



# PRACTICA 08

## MAPA DE MEMORIA

Rogelio Daniel Gonzalez Nieto

FACULTAD DE INGENIERIA AREA DE CIENCIAS DE LA COMPUTACION



# OBJETIVO

Crear el mapa de memoria en base al código objeto obtenido en la practica no. 6

# DESARROLLO

Para el desarrollo de esta práctica primero se separa el código objeto del programa en grupos de 2 caracteres que representan 1 bit de memoria con la siguiente función:

```
private List<string> separarCodigo(string linea)
{
    string n = "";
    string separado = linea.Substring(9);
    List<string> lista = new List<string>();

    foreach (var s in separado)
    {
        n += s;
        if (n.Length == 2)
        {
            lista.Add(n);
            n = "";
        }
    }

    return lista;
}
```

Se uso un diccionario como estructura de datos para el manejo de memoria, como key del diccionario se usó la dirección de inicio de escritura y como value se uso una lista de string con el código objeto ya separado, en base a la dirección en el key y a la dirección de inicio del código objeto se determina donde empezar a guardar los datos en el diccionario por medio del siguiente código:

```
private void llenarMapa(string direccion, List<string> codigo)
{
    bool bandera = false;
    int iCount = 0;
    foreach (var e in dMapaMemoria)
    {
        for (int j = 0; j < e.Value.Count; j++)
        {
            if (e.Value[j].Contains(direccion))
            {
                bandera = true;
            }

            if (bandera == true)
            {
                e.Value[j] = codigo[iCount];
                iCount++;
            }

            if (iCount >= codigo.Count)
            {

```

```

        bandera = false;
    }
}
}

```

Después de crearse el diccionario este se muestra en un datagrid con el siguiente código:

```

private void mapaMemoria()
{
    this.dataGridView4.Rows.Clear();
    int inicioProgrma = Convert.ToInt32(this.sTamaInicial, 16);
    int finalProgrma = Convert.ToInt32(this.sTamaFinal, 16);
    int inicio = inicioProgrma;
    int final = this.RoundUp(finalProgrma);
    string sDirIni = $"{inicio:X}";
    string sDirFin = $"{final:X}";
    string direccion = "";
    string dirInicio = "";
    List<string> lAux = new List<string>();

    for (int i = inicio; i < final; i += 16)
    {
        direccion = $"{i:X}";
        lAux = new List<string>();
        lAux.Add(direccion);
        lAux.Add("X" + direccion.Substring(0, 3) + "1X");
        lAux.Add("X" + direccion.Substring(0, 3) + "2X");
        lAux.Add("X" + direccion.Substring(0, 3) + "3X");
        lAux.Add("X" + direccion.Substring(0, 3) + "4X");
        lAux.Add("X" + direccion.Substring(0, 3) + "5X");
        lAux.Add("X" + direccion.Substring(0, 3) + "6X");
        lAux.Add("X" + direccion.Substring(0, 3) + "7X");
        lAux.Add("X" + direccion.Substring(0, 3) + "8X");
        lAux.Add("X" + direccion.Substring(0, 3) + "9X");
        lAux.Add("X" + direccion.Substring(0, 3) + "AX");
        lAux.Add("X" + direccion.Substring(0, 3) + "BX");
        lAux.Add("X" + direccion.Substring(0, 3) + "CX");
        lAux.Add("X" + direccion.Substring(0, 3) + "DX");
        lAux.Add("X" + direccion.Substring(0, 3) + "EX");
        lAux.Add("X" + direccion.Substring(0, 3) + "FX");
        this.dMapaMemoria.Add(direccion, lAux);
    }

    foreach (var l in lRegistros)
    {
        if (l != lRegistros[0] && l != lRegistros[lRegistros.Count - 1] && l.Length > 10)
        {
            List<string> objs = this.separarCodigo(l);
            string sDireccion = l.Substring(1, 6);
            int iDireccion = Convert.ToInt32(sDireccion, 16);
            sDireccion = $"{iDireccion:X}";
            this.llenarMapa(sDireccion, objs);
        }
    }

    int j = 0;

