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

Examen de Fundamentos de los computadores. *Área de Arquitectura y Tecnología de Computadores* **Segundo Parcial: 04-06-2005**

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.

El fabricante de automóviles Mercedes ha contratado a la Universidad de Oviedo para que diseñe el sistema informático para su nuevo modelo de coche. Para construir el primer prototipo se va a utilizar una CPU teórica a la que se conectan, al menos, los siguientes periféricos:

- Una pantalla, cuyo funcionamiento e interfaz es el visto en clase
- Un conjunto de sensores que miden el nivel de gasolina, aceite, agua, etc... del coche

Todos los sensores del coche se conectan al mismo interfaz (interfaz del coche), que ocupa 4K en el E.D. y uno de cuyos registros indica el estado en el que se encuentran dichos sensores. Se sabe que el bit 0 del registro se pone a 1 si hay un nivel bajo de gasolina y que el bit 11 se pone a 1 si hay un nivel bajo de aceite. Los demás bits del registro indican los niveles de otras variables.

Cada vez que cambia el contenido del registro de estado se genera una interrupción. Cuando se produce la interrupción, la CPU borra el contenido de la pantalla y a continuación muestra un mensaje de alerta por cada sensor que ha detectado un nivel bajo (es decir, con su bit de estado a 1). Para mostrar el mensaje en pantalla se tiene un procedimiento, llamado *escribe*, que recibe como parámetro la dirección de memoria en la que se encuentra el texto de la alerta.

La rutina de servicio para la interrupción generada por la interfaz del coche es la que se muestra a continuación:

```
PROCEDIMIENTO rut_alertas

;Guardar el valor original de los registros
utilizados por la rutina
    PUSH R0
    PUSH R1
    PUSH R2

;Cargar en R0 el contenido del registro de
estado de los sensores del coche
    MOVL R0, 12h
    MOVH R0, F1h
    MOV R0, [R0]

;Cargar en R1 la dirección del registro de
control de la pantalla
    --- 1 ---

;Borrar el contenido de la pantalla
    MOVL R2, -xx-
    MOVH R2, -xx-
    MOV [R1], R2

;Cargar en R1 la máscara para detectar si
está bajo el nivel de gasolina
    MOVL R1, -xx-
    MOVH R1, -xx-

;Si está activa la alerta de gasolina,
mostrar el mensaje de aviso en pantalla
    AND R1, R0, R1
    BRZ continua_1
    --- 2 ---
    CALL escribe
    INC R7

continua_1:
;Procesar el resto de alertas, leyendo su
estado y mostrando los mensajes en pantalla
;...

;Finalizar la rutina de servicio
    --- 3 ---
FINP
```

– ¿Qué máscaras se tienen que utilizar para detectar un nivel bajo de gasolina y un nivel bajo de aceite? Expresar el resultado en hexadecimal

Gasolina:	0001h
Aceite:	0800h

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

POP R2

POP R1

POP R0

IRET

– Teniendo en cuenta que la segunda línea de la pantalla empieza en la posición de memoria *A10Fh*, ¿qué instrucción o instrucciones falta/n en **--- 1 ---**?

MOVL R1, 78h

MOVH R1, 0A1h

– Las direcciones del E.D. que no están ocupadas por los interfaces de E/S se quieren cubrir utilizando **únicamente** módulos de memoria de tamaño 16K. ¿Qué rango de direcciones estará ocupado por el sistema de memoria? Ejemplo: 0100h - 0AFFh

0000h - 7FFFh

– Si el bit 0 del registro de control de la pantalla sirve para apagar/encender la pantalla, y el bit 1 sirve para borrar su contenido, ¿qué palabra tiene que escribir la rutina de servicio en el registro de control de la pantalla? Contestar en hexadecimal

0003h

A

– Justo antes de ejecutarse la instrucción AND R1, R0, R1 de la rutina de servicio "rut_alerta" el contenido de la pila era el siguiente (ordenado de posiciones de memoria inferiores a superiores): 02FFh, C000h, 230Fh, 2F53h, 0001h, 3A00h. La interrupción que desencadenó la llamada a "rut_alerta" se produjo mientras la CPU estaba ejecutando la instrucción BRC +10h, que forma parte del programa principal. Teniendo todo esto en cuenta, ¿en qué posición de memoria se encuentra la instrucción que se estaba ejecutando cuando se produjo la interrupción?

