH APRIL)

- Test the final version of the software and adjust as problems are found
  - ***MEETING: Supervisor Meeting to discuss current progress***

WEEK 12 (2ND MAY)

- ***MILESTONE: Hand in Final Report and Completed Software***

Reference:

[1] FTK - Forensic Tool Kit https://en.wikipedia.org/wiki/Forensic_Toolkit

[accessed: 20.01.2016]

[2] "file" - http://linux.die.net/man/1/file

[accessed: 17.01.2016]

[3] Foremost - https://en.wikipedia.org/wiki/Foremost_(software)

[accessed: 18.01.2016]

[4] JVM (Java Virtual Machine) - https://en.wikipedia.org/wiki/Java_virtual_machine

[accessed: 23.01.2016]

[5] MD5 Hashing - https://en.wikipedia.org/wiki/MD5

[accessed: 23.01.2016]