

Práctica 4

Objetivos

Objetivo General

- Aplicar los conocimientos adquiridos en el curso sobre el lenguaje ensamblador

Objetivos Específicos

- Aplicar el conocimiento de operaciones básicas a nivel ensamblador
- Conocer el funcionamiento de las interrupciones
- Comprender el uso de la memoria en los programas escritos en ensamblador
- Aplicar el manejo de archivos a bajo nivel
- Aplicar el uso de registros bandera
- Manejar entrada, proceso y salida de datos numéricos en ensamblador
- Adquirir y aplicar conocimiento sobre manejo de números 16 bits con signo.

Descripción

Se requiere el desarrollo de una aplicación en consola, utilizando programación a bajo nivel (lenguaje ensamblador), que tenga las funcionalidades de una calculadora que realice las siguientes operaciones:

- Suma
- Resta
- Multiplicación
- División
- Potenciación

Además de poder obtener los datos estadísticos de los números ingresados, por medio de archivo o directamente; estos serán:

- Media
- Mediana
- Moda

- Cantidad de números pares
- Cantidad de números impares
- Cantidad de numeros tipo
 - Fibonacci
 - Lucas

Especificaciones

Símbolos a utilizar

Operación	Símbolo1	Símbolo2
Suma	+	mas
Resta	-	menos
Multiplicación	*	por
División	/	entre
Potenciación	^	potencia

Estadísticos

Estadístico	Observaciones
Media	Salida parte entera
Mediana	Salida parte entera
Moda	
Números pares	
Números impares	
Números Fibonacci	Se debe calcular
Números Lucas	Se debe calcular

Los estadísticos se calculan sobre las entradas recibidas y no sobre los resultados, finales o parciales, de las operaciones.

Desarrollo de la práctica

Identificación del desarrollador

Cada vez que se ejecute la práctica se deberá mostrar la identificación del desarrollador, ejemplo:

```
Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Arquitectura de Compiladores y ensambladores 1
Sección <A|B>
< Nombre del desarrollador >
< Registro Académico >
```

Luego de la identificación deberá esperar a que se presione Enter para poder continuar hacia el menú.

Menú

El menú del desarrollo debe incluir las siguientes opciones

1. Calculadora
2. Archivo
3. Salir

Calculadora

Mostrará en consola la opción de realizar operaciones matemáticas.

Archivo

Permitirá introducir la ruta del archivo de entrada en el que se almacenaron las operaciones a realizar.

Salir

De no haber generado el reporte de ejecución (explicado más adelante) se generará en este momento y terminará la ejecución del programa.

Detalle Funciones del menú

Calculadora

Permitirá realizar operaciones matemáticas, desde línea de comandos, en notación infija e ingresar instrucciones que muestran reportes específicos.

las respuestas deben estar identificadas con el caracter »

Ejemplos de operaciones directas de calculadora

Operación: $8 + 25$

Calculadora

> 8+25

Otra forma

Calculadora

> 8 mas 25

Resultado

Calculadora

» 33

Operación: $((12 + (8^2)) / 4)$

Calculadora

> ((12 mas (8 ^ 2)) entre 4)

Resultado

Calculadora

» 19

>

A la hora de alizar una operación no debe limpiarse la pantalla.

Instrucciones reconocidas

Instrucción	Función
RESET	<ul style="list-style-type: none">- Reiniciará todos los estadísticos tomados hasta el momento- Iniciará con una nueva operación- Si se ha cargado una operación desde archivo volverá a la posibilidad de usar la consola como calculadora o evaluar un nuevo ID
MOSTRAR MEDIA	Mostrará la media de los números reconocidos al momento
MOSTRAR MODA	Mostrará la moda de los números reconocidos al momento
MOSTRAR MEDIANA	Mostrará la mediana de los números reconocidos al momento
MOSTRAR IMPARES	Mostrará la cantidad de números impares reconocidos
MOSTRAR PARES	Mostrará la cantidad de números pares reconocidos
MOSTRAR FIBONACCI	Mostrará la cantidad de números Fibonacci reconocidos
MOSTRAR LUCAS	Mostrará la cantidad de números Lucas reconocidos
EXEC [ID]	<ul style="list-style-type: none">- [ID] será reemplazado por el nombre de la operación cargada desde archivo- Se analizará la operación en cuestión y realizará todos los cálculos- Mostrará el resultado de la evaluación completa de la operación- quedará en espera de la instrucción MOSTRAR deseada- no podrá realizar cálculos matemáticos si se ha cargado un ID
SALIR	Salir al menú principal

Ejemplo de instrucciones:

Calculadora

> EXEC oper1

El resultado es: 33

Calculadora

> MOSTRAR media

La media desde archivo es: 7

Si se están haciendo cálculos y se usa alguna instrucción MOSTRAR solo se podrán seguir haciendo cálculos si se ingresa RESET para volver a empezar de cero con una nueva operación.

Si se trata de una operación desde archivo se refiere únicamente a los datos de la operación referenciada por nombre, aunque el archivo contenga varias operaciones.

Si se está manipulando un archivo, los estadísticos se refieren los números identificados en todas las operaciones que en él se encuentren; si se llama a una operación como parte de otra, el resultado de la operación llamada no debe ser incluido en los estadísticos.

