

## Sistemas de E/S

### Ejercicios resueltos

**Ejercicio 1.** Se desea desarrollar un controlador para un horno microondas. El controlador dispone de un procesador de 32 bits, mapa de E/S común y juego de instrucciones del MIPS 32. A este procesador se le conectan un módulo de E/S, que controla el funcionamiento del horno. El módulo dispone de los cinco siguientes registros de 32 bits:

- Registro con dirección 1000. En este registro se carga el valor correspondiente a la cuenta atrás en segundos.
- Registro con dirección 1004. En este registro se carga un 1 cuando se quiere comenzar la cuenta atrás. Mientras haya un 0 no se inicia la cuenta atrás.
- Registro con dirección 1008. Cuando la cuenta atrás llega a 0, en este registro el controlador almacena un 1. Mientras se está realizando la cuenta atrás el valor que se encuentra almacenado en este registro es 0.
- Registro con dirección 1012. En este registro se carga el valor correspondiente a la potencia deseada. El valor es un entero de 32 bits en complemento a dos.
- Registro con dirección 1016. Cuando se carga en este registro el valor 1, el microondas se pone en funcionamiento con la potencia almacenada en el registro anterior. Cuando en este registro se carga un 0, el microondas se para independientemente de que se haya llegado al final de la cuenta atrás.

Se desea escribir una función en ensamblador, denominada `Controlador_Horno` que permita controlar el funcionamiento del horno. Esta función recibirá dos parámetros de entrada: un valor entero indicando el número de segundos que ha de permanecer encendido el horno y un valor entero que representa la potencia del horno. La función se encargará de programar el controlador anterior con la potencia y segundos pasados como parámetros, encargándose de poner en funcionamiento el horno y apagarlo cuando haya transcurrido el tiempo necesario. La función devolverá los siguientes posibles valores:

- Un valor igual a 0 que representa que no ha habido ningún error y que se devolverá cuando la función haya parado el horno.
- Un valor igual a -1, cuando la potencia pasada como parámetro sea menor de 100 y mayor de 1000. En este caso no se ha de programar el controlador del horno y la función debe retornar inmediatamente el valor -1.

Se pide:

- Indique en qué registros se han de pasar los parámetros y en qué registro se debe devolver el resultado.
- Invoque a la función anterior para un valor de potencia de 800 W y una duración de 90 segundos. Escriba a continuación el código que permita imprimir por pantalla el resultado que devuelve la función.
- Escriba el código correspondiente a la función `Controlador_Horno`.

**Solución:**

```
controlador_Horno:
    li    $t0, 100
    blt   $a0, $t0, fin_error
    li    $t0, 1000
    bgt   $a0, $t0, fin_error
    sw    $a1, 1000        ; cuenta atrás
    sw    $a0, 1012        ; potencia
    li    $t0, 1
    sw    $t0, 1016        ; arranque motor
    sw    $t0, 1004        ; iniciar cuenta atrás
bucle:   lw    $t1, 1008    ; leer estado cuenta
    beq   $t1, $0, bucle
    sw    $0, 1016        ; parar el motor
    move  $v0, $0
    jr    $ra
fin_error: li $v0, -1
           jr $ra
```

**Ejercicio 2.** Se desea desarrollar un controlador para un dispositivo que permite medir la tensión arterial de una persona. El módulo de E/S del medidor de tensión se conecta a un computador que dispone de un procesador de 32 bits, mapa de E/S común y juego de instrucciones del MIPS 32. El módulo dispone de los siguientes registros de 32 bits:

- Registro con dirección 1000. Cuando se carga en este registro el valor 1, se le indica al módulo que se quiere comenzar una medida de la tensión.
- Registro con dirección 1004. Cuando la medición ha acabado en este registro se almacena un 1, mientras el dispositivo se encuentra realizando la medición, su valor es 0.
- Registro con dirección 1008. Cuando el dispositivo finaliza la medición de la tensión, en este registro se almacena el valor correspondiente a la tensión sistólica.
- Registro con dirección 1012. Cuando el dispositivo finaliza la medición de la tensión, en este registro se almacena el valor correspondiente a la tensión diastólica.

