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.

- Una aplicación de videoconferencia transmite de un cliente a otro, fotograma de dimensiones 320 x 240 píxeles (largo x alto) en escala de grises. Cada píxel de la imagen se representa mediante un valor en el rango [0,255]. La imagen original pasa previamente antes de llegar a su destino por un servidor que entre otras funciones agrega un logotipo a la imagen en la posición (x,y)[1] = (5,5), que llamaremos "mosca". La "mosca" en este caso es el logotipo de la Universidad de Oviedo. Las dimensiones de la mosca son de 10 x 10 píxeles. Dado que la imagen original será procesada con más finalidades, la nueva imagen con el logotipo es una copia de la original.

En la figura se puede observar un ejemplo.

Dado que la aplicación que reside en el servidor es poco eficiente debido a la calidad del código generado, utilizado en el desarrollo de dicha aplicación, se ha pensado en migrar esta funcionalidad a lenguaje ensamblador de Intel x86.

Las imágenes (original y mosca) se almacenan en dos vectores de manera lineal. Primero se almacena la primera línea, luego la segunda, …

[1] La coordenada (0,0) es la esquina superior izquierda

El nuevo programa mostrado a continuación tiene dos procedimientos. El primero, **CalculaInicio**, calcula el número de posiciones de memoria "a saltar" hasta la posición donde se comenzará a escribir la "mosca". Este procedimiento recibe como parámetros:

- Coordenada X donde se empieza a escribir la "mosca".
- Coordenada Y donde se empieza a escribir la "mosca".
- Dirección de la variable Npos_saltar que es la posición donde se va a escribir la "mosca"

El segundo, **CreaImagen**, copia la imagen original junto con la mosca a una nueva imagen. Recibe los siguientes parámetros en este orden

- Dirección a partir de la cual está almacenada la imagen original
- Dirección a partir de la cual se guardará la imagen final
- Dirección a partir de la cual está almacenada la "mosca"
- Posición en la que comienza a escribirse la "mosca".
- Nº elementos de la imagen "mosca" (tamaño)
- Nº de elementos de la imagen original (tamaño)

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

.DATA
ImagenOrigen   DB 152,0,120,30,57,93,[…]
ImagenFinal    DB 76800 DUP (0)
Mosca           DB 0,0,0,0,0,0,12,158,26[…]
Npos_saltar    DB 0
   ------- HUECO 1 -------

     ;ponemos parámetros en la pila
   ------- HUECO 2 -------

     call CalculaInicio

     ;ponemos parámetros en la pila
   ------- HUECO 3 -------
     call CreaImagen
      ; Retorno al S. O.
      push 0
      call ExitProcess
      NOP

CreaImagen PROC

      push ebp
      mov ebp,esp

;Guardamos en pila registros que usaremos
      push edi ; dirección imagen origen
      push edx ; dirección imagen final
      push ebx ; dirección mosca
      push ecx ; No. elem. imagen original
      push eax ; Variable temporal
      push esi ; No. elementos mosca

;recogemos de la pila alguno de los parámetros
    ;edi el puntero a la imagen origen
    ;ebx el puntero a mosca
    ;edx puntero a imagen final
    ;ecx No. elem. iImagen original
    ;esi No. elem. imagen mosca

    ------- HUECO 4 -------

;Se realiza una copia de la imagen original en la
imagen final

COPIA:
     mov al, [edi]
     mov [edx], al
      inc edi
      inc edx
      loop COPIA

      ;Se inicializa de nuevo el valor del
registro     ecx con el número de posiciones de
memoria que hay que saltar para escribir la
"mosca" y edx con la dirección de la primera
posición de memoria de la imagen final (puntero a
imagen final)

    ------- HUECO 5 -------

; Sobre la imagen final se saltan las posiciones
de memoria necesarias para escribir la "mosca"

SALTO_POSS:
      inc edx
      loop SALTO_POSS
```

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

Examen de Fundamentos de los computadores. *Área de Arquitectura y Tecnología de Computadores* **Segundo Parcial: 10-06-2006**

```
;Se escribe una línea de la "mosca"
;En la 1ª iteración se escribe la línea 1

MAS_LINEAS:
      mov ecx, 10

LINEA_MOSCA:
    ------- HUECO 6 -------
      inc edx
      inc ebx
      loop LINEA_MOSCA
      ;Avanzar sobre la imagen final hasta la
siguiente posición donde se escribiría la
siguiente línea de la mosca

    add edx,136h ;; 136h = 310d

    ;Comprobar si quedan más líneas de la mosca
por escribir en cuyo caso pasar a la siguiente
línea y escribirla
    ------- HUECO 7 -------
      pop esi
      pop eax
      pop ecx
      pop ebx
      pop edx
      pop edi
      pop ebp

;Finalizar el procedimiento y limpieza de
parámetros de la pila
     ------- HUECO 8 -------
CreaImagen    ENDP


CalculaInicio PROC
;; CODIGO OMITIDO INTENCIONADAMENTE
ret 12
CalculaInicio ENDP

END inicio
```

– ¿Qué instrucción (es) faltan en el **HUECO 1**?

