



Apellidos: _____ Nombre: _____ DNI: _____

Examen de Fundamentos de Computadores – 5 de septiembre de 2009

Cada respuesta correcta suma un punto. Cada respuesta incorrecta, ilegible o vacía no suma ni resta. Para obtener la nota del examen, el total de puntos se dividirá entre el total de preguntas y se multiplicará por 10.

Se ha escrito un programa en ensamblador de la arquitectura x86-32 que calcula cuál es el máximo de la longitud de varias cadenas, considerando que están codificadas en UTF-8. Para ello utiliza el procedimiento LongitudUTF8 que recibe por la pila la dirección de una cadena y devuelve en eax su número de caracteres considerando que está codificada en UTF-8. El código del programa se muestra a continuación:

```
.386
.model flat, stdcall
ExitProcess proto, ExitCode:dword

.data
Cadena1 DB 69h, 0C3h, 0B3h, 6Eh, 0
Cadena2 DB 0C3h, 0A1h, 0C3h, 0A9h, 0C3h,
        0ADh, 0
Cadena3 DB 53h, 54h, 069h, 070h, 6Eh, 0
ArrayCad ?? 0, 0, 0
Maximo DD 0

.code

; Procedimiento que devuelve en eax el nº
; de caracteres de una cadena UTF-8
LongitudUTF8 proc
    push ebp
    mov ebp, esp
    push edi
    push ecx
    mov edi, [ebp+8] ; Mover a edi el 1er parámetro

    xor eax, eax ; Inicializar longitudes a cero

bucleContador:
    mov cl, [edi]
    cmp cl, 0
    je short fuera
    and cl, ??
    cmp cl, 80h
    -- HUECO 1 --
```

```
    inc eax
car_no_nuevo:
    inc edi
    jmp short bucleContador

fuera:
    pop ecx
    pop edi
    pop ebp
    ret 4
LongitudUTF8 endp

; Programa principal
Inicio:
    mov [ArrayCad], OFFSET Cadena1
    mov [ArrayCad+4], OFFSET Cadena2
    mov [ArrayCad+8], OFFSET Cadena3

    mov edx, 0 ; edx: máx. temporal
    mov esi, 0
    mov ecx, 3
bucle:
    -- HUECO 2 --
    call LongitudUTF8
    cmp edx, eax
    jae no_hay_nuevo_max
    mov edx, eax
no_hay_nuevo_max:
    inc esi
    loop bucle

    mov [Maximo], edx

    push 0
    call ExitProcess
end Inicio
```

Listado1.asm - Programa Principal

— ¿Qué debería haber en lugar de las interrogaciones en la definición de la variable ArrayCad?

DD

— ¿Qué falta en las interrogaciones de la instrucción and cl, ?? del procedimiento LongitudUTF8?

0C0h

— ¿Qué instrucción o instrucciones faltan en el Huevo 1 del procedimiento LongitudUTF8?

jz car_no_nuevo

— ¿Qué instrucción o instrucciones faltan en el Huevo 2 del programa principal?

push [ArrayCad + esi*4]

— El registro ESP vale 0012FFC0h justo antes de ejecutar la instrucción call ExitProcess. ¿Cuál es el valor más pequeño que contendrá ESP durante la ejecución del programa? Responder en hexadecimal.

0012 FFB0h

— ¿Qué valor contendrá Maximo al final de la ejecución del programa? Responder en hexadecimal.

05h

— ¿Cuál es la codificación del -7,25 en IEEE-754? Responder en hexadecimal.

C0E80000h



— En una ALU de 5 bits similar a la de la CPU teórica se introduce en el primer operando el número -3 codificado en C-2 y en el segundo operando, el mismo número codificado en signo magnitud. Si se selecciona la operación de suma, ¿cuánto valen los flags de estado?

