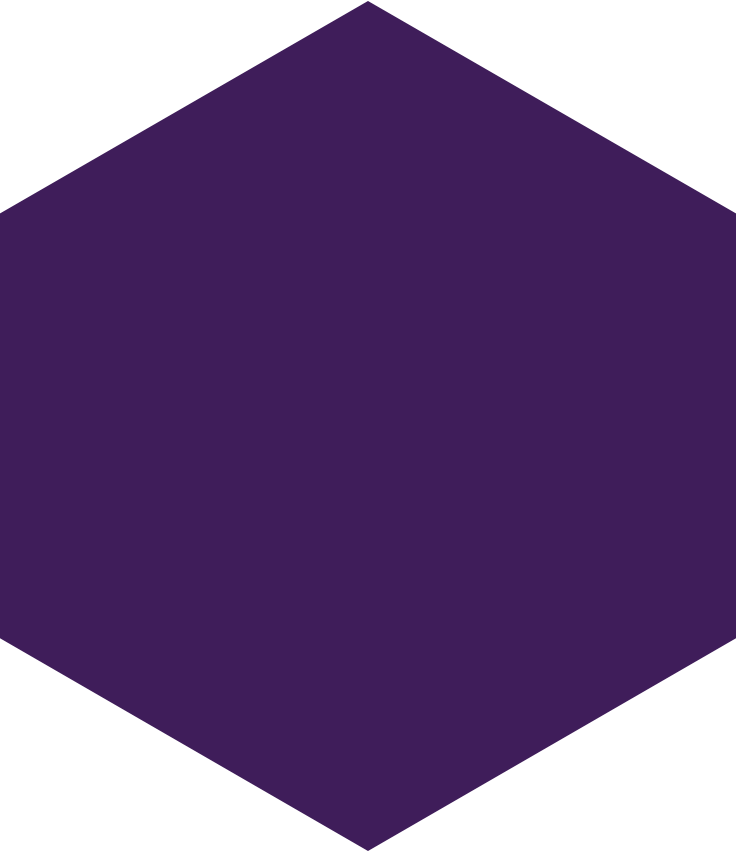
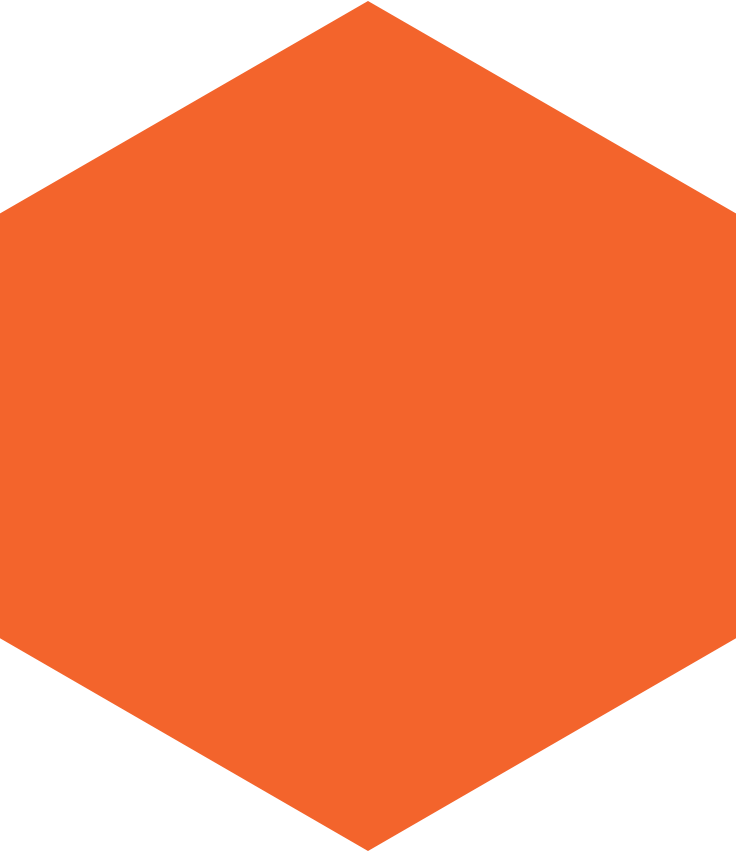
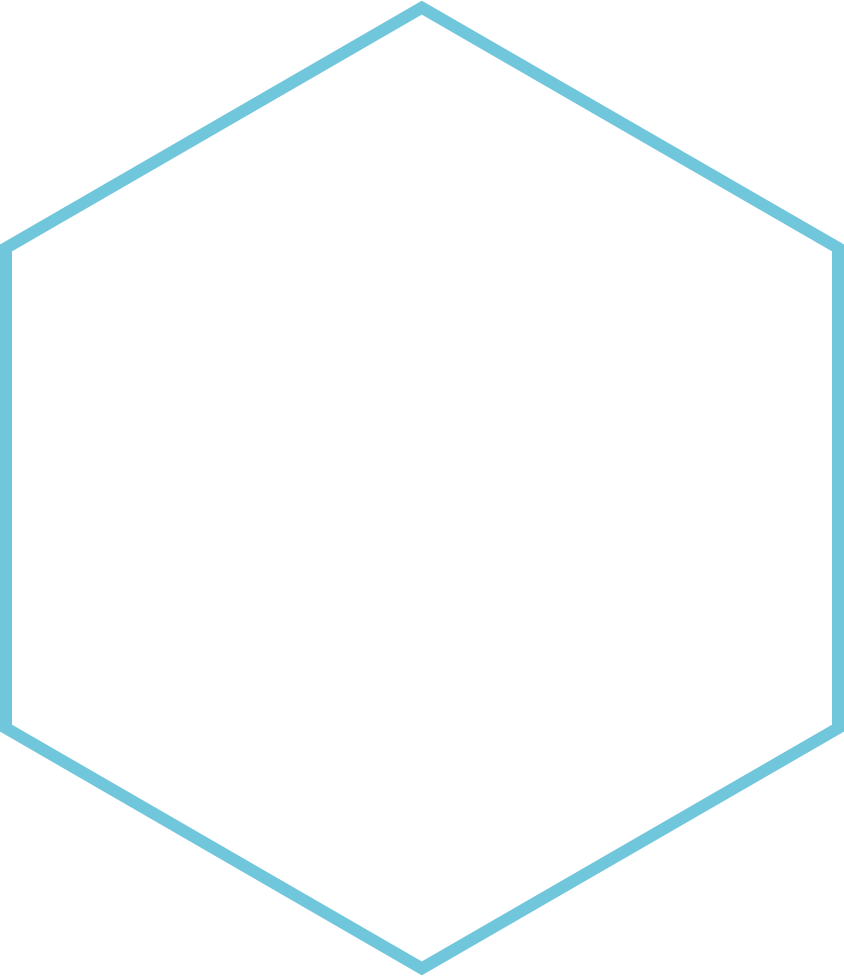
|  |
| --- |
| **Arquitectura y Organización de Computadores** |
| Taller 1 Desarrollado en Emu8086 |
| Desarrollador: Rodrigo Dominguez M.  Profesor: Ricardo Pérez S. |
|  |