```
.CODE
inicio:
```

– ¿Qué instrucción (es) faltan en el **HUECO 2**?

```
push 5
push 5
push OFFSET Npos_saltar
```

– ¿Qué instrucción (es) faltan en el **HUECO 3**?

```
push OFFSET ImagenOrigen
push OFFSET ImagenFinal
push OFFSET Mosca
push Npos_saltar
push 64h   ;; 100 en decimal
push 12C00h  ;; 76800 en decimal
```

– ¿Qué instrucción (es) faltan en el **HUECO 4**?

```
mov edi,  [ebp+28]
mov ebx, [ebp+20]
mov edx, [ebp+24]
mov ecx, [ebp+8]
mov esi, [ebp+12]
```

– ¿Qué instrucción (es) faltan en el **HUECO 5**?

```
mov ecx, [ebp+16]
mov edx, [ebp+24]
```

– ¿Qué instrucción (es) faltan en el **HUECO 6**?

```
mov al, [ebx]
mov [edx], al
dec esi   ;; Para no restar 10 en hueco 6
```

– ¿Qué instrucción (es) faltan en el **HUECO 7**?

```
sub esi, 0Ah ;; (10d)
cmp esi,0 ;; Para comparar com 0 si se resta en el hueco 6
jnz MAS_LINEAS
```

– ¿Qué instrucción (es) faltan en el **HUECO 8**?

```
ret 24
```

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

A) Justo después de retornar de una rutina de interrupción, el flag IF está siempre a uno.
B) Las interrupciones son la única forma de sincronizar la comunicación entre la cpu y los periféricos.
C) El valor de un vector de interrupción indica la prioridad de la interrupción asociada a ese vector.
D) La rutina de servicio de una interrupción puede recibir parámetros a través de la pila.

A

El espacio de direcciones de un ordenador basado en la CPU elemental está organizado de la forma siguiente:

- La zona de direcciones más bajas está ocupada por un módulo de memoria de 16K
- Justo a continuación se encuentra un módulo de memoria de 32K
- Se reservan las últimas 16K posiciones de memoria para mapear diferentes periféricos.

– En este ordenador se quiere cargar un programa que ocupa 1K en memoria. ¿Qué rango de valores se puede utilizar en la directiva ORIGEN de dicho programa? Ejemplo: 0A00h-A000h

0100h - BC00h

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

Examen de Fundamentos de los computadores. *Área de Arquitectura y Tecnología de Computadores* **Segundo Parcial: 10-06-2006**

– Se dispone de chips de memoria de 4Kx4 para construir un módulo de memoria que cubra el espacio de direcciones completo de la CPU teórica. Si cada chip cuesta 0.50€, ¿cuánto costarán los chips necesarios para construir el módulo de memoria?

32€

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

A) En Intel son necesarias cuatro posiciones de memoria para almacenar un número entero del mismo tamaño que el número entero más grande que se puede almacenar en una sola posición de memoria de la CPU elemental.
B) La directiva **.MODEL** de Intel indica el modelo de CPU en el que se tiene que ejecutar el programa
C) La instrucción **SAL** de Intel es equivalente a la instrucción **SHL**
D) El formato *little endian* significa que los bytes de peso más bajo se almacenan en posiciones de memoria más bajas.

C, D

– ¿Cuántos bancos son necesarios para completar la memoria de un dispositivo que recibe 22 líneas de dirección y 8 líneas de datos si se utilizan chips de 512Kx1? ¿Y cuántos chips?

Bancos: 8 Chips: 64

– Una función comienza por las intrucciones **push ebp** y **mov ebp, esp** y finaliza con **pop ebp** y **ret**. Justo antes de ejecutar la instrucción **ret** se conocen los siguientes valores de registros: ESP=12FFB4h, EBP=12FFF0h y EIP=401049h. ¿Cuánto valía ESP justo antes de la instrucción **mov ebp, esp**? Responder en hexadecimal.

ESP=12FFB0h

– En la tabla se representa el estado inicial de la zona de memoria especificada antes de la ejecución del siguiente fragmento de código:

