



PRÁCTICA NO. 4

Objetivos:

- General

- Desarrollar una aplicación de consola en lenguaje ensamblador para la arquitectura ARM64, que permita implementar y ejecutar distintos algoritmos de ordenamiento

- Específicos

- Implementar diferentes algoritmos de ordenamiento en ensamblador
- Permitir que el usuario seleccione la velocidad de visualización del proceso de ordenamiento, con opciones que van desde visualización sin retardos hasta una visualización detallada y lenta.
- Generar un reporte de tiempos de ejecución: Implementar una funcionalidad que mida y reporte el tiempo de ejecución de cada algoritmo cuando se elige la opción sin retardos, permitiendo al usuario comparar la eficiencia de los distintos métodos.
- Aplicar técnicas de optimización a nivel de ensamblador para mejorar el rendimiento del programa, aprovechando las características específicas de la arquitectura ARM64.

Descripción:

En esta práctica, los estudiantes deberán desarrollar una aplicación de consola que funcione en sistemas basados en la arquitectura ARM64. La aplicación deberá de poder ordenar un listado de números por diferentes algoritmos de ordenamiento. El objetivo es familiarizarse con el entorno de programación en sistemas ARM64, practicar el desarrollo de aplicaciones de consola y aplicar conceptos de programación como funciones, manejo de entradas y salidas, y validación de datos.

Funcionamiento

Al iniciar la aplicación se desplegará un encabezado con los siguientes datos:

<i>Universidad de San Carlos de Guatemala</i> <i>Facultad de Ingeniería</i> <i>Escuela de Ciencias y Sistemas</i> <i>Arquitectura de Computadores y Ensambladores 1</i> <i>Sección A</i> <i>Nombre del estudiante</i> <i>Carnet del Estudiante</i>
--

Se deberá de esperar que el usuario presione la tecla enter para continuar. Al presionar la tecla enter se desplegará un menú con las siguientes opciones.

- | |
|---|
| <ol style="list-style-type: none">1. Ingreso de lista de números2. Bubble Sort3. Insertion sort4. Finalizar programa |
|---|

Ingreso de lista de números

En esta parte se deberá de ingresar el listado de los números a ordenar para los siguientes algoritmos y se podrá realizar de 2 formas:

1. Ingresando los números de forma manual separados por una coma, por ejemplo:
1,2,8,21,5,1,6,7,8,122,36
2. Ingresar un archivo csv donde de igual forma los números estarán separados por medio de una coma

Aquí queda a discreción del estudiante como quiere realizar la solicitud del ingreso de la lista de números sin embargo se recomienda el uso de un submenú para poder hacer el ingreso de estos.

Ejemplos:

1. De forma manual
2. Carga de Archivo csv
3. Regresar al menú anterior

Ordenamientos

Algoritmos de Ordenamiento

Selecciona entre diferentes algoritmos para ordenar tu conjunto de datos. Los algoritmos disponibles son:

- **Bubble Sort:** Un algoritmo sencillo que compara pares de elementos adyacentes y los intercambia si están en el orden incorrecto.
- **Quick Sort:** Un algoritmo rápido que divide el conjunto en dos subarreglos alrededor de un pivote.
- **Insertion Sort:** Un algoritmo de clasificación simple que funciona insertando iterativamente cada elemento de una lista sin ordenar en su posición correcta en una porción ordenada de la lista
- **Merge Sort:** Un algoritmo de ordenamiento externo estable basado en la técnica divide y vencerás.

Al seleccionar cualquier de estos 4 algoritmos en el menú principal, le solicitara al usuario el ingreso de 2 parámetros que son la forma en el que va a ordenar si ascendente o descendente y si mostrara paso a paso el ordenamiento de la lista.

Forma en la que se va a ordenar

El usuario puede elegir en que forma quiere hacer el ordenamiento de la lista de números.

Visualización del Proceso

Durante el proceso de ordenamiento, podrás ver cómo cambian los elementos en la lista en cada paso del algoritmo de ordenamiento.