Escriba una función en ensamblador, denominada `Controlador_tension` que permita controlar el funcionamiento de este equipo. Esta función no recibirá ningún parámetro de entrada. La función se encargará de realizar una medición de la tensión y devolverá como parámetros de salida el valor de la tensión sistólica y el valor de la tensión diastólica.

#### Solución:

```
ControladorTension:    li    $t0, 1
                      sw    $t0, 1000

                      Bucle: lw    $t0, 1004
                      beqz   $t0, bucle

                      lw    $v0, 1008
                      lw    $v1, 1012

                      jr     $ra
```

**Ejercicio 3.** Se desea adquirir un disco duro para un equipo doméstico que dé la mejor relación calidad/precio. Dado que el disco se va a utilizar de manera intensiva para la grabación y reproducción de películas, se considera más importante el tiempo medio de acceso y la capacidad de almacenamiento que el coste por MB del disco.

Se dispone de dos posibles discos candidatos con los siguientes parámetros:

Criterio	Disco A	Disco B
Platos	12 platos/disco	24 platos/disco
Superficies por plato	2 superficie/plato	2 superficie/plato
Pistas	10000 pista/cilindro	50000 pista/cilindro
Tamaño Pista	64 sector/pista	64 sector/pista
Tamaño Sector	512 bytes/sector	512 bytes/sector
Velocidad Rotación	5400 r.p.m.	7200 r.p.m.
Tiempo (Disco) es proporcional a la distancia	0,001 seg/pista	0,003 seg/pista
Coste (precio final)	24,26 Euros	33,49 Euros

El disco A está dotado de una tecnología más antigua que el disco B, como se puede apreciar por la velocidad de rotación del disco.

Se pide:

- Calcular la capacidad de cada disco si se formatea con cuatro zonas de igual tamaño con densidades de 64, 128, 256 y 512 sectores por pista.
- Calcular el tiempo medio de acceso de cada disco con dicho formato por zonas.
- Calcular el coste por MB de cada disco y justifique, en función de todos los datos calculados, cuál es el mejor disco para comprar.

#### Solución:

- Calcular la capacidad de cada disco si se formatea con cuatro zonas de igual tamaño con densidades de 64, 128, 256 y 512 sectores por pista.

Hay que tener en cuenta que la fórmula a utilizar es:

$$\text{Nº Platos} * \text{Nº Superficies} * \left( \frac{\text{Nº Pistas}}{4} \right) * (64 * \text{Tam(Sector)} + 128 * \text{Tam(Sector)} + 256 * \text{Tam(Sector)} + 512 * \text{Tam(Sector)})$$

Apdo. b:	C <sub>busca</sub> (Disco A)		C <sub>busca</sub> (Disco B)	
C <sub>busca</sub> (l) en bytes	254.912.000.000,00	bytes	254.912.000.000,00	bytes
C <sub>busca</sub> (l) en KB	254.900.000,00	KB	254.900.000,00	KB
C <sub>busca</sub> (l) en MB	254.900,00	MB	254.900,00	MB
C <sub>busca</sub> (l) en GB	254,90	GB	254,90	GB

b) Calcular el tiempo medio de acceso de cada disco con dicho formato por zonas.

