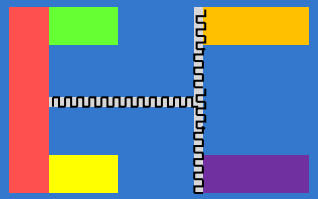


PRÁCTICA 4.
PROGRAMA EN ENSAMBLADOR

PRÁCTICA 4. PROGRAMA EN ENSAMBLADOR

Índice

- Práctica 4.1: Cifrado/Descifrado TEA (3 sesiones)
- Práctica 4.2: Determinante de una matriz (3 sesiones)

PRÁCTICA 4.1. PROGRAMA EN ENSAMBLADOR

Desarrollo y entrega

- En esta práctica se va a utilizar el lenguaje ensamblador de MIPS para implementar el algoritmo de codificación y decodificación TEA.
- El alumno debe entregar para cada apartado de la práctica la siguiente documentación:
  - Una memoria documental en la que se explique cómo se ha implementado la práctica, el código utilizado y volcados de pantalla que demuestren su funcionamiento.
  - Los archivos asociados a la implementación realizada para que el profesor pueda ejecutarlos.

PRÁCTICA 4.1. PROGRAMA EN ENSAMBLADOR

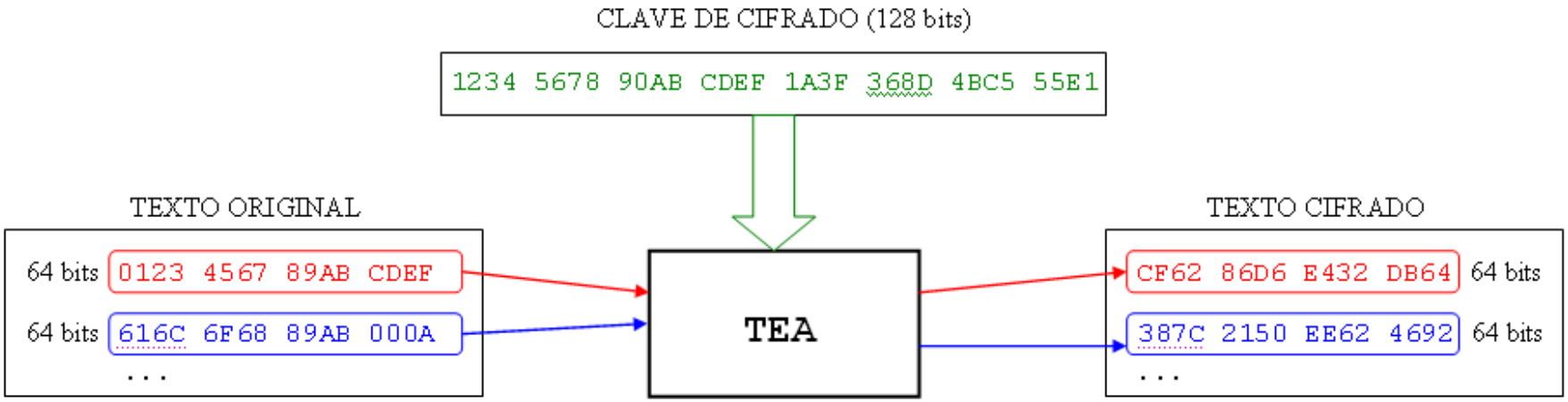
Introducción

- TEA (Tiny Encryption Algorithm) es un algoritmo criptográfico utilizado para el cifrado de información.
- Es un algoritmo de cifrado libre de patentes, sencillo de implementar y que utiliza cifrado por bloques de 64 bits (toma el texto original y para cada bloque de 64 bits genera un nuevo bloque cifrado de 64 bits).
- El tamaño del texto cifrado es igual que el tamaño del texto original y permite la paralelización del proceso de cifrado.
- Utiliza una clave de cifrado de 128 bits.
- Utiliza al menos 32 rondas de ejecución para la realización del cifrado.

