

TALLER PROGRAMACION DE SISTEMAS P1

PRÁCTICA 1: IDENTIFICAR LAS LINEAS DE LENGUAJE ENSAMBLADOR

Módulo 2 Diseño del ensamblador.

Practica no 1. Identificación de las partes de una línea de lenguaje ensamblador. Implementar el algoritmo para separar las partes de una línea de lenguaje ensamblador. Las prácticas se realizaran de manera individual. Las practicas pueden desarrollarse en lenguaje C, Java, C #.

Descripción de la práctica:

1. Capturar el siguiente ejemplo con un editor de texto puro (por ejemplo, bloc de notas)

```
; comentario numero 1 =TALLER=
    ORG    %00001111
Et1   equ    $ffff
dos   LDAA   @4732
      SWI
      DS.b   %0011000011111100
;comentario numero 2 programación!
;comentario numero 3
tres  sWi
      End
```

2. Al capturar el ejemplo utilizar espacios en blanco y tabuladores de manera alternada para crear varias opciones de separación de palabras.
3. Grabar el archivo con el siguiente nombre P1ASM.TXT
4. Cada una de las líneas capturadas es un ejemplo ficticio de un código en lenguaje ensamblador para la arquitectura HC12. Cada una de estas líneas puede ser de dos formas:
 - a. Línea de comentario
 - b. Línea de ETIQUETA, CODIGO DE OPERACIÓN (CODOP) y OPERANDO
5. Las líneas de comentario tienen las siguientes reglas de escritura.
 - a. Comienza con el carácter de “;”.
 - b. Este carácter de “;” solo puede estar en la primera posición de la línea.
 - c. Después del carácter de “;” pueden seguirle cualquier carácter. En caso de letras pueden ser indistintamente mayúsculas y minúsculas.
 - d. La longitud máxima es de 80 caracteres.

Taller de Programación de Sistemas

Práctica 1: Identificar las líneas del lenguaje ensamblador.

- e. El delimitador de la línea de comentario es el retorno de carro ("enter").
- 6. Las líneas que están formadas por ETIQUETA, CODOP, y OPERANDO tienen las siguientes reglas de escritura;
 - a. ETIQUETAS;
 - i. Comienza con letra mayúsculas o minúsculas (son validos los dos casos)-
 - ii. Esta primera letra debe de estar solo en la primera posición de la línea.
 - iii. Su longitud es de 8 caracteres máximo.
 - iv. Después de la primera letra le pueden seguir mas letras, números (0...9) o guiones bajos "_".
 - v. Cualquier otro carácter representado es un error.
 - b. CODOPS (códigos de operación)
 - i. Comienzan con letra mayúsculas o minúsculas (son validos los dos casos)
 - ii. El otro carácter valido es el punto "." Y solo puede haber uno solo en la palabra.
 - iii. Su longitud máxima es de 5 caracteres.
 - iv. Cualquier otro carácter es un error.
 - c. OPERANDOS:
 - i. Pueden comenzar con cualquier carácter (en las siguientes practicas se revisaran las excepciones).
 - ii. Pueden tener cualquier longitud.
- 7. Este tipo de líneas pueden tener las siguientes combinaciones:
 - a. ETIQUETA, CODOP, OPERANDO
 - b. ETIQUETA, CODOP
 - c. CODOP, OPERANDO
 - d. CODOP
- 8. Para el primer caso: Etiqueta, CODOP, OPERANDO, debemos de considerar que:
 - a. Si la línea comienza con ETIQUETA, el primer carácter de la línea debe de ser entonces una letra (mayúscula o minúscula).
 - b. Entre ETIQUETA y CODOP solo podemos tener separadas estas dos palabras por espacios en blanco o tabuladores (considerar los dos casos)
 - c. Entre CODOP y OPERANDO solo podemos tener separadas dos palabras en blanco o tabuladores (considerar los dos casos)
 - d. Después de la palabra OPERANDO solo podemos tener el retorno de carro o "enter" este siempre será el delimitador de cada sentencia de lenguaje ensamblador.
- 9. Para el segundo caso: ETIQUETA, CODOP, debemos considerar que:
 - a. Si la línea comienza con ETIQUETA, el primer carácter de la línea debe de ser entonces una letra (mayúscula o minúscula).

Taller de Programación de Sistemas

Práctica 1: Identificar las líneas del lenguaje ensamblador.