Ejemplos de Visualización en Consola

Bubble Sort (Orden Ascendente, Paso a paso)

Antes de iniciar el ordenamiento, verías algo así:

Conjunto inicial: [5, 3, 8, 1, 2]

Durante el proceso, se mostrarían los intercambios paso a paso:

Paso 1: [3, 5, 1, 2, 8]

Paso 2: [3, 1, 2, 5, 8]
Paso 3: [1, 2, 3, 5, 8]
Ordenamiento completado.

Quick Sort (Orden Descendente, Paso a paso)

Conjunto inicial: [4, 6, 2, 8, 1]
Dividiendo por pivote (6):
Paso 1: [6, 4, 8, 2, 1]
Dividiendo por pivote (4):
Paso 2: [6, 4, 8, 2, 1]
...
Ordenamiento completado: [8, 6, 4, 2, 1]

Reporte

Luego de ordenar los datos se deberá de generar un reporte general del ordenamiento. El reporte se deberá de ir generando en un archivo .txt donde se visualicen al menos los siguientes datos:

1. Tipo de ordenamiento
2. Conjunto inicial de datos
3. Conjunto ordenado
4. Tiempo de ejecución

Además, el reporte en la parte superior deberá de incluir el encabezado con su información y la fecha y hora de creación.

Conjunto inicial: [12, 4, 9, 2, 7]
Ordenando con Bubble Sort...
Conjunto ordenado: [2, 4, 7, 9, 12]
Tiempo de ejecución: 150 ms

Ordenando con Quick Sort...
Conjunto ordenado: [2, 4, 7, 9, 12]
Tiempo de ejecución: 70 ms

Ordenando con Merge Sort...
Conjunto ordenado: [2, 4, 7, 9, 12]
Tiempo de ejecución: 85 ms

Finalizar programa

Esta opción del menú termina con la ejecución del programa.

Criterios de evaluación:

- Correcto funcionamiento de la aplicación en sistemas ARM64.
- Manejo adecuado de errores y validaciones.
- Interacción fluida con el usuario a través de la consola.
- Se debe validar que los números ingresados sean correctos.

Puntos Importantes A Tomar En Cuenta:

- ➔ Se trabajará con números enteros positivos y negativos.
- ➔ Es importante que el encabezado con sus datos ya que de lo contrario se considera copia.
- ➔ La práctica se desarrollará de forma individual.

Entrega:

- La entrega será por medio de la plataforma UEDI adjuntando el link del repositorio de GitHub.
- El nombre del repositorio deberá ser el siguiente:
 - ACYE1_SECCION_2S24_#carne
 - **Ejemplo:** ACYE1_A_2S24_202002794
- Crear una carpeta llamada Práctica 4, ya que el mismo repositorio se estará utilizando para el resto del contenido de ARM
- Debe existir más de un commit.
- Realizar el último commit y hacer su entrega en UEDI antes de las 23:59 horas del 03 de octubre.
- Se debe agregar al auxiliar de su sección como colaborador del repositorio.
 - Sección A: hugosmhd
 - Sección B: jcas1020

Entregables:

- Código de usado para el desarrollo de la práctica.
- Manual Técnico con la descripción del funcionamiento de cada característica en formato Markdown o PDF.
 - Agregar capturas del funcionamiento.
 - Agregar cualquier otro aspecto que consideren importante, explicación de código de ser necesario, etc.

Observaciones Y Restricciones:

- La práctica será desarrollada de forma INDIVIDUAL.
- El día de la calificación se harán preguntas sobre la elaboración de la práctica las cuales se considerarán en la nota.
- Las copias parciales o totales tendrán una nota de 0 puntos y los involucrados serán reportados a la Escuela de Ciencias y Sistemas, incluyendo ambas secciones.
- Se revisan que no existan copias de prácticas entre ambas secciones.
- Entregas tarde serán penalizadas con el 25% de la nota obtenida por cada 15 minutos después de la hora límite.