Undécima Clase

Presentación

En la clase anterior hemos aprendido el formato de las direcciones y la segmentación de memoria y también hemos aprendido como llamar a subrutinas que agrupan cálculos que se realizan frecuentemente.

En esta clase presentaremos la forma de ejecutar módulos de un sistema operativo mediante interrupciones por software y por hardware.

Es importante relacionar los contenidos de esta con lo desarrollado en Sistemas de Computación I y en las dos primeras unidades de esta asignatura.

La propuesta del Trabajo Práctico Requerido Nº 3 lo invitará a recuperar conocimientos y habilidades para codificar y programar en Assembler, agregando un nuevo reto: llamar a subrutinas mediante las instrucciones CALL e INT.

A través del estudio de esta unidad esperamos que usted adquiera la capacidad para:

- Conocer el significado de una interrupción, su utilización y su desarrollo.
- Vincular las interrupciones y los sistemas operativos.
- Comprender las diferencias entre la ejecución de una interrupción por hardware y una interrupción por software.

A continuación, le presentamos un detalle de los contenidos y actividades que integran esta unidad. Usted deberá ir avanzando en el estudio y profundización de los diferentes temas, realizando las lecturas requeridas y elaborando las actividades propuestas.

Contenidos y Actividades

1. Interrupciones

Lectura Sugerida

- Ginzburg, M.; **Como el computador arranca y carga el Sistema Operativo**. En su: **La PC por dentro** 6ª ed. Buenos Aires: Biblioteca Técnica Superior, 2012, p.1-46 a 1-53
- Ginzburg, M.; Apéndice 2 de la Unidad 1. **Sistemas operativos: su lugar y funciones como parte del Software.** En su: La PC por dentro. 4ª ed. Buenos Aires: Biblioteca Técnica Superior, 2006, p.153.

Lectura requerida

- Ginzburg, M.; **Interrupciones por Software**. En su: **Assembler desde cero e interrupciones**. 3ª ed. ampliada. Buenos Aires: 2010.

2.1. Desarrollo de una interrupción por software

Lectura requerida

- Ginzburg, M.; Utilización del programa Assembler. En su: **Assembler desde cero e interrupciones**. 3ª ed ampliada. Buenos Aires: 2010.

2.2 Desarrollo de una interrupción por hardware

Lectura requerida

- Ginzburg, M.; Utilización del programa Assembler. En su: **Assembler desde cero e interrupciones**. 3ª ed ampliada. Buenos Aires: 2010.