|  |
| --- |
| Introducción. |
|
| Objetivo. |

El objetivo del siguiente taller es programar en bajo nivel utilizando para ello un emulador de procesador Emu8086, además de entender el manejo de interrupciones del sistema.

Contexto del taller.

El programa desarrollado en este taller consta de las siguientes funciones:

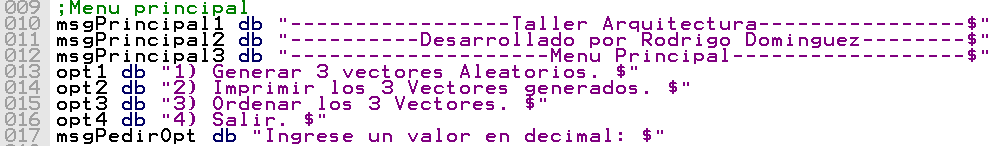
* Un menú principal con 4 opciones:
  + La opción 1 genera 3 vectores de 7 valores aleatorios entre 0 y 99.
  + La opción 2 imprime los 3 vectores generados en la opción 1
  + La opción 3 reordena los 3 vectores de acuerdo con el algoritmo siguiente:
    - Se desplazan los 21 valores un espacio hacia abajo en la memoria del computador y el ultimo valor debe ser llevado a la primera posición.

