Sistemas Operativos

Interrupciones y llamadas al sistema

Daniel Ochoa

{dochoa}@fiec.espol.edu.ec

Objetivos: Conocer como monitorear las interrupciones

Requerimientos: Un computador con GNU/Linux.

|  |
| --- |
| **Actividad 1: Interrupciones** |

Sabemos que el SO ejecuta sus funciones cuando recibe una señal. Algunas señales son generadas por dispositivos otras por los procesos. Las generadas por dispositivos (y que llegan al procesador) se conocen también como interrupciones. El número de interrupciones depende del procesador y son identificadas por un IRQ (canal). Es posible conocer cuantas y de que tipo son las interrupciones que se generan durante la operación de un sistema de cómputo. Realice el siguiente experimento:

En un terminal ejecute el comando: *more -f /proc/interrupts*, responda las siguientes preguntas:

1) En el comando "more -f /proc/interrupts" identifique el programa que se ejecuta, los parámetros del programa y la fuente de informaciòn de las interrupciones. ¿Qué hace este comando?

|  |  |
| --- | --- |
| **A** | El programa ejecutado es “more”. Se pasó como parámetro la ruta de un archivo. La fuente de información es el archivo con las interruption requests registrada/La fuente de información son los núcleos del procesador. El comando muestra por pantalla los núcleos como columnas y como filas las diferentes interruption request. Debajo de cada núcleo se detalla la cantidad de veces que una cierta interrupción se lanzó. |

2) Si usted ejecuta varias veces el comando va a notar que el número de interrupciones registradas aumenta en diferente proporcion. Qué evidencia proporciona la salida del comando para comprobar si su sistema implementa mecanismos de:

|  |  |
| --- | --- |
| **B: timesharing** | El número de diferentes interrupciones en un núcleo aumenta, lo que nos dice que un núcleo ha ejecutado diferentes procesos que lanzaron diferentes interrupciones. |

|  |  |
| --- | --- |
| **C: Paralelismo** | El número de interrupciones en todos los núcleos ha aumentado, evidenciando que todos los núcleos han estado ejecutando diferentes procesos. |

|  |  |
| --- | --- |
| **D: Acceso a dispositivos** | El Contador para la interrupción 1 aumenta, esta interrupción es lanzada por el controlador del teclado. |

|  |
| --- |
| **Actividad 2: Llamadas al sistema** |

Cuando el procesador, o un proceso genera una señal se dice que se tienen excepciones (que son manejadas de forma identica que las interrupciones). En el caso de las llamadas al sistema es posible conocer cuando se generan en un programa. Descarge estos archivos: [hello\_world.c](https://www.sidweb.espol.edu.ec/courses/40809/files/2769262/download?wrap=1) , [hello\_world.s](https://www.sidweb.espol.edu.ec/courses/40809/files/2769263/download?wrap=1)

¿Solo revisando sus còdigos fuentes cuáles son las diferencias y similudes de ambos programas?

|  |  |
| --- | --- |
| **E** | Las diferencias es que están escritos en lenguajes distintos y la manera en que hacen las llamadas al sistema. Similitudes es que ambos lanzan llamadas al sistema, el archivo .c con la librería stdio mientras que el archivo .s con syscall; otra similitud es que ambos tienen el mismo resultad, que es mostrar por pantalla un mensaje. |

El primero se compila así: gcc -o hello\_world hello\_world.c y el segundo asi:gcc -nostdlib  hello\_world.s. Luego ejecute el comando *strace Nombre\_programa\_ejecutable,* para ambos casos. Indique que hace el programa strace

|  |  |
| --- | --- |
| **F** | Para ambos programas muestra las llamadas al sistema que invocadas. |

¿Que llamada al sistema escribe en la pantalla?

|  |  |
| --- | --- |
| **G** | La llamada al sistema es write |

¿Que número tiene esa llamada al sistema?

|  |  |
| --- | --- |
| **H** | Es el 1 |

¿Por qué la cantidad de llamadas al sistema es diferente si se ejecuta uno u otro programa?

|  |  |
| --- | --- |
| **I** | El programa compilado desde el archivo .c hace uso de stdlib la cual internamente realiza más llamadas al sistema para validaciones y demás acciones realizar su función. Por otro lado el archivo .s escrito en assembly se invoca a escribir por pantalla directamente con syscall |