Apellidos _____ **Nombre** _____ **DNI** _____ **A**

Examen de Fundamentos de los computadores. *Área de Arquitectura y Tecnología de Computadores* **Segundo Parcial: 29-05-2004**

A

Instrucciones generales para la realización de este examen

La respuesta debe escribirse en el hueco existente a continuación de cada pregunta **con letra clara**.

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

A un computador basado en la CPU de ejemplo se le conecta una interfaz de E/S que permite trabajar con cámaras digitales de fotografía como periféricos del computador. Dicha interfaz de E/S contiene dos elementos fundamentales:

- Una memoria interna de 3,5Kposiciones. Esta memoria se utiliza para la transferencia de imágenes desde la cámara hasta la memoria principal. Las imágenes, que siempre serán de tamaño superior a las 3,5Kposiciones se irán transfiriendo en trozos de 3,5Kposiciones.
- Un registro de estado/control. Con los bits de este registro se activan se realizan las operaciones indicadas:
  1. Bit 0: Transferir la última imagen desde la memoria de la cámara hasta la memoria de la interfaz.
  2. Bit 9: Indica que la memoria de la interfaz contiene un trozo de la imagen que se está transfiriendo.
  3. Bit 10: Se pone a uno cuando la cámara ha terminado de transferir toda la imagen solicitada.
  4. Bit 15: Capturar una imagen. Le ordena a la cámara que capture una imagen y la almacene en su tarjeta de memoria interna.

Para transferir la última imagen tomada desde la cámara hasta la memoria del computador es necesario seguir los siguientes pasos:

1. Activar el bit 0.
2. Comprobar si el bit 9 está activo. En ese caso, tenemos un trozo de 3,5Kposiciones de imagen que podemos copiar a la memoria del computador.
3. Mientras el bit 10 no se active, hay que seguir transfiriendo trozos desde la memoria de la interfaz hasta la memoria del sistema.

Para situar la interfaz en la memoria del computador es necesario asignarle un hueco de 4Kposiciones. La memoria de la interfaz aparecerá en las primeras 3,5Kposiciones mientras que el registro de estado/control aparecerá en la última posición de ese hueco de 4Kposiciones.

A continuación se muestra el código necesario para tomar una fotografía y transferirla a la memoria del computador (a la dirección apuntada por la etiqueta `IMAGEN`). Sabiendo que la interfaz se ha mapeado a partir de la posición de memoria D000h, contestar a las preguntas planteadas.

```
MOVL R0, -XX- ; Cargamos en R0 la dirección
MOVH R0, -XX- ; del registro de estado/control
XOR R1, R1, R1
MOVH R1, 80h  ; Enviamos la orden de tomar
MOV [R0], R1   ; una foto

XOR R1, R1, R1
MOVL R1, 1    ;Enviamos la orden de transferir
MOV [R0], R1  ; la imagen
; Cargamos R5 con la dirección de la
; memoria de la interfaz
--- 1 ---

MOVL R6, BYTEBAJO DIRECCION IMAGEN
MOVH R6, BYTEALTO DIRECCION IMAGEN
XOR R1, R1, R1
XOR R3, R3, R3
;Cargamos en R1 la máscara para saber si ha
;llegado un trozo de la imagen
--- 2 ---
;Cargamos en R3 la máscara para saber si se ha
;terminado la transferencia
--- 2 ---
Bucle:
---   3   ---       ;   R4   =   reg.
Estado/control
                ;Comprobamos si ha llegado
    BRNZ un_trozo   ; un trozo de imagen
    AND R4, R4, R3;Comprobamos si ha terminado
    BRZ BUCLE     ; la transferencia
    JMP fin_trans
un_trozo:
;Copiamos un trozo de imagen a su lugar
    PUSH R5
    PUSH R6
    CALL TRANSFIERE_IMG
    JMP BUCLE
fin_trans :
```

– ¿En qué dirección se encuentra mapeado el registro de estado/control de la interfaz?

DFFFh

– ¿Qué instrucción o instrucciones falta/n en **--- 1 ---**?

```
MOVL R5, 0

MOVH R5, 0D0h
```

– ¿Qué dos instrucciones faltan en los huecos marcados como **--- 2 ---**?

```
MOVH R1, 2


MOVH R3, 4
```

– ¿Qué instrucción o instrucciones falta/n en **--- 3 ---**?

```
MOV  R4, [R0]

AND R2, R4, R1
```

Se quiere diseñar un dispositivo de memoria que cubra todo el rango de direcciones de la CPU teórica. Para hacer el dispositivo se emplean chips de 4Kx8.

– ¿Cuántos chips harían falta?

32

– ¿Qué decodificador llevaría el dispositivo? Ejemplo de respuesta: 3:8.

