

Y un registro de datos (Sólo lectura. Dirección = Base + 1).

andi \$t0, \$t0, 0x01

1bu \$t0, 1(\$t1)

B

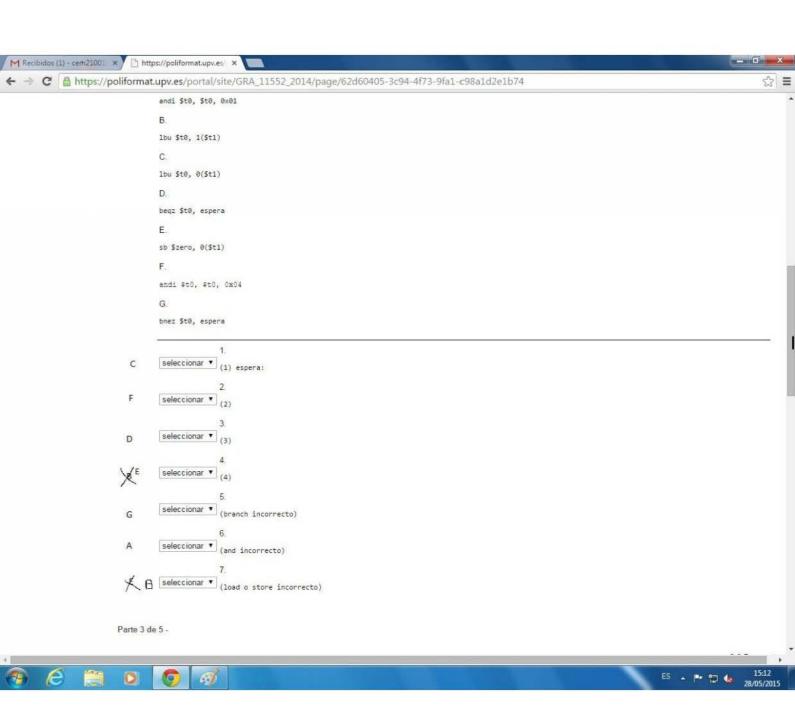
• COD (bits 7...0). Côdigo ASCII de la tecla pulsada. Leer de este registro provoca que R = 0.

borre el bit R. Asumir que en \$11 hemos cargado la dirección base.

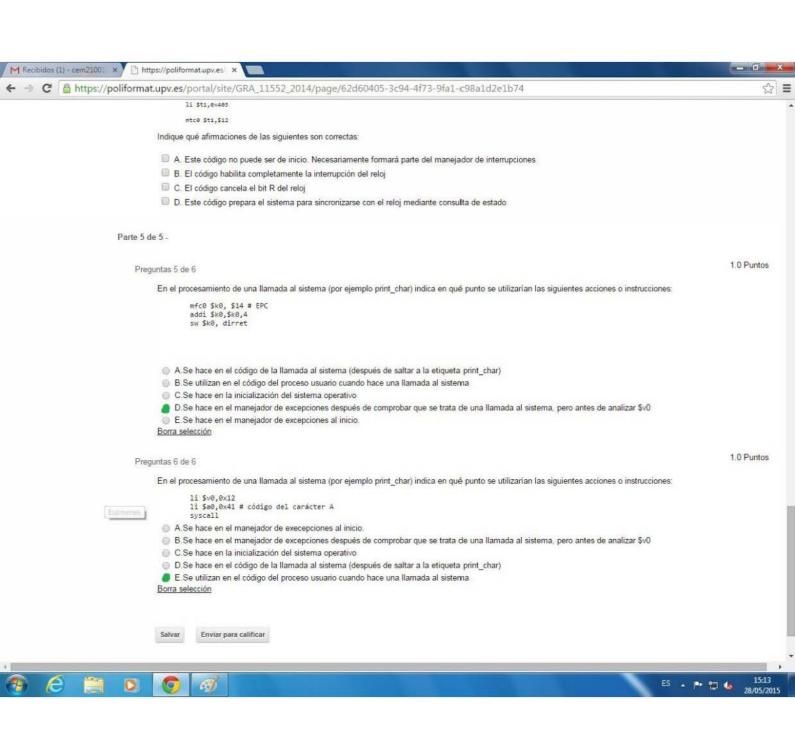
NOTA: el programa sólo tiene 4 instrucciones, hay 3 que no son correctas (un branch, un and y una load o store)

• E: (bit 0, lectura/escritura). Habilitación de la interrupción (mientras E = 1, el valor R = 1 activa la linea de interrupción del dispositivo). Línea INT3*.

Seleccionar las instrucciones que usarías indicando el orden para realizar un bucle de consulta de estado que espere hasta que se haya pulsado una tecla y posteriormente



M Recibidos (1) - c	tem21003 × https://poliformat.upv.es/ ×	_
← → C 🖁	https://poliformat.upv.es/portal/site/GRA_11552_2014/page/62d60405-3c94-4f73-9fa1-c98a1d2e1b74	☆ :
	Preguntas 3 de 6	2.0 Puntos
	Completad las siguientes instrucciones de un fragmento de código de inicialización para que habilite las interrupciones generales y las línea	a INT4 y active el modo usuario:
	li sto, ox 1003	
	MTC0 \$t0, \$12	
	Completad las siguientes instrucciones de un fragmento de código para que habilite las interrupciones generales, habilite la linea INT2, de resto del registro de estado sin modificar.	shabilite la INT1 e INT3 y deje el
	mfc0 \$t0, \$12	
	ori \$t0, \$t0, 0x 0801 0401	
	andi \$t0, \$t0, 0x F9FF F5FF	
	mtc0	
	Parte 4 de 5 - Preguntas 4 de 6	2.0 Puntos
	La interfaz del reloj de PCSpim es la siguiente:	
	Dirección Base=FFFF0010, con un único registro de ôrdenes y estado. Interrumpe por la línea Int2	
	Registro de órdenes y estado (Lectura/escritura)	
	 R (bit 1, lectura/escritura). Indicador de dispositivo preparado: R = 1 a cada segundo. E (bit 0, lectura/escritura). Habilitación de la interrupción (mientras E = 1, el valor R = 1 activa la línea de interrupción del dispositivo) 	
	Considere el siguiente código de inicio del sistema:	
	la \$t0,0xFFFF0010	
	li \$t1,3	
	sb \$t1,e(\$te)	
	li \$t1,0x403	
	ntce \$t1,\$12 Indique qué afirmaciones de las siguientes son correctas:	
	A. Este código no puede ser de inicio. Necesariamente formará parte del manejador de interrupciones	
	B. El código habilita completamente la interrupción del reloj C. El código cancela el bit R del reloj	
	D. Este código prepara el sistema para sincronizarse con el reloj mediante consulta de estado	
		ES A P 15:12



```
Parte 1 de 5 -
    Preguntas 1 de 6
          Tenemos un interfaz de consola en la dirección 0xFFFF0100 con los siguientes registros:
          Registro de órdenes y estado (Lectura/escritura. Dirección = Base).
                 • R (bit 2, lectura/escritura). Indicador de dispositivo preparado: el hardware hace R = 1 cuando la consola está disponible. Para cancelar, el programa ha de escribir
                 R=0

    E: (bit 7, lectura/escritura). Habilitación de la interrupción (mientras E = 1, el valor R = 1 activa la línea de interrupción del dispositivo, línea INT1*).

          Registro de datos (Sólo escritura. Dirección = Base + 1).
          Ambos registros son de 8 bits.
          El siguiente programa escribe el carácter de la posición de memoria "Letra" en la consola, sincronizando por consulta de estado (la interrupción estará deshabilitada).
          Completad las instrucciones que hay detrás del bucle de consulta de estado (que no se muestra) que primero deben cancelar el bit R y luego escribir el carácter en la
          consola
                    .data 0x1000000
          Letra: .ascii 'p'
                    la $t0, 0xFFFF0100
               bucle de consulta de estado
                                                         0($t0)
                         $zero
                                 letra
                    lbu $t1,
                                4(St0)
```

2.0 Puntos

Tenemos una interfaz de reloj en DB=0xFFFF0010 que tiene un único registro de Estado/Ordenes accesible para lectura y escritura. Su contenido es :

- R (bit 2, lectura/escritura). Indicador de dispositivo preparado: R = 1 a cada segundo.
 - o Cancelación: se escribirá un 0 el en bit R .
- E: (bit 0, lectura/escritura). Habilitación de la interrupción (mientras E = 1, el valor R = 1 activa la línea de interrupción del dispositivo)

El dispositivo está conectado a la línea de interrupción INT2*.

