1er Control Estructura de Computadores II

curso 2006-2007 Q2

Nombre: . . . . . .

Problema 1 (1 punto)

Traducid a ensamblador del IA32 la funciones escritas en C:

```
char Lower(char c) {
  if ((c >= 'A') && (c <= 'Z'))
    return c - 'A' + 'a';
  else
    return Low(c);   /* la definición de Low es "char Low(char c);" */
}
```

Problema 2 (1'5 puntos)

Disponemos de una matriz de enteros de 100 x 100 elementos, almacenada por filas a partir de la dirección simbólica M. **Escribid** un programa, lo más óptimo posible, en ensamblador de IA32, que copie los elementos de la diagonal de la matriz M a un vector de 100 elementos que se encuentra en memoria a partir de la dirección simbólica V (Nota: la instrucción "movl $M, %ebx" carga la dirección de inicio de M en ebx).

Problema 3 (3,5 puntos)

Dado la siguiente porción de código escrito en C:

```
typedef struct {
   char vc[6];
   int i1;
} S1;
typedef struct {
  char c3;
  S1 vs[10];
  short int s4;
  char *pc2;
} S2;

int XXX(int *pj, int i);
int ec2(int M[100][100], int j, S1 ss, int *pi) {
  S2 dos;
  int v[100];
  int tmp;
  ...
}
```

Se pide:

a) **Dibujad** cómo quedaría la estructura de datos S2 en Linux, indicando claramente los desplazamientos con respecto a la dirección de inicio de la estructura, el tamaño de cada campo y el tamaño total de la estructura.

b) **Dibujad** el bloque de activación de la subrutina, indicando claramente los desplazamientos a partir del registro ebp, así como el tamaño de cada parámetro y variable local.

c) **Traducid** a ensamblador del IA32 la siguiente sentencia en C, suponiendo que está dentro de la función ec2:

```
M[*pi][j] = XXX(&tmp, j);
```

d) **Traducid de forma óptima** a ensamblador del IA32 la siguiente sentencia en C, suponiendo que está dentro de la función ec2:

```
for (tmp=0; tmp<100; tmp++)
   M[0][tmp] = v[tmp] | dos.vs[j].i1;
```

Problema 4 (2 puntos)

1) Dado el siguiente formato de punto flotante:

S	Exponente			Mantisa		
12	11		8	7		0

Donde: S es el bit de signo, el exponente se representa en exceso 8, hay bit implicito y la mantisa está normalizada.

Representad en este formato el número 5,3

2) **Escribid** una secuencia de microoperaciones que se comporte de la misma forma que la siguiente instrucción IA32: **popl -35(%ebx, %eax,4)**. Las microoperaciones que podéis utilizar son las siguientes:

```
Ri = Rj op Rk                    Ri = GetDesplazamiento(IR)
Ri = M[Rj]                        Ri = GetFactorEscala(IR)
M[Rj] = Ri                       Ri = GetDesplazamiento(IR)
```

3) **Escribid** una secuencia de instrucciones para ejecutar la operación a = (b-c) /(a+d) en una arquitectura de tipo pila en la que las operaciones se describen de la sigiente forma:

```
add/sub/mul/div   #pila[sp+1]=pila[sp] op pila[sp+1]; sp=sp+1
push @            #sp=sp-1; pila[sp]=M[@]
pop @             #M[@]=pila[sp]; sp=sp+1
```

4) **Escribid** la secuencia de comandos Linux necesaria para extraer los ficheros contenidos en "Programas.Sesion05.tar.gz".

Problema 5 (2 puntos)

Responded a las siguientes afirmaciones poniendo una X en el recuadro correspondiente (en la columna C si la afirmación es cierta o en la columna F si la afirmación es falsa). Cada respuesta contestada correctamente SUMA 0,2 puntos. **Cada respuesta incorrecta RESTA 0,2 puntos**. Las respuestas no contestadas no se tienen en cuenta.

C	F	Afirmación
		El rango de un short int es -2^7 .. 2^7 -1
		La instrucción movl %eax, $6(%ebx,%ecx,2) produce un error de compilación
		El registro %bh se puede modificar en el interior de una subrutina sin necesidad de salvarlo previamente
		Los bits de condición no se activan con la instrucción leal
		La instrucción pushl suma 4 al valor del registro %esp
		Un short int ocupa 2 bytes cuando se almacena en la pila como parámetro de una subrutina
		Los 16 bits de menor peso del registro %esi NO pueden ser accedidos como dos registros de 8 bits, el %sh y el %sl
		En C las matrices se almacenan por filas en posiciones consecutivas de memoria
		Los operandos inmediatos en IA32 se almacenan temporalmente en el registro %eax
		En C todos los tipos de datos estructurados, incluidos los struct, se pasan por referencia cuando son parámetros de una subrutina