Apdo. b: Cálculos previos		
T <sub>busqueda</sub> (l)	T <sub>busqueda</sub> (Disco A)	T <sub>busqueda</sub> (Disco B)
Peor Caso: de la pista 0 a la pista 99999	100000 pistas	5000 pistas
Mejor Caso: se está en la pista	0 pistas	0 pistas
Caso medio: se está a la mitad de pistas entre ambas	50000 pistas	2500 pistas
T <sub>busqueda promedio</sub> (Disco A)=Sumatorio (Caso*su tiempo)/3casos	50 segundos	7,5 segundos
T <sub>busqueda promedio</sub> (Disco A)=Sumatorio (Caso*su tiempo)/3casos	50000 mseg.	7500 mseg.
T <sub>rotacion</sub> (l)	T <sub>rotacion</sub> (Disco A)	T <sub>rotacion</sub> (Disco B)
Velocidad rotación en segundos(v <sub>rot</sub> )	90 r.p.seg.	120 r.p.seg.
Tiempo una rotación (1/v <sub>rot</sub> ) en segundos	0,011111111 segundos	0,008333333 segundos
Tiempo una rotación (1/v <sub>rot</sub> ) en mseg	11,11111111 mseg.	8,33333333 mseg.
Calculando el T <sub>transferencia</sub>		
T <sub>transferencia</sub> (l) con zonas de igual número de pistas y diferente densidad: 64, 128, 256 y 512 sectores/pista		
T <sub>transferencia</sub> (l) con zona de <b>64 sectores/pista</b>	Disco A	Disco B
Tiempo recorrer todos los sectores de una pista	11,11111111 mseg.	8,33333333 mseg.
Número de sectores por pista	64 sector/pista	64 sector/pista
T <sub>transferencia</sub> (l) promedio con zona de <b>64 sectores/pista</b>	0,173611111 mseg.	0,130208333 mseg.
T <sub>transferencia</sub> (l) con zona de <b>128 sectores/pista</b>	Disco A	Disco B
Tiempo recorrer todos los sectores de una pista	11,11111111 mseg.	8,33333333 mseg.
Número de sectores por pista	128 sector/pista	128 sector/pista
T <sub>transferencia</sub> (l) promedio con zona de <b>128 sectores/pista</b>	0,086805556 mseg.	0,065104167 mseg.
T <sub>transferencia</sub> (l) con zona de <b>256 sectores/pista</b>	Disco A	Disco B
Tiempo recorrer todos los sectores de una pista	11,11111111 mseg.	8,33333333 mseg.
Número de sectores por pista	256 sector/pista	256 sector/pista
T <sub>transferencia</sub> (l) promedio con zona de <b>256 sectores/pista</b>	0,043402778 mseg.	0,032552083 mseg.
T <sub>transferencia</sub> (l) con zona de <b>512 sectores/pista</b>	Disco A	Disco B
Tiempo recorrer todos los sectores de una pista	11,11111111 mseg.	8,33333333 mseg.
Número de sectores por pista	512 sector/pista	512 sector/pista
T <sub>transferencia</sub> (l) promedio con zona de <b>512 sectores/pista</b>	0,021701389 mseg.	0,016276042 mseg.
	T <sub>transferencia</sub> (Disco A)	T <sub>transferencia</sub> (Disco B)
T <sub>transferencia</sub> (l) con zonas (promediando)	0,081380208 mseg.	0,061035156 mseg.
Calculando T <sub>acceso</sub> requerido en los apartados b y d		
T <sub>busqueda</sub> (l)	50000 mseg.	7500 mseg.
T <sub>rotacion</sub> (l)	11,11111111 mseg.	8,33333333 mseg.
T <sub>transferencia</sub> (l) con zonas (promediando)	0,081380208 mseg.	0,061035156 mseg.
T <sub>acceso</sub> (l) con Zonas	50011,19249 mseg.	7508,394368 mseg.

c) Calcular el coste por MB de cada disco y justifique, en función de todos los datos calculados, cuál es el mejor disco para comprar.

El coste por GB de cada disco es:

Calculando el coste por almacenamiento de cada disco	Disco A	Disco B
Tamaño con CAV	73.24 GB	73.24 GB
Tamaño con pistas de tamaño variable	274.66 GB	274.66 GB
Coste por GB con CAV	0.33 Euros/GB	0.46 Euros/GB
Coste por GB con pistas de tamaño variable	0.09 Euros/GB	0.12 Euros/GB

Dado que ambos discos tendrían el mismo tamaño, ya sea como CAV o con distribución variable de sectores por pista, el tamaño del disco no es un dato significativo para la decisión.

Fijándose que el tiempo de acceso del disco A es mayor que el tiempo de acceso del disco B pero el coste por GB es menor en el disco A, se tendrá que tomar la decisión entre qué característica es más deseable: si es la velocidad entonces habrá que adquirir el disco B, si es la economía entonces habrá que adquirir el disco A.

**Ejercicio 4.** Sea un computador de 32 bits de 500 MIPS, con el juego de instrucciones del MIPS32 y un mapa de entrada/salida común, con una frecuencia de reloj de 1 GHz. Este computador dispone de un dispositivo que produce 5000 interrupciones por segundo. Cada interrupción supone la ejecución de una rutina de tratamiento de la interrupción que requiere 2000 instrucciones máquina. Calcule el porcentaje de tiempo que este computador dedica a tratar las interrupciones de este dispositivo.