Trabajo Práctico Requerido

- Trabajo Práctico Nº 3: **Llamado a Subrutinas e Interrupciones por Software.**

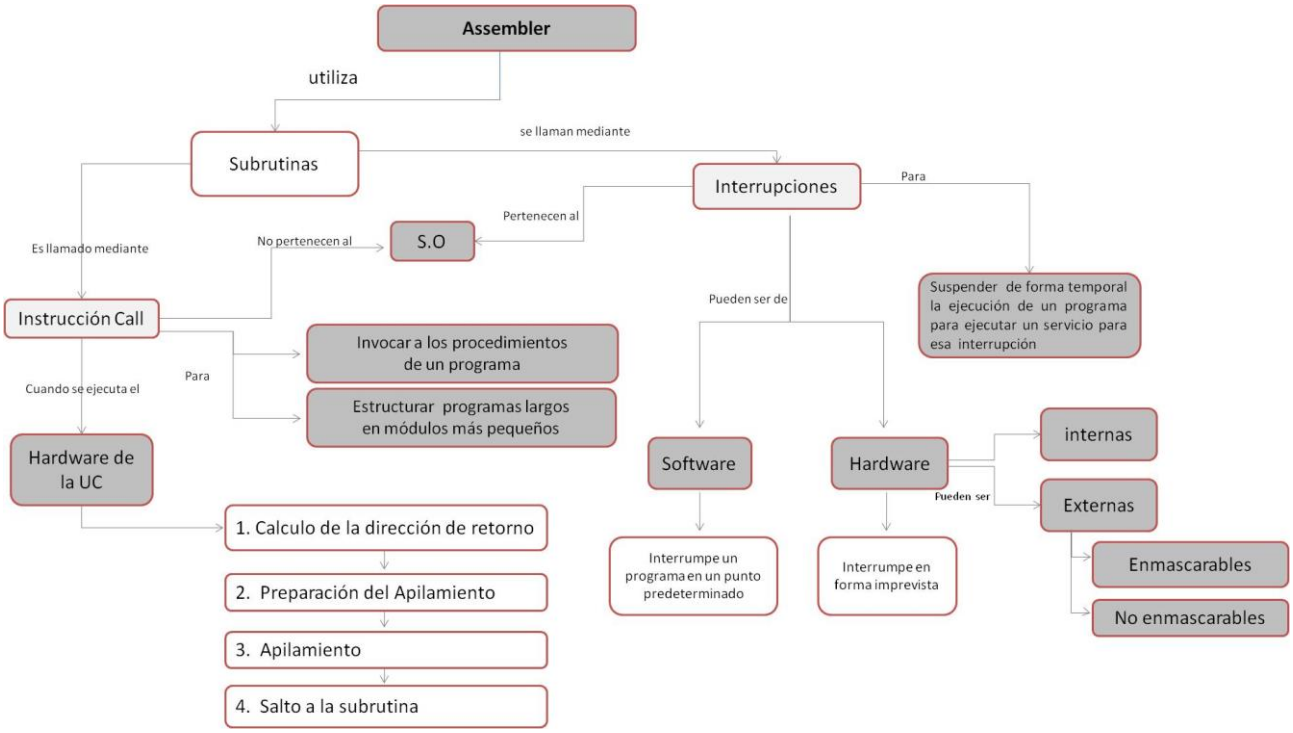
BIBLIOGRAFÍA OBLIGATORIA

Ginzburg, Mario; **Assembler desde Cero e interrupciones** 3ª ed ampliada. Buenos Aires: Biblioteca Técnica Superior, 2003.

Ginzburg, Mario; La PC por Dentro 6ª ed. Buenos Aires: Biblioteca Técnica Superior, 2012

Ginzburg, Mario; **La PC por Dentro** 4ª ed. Buenos Aires: Biblioteca Técnica Superior, 2006

Organizador Gráfico



Lo/a invitamos a comenzar con el estudio de los contenidos que conforman esta séptima clase

1. Interrupciones

Hasta el momento hemos trabajado con subrutinas de usuario utilizando la instrucción CALL. Ésta, entre otras cosas, llama o invoca una subrutina para calcular una raíz cuadrada o para ordenar números. Este tipo de subrutinas no pertenecen al sistema operativo.

Por razones de seguridad y practicidad existen otras subrutinas que forman parte del sistema operativo. Éstas se "llaman" mediante **interrupciones**. Como su nombre lo indica, suponen la suspensión temporaria de la ejecución de un programa, para pasar a ejecutar una subrutina que da servicio a esa interrupción. Luego de ejecutarse se reanuda la ejecución del programa en el punto donde quedó interrumpido.

Para facilitar la comprensión de este mecanismo particular es importante recordar que el sistema operativo (S.O) se encarga de administrar los cuatro recursos de un sistema:

- El manejo de memoria
- Prioridades relativas al orden en que la UC ejecuta los programas que están en memoria.
- El manejo de archivos
- El manejo de periféricos.

Le acercamos el siguiente material que lo ayudará a establecer puentes significativos entre sus conocimientos previos y los contenidos que desarrollaremos a continuación.

Lectura Sugerida

Ginzburg, M.; Apéndice 2 de la Unidad 1. Sistemas operativos: su lugar y funciones como parte del Software. En su: **La PC por dentro.** 4ª ed. Buenos Aires: Biblioteca Técnica Superior, 2006, p.153.

