

Exámenes

(ETC) P12 - Test de sesión (curso 20/21)

[Volver a la Lista de Exámenes](#)



Parte 1 de 1 -

10.0 Puntos

Preguntas 1 de 5

2.0 Puntos

Interfaz de teclado (DB = 0xFFFF0000)

Nombre	Dirección	Acceso	Estructura
Estado/órdenes	DB	L/E	
Datos	DB+4	L	

Registro de órdenes y estado (Lectura/escritura. Dirección = Base).

- R (bit 0, sólo lectura). Indicador de dispositivo preparado: R = 1 cada vez que se pulsa una tecla. Para hacer R = 0 es necesario realizar un acceso de lectura en el registro de datos.
- E: (bit 1, lectura/escritura). Habilitación de la interrupción (mientras E = 1, el valor R = 1 activa la línea de interrupción del dispositivo).

Registro de datos (Sólo lectura. Dirección = Base + 4).

- COD (bits 7...0). Código ASCII de la tecla pulsada. Leer de este registro provoca que R = 0.

Dada la estructura del interfaz de teclado que se muestra en la figura, indíquese la secuencia de código correcta para realizar el bucle de sincronización por consulta de estado. Se asume que \$t0 se halla inicializado con la dirección base del interfaz.

• ☐

espera: lb \$t1, 0(\$t0)

beqz \$t1, espera

• ☐

espera: lb \$t1, 0(\$t0)

andi \$t1, \$t1, 2

beqz \$t1, espera

• ☐

espera: lb \$t1, 0(\$t0)

ori \$t1, \$t1, 1

beqz \$t1, espera

• ☒

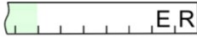

espera: lb \$t1, 0(\$t0)

andi \$t1, \$t1, 1

beqz \$t1, espera

Respuesta correcta: D

Interfaz de teclado (DB = 0xFFFF0000)

Nombre	Dirección	Acceso	Estructura
Estado/órdenes	DB	L/E	
Datos	DB+4	L	

Registro de órdenes y estado (Lectura/escritura. Dirección = Base).

- R (bit 0, sólo lectura). Indicador de dispositivo preparado: R = 1 cada vez que se pulsa una tecla. Para hacer R = 0 es necesario realizar un acceso de lectura en el registro de datos.
- E: (bit 1, lectura/escritura). Habilidad de la interrupción (mientras E = 1, el valor R = 1 activa la línea de interrupción del dispositivo).

Registro de datos (Sólo lectura. Dirección = Base + 4).

- COD (bits 7...0). Código ASCII de la tecla pulsada. Leer de este registro provoca que R = 0.

Dada la estructura del interfaz de teclado que se muestra en la figura, indíquese cómo se realizaría la cancelación del bit R (R=0). Se asume que \$t0 se halla inicializado con la dirección base del interfaz.

- ☐

```
li $t1,1
sb $t1, 0($t0)
```
- ☐

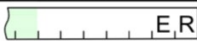

```
li $t1, 2
sb $t1, 0($t0)
```
- ☒

```
lb $a0, 4($t0)
```
- ☐

```
sb $zero, 0($t0)
```

Respuesta correcta: C

Interfaz de teclado (DB = 0xFFFF0000)

Nombre	Dirección	Acceso	Estructura
Estado/órdenes	DB	L/E	
Datos	DB+4	L	

Registro de órdenes y estado (Lectura/escritura. Dirección = Base).

- R (bit 0, sólo lectura). Indicador de dispositivo preparado: R = 1 cada vez que se pulsa una tecla. Para hacer R = 0 es necesario realizar un acceso de lectura en el registro de datos.
- E: (bit 1, lectura/escritura). Habilitación de la interrupción (mientras E = 1, el valor R = 1 activa la línea de interrupción del dispositivo).

Registro de datos (Sólo lectura. Dirección = Base + 4).

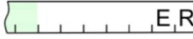
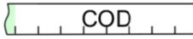
- COD (bits 7...0). Código ASCII de la tecla pulsada. Leer de este registro provoca que R = 0.