PRÁCTICA 4.1. PROGRAMA EN ENSAMBLADOR

Introducción

- El esquema del proceso de cifrado del algoritmo TEA es el siguiente:



Cifrado TEA

- El algoritmo de cifrado TEA escrito en C es el siguiente:

```
void CIFRAR (unsigned long* v, unsigned long* k)
        {
        unsigned long y=v [0], z=v [1], sum=0, i=0,    /* inicialización */
        delta=0x9e3779b9,                              /* constante necesaria */
        n=32;                                          /* número de rondas de cifrado */
        for (i=0; i< n; i++)   {              /* Ciclo básico */
                               sum += delta;
                               y +=  ((z<<4)+k[0])  xor  (z+sum)  xor  ((z>>5)+k[1]);
                               z +=  ((y<<4)+k[2])  xor  (y+sum)  xor  ((y>>5)+k[3]);
                                }             /* fin del ciclo */
        v [0]=y;   v [1]=z;
        }
```

- Donde v[0] y v[1] son los 64 bits a codificar y k es la clave de cifrado de 128 bits.

PRÁCTICA 4.1: CIFRADO TEA

Enunciado

- Implementar en el ensamblador de MIPS el algoritmo de cifrado TEA que se ha explicado. Hay que tener en cuenta las siguientes consideraciones:
  - La información a cifrar será un bloque de 64 bits que estará almacenado en memoria (2 words).
  - El resultado del cifrado será un bloque de 64 bits que estará almacenado en memoria (2 words).
  - La clave de cifrado serán 128 bits que estarán almacenados en memoria (4 words).
  - Se utilizarán 32 rondas para realizar el cifrado de la información.

PRÁCTICA 4.1: CIFRADO TEA

Ejemplos

- Para poder comprobar el correcto funcionamiento del cifrado TEA implementado por el alumno, se pueden probar los siguientes ejemplos de cifrado (expresados en hexadecimal):

Clave de cifrado: 1111 1111 2222 2222 3333 3333 4444 4444	
Texto original	**Texto cifrado**
0123 4567 89ab cdef	cf62 86d6 e432 db64
616c 6f68 89ab 000a	387c 2150 ee62 4692

Clave de cifrado: 1234 5678 90ab cdef fedc ba09 8765 4321	
Texto original	**Texto cifrado**
475f 4345 4f44 4152	c450 058a 65bc ed2e
434e 5546 414e 4f49	eaad 7984 167f e724

Descifrado TEA

- El algoritmo de descifrado TEA escrito en C es el siguiente:

```
void DESCIFRAR (unsigned long* v, unsigned long* k)
        {
        unsigned long y=v [0], z=v [1], sum=0xc6ef3720, i=0,    /* inicialización */
        delta=0x9e3779b9,                                       /* constante necesaria */
        n=32;                                                   /* número de rondas de cifrado */
        for (i=0; i< n; i++)   {                /* Ciclo básico */
                               z -=  ((y<<4)+k[2])  xor  (y+sum)  xor  ((y>>5)+k[3]);
                               y -=  ((z<<4)+k[0])  xor  (z+sum)  xor  ((z>>5)+k[1]);
                               sum -= delta;
                                }               /* fin del ciclo */
        v [0]=y;   v [1]=z;
        }
```

- Donde v[0] y v[1] son los 64 bits a decodificar y k es la clave de cifrado de 128 bits.

PRÁCTICA 4.1: DESCIFRADO TEA

Enunciado

- Implementar en el ensamblador de MIPS el algoritmo de descifrado TEA que se ha explicado. Tener en cuenta las siguientes consideraciones:
  - La información a descifrar será un bloque de 64 bits que estará almacenado en memoria (2 words).
  - El resultado del descifrado será un bloque de 64 bits que estará almacenado en memoria (2 words).
  - La clave de cifrado serán 128 bits que estarán almacenados en memoria (4 words).
  - Se utilizarán 32 rondas para realizar el descifrado de la información.