4:16

Durante la ejecución de la primera instrucción del código que se muestra a continuación, la CPU del computador

elemental recibe una interrupción.

```
    ADD R0, R1, R2
    BRNZ no_cero
    MOVL R3, 3
    JMP sigue
no_cero:
    MOVL R3, 6
sigue:
```

En la fase de aceptación de esa interrupción y antes de que se ejecutara ninguna instrucción de la rutina de tratamiento de la interrupción, aparecieron, **no necesariamente en ese orden**, en el bus de direcciones del sistema los siguientes valores: 3A68h, 000Ch, 3A67h. Durante la misma fase, en el bus de datos del sistema aparecieron, por orden, los siguientes datos: 0001h, 2F53h, 000Ch, 7BD0h.

Con estos datos, contestar a las siguientes preguntas:

– ¿Qué valor tendrá el registro R3 al finalizar la ejecución de ese trozo de código?

6

– ¿Cuál será el valor del registro R7 al inicio de la ejecución de la instrucción de salto condicional?

3A69

– ¿Cuál es el número de interrupción que se ha generado? Contestar en decimal.

12

– Teniendo en cuenta los datos del problema y suponiendo que en el registro R0 tenemos el valor del número del vector de interrupción asociado a la interrupción generada (no siendo necesario, por tanto, inicializar ese registro) escribir el código necesario para instalar la rutina de tratamiento de la interrupción.

MOVL Rx, 0D0h

MOVH Rx, 7Bh

MOV [R0], Rx

Se ha cubierto parte del espacio de direcciones de la CPU elemental de la siguiente manera:

1. Se ha mapeado un dispositivo de memoria RAM que cubre la tabla de vectores de interrupción.
2. A continuación se ha mapeado un dispositivo E/S que tiene 4 registros mapeables.
3. El 50% superior del espacio de direcciones está cubierto por otro dispositivo de RAM.

Se quiere mapear otro dispositivo de E/S que necesita una posición del espacio de direcciones.

– ¿Cuál sería la posición ocupada por este dispositivo de E/S si se situase en las direcciones más bajas posibles? **Responder en hexadecimal.**

0104h

– ¿Cuál o cuáles de las siguientes afirmaciones son CIERTAS (puedes responder "ninguna" o “todas” si así lo consideras)?

A) En la CPU teórica, los periféricos envían a través de la línea INT el número de vector de interrupción de su rutina de servicio.
B) Las entradas del circuito de activación de dispositivos de memoria se conecta sólo a líneas del bus de direcciones de la CPU.
C) En Intel, la instrucción **ADD ESP, 4** es totalmente equivalente a **POP EAX**.
D) Los periféricos se conectan a su interfaz y a los buses de datos, direcciones y control de la CPU teórica.

B

Se ha realizado un dispositivo de memoria a partir de 4 chips de 16Kx2. Los chips se han organizado de manera que el dispositivo tuviese el mayor tamaño de palabra posible.

– ¿Cuál sería el mayor número que se podría representar con complemento a 2 en una palabra de ese dispositivo? **Responder en decimal.**

127

– ¿Cuál o cuáles de las siguientes afirmaciones son CIERTAS (puedes responder "ninguna" o “todas” si así lo consideras)?

A) Si EBX vale 00000010h, ESI vale 00000020h y la etiqueta **temp** apunta a la dirección 00AB0020h, al ejecutar **MOV [temp+EBX+ESI*2], BL** se escribirá en la dirección 00AB0070h.
B) La instrucción **JA salto** hace lo mismo que la instrucción **JNBE salto**.
C) Como la arquitectura Intel x86 tiene 32 líneas de direcciones y en cada posición de memoria se almacena una palabra (16 bits), si todo el espacio de direcciones estuviese rellenado con memoria se podrían tener 8 GBytes de memoria.
D) Un chip de memoria de 16Kpalabras se puede mapear a partir de 8 posiciones distintas del espacio de direcciones de la CPU teórica.

A, B

Se han utilizado 4 chips de memoria de 2Kx16 para diseñar un dispositivo de memoria. Este dispositivo se ha conectado a una CPU con un bus de direcciones de 14 líneas. Se ha visto que el dispositivo cubre el rango de direcciones 0000h-0FFF.

– ¿Cuántos bancos tiene el dispositivo?

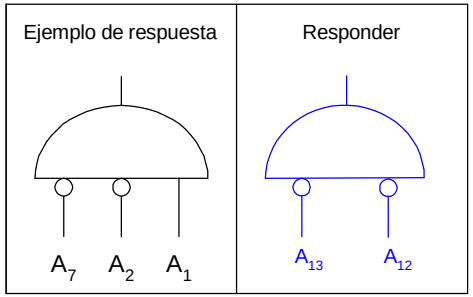
2

A

Apellidos _____ **Nombre** _____ **DNI** _____ **A**

Examen de Fundamentos de los computadores. *Área de Arquitectura y Tecnología de Computadores* **Segundo Parcial: 29-05-2004**

– Dibujar la puerta AND que debe actuar como circuito de activación del dispositivo. (En el ejemplo de respuesta se muestran tres líneas de dirección, pero puede ser cualquier otro número; indicar su peso y si entran negadas o no a la puerta.)



Se ha creado un programa en ensamblador para los procesadores Intel x86 que es capaz de invertir cadenas de caracteres terminadas con el valor de un byte 00h.