Las interrupciones pueden ser **por software o por hardware.** Las primeras se originan cuando se ejecuta la instrucción que para el Assembler de Intel es **INTxx** (XX es un número en hexa). Por ejemplo, se puede emplear para llamar a una subrutina que abra o guarde un archivo.

Tienen su origen en la necesidad de que un programa se autointerrumpa en un punto ya predeterminado. La instrucción que se pasa a ejecutar luego de INTxx es la primera de una subrutina del SO., y de manera semejante a CALL, cuando se termina de ejecutar esta subrutina se retorna a la instrucción que sigue en memoria a INTxx.

Por otro lado, una **interrupción por hardware** interrumpe imprevistamente un programa de usuario en ejecución para que se pase a ejecutar una subrutina de S.O o del BIos, que no se relaciona con los objetivos del programa interrumpido. Estas interrupciones no se originan en ninguna instrucción que llama al S.O, sino por instrucciones que la UCP no puede ejecutar (internas a la UCP), o (externas a la UCP) porque un periférico avisa mediante su interfaz, que una subrutina del S.O o del BIOS debe atender a un evento ocurrido. Un claro ejemplo es la terminación de escritura de un sector en el disco por ADM o problemas de papel con la impresora.

Por lo afirmado hasta aquí se pueden identificar dos tipos de interrupciones por hardware:

- **Internas** (a la UCP) por ejemplo, ejecución de una instrucción de código de operación inexistente, u otra de división cuyo resultado no entra en el formato predeterminado, u otra que ordena escritura a celdas de memoria no permitidas por razones de seguridad, para no destruir información que debe conservarse. En todos estos casos, la instrucción en cuestión no puede terminar de ejecutarse, y se pasa a ejecutar una subrutina del S.O para atender el evento ocurrido.
- **Externas** (a la UCP) que a su vez pueden ser:

  Enmascarables se puede demorar su detención, por ejemplo: la activación de una línea **IRQ** de un bus, que es la forma más corriente de una interrupción por hardware, que también llama a una subrutina del S.O o del BIOS, para que atienda dicha activación. Ésta, como se anticipó, puede originarse en algún evento que avisa un periférico mediante su interfaz que activa su línea IRQ.

  No enmascarables deben ser atendidas sin demora. Por ejemplo: las acciones a seguir cuando cesa accidentalmente el suministro de energía de un computador.

Supongamos que un programa "A" solicitó una escritura en un disco por medio de una interrupción por software (ejecución de una instrucción INTxx) y quedó autointerrumpido. Cuando se complete la escritura, la interfaz del disco originará una interrupción por hardware activando una línea IRQ del bus al que está conectada. Esta activación implicará una solicitud de interrupción (Interrupt ReQuest = IRQ) para que se interrumpa el programa "Z" en ejecución en ese momento, a fin de que una subrrutina del S.O o del BIOS verifique que la grabación pedida por el programa "A", se haya concretado correctamente.

Entonces, resulta que este programa "Z" fue interrumpido, a partir de la activación de esa IRQ para que una subrutina del S.O lleve a cabo algo relacionado con otro programa "A", y no con el programa Z.

De este modo, las interrupciones tipo IRQ llaman a subrutinas del S.O que atienden a eventos que **no** están contemplados en el programa que se está ejecutando y que, típicamente, no están relacionados con él.

Luego de esta breve introducción sobre qué es una interrupción y cómo se clasifican, lo invitamos a continuar con la siguiente actividad de lectura.

Lectura requerida

- Ginzburg, M.; **Interrupciones por Software**. En su: **Assembler desde cero e interrupciones**. 3ª ed. ampliada. Buenos Aires: 2010. p.72 - 80
- Ginzburg, M.; **Interrupciones por Hardware**. En su: **Assembler desde cero e interrupciones**. 3ª ed. ampliada. Buenos Aires: 2010. p. 80 - 82
-

Guía para la lectura

- ¿Qué son las interrupciones?
- ¿Cómo opera una interrupción por hardware externa?
- ¿Cómo operan las interrupciones por Software?
- ¿Cómo se retorna al programa interrumpido?
- ¿Qué es la zona de memoria principal denominada pila"?
- ¿Qué es la zona de memoria de "vectores interrupción"?

