Proyecto Programado I

Profesor: Santiago Núñez Corrales

11 de abril de 2011

1. Introducción

En el año 4 a.C. Julio César, el emperador de Roma más importante de la historia, fallecía dejando tras de sí uno de los imperios mejor consolidados de la historia. Muerto finalmente a manos de Bruto, se convierte en una leyenda que llega a nuestros días en la forma de la obra de teatro Julius Caesar de William Shakespeare.

Parte del trabajo del César era el envío de comunicados a través de la República por medio de mensajeros a caballo, usualmente armados y expertos combatientes. Aún cuando el mensajero fuese diestro en el uso de las armas, en muchas ocasiones eran interceptados por flechas y grupos de enemigos de la República Romana. El mensaje era descifrado y la ventaja estratégica perdida. Esto costó al César numerosas batallas y por tanto se decidió encontrar un método para ocultar la información a la vista: no importa si se tiene el texto del documento, este es ilegible sin una *clave*, una llave que permita comprender cómo se organizan los caracteres.

La criptografía es la disciplina que involucra matemática y teoría de la información con el objetivo de asegurar que un mensaje será entregado y descifrado solo por las partes que han acordado estar involucradas en el proceso de comunicación. No solamente en Roma, sino a través el mundo, la presión militar y diplomática ha forzado a las organizaciones y a los paises a crear mecanismos de seguridad de datos, y con ellos, mecanismos cada vez más efectivos para romperlos. Allan Turing antes de su trabajo en computabilidad fue el personaje estratégico para decodificar el funcionamiento de la máquina Enigma y así tener los mensajes del Riech decodificados antes de que llegaran a las flotas navales de U-232 en el Atlántico Norte.

Este proyecto programado usa el método criptográfico de César para desarrollar habilidades y destrezas en desarrollo de código en lenguaje ensamblador.

2. Descripción del Proyecto

El proyecto consiste en la implementación de la cifra modificada de César en lenguaje ensamblador para x86, sintaxis AT&T tal como se describe en el libro Programming Ground Up, disponible en el Tec Digital. Para el desarrollo del programa, se utilizará un código base en lenguaje en C que facilite la comprensión del algoritmo.

2.1. Ejecucíon del Programa

Cuando el programa se ejecute, deberá recibir los siguientes parámetros en el orden aquí descrito:

1. **Archivo de entrada**. Una cadena de caracteres que representa la dirección relativa[1] del archivo que se desea procesar. El archivo debe estar en formato texto simple codificación ASCII estandar.

2. **Archivo de salida**. Una cadena de caracteres que representa la dirección relativa del archivo que se creará durant el proceso de encripción. El archivo deberá construirse en formato texto simple codificación ASCII estandar.

3. **Operación**. El programa recibirá una bandera de acuerdo a las operación que se desee efectuar, siendo estas:

   - Encriptar: con la bandera `-e` y un valor entre 0 a 35 se efectuará el proceso de encripción.
   - Desencriptar: con la bandera `-d` y un valor entre 0 a 35 se efectuará el proceso de desencripción.

2.2. Código Fuente

El código fuente debe compilarse y ejecutarse en un sistema GNU/Linux con el ensablador GAS, sintaxis AT&T.

2.3. Documentación

La tarea programada debe venir acompañada de documentación externa en formato IEEE Transactions (LaTeX o plantilla para Word) que contenga una descripción del problema, la estrategia de solución implementada y un conjunto de pruebas, así como una copia verbatim (completa) del codigo fuente de la aplicación. La plantilla de documentación puede descargarse de `http://www.ieee.org/documents/TRANS-JOUR.doc`.

3. Evaluación

1. **Correctitud**. El programa compila, se ejecuta y generera resultados correctos sin errores o advertencias. [50 %]

2. **Completitud**. El programa cumple con la especificación de la tarea programada. [30 %]

3. **Estructura y organización**. El código fuente está apropiadamente documentado; existe una organización lógica en funciones que cumplen objetivos específicos; los nombres de los procedimientos permiten leer fácilmente el código. [10 %]

4. **Diseño y proceso**. La documentación externa es apropiada. El diseño del programa es modular y comprensible. El entregable se envia en un archivo tar.gz. [10 %]

4. Entrega

El proyecto programado deberá entregarse el día 5 de Mayo de 2011 al correo electrónico `arquitectura.santaclara@gmail.com` con hora máxima del servidor 23:59:59.

[1] La direccion relativa siempre es aquella dada a partir del directorio de trabajo actual.

5. Puntos adicionales

1. Extensión para programación. En la actualidad, el programa solo soporta 36 símbolos. Extienda la tabla de símbolos para soportar caracteres necesarios en C. (15 pts)

2. Ruptura de código. El sistema de encripción utilizado en este proyecto puede ser fácilmente roto debido a que es solo una transposición de los caracteres de manera circular. Así, a partir de una tabla de frecuencias de referencia, se puede calcular la desviación estándar de la diferencia entre las frecuencias esperadas para el texto (usualmente dependientes del idioma). Al encontrar el desplazamiento que minimice la frecuencia observada vs. esperada para la desviación estándar

$$\chi^2 = \sum_{i=0}^{n-1} \frac{(obs_i - esp_i)^2}{esp_i}$$

   Construya una función que implemente la ruptura de código fuente utilizando el conjunto de instrucciones de punto flotante SSE2. Utilice un texto en inglés como referencia. Sugerencia: obras completas de William Shakespeare. [2](35pts)

3. Parametrización de frecuencias de ruptura de código. Desarrolle una función que permita al usuario ingresar un texto de referencia para establecer las frecuencias esperadas de los caracteres. (25pts)

4. Perfiles de parametrización. Construya una función que reciba varios texto de referencia (Español, Inglés, código fuente) de tal forma que el programa indique qué tipo de texto es el más cercano al documento codificado. (25pts)

[2] http://www.gutenberg.org/cache/epub/100/pg100.txt