- b. Entre Etiqueta y CODOP solo podemos tener separadas estas dos palabras por espacios en blanco o tabuladores (considerar los dos casos).
 - c. Después de CODOP solo podemos tener el retorno de carro o “enter” este siempre será el delimitador de cada sentencia de lenguaje ensamblador.
- 10. Para el tercer caso: CODOP, OPERANDO, debemos de considerar que:
 - a. Antes de la palabra CODOP solo podemos tener espacios en blanco o tabuladores (considerar los dos casos). Es decir, el primer carácter de la línea será o un espacio en blanco o un tabulador.
 - b. Entre las palabras CODOP y OPERANDO solo podemos tener espacios en blanco o tabuladores, considerar los dos casos.
 - c. Después de la palabra OPERANDO solo podemos tener el retorno de carro “enter” este siempre será considerado como el delimitador de cada sentencia del lenguaje ensamblador.
- 11. Para el cuarto caso: CODOP debemos de tener las siguientes consideraciones:
 - a. Antes de la palabra CODOP solo podemos tener espacios en blanco y tabuladores considerar los dos casos. Es decir, el primer carácter de la línea será o un espacio en blanco o un tabulador.
 - b. Después de la palabra CODOP solo podemos tener el retorno de carro o “enter” este debe de ser considerado siempre como el delimitador de una línea de lenguaje ensamblador.
- 12. Se debe de observar que siempre debe de haber por lo menos la palabra que conforme el CODOP ya que de otra forma sería un error.
- 13. La última línea del ejemplo siempre debe tener la palabra END (ya sea escrita con mayúsculas, minúsculas o una combinación de mayúsculas y minúsculas). Es decir, al validar la línea que tiene la palabra END debemos de considerar que este es el final del archivo con el ejemplo de lenguaje ensamblador.

Para hacer el programa:

- 1. Hacer un programa que abra el archivo creado previamente (P1ASM.TXT)
- 2. Leer línea por línea el archivo de ejemplo, pudiendo:
 - a. Leer cada línea directamente del archivo, de una por una y procesar cada línea que se va leyendo; o bien,
 - b. Leer todo el archivo con un ciclo y conforme se lee el archivo almacenarlo en una estructura de datos temporal, posteriormente leer de esta estructura de datos.
 - c. En ambos casos existen ventajas y desventajas. Deberás de elegir la que sea más conveniente, recuerda que el programa debe de irse complementando con las siguientes prácticas.

Taller de Programación de Sistemas

Práctica 1: Identificar las líneas del lenguaje ensamblador.

3. Una vez que identificamos la primera línea se debe de comenzar a leer carácter por carácter para identificar las palabras e ignorar los espacios en blanco y tabuladores.
4. Una vez que se identifique la primera palabra de una línea se debe de determinar si esta fue ETIQUETA, CODOP u OPERANDO.
5. Se debe de utilizar tres variables que se deben de llamar, precisamente, como “etiqueta”, “codop” y “operando” que deben de ser de tipo cadena ya que estas leyendo de un archivo de texto. Una vez que se identifiquen plenamente las palabras de una línea debes de imprimir en pantalla el valor de cada variable. No olvides que antes de imprimir debes poner el terminador de cadena a la variable para que no almacene basura.
6. Si alguna variable no tuviera ningún valor debes de asignarle un valor “NULL” (cadena de texto con el mensaje NULL).
7. Una vez impreso el valor de las variables en el monitor, debes de inicializar las variables con el carácter de nulo, para que no se almacene basura en las variables. Una vez impreso el valor de las palabras identificadas por una línea debes de leer la siguiente línea del archivo y hacer el mismo procedimiento.
8. No olvides que para el caso de las líneas de comentario solo debes de imprimir en pantalla el mensaje “comentario”.

Ejemplo de salida al monitor para el archivo P1ASM.TXT propuesto:

COMENTARIO

ETIQUETA=null
CODOP=ORG
OPERANDO=%00001111

ETIQUETA=Et1
CODOP=equ
OPERANDO=\$ffff

ETIQUETA=dos
CODOP=LDAA
OPERANDO=@4732

ETIQUETA=null
CODOP=SWI
OPERANDO=null

ETIQUETA=null
CODOP=DS.b
OPERANDO=%0011000011111100

Taller de Programación de Sistemas

Práctica 1: Identificar las líneas del lenguaje ensamblador.

COMENTARIO

COMENTARIO

ETIQUETA=tres

CODOP=sWi

OPERANDO=null

ETIQUETA=null

CODOP=End

OPERANDO=null

Características de Entrega.

1. Entregar código fuente y ejecutable
2. Entregar reporte
 - a. No faltas de ortografía
 - b. Debe tener nombre del alumno, código, grupo, fecha de entrega
3. Descripción del reporte
 - a. Describir el algoritmo utilizado para leer el archivo. Describir el algoritmo para identificar las palabras.