

Procesadores de lenguajes

Practica 2

Luis José Quintana Bolaño

Junio, 2014

Abstract

Soluciones a las cuestiones teóricas de la segunda practica sobre ensamblador.

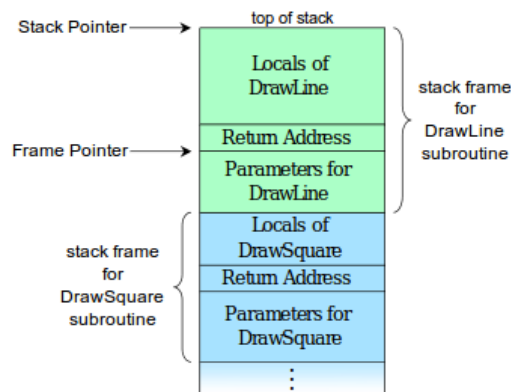
1 Registros de proposito general en Intel 32

General-Purpose Registers						
31	16	15	8	7	0	
			AH		AL	AX 16-bit EAX 32-bit
			BH		BL	BX 16-bit EBX 32-bit
			CH		CL	CX 16-bit ECX 32-bit
			DH		DL	DX 16-bit EDX 32-bit
			BP			EBP 32-bit
			SI			ESI 32-bit
			DI			EDI 32-bit
			SP			ESP 32-bit

Figure 1: Esquema de los registros de proposito general de la arquitectura Intel-32.

2 Los registros EBP y ESP

El registro **EBP** (Frame pointer) es un registro de 32 bits que mantiene la dirección de la pila de llamadas en uso. El registro **ESP** (Stack Pointer), también de 32 bits, contiene la dirección del tope de la pila.



3 Métodos de direccionamiento

Métodos de direccionamiento y ejemplos:

- Por registro
 - mov ecx, ebx
- Inmediato
 - mov eax, 26
- Directo
 - mov eax, var_name
- Indirecto
 - mov ecx, [ebx]
- Desplazamiento
 - mov eax, [esp+4]
- Indexado
 - mov eax, [esp][esi]

4 Estudio del ensamblador de suma

Procedemos a analizar el código ensamblador del programa propuesto, marcando en rojo las pseudoinstrucciones y en azul las directivas de compilación:

```

.file      "suma.c"
.comm     a,4,4
.comm     b,4,4
.comm     c,4,4
.section   .rodata
.LC0:
.string    "%d + %d = %d \n"
.text
.globl     main
.type      main, @function
main:
.LFB0:
.cfi_startproc
pushl      %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
movl       %esp, %ebp
.cfi_def_cfa_register 5
andl       $-16, %esp
subl       $16, %esp
movl       $5, a
movl       $7, b
movl       a, %edx
movl       b, %eax
addl       %edx, %eax

```

```
movl    %eax, c
movl    c, %ecx
movl    b, %edx
movl    a, %eax
movl    %ecx, 12(%esp)
movl    %edx, 8(%esp)
movl    %eax, 4(%esp)
movl    $.LC0, (%esp)
call    printf
movl    $0, %eax
leave
.cfi_restore 5
.cfi_def_cfa 4, 4
ret
.cfi_endproc
.LFE0:
.size    main, .-main
.ident   "GCC: (Ubuntu 4.8.2-19ubuntu1) 4.8.2"
.section .note.GNU-stack,"",@progbits
```