

Las **INTERRUPCIONES** en **MULTIPROGRAMACION**

-- 0 --

LAS INTERRUPCIONES Y EL BAILE DE LAS PSW

Seguramente todos los programadores se han preguntado alguna vez cómo el ordenador puede gestionar tantas tareas a un mismo tiempo sin perder el control. Este artículo intenta contestar esta cuestión exponiendo el modo como lo resolvió el D.O.S., el Sistema Operativo para grandes sistemas de IBM antecesor del MVS.

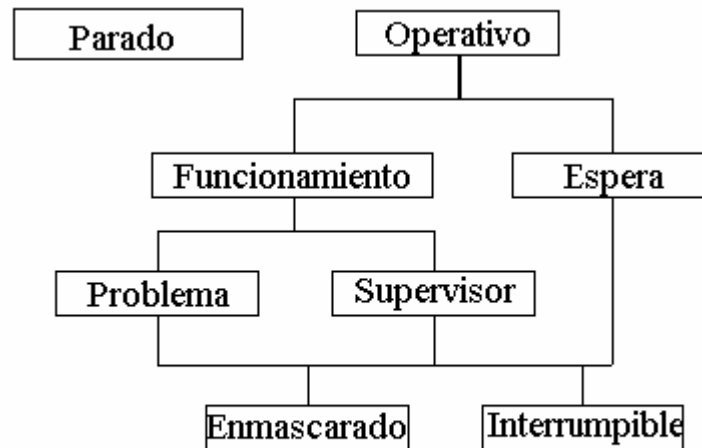
Introducción:

Como ya se comentó en la introducción de los Sistemas Gestores de Bases de datos, se entiende por *multiprogramación* el tratamiento en forma concurrente de más de un programa en tiempo real. Este concepto es el que nos ocupa, pues pretende estudiar cómo un único procesador o CPU puede ejecutar al mismo tiempo, al menos dos programas, el supervisor y un programa de usuario, o programa problema. Hay que tener en cuenta que en un gran sistema, para que se pueda ejecutar un programa en el ordenador, éste debe contar con un programa que controle las operaciones de entrada/salida necesarias para el tratamiento, explotación y puesta a punto de los programas realizados por el usuario o por el constructor. Este programa de control se llama **supervisor** y se ubica en la memoria principal, concretamente en las posiciones mas bajas, está siempre cargado, y tiene por misión:

- planificar la ejecución de todo tipo de programas
- controlar y coordinar las operaciones de E/S de todos los programas problema.
- Manejar todas las interrupciones.

Programa problema: recibe esta denominación todo programa que contenga operaciones de entrada de datos, proceso de esos datos, y salida de los datos elaborados, y que además se ejecute interactivamente con el supervisor. Se podría decir que los programas problema son los programas escritos por los programadores(nomina, facturación, etc..) , y algunos programas ofrecidos por los constructores (compiladores, sort, etc.). El supervisor se diferencia de estos programas porque no elabora información, solo controla el sistema.

Estados de programa de la CPU



Estados en los que se puede encontrar la CPU

Figura 1

La CPU puede encontrarse en diferentes situaciones o estados a lo largo de un trabajo, existiendo los 8 estados de programa que se muestran en la figura 1.

No todos los estados son excluyentes, por lo que la CPU puede estar al mismo tiempo en más de un estado.

Ejemplo: Cuando la CPU se encuentra en estado *operativo*, de *funcionamiento* y de *problema*, puede estar también en estado *enmascarable* o en estado *interrumpible*. En cambio, si la CPU se encuentra en estado de *parada* no puede estar en otro estado pues no puede ejecutar instrucciones.

- La CPU se encuentra en estado de *supervisor* cuando esta ejecutando instrucciones de supervisor.

- La CPU se encuentra en estado de *espera* cuando no se procesan instrucciones pero se mantiene a punto de estar en funcionamiento.

- La CPU se encuentra en estado *operativo* cuando esta o ejecutando instrucciones o en un estado de *interrupción*, es decir, o esta en estado de *funcionamiento* o esta en estado de *espera*.

Los 8 estados de programa se diferencian entre sí por los siguientes tres motivos:

- el modo como funciona la CPU: en cada estado de programa, hay ciertas cosas que la CPU puede hacer y otras que no.