PRÁCTICA 4.1: INTERFAZ PARA USUARIO

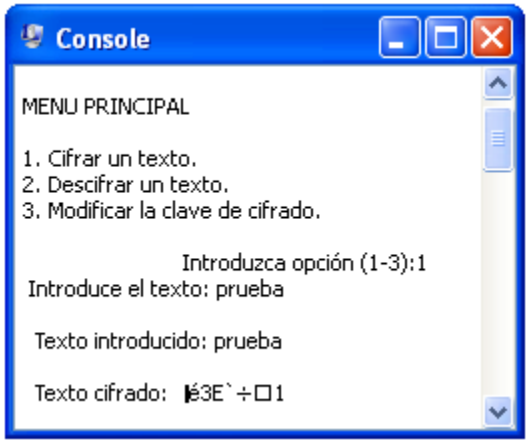
Enunciado

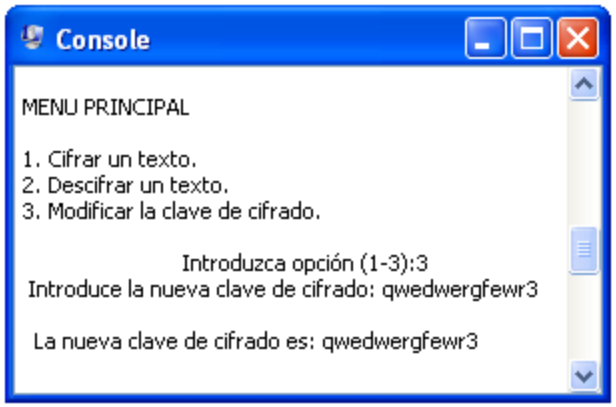
- Implementar en el ensamblador de MIPS un programa completo de cifrado y descifrado TEA que permita al usuario realizar las siguientes operaciones:
  1. Cifrar un texto de 64 bits introducido por el usuario.
  2. Descifrar un texto de 64 bits introducido por el usuario.
  3. Modificar la clave de cifrado.
- Tras finalizar la ejecución volverá al principio para solicitar una nueva operación al usuario.
- Si el usuario introduce cualquier operación diferente a las 3 anteriores, se mostrará por consola el mensaje "*HA SELECCIONADO UNA OPERACIÓN INCORRECTA*" y volverá a solicitar una nueva operación.

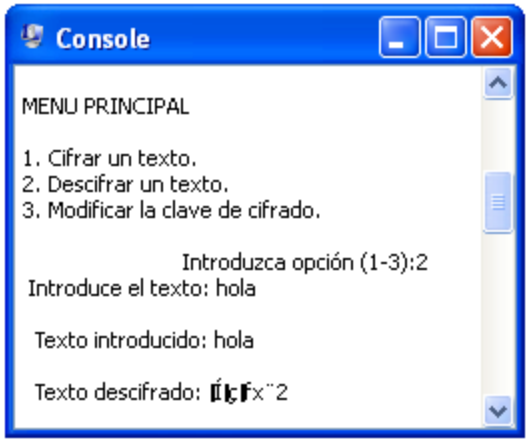
PRÁCTICA 4.1: INTERFAZ PARA USUARIO

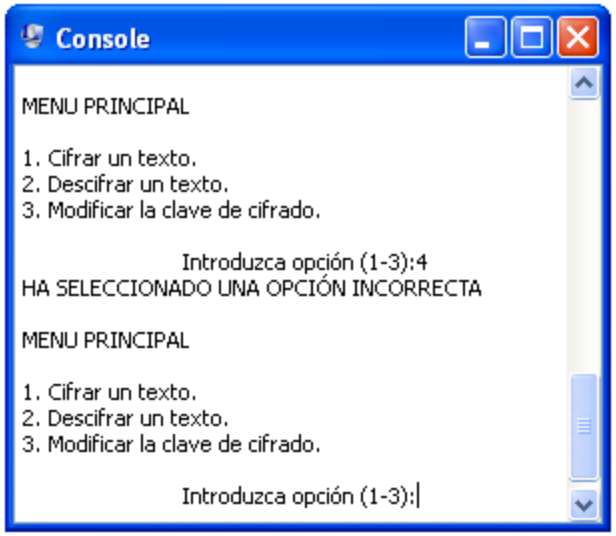
Enunciado

- La interfaz para el usuario debe ser similar a esta:









PRÁCTICA 4.2: DETERMINANTE DE UNA MATRIZ

Enunciado

- La práctica consiste en realizar un programa en ensamblador MIPS que implemente el determinante de una matriz 3x3.
- Para ello:
  - los valores de la matriz se deben introducir por teclado
  - el resultado debe salir por pantalla.