

Laboratorio 1

Competencias a desarrollar

En este laboratorio el estudiante aplicará los conocimientos de ingreso de datos, subrutinas, operaciones de punto flotante y manejo de memoria mediante un programa estructurado con subrutinas, en el lenguaje ensamblador ARM.

El siguiente proyecto se trabajará en parejas asignadas por su profesor.

Introducción

La generación de secuencias de números aleatorios es un tema de gran interés en la computación y un área de investigación activa. En muy pocos casos se puede decir que una secuencia es verdaderamente aleatoria y las técnicas aplicadas en estos casos son muy variadas: física cuántica, imágenes de lámparas de lava, circuitos inciertos¹ e incluso el flujo de electrones en una memoria que almacena la información variando valores de resistencias (no voltajes)². Es por eso que ordinariamente se utilizan técnicas para generar secuencias pseudo-aleatorias, es decir secuencias que son reproducibles pero que casi no presentan patrones notables.

Nuestra sociedad moderna depende mucho de los números aleatorios (Criptografía, seguridad en transmisión de datos, verificación de integridad de paquetes TCP/IP, simulaciones computacionales, sistemas de interferencia de telecomunicaciones, etc.). El objetivo de este laboratorio es comprender e implementar un **Linear Feedback Shift Register (LFSR)**, uno de los métodos más comunes de generación de secuencias pseudo-aleatorias.

Instrucciones

1. Lea cuidadosamente los conceptos involucrados en el funcionamiento del LFSR en la página de [wikipedia](http://es.wikipedia.org/wiki/Linear_Feedback_Shift_Register). Asegurese de comprender los conceptos de periodo y polinomio de longitud máxima.
2. Implemente en un archivo llamado `random.s` las siguientes subrutinas utilizando el estándar ABI:
 - a. **lfsr**: Un generador LFSR Galois de 32 bits que utilice un polinomio de longitud máxima. La subrutina recibirá en R0 la dirección de un vector de memoria con variables tipo word en donde almacenará los números generados, en R1 el tamaño del vector y en R2 la semilla para el LFSR.

¹ <http://spectrum.ieee.org/computing/hardware/behind-intels-new-randomnumber-generator>

² <http://spectrum.ieee.org/semiconductors/memory/a-memristor-true-randomnumber-generator>

- b. **min**: Una subrutina que retorna en R0 el valor mínimo de un vector punto flotante de precisión simple. Los parametros de entrada serán: R0 la dirección del vector y R1 el tamaño del vector.
- c. **max**: Una subrutina que retorna en R0 el valor máximo de un vector punto flotante de precisión simple. Los parametros de entrada serán: R0 la dirección del vector y R1 el tamaño del vector.
- d. **avg**: Una subrutina que retorna en R0 el valor promedio de un vector punto flotante de precisión simple. Los parametros de entrada serán: R0 la dirección del vector y R1 el tamaño del vector.
- e. **norm**: Una subrutina que normaliza un vector de punto flotante de precisión simple. Los parametros de entrada serán: R0 la dirección del vector, R1 el tamaño del vector, R2 el valor mínimo del vector y R3 el valor máximo del vector. (los valores normalizados se sobre escriben en el vector de entrada).
- f. **printVec**: Imprime los valores de un vector con valores punto flotante. Los parametros de entrada serán: R0 la dirección del vector de precisión simple y R1 el tamaño del vector.

3. Implemente un archivo llamado `main.s` que ejecute los siguientes pasos:





- a. Utiliza como semilla un valor entero ingresado por el usuario para generar un vector de 2^{20} valores pseudo-aleatorios de 32 bits. **Importante: Debe reservar un espacio de memoria de tamaño adecuado antes de llamar a la subrutina `lfsr`.**
- b. Convierte los valores aleatorios almacenados en el vector al formato punto flotante de precisión simple. (Es decir convierte un valor entero 1 a 1.00)
- c. Imprime el mínimo y máximo del vector punto flotante.
- d. Normaliza el vector punto flotante a valores entre 1 y 0.
- e. Imprime el valor promedio del vector normalizado.
- f. Imprime los valores del vector normalizado.

