Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

IC-3101 Arquitectura de Computadores

Examen I

Profesor:

Esteban Arias Méndez

Estudiante:

Luis Carlos Araya Mata

Carné:

2020205005

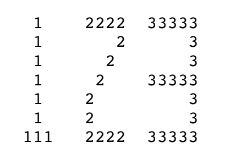
Fecha de Entrega:

24/04/2021

Enunciado:

Usando el lenguaje ensamblador NASM, escriba un programa completo tal que reciba un número positivo o negativo de máximo 16 bits de representación.

Su programa deberá mostrar dicho número ingresado como un banner en modo texto, como el ejemplo siguiente: Para una entrada: “123” mostrará un banner similar a:



Observe que cada dígito está compuesto por caracteres del mismo dígito, además todo el banner se muestra de forma horizontal, no se deben mostrar los valores de forma vertical. En caso de los números negativos, debe mostrarse el guión de menos - en el banner.

En caso de que se reciba un valor mayor al rango de 16 bits para positivos y negativos deberá mostrar el error de rango en pantalla y preguntar al usuario si desea intentarlo de nuevo.

Debe explicar en su reporte PDF la forma de organización de los datos requeridos, explicar en sus palabras la estrategia para armar el banner de la forma solicitada. Su solución debe tomar en cuenta el ahorro de recursos de memoria,  usando la menor cantidad de datos en memoria que considere posible, así como una ejecución eficiente del programa.

El diseño de cada digito puede cambiar, pero cada digito debe ser formada por los caracteres del mismo dígito que representan. Además, en el programa, debe almacenarse una única copia de cada dígito, es decir, por ejemplo, en la cifra 121, el dígito 1 se usa 2 veces, pero el texto para el dígito 1 debe ser almacenad1 en el programa una única vez.  Esto para hacer más pequeño y eficiente el programa.

Para incentivar el código eficiente y la buena programación, los 5 programas ensamblados más pequeños, que sean funcionales y cumplan con todos los aspectos de la especificación de esta tarea serán un cuadro de honor en las notas pudiendo merecer un 100 en esta actividad, si todo está según lo solicitado acá.

Solución:

Descarte inicial:

El primer problema al que me enfrenté fue a como iba a imprimir los números, ya que según como vi en el enunciado, el resultado final abarca varias líneas de la terminal, por lo que descarté de una vez el hecho de imprimir de una vez el dígito que voy leyendo, debido a que eso implicaría subir y bajar en las líneas de la terminal, lo cuál no sé realizar entonces opté por descartar esta opción.

Diseño de los números:

El diseño de los números los realicé de una altura máxima de 5 líneas, esto con el fin de facilitarme el hecho de que se descuadraran los números y evitar validaciones y código extra para dibujar cada número. El diseño de cada número está realizado de tal manera que se pueda reutilizar cada línea de diseño con el fin de evitar crear más variables y así ahorrar memoria. Cada línea se representa como un string para poder ser utilizado a la hora de imprimirlo en la línea y que quede bien.

Variables:

En la declaración de variables de la sección “.DATA”, están el mensaje de inicio que corresponden a desplegar los mensajes para que el usuario ingrese el número y el mensaje final que le muestra al usuario cuál es el número.

Están declaradas cada “capa” que representa cada línea de cada número, separadas por número.

En la declaración de variables de la sección “.UDATA” está solamente reservada la variable para el número que ingresa el usuario.

Core del Algoritmo:

Debido a que cada número está diseñado de una altura máxima de 5 líneas, implementé 2 ciclos, uno anidado dentro de otro. Procedo a explicar qué hace cada uno:

Ciclo de afuera:

Este ciclo es de tipo “loop”, lo cuál va condicionado por el valor del registro CX, el cuál iniciando el programa corresponderá a 5, debido a que esta es la altura máxima del diseño de los números en cuanto a las líneas. En fin, este ciclo nos dicta cuantas veces vamos a leer el número entero, ósea siempre van a ser 5, debido a que debemos imprimir las 5 líneas de capa de cada número. Aquí tuve que recurrir a realizar saltos de poco a poco debido a la extensión del código, ya que como se sabe, la directiva loop tiene un límite de distancia de líneas.

Ciclo de adentro:

Es condicional, y es el que me permite leer cada dígito del número. Aquí se pregunta cúal dígito del 0 al 9 y también el guión incluido es el dígito que se está analizando. Una vez se sabe qué digito es, se le pregunta al CX su valor por medio de una comparación para saber por cuál línea se va recorriendo, esto dicta con base a cada número cuál línea del número se debe pintar. Se procede a imprimir la línea que corresponde, y se reinicia el ciclo, para seguir moviéndome por cada carácter del número y termina hasta que se recorra el último carácter del número.

Entonces al final cada línea se va imprimiendo hasta llegar a la quinta que sería la última. En ese entonces ya el resultado estará en pantalla.