Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
Curso: Organización del Computador (CI-3815)
Trimestres: Enero-Marzo 2012

Proyecto 2 (10%) Manejo de Interrupciones y Excepciones

Objetivo General

El objetivo de esta asignación es que el estudiante se familiarice con los principios básicos de interrupciones.

Objetivos Específicos

1. Implementar manejadores para las interrupciones de teclado (Receiver) y de reloj (Timer).

Enunciado

Se desea que usted modifique el archivo exception.s que trae SPIM para crear un nuevo programa exceptest.s que funcionará como un simulador de un planificador de un sistema operativo. Un planificador o *scheduler* es un programa que se encarga de que otros programas se ejecuten.
El simulador exceptest.s a su vez ejecutará un programa de nombre progini.s que contendrá información de dos o mas programas a ejecutarse y que compartirán el CPU, gracias al manejo de interrupciones, las estructuras y las estrategias que usted implementará en el programa exceptest.s

Para ejecutar los programas que estén en progini.s usted empleará un algoritmo de tiempo compartido o *time sharing*, en este algoritmo, al conjunto de estructuras asociadas a cada programa le daremos el nombre de proceso. A cada uno de los procesos *Pi* se le permite ejecutarse por un tiempo *t* (llamado quantum), luego del cual se suspende su ejecución y se pasa a ejecutar al siguiente proceso Pj de una lista de procesos listos o *ready*, al llegar al final de la lista de procesos listos, se vuelve a comenzar con el primero de ellos.

Si alguno de los procesos termina su ejecución antes de haber terminado su tiempo éste libera el CPU y debe dar paso al siguiente proceso de la lista de procesos listos para ejecutarse, una vez que un proceso ha culminado su ejecución no debe ser tomado más en cuenta a la hora de elegir el próximo proceso a ejecutarse.

Todo proceso, al que le sea suspendida su ejecución, cuando vuelva a reanudarse su ejecución, ésta debe continuar en el mismo punto en el que quedó cuando su ejecución fue suspendida.

Una vez que todos los programas que están en progini.s han finalizado sus instrucciones, el programa que usted elabore debe emitir un mensaje en el que se indique la finalización de todo el trabajo que se estaba ejecutado y terminar su ejecución.

Para poder cumplir con las especificaciones del proyecto, su programa debe mantener una lista de los programas que están en posibilidad de ejecutarse, para que un programa Pi pueda ser colocado en esta lista se debe invocar a la instrucción break 0x2 pasando como parámetro su dirección de comienzo en el registro $a0 (usando un mecanismo similar al de los syscall).

Un programa que desea terminar su ejecución debe invocar a la instrucción break 0x4.

Su programa debe ejecutar cada uno de los programas que se registren un tiempo de k milisegundos. Por razones de la implementación este valor k solo podrá estar en múltiplos de 10 milisegundos, por ejemplo: 2000 milisegundos o 100 milisegundos. Luego de que transcurra el tiempo k, se debe suspender la ejecución del proceso actual, para dar paso al siguiente. Los procesos a su vez pueden voluntariamente darle paso al próximo proceso si ejecutan la instrucción break 0x8.

Para consultar información de los mecanismos de excepciones e interrupciones y del formato de la instrucción break y su funcionamiento, pueden consultar el Apéndice A, que está publicado aula virtual en la sección de documentos.