Este algoritmo debe realizarse las veces necesarias para que la suma del vector 1 sea menor a la del vector 2 y a la vez la suma del vector 2 sea menor a la del vector 3.

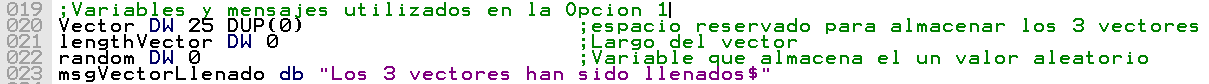
* + La opción 4 termina el programa.

Explicación de las funciones utilizadas.

* Variables y Mensajes.



En esta posición del código se almacenan los mensajes utilizados en el menú principal.

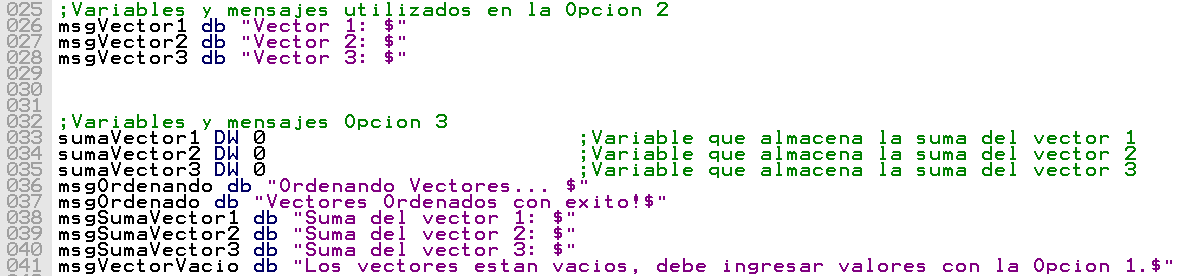


En esta posición almacenamos las variables y mensajes utilizados en la opción 1.

Donde la variable Vector almacena 25 posiciones para no tener problemas en el algoritmo de ordenación.

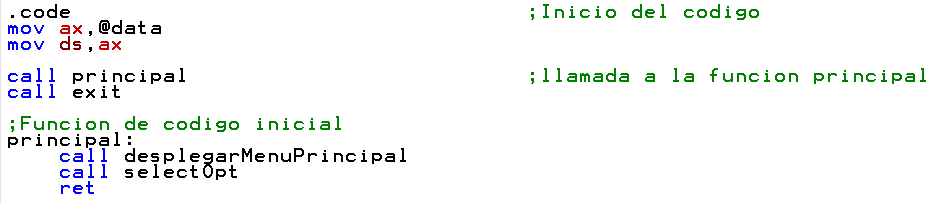
La variable lengthVector es utilizada para almacenar el tamaño del vector.

La variable random es utilizada para almacenar el numero aleatorio.

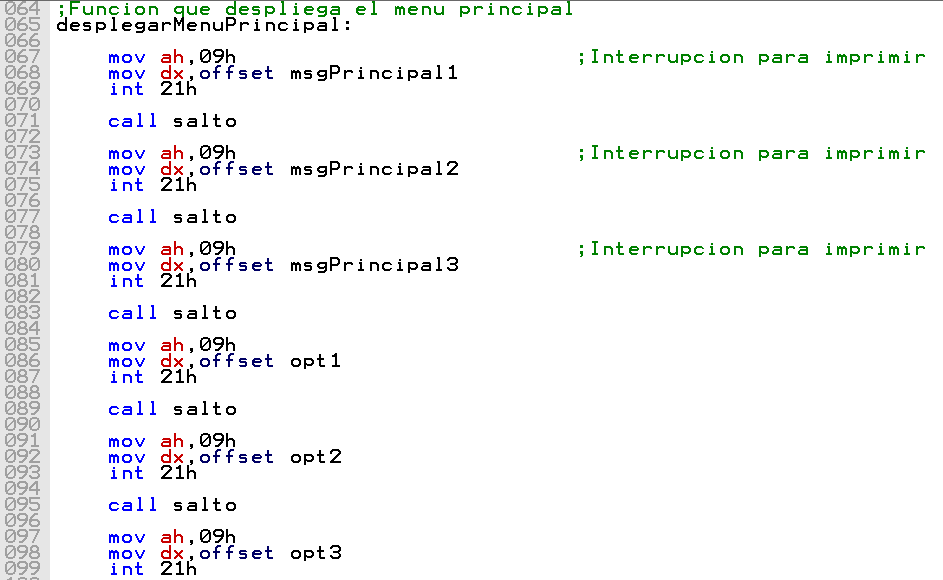


En esta posición se almacenan los mensajes y variables de las opciones 2 y 3.