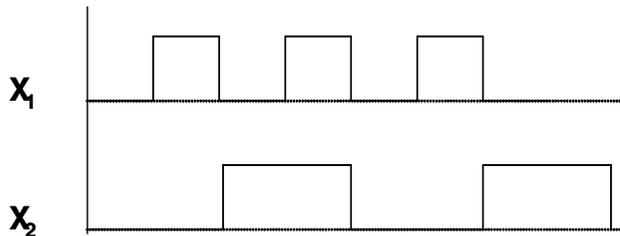
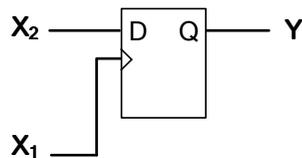
Z=0 C=1 O=0 S=1

— Se tiene un formato de coma fija con 4 bits para la parte entera y 4 bits para la parte decimal. ¿Qué error se ha cometido al representar el número 0,8 en este formato? Responder en decimal.

0,05

— En el circuito de la figura, inicialmente el biestable almacena el valor 0. Si la señal X_1 tiene una frecuencia de 2 Hz, ¿cuánto tarda la salida Y en valer 1? Indicar las unidades en la respuesta.

0,75 s



— ¿Cuántos minterms generan un valor 1 en la forma canónica de la función $S = A + (\overline{BC})$?

7

— Se tiene un dispositivo de memoria con 2 bancos y 16 chips de 32Kx4, ¿cuál es la organización del dispositivo?

64Kx32

— Si en un momento determinado el valor de SR es C=0, O=0, Z=1, S=0, ¿qué señales de control se generan en el paso T4 de la instrucción BRNZ?

FIN

— Indica cuál o cuales de las siguientes afirmaciones son ciertas. Contesta “Ninguna” si crees que ninguna es cierta.

- El código en hexadecimal del mayor número representable en formato IEEE 754 simple precisión es 7F7FFFFh.
- Los biestables R-S síncronos cambian su estado en cualquier momento a lo largo del tiempo.
- En el registro PC de la CPU teórica se almacena la dirección de los operandos de la siguiente instrucción a ejecutar por la CPU.
- UNICODE se caracteriza porque cada carácter tiene asignado un código único.

a) y d)

— Indica cuál o cuales de las siguientes afirmaciones son ciertas. Contesta “Ninguna” si crees que ninguna es cierta.

- Se puede acceder al byte menos significativo del registro EBP a través del registro BL.
- La codificación con 1 bit en complemento a 2 del número -1 es 0.
- En la sincronización de E/S programada mediante muestreo periódico, es la CPU la que activa la señal INTA.
- Un dispositivo de memoria de organización 16Kx8 a partir de chips de organización 2Kx4 necesita un decodificador tipo 2:4.

Ninguna

— Se define una nueva instrucción para la CPU elemental que permite calcular el complemento a 1 sobre una posición de memoria y almacenar el resultado en un registro dado. La sintaxis es la siguiente;

NOT Rx,[Ry]

¿De qué señales de control se compondría la instrucción? Omitir los tres primeros pasos.

Paso	Señales
4	Ry-IB, IB-MAR, READ
5	Ciclo de Espera
6	MDR-IB, TMPE-SET, XOR, ALU-TMPS, ALU-SR
7	TMPS-IB, IB-Rx, FIN



Apellidos: _____ Nombre: _____ DNI: _____

Examen de Fundamentos de Computadores – 5 de septiembre de 2009

Se ha escrito un programa en ensamblador de la CPU elemental que cuenta el número de veces que aparece una cadena en Texto. En este ejercicio la cadena que se busca es “no”. Para ello se llama al procedimiento Cuenta_no, al que se le pasan dos parámetros;

- a) Dirección del texto donde se quiere buscar la cadena. El texto finaliza con 0.
- b) Dirección de memoria donde se almacena el número de veces que se encuentra la cadena en el texto.