1b \$t1,0(\$t0)

Ordena las instrucciones para que el código realice un bucle de consulta de estado que espere hasta que haya pasado un segundo para (1) cancelar el bit R y (2) incrementar la variable tiempo de tipo word. Asumir que en \$10 está cargada la dirección base de la interfaz.

```
A.
la $t0, tiempo
B.
                                                         H,B,D,G,A,F,C,E,
andi $t1,$t1,4
C.
addiu $t1,$t1,1
D.
begz $t1, bucle
E
sw $t1,0($t0)
F.
lw $t1,0($t0)
G.
sb $zero, 0 ($t0)
H.
```

Parte 3 de 5 -

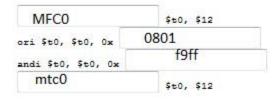
Preguntas 3 de 6

2.0 Puntos

Completar las siguientes instrucciones de un fragmento de código de inicialización que habilite las interrupciones generales, la INT3 y activa el modo usuario:



Completar las siguientes instrucciones de un fragmento de código que habilita las interrupciones generales, la INT1,INT3 y deshabilita la INT2 dejando el resto sin modificar:



Preguntas 4 de 6

La interfaz del reloj de PCSpim es la siguiente:

Dirección Base=FFFF0010, con un único registro de órdenes y estado. Interrumpe por la línea Int2

Registro de órdenes y estado (Lectura/escritura)

- R (bit 1, lectura/escritura). Indicador de dispositivo preparado: R = 1 a cada segundo.
 Cancelación (R = 0): se escribirá en bit R un 0.
- E: (bit 0, lectura/escritura). Habilitación de la interrupción (mientras E = 1, el valor R = 1 activa la línea de interrupción del dispositivo)

Considere el siguiente código de inicio del sistema:

```
la $t0,0xFFFF0010

li $t1,3

sb $t1,0($t0)

li $t1,0x403

mtc0 $t1,$12
```

Indique qué afirmaciones de las siguientes son correctas:

- A. Este código no puede ser de inicio. Necesariamente formará parte del manejador de interrupciones.
- B. Este código prepara el sistema para sincronizarse con el reloj mediante consulta de estado
- C. El código cancela el bit R del reloj
- D. El código habilita completamente la interrupción del reloj

Parte 5 de 5 -

Preguntas 5 de 6

1.0 Puntos

.

En el procesamiento de una llamada al sistema (por ejemplo print_char) indica en qué punto se utilizarían las siguientes acciones o instrucciones:

```
.set noat
sw $at,0($k1)
.set at
```

- A.Se hace en la inicialización del sistema operativo
- B.Se hace en el manejador de excepciones después de comprobar que se trata de una llamada al sistema, pero antes de analizar \$v0
- O C.Se utilizan en el código del proceso usuario cuando hace una llamada al sistema
- D.Se hace en el manejador de excepciones al inicio.
- 💓 E.Se hace en el código de la llamada al sistema (después de saltar a la etiqueta print_char)

Borra selección

En el procesamiento de una llamada al sistema (por ejemplo print_char) indica en qué punto se utilizarían las siguientes acciones o instrucciones:

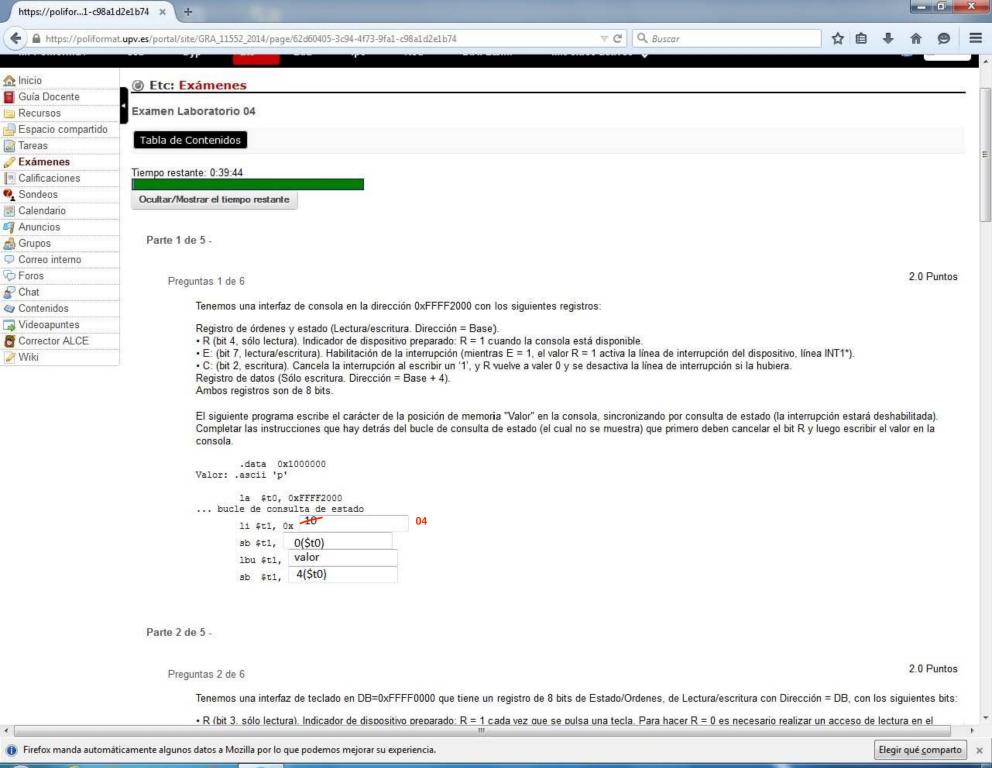
```
li $v0,0x12
li $a0,0x41 # código del carácter A
syscall
```

- A.Se hace en el manejador de excepciones después de comprobar que se trata de una llamada al sistema, pero antes de analizar \$v0
- B.Se hace en el manejador de execepciones al inicio.
- C.Se hace en la inicialización del sistema operativo
- D.Se hace en el código de la llamada al sistema (después de saltar a la etiqueta print_char)
- E.Se utilizan en el código del proceso usuario cuando hace una llamada al sistema