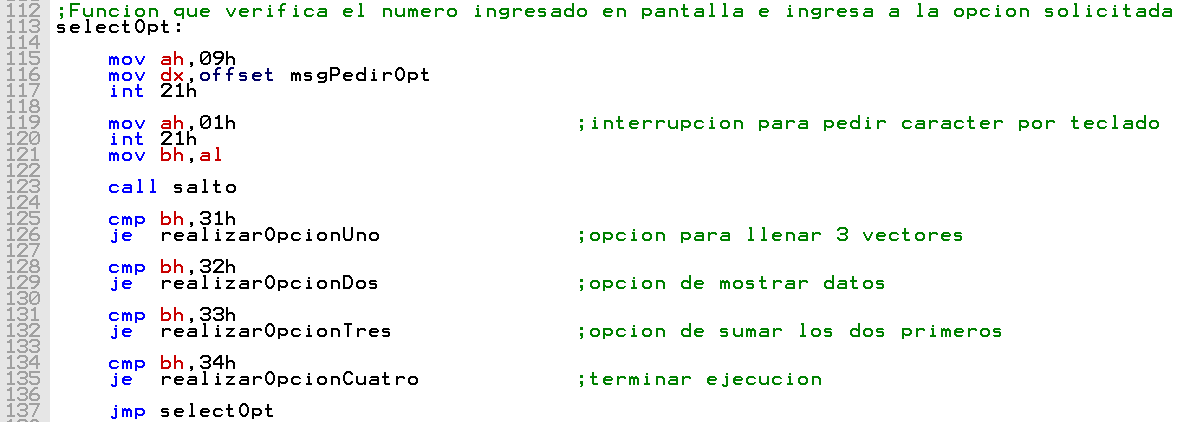
La variable sumaVector# es utilizada para almacenar las sumas de los vectores 1, 2 y 3.



Inicio del programa (esta es la función principal donde se mantiene el ciclo del programa)



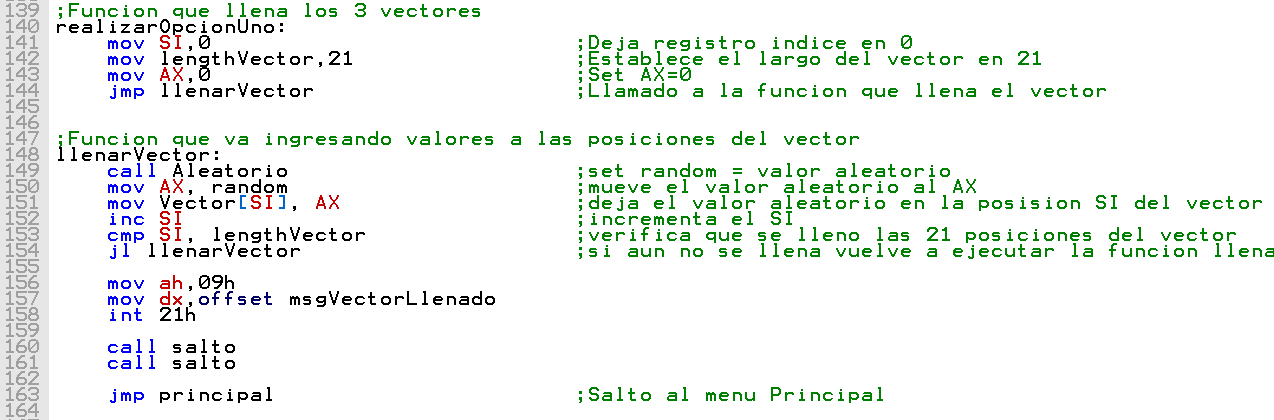
Función que despliega el menú principal utilizando la interrupción 21h junto a la función 09H que permite pasar al registro AX una cadena de caracteres e imprimir por pantalla dicha cadena de caracteres.