1.1. Desarrollo de una interrupción por software

La ejecución de una instrucción de **INTxx** da lugar a que la UC realice un proceso que permite localizar la subrutina que atiende a esa interrupción, dado que durante éste, en la pila del programa interrumpido, la UC guardará los registros CS, IP y RE.

La ultima instrucción **IRET** de la subrutina del SO ordenará volver a la instrucción que sigue a INTxx, permitiendo que se reanude la ejecución del programa que fue autointerrumpido por INTxx.

Para conocer más en detalle el desarrollo de una interrupción por software, le proponemos realizar la siguiente actividad. En ella encontrará, también, ejemplos ilustrativos que pueden facilitar su comprensión.

Lectura requerida

Ginzburg, M.; Interrupciones por Software. En su: **Assembler desde cero e interrupciones**. 3ª ed. Buenos A2010 p. 44 a 49.

Guía para la lectura

- Una instrucción tiene como componentes CS: IP = 2040:328A. Determinar su dirección efectiva indicándola con cinco cifras en hexa.
- ¿Qué contiene un vector interrupción? ¿Cuál es su contenido y para qué sirve? ¿Cómo se obtiene su dirección?
- ¿Cuál es siempre para Intel en modo real la componente izquierda de la dirección de un vector?
- ¿En qué modo opera la CPU cuando se pasa a ejecutar la subrutina que llamó una instrucción INTxx?
- ¿Qué diferencia existe entre las intrucciones CALL e INTxx.
- ¿Qué es modo "user" y "modo kernell"? y cómo se pasa de uno a otro?

1.2 Desarrollo de una interrupción por hardware

Como se ha afirmado, la activación de una línea **IRQ** de un bus por parte de una interfaz de un periférico origina una interrupción por hardware del **tipo enmascarable**. La UC según se verá interrumpirá al programa en ejecución, siempre y cuando, éste no sea del S.O y, luego, tendrá lugar un proceso semejante al que ocurre en una interrupción por software. En el caso que la solicitud de interrupción IRQ no sea habilitada por la UC, ésta deberá esperar hasta que la UC la autorice. Esta espera o "enmascaramiento" temporario no ocurre en ningún otro tipo de interrupciones. La ejecución de una instrucción INTxx (interrupción por software) no puede demorarse, como ocurre con cualquier instrucción.

Los pasos que suceden en una interrupción de tipo IRQ se desarrollan en el siguiente material de lectura.

Lectura requerida

Ginzburg, M.; Interrupciones por Hardware. En su: **Assembler desde cero e interrupciones**. 3ª ed. Buenos Aires: 2010.

Guía para la lectura

- Describir los pasos que ocurren durante el proceso de una interrupción por hardware hasta que se retorna al programa interrumpido.
- ¿Cómo se localiza el vector interrupción en una interrupción tipo IRQ?
- ¿Qué sucede cuando el flag I no permite la interrupción?
- A partir de qué momento tienen igual desarrollo las interrupciones por Hardware y las interrupciones por software?
- ¿Por qué las subrutinas que atienden a las interrupciones por software o por hardware deben terminar con la instrucción IRET en lugar de la intraucción RET como ocurre en las subrutinas llamadas por CALL?

Luego de analizar y estudiar los contenidos correspondientes a esta unidad le proponemos realizar el Trabajo Practico Requerido.

Trabajo Práctico Requerido

Trabajo Práctico Nº **: Llamado a Subrutinas e Interrupciones por Software**

Cierre de la unidad

Como cierre de Unidad, le sugerimos armar un cuadro comparativo identificando las principales diferencias entre interrupciones por software y por hardware y del llamado a subrutinas mediante CALL e INTxx. Esta actividad le ayudara a realizar una síntesis de los contenidos más importantes de la unidad