```

```

int k = 0;
List<string> sHeader = new List<string>();
for (int i = inicio; i < final; i += 16)
{
    direccion = $"{i:X}";
    sHeader.Add(direccion);
}

foreach (var d in this.dMapaMemoria)
{
    this.dataGridView4.Rows.Add(sHeader[j], d.Value[0], d.Value[1], d.Value[2], d.Value[3],
d.Value[4], d.Value[5], d.Value[6], d.Value[7], d.Value[7], d.Value[8], d.Value[9], d.Value[10],
d.Value[11], d.Value[12], d.Value[13], d.Value[14], d.Value[15]);
}
}

```

Ensamblador de SIC

Programa

```

SUM  START  4000H
FIRST LDA    ZERO
LOOP  ADD    TABLE,X
      TIX    COUNT
      JLT    LOOP
      STA    TOTAL
TABLE RESW   2000
COUNT RESW  1
ZERO   WORD  0
TOTAL  RESW  1
      END    FIRST

```

Direcciones

CP	Simbolo	Instruccion / Directiva	Direccion	Codigo OBJ	Direccionamiento
4000H	SUM	START	4000H	---	---
4003	FIRST	LDX	ZERO	04578B	DIRECTO
4006		LDA	ZERO	00578B	DIRECTO
4009	LOOP	ADD	TABLE,X	185785	INDEXADO
400C		TIX	COUNT	2C5788	DIRECTO
400F		JLT	LOOP	384009	DIRECTO
4012		STA	TOTAL	0C578E	DIRECTO
4015		RSUB		4C0000	DIRECTO
5785	TABLE	RESW	2000	---	---
5788	COUNT	RESW	1	---	---
578B	ZERO	WORD	0	000000	DIRECTO
578E	TOTAL	RESW	1	---	---
5791		END	FIRST	---	---

Tabla de Simbolos

Simbolo	Direccion
SUM	4000H
FIRST	4003
LOOP	4009
TABLE	5785
COUNT	5788
ZERO	578B
TOTAL	578E

Errores

Errores de direcciones:

El Programa No Contiene Errores de Etiquetas repetidas  
Errores de instrucciones y/o directivas  
Sin errores de instrucciones y/o directivas

Mapa de Memoria

C.P.	0	1	2	3	4	5	6	7	8	9	A	B
4000	04	57	8B	00	57	8B	18	57	57	85	2C	57
4000	57	8E	4C	00	00	X4015X	X4016X	X4017X	X4017X	X4018X	X4019X	X4
4000	4020	X4021X	X4022X	X4023X	X4024X	X4025X	X4026X	X4027X	X4027X	X4028X	X4029X	X4
4000	4030	X4031X	X4032X	X4033X	X4034X	X4035X	X4036X	X4037X	X4037X	X4038X	X4039X	X4
4000	4040	X4041X	X4042X	X4043X	X4044X	X4045X	X4046X	X4047X	X4047X	X4048X	X4049X	X4
4000	4050	X4051X	X4052X	X4053X	X4054X	X4055X	X4056X	X4057X	X4057X	X4058X	X4059X	X4
4000	4060	X4061X	X4062X	X4063X	X4064X	X4065X	X4066X	X4067X	X4067X	X4068X	X4069X	X4
4000	4070	X4071X	X4072X	X4073X	X4074X	X4075X	X4076X	X4077X	X4077X	X4078X	X4079X	X4
4000	4080	X4081X	X4082X	X4083X	X4084X	X4085X	X4086X	X4087X	X4087X	X4088X	X4089X	X4
4000	4090	X4091X	X4092X	X4093X	X4094X	X4095X	X4096X	X4097X	X4097X	X4098X	X4099X	X4
4000	40A0	X40A1X	X40A2X	X40A3X	X40A4X	X40A5X	X40A6X	X40A7X	X40A7X	X40A8X	X40A9X	X4

Programa Objeto

```

HSUM  0040001791
T0040001504578B00578B1857852C57883840090C578E4C000
0
T00578515
T005788178B000000
E4003

```

Conjunto de Instrucciones

Nemotico	Codigo de Operacion
ADD	18
AND	40
COMP	28
DIV	24
J	3C
JEQ	30

Procesar Programa

Las celdas con valores con X son solo celdas de ayuda para la posición de memoria, las demás son el código objeto anteriormente obtenido