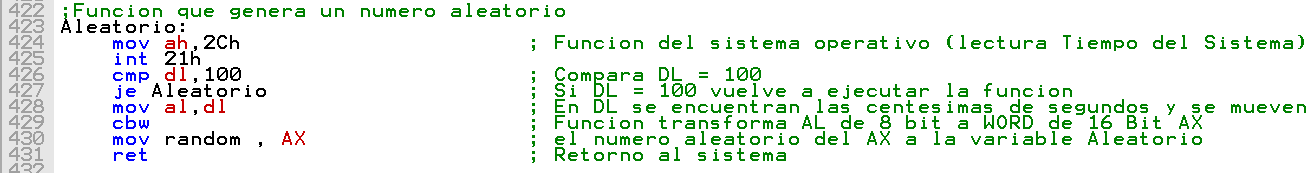


Función para verificar el ingreso de un numero por teclado utilizando la interrupción 21h junto a la función 01h que permite la entrada de un carácter almacenándolo en el registro AX en la parte Low (AL).

Dicho carácter ingresado es comparado con los valores en ascii (1,2,3 y 4) realizando las llamadas correspondientes a las opciones 1, 2, 3 o 4.

Mediante la instrucción CMP verificamos el carácter y mediante la instrucción JE saltamos a dicha opción si es igual al carácter.



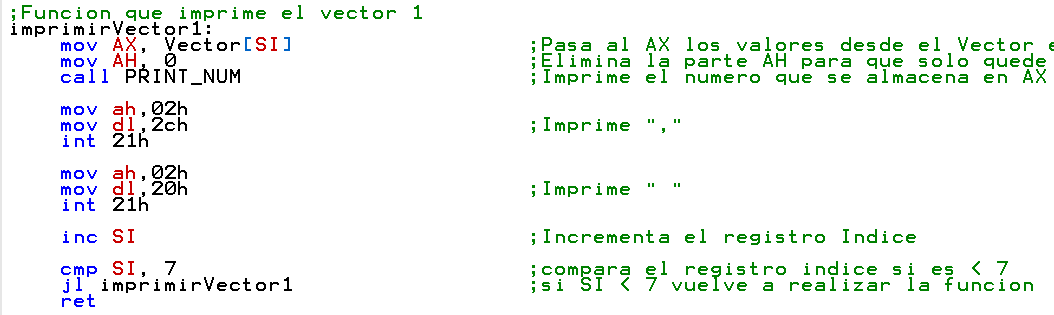
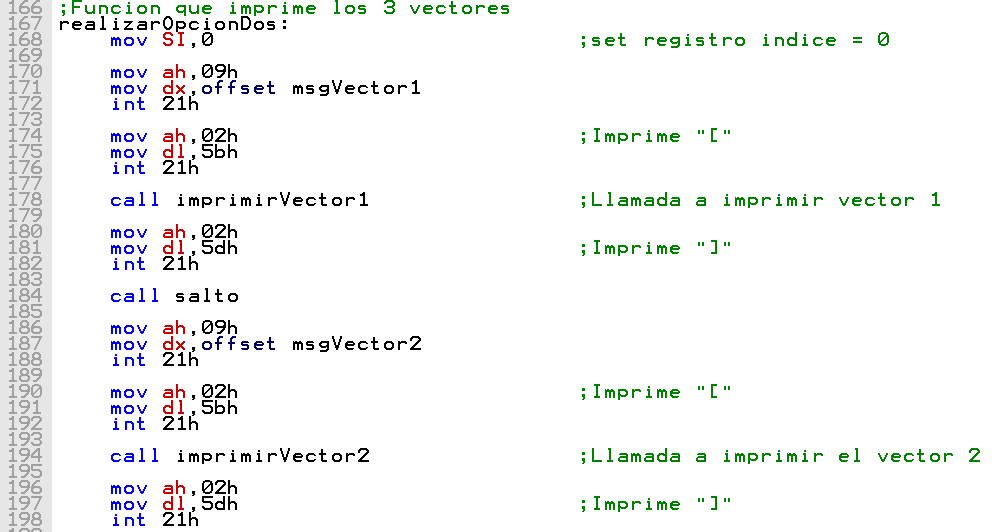