**Solución:**

El dispositivo interrumpe 5000 veces por segundo, por tanto, el número de instrucciones que se ejecuta por segundo como parte del tratamiento de las interrupciones es de  $2000 \times 5000 = 10 \text{ MIPS}$ . El porcentaje de tiempo dedicado a atender las interrupciones es  $10/500 \times 100 = 2\%$ .

**Ejercicio 5.** Se desea desarrollar un controlador para un semáforo. El controlador dispone de una CPU de 32 bits, mapa de E/S **separado** y juego de instrucciones del MIPS 32. A esta CPU se le conectan dos módulos de E/S. El primero es un cronómetro y el segundo es el módulo de E/S que controla el funcionamiento del semáforo. El módulo cronómetro dispone de los tres registros siguientes:

- Registro con dirección 1000. En este registro se carga el valor correspondiente a la cuenta atrás en segundos.
- Registro con dirección 1004. En este registro se carga un 1 cuando se quiere comenzar la cuenta atrás.
- Registro con dirección 1008. Cuando la cuenta atrás llega a 0, en este registro se carga un 1. Mientras se está realizando la cuenta atrás el valor de este registro es 0.

El módulo de E/S que controla el semáforo dispone de tres registros:

- Registro con dirección 1012. En este registro se codifica el valor correspondiente al color del semáforo: 100 para el rojo, 010 para el amarillo y 001 para el verde.
- Registro con dirección 1016. Es un registro que indica la duración en segundos correspondiente al rojo.
- Registro con dirección 1020. Es un registro que indica la duración correspondiente al amarillo, que ocurre en la transición rojo-verde.
- Registro con dirección 1024. Es un registro que indica la duración en segundos del verde.

Se pide:

- Escriba el programa ensamblador que controla el funcionamiento de este semáforo. El semáforo siempre comienza su funcionamiento en rojo.
- ¿Qué ineficiencia se identifica en el programa anterior? ¿Cómo se podría resolver?

**Solución:**

ROJO:

```
li    $t0, 100
OUT   $t0, 1012
IN     $t0, 1016
OUT   $t0, 1000
IN     $t0, 1
OUT   $t0, 1004
```



```

ESPERA_ROJO:
    IN    $t0,1008
    li    $t1,1
    beq   $t0,$t1,ESPERA_ROJO
AMARILLO:
    li    $t0, 010
    OUT   $t0, 1012
    IN    $t0, 1020
    OUT   $t0, 1000
    IN    $t0,1
    OUT   $t0,1004
ESPERA_AMARILLO:
    IN    $t0,1008
    li    $t1,1
    beq   $t0,$t1,ESPERA_AMARILLO
VERDE:
    li    $t0, 010
    OUT   $t0, 1012
    IN    $t0, 1024
    OUT   $t0, 1000
    IN    $t0,1
    OUT   $t0,1004
ESPERA_VERDE:
    IN    $t0,1008
    li    $t1,1
    beq   $t0,$t1,ESPERA_VERDE
    b     ROJO

```

La principal causa de ineficiencia es consumo de CPU en el tiempo de espera del contador. La solución sería el empleo de E/S mediante interrupciones.

**Ejercicio 6.** Un computador con una frecuencia de reloj de 3 GHz dispone de un disco duro conectado con una velocidad de transferencia máxima de 20 megabits/s. La latencia promedio entre el envío de peticiones al disco y el comienzo de la transferencia es de 10 milisegundos. Las transferencias se realizan mediante entrada/salida programada. La preparación y envío de una orden de lectura requiere 3000 ciclos de reloj. Se Las peticiones de lectura son para bloques de 4 KB. Se desea conocer el tiempo total de ejecución de una operación de entrada/salida.

#### Solución:

El tiempo total  $T_t$  de ejecución de una operación de entrada/salida vendrá dado por el tiempo de preparación  $T_p$ , la latencia hasta el inicio de la transferencia  $T_l$  y el tiempo de transferencia  $T_{tr}$ .

$$T_p = 3000 \text{ ciclos} / 3 \cdot 10^9 \text{ ciclos/s} = 10^{-6} \text{ secs.}$$

$$T_l = 10 \cdot 10^{-3}$$

$$T_{tr} = (4 \cdot 1024 \cdot 8) \text{ bits} / 20 \cdot 10^6 \text{ bits/sec} = 32768 / 2 \cdot 10^7 = 0.0016384 \text{ secs.}$$

**Ejercicio 7.** Sea un computador de 32 bits similar al MIPS R2000 utilizado en la práctica de la asignatura. Dicho computador tiene un mapa único para memoria y entrada/salida. El espacio de direcciones de memoria está en el rango 0x00000000 – 0x7FFFFFFF. El espacio de direcciones para entrada/salida está en el rango 0x80000000 – 0xFFFFFFFF.

