

CICS (I)

El lector debe remontarse a los años 60 para conocer las necesidades que se planteaban en aquella época y los recursos de que se disponía. Así, informáticamente hablando, solo había procesos batch. Pero una compañía aérea, concretamente la TWA, planteo un problema que no podía resolverse con los elementos existentes, razón por la que hubo que crear unas herramientas o productos nuevos, entre los que se encontró el CICS.

0.- Preliminares:

Para conocer la importancia de este tema, es necesario remontarse a finales de los años 50, ya que es en estos momentos en los que aparece un nuevo periférico: El *disco magnético*.

No se llamó '*disco*', sino DAAD⁽¹⁾ (dispositivo de almacenamiento de acceso directo) pues previamente se había creado otro tipo de discos y tambores magnéticos que recibían esta denominación. No obstante, a pesar de disponer de este nuevo periférico, durante algún tiempo la filosofía de las aplicaciones siguió siendo secuencial, y los DAAD solo se empleaban para contener las bibliotecas del sistema y como ficheros temporales de datos.

Así estaba el panorama informático de la época, cuando aparecieron unos clientes con unas necesidades de procesos nuevos, concretamente las líneas aéreas americanas (entre las que se encontraba la TWA). Estas compañías demandaban procesos ON-LINE, pues básicamente tenían 3 tipos de transacciones básicas:

- Disponibilidad de plazas.
- Venta de plazas.
- Cancelación de reservas.

Por lo tanto, los requisitos básicos que necesitaba la compañía aérea eran:

- Un fichero maestro de plazas, actualizado al instante.
- Una buena velocidad de respuesta.

Y, como el hardware y el software de la época, no permitían esto, aunque había periféricos que sí, la solución que se adoptó consistió en desarrollar un hardware y un software nuevo.

Las características solicitadas al nuevo Hardware eran:

- 1.- CPU mas rápida.
- 2.- Desarrollo de periféricos nuevos (Terminales).
- 3.- Desarrollo de todo un sistema de comunicaciones que pudiese enlazar los terminales con el ordenador. La infraestructura de las comunicaciones ya existía, pero eran comunicaciones entre terminal y terminal, no entre CPU y CPU.

La solución dada al software fue más revolucionaria, pues se buscó un programa de control que atendiera simultáneamente el máximo número de transacciones posibles. Se denominó : "*proceso concurrente de transacciones*".

La figura 1 muestra de forma comparativa las instalaciones anteriores y posteriores a este momento. En la **UC** (Unidad de Control) se encuentra el interface universal que hace posible la comunicación entre el ordenador y la impresora. Mientras que la **TCU** (Transmision Control Unit) es la unidad de control a la que llegan las líneas telefónicas que en su otro extremo tienen conectados los terminales que permiten interactuar con el ordenador central.

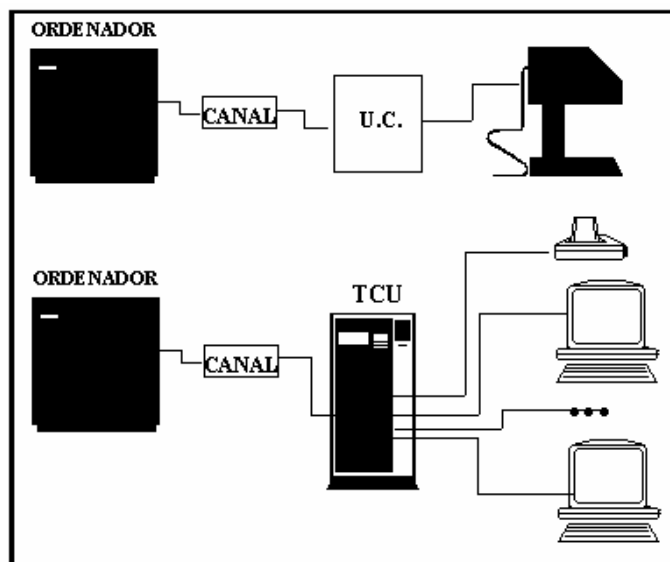


Figura 1: Instalación típica anterior y posterior a la aparición de los procesos ON-LINE

Pues bien, la idea básica de esta innovación consistía en tener residente en el ordenador un programa maestro cuya función fuera detectar, encolar y tratar las transacciones que llegaran desde los distintos terminales de la red; y, en función del código de la transacción recibida, cargar en memoria, junto a él, el programa de contabilidad, reserva de vuelo, etc., que permitiera tratarla.

De esta forma, se pretendía conseguir que los tiempos libres que deja la CPU al seleccionar una transacción se pudieran aprovechar para el proceso de una nueva transacción, o de una vieja retenida temporalmente.

Es de notar que en los procesos ON-LINE se fija un tope o umbral en tiempo de respuesta ante una transacción, no importando la cantidad de información procesada, a fin de impedir que una transacción entre en el 'abrazo de la muerte'⁽²⁾ y se adueñe de todo el sistema.