En la búsqueda de la cadena se considera la posibilidad de encontrar palabras con dos enes. El código del programa se muestra a continuación:

```

ORIGEN 500h
INICIO ini
.PILA 20h
.DATOS
Texto VALOR "En un lugar de la mancha de "
Num_no VALOR 0

.CODIGO
ini:
MOVH R0, BYTEBAJO DIRECCION Texto
MOVH R0, BYTEALTO DIRECCION Texto
PUSH R0 ; Apilar dirección de Texto

MOVH R1, BYTEBAJO DIRECCION Num_no
MOVH R1, BYTEALTO DIRECCION Num_no
PUSH R1 ; Apilar dirección de Num_no

Call Cuenta_no

-- HUECO 1 --
JMP -1

PROCEDIMIENTO Cuenta_no
PUSH R6
MOV R6,R7

PUSH R0
PUSH R1
PUSH R2
PUSH R3

```

```

PUSH R4
PUSH R5

; Dirección número de "no".
INC R6
INC R6
MOV R1,[R6]

; Dirección del texto donde se busca.
INC R6
MOV R3,[R6]

XOR R5,R5,R5
XOR R0,R0,R0

MOVH R2,00h
MOVL R2,'n'

Bucle: MOV R4,[R3]
no_o: COMP R4,R2
BRNZ no_n
INC R3
MOV R4,[R3]
PUSH R2
MOVH R2,00h
MOVL R2,'o' ; ASCII 6Fh

-- HUECO 2 --

BRNZ no_o
INC R5

no_n: COMP R4,R0
BRZ Final

-- HUECO 3 --

JMP Bucle

Final: MOV [R1],R5

POP R5
POP R4
POP R3
POP R2
POP R1
POP R0

POP R6
RET

```

FINP
FIN

— ¿Qué instrucción o instrucciones faltan en **Hueco 1**?

INC R7
INC R7

— ¿Qué instrucción o instrucciones faltan en **Hueco 2**?

COMP R4, R2
POP R2

— ¿Qué instrucción o instrucciones faltan en **Hueco 3**?

INC R3

— Se sabe que en la posición 0543h se encuentra el código de la instrucción D002h. ¿Qué dirección de memoria se modifica al ejecutarse la instrucción MOV [R1], R5? Contestar en hexadecimal.

Dirección: **053Ch**

— Se sabe que inicialmente R7=0594h. Cuando R7 toma el valor 0591h, ¿cuál es el contenido de ésta dirección de memoria? Contestar en hexadecimal

0544h

— Expresar el valor en hexadecimal del contenido de la dirección de memoria que corresponde a la posición de pila en el momento de mayor ocupación. ¿Qué codifica dicho valor?

Valor: **006Eh** Codifica: **n**



— El número de veces que se ha encontrado la cadena “no” en el programa de la CPU elemental se quiere visualizar en una pequeña pantalla similar a la de las calculadoras. Esta pantalla está conectada a una interfaz que contiene un registro que contiene el dato a mostrar y se ocupa de generar las señales adecuadas para que el resultado aparezca en la pantalla. El registro de datos se mapea en la dirección FA20h. Escribir la(s) instrucción(es) que sería(n) necesaria(s) al final del programa para mostrar el resultado de la cuenta.

MOVH Rx, 0FAh

MOVL Rx, 20h

MOV Ry, [R1]

MOV [Rx], Ry

Este programa lee el estado de los interruptores del periférico luces y actualiza el estado de los led de dicho periférico de acuerdo al siguiente criterio: muestra en los led el estado inverso de los interruptores. Para ello utiliza una rutina de tratamiento de interrupción denominada `rutluces`. El periférico luces se mapea en la dirección F000h.

```
ORIGEN 0A000h
INICIO main
.PILA 10

.CODIGO

main:
MOVL R0, BYTEBAJO DIRECCION rutluces
MOVH R0, BYTEALTO DIRECCION rutluces
MOVL R1, 27h
MOVH R1, 0
MOV [R1], R0
STI
JMP -1

PROCEDIMIENTO rutluces
PUSH R0
PUSH R1

-- HUECO --

POP R1
POP R0
IRET
FINP

FIN
```

— ¿Qué instrucción o instrucciones faltan en el hueco del procedimiento `rutluces`?

MOVL R0, 00h

MOVH R0, 0F0h

MOVR1, [R0]

NOT R1

MOV [R0], R1