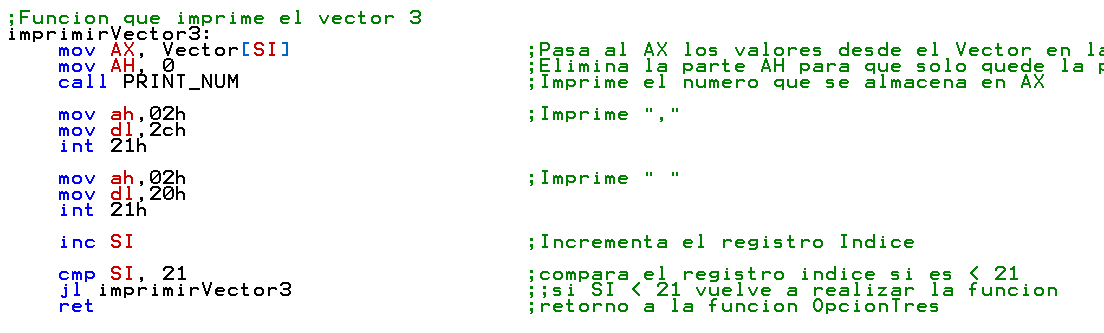
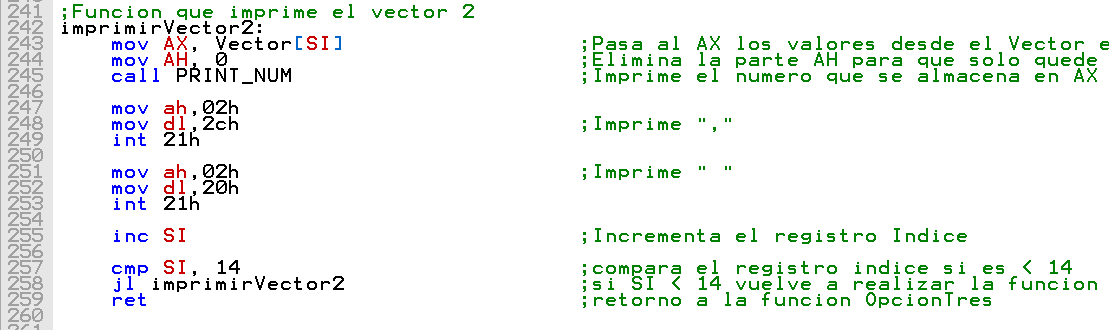
La opción 1 que permite llenar los 3 vectores con valores aleatorios, hace uso de 3 Funciones.

* La Función “realizarOpcionUno” setea los registros AX y SI a 0, setea la variable lengthVector y llama a la función llenarVector.
* La función llenarVector hace uso de la función Aleatorio la cual genera un valor aleatorio y lo coloca en la variable random, luego va recorriendo las 21 posiciones del vector para ir llenando cada valor con un numero aleatorio.

Las posiciones del vector se van recorriendo gracias al uso del registro SI el cual va aumentando en cada ciclo, utilizando la instrucción CMP y JL verificamos que se a llenado todo el vector (las 21 posiciones).