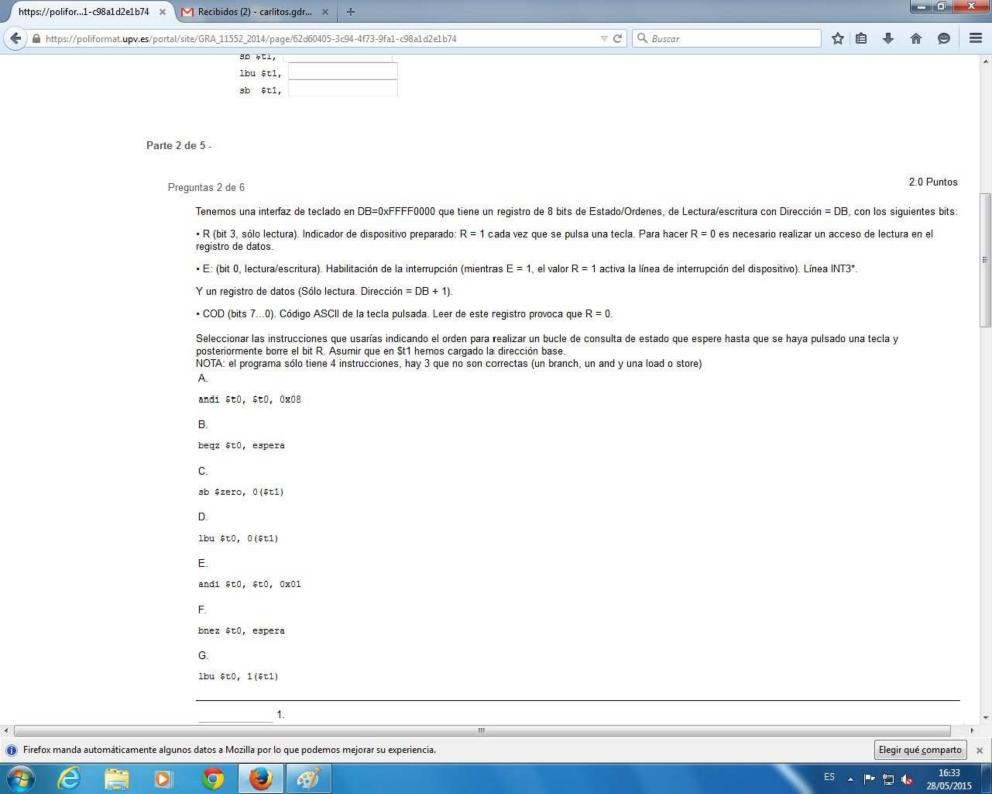
Borra selección

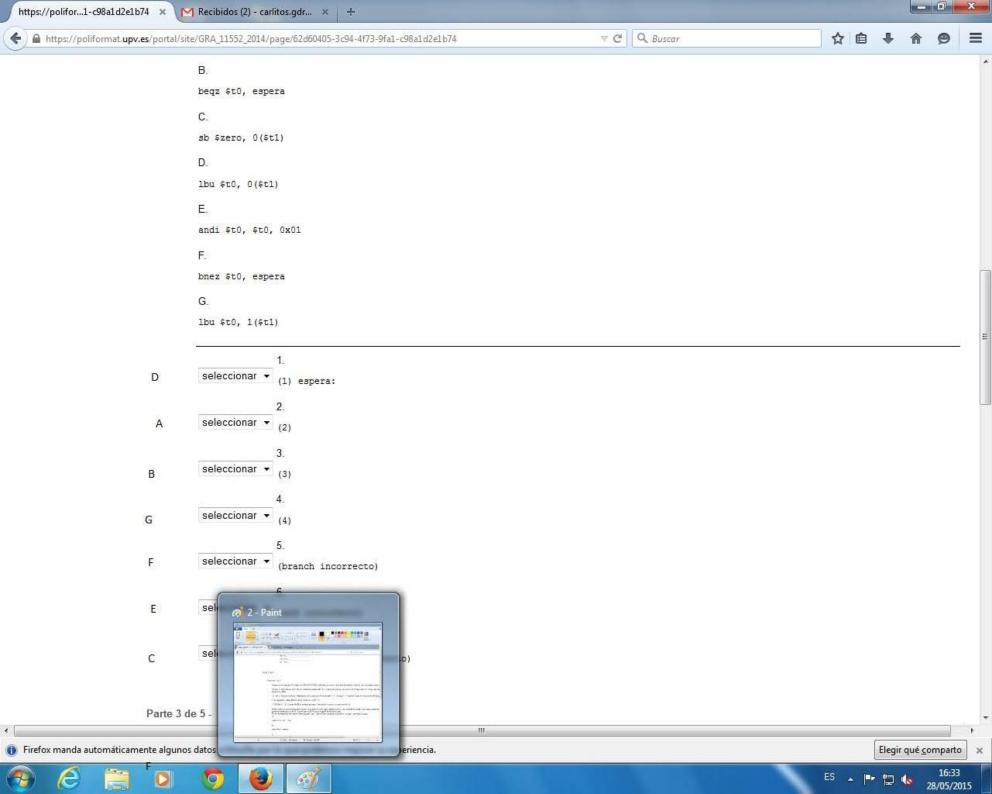
Salvar

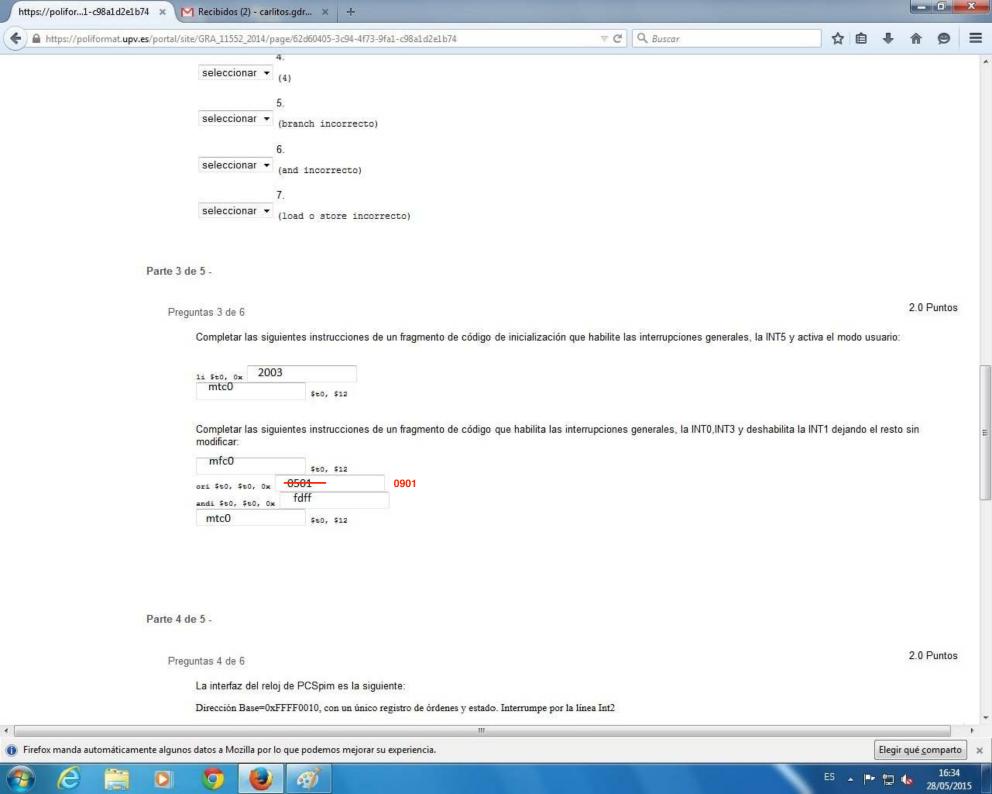
Enviar para calificar

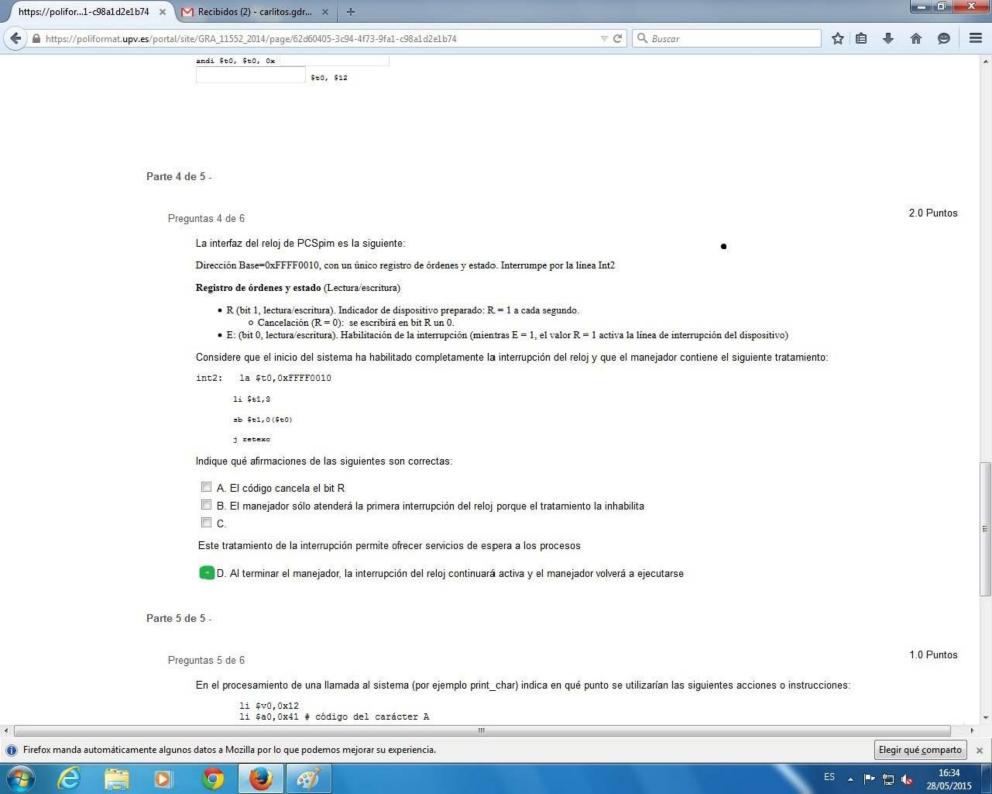


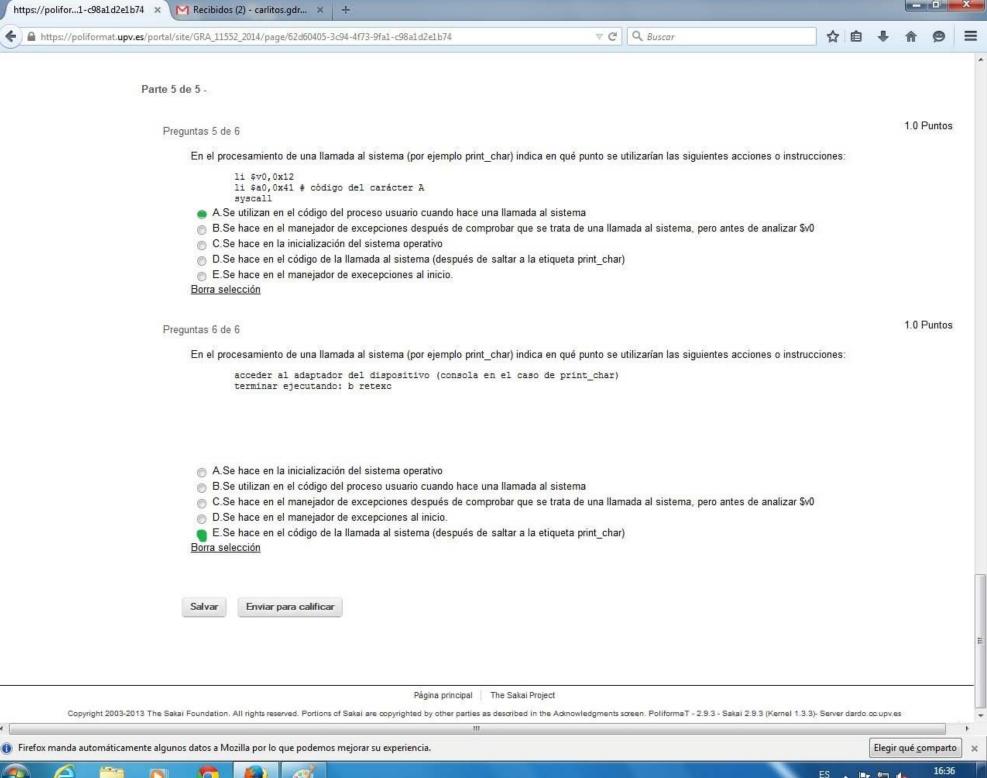
16:30 28/05/2015



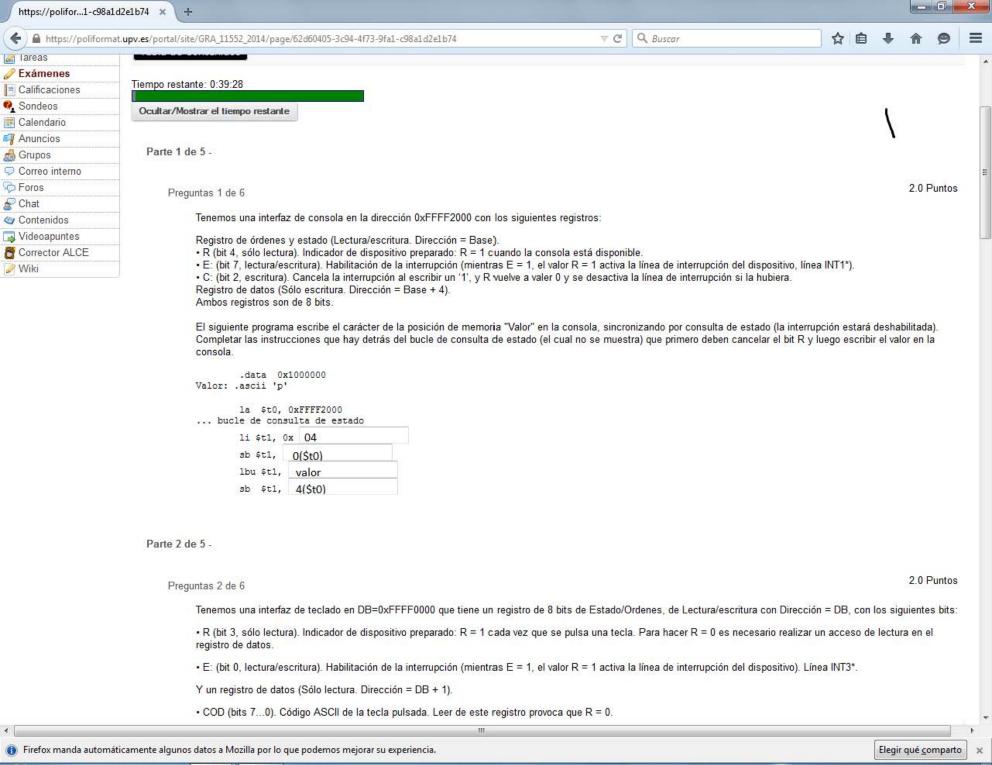








16:36 05/2015



29/05/2015

Parte 2 de 5 -

Preguntas 2 de 6

2.0 Puntos

Tenemos una interfaz de teclado en DB=0xFFFF0000 que tiene un registro de 8 bits de Estado/Ordenes, de Lectura/escritura con Dirección = DB, con los siguientes bits:

- R (bit 3, 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 0, lectura/escritura). Habilitación de la interrupción (mientras E = 1, el valor R = 1 activa la línea de interrupción del dispositivo). Línea INT3*.

Y un registro de datos (Sólo lectura. Dirección = DB + 1).

