|  |  |
| --- | --- |
| **Solucionado** |  |
| **En proceso** |  |
| **Sin corregir** |  |

**Puntos a tratar**

**Contador (Codigo fuente)**

1. El Contador del stack segment debe UNICAMENTE debe incrementarse con líneas del tipo

“dw \*numero\* DUP(\*Numero o string\*)”. No se debe incrementar si:

En vez de dw es db

En lugar de DUP es DOBLE

1. En el code segment el contador NO debe incrementarse con líneas que estén marcadas como “Error”

**Codificacion**

1. Crear un diccionario de datos para almacenar cada etiqueta con su respectiva direccion.
2. A las instrucciones de salto, agregar la dirección de la etiqueta en la codificación
3. Agregar una validación para que aquellas instrucciones de salto que usen una etiqueta que no exista o este definida después de esta instrucción, se marque como “Error”
4. Crear un diccionario de variables, que incluya el nombre de las variables con su respectiva direccion. (Las variables que están en la tabla de símbolos junto con su respectiva direccion)

**Errores dados en la última versión**Se descarto la versión anterior. Se desarrollo una nueva versión. En la versión anterior, el programa no contemplaba el metodo generate\_machine\_code

**La nueva versión presenta los siguientes errores:**

- En el Código fuente, el programa no valida como erróneas aquellas variables cuyo nombre supera los 10 caracteres.

- En el Código fuente y en la tabla de símbolos, el programa no detecta correctamente las constantes.

- No hay correspondencia entre las validaciones en el Código fuente y en la tabla de símbolos. Es decir, hay variables que están clasificadas como error en el código fuente, pero en la tabla de símbolos si son consideradas, y viceversa.

- Las variables en la tabla de símbolos no tienen bien asignada su dirección

- Hay instrucciones de salto que pese a estar marcadas como error, aumentan el contador. Es probable que esten aumentando el contador debido a que manejan etiquetas que existen en el diccionario.

**IMPORTANTE**

1. Añadir mas validaciones y mejorar la codificación de instrucciones para que no marque como error instrucciones que hagan uso de variables. (La línea de la imagen debería codificarse)

Es importante revisar que las instrucciones tengan el numero correspondiente de operandos, por ejemplo, en la imagen CMP puede usar dos operandos, sin embargo, no los esta reconociendo correctamente y marca la línea como error. Revisar porque esto sucede y dar solución.

Para las instrucciones que hacen uso de 2 operandos, dichas instrucciones tienen que funcionar cuando alguno de los operandos es una variable que está declarada dentro del data segment, y si la variable existe y esta en el data segement, en la codificación de la instrucción se debe incluir la direccion de la misma. De forma similar a como se hizo con las instrucciones de salto.

Opcionalmente, el programa debe identificar y codificar la instrucción en base a los registros que se estén usando, de manera similar a como lo vimos en clase, así mismo, debe cambiar la codificación si se esta usando una variable, y trabajar para los casos que vimos en clase, reg/reg para cuando trabajamos con 2 registros, o reg/mem cuando trabajamos con un registro y una variable.



1. En lugar de clasificar la instrucción como “etq”, debe mostrar “Correcto”



**No se si la profa revise a detalle la codificación, porque creo que no se hace como debería la codificación, es decir, no se codifica según los casos de reg/mem, reg/reg, mem/reg**