- el modo como se indica la situación de la CPU, es decir cómo se sabe que la CPU se encuentra en cierto modo: La forma de indicar el estado de la CPU es bien por medio de alguna luz de la consola o bien por medio del estado de cierta o ciertas posiciones en algún lugar de memoria (PSW).

- El modo como se pasa a dicho estado: El modo de pasar de un estado a otro se hace bien por medio de alguna tecla de consola, o bien por lo que se llama **interrupciones**.

Normalmente el sistema estará en estado de *funcionamiento*, ejecutando instrucciones del programa problema o del supervisor. Pero, ocasionalmente ocurrirán operaciones que obliguen a la CPU a pasar un programa al estado de *espera*, es decir a parar la ejecución de sus instrucciones.

Interrupciones:

Cuando un mensaje llega al ordenador, su llegada debe crear una interrupción en el proceso que se está ejecutando, de forma tal que se pueda leer el mensaje y almacenarlo hasta que pueda ser tratado.

Realmente una interrupción es una solicitud hecha a la CPU para cambiar el estado del programa que se está ejecutando. Pero este cambio de estado de funcionamiento a espera, se puede realizar tanto por la aparición de una interrupción como por la ejecución de la instrucción de ensamblador **LOAD PSW**. Por ejemplo, siempre que se deba atender una operación de E/S la CPU deberá estar en estado *Supervisor*, ya que es este programa el que controla estas operaciones. Basta pensar que antes de que apareciera la operación de E/S la CPU estaba ejecutando instrucciones del programa problema y por lo tanto, se encontraba en estado de *problema*; pero, al aparecer la operación de E/S, este programa problema solicita que la CPU pase del estado de *problema* al estado de *supervisor*, con lo cual se habrá ocasionado una interrupción de llamada al supervisor, para poder realizar la operación de E/S.

Clases de interrupciones:

- E/S : se produce cuando termina una operación de E/S bien desde un dispositivo o hacia un dispositivo.
- Externa: se produce cuando el operador aprieta la tecla de interrupción que tiene la consola. En este tipo de interrupción la CPU pasa del estado de espera al estado de funcionamiento. Por ejemplo una señal del contador de tiempos.
- De programa: Se produce cuando un resultado aritmético es demasiado largo para entrar dentro del registro o del área de almacenamiento prevista, o cuando se quiere tratar caracteres alfabéticos como numéricos, etc.
- Por error de maquina: se produce cuando existe una deficiencia de funcionamiento en unidades que no son de E/S. Un ejemplo: pudiera ocurrir que al transferir datos de un sitio a otro, los circuitos ganasen o perdiesen un bit, generándose un error de paridad.
- De llamada al supervisor: esta interrupción se produce cuando el programa problema ejecuta la instrucción de ensamblador **SVC** (SUPERVISOR CALL) de llamada al supervisor. Con esta instrucción se pasa del estado de problema al estado supervisor.

Los programas problema pueden contener muchas condiciones que exijan pasar al estado de supervisor, pues éste debe gobernar todo el sistema, y por tanto, debe recibir el control cuando se quieran leer o escribir datos, cargar nuevos programas a ejecutar, poner a punto el contador de tiempos, cancelar programas que se están ejecutando, etc...

PSW Programa Status Word : Palabra de estado de programa.

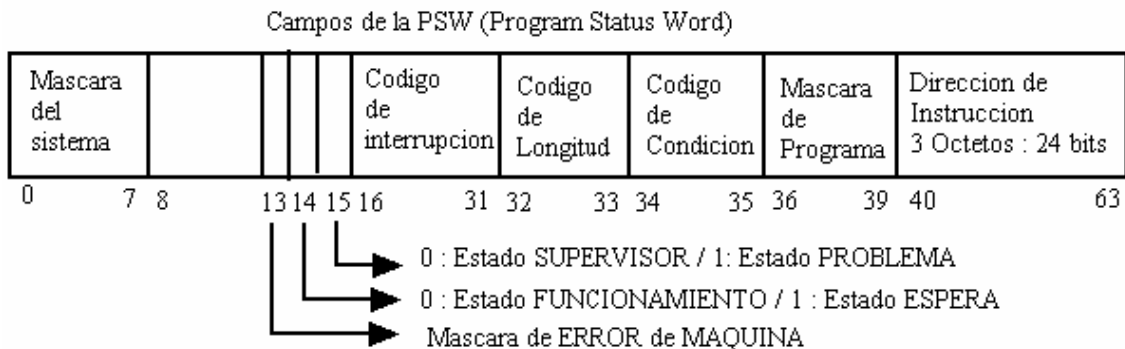


Figura 2

Esta es una doble palabra (*) de información, cuyo contenido, mostrado en la figura 2, refleja la situación lógica del ordenador respecto al programa problema que se está ejecutando, y dispone entre otros, de los campos que reflejan las condiciones lógicas siguientes:

- *Estado* en que se encuentra la CPU
- *Dirección* de la próxima instrucción a ejecutar
- *Relación* (Mayor, menor o igual) entre los últimos datos comparados en el programa (código de condición).