La figura 2 representa esquemáticamente la interrelación entre la CPU al ejecutar el programa de aplicación y el canal, encargado de realizar las operaciones de entrada/salida. Es importante resaltar la figura del canal, ya que la ejecución de sus instrucciones no las realiza el procesador central, por eso mientras se procesa el programa de canal, la CPU puede dedicarse a ejecutar otras instrucciones de otra región o de otra tarea que estuviera en espera.

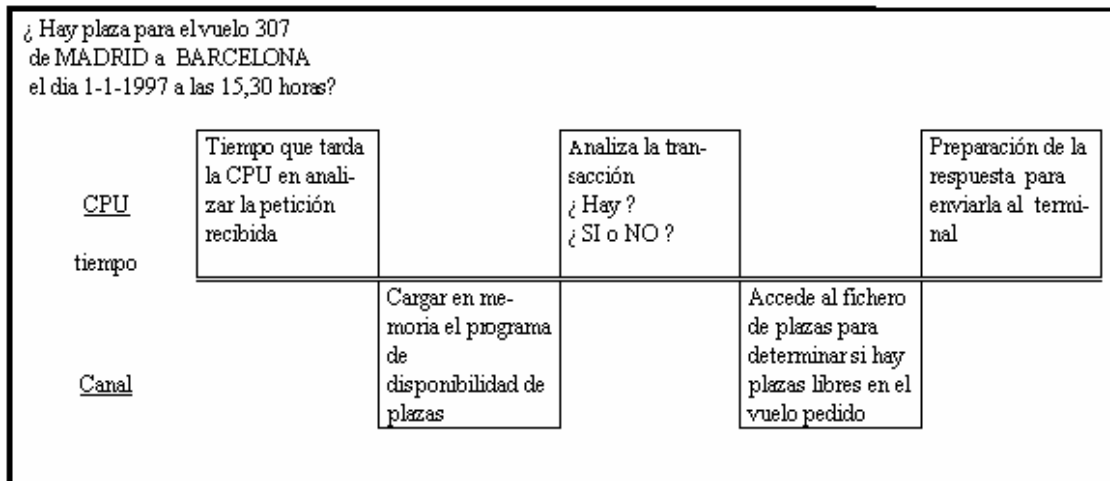


Figura 2: Esquema del reparto de trabajo ejecutado por la CPU y el canal para la ejecución de una transacción.

1.- Introducción

CICS es el acrónimo de *Customer Information Control System* o Sistema de Control de Información de Clientes.

IBM define este producto como un sistema de Base de datos/Comunicación de uso general, pues ofrece soporte para sistemas ON-LINE de forma similar a como lo hace el sistema operativo. Pero esto no debe hacernos creer que le puede sustituir, pues se ejecuta bajo el control del sistema operativo MVS (antes DOS/VS) junto con el resto de aplicaciones de la instalación, incluidos los procesos por lotes (Batch).

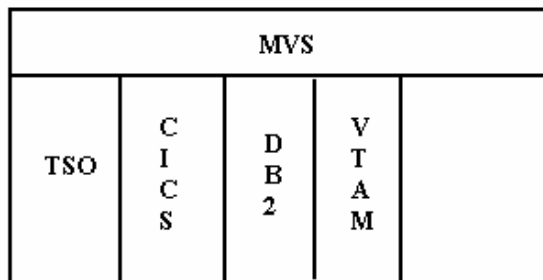


Figura 3 Regiones o espacios de direcciones del HOST

La figura 3 presenta las distintas particiones o espacios de direcciones (address space) en los que se encuentra organizado el sistema central. Así, de forma muy esquemática, se puede decir que las posiciones más bajas de memoria se encuentran ocupadas por el sistema operativo, el cual se encarga de dar servicio al resto de las regiones en las que se encuentra dividido el sistema. Cada una de estas regiones constituye un subsistema, de modo que en una puede correr el CICS, en otra DB2, en otra un núcleo ADABAS, etc...

De esta forma, cuando CICS quiere un dato de un fichero, si le tiene asignado a su entorno, invoca al S.O. para que sea este quien invoque al canal y obtenga una copia de dicho registro en memoria. Pero, si quiere un dato de ADABAS, le pide al S.O. que le pida a ADABAS dicho dato, siendo ADABAS quien se encarga de llamar al S.O. para leer dicho dato de los ficheros físicos en los que guarda la información por él gestionada, y así sucesivamente.

Otro componente que debe ser mencionado para conocer el funcionamiento de este producto, es el **VTAM** (Virtual Telecommunications Access Method), componente encargado de proporcionar los servicios necesarios para el control de transmisión: Acepta las peticiones de inicio de comunicación, solicita la conexión de terminales para la iniciación automática de tareas, etc. En una palabra, se podría decir que es el responsable de las comunicaciones. Corre en el procesador principal, en un espacio de direcciones propio, y no debe confundirse con el **NCP** (Network Control Program) que es el programa encargado igualmente de las comunicaciones, pero que corre en el controlador de comunicaciones. (3)

El VTAM permite combinar dispositivos por lotes e interactivos en la misma línea de transmisión, sin degradación de la respuesta interactiva. Los recursos del VTAM no están sujetos a limitaciones de tiempos para el programador de aplicaciones.

Una característica importante de CICS es que el diseñador