Si existiese más de una moda se tomará la primera.

Archivo

Permite ingresar el nombre del archivo que contiene las operaciones a evaluar

Desde archivo

Nombre de archivo: _

El contenido del archivo será XML, el padre siempre será "calculos" y cada operación deberá ser identificada con "operación", el nombre de la operación con "nombre" y la operación a evaluar con "contenido".

Ejemplo del archivo de entrada

```
<calculos>
  <operacion>
    <nombre>oper1</nombre>
    <contenido>8 + 25</contenido>
  </operacion>
  <operacion>
    <nombre>ejemplo2</nombre>
    <contenido>((12 + (8 potencia 2)) entre 4)</contenido>
  </operacion>
  <operacion>
    <nombre>calculos</nombre>
    <contenido>((5 mas 9) * 2) + (6*5) - oper1</contenido>
  </operacion>
</calculos>
```

Reporte de ejecución

Se generará un archivo con contenido de tipo XML que detalle los siguientes aspectos:

- Datos del alumno
 - Nombre
 - Carnet
 - Curso
 - Sección
- Fecha de generación de reporte
 - Día
 - Mes
 - Año
- Hora de generación de reporte
 - Hora
 - Minuto
 - Segundo
- Resultados:
 - media
 - mediana
 - moda
 - pares
 - impares
 - fibonacci
 - lucas
- Operaciones:
 - Resultados de cada operación

Ejemplo:

```
<reporte>
  <datos>
    <nombre>Nombre del estudiante</nombre>
    <carnet>20XYABCDE</carnet>
    <curso>Arquitectura de computadores y ensambladores 1</curso>
    <seccion>A|B</seccion>
  </datos>
  <fecha>
    <dia>09</dia>
    <mes>08</mes>
    <anio>2022</anio>
```

```
</fecha>
<tiempo>
  <hora>07</hora>
  <minuto>02</minuto>
  <segundo>47</segundo>
</tiempo>
<estadisticos>
  <media>7</media>
  <mediana>6</mediana>
  <moda>8</moda>
  <fibonacci>6</fibonacci>
  <lucas>3</lucas>
</estadisticos>
<operaciones>
  <oper1>33</oper1>
  <ejemplo2>19</ejemplo2>
  <calculos>25</calculos>
</operaciones>
</reporte>
```

En el caso que no se haya cargado un archivo de entrada, en operaciones habrá únicamente una y su resultado será el último generado por la calculadora.

Siempre se mostrarán los resultados del último análisis generado, ya sea de archivo o de cálculos directos.

Los cálculos directos también generan los datos estadísticos.

Referencias

Titulo	Referencia
Tabla ASCII	https://elcodigoascii.com.ar
Libro de Texto	Brey, Barry B. Microprocesadores Intel 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, and Pentium 4 Arquitectura, Programación e Interfaz. Pearson Prentice Hall. Séptima Edición.
Libro sugerido	Assembly language for x86 processors. Irvine - Pearson - 2015

Entregables

- Manual Técnico (elaborado en MD).
- Manual de usuario (elaborado en MD).
- Código funcional
- Link al repositorio de gitlab.

Entrega

- La entrega, en UEDI, será el link al repositorio de gitlab
- El nombre del repositorio responde a la siguiente estructura:
 - ACE1-
 - Año
 - Período
 - Código curso
 - Sección
 - Número de carnet.
 - PRAC4

ej. ACE1-222S0778B200012345PRAC4

- Realizar el último commit y hacer su entrega en UEDI antes de 23:59 horas del 23 de septiembre.
- Se ejecutará un checkout hacia el último commit hecho antes de la fecha de entrega.
- Se debe agregar al auxiliar de su sección como miembro del repositorio
 - Sección A: @Orbp1403
 - Sección B: @pinedaMario

Observaciones y Restricciones

- Lectura de archivo (para el modo archivo)
- Generación de reporte
- Se debe entregar los manuales técnico y de usuario, de lo contrario se asumirá que el estudiante copió.

Debido a la naturaleza de ensamblador, los resultados completos solo se pueden evaluar si se logran cumplir con los dos primeros puntos

- **La realización de la práctica es de forma individual**
- Se evaluarán números de 16 bits **con signo**
- Todas las acciones que amerite mostrar mensajes de error deberá mostrarlo, de lo contrario tendrá penalización en puntos.
- El código del programa debe ser estrictamente ensamblador, no se permite el uso de alguna librería.
- Los números Fibonacci o Lucas no pueden estar quemados en el código, se debe implementar un algoritmo para su identificación.
- Los números de fecha y tiempo del reporte, en caso de ser menores a 10, deben incluir un cero antes del número (ej. 9 -> 09).
- Se debe presentar el proyecto en Dosbox.
- Lenguaje ensamblador a utilizar MASM 6.11
- No está permitido el uso de estructuras de control if o if else, while, repeat, for.
- No está permitido el uso de STRUCT

- El día de la calificación se harán preguntas, modificación de código sobre aspectos utilizados en la elaboración del proyecto, las cuales se considerarán en la nota final.
- **Copias parciales o totales tendrán una nota de 0 puntos y los involucrados serán reportados a la Escuela de Ciencias y Sistemas**