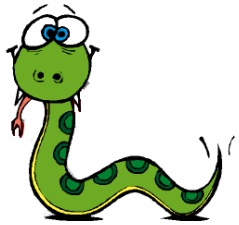
Miniprojekt ”forensic tool”



Sista laborationen är ett miniprojekt där ni arbetar i grupp om 3-4 studenter. I projektet skall gruppen göra ett program där gruppen skall använda alla kunskaper som har inhämtats under kursens gång. En kortfattat specifikation för programmet finns i slutet av dokumentet.

Projektgrupper

Ni får gärna jobba i samma grupp som ni har haft under kursen. Om du inte har gjort mist 4 av 5 laborationerna får du inte göra miniprojektet. Du måste göra laborationerna först. Du får höra av dig till mig för en plan.

Redovisning

Ni skall redovisa genom att skriva en kort labbrapport där ni beskriver ert program. Rapporten utan källkod skall vara 2-3 sidor inte mer. Koden och labbrapporten skall packas ihop och laddas upp på blackboard senast 9/3.
Bara en gruppmedlem skall ladda upp rapporten därför är det viktigt att studenternas namn skall stå tydligt på första sidan av rapporten.
Miniprojekten skall också redovisas muntlig med en redovisningsseminare 9/12.

Rapporten skall innehålla:

1. Namn på studenterna som har bidragit till projektet.
2. Backgrund: Kort beskrivning av uppgiften och varför det verkar intressant/relevant för er.
3. Kravspecifikation: En lista med vilka funktioner som uppgiften skall lösa.
4. Genomförande: Beskriv hur ni har delat arbetet, hur ni har jobbat, vilket material(websidor med tutorials,

programmeringsforum, mm) som ni har tagit inspiration från för att lösa uppgiften.
5. Systembeskrivning- Flödeschema som visar en översikt över lösningen samt en kort motivering av det.
6. Test, förklara hur ni har testat varje funktion och det slutliga programmet.
7. Testresultat- Skriv en lista med eventuella runtime-error och för vilka indata dessa error visar sig. Om inga error då skriver ni det.
8. Diskussion-

Ett välgjort miniprojekt och rapport **kan höja betyget** på kursen om det händer att ni ligger mellan två olika betyg.

Problembeskrivning

Uppgiften handlar om att implementera ett paket med funktioner som kan hjälpa dig söka efter information i en dator, kryptera filer, mm.

Det är mycket information du kan extrahera från en hårddisk. Hur mycket, beror på din nyfikenhet och ambition.

Det viktigaste är att du kommer att lära dig mer programmering och har roligt under tiden.

En del av dessa funktioner/metoder har du helt eller delvis redan implementerad i tidigare laborationer.

Följande är exempel på funktioner som ditt forensiska verktyg bör erbjuda:

- En funktion som hittar alla filer i en katalog (eller på hela hårddisken). Inputen till funktionen är en path till en viss katalog man vill lista ut, t.ex. ”C:/nina/java”. Funktionen skall skapa en tupler eller lista med alla filer som hittas. Tänk på att du måste gå ner i alla underkataloger som finns i respektive katalog mm.
- En funktion som kan hitta och lista ut alla filer av en viss typ (alltså filerna som har en viss ändelse pdf, txt, doc, ljudfiler, bildfiler, mm). Indata till funktionen är just filändelsen och mappen du skall leta i. Utdata är en lista med alla filer med just den filändelsen.

- En funktion som söker efter en viss information i en viss typ av fil. T.ex. namn, personnummer, mm i textfiler. Funktionen returnerar alla filnamn där filerna innehåller respektive information.
- För vissa typer av filer är det svårt att läsa innehållet. Ta reda på om det inte finns program som konverterar dessa filer så att du sedan kan hitta information. T.ex. pdf filer kan konverteras till ps innan läsning.
- En funktion som söker efter modifierade filer senast ett viss datum
- En funktion som kryptetar filer. Du skall kunna kryptera och dekryptera filer. För denna kryptering skall du använda datastrukturen Dictionary(kap 9) på följande sätt:
  -skapa en dictionary där du associerar koder till varje bokstav, dvs, en bokstav ersätts med en annan bokstav, siffra eller tecken, t.ex
  codes= {'A': 'p', 'B': '7', 'C': '#' …… 'a': 'B' , ….m}
  Med detta exempel bokstaven " A" kommer att ersättas vid kryptering med "p" osv.
- Filanalys. Datafiler är oftast mycket stora. Många gånger i forensiskt arbete vill man kunna se skillnader mellan innehållet i två filer eller hitta en fil som är närmast likt någon annan fil som man har som undersökningsfil. I ditt program skall du kunna jämföra två olika filer för att se skillnader i innehåll. För detta skall du använda datastrukturen Set (kap 9) och operationerna som erbjuds.
- Outputen kan vara t.ex en lista med orden som inte i filen A men inte i B samt orden som finns i filen B men inte i A. Du kan också visualisera skillnaden på annat sätt.

Programmet skall kunna användas med ett GUI samt ett textuellt interface (meny). Man skall kunna starta programmet med GUI eller textuell interface genom att skicka olika argument till programmet till exempel

>>foresnik.py txt

>>foresnik.py gui

Extra:

HashSet

- Hashset är något som används flitigt inom IT-forensiskt arbete. Ett hashset är en fil som innehåller hashsummor för olika typer av filer, dessa filer kan vara filer som kan anses suspekta eller så kan det vara kända ofarliga filer tex omodifierade operativsystemfiler. Er uppgift är att när filer läses in körs en funktion som tar fram hashvärdet för varje fil, detta kontrolleras sedan mot ert egenskapade hashset. Ni kan själva välja om ni vill kontrollera mot suspekta filer eller mot kända ofarliga filer eller båda. Om ni väljer suspekta bör dessa kanske skrivas ut för sig eller markeras på något sätt. Om ni väljer kända ofarliga filer så kan ni välja att inte skriva ut dessa eller markera dessa på något annat sätt.

Hårdvaru-mjukvaro egenskaper

- Inom IT-forensik är det viktigt att dokumentera så mycket som möjligt. Detta görs främst för att kunna återskapa utvinningen och kunna bevisa att allt gjordes enligt alla konstens regler. Det kan därför vara bra att ta fram alla uppgifter som går att få tag på om utrustningen som används. Er uppgift är att skriva ut information om datorn som programmet körs på. Information som kan vara intressent är vilken hårdvara (ram, cpu, gpu etc) och mjukvara (operativsystem etc).

Extra Extra: Automastik generering av rapport

- I det IT-forensiska arbetet ingår det nästan alltid rapportgenerering, dvs en rapport som innehåller information som kan vara intressant för tex en rättegång och eller en individ. Er uppgift är att användaren av programmet kan välja och markera filer som kan anses suspekta och spara dessa. Användaren ska också kunna skriva in egna kommentarer om hen så önskar. Rapporten skall sedan sparas i ett lämpligt format så som pdf eller dylikt, om bilder är markerade som suspekta så skall

dessa visas som bilder i dokumentet och till sist ska användaren kunna bestämma var rapporten ska sparas.