Dada la estructura del interfaz de teclado que se muestra en la figura, indíquese cuándo se realizaría la lectura del puerto de Datos de dicho interfaz.

- ☐ A la salida del bucle de sincronización y siempre inmediatamente después de cancelar R
- ☐ A la salida del bucle de sincronización y siempre antes de cancelar R
- ☐ Antes del bucle de sincronización
- ☒ Después del bucle de sincronización

Respuesta correcta: D

Interfaz de teclado (DB = 0xFFFF0000)

Nombre	Dirección	Acceso	Estructura
Estado/órdenes	DB	L/E	
Datos	DB+4	L	

Registro de órdenes y estado (Lectura/escritura. Dirección = Base).

- R (bit 0, sólo lectura). Indicador de dispositivo preparado: R = 1 cada vez que se pulsa una tecla. Para hacer R = 0 es necesario realizar un acceso de lectura en el registro de datos.
- E: (bit 1, lectura/escritura). Habilitación de la interrupción (mientras E = 1, el valor R = 1 activa la línea de interrupción del dispositivo).

Registro de datos (Sólo lectura. Dirección = Base + 4).

- COD (bits 7...0). Código ASCII de la tecla pulsada. Leer de este registro provoca que R = 0.

Dada la estructura del interfaz de teclado que se muestra en la figura, complétese el código para implementar la función

char getchar() - obtiene sobre el registro \$v0 el carácter introducido a través de la interfaz de teclado

```
li $t0, 0x ☒ ffff0000
```

```
espget: ☒ lb $t1, 0($t0)
```

```
☒ andi $t1, $t1, 0x ☒ 1
```

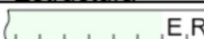

```
☒ beqz $t1, espget
```

```
☒ lb $ ☒ v0 , 0x ☒ 4 ($t0)
```

```
jr $ra
```

Respuesta correcta: ffff0000, lb, andi|and, 1|01, beqz|beq, lb, v0|vo, 4|04

Interfaz de consola (DB = 0xFFFF0008)

Nombre	Dirección	Acceso	Estructura
Estado/órdenes	DB	L/E	
Datos	DB+4	E	

Registro de órdenes y estado (Lectura/escritura. Dirección = Base).

- R (bit 0, sólo lectura). Indicador de dispositivo preparado: R = 1 cuando la consola está disponible. R = 0 cuando se escribe en el registro de datos.
- E: (bit 1, lectura/escritura). Habilitación de la interrupción (mientras E = 1, el valor R = 1 activa la línea de interrupción del dispositivo).

R = 1 activa la línea de interrupción del dispositivo.

Registro de datos (Sólo escritura. Dirección = Base + 4).

- COD (bits 7...0). Código ASCII de del carácter que se ha de escribir en la consola. Escribir en este registro provoca que R = 0 mientras se procesa la operación.

Dada la estructura del interfaz de consuela que se muestra en la figura, indíquese la secuencia correcta de código para implementar la función:

void putchar(char \$a0) - imprime un carácter a través de la interfaz de la consola

- ☐

```

li $t0, 0xffff0008
sb $a0, 4($t0)
espput:  lb $t1, 0($t0)
        andi $t1, $t1, 2
        beq $t1, $zero, espput
        jr $ra

```
- ☐

```

li $t0, 0xffff0008
li $v0, 11
syscall          # print_char
espput:  lb $t1, 0($t0)
        andi $t1, $t1, 1
        beq $t1, $zero, espput
        jr $ra

```
- ☒

```

li $t0, 0xffff0008
espput:  lb $t1, 0($t0)
        andi $t1, $t1, 1
        beq $t1, $zero, espput
        sb $a0, 4($t0)
        jr $ra

```
- ☐

```
li $t0, 0xffff0008
espput: lb $t1, 0($t0)
        andi $t1, $t1, 1
        beq $t1, $zero, espput
li $v0, 11
syscall      # print_char
jr $ra
```

Respuesta correcta: C

- [PoliformaT](#)
- [UPV](#)
- [Powered by Sakai](#)
- Copyright 2003-2021 The Sakai Foundation. All rights reserved. Portions of Sakai are copyrighted by other parties as described in the Acknowledgments screen.