**Universidad de Guadalajara**

**Centro Universitario de Ciencias Exactas e Ingenierías**

**Taller de programación de sistemas**

**Reporte # 4**

**Romero Gastelu, María Elena**

**NRC: 02316**

**ALUMNO:**

GÓMEZ TOVAR EDGAR IVÁN

CODIGO: 303526879

27-10-2008

**Funciones y variables utilizadas**

* **Función:** directivas ()

Esta función se encarga de realizar las diversas validaciones a las directivas; cuando valida totalmente una de ellas llama a la función a\_temporal () para que inserte los datos correspondientes en el TEMPORAL.txt.

La función cuenta con las siguientes variables:

* **Enteras:** aux, correcto, i, signo.

Las variables aux, correcto y signo actúan como banderas. La variable i es un contador.

* **Función:** a\_temporal ()

Esta función es llamada por la función directivas () al validar una directiva (valga la redundancia) y se encarga de crear el TEMPORAL (si es necesario) y de introducir los datos pertinentes en el. También se encarga de llamar a la función a\_tabsim () para hacer lo propio con las etiquetas.

La función cuenta con las siguientes variables:

* **Enteras:** i, aux, aux2, posición.
* **Carácter:** nombre\_t [20].
* **Función:** a\_tabsim ()

Esta es la función encargada de introducir las etiquetas al TABSIM.txt previa verificación de que estas no se encuentran ya en el archivo. Es llamada dentro de la función a\_temporal cuando tal función ya realizo su trabajo.

* **Enteras:** i, aux, aux2, posicion.
* **Carácter:** temporal, etiqueta\_t [20].
* **Función:** dec\_hex ( )

La función se encarga de convertir un número entero que recibe, a su equivalente hexadecimal pero ahora en forma de cadena de caracteres. También se encarga de dejar tal cadena en un formato de 2 bytes, rellenando con ceros si es necesario.

* **Enteras:** i ,j, k, bandera, todavia.
* **Carácter:** aux\_num\_hex [20], num\_hex [20],

**Formato de los archivos auxiliares y validación de las directivas**

Básicamente cuando se detecta que un codop cumple con las reglas de escritura permitidas se manda a llamar a la función directivas (), esta función se encargara de verificar si ese codop es en realidad una directiva y si no lo es se manda a llamar a la función busqueda\_tabop () para que verifique si es un código de operación. La función directivas primero verifica que tipo de directiva es, si es un ORG, DW, DB, DC.W, DC.B, FCB, FDB, DS, DS.B, DS.W, RMB, RMW o EQU comienza a validar que tenga un operando numérico y que el valor numérico de este se encuentre dentro de los parámetros permitidos para cada directiva particular. Si es un ORG le suma un 1 a un contador que indica las incidencias de tal directiva (recordemos que solo puede haber un ORG por cada archivo a ensamblar) también verifica el valor de su operando y si es correcto se lo asigna a DIR\_INIC la cual a su vez se le asigna al CONTLOC, si en cambio es un EQU verifica que este tenga una etiqueta y un operando. Si en cambio la directiva es un FCC verifica que tenga un operando en forma de cadena de caracteres encerrados entre comillas dobles y si es un END verifica que esta no tenga operandos. Si el rango del operando esta correcto se manda a llamar a la función a\_temporal ().

Al llegar a la función a\_temporal () lo primero que se hace es copiar el nombre del archivo que se tiene ensamblando en el momento, se le extraen los caracteres hasta que se encuentra un dígito y se le concatena la cadena “tmp.txt”; se guarda la posición actual del cursor del archivo abierto, se cierra tal archivo y se trata de abrir el archivo TEMPORAL correspondiente, si este no se puede abrir o no existe se procede a crear uno nuevo con el mismo nombre. Al abrirse el archivo TEMPORAL se verifica si el codop actual es una directiva o un código de operación en base a una bandera, si es una directiva se escribe en el archivo el valor que regrese la función dec\_hex a la cual se le enviara el valor del CONTLOC (hay que recordar que tal función retorna en forma de cadena el valor hexadecimal de un entero), se escribe un tabulador, luego el valor de la etiqueta, luego otro tabulador, el valor del codop, un último tabulador, el valor del operando y por ultimo un salto de línea; en este punto (antes de incrementar el CONTLOC) se manda a llamar a la función a\_tabsim () la cual lo primeo que verifica es si existe una etiqueta, si esto no se cumple retorna a donde fue llamada pero en cambio si la etiqueta no es nula guarda la posición del cursor del archivo en uso, se cierra tal archivo y se verifica si el TABSIM existe o no intentando abrirlo, y si no es así crea uno nuevo luego, ya abierto el TABSIM, se hace una búsqueda intentando encontrar concordancia alguna con la etiqueta actual en caso de que esto ocurra la función solo retorna pero si la etiqueta no está en el TABSIM escribe el valor del CONTLOC y la etiqueta regresando el control a la función a\_temporal (). Regresando a la función, y en base a la directiva actual, se procede a incrementar el CONTLOC de la manera correspondiente; después se verifican la cantidad de ORG que han aparecido hasta el momento, si esta cantidad es 0 y la directiva actual no es un EQU se manda el error a pantalla indicando que antes de un ORG solo puede estar una directiva EQU, también se verifica el rango del CONTLOC y sus posibles errores.

Si por el contrario no se tuviera una directiva pero si un código de operación el procedimiento seria el mismo pero incrementando el valor del CONTLOC de acuerdo a lo que indique el TABOP.txt con el parámetro de total de bytes.