Existen varias PSW, sin embargo en un momento determinado solamente actúa una, recibiendo ésta la denominación de **PSW ACTUAL**. Todas las PSW están almacenadas en lugares concretos y predeterminados de la memoria, formando la PSW actual parte de la circuitería interna de la máquina.

En la misma figura 2 se puede apreciar cómo el valor de los bits 13, 14 y 15 se relaciona con los estados de la CPU, aunque estos valores tienen sentido solo en la **PSW ACTUAL**.

Por otra parte, y como ya se ha dicho anteriormente, el programa puede pasar al estado de espera o cuando se produce una interrupción, o cuando se ejecuta la instrucción **LOAD PSW**. Esta instrucción, que se ejecuta siempre en estado supervisor, escoge una PSW de la memoria, y la transfiere a otras posiciones de memoria, concretamente a las posiciones destinadas a la PSW actual. Así, cuando el programa problema pretende realizar una operación de E/S ejecutará entre otras, la instrucción de ensamblador **SVC**, que provocará la interrupción, con lo que la PSW actual será sustituida por una PSW nueva que tendrá un 0 en el bit 15.

Otro problema con el que se encuentra la CPU es que no siempre puede atender a todo tipo de interrupciones. Así, si el supervisor está procesando una interrupción de entrada/salida, toda interrupción de E/S que se produzca procedente de los canales(**) quedará enmascarada, es decir, pendiente de ejecución.

El enmascaramiento se logra por la presencia de bits 0 en los campos de máscara del sistema de la PSW actual, dedicándose el bit 7 para las interrupciones externas, y el resto para las de E/S.

Para enmascarar la CPU contra las deficiencias de funcionamiento de la máquina, se incluye el bit 0 en la *posición 13* de la PSW actual, es decir, en el campo de máscara de *error de máquina*.

Por su parte, las interrupciones de programa, provocadas por resultados aritméticos excepcionales y por otras condiciones se enmascaran incluyendo bits 0 en el campo de máscara del programa.

La interrupción de supervisor no tiene máscara, por lo que siempre era atendida.

El paso de 0 a 1 o de 1 a 0 de estos bits, se realiza mediante la ejecución de determinadas instrucciones de ensamblador. Así, para poner ceros o unos en la *máscara de programa*, el programa problema puede ejecutar la instrucción **SET PROGRAMA MASK**. Y, para cambiar

los bits del campo *mascara del sistema*, se ejecutará la instrucción **SET SYSTEM MASK**. Esta instrucción solo se puede ejecutar en estado de supervisor, aunque funciona igual que la **SET PROGRAMA MASK**.

Baile de las PSW.

Ya se ha comentado que hay movimientos de PSW, lo cual implica que hay varias PSW, pero una única es la que influye sobre el sistema en un momento determinado, es la **PSW ACTUAL**. Estos movimientos de PSW tienen su justificación, ya que se debe preservar el estado lógico del sistema antes de la interrupción para poder restablecerlo después de la interrupción.

Al sustituir la PSW actual por una nueva, aquella no se pierde, solo se almacena en otra posición concreta de memoria. Pero como una vez almacenada su contenido ha perdido actualidad, pasa a llamarse **PSW anterior**.

Así pues, se pueden apreciar los siguientes términos:

PSW nueva: Es la PSW que, como consecuencia de la interrupción, sustituye y se convierte en PSW actual.

PSW anterior: Es la PSW que fue sustituida y queda almacenada en espera de que se necesite de nuevo.

PSW actual: es la PSW que se usa durante la ejecución de una instrucción. Es la que influye sobre el funcionamiento del sistema.

Hay 5 PSW nuevas, una por cada tipo de interrupción, y las posiciones de memoria predeterminadas para contener las distintas PSW son las especificadas en la figura 3.

