

Examen práctico Fundamentos de Computadores – Gijón. Curso 2009-2010

Apellidos y nombre: _____ DNI: _____ Fecha: _____ Grupo: ____

Instrucciones generales

- Desde el explorador de Windows, comprueba que esté vacío el contenido del directorio **C:\Temp** y, si no lo está, bórralo.
- Crea una carpeta con tus apellidos, nombre y DNI. Por ejemplo: **Diaz Diaz José – 10999888J**. Guarda en esa carpeta todos los archivos que hagas.
- Tras terminar el ejercicio, y cuando creas que todo está bien, desde el Explorador de Windows pulsa con el botón derecho sobre la carpeta y escoge la opción del programa PowerArchiver “**Compress To ‘TuCarpeta.zip’**” para comprimirla.
- Dale a **OK**, con lo que se generará un fichero **.zip** que deberás copiar a la carpeta **\\lear\exámenes\Estructura de Computadores\PLxx**. Una vez que escribas en esa carpeta no podrás cambiar, leer ni volver a escribir el archivo, así que asegúrate de que está correcto **antes** de copiarlo.
- Borra el directorio temporal antes de cerrar la sesión.

Ejercicio

- Completar el esqueleto del programa incluido en el fichero **eval.ens**, **rellenando los huecos marcados con interrogantes**. Los interrogantes se corresponden con una instrucción (o con un trozo de una instrucción incompleta). Lee atentamente los comentarios incluidos en el fichero para comprender el funcionamiento global del programa

- ¿Cuál es la dirección de mapeo de la interfaz Teclado y el vector de interrupción utilizado? ¿En qué dirección estará mapeada la posición de la memoria de vídeo de la interfaz Pantalla donde ha de empezar a escribirse? (Responder en hexadecimal).

Dirección de mapeo:

Vector de interrupción:

Dirección de la primera posición donde escribir en la memoria de vídeo:

- Una vez hayas obtenido el fichero ejecutable, carga el programa en la CPU elemental, **configura y mapea los periféricos necesarios de forma adecuada, llenando el espacio de direcciones del computador con tantos módulos de memoria como sea posible**. Ejecuta el programa y comprueba que funciona correctamente. Guarda el estado de la CPU elemental con el nombre **eval.sim** en un momento cualquiera de la ejecución del programa.
- Antes de entregar el examen comprueba que en el archivo comprimido tienes al menos los ficheros **eval.ens**, **eval.eje** y **eval.sim**.

Examen práctico Fundamentos de Computadores – Gijón. Curso 2009-2010

Apellidos y nombre: _____ DNI: _____ Fecha: _____ Grupo: _____

```
; Este programa utiliza 2 periféricos: Teclado y
; Pantalla. El programa principal
; lleva a cabo las siguientes acciones:
; 1- Instala la rutina de interrupcion en
; memoria y habilita el tratamiento de
; interrupciones.
; 2- Inicializar la variable posicionPantalla
; 3- Ejecuta un bucle infinito
; Cuando se genere una interrupcion desde el
; periférico Teclado, la rutina de tratamiento
; asociada al periférico debera escribir el
; carácter en la pantalla en la posicion
; indicada por la variable posicionPantalla,
; actualizando esta variable a la siguiente
; posicion que corresponda.
; Se escribirán caracteres en la DIAGONAL de
; la pantalla, empezando por la ESQUINA SUPERIOR
; IZQUIERDA y bajando en la diagonal. Después de
; escribir en la ESQUINA INFERIOR DERECHA, debe
; llamarse al procedimiento que borra la
; pantalla y comenzar de nuevo en la esquina
; superior izquierda.
```

```
ORIGEN 0E000h
INICIO primera
.PILA 15h
```

```
.DATOS
posicionPantalla VALOR 0FF00h
```

```
.CODIGO
PROCEDIMIENTO borra_pantalla
????
????
????
```

```
; Carga registro de control de la pantalla
????
????
MOV R1, [R0]
```

```
; Carga la mascara y se la aplica al registro
; de control
MOVL R2, 1
MOVH R2, 0
OR R1, R1, R2
```

```
; Actualiza registro de control
MOV [R0], R1
```

```
POP R2
POP R1
POP R0
????
FINP
```

```
PROCEDIMIENTO RutinaTeclado
PUSH R0
PUSH R1
PUSH R2
PUSH R3
```

```
; Cargar el caracter escrito en teclado
MOVL R0, 0C0h
MOVH R0, 0FFh
MOV R0, [R0]
```

```
; Aplica el atributo: color fondo blanco,
; color caracter azul
MOVH R0, 39H
```

```
; Cargar la posicion de pantalla donde se va a
; escribir
MOVL R1, BYTEBAJO DIRECCION posicionPantalla
MOVH R1, BYTEALTO DIRECCION posicionPantalla
MOV R2, [R1]
```

```
; Cargar la posicion equivalente a salirse de
; la pantalla por abajo
MOVL R3, 136
MOVH R3, 0FFh
```

```
; Comprobar si la posicion actual esta dentro
; de los limites de la
; pantalla; si no es asi, borrar la pantalla y
; actualizar a la posicion de la esquina
; superior izquierda de la pantalla
```

```
COMP R2, R3
BRNZ escribe_caracter
CALL borra_pantalla
MOVL R2, 0
MOVH R2, 0FFh
```

```
escribe_caracter:
; Escribir el caracter en pantalla
MOV [R2], R0
```

```
; Avanza a la siguiente posicion de pantalla a
; escribir
MOVL R3, 17
MOVH R3, 0
ADD R2, R2, R3
MOV [R1], R2
```

```
????
????
????
????
????
FINP
```

```
primera:
; Instalar rutina de teclado en memoria
MOVL R0, 4
MOVH R0, 0
MOVH R1, BYTEALTO DIRECCION RutinaTeclado
MOVL R1, BYTEBAJO DIRECCION RutinaTeclado
MOV [R0], R1
```

```
; Inicializar la posicion en pantalla
MOVL R0, ??
MOVH R0, ??
MOVL R1, ??
MOVH R1, ??
????
```

```
; Activar las interrupciones
STI
```

```
bucle:
JMP bucle
FIN
```