A dicho computador se ha conectado una caja registradora de supermercados que tiene un cajón donde se guarda el dinero, una impresora y un teclado. Así mismo también se ha conectado un lector de código de barras.

El módulo de salida de la caja registradora tiene los siguientes registros:

- ESTADO\_TECLADO (dirección 0x90000000): Registro de solo lectura en el que se puede consultar el estado del teclado, que podrá ser ESPERANDO\_TECLA (valor 0) TECLA\_PULSADA (valor 1).
- DATOS\_TECLADO (dirección 0x90000004): Registro en el que se almacena el código de la última tecla pulsada. Los valores posibles son TECLA\_INICIO (1) y TECLA\_TOTAL (2).



- ESTADO\_CAJON (dirección 0x90000008): Registro de solo lectura en el que se puede consultar el estado del cajón, que podrá ser CERRADO (valor 0) o ABIERTO (valor 1).
- ESTADO\_IMPRESORA (dirección 0x9000000C): Registro de solo lectura en el que se puede consultar el estado de la impresora que podrá ser ESPERANDO (valor 0) o IMPRIMIENDO (valor 1).
- CONTROL (dirección 0x90000010): Registro en el que se pueden escribir las órdenes ESPERAR\_TECLA (0), ABRIR\_CAJON (1) o IMPRIMIR (2).
- DATOS\_IMPRESION (direcciones 0x90002000 y siguientes): Conjunto de 14 registros donde se almacenarán los códigos ASCII del código de barras a imprimir (13 primeros registros) y el precio (decimocuarto registro IEEE 754 de simple precisión).

Cuando se desea leer una tecla, se suministra al módulo la orden ESPERAR\_TECLA y el estado del teclado pasa a ser ESPERANDO\_TECLA, hasta que se pulsa una tecla (estado TECLA PULSADA) y su código se almacena en el registro DATOS\_TECLADO. Una vez consultado el código de la tecla, será necesario enviar una nueva orden ESPERAR\_TECLA para iniciar la lectura de una nueva tecla. Si en algún momento se desea abrir el cajón del dinero, se deberá enviar la orden ABRIR\_CAJON con lo que el cajón pasará de CERRADO a ABIERTO. Cuando se cierre manualmente el cajón, su estado volverá directamente a CERRADO. Por último para imprimir un código de barras y su precio, se colocarán dichos datos en el registro DATOS\_IMPRESION y se dará la orden IMPRIMIR para que ESTADO\_IMPRESORA pase de ESPERANDO a IMPRIMIENDO. Cuando se haya impreso la línea se volverá al estado ESPERANDO. Si se desea imprimir un total se indicará el código de barras con los 13 dígitos con código ASCII igual a 0.

El módulo de entrada/salida del lector de código de barras tiene los siguientes registros:

- ESTADO (dirección 0x8000100A): Registro de solo lectura en el que se puede consultar el estado del dispositivo, que podrá ser DESACTIVADO (valor 0), ESPERANDO\_LECTURA (valor 1), LECTURA\_REALIZADA (valor 2).
- CONTROL (dirección 0x8000100B): Registro en el que se pueden escribir las órdenes DESACTIVAR (valor 0), ACTIVAR (valor 1) o LEER (valor 2).
- DATOS (direcciones 0x8000100C y siguientes): Conjunto de 13 registros que contiene los códigos ASCII de los 13 dígitos del último código de barras leído.

Inicialmente el dispositivo se encuentra DESACTIVADO, cuando se le suministra la orden ACTIVAR, el dispositivo pasa al estado ESPERANDO\_LECTURA. Tras realizar una lectura óptica de un código de barras, el dispositivo pasará al estado LECTURA\_REALIZADA y el código de barras leído pasará a estar almacenado en los 13 registros de datos. Cuando se desee realizar una nueva lectura será necesario suministrar la orden LEER, con lo que se volverá al estado ESPERANDO\_LECTURA. Cuando se desee finalizar la utilización del dispositivo se suministrará la orden DESACTIVAR con lo que el dispositivo pasará al estado DESACTIVADO.