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

Seleccionar las instrucciones que usarías indicando el orden para realizar un bucle de consulta de estado que espere hasta que se haya pulsado una tecla y posteriormente borre el bit R. Asumir que en \$t1 hemos cargado la dirección base.

NOTA: el programa sólo tiene 4 instrucciones, hay 3 que no son correctas (un branch, un and y una load o store)

A.

sb \$zero, 0(\$t1)

B.

andi \$t0, \$t0, 0x01

C.

andi \$t0, \$t0, 0x08

D.

lbu \$t0, 1(\$t1)

E.,

bnez \$t0, espera

F.

begz \$t0, espera

G.

lbu \$t0, 0(\$t1)

1.

seleccionar -

(1) espera:

2.

Firefox manda automáticamente algunos datos a Mozilla por lo que podemos mejorar su experiencia.

Elegir qué comparto





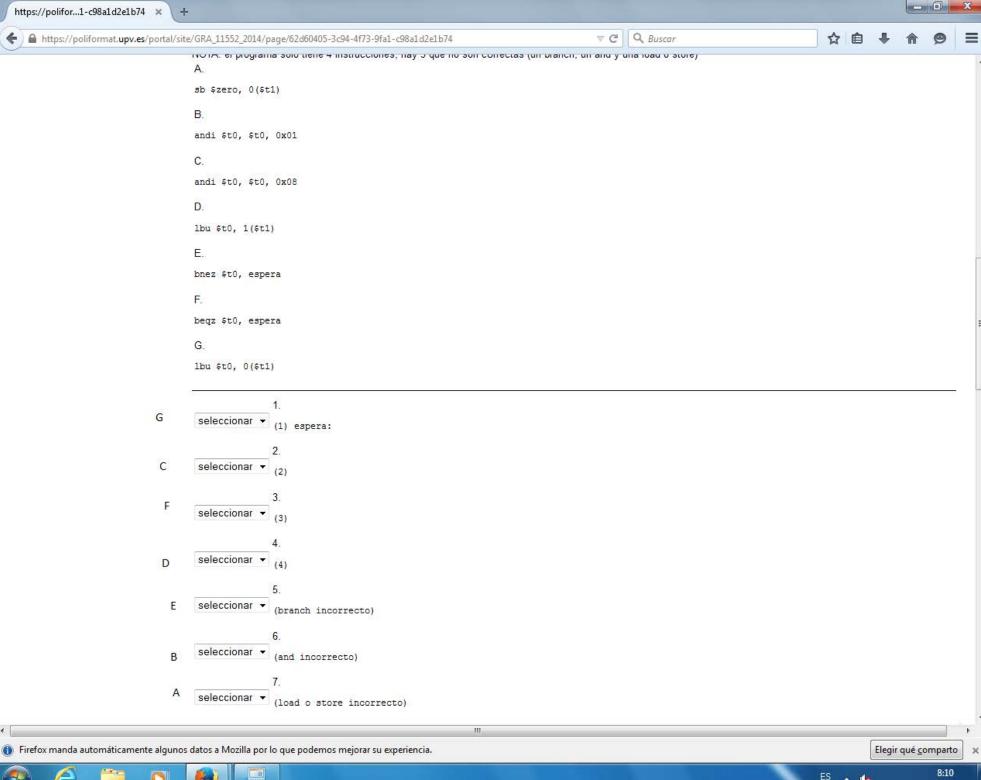




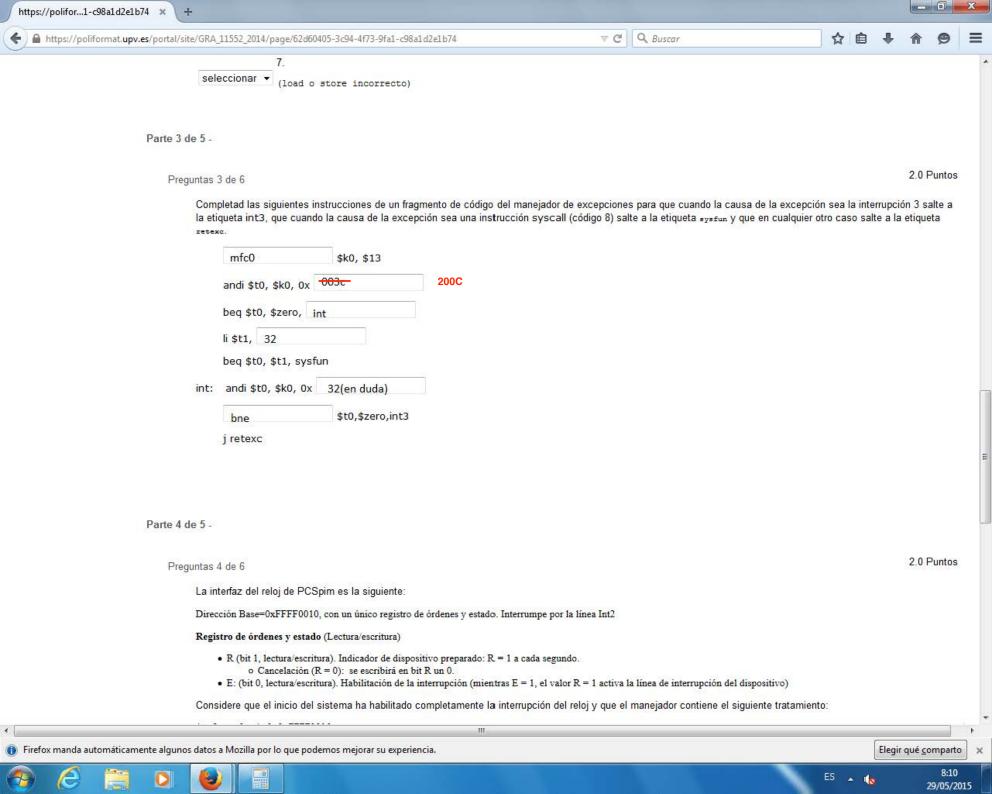


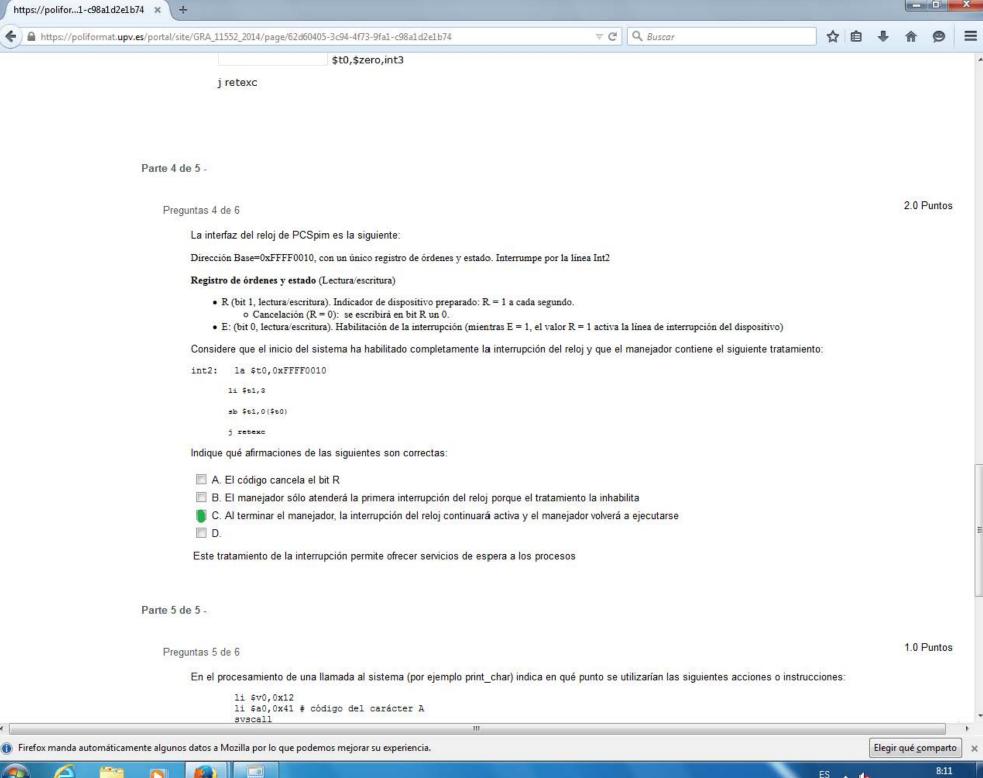




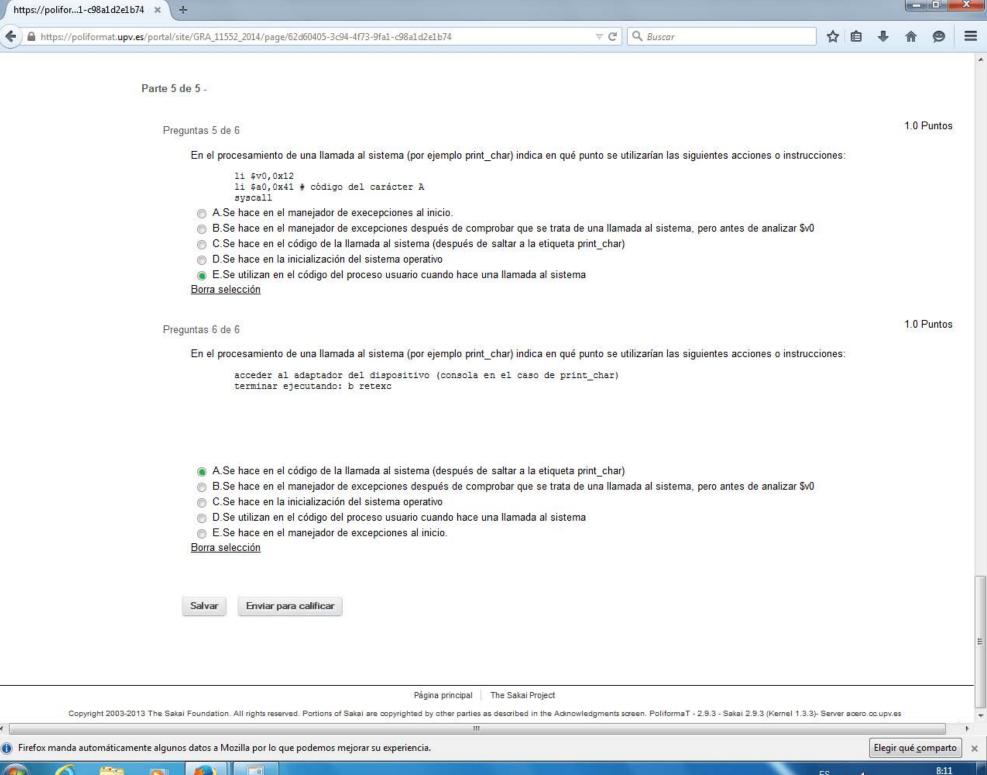


8:10 29/05/2015

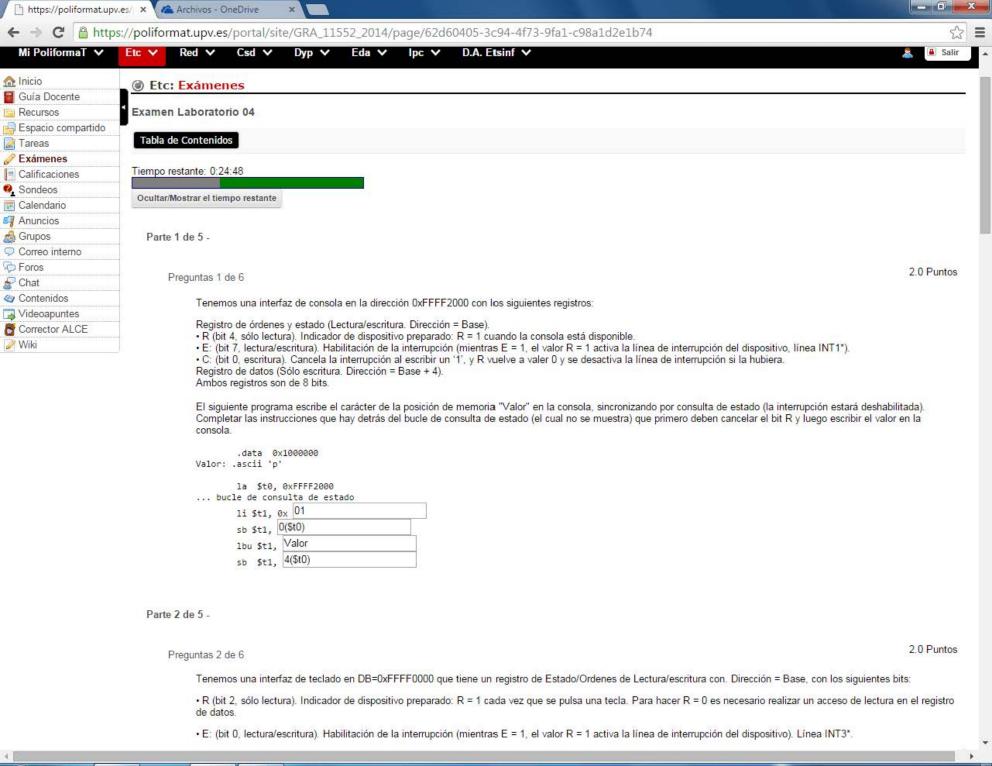




8:11 29/05/2015



29/05/2015



17:18 28/05/2015



2.0 Puntos