2F52h

– Suponiendo que en la sección de datos se tiene definida la etiqueta *mensaje_1* para el mensaje de alerta "Nivel bajo de aceite", ¿qué instrucción o instrucciones faltan en **--- 2 ---**?

MOVL Rx, BYTEBAJO DIRECCION mensaje_1

MOVH Rx, BYTEALTO DIRECCION mensaje_1

PUSH Rx

NOTA: Rx con X=0,1,2

A continuación se muestra parte del código que se utiliza para que la CPU sea capaz de procesar las interrupciones generadas por la interfaz del coche.

```
--- 4 ---
MOVL R1, BYTEBAJO DIRECCION rut_alertas
MOVH R1, BYTEALTO DIRECCION rut_alertas
MOV [R0], R1
```

– ¿Qué instrucción o instrucciones falta/n en **---4---** para que la rutina de servicio se instale en la última posición de la TVI?

MOVL R0, ffh

MOVH R0, 00h

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

A) La técnica de Entrada/Salida programada consume menos recursos de CPU que la técnica de Entrada/Salida basada en interrupciones.
B) Las entradas del circuito de activación de dispositivos interfaces de periféricos mapeables en el espacio de direcciones del computador elemental, se conectan sólo a líneas del bus de direcciones del sistema.
C) Un Lenguaje Ensamblador es un lenguaje de programación que utiliza sólo instrucciones de una determinada CPU y, por tanto, se define siempre para una CPU concreta.
D) La implementación de un bucle de código para la arquitectura Intel necesita, obligatoriamente, que el número de iteraciones de dicho bucle sea cargado en el registro ECX.

B, C

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

A) La instrucción ***CALL Etiqueta*** funciona de forma idéntica a la instrucción ***JMP Etiqueta***, sólo que después de saltar guarda el valor del PC renovado en la pila.
B) El formato de almacenamiento *Big Endian*, utilizado por la arquitectura Intel, significa que la dirección que identifica en memoria a las palabras y a las dobles palabras es la de su byte menos significativo.
C) En la instrucción de retorno de un procedimiento de la arquitectura Intel, ***RET n***, ‘*n*’ indica el número de parámetros (habitualmente de 32 bits) que serán eliminados de la pila al retornar de dicho procedimiento.
D) El tamaño del espacio direccionable de un computador depende del ancho de su bus de direcciones y del ancho de su bus de datos.