Se pide:

- Escriba una subrutina utilizando el ensamblador del MPIS R2000 que comience a leer códigos de barra cuando se pulse la tecla TECLA\_INICIO y vaya realizando lecturas hasta que se pulse la tecla TECLA\_TOTAL. Los códigos se almacenarán en posiciones de memoria consecutivas a partir de la dirección de memoria pasada en el parámetro \$a0. Cada código de barras se comenzará a almacenar en una dirección con alineamiento de palabra y usando a partir de ahí, bytes consecutivos. La subrutina devolverá de \$v0 el número de códigos leídos.
- ¿Qué modelo de entrada/salida implican los dispositivos presentados en el enunciado?
- ¿Qué modelo de entrada/salida sería el más apropiado para el lector de códigos de barra? ¿Por qué?
- Si la caja registradora usase tres módulos de entrada/salida (uno para el cajón, otro para el teclado y otro para la impresora) ¿Qué modelo de entrada/salida seleccionaría para cada uno de ellos?

**Solución:**

a)

```
LEE_CODIGOS:
    move    $t7, $a0

    li      $t0, ESPERAR_TECLA
    sw      $t0, CONTROL
```





```

    li    $t0, ESPERANDO_TECLA

MIENTRAS_NO_PULSADO
    lw    $t1, ESTADO_TECLADO
    beq   $t0, $t1, MIENTRAS_NO_PULSADO

    li    $t0, TECLA_INICIO
    lw    $t1, DATOS_TECLADO
    bne   $t0, $t1, LEE_CODIGOS

    li    $t0, ACTIVAR
    sw    $t0, CONTROL_BARRAS

    li    $t0, ESPERANDO_LECTURA

MIENTRAS_NO_ACTIVADO:
    lw    $t1, ESTADO_BARRAS
    bne   $t0, $t1, MIENTRAS_NO_ACTIVADO

    li    $t0, ESPERAR_TECLA
    sw    $t0, CONTROL

    li    $t0, LECTURA_REALIZADA
    li    $t1, TECLA_PULSADA

MIENTRAS_ESPERA_EVENTO
    lw    $t2, ESTADO_TECLADO
    lw    $t3, ESTADO_BARRAS
    beq   $t3, $t1, TRATAR_CODIGO
    beq   $t2, $t0, TRATAR_TECLA
    b     MIENTRAS_ESPERA_EVENTO

TRATAR_CODIGO:
    la    $t4, DATOS_BARRAS
    li    $t6, 13

BUCLE_BARRAS:
    lb    $t5, ($t4)
    sb    $t5, (t$t7)
    add   $t4, $t4, 1
    add   $t7, $t7, 1
    sub   $t6, $t6, 1
    bnez  $t6, BUCLE_BARRAS
    add   $t7, $t7, 3
    li    $t5, LEER
    sw    $t5, CONTROL_BARRAS
    bne   $t2, $t0, MIENTRAS_ESPERA_EVENTO

TRATAR_TECLA:
    li    $t5, TECLA_TOTAL
    lw    $t6, ESTADO_TECLADO
    beq   $t5, $t6, FINALIZAR
    li    $t5, ESPERAR_TECLA
    sw    $t5, CONTROL
    b     MIENTRAS_ESPERA_EVENTO

FINALIZAR:
    li    $t0, DESACTIVAR
    sw    $t0, CONTROL_BARRAS
    li    $t0, DESACTIVADO

```

```
MIENTRAS_NO_DESACTIVADO:  
    lw    $t1, ESTADO_BARRAS  
    bne   $t0, $t1, MIENTRAS_NO_DESACTIVADO  
    sub   $v0, $t7, $a0  
    div   $v0, $v0, 4  
    jr    $ra
```

b) Entrada/salida programada

- a) Acceso directo a memoria. Sería el que menos ocupación de CPU requeriría, notificando cuando el código estuviese almacenado en memoria principal.
- d) Una opción sería:
- Cajón: Entrada/salida por interrupciones
  - Teclado: Entrada/salida por interrupciones.
  - Impresora: Entrada/salida por acceso directo a memoria.