El programa utiliza dos procedimientos:

- Procedimiento `longitud`: recibe un único parámetro de 32 bits a través de la pila (la dirección inicial de la cadena de caracteres) y devuelve la longitud de la cadena de caracteres en el registro `eax` (interpretándose la longitud como un entero sin signo de 32 bits). Si, por ejemplo, se definiese en la sección de datos cadena DB "abc", 0 , el procedimiento devolvería a través de `eax` el valor `00000003`.
- Procedimiento `invierte`: recibe dos parámetros a través de la pila, que deben ser apilados por el procedimiento llamador en el orden en que son comentados a continuación:
  - El primer parámetro es un valor de 32 bits que el procedimiento interpreta como la dirección inicial de la cadena de caracteres a invertir.
  - El segundo parámetro es la longitud de la cadena anterior, especificada como un entero sin signo de 32 bits.

El procedimiento recorre la cadena, tomando inicialmente el primer carácter e intercambiándolo con el último, para después continuar con el segundo e intercambiarlo con el penúltimo, y así consecutivamente hasta llegar a la mitad de la cadena. En ese instante, se detiene el recorrido de la cadena pues ya estarán intercambiados todos los caracteres que la conformaban inicialmente.

El programa que ha sido creado, se encarga de ir llamando a estos dos procedimientos, de forma que se inviertan tres cadenas definidas en la sección de datos.

A continuación, se muestra el código fuente de este programa, en el que faltan algunas instrucciones.

```
.386
.MODEL FLAT
EXTERN ExitProcess:PROC
.DATA
cadena1 DB "Hola amigos 123",0
cadena2 DB "29-mayo-2004",0
cadena3 DB "Asturias patria querida",0

.CODE
; Procedimiento que calcula la longitud
; de una cadena
longitud PROC
  push ebp
  mov ebp, esp
  push edi
  push ebx
  mov edi, [ebp+8] ; Mover a edi 1er par.
  xor  eax,  eax  ;Inicializar  longitud  a
cero

bucle_contador:
     mov bl, [edi]
     cmp bl, 0
     je SHORT fuera
     inc eax
     inc edi
     jmp SHORT bucle_contador

fuera:
   ------- HUECO 1 -------
longitud ENDP

;    Procedimiento    que    invierte    los
caracteres
; de una cadena
invierte PROC
  push ebp
  mov ebp, esp
  push ebx
  push ecx
  push edi
  push esi

  ; Copiar la long. de la cad. desde la
  ; pila a ECX y copiar la dir. inicial
  ; de la cadena desde la pila a EDI
   ------- HUECO 2 -------

  ; Hacer que el registro esi apunte al
  ; ultimo carácter de la cadena
  mov esi, edi
  add esi, ecx
  dec esi

  ; Dividir entre dos la longitud de la
  ; cadena para saber cuándo parar de
  ; intercambiar caracteres
  shr ecx, 1



   ; Bucle principal de intercambio
   ; de caracteres
intercambiar:
  ; Intercambiar los 2 caracteres que toquen
  mov bl, [esi]
  mov bh, [edi]
  mov [esi], bh
  mov [edi], bl

  ; Hacer que esi y edi apunten a los 2
  ; caracteres siguientes a intercambiar

   ------- HUECO 3 -------

  loop intercambiar

  pop esi
  pop edi
  pop ecx
  pop ebx
  pop ebp
   ------- HUECO 4 -------
invierte ENDP

; Programa principal
inicio:
; Invertir cadena1
```

```
        push OFFSET cadena1
        call longitud
        push OFFSET cadena1
        push eax
        call invierte

; Invertir cadena2
        push OFFSET cadena2
        call longitud
        push OFFSET cadena2
        push eax
        call invierte

; Invertir cadena3
        push OFFSET cadena3
        call longitud
        push OFFSET cadena3
        push eax
        call invierte

        push 0
        call ExitProcess
END inicio
```

La figura muestra una captura de pantalla en un instante determinado de la ejecución del programa dentro del depurador Turbo Debugger (TD32.EXE).

– ¿Cuál es el valor inicial del contador del programa al comenzar a ejecutar el programa?

00401043h

– ¿Qué instrucción o instrucciones faltan en el **HUECO 1**?

```
pop ebx

pop edi

pop ebp

ret 4
```

– ¿Qué instrucción o instrucciones faltan en el **HUECO 2**?

```
mov ecx, [ebp+8]
```

```
mov edi, [ebp+12]
```

– ¿Qué instrucción o instrucciones faltan en el **HUECO 3**?

```
inc edi

dec esi
```

– ¿Qué instrucción falta en el **HUECO 4**?

```
ret 8
```

– ¿Cuál es el valor del registro puntero de pila justo después de ejecutarse la primera instrucción **call invierte** del programa principal?

0012FFB8h

– ¿Cuál es la magnitud del salto (valor que se suma al registro contador de programa) en la segunda instrucción **call invierte** del programa principal? **Responder en DECIMAL**.

-82

– ¿Cuál es el código ASCII de la letra **g**? **Responder en hexadecimal**.

67h