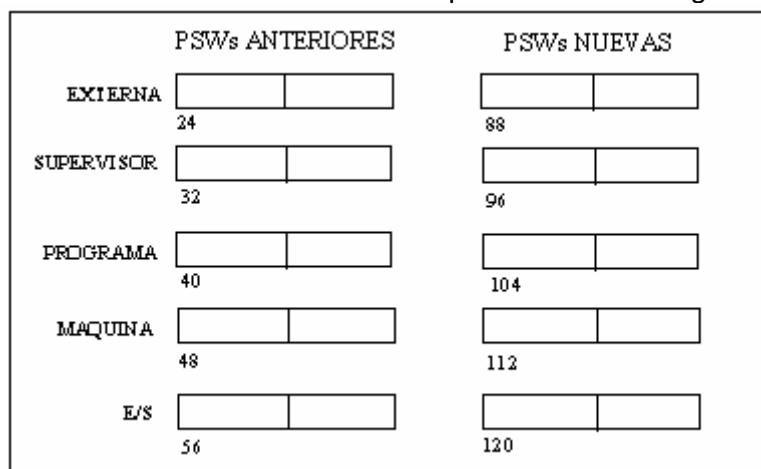


Figura 3

Que ocurre cuando se produce una interrupción:

El concepto clave para entender este entramado de PSWs es que dependiendo de cual haya sido la clase de interrupción, la **PSW actual** se almacenará en la posición de memoria correspondiente a la **PSW anterior** de la interrupción producida; y, la **PSW nueva** que contiene, entre otras cosas, la dirección de comienzo de la rutina encargada de tratar la interrupción producida, será la que se cargue como PSW actual. Es decir, la PSW actual se convierte en PSW anterior, y se almacena en el lugar previsto para la clase de interrupción generada; y la PSW nueva correspondiente a este tipo de interrupción, se convierte en PSW actual.

Es lógico, por tanto, que al haber 5 PSW nuevas, una por cada tipo de interrupción, cada una apunte a la primera instrucción de la rutina del supervisor que se ocupa de esta clase de interrupción.

EJEMPLO: Si se especifica en el programa problema el tratamiento aritmético de un campo alfabético, se producirá un error que generara una interrupción, que por su naturaleza será de *programa*, por lo que habrá que buscar la nueva PSW en la posición 104 de memoria, la cual contiene la *dirección* de la primera instrucción de la rutina del supervisor que maneja las interrupciones de programas.

Dirección de instrucción:

El campo *dirección de instrucción* contiene la dirección de la próxima instrucción a ejecutar por la CPU. Ahora bien, su contenido depende de que sea PSW nueva, en cuyo caso contiene la dirección de comienzo de la rutina que trata la interrupción, o que se corresponda con la ejecución de un programa, en cuyo caso contiene la dirección de la próxima instrucción del programa problema que se esta ejecutando. En cualquier caso la CPU solo ejecuta la instrucción a la que apunta la dirección de la PSW ACTUAL.

Código de longitud:

Generalmente las instrucciones se ejecutan en la misma secuencia en la que se encuentran almacenadas. Las instrucciones tienen varias longitudes, de 1 , 2 o 3 medias palabras. Pues bien, uno de los campos de la PSW contienen la longitud, en medias palabras, de la instrucción que se esta interpretando. Este campo se llama *código de longitud de instrucción*.

Código de condición

Por otra parte, en los programas suelen existir puntos donde la secuencia de las instrucciones cambia. A estos cambio de secuencia se les llama bifurcaciones, y pueden ser condicionales e incondicionales.

Las bifurcaciones condicionales son aquellas que están basadas en el resultado de una operación de comparación anterior de datos. En cambio, las bifurcaciones incondicionales se producen siempre que se ejecuta una sentencia de bifurcación, y son independientes de cualquier condición.

También se produce un cambio en la secuencia de instrucciones, cuando se pasa del estado de problema a estado de supervisor, pues la ejecución de instrucciones pasara a otra dirección diferente ya que ha habido cambio de PSW. Este cambio de secuencia no es propiamente una bifurcación, pero la información de comparación generada en los pasos anteriores en el caso de una bifurcación condicional, debe guardarse hasta que se interprete la siguiente instrucción del programa que se esta ejecutando. Por esta razón, en la PSW existe un campo que guarda esta información, y que se llama *código de condición*.