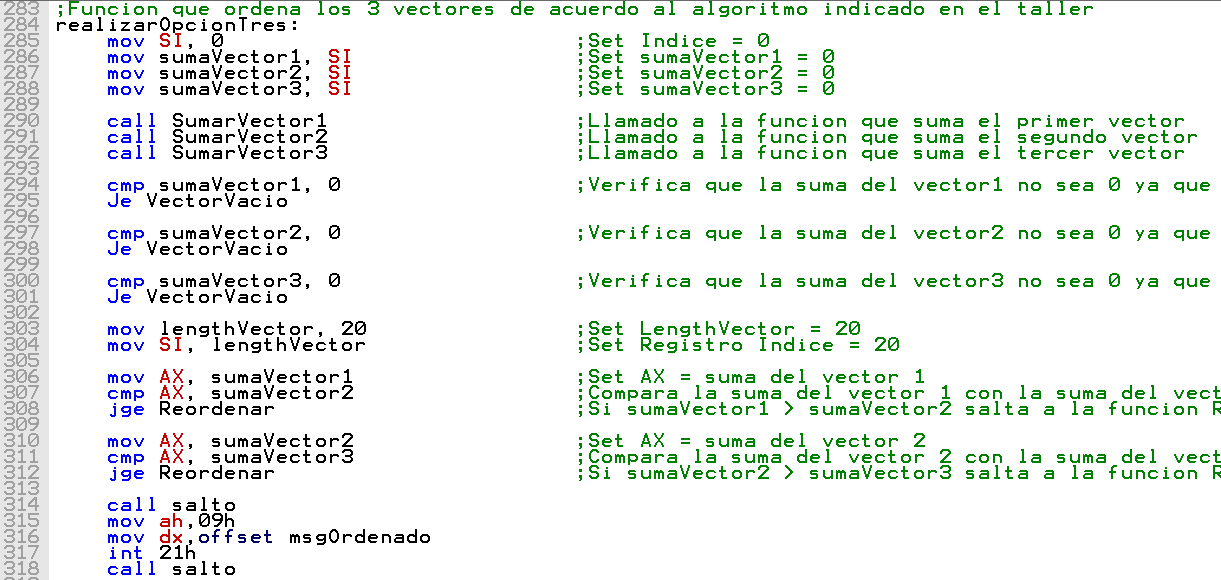
* La función Aleatorio hace uso de la interrupción 21H junto a la función 2CH la cual coloca en los registros AH, AL, CH, CL, DH, DL los valores correspondientes a la fecha y hora del sistema, en donde en el registro DL podemos encontrar las centésimas de segundo, al ser este valor un numero que se actualiza con mucha rapidez es técnicamente un numero pseudoaleatorio.