Tenemos una interfaz de teclado en DB=0xFFFF0000 que tiene un registro de Estado/Ordenes de Lectura/escritura con. Dirección = Base, con los siguientes bits:

• R (bit 2, 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 0, lectura/escritura). Habilitación de la interrupción (mientras E = 1, el valor R = 1 activa la línea de interrupción del dispositivo). Línea INT3*.

Y un registro de datos (Sólo lectura. Dirección = Base + 1).

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

Seleccionar las instrucciones que usarías indicando el orden para realizar un bucle de consulta de estado que espere hasta que se haya pulsado una tecla y posteriormente borre el bit R. Asumir que en \$11 hemos cargado la dirección base.

NOTA: el programa sólo tiene 4 instrucciones, hay 3 que no son correctas (un branch, un and y una load o store)

A.
andi \$t0, \$t0, 0x04
B.
lbu \$t0, 1(\$t1)
C.
sb \$zero, 0(\$t1)
D.
bnez \$t0, espera

andi \$t0, \$t0, 0x01

beqz \$t0, espera

G.

F.

Preguntas 2 de 6

lbu \$t0, 0(\$t1)

G v (1) espera:

2.
A v (2)
3.
F v (3)
4.
B v (4)
5.
D v (branch incorrecto)
6.
E v (and incorrecto)
7.

1.









