UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y Ciencias de la Computación

Programación de Sistemas y Concurrencia Práctica nº2.

Un algoritmo de cifrado por bloques opera con bloques de 64 bits y una clave de 128 bits siguiendo el siguiente procedimiento.

Para cada bloque de 64 bits (`unsigned int v[2]`), sea la clave *k* (`unsigned int k[4]`), y *delta* una constante igual a 0x9e3779b9:

Inicializar *sum* a 0xC6EF3720

Repetir 32 veces:

Restar a `v[1]` la aplicación del operador XOR (^) a
`(v[0]` desplazado a la izquierda 4 bits +`k[2])`,
`(v[0]` + `sum`) y
(`v[0]` desplazado a la derecha 5 bits)+`k[3]`

Restar a `v[0]` la aplicación del operador XOR (^) a:
`(v[1]` desplazado a la izquierda 4 bits + `k[0])`
`(v[1]`+ `sum`) y
(`v[1]` desplazado a la derecha 5 bits)+`k[1]`

Restar a `sum` el valor de `delta`.

Al final de las 32 iteraciones, se tendrá en `v` el valor desencriptado de los 64 bits.

Realizar un programa en C que cargue en memoria dinámica el contenido de un fichero encriptado (en formato little-endian), realice su desencriptado siguiendo el procedimiento descrito y lo almacene en un fichero de salida (recordar liberar la memoria dinámica al final). Tanto el nombre del fichero de entrada como el del fichero de salida son indicados como argumentos del programa en la línea de comandos.

Como el tamaño del fichero original no tiene porqué ser múltiplo de 8 y el algoritmo de desencriptado trabaja con bloques de 8 bytes (64 bits) hay que tener en cuenta las siguientes indicaciones:

- Al comienzo del fichero encriptado se encuentra almacenado (`unsigned int`) el tamaño del fichero original desencriptado. Este tamaño no hay que almacenarlo en memoria dinámica.
- Al hacer el encriptado, si el tamaño del fichero original no era múltiplo de 8, se tiene al final un bloque incompleto para aplicar el cifrado, por lo que artificialmente se completa hasta tener 8 bytes. Estos valores de relleno están almacenados también en el fichero encriptado, y deben también ser desencriptados, pero no deben escribirse en el fichero de salida.

Por ejemplo, si el fichero original ocupa 30 bytes, al hacer el encriptado se tuvieron que utilizar 32 bytes, esto es, se pusieron 2 valores de relleno. La longitud total del fichero encriptado es 4 (almacenamiento de la longitud del fichero encriptado) + 30 + 2 (posiciones de relleno), esto es 36 bytes. Al hacer el desencriptado, se hará de 32 bytes, pero sólo se escriben 30 en el fichero de salida.

Para realizar el desencriptado se recomienda definir una función con la siguiente cabecera:

```
void decrypt(unsigned int* v, unsigned int* k);
```

donde `v` es el array de 2 `unsigned int` que se va a desencriptar y `k` es la clave consistente en un array de 4 `unsigned int` con los siguientes valores: `{128, 129, 130, 131}`.

Funciones de C que pueden resultar útiles para la práctica (no es necesario utilizar todas): `fopen`, `fread`, `fwrite`, `fclose`, `malloc`, `free` y `memcpy`.