**Universidad de Guadalajara**

**Centro Universitario de Ciencias Exactas e Ingenierías**

**Taller de programación de sistemas**

**Reporte # 10**

Romero Gastelu, María Elena

**NRC**: 02316

**Alumno:**

Gómez Tovar Edgar Iván

**Código:**

303526879

8-12-2008

**Breve descripción de las funciones y variables utilizadas**

* **Función:** S19 (char registro[ ], char longitud[ ], char dir[ ], char datos[ ], char ch[ ])

Esta función se encarga de escribir los diversos registros en el archivo .S19; lo primero que hace es verificar lo que contiene la variable registro, si esta es “S0” crea el archivo .S19 sobrescribiendo cualquier otro archivo .S19 que se hubiera generado para el nombre del archivo especificado.

La función cuenta con las siguientes variables:

* **Enteras:** tipo, posición.
* **Carácter:** char registro [ ], char longitud [ ], char dir [ ], char datos [ ], char ch [ ].
* **Función:** checksum (char longitud [ ], char dir [ ], char datos [ ])

Esta función obtiene el campo de checksum en base a los parámetros que recibe; lo que hace es tomar pares de caracteres de cada variable, las transforma a su representación decimal (recordemos que todas ellas están en hexadecimal) y le suma esos valores a una variable llamada “suma”. Cuando se han tomado todos los pares de las variables se transforma lo que contiene suma en binario y se obtiene su complemento a la base, por último se toman los 8 bits menos significativos, se transforman en hexadecimal y esto se almacena en ch.

La función cuenta con las siguientes variables:

* **Enteras:** suma.
* **Carácter:** char longitud [ ], char dir [ ], char datos [ ], char ch [3].

Además de las variables usadas en las funciones, utilice:

* **Enteras:** línea\_S19, largo.
* **Carácter:** tamano [3], dirección [5], datos [32], cod\_maq2 [80].

**Descripción de la forma de escritura en el archivo .S19**

Cuando se encuentra una directiva ORG se comienza a obtener el ASCII en 2 bytes de cada carácter del nombre del archivo que se está ensamblando, a esto se le concatena el valor ASCII del carácter de control “LF” y se le pone su correspondiente fin de cadena; en este punto tenemos listo lo que se escribirá en el campo de datos del registro S0, ahora obtenemos la longitud de este registro de datos y la dividimos entre 2 (ya que necesitamos la cantidad de caracteres, y en esta cadena cada carácter está representado por un ASCII en 2 bytes), a esto le sumamos 3 (correspondientes al campo de checksum y direccion) para dejar listo el campo de longitud del registro S0.

Llamamos a la función checksum ( ) y le mandamos los valores de longitud y datos que previamente obtuvimos, además de la cadena “0000” como campo de dirección; esto dejara el valor del checksum en la variable ch.

Inmediatamente después mandamos llamar a la función S19 mandándole los mismos parámetros que le mandamos a la función checksum ( ), más la variable ch y la cadena “S0” como identificador de registro. Inicializamos la variable “largo” en cero, esta variable nos servirá para ir midiendo el campo de datos del registro S1.

Ahora que hemos escrito el registro S0 e inicializado la variable largo podemos comenzar a completar el registro S1. Cada vez que se genere código maquina en una línea se le asignara un 2 a la variable “línea\_S19” y ese código maquina se almacenara en la variable “cod\_maq2” además se le asignara el valor del CONTLOC en formato hexadecimal de 4 bytes a la variable direccion, en contraparte si se encuentra una directiva de reserva de espacio se le asignara un 1 a la variable línea\_S19, por ultimo si no se genera código maquina pero tampoco nos encontramos con una directiva de reserva de espacio se le asignara un 0 a la variable línea\_S19.

Si línea\_S19 es igual a 2, existen 3 posibilidades:

1. Si la longitud de cod\_maq2 más el valor de la variable largo es menor a 16: si largo es igual a cero copiamos lo que contenga cod\_maq2 en datos, de otra forma le concatenamos cad\_maq2 a datos; de cualquier forma le sumamos a largo la longitud de la variable cod\_maq2.
2. Si la longitud de cod\_maq2 más el valor de la variable largo es igual a 16: le concatenamos cod\_maq2 a datos, a largo le asignamos 19 y obtenemos su representación en hexadecimal completada a 2 bytes para guardarla en la variable “tamano”; le mandamos a la función checksum ( ) las variables tamano, datos y dirección. Ya podemos mandar a llamar a la función S19 ( ), después de lo cual a largo le asignamos un cero.
3. Si la longitud de cod\_maq2 más el valor de la variable largo es mayor a 16: guardamos el valor que obtenemos de sumar la longitud de cod\_maq2 con la variable largo y a esto restarle 16. Ahora a la longitud de la variable cod\_maq2 le restamos este ultimo valor que obtuvimos, esto nos dará la cantidad de elementos de cod\_maq2 que podemos almacenar en datos antes de llegar a los 16 permitidos. Mientras que largo sea menor a 16 concatenamos los caracteres de cod\_maq2 en datos, en su posición largo. Luego a largo le asignamos un 19 y obtenemos su representación hexadecimal en 2 bytes que almacenaremos en tamano, mandamos llamar a checksum ( ) y le mandamos tamano, dirección y datos y ahora llamamos a S19 mandandole tamano, dirección, datos y ch; le asignamos un cero a largo y repetimos el proceso de asignarle los caracteres que quepan en datos de cod\_maq2, y a la representación en decimal del valor de la dirección le sumamos aquel primer valor que calculamos y que se refería a la cantidad de elementos de cod\_maq2 que podíamos almacenar en datos antes de llegar a los 16 permitidos, esto lo sacamos en su representación hexadecimal en 4 bytes para dejarla como la nueva dirección. Ahora, si en este punto largo fuera igual a 16 (si hubiera que hacer otro registro S1 por que el cod\_maq2 fuera muy largo, por ejemplo con una cadena de la directiva FCC) haríamos lo mismo hasta que largo fuera distinto a 16.

Si línea\_S19 fuera igual a 1, a largo le sumariamos 3 y con los datos que tenemos mandamos llamar a checksum ( ) y posteriormente a S19 ( ), para luego poner en cero la variable largo.

Por último cuando se encuentre una directiva END, se procederá a escribir los datos que tengamos pendientes en un registro S1 para después mandar llamar a la función S19 ( ) con los siguientes datos constantes: tamano = “03”, dirección = “0000” y ch = “FC”.