C





▼ (load o store incorrecto)

17:18

https://poliformat.upv.es/ × Archivos - OneDrive

17:19 28/05/2015

Página principal | The Sakai Project











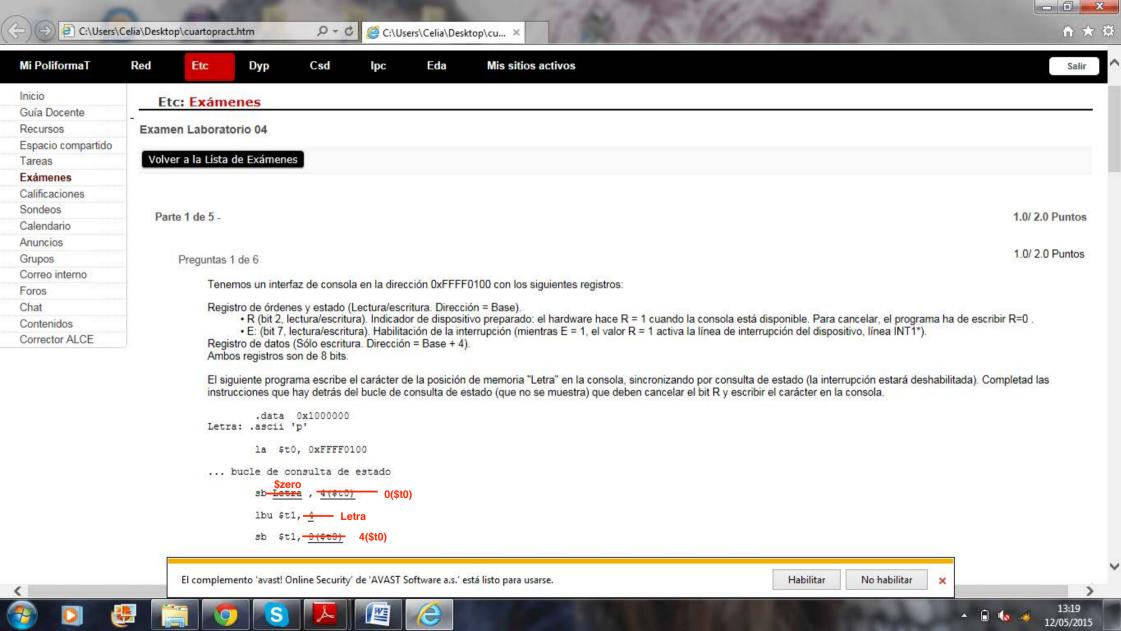




3) li 9to, 0x 2003 mtc0 \$t0,\$12 mgcp \$t0,91

mgcp \$40,912 ions x v, otto, otto into andi sto, sto, ox part कलेटके ब्रेनंग, र्डाय Habilian i. generale, to

sharengo, i atilidad statistication allo association allo



sb \$zero, 0 (\$t0) B. sw \$t1,0(\$t0) C. addui \$t1,\$t1,1 D. andi \$t1,\$t1,2 la \$t0, tiempo F lw \$t1,0(\$t0) G.













lb \$t1,0(\$t0)