Este campo refleja las condiciones existentes después de ejecutar una instrucción. No todas las instrucciones modifican el código de condición de la PSW. Este campo se encuentra en las posiciones 34-35 de la PSW y nos puede indicar :

- 1.-la relación (mayor, menor o igual) de una cantidad con otra.
- 2.-si una cantidad es mayor, menor o igual a cero.
- 3.-si una cantidad es distinta de cero.
- 4.-si el comienzo de una operación de E/S ha tenido éxito o no .

Contador de tiempos:

Otra de las funciones del supervisor es la de suministrar servicios de contador de tiempo, para lo cual se coloca en la memoria, concretamente en la posición 80 hexadecimal (128 decimal) cualquier valor de tamaño palabra. Este valor ira disminuyendo automáticamente en base a la señal de sincronismo del sistema, y cuando llegue a 0 generará una interrupción externa. Así mismo, esta función permite al programa problema pedir al supervisor información sobre el contenido del campo que señala el tiempo.

Prioridades de interrupciones:

Para conocer la prioridad con la que se tratan las interrupciones, se debe suponer que se producen al mismo tiempo los cinco tipos de interrupción descritos.

En primer lugar, hay que indicar que esto no es posible, ya que las interrupciones de supervisor y de programa son excluyentes. Esto es así porque para que se produzca una interrupción de supervisor se debe ejecutar la instrucción de programa **SVC**, y si esta instrucción es correcta, no se puede motivar un **PROGRAMA CHECK** (error de programa) pues si se ejecuta la **SVC**, al ser una instrucción de programa, es porque no se ha interrumpido el programa.

Supongamos entonces que se producen simultáneamente 4 interrupciones

- externa
- supervisor o programa (son excluyentes)
- maquina
- E/S

Las peticiones de interrupción se apilan en un orden determinado de antemano:

- maquina
- programa- supervisor
- externa
- E/S

Sin embargo, se atienden de acuerdo con otro orden también predeterminado:

- maquina
- E/S
- Externa
- Programa-supervisor

Como se ve, las interrupciones de error de maquina constituyen un caso especial, y esa razón es la que hace que aparezcan en cabeza de ambas listas. Si llegan las 4 interrupciones a la vez, la de maquina es la primera en apilarse, y la que comienza a ejecutarse, no atendiendo las otras tres. La razón es clara, pues la interrupción de maquina indica que la circuitería del ordenador está fallando, y por lo tanto, toda operación que se realice después no es fiable.

Generalmente, cuando se produce una *interrupción de maquina*, cesa de ejecutarse la instrucción en curso y se graba un tipo especial de información que posteriormente se consultará para su diagnostico y reparación, mientras que cuando se producen los otros tipos de interrupción, se atienden cuando la CPU termina la operación que esta ejecutando, y antes de que comience la siguiente.

Así pues, se reduce la petición simultánea a 3 interrupciones: E/S, P-S, y Externa, y supongamos que la PSW actual, apuntaba a la dirección 003020 en el momento de producirse las tres interrupciones.

