

**TAREA ACADÉMICA DEL LABORATORIO DE
ORGANIZACIÓN Y ARQUITECTURA DE COMPUTADORAS**

| | | | |
|-------------------|---|--------------|--|
| SESIÓN N°: | 2 | TEMA: | PROGRAMACIÓN EN ENSAMBLADOR DE 64 BITS Y RENDIMIENTO |
|-------------------|---|--------------|--|

INDICACIONES GENERALES

- Leer la guía teórica correspondiente a esta sesión para complementar su desarrollo.
- Realizar sus consultas y/u observaciones a los jefes de práctica.
Hector Bustamante Ruiz: **hbustamanter@pucp.pe**
Giancarlo Villena Prado: **gvillenap@pucp.edu.pe**
con copia al profesor del curso Stefano Romero Gutierrez (**sromerog@pucp.pe**).
- Estructurar su archivo con el formato **T2_codigo.zip** (i.e. T2.20082169.zip). En el archivo comprimido debe adjuntar el informe y los programas codificados en el siguiente formato: **Informe_codigo.pdf**, **asmSTD.asm**, **Pregunta1.c** y **Pregunta2.c** y **Bench.c** (i.e. Informe_20082169.pdf, asmSTD.asm, Pregunta1.c, Pregunta2.c, bench.c). El informe debe contener el diagrama de flujo de su solución, la descripción de su desarrollo y la respuesta a las preguntas de la guía. Los códigos deben tener comentarios relevantes para su entendimiento.
- Presentar su solución antes del 28/09/2019 a las 16:00 en la intranet (ruta: **/Laboratorios/Laboratorio 2/Tarea/Soluciones**).

OBJETIVOS

- Brindar al alumno claridad de conceptos acerca de la programación en assembler de 64 bits y rendimiento.
- Codificar programas en lenguaje ensamblador de 64 bits utilizando sintaxis Intel.
- Aprender a utilizar el concepto de *Calling Convention*.
- Aprender a utilizar las formulas para medir el rendimiento de un programa C.

CUESTIONARIO

PREGUNTA N°1

(5 pts)

El promedio de los elementos de un vector $\vec{x}=(x_1, x_2, \dots, x_n)$ se define en la Ecuación (1).

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (1)$$

Donde μ es la media aritmética, N el número de elementos y x_i es el valor de x en posición i -ésima.

Así mismo, la desviación estandar se define como lo expresado en la Ecuación (2):

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (|x_i - \bar{x}|^2)}{N}} \quad (2)$$

Siendo así, se le pide evaluar el rendimiento del sistema calculando la **desviación estandar** de un vector (**v**) que contenga valores de coma flotante de los primeros 2048 numeros. Para ello, debe realizar lo siguiente:

- Codificar el programa principal en lenguaje C **Pregunta1.c**. En este programa se inicializará y llenará el vector de datos. Dentro de este programa debe calcular la desviación estandar a través de la función **cSTD** la cual deberá tener como entrada los siguientes parámetros:

- **V1 (coma flotante)**: Vector en coma flotante.
- **N (entero positivo)**: Cantidad de elementos del vector
- **V2 (coma flotante)**: Resultado del cálculo de la desviación estandar.

Finalmente, el programa deberá imprimir el siguiente mensaje:

El resultado de la desviación estandar en C es

- Codificar una función en lenguaje ensamblador de 64 bits que calcule la desviación estandar. Para ello, debe crear la función **asmSTD.asm** y considerar que los parámetros de entrada son los mismos que la función **cSTD**.

Para validar su código, debe llamar a la función desde el programa principal **Pregunta2.c**, el cual debe estar basado en **Pregunta1.c**, y debe imprimir el siguiente mensaje:

El resultado de la desviación estandar en ASM es

- Codificar el programa principal **Bench.c** en lenguaje C . Este programa ejecutará 100 veces las funciones **cSTD** y **asmSTD** con el vector de datos **v**. En cada iteración

se generará un tiempo de ejecución para cada programa los cuales deben ser almacenados en dos arreglos independientes de manera ordenada. Al finalizar las 100 iteraciones, se deberá calcular la media, la desviación estándar y la mediana de ambos arreglos. Finalmente debe imprimir los siguientes mensajes en el terminal, una vez que se ejecuten:

La media de los 100 tiempos de ejecución en ASM es
La std de los 100 tiempos de ejecución en ASM es
La mediana de los 100 tiempos de ejecución en ASM es
La media de los 100 tiempos de ejecución en C es
La std de los 100 tiempos de ejecución en C es
La mediana de los 100 tiempos de ejecución en C es

Posteriormente, en su informe, comentar qué función se ejecuta más rápido y qué medida de rendimiento (media o mediana) ha sido más útil para determinarlo.