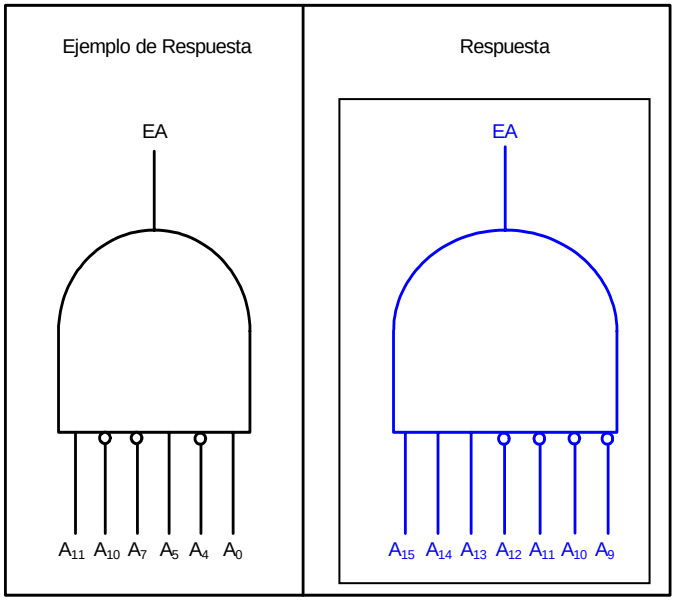
Ninguna

En los últimos meses se han registrado en España varias retenciones kilométricas debidas, por un lado, a causas meteorológicas y, por otro, a salidas y regresos de vacaciones. Por ello, y para implementar un sistema capaz de monitorizar el tráfico en los principales puntos conflictivos del país, la DGT estudia la posibilidad de utilizar un computador elemental. A este computador pretende conectar un dispositivo periférico encargado de recibir información de cada uno de dichos puntos de la red viaria nacional. El interfaz de este periférico consta de 296 registros de datos y 1 registro de control, todos ellos elementos mapeables. Cada uno de los registros de datos almacena el número de kilómetros de retención que hay en el punto conflictivo al que representa.

– ¿Cuál es el número mínimo de líneas del bus de direcciones necesario para direccionar todos los registros de este interfaz?

$2^a \geq 297 \rightarrow a = 9$

– ¿Cuál sería el circuito de activación del interfaz necesario para mapear el dispositivo a partir de la dirección E000h?



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

Examen de Fundamentos de los computadores. *Área de Arquitectura y Tecnología de Computadores* **Segundo Parcial: 04-06-2005**

Se desea construir un dispositivo de memoria para la CPU elemental que cubra el 25% superior de su espacio direccionable. Para la construcción de este dispositivo se utilizan chips cuya organización es 4096x8.

- ¿Cuántos chips serían necesarios?

((64Kx0.25)/4K)x(16/8) = 8 chips

- Tras la construcción del dispositivo se detecta un fallo en el byte más significativo de la palabra almacenada en la dirección D3DFh, ¿qué chip está funcionando de forma incorrecta? [Los chips se numeran de 1 a N, de izquierda a derecha por columnas y de arriba a abajo por filas].

3

Tras la ejecución del siguiente conjunto de instrucciones para la arquitectura Intel:

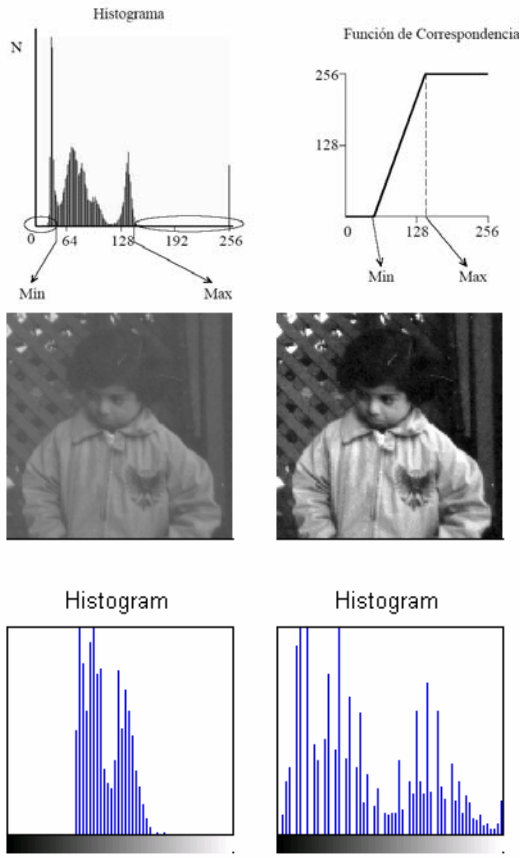
```
        XOR     EAX, EAX
        MOV     EBX, 10h
        MOV     ECX, 08h

bucle:  INC     EAX
        SHR     EBX, 1
        JZ      fin
        LOOP    bucle

fin:    MOV     EDX, EAX
```

- ¿Cuál será el contenido del registro EDX?

