**BITACORA CÓDIGO ENSAMBLADOR CALCULADORA**

Iván Darío Guerrero Chávez

Código: 1.004.731.375

Nicolas Ceballos Arias

Código: 1.098.306.373

Este es un código ensamblador para la ejecución de una calculadora básica (+, -, \*, /) desarrollada en un entorno como Linux. A continuación, la explicación del código y su función.

**Section .data y .bss**

1. .data: Define un mensaje para mostrar al usuario.

* msg\_intro, prompt\_num1: contiene los mensajes de bienvenida, de entrada, de texto y de números, entrada y salida.

1. .bss: Espacio de memoria para variables.

* num1 y num2: Guardar los números dados.
* result: Guardar el resultado de la operación.
* buffer: Entrada del usuario.
* operation: guardar el operador (+, -, \*, /)

**Section .text y para iniciar \_start**

1. \_start: Mensaje de bienvenida e inicia un bucle de cálculo (loop).

**Bucle principal (loop)**

1. Solicita el primer número:

* Muestra el mensaje prompt\_num1, lee la entrada en num1, y llama a parse\_number1 para convertir el texto en un número entero.

1. Solicita el segundo número:

* Similar al primer número.

1. Solicita la operación.

* Muestra el mensaje prompt\_op, lee el operador en operation, y verifica qué operación debe realizarse.

**Operaciones**

Para cada operación, el código evalua operation.

1. Suma: Toma num1 y num2 e eax y muestra el resultado.
2. Resta: Mismo procedimiento que la suma
3. Multiplicación: imul para multiplicar los valores.
4. División: Verifica si num2 es cero (para evitar la división por cero), y si no, divide num1 entre num2.
5. Modulo: Calcula el residuo de num1 dividido entre num2 usando idiv y almacenando el residuo en eax.

Si la operación es invalida se muestra un mensaje de error (error\_op) y se repite el bucle.

**Mostrar el resultado**

1. Convierte el resultado en eax de número a ASCII.
2. Muestra msg\_res seguido del result.

**Continuar y salir**

1. Muestra el mensaje exit\_msg, y lee la entrada del usuario.
2. Si el usuario escribe "exit", salta a salir, termina el programa, y cierra con el sistema (int 0x80).

**Funciones para convertir (parse\_number 1, parse\_number2)**Se convierte de ASCI a entero.

1. parsi\_number1: Lee num1, detecta signo negativo, y convierte a número entero.
2. parsi\_number2: Realiza lo mismo que el anterior.