Arquitectura de computadores
EJERCICIO EN CLASE

Nombre Ricardo Medina	Viernes 26 de abril 2024
Nombre Alejandro Osejo	
Código: A00369009 - A00372469	Duración: 1.0 hora

En la figura 1 se muestra un programa en lenguaje alto nivel donde desde el main, se llama una función ***cal(n)*** que realiza una operación recursiva sobre un entero, lamentablemente el programa tiene una línea de código incompleta, la línea 6 [*return ????????;*], y por lo tanto no podemos saber qué valor retorna.

```
1  int Result;
2  int cal(int n)
3  {
4      if (n <= 1)
5          return n;
6      return ????????; //linea incompleta
7  }
8
9  int main()
10 {
11     int tmp;
12     int n = 7;
13     tmp=cal(n);
14     Result=tmp;
15     return 0;
16 }
17
```

Figura 1

Recuadro 1
Los valores de los registros al inicio de la ejecución:

EAX = 0105FC9C
EBX = 00CCD000
ECX = 00F5100A
EDX = 00F5100A
ESI = 00F5100A
EDI = 00F5100A
EIP = 00F5101C
ESP = 0105FC44
EBP = 0105FC50

```
00F5101C main:push      ebp
00F5101D      mov       ebp,esp
00F5101F      call      L0
00F51024  L0: pop       ebx
00F51025      sub       esp,8
00F51028      mov       dword ptr [ebp-4],7
00F5102F      mov       eax,dword ptr [ebp-4]
00F51032      push      eax
00F51033      call      cal (0F5104Ah)
00F51038      add       esp,4
00F5103B      mov       dword ptr [ebp-8],eax
00F5103E      xor       eax,eax
00F51040      mov       esp,ebp
00F51042      pop       ebp
00F51043      ret

00F5104A cal: push      ebp
00F5104B      mov       ebp,esp
00F5104D      push      esi
00F5104E      cmp       dword ptr [ebp+8],1
00F51052      jg        LN2 (0F51059h)
00F51054      mov       eax,dword ptr [ebp+8]
00F51057      jmp       LN1 (0F5107Bh)
00F51059 LN2: mov       eax,dword ptr [ebp+8]
00F5105C      sub       eax,1
00F5105F      push      eax
00F51060      call      cal (0F5104Ah)
00F51065      add       esp,4
00F51068      mov       esi,eax
00F5106A      mov       ecx,dword ptr [ebp+8]
00F5106D      sub       ecx,2
00F51070      push      ecx
00F51071      call      cal (0F5104Ah)
00F51076      add       esp,4
00F51079      add       eax,esi
00F5107B LN1: pop       esi
00F5107C      pop       ebp
00F5107D      ret
```

Afortunadamente se contaba con el archivo ejecutable y por medio del proceso de desemsambaldo se recuperó el archivo ensamblador correspondiente al programa alto nivel.

Powered by CamScanner

A. Sobre el código ensamblador suministrado, encierre e indique las instrucciones ensamblador correspondientes a las siguientes líneas de código de alto nivel: [10%]

```
int n = 7;
tmp=cal(n);
if (n <= 1)
```

Se dará cuenta que en el programa ensamblador MASM falta las instrucciones correspondientes a la línea de código 14: `Result=tmp;`
Escriba entonces la(s) instrucción(es) faltante(s):

Mov dword ptr (result), eax

B. En la tabla adjunta, realice un análisis dónde muestre el contenido de la pila cuando se finaliza la ejecución del programa, para ello asuma los valores iniciales de los registros que se les suministra en el recuadro 1. Una vez se ejecute el programa, indique el valor que se almacena en la variable **tmp**, indique en la tabla dónde está almacenada dicha variable (tmp) y especifique el contenido final de los registros justo antes del ret del main[55%]

EAX = 00000000 EBX = _____ ECX = 00000006
EDX = 00F5100A ESI = 00000001 EDI = 00F5100A
ESP = 0105FC14 EBP = 0105FBE4 EIP = _____

C. Ahora, después del análisis del comportamiento de la pila y el seguimiento al programa ensamblador y el resultado que devuelve la función cal(n), complete la línea de alto nivel faltante: [15%]

```
int cal(int n)
{
 if (n <= 1)
   return n;
return cal(n-1) + cal(n-2);   ← complete esta línea
}
```

----------Fin----------

Anexo

DECIMAL	HEX	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Powered by CamScanner

STACK					Instrucciones que escriben	Instrucciones que leen
0x0105FBAC						
0x0105FBB0						
0x0105FBB4						
0x0105FBB8						
0x0105FBBC						
0x0105FBC0						
0x0105FBC4						
0x0105FBC8						
0x0105FBCC						
0x0105FBD0						
0x0105FBD4						
0x0105FBD8						
0x0105FBDC						
0x0105FBE0	6A	10	F5	00	Push ESI	Pop ESI
0x0105FBE4	E4	FB	05	01	Push EBP	Pop EBP
0x0105FBE8	65	10	F5	00	Call cal (0F5104Ah)	
0x0105FBEC	01	00	00	00	Push EAX	
0x0105FBF0	0A	10	F5	00	Push ESI	
0x0105FBF4	04	FC	05	01	Push EBP	
0x0105FBF8	65	10	F5	00	Call Cal (0F5104Ah)	
0x0105FBFC	02	00	00	00	Push EAX	
0x0105FC00	0A	10	F5	00	Push ESI	
0x0105FC04	14	FC	05	01	Push EBP	
0x0105FC08	65	10	F5	00	Call Cal (0F5104Ah)	
0x0105FC0C	03	00	00	00	Push EAX	
0x0105FC10	0A	10	F5	00	Push ESI	
0x0105FC14	24	FC	05	01	Push EBP	
0x0105FC18	65	10	F5	00	Call Cal (0F5104Ah)	
0x0105FC1C	04	00	00	00	Push EAX	
0x0105FC20	0A	10	F5	00	Push ESI	
0x0105FC24	34	FC	05	04	Push EBP	-
0x0105FC28	65	10	F5	00	Call Cal (0F5104Ah)	
0x0105FC2C	05	00	00	00	Push EAX	
0x0105FC30	0A	10	F5	00	Push ESI	
0x0105FC34	44	FC	05	01	Push EBP	-
0x0105FC38	65	10	F5	00	Call Cal (0F5104Ah)	
0x0105FC3C	06	00	00	00	Push EAX	
0x0105FC40	0A	10	F5	00	Push ESI	
0x0105FC44	54	FC	05	01	Push EBP	-
0x0105FC48	38	10	F5	00	Call Cal (0F5104Ah)	
0x0105FC4C	07	00	00	00	Push EAX	
0x0105FC50	07	00	00	00	Call L0	Pop EBX
0x0105FC54	50	FC	05	01	Push EBP	

Powered by CamScanner