### Esqueleto básico

TITLE Program Template (OpArrArg.asm)  
  
; Irvine Library procedures and functions  
INCLUDE \masm32\Irvine\Irvine32.inc  
INCLUDELIB \masm32\Irvine\Irvine32.lib  
INCLUDELIB \masm32\Irvine\User32.lib  
INCLUDELIB \masm32\Irvine\Kernel32.lib  
; End Irvine  
  
;SIMBOLOS  
mcr=0dh  
mlf=0ah  
mnul=0h  
  
.DATA  
; PROC main  
  
  
; PROC sycArrdw, variables locales  
  
  
.CODE  
main PROC  
  
 EXIT  
main ENDP  
  
sycArrdw PROC  
  
 RET   
sycArrdw ENDP  
  
END main

### Instrucciones Irvine

|  |  |  |  |
| --- | --- | --- | --- |
| Instrucción | Descripción | Params | **Return** |
| *DumpRegs* | Despliega registros y banderas | - | - |
| *DumpMem* | Despliega rango de memoria | **ESI:** Dirección inicial<br />**ECX:** # elementos<br />**EBX:** tamaño de elementos (1B, 2B) | - |
| *WriteInt* | Escribe entero signado de 32 bits | **EAX:** Número a escribir | - |
| *WriteHex* | Escribe entero no signado en hexa | **EAX:** Número | - |
| *ReadHex* | Lee entero sin checar validez | - | **EAX:** Número leído |
| *WriteString* | Escribe cadena, terminada en 0 | **EDX:** Offset de la cadena | - |
| *ReadString* | Lee cadena hasta el sig. enter | **EDX:** Offset donde se va a guardar<br />**ECX:** max. # de caracteres (+1)<br /> | **EAX:** # de caracteres |
| *Gotoxy* | Coloca el cursor en la posición x,y | **DH:** x (renglón)<br />**DL:** y (columna) | - |
| *WriteChar* | Escribe el caracter | **AL:** caracter a escribir | - |

### Notas

* **LENGTHOF** = # de elementos
* **SIZEOF** = # bytes

#### Tipos de Direccionamiento Indirecto

* Operando Indirecto: Es mejor para procedimientos!
* .DATA  
  array SDWORD 2,3,5,6  
    
  .CODE  
  MOV ESI, OFFSET array  
    
  MOV EAX, [ESI]  
  ADD ESI, TYPE array
* Operando Indexado
  + Normal
  + .DATA  
    array SDWORD 2,3,5,6  
      
    .CODE  
    MOV ESI, 0 ; indice = 0, 4, 8, ...  
      
    MOV EAX array[ESI] ; opcion 1  
    ADD ESI, TYPE array  
      
    MOV EAX [array + ESI] ; opcion 2
  + Escalado
  + .DATA  
    array SDWORD 2,3,5,6  
      
    .CODE  
      
      
    MOV ESI, OFFSET array ; dirección del arreglo  
    MOV EBX, 0 ; indice = 0, 1, 2, 3, ...  
      
    MOV EAX [ESI+EBX\*TYPE array]  
      
    INC EBX ; aumento indice
  + Desplazamiento
  + .DATA  
    array SDWORD 2,3,5,6  
      
    .CODE  
      
    MOV ESI, OFFSET array ; dirección del arreglo  
    MOV EBX, 0 ; indice = 0, 4, 8, ...  
      
    MOV EAX [ESI+EBX]  
      
    ADD EBX, TYPE array