00000005h

Pensando en la optimización de un programa de tratamiento de imágenes se ha pensado en migrar alguno de las efectos que proporciona el programa a código ensamblador para los procesadores Intel x86.

Se ha comenzado por una función que mejora los niveles de contraste de la imagen, se trata de la **expansión de histograma**. Esta función permite que los tonos de la imagen origen ocupen todo el rango de niveles de gris que el formato de la imagen permita. Se va a trabajar con un formato de imagen que tiene 256 niveles de grises: desde el tono negro que tomará el valor cero hasta el nivel blanco que tendrá el valor 255. En la figura siguiente se muestra esquemáticamente lo que realiza esta función así como el efecto de su aplicación:



La **expansión de histograma** utiliza una función de correspondencia para convertir cada tono de la imagen original a la que le corresponda de manera que la imagen ocupe todos los niveles de grises. Esta función de correspondencia tiene la expresión:

$$y = A \cdot x - A \cdot Min\,, \qquad \text{donde } A = 255 \big/ (Max - Min);$$

Max es el mayor nivel de gris que presenta la imagen origen, Min es el menor nivel de gris diferente de cero de la imagen origen, ***x*** es el nivel de gris de cada píxel de la imagen origen e ***y*** representará el nivel de gris de la imagen final.

Para realizar la **expansión de histograma** vamos a utilizador dos procedimientos:

- Procedimiento *BuscaMaximoMinimo*, realizará una búsqueda por todos los píxeles de la imagen origen buscando el nivel de gris máximo y el nivel mínimo de gris no nulo. Recibe como parámetros y en el orden en que son comentados:
  - Dirección donde se almacena el ***Mínimo***
  - Dirección donde se almacena el ***Máximo***
  - Dirección donde se encuentra la imagen origen
  - Número de píxeles que tiene la imagen origen
- Procedimiento *AplicaFuncionCorrespondencia,* recorrerá la imagen origen y a cada píxel convierte su nivel de luminosidad de acuerdo a la función que anteriormente hemos descrito. Recibe como parámetros y en el orden en que son comentados:
  - Dirección donde se encuentra el ***Mínimo*** de la luminosidad de la imagen origen.
  - Dirección donde se encuentra el ***Máximo*** de la luminosidad de la imagen origen.
  - Dirección donde se encuentra la imagen origen
  - Dirección donde se encuentra la imagen destino
  - Número de píxeles que tiene tanto la imagen origen como la imagen destino

.

Para comprobar el funcionamiento de los dos procedimientos se ha creado un programa cuyo código de forma parcial se reproduce a continuación. Se define un par de vectores que harán las veces de imagen origen y de imagen final. Cada elemento del vector representa un píxel de la imagen. Sobre

