

## Curso de Organización del Computador CI3815: Problemario de ejercicios

### Interrupciones y Excepciones

El objetivo de este y otros problemarios que estamos colocando a disposición de los estudiantes del curso de organización del computador, es que puedan poner en práctica los conocimientos adquiridos en las clases de teoría.

Si usted considera que hay un error en este material por favor comuníquese a los profesores de teoría para aclarar cualquier tipo de inconveniente, y de ser el caso, realizar las correcciones que sean necesarias.

### Interrupciones y Excepciones

- 1) El presente programa es un manejador de interrupciones, extremadamente simple, para SPIM, donde se usa la sub-rutina `print_exc` que solamente imprime valores que recibe como parámetros en su llamada. Se le pide que usted describa: ¿Cuál tratamiento es el que está haciendo este programa de tratamiento de interrupciones cuando ocurre una interrupción o una excepción en el sistema?

```
        .ktext 0x0000080 # (0)
        sw $a0,save0 # (1)
        sw $a1,save1
        mfc0 $k0,$13
        mfc0 $k1,$14
        sgt $v0,$k0,0x44 # (3)
        bgtz $v0,done
        mov $a0,$k0 # (4)
        mov $a1,$k1
        jal print_exc

done:
        lw $a0,save0 # (5)
        lw $a1,save1
        addiu $k1,$k1,4
        rfe # (6)
        jr $k1 # (7)
        .kdata
save0:  .word 0
save1:  .word 1
```

- 2) Dada la siguiente lista de 5 interrupciones y excepciones que pueden ocurrir en un computador en algún momento
  - i) Excepción por división por cero
  - ii) Interrupción de *Timer*
  - iii) Interrupción de Teclado
  - iv) Interrupción por error en la Memoria
  - v) Excepción por instrucción inválida

Se le pide que usted indique cuáles de estas interrupciones o excepciones serán atendidas y cuáles estarán enmascaradas si el registro de Estado (Registro12 o *Status*) contiene alguno de los siguientes valores:

a) 0x0000FF01   b) 0x00EF00C1   c) 0x0000FF81   d) 0x0000F003

- 3) Suponga que un usuario corre su programa y se da cuenta de que durante su ejecución ocurre una excepción. Suponga ahora que usted recibe un *log* de la ejecución de ese programa (obtenido mediante la opción *Save Log File* de QtSpim), del instante justo después de que ha ocurrido la excepción, y recibe también el programa original en un archivo de texto. En esta situación se le pide que usted responda a las siguientes preguntas:

- a) ¿Estará usted en la capacidad de detectar que instrucción del programa ocasionó la excepción?
- b) ¿Estará usted en capacidad de arreglar el programa, de forma tal que la excepción no vuelva a ocurrir?
- c) Si su respuesta a la pregunta anterior es afirmativa indique de qué manera puede hacerlo.

Puede asumir que este archivo de *log* contiene todos los contenidos de la memoria, los registros, así como las secciones de texto y de datos del programa original que estaba siendo ejecutado.

- 4) Suponga que se dan las mismas condiciones que se expusieron en la pregunta anterior. En este caso, si el programa del usuario es interrumpido por una interrupción, en lugar de una excepción, responda a las siguientes preguntas:

- a) ¿Usted pudiera ubicar que instrucción del programa del usuario es la que ocasiono la interrupción?
- b) ¿Usted pudiera arreglar el programa, de forma tal de garantizar que la interrupción no vuelva a ocurrir durante la ejecución de ese programa?

- 5) Normalmente en SPIM al finalizar una rutina de tratamiento de interrupciones, se utiliza la instrucción *eret* para retornar al programa Px que se estaba ejecutando (en versiones anteriores de SPIM se usaba *rfe*). En un programa de tratamiento de interrupciones que usted pudiera desarrollar, ¿Sería posible que la instrucción *eret* fuera sustituida por una instrucción *jr \$ra ?*, de ser afirmativa su respuesta, indique que instrucciones deberían ser adicionadas al final de la rutina de tratamiento de interrupciones para que el flujo de ejecución del computador, retome adecuadamente la ejecución del programa Px que estaba siendo servido antes de que ocurriera la interrupción.

- 6) Si durante la ejecución de un programa Py ocurre una interrupción o excepción y justo en el momento en el que comienza a ejecutarse el programa de tratamiento de interrupciones usted puede ver el contenido del registro Causa (Registro 13 o *Cause*). ¿Pusiera usted decir cuántas y cuales interrupciones o excepciones han ocurrido hasta ese instante?

Si el registro Causa (Registro 13 o *Cause*) contuviera alguno de los siguientes valores:

a) 0x80000000   b) 0x00000020   c) 0x80007030

¿Qué información pudiera usted deducir de lo que ha ocurrido recientemente en ese computador?, para cada uno de los valores del registro Causa antes expuestos.

- 7) Un computador puede tener un mecanismo único para atender las interrupciones que luego bifurque a una única rutina manejadora de interrupciones o excepciones, como ocurre con SPIM, o puede tener un vector de interrupciones en el que cada posición del vector permite bifurcar a una rutina de tratamiento de interrupciones y excepciones diferentes. ¿Qué ventajas puede brindar cada uno de estos esquemas al administrador de un computador? ¿Cuáles pudieran ser las desventajas de cada uno de los dos esquemas planteados?
- 8) Si las excepciones provienen de situaciones que están presentes en los programas de los usuarios ¿Por qué se utiliza un mecanismo general para hacer el tratamiento de estas situaciones, en lugar de dejarle a los usuarios la responsabilidad de la atención de esta situaciones, como parte del código normal que su programa debe ejecutar?

Material elaborado por: Prof. Ricardo Gonzalez 2013