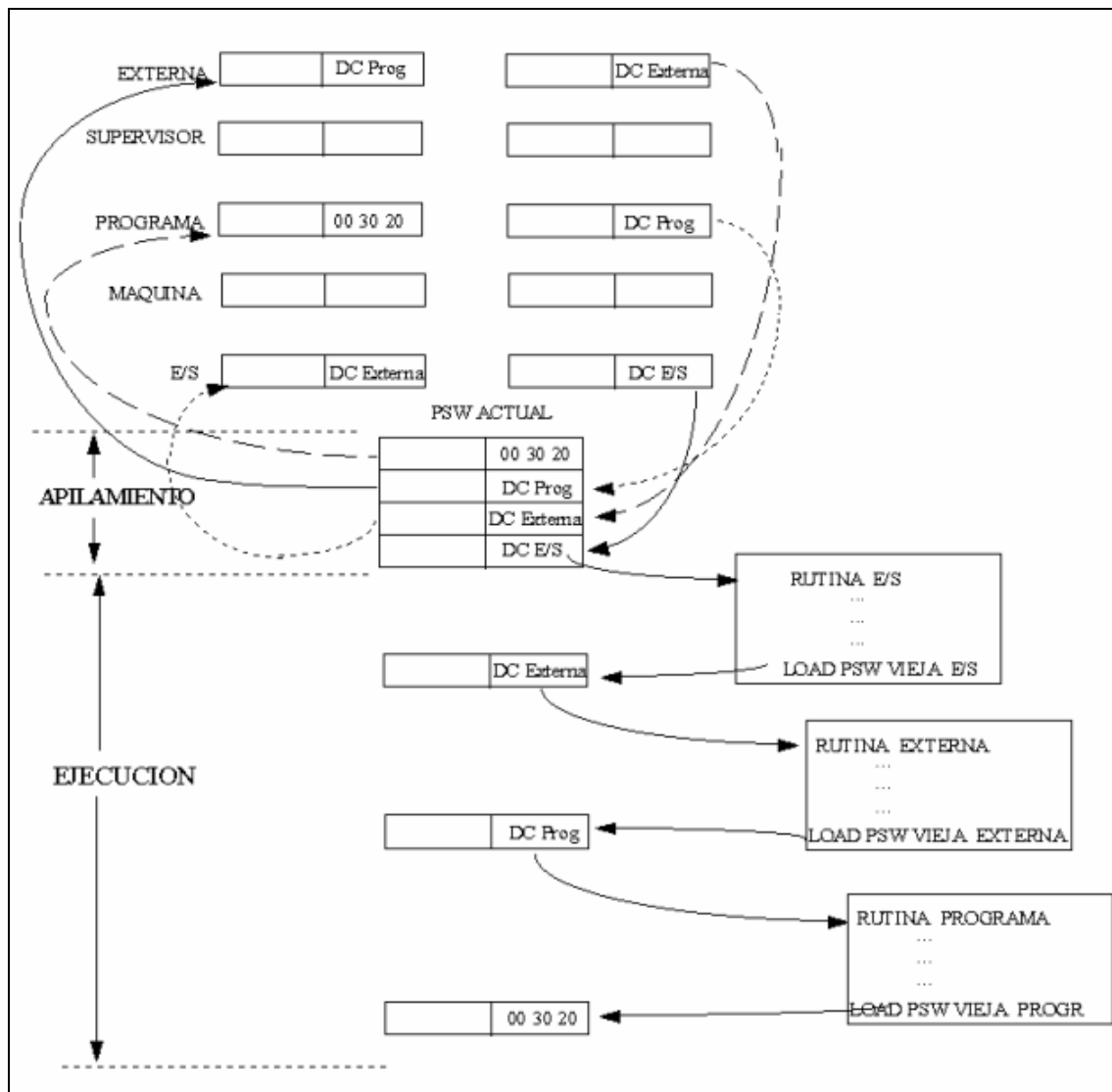


Figura 4

La primera, apilada según muestra la figura 4, es la de **programa**, por lo tanto la PSW actual pasa a PSW anterior de programa, y la PSW nueva de programa pasa a ser actual; pero como hay otra interrupción encolada, pasa a tratarse la **externa** pendiente, para lo cual ocurrirá lo mismo que con la de programa, es decir, la PSW actual pasa a PSW anterior externa, y la PSW nueva externa, pasa a actual. Pero entonces se pasa a atender a la de **E/S**, por lo que la PSW actual (que contiene la PSW nueva externa) pasa a PSW anterior de E/S, y la PSW nueva de E/S pasa a actual.

Llegados a este punto, y como no hay mas interrupciones pendientes, la CPU pasa a ejecutar las instrucciones de la rutina que trata las **interrupciones de E/S**, pues el campo de dirección de instrucción apunta a la primera instrucción de esta rutina, la cual termina cargando la PSW anterior de E/S, por lo que una vez ejecutada comienza a ejecutarse la rutina que trata **las interrupciones externas**, y así sucesivamente hasta terminar cargando la PSW anterior de interrupciones externas, que contiene la PSW nueva de **interrupción de programa**, que a su vez terminara cargando la PSW anterior de interrupción de programa que contiene la **PSW actual del programa** que se estaba ejecutando en el momento de producirse las interrupciones, por lo que proseguirá con la ejecución del programa desde donde lo había dejado.

Este baile de PSWs muestra cómo el orden de ejecución de interrupciones resulta ser el inverso al de apilamiento, pero sobre todo muestra el mecanismo diseñado para poder controlar la

ejecución de al menos dos programas de forma concurrente, tal y como se había fijado como objetivo de este artículo.