4. Puntos extra: Aplicar aritmetica vectorial en las operaciones de normalización del vector.

Material a entregar

1. Archivos `main.s` y `random.s`

Evaluación

Criterios de evaluación	Nivel 4 Distinguido 	Nivel 3 Experto 	Nivel 2 Aprendiz 	Nivel 1 Novato 
Pre-laboratorio (si aplica, sino los puntos van en funcionamiento) 10%	Los estudiantes enviaron el programa solicitado y funcionando en todos los requerimientos. 10%	No aplica.	No aplica.	No enviaron el pre-laboratorio o no funciona. 0%
Funcionamiento del Programa 40%	El programa funciona con todos sus requerimientos: ingreso de datos, despliegue de resultados y salida correcta al sistema operativo. 40%	El programa funciona al menos en el 80% de sus requerimientos: ingreso de datos, despliegue de resultados y salida correcta al sistema operativo. 30%	El programa funciona entre el 70% y 60% de sus requerimientos: ingreso de datos, despliegue de resultados y salida correcta al sistema operativo. 20%	El programa funciona en menos del 50% de sus requerimientos: ingreso de datos, despliegue de resultados y salida correcta al sistema operativo. 15%
Programación defensiva (si aplica, sino los puntos van en funcionamiento) 10%	El programa proporciona mensajes oportunos ante situaciones inesperadas o ingreso de datos incorrectos 10%	El programa proporciona algunos mensajes oportunos ante situaciones inesperadas o ingreso de datos incorrectos 8%	El programa proporciona pocos mensajes oportunos ante situaciones inesperadas o ingreso de datos incorrectos 6%	El programa no proporciona mensajes oportunos ante situaciones inesperadas ni con ingreso de datos incorrectos 0%
Documentación del programa 5%	La documentación incluye encabezado y comentarios representativos en los bloques de código más importantes. Los nombres de las variables son significativos. 5%	La documentación incluye encabezado pero no hay comentarios representativos en todos los bloques de código más importantes. Algunos nombres de las variables son significativos. 4%	Falta documentación en el encabezado o en bloques de código Los nombres de las variables son medianamente significativos. 3%	Falta más del 50% de la documentación del código Los nombres de las variables no expresan ningún significado. 1%
Orden del programa 5%	La presentación del programa es muy clara y ordenada, y utiliza una tabulación adecuada. 5%	La presentación del programa es aceptable, y utiliza una tabulación más o menos adecuada. 4%	La presentación del programa es regularmente clara y ordenada. La tabulación es regular. 3%	La presentación del programa es confusa y desordenada. No hay tabulación de las instrucciones. 1%
Uso de subrutinas 10%	El programa es 100% estructurado con subrutina(s) que usan el estándar ABI y que hacen que el código sea más eficiente, comprensible y ordenado. Puede incluir macros que facilitan el entendimiento del programa. Las subrutinas están en un archivo diferente al main. 10%	Al menos el 80% del programa es estructurado con subrutinas que usan el estándar ABI y que hacen que el código sea más eficiente, comprensible y ordenado. Puede incluir macros que facilitan el entendimiento del programa. Las subrutinas están en un archivo diferente al main. 8%	Entre el 70-60% del programa es estructurado con subrutinas que usan el estándar ABI y que hacen que el código sea más eficiente, comprensible y ordenado. Puede incluir macros que facilitan el entendimiento del programa. Las subrutinas están en el mismo archivo del main. 6%	Las subrutinas no usan el estándar ABI o el programa no está estructurado. No se hizo la subrutina solicitada 0%
Diario reflexivo 10%	El estudiante realizó una reflexión completa con aportes significativos para mejorar su aprendizaje en/antes de la fecha solicitada y contestó todas las preguntas. 10%	El estudiante realizó una reflexión con aportes significativos para mejorar su aprendizaje en/antes de la fecha solicitada y contestó el 80% de las preguntas. 8%	El estudiante realizó una reflexión con algunos aportes significativos para mejorar su aprendizaje en/antes de la fecha solicitada y contestó el 60% de las preguntas. 6%	El estudiante realizó una reflexión con muy pocos aportes significativos para mejorar su aprendizaje en/antes de la fecha solicitada y contestó menos del 50% de las preguntas. 2%