```
XOR EAX,EAX
XOR EDI,EDI
MOV EDI,1h
MOV AX,[lista]
MOV [lista + EDI*4],AL
```

Sabiendo que la etiqueta lista hace referencia a la posición de memoria 0040200h, completar la tabla tras la ejecución del fragmento de código.

	Antes	Después
00402000h	ABh	ABh
00402001h	12h	12h
00402002h	CDh	CDh
00402003h	34h	34h
00402004h	EFh	ABh
00402005h	56h	56h

– La empresa CONTROLA, S.A. nos ha encargado una colaboración en un proyecto en el cual está desarrollando un novedoso sistema de control de acceso a recintos. El sistema constará de un lector de tarjetas conectado a un computador basado en la CPU teórica que lee un código de 16 bits almacenado en cada tarjeta por radiofrecuencia, no siendo necesario que la persona que la lleve se acerque al lector. El sistema detectará de manera automática si la persona tiene o no acceso autorizado mostrando esta información en una pantalla y abriendo la puerta en caso positivo.

Para realizar un primer prototipo utilizaremos el modelo de CPU teórica al cual se conectarán los siguientes elementos de E/S:
- Un lector de tarjetas que consta de:
  1. Un registro de datos de 16 bits.
  2. Un registro de control/estado, de 16 bits, que dispondrá de los siguientes bits:
     a) Bit 0: se activa (pone a 1) cuando se desee abrir la puerta.
     b) Bit 2: se activa (pone a 1) cuando el lector lee una tarjeta y su código de acceso almacenado en el registro de datos. Se desactiva (pone a 0) cuando la CPU lee el registro de datos.

- Una pantalla similar a la vista en clase, pero con distintas dimensiones: 5 filas por 25 columnas.

Para controlar el acceso al recinto se realizarán las siguientes operaciones:
1.- Mediante muestreo periódico del registro de control/estado se detecta que se ha leído una tarjeta y su código ha sido almacenado.
2.- En este caso se ejecutará el procedimiento acceso, que comprueba si el código está en la lista de códigos autorizados en cuyo caso muestra la información de aceptación o rechazo en pantalla y abre la puerta en caso necesario.
Los registros de datos y de control/estado del lector se mapean en ese orden y a partir de la dirección C000h.
A continuación se muestra una parte del código de procedimiento de acceso:

```
PROCEDIMIENTO acceso

;Guardar el valor de los registros utilizados
por la rutina
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4
PUSH R5

; Cargar en R1 la dirección del registro de
estado/control de la pantalla
MOVH R1, FFh
MOVL R1, FEh

; Borrar el contenido de la pantalla
MOVL R2, 03h
MOVH R2, 00h
MOV [R1], R2

; Cargar en R0 el contenido del registro de
datos del lector
```

--1--

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

Examen de Fundamentos de los computadores. *Área de Arquitectura y Tecnología de Computadores* **Segundo Parcial: 10-06-2006**

```
; Cargar en R2 la dirección del primer código de la lista
                    --2--
;Inicialización número de códigos de la lista
MOVL R3, 16h
MOVH R3, 00h

;Cargar en R5 la primera posición de pantalla
                    --3--
;Bucle de aceptación o denegación de permiso
bucle:
MOV R4, [R2]
COMP R0, R4
BRNZ salto

;resto de código hasta terminar.............
FINP
```

– ¿Qué instrucción/es falta/n en el hueco **-1-**?

MOVL Rx, 00h

MOVH Rx, 0C0h

MOV R0, [Rx]

– Teniendo en cuenta que los códigos de acceso autorizado se definen en la sección de datos con la etiqueta `lista`. ¿Qué instrucción/es falta/n en **-2-** ?

MOVL R2, BYTEBAJO DIRECCION lista

MOVH R2, BYTEALTO DIRECCION lista

– ¿Qué instrucción/es falta/n en el hueco **–3-**?

MOVH R5, FFh

MOVL R5, 81h

– Las direcciones del E.D. no ocupadas no ocupadas por interfaces se pretenden cubrir utilizando únicamente módulos de memoria de 16k. ¿Qué rango de direcciones ocupará el sistema de memoria?. Ejemplo 0000h-1000h.

0000h-BFFFh

– Justo antes de ejecutarse la instrucción MOV R4, [R2], parte del contenido de la pila era el siguiente (según el orden de llenado de la pila, terminando en el valor de [R7]):A001h, B010h, 1000h, 0001h, F0FFh, 8147h. ¿Cuál era el valor del registro R4 en el programa principal justo antes de acceder al procedimiento?

F0FFh

– Se ha modificado el lector de tarjetas para que en caso de detectar una tarjeta, genere automáticamente una interrupción. A la rutina de interrupción se le ha asignado el vector de interrupción 7.

Completar el código necesario para instalar “acceso” como rutina de interrupción.

MOVL R0, 07h

MOVH R0, 00

MOVL Rx, BYTEBAJO DIRECCION acceso

MOVH Rx, BYTEALTO DIRECCION acceso

MOV [R0], Rx

– ¿Qué instrucción de “`acceso`” debe modificarse y por qué nueva instrucción, para que pueda funcionar como una rutina de interrupción?

Antigua: RET Nueva: IRET

– En la primera ejecución del nuevo programa se produce una interrupción y durante la fase de desencadenamiento (antes de ejecutar ninguna instrucción) de la misma aparecen en la pila los siguientes datos (ordenados según orden de llenado terminando en la posición de [R7]) F001h, 0001h, 0502h.

¿Qué dato enviará el lector a través del bus de datos en esta fase de aceptación de la interrupción?.(responder en hexadecimal).

0007h

– ¿Cuál era dirección de la instrucción que se estaba ejecutando en el programa principal cuando se produce la interrupción?.

0501h

En un computador con un bus de direcciones de 8 líneas, se pretende ubicar un dispositivo de memoria de 8 palabras. El rango del espacio de direcciones elegido para dicho dispositivo es 50h-57h ¿Cuál es el circuito de activación del dispositivo de memoria, de modo que éste quede ubicado en el rango de direcciones indicado? Deben dibujarse todas las líneas que llegan a la puerta AND, indicar su peso y si entran negadas o no a la puerta.