Estructuras de Datos y Algoritmos

Práctica II - Curso 2016/17

Ataque Inminente (II)

Descripción y objetivos

En la sesión de evaluación de la práctica se comprobó que el cambiar la estructura del programa de la forma siguiente podía conseguir que el tiempo de ejecución fuera 1/3 del que obteníamos con la estructura original:

```
datos[] ← lectura del fichero
busq[] ← texto que se busca (traducido a vector numérico)
bucle clave ← 0..65535
      busq_ofus[] ← copia de busq[]
      ofuscar(busq_ofus[], clave)
      bucle búsqueda de busq_ofus[] en datos[]
            si encontrado en posición pos:
                  trozo[] ← datos[pos-100..pos+500]
                  ofuscar(trozo[], clave)
                  texto  ← conversión a cadena de trozo[]
                  mostrar pos, clave y texto por pantalla
```

Esta mejora, sin embargo, no parece suficiente para que la historia tenga un final feliz: Con el mejor ordenador y lenguaje de programación disponible se tarda unos 10 segundos en procesar un fichero de 100K, lo que dado que el tiempo es lineal con el tamaño del fichero nos indica que se tardarían unos 3 días en procesar un fichero de 2 Gigas (el tamaño típico de un fichero de gestor de correos electrónicos). Esto es inaceptable.

El objetivo de ésta segunda práctica es conseguir multiplicar la velocidad de ejecución de la aplicación al menos 1000 veces para que se pueda resolver el problema en minutos, no en días. Para ello la clave consistirá en diseñar una estructura de datos que tenga las propiedades adecuadas para tratar este problema.

Existen varias alternativas que permiten resolver el problema (tanto basadas en estructuras que se contemplan en la asignatura como en otras que no aparecen en el temario), se sugiere al alumno que realice una investigación previa basándose en el tipo de problema que queremos resolver. Hay una estructura particularmente adecuada que se menciona de pasada en una de las transparencias de la asignatura y que no requiere de más de 30 líneas de código para implementarse.

Una idea clave para conseguir el resultado podría ser la siguiente: Sea m la longitud de la cadena de búsqueda (típicamente unos 10 caracteres), si conseguimos una estructura de datos que pueda almacenar las 65.536 versiones ofuscadas de la cadena de búsqueda y realizar la comprobación de si un trozo de los datos de tamaño m coincide con alguna de esas versiones en un tiempo $O(m)$, conseguiríamos una mejora de varios miles de veces respecto al algoritmo original.

La estructura del programa sería la siguiente (el acrónimo EDM hace referencia a esa estructura de datos "mágica"):

```
datos[0..n-1] ← lectura del fichero
busq[0..m-1] ← texto que se busca (traducido a vector numérico)
bucle clave ← 0..65535
      busq_ofus[] ← copia de busq[]
      ofuscar(busq_ofus[], clave)
      insertar (busq_ofus, clave) en EDM
fin-bucle
bucle pos ∈ [0..n-m]
      si datos[pos..pos+m-1] ∈ EDM
            trozo[] ← datos[pos-100..pos+500]
            ofuscar(trozo[], clave)
            texto  ← conversión a cadena de trozo[]
            mostrar pos, clave y texto por pantalla
fin-bucle
```

Presentación y Evaluación de la práctica

Se pueden utilizar los lenguajes Java, Python, C, Pascal, Haskell o R para la implementación de la aplicación. Si se desea utilizar otro lenguaje debe obtenerse primero la autorización del profesor.

Para una correcta evaluación de la práctica el alumno deberá:

1. Presentar electrónicamente (por el Aula Virtual de la Escuela o por correo electrónico), antes de las 23:59 del 27 de noviembre de 2016, un fichero comprimido que contenga el código fuente de la aplicación utilizada para resolver el problema planteado. En el código fuente debe aparecer (como comentario) el nombre de quienes han realizado la práctica.
2. Presentarse a la sesión de evaluación que le corresponda según su grupo de laboratorio en la semana del 28 al 30 de noviembre de 2016. En esta sesión se probará la aplicación realizada. Es posible que en la sesión se solicite la modificación del código de la aplicación.

En el caso de realización por parejas (la situación habitual), tan sólo es necesario que uno cualquiera de ellos realice la presentación electrónica. En la evaluación, sin embargo, si es necesaria la presencia de ambos y la evaluación puede ser distinta para cada uno de ellos.