

Instituto Tecnológico de Costa Rica Área Académica de Ingeniería en Computadores CE 4301 — Arquitectura de Computadores I

## Tarea 1 Programación en Ensamblador con ARM

Fecha de asignación: 18 de septiembre, 2020 Fecha de entrega: 27 de septiembre, 2020 Grupos: 1 persona Profesor: Luis Chavarría Zamora

Para los siguientes ejercicios de programación, desarrolle la solución al problema planteado de manera clara y correcta. Debe incluir en cada caso un diagrama de flujo y el código en lenguaje ensamblador ARM (ARMv4) que soluciona cada problema. Debe ser desarrollado en VisUAL, el programa debe tener los comentarios necesarios. Debe quedar clara la solución de cada problema:

1. Un programa de encriptación y desencriptación sencillo. Como parte de este ejercicio se usará el programa desarrollado en el Taller 1 (generador de números pseudo-aleatorios). Se encriptará la típica frase: **Hello, World!**. Coloque al final de la frase el valor en ASCII 0x0ah que significa *End of Line* (EOL). La/el estudiante transformará cada caracter de ASCII a su representación de 8 bits, coloque los caracteres desde la posición de memoria 0x100 en adelante <sup>12</sup>. A continuación se explica el proceso:

## a) Encriptación:

- 1) Genera el primer número pseudoaleatorio (semilla, primera letra del apellido en mayúscula)  $^3$ .
- 2) Toma el primer caracter de la frase a encriptar, le suma 29d y le aplica XOR con el valor semilla.
- 3) Toma el resultado y lo sobreescribe, es decir, si encripta el caracter en la posición 0x100, el resultado de la operación se guarda en 0x100.
- 4) Se genera el siguiente valor aleatorio usando el polinomio LFSR:  $x^8 + x^6 + x^5 + x^4 + 1$ .
- 5) Se encripta (XOR) el siguiente caracter de la frase a encriptar con el valor aleatorio generado en el paso anterior. Sobreescribe el resultado.
- 6) Repita desde el paso 1b4 hasta llegar al EOL (tiene que encriptar este también).
- 7) Guarde el largo de la frase (incluyendo EOL) en la posición 0x200.

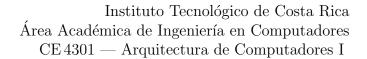
## b) Desencriptación:

1) Genera el primer número pseudoaleatorio (semilla, primera letra del apellido en mayúscula).

 $<sup>^1{\</sup>rm La}$  posición de memoria 0x100, tome en cuenta que el índice de la posición de memoria en la instrucción se escribe en decimal, entonces:  $100H \to 256D$ 

<sup>&</sup>lt;sup>2</sup>use un byte por palabra (4 bytes)

<sup>&</sup>lt;sup>3</sup>Por ejemplo: Xi,  $X \rightarrow 088D \rightarrow 58H \rightarrow 01011000B$ 





- 2) Toma el primer caracter de la frase encriptada, le aplica XOR con el valor semilla y luego le resta 29d al resultado.
- 3) Toma el resultado y lo sobreescribe, es decir, si encripta el caracter en la posición 0x100, el resultado de la operación se guarda en 0x100.
- 4) Se genera el siguiente valor aleatorio usando el polinomio LFSR:  $x^8 + x^6 + x^5 + x^4 + 1$ .
- 5) Se desencripta (XOR y restar 29d) el siguiente caracter de la frase encriptada con el valor aleatorio generado en el paso anterior. Sobreescribe el resultado.
- 6) Repita desde el paso 1b4 hasta que cumpla el largo de la frase guardado en 0x200.

Todo esto debe estar en un solo programa. El resultado de la desencripción será la frase Hello, World!.

2. Un verificador de números primos. Debe usar el tamiz de Erastótenes (Sieve of Eratosthenes), este tamiz lo debe llenar con posiciones de memoria <sup>4</sup> a partir de 0x100, debe indicar un 1h cuando no es primo. Debe usar la posición para detectar el número, por ejemplo el número 2 es 0x100, el 3 es 0x104 y así con los siguientes. Debe usar el mismo LFSR y valor semilla del Taller 1, genere 100 números y revise si cada número es primo de acuerdo con el tamiz. Si el número es primo escríbalo a partir de la posición 0x700.

El trabajo debe seguir los siguientes lineamientos:

- Debe usar la herramienta VisUAL. No se aceptarán trabajos en otras herramientas.
- Se deben enviar dos \*.S correctamente documentado internamente. Colóquelos en un comprimido junto con los diagramas de flujo de cada problema.
- Como cualquier documento técnico, la ortografía y el formato deben ser impecables.
- La entrega se debe realizar por medio del TEC-Digital en la pestaña de evaluación, no más tarde de la fecha de entrega a las 11:59 pm. No se aceptarán trabajos después de esta hora y fecha.
- Los documentos serán sometidos a control de plagios para eliminar cualquier intento de plagio con trabajos anteriores similares o copias textuales.

Tarea 1 Ensamblador en ARM Página 2 de 2

<sup>&</sup>lt;sup>4</sup>use un byte por palabra (4 bytes)