estos vectores se ejecutaran los dos procedimientos para la búsqueda del máximo y mínimo y la aplicación de la función de correspondencia.

```
.386
.Model flat
Extern ExitProcess:PROC


.DATA
ImagenOrigen  DB  152, 0, 120, 30,57, 67,
                  64, 250
ImagenFinal   DB 8 DUP(0FFh)
ElMaximo      DB  0
ElMinimo      DB  0

.CODE
inicio:

    ;ponemos parámetros en la pila

   ------- HUECO 1 -------

    call BuscaMaximoMinimo


    ;ponemos parámetros en la pila

   ------- HUECO 2 -------

    call AplicaFuncionCorrespondencia


    ; Retorno al S. O.
    push 0
    call ExitProcess
    NOP


BuscaMaximoMinimo PROC

    push ebp
    mov ebp,esp

    ;Guardamos   en   pila   registros   que
usaremos
    push edi ; dirección imagen origen
    push ebx ; dirección del máximo
    push edx ; dirección del mínimo
    push ecx ; No. Pixeles imagen
    push esi ; indice para recorrer imagen
    push eax ; registro auxiliar

    ;recogemos de la pila los parámetros

    ;edi el puntero a la imagen origen
    ;ebx dirección de escritura del máximo
    ;edx dirección escritura del mínimo
    ;ecx No. Píxeles imagen

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

    ;limpiamos los registros que usaremos
    xor esi,esi
    xor eax,eax

    ;inicializamos el máximo
    mov [ebx],BYTE PTR 00h
    ;inicializamos el mínimo
    mov [edx],BYTE PTR 0FFh

bucle: ;bucle para recorrer la imagen

    ;al almacenamos luminosidad del pixel
    mov al,BYTE PTR [edi+esi]

    ;comprobamos si es cero la luminosidad
    cmp al,BYTE PTR 00h
    JE SHORT salto

    ;comprobamos si es mayor que el máximo
    ;actual
    ------- HUECO 4 -------

    ;comprobamos si es minimo
comprobar_minimo:
    cmp al,[edx]
    jb SHORT ES_MINIMO
    jmp SHORT salto

ES_MAXIMO:
    ;Fue máximo ⇔ actualizamos el máximo y
    ;tenemos que comprobar si fue mínimo
    ------- HUECO 5 -------

ES_MINIMO:
    ;Fue mínimo ⇔ actualizamos el mínimo
    mov [edx],al

salto:
    ;avanzamos por la imagen
    inc esi
    ;comprobamos si hemos recorrido toda la
    ;imagen
    loop bucle
    ;recuperamos registros de la pila
    pop eax
    pop esi
    pop ecx
    pop edx
    pop ebx
    pop edi
    pop ebp

    ;retornamos y limpiamos los parámetros
    ------- HUECO 6 -------

BuscaMaximoMinimo    ENDP

AplicaFuncionCorrespondencia PROC

;;;;;;;;;;;;;;;;;;;;;;;;;
; CODIGO OMITIDO INTENCIONADAMENTE
;;;;;;;;;;;;;;;;;;;;;;;;;

AplicaFuncionCorrespondencia ENDP

END inicio
```

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

```
push OFFSET ElMinimo
push OFFSET ElMaximo
push OFFSET ImagenOrigen
push 8
```

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

```
push OFFSET ElMinimo
push OFFSET ElMaximo
push OFFSET ImagenOrigen
push OFFSET ImagenFinal
push 8
```

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

```
mov edi,[ebp+12]
mov ebx,[ebp+16]
mov edx,[ebp+20]
mov ecx,[ebp+8]
```

– ¿Qué instrucción o instrucciones faltan en el **HUECO 4**?

```
cmp al,[ebx]
ja SHORT ES_MAXIMO
```

– ¿Qué instrucción o instrucciones faltan en el **HUECO 5**?

```
mov [ebx],al
jmp SHORT comprobar_minimo // jmp SHORT salto
```

– ¿Qué instrucción falta en el **HUECO 6**?

```
ret 16
```

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

– ¿Cuánto valdrá el contador de programa después de ejecutar la instrucción **JMP SHORT salto** del procedimiento *BuscaMaximoMinimo*? Expresar en hexadecimal

0040106Fh

– ¿Cuál es el valor inicial del puntero de pila antes de ejecutar la primera instrucción del procedimiento *BuscaMaximoMinimo*? Expresar en hexadecimal

0012FFB0h

En los laboratorios de la universidad se está diseñando un nuevo computador destinado a resolver un tipo de problema específico. El espacio direccionable de este computador es la cuarta parte del espacio direccionable de la arquitectura Intel. Se decide llenar todo este espacio con memoria RAM, utilizando dispositivos construidos a partir de chips cuya organización es 256Mx4. Teniendo en cuenta que cada posición de memoria del nuevo computador almacena el doble de información que una posición de memoria de un computador basado en la arquitectura Intel,

– ¿cuántos chips serían necesarios para implementar su sistema de memoria?

((4096/4)/256)x(16/4) = 16