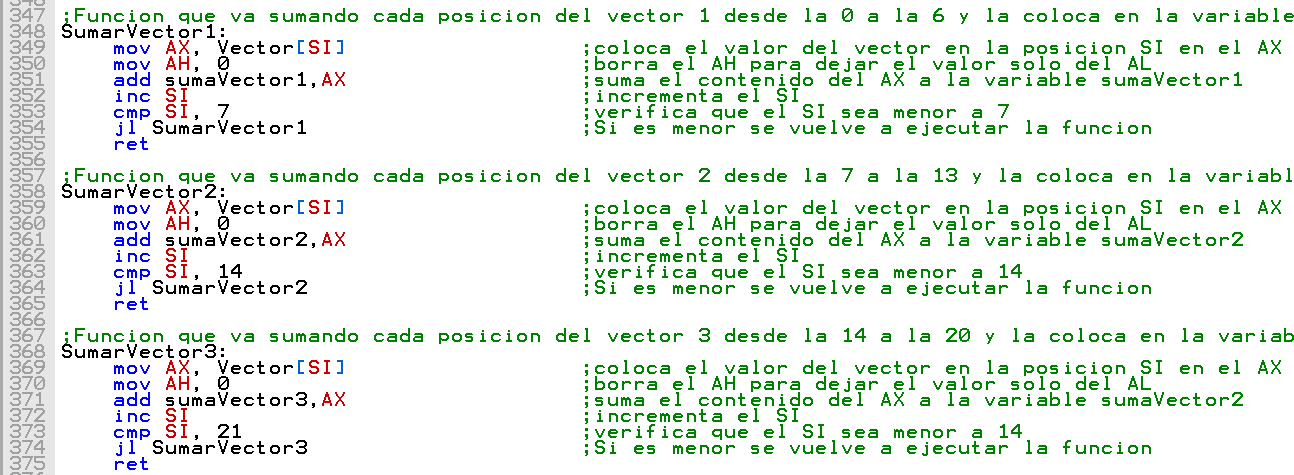


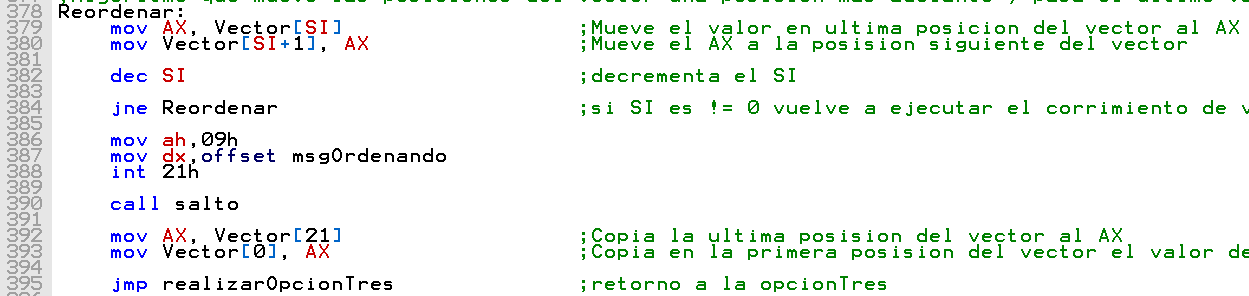


La opción 2 que permite imprimir los 3 vectores consta de 4 funciones:

* La función realizarOpcionDos es el ciclo principal de la función la cual llama a las otras 3 funciones para realizar las búsquedas e impresión por pantalla de los números. Podemos observar que esta función hace uso de la interrupción 21H junto a las funciones 09H y 02H las cuales permiten imprimir en pantalla una cadena de caracteres y un carácter por si solo.
* Las funciones ImprimirVector1, 2 y 3 van recorriendo las posiciones del vector desde la 0 a la 20 y utilizando la función PRINT\_NUM vemos por pantalla el valor que almacena el vector.

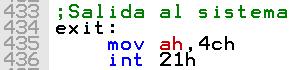






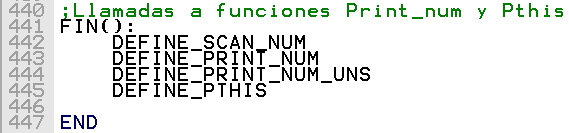
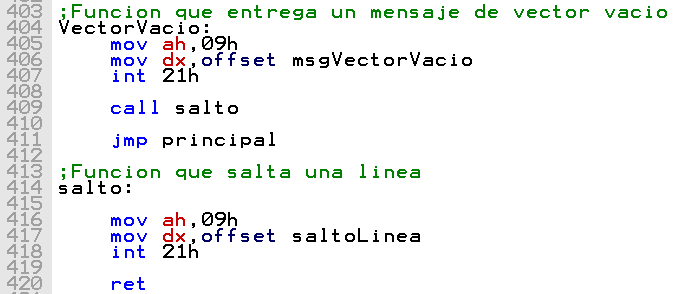
La opción 3 que permite reordenar el vector mediante un algoritmo de ordenamiento el cual se basa en correr todos los valores una posición hacia abajo y colocar el ultimo valor al principio, consta de 5 funciones:

* La función realizarOpcionTres permite llamar a las funciones de suma y luego verificar si dichas sumas están ordenadas, en caso de que no esten ordenadas se volverá a ejecutar esta función las veces necesarias hasta que el vector este ordenado.
* Las funciones SumaVector1, 2 y 3 recorren el vector principal desde las posiciones 0 a 6 y suman cada valor de cada posición añadiéndolo a la variable sumaVector1, luego realiza el mismo procedimiento desde la posición 7 a la 13 y por último desde la posición 14 a la 20
* Luego de que se realizan las 3 sumas se utiliza la función CMP y JGE para verificar que las sumas esten en el orden deseado (Suma vector 1 < suma vector 2 < suma vector 3) en caso de no estarlo se llama a la función reordenar.
* La función reordenar hace uso de el registro SI desde la posición 20 hasta la 1 utilizando la instrucción DEC que va decrementando el valor de dicho registro, donde cada posición desde la 20 va almacenándose en la posición contigua, por ultimo el valor en la posición 21 es llevado a la posición 0 completando el algoritmo de reordenamiento 1 vez.



La opción 4 es un retorno al sistema operativo utilizando la interrupción 21H junto a la función 4CH la cual termina con la ejecución del bloque de código.

* Funciones auxiliares utilizadas:



* La función VectorVacio verifica que el vector no contenga valores vacíos antes de realizar un ordenamiento.
* La función salto imprime por pantalla un salto de línea.

Se adjunta código en formato .asm junto a este informe.