beqz \$t1, bucle

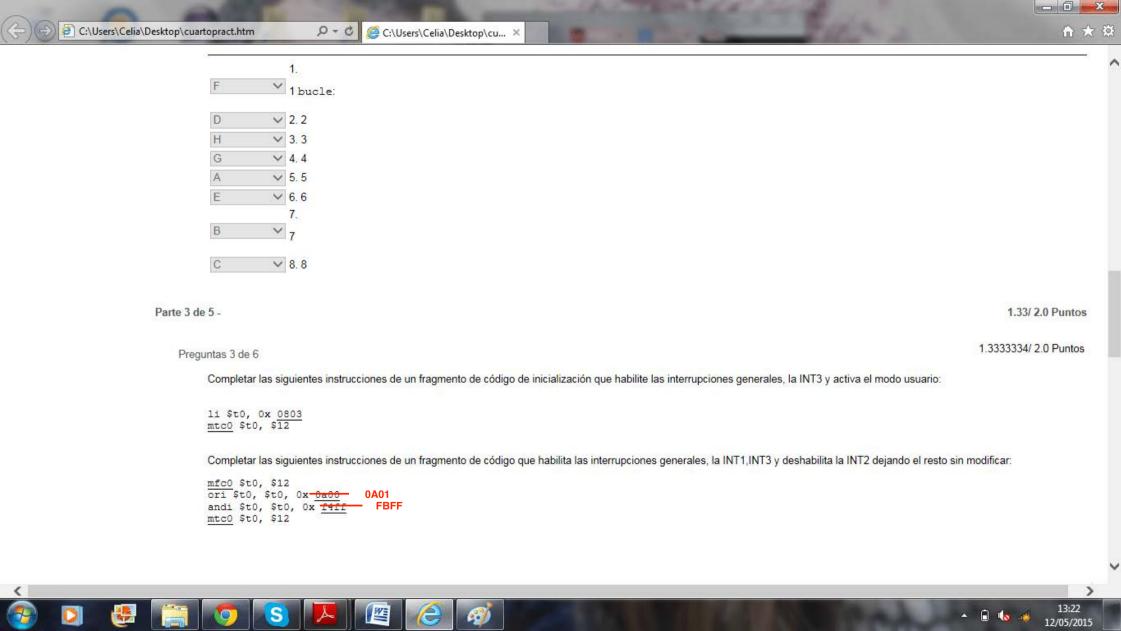


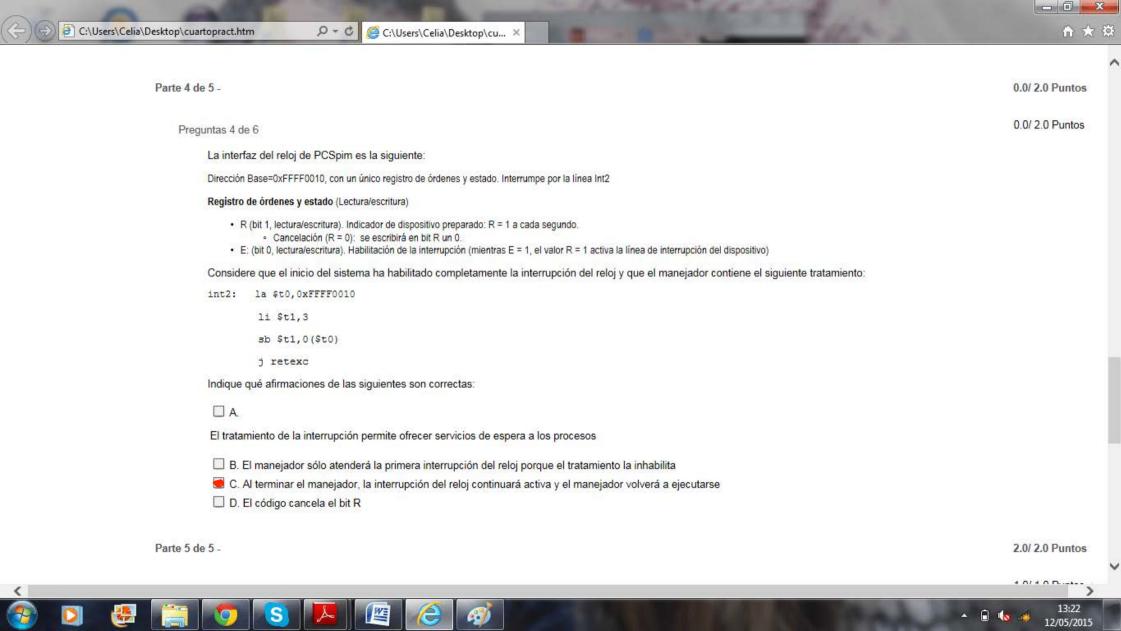


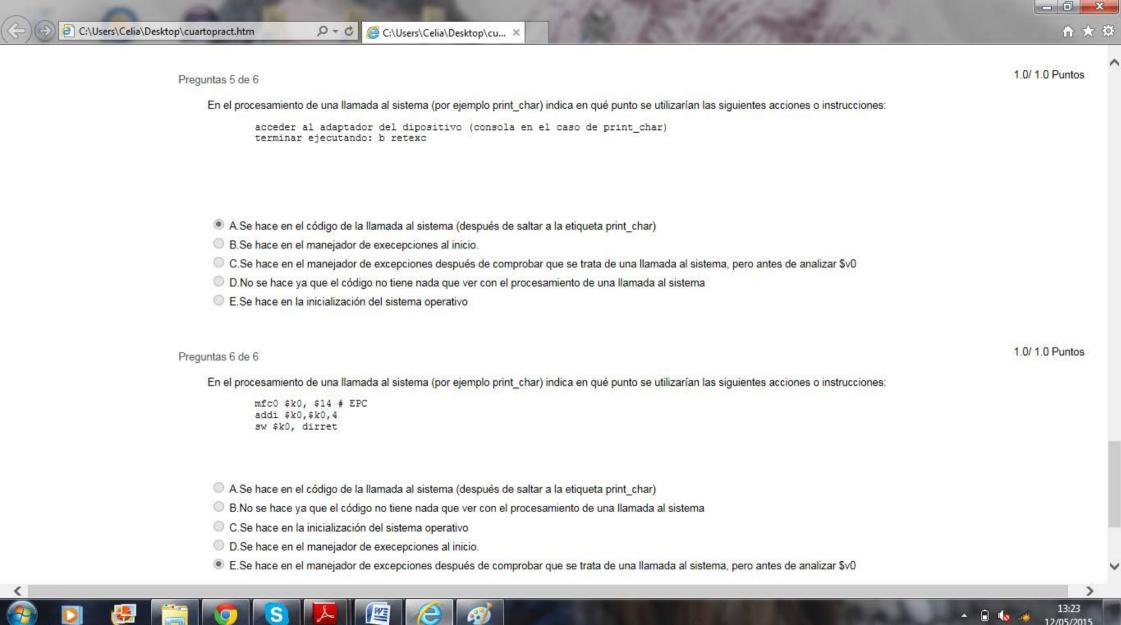












Tenemos una interfaz de consola en la dirección 0xFFFF2000 con los siguientes registros:

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

- R (bit 4, sólo lectura). Indicador de dispositivo preparado: R = 1 cuando la consola está disponible.
- E: (bit 7, lectura/escritura). Habilitación de la interrupción (mientras E = 1, el valor R = 1 activa la línea de interrupción del dispositivo, línea INT1*).
- C: (bit 2, escritura). Cancela la interrupción al escribir un '1', y R vuelve a valer 0 y se desactiva la línea de interrupción si la hubiera.

Registro de datos (Sólo escritura. Dirección = Base + 4). Ambos registros son de 8 bits.

El siguiente programa escribe el carácter de la posición de memoria "Valor" en la consola, sincronizando por consulta de estado (la interrupción estará deshabilitada). Completar las instrucciones que hay detrás del bucle de consulta de estado (el cual no se muestra) que primero deben cancelar el bit R y luego escribir el valor en la consola.

```
... bucle de consulta de estado

li $t1, 0x

sb $t1,

lbu $t1,

sb $t1,

sb $t1,
```

Parte 2 de 5 -

Preguntas 2 de 6

2.0 Puntos

Tenemos una interfaz de teclado en DB=0xFFFF0000 que tiene un registro de 8 bits de Estado/Ordenes, de Lectura/escritura con Dirección = DB, con los siguientes bits:

- R (bit 3, 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 0, lectura/escritura). Habilitación de la interrupción (mientras E = 1, el valor R = 1 activa la línea de interrupción del dispositivo). Línea INT3*.

Y un registro de datos (Sólo lectura. Dirección = DB + 1).

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

Seleccionar las instrucciones que usarías indicando el orden para realizar un bucle de consulta de estado que espere hasta que se haya pulsado una tecla y posteriormente borre el bit R. Asumir que en \$t1 hemos cargado la dirección base.

NOTA: el programa sólo tiene 4 instrucciones, hay 3 que no son correctas (un branch, un and y una load o store)

```
A.
sb $zero, 0($t1)
B.
beqz $t0, espera
C.
lbu $t0, 0($t1)
D.
lbu $t0, 1($t1)
E.
andi $t0, $t0, 0x01
F.
andi $t0, $t0, 0x08
G.
bnez $t0, espera
1. C
(1) espera:
2. F
(2)
3. B
(3)
4. D
(4)
5. G
(branch incorrecto)
6. E
```

(and incorrecto)

(load o store incorrecto)

Parte 3 de 5 -

Preguntas 3 de 6

2.0 Puntos

Completar las siguientes instrucciones de un fragmento de código de inicialización que habilite las interrupciones generales, la INT5 y activa el modo usuario:



Completar las siguientes instrucciones de un fragmento de código que habilita las interrupciones generales, la INT0,INT3 y deshabilita la INT1 dejando el resto sin modificar:



Parte 4 de 5 -

Preguntas 4 de 6

2.0 Puntos

La interfaz de teclado del simulador PCSPIM contiene dos registros de un byte ubicados a partir de la direción base DB = 0xFFFF0000 y tiene la posibilidad de interrumpir por la linea irq0*. La descripción de los registros es la siguiente:

Registro de Estado/Órdenes (lectura/escritura, dirección = DB)

- R (bit 0, sólo lectura). Indicador de dispositivo preparado. R=1 cada vez que se pulsa una tecla.
 - o Cancelación: Para hacer que R=0, es necesario leer el registro de datos.
- E (bit 1, lectura/escritura). Habilita la petición de interrupciones del dispositivo. Mientras E=1, se activará irq0* cada vez que R=1.

Registro de Datos (Sólo lectura, dirección = DB + 4)

• COD (bits 0..7). Código ASCII de la última tecla pulsada. Leer de este registro provoca la puesta a cero del bit R.

Supón que, en el arranque del sistema, se ha habilitado completamente la interrupción del teclado. Considere la siguiente rutina de servicio de la interrupción 0:

```
irq0: la $t0,0xFFFF0000
    lb $t1,0($t0)
    andi $t1,$t1,2
    sb $t1,0($t0)
    b retexc
```

Indique qué afirmaciones de las siguientes son correctas sobre esta rutina de servicio:

- A. No cancela correctamente la petición de interrupción.
- ☐ B. El carácter leido se almacena en el registro \$t1.
- C. Accede al teclado por consulta de estado.
- D. Cancela correctamente la petición de interrupción.

Parte 5 de 5 -

Preguntas 5 de 6

1.0 Puntos

En el procesamiento de una llamada al sistema (por ejemplo print_char) indica en qué punto se utilizarían las siguientes acciones o instrucciones:

```
li $v0,0x12
li $a0,0x41 # código del carácter A syscall
```

- A.Se utilizan en el código del proceso usuario cuando hace una llamada al sistema
- B.Se hace en el manejador de excepciones después de comprobar que se trata de una llamada al sistema, pero antes de analizar \$v0
- C.Se hace en el código de la llamada al sistema (después de saltar a la etiqueta print_char)
- O D.Se hace en la inicialización del sistema operativo
- © E.Se hace en el manejador de execepciones al inicio.

Borra selección

Preguntas 6 de 6

1.0 Puntos

En el procesamiento de una llamada al sistema (por ejemplo print char) indica en qué

punto se utilizarían las siguientes acciones o instrucciones:

mfc0 \$k0, \$14 # EPC addi \$k0, \$k0, 4 sw \$k0, dirret

- A.Se hace en el código de la llamada al sistema (después de saltar a la etiqueta print_char)
- B.Se hace en el manejador de excepciones después de comprobar que se trata de una llamada al sistema, pero antes de analizar \$v0
- C.Se hace en la inicialización del sistema operativo
- O D.Se utilizan en el código del proceso usuario cuando hace una llamada al sistema
- C E.Se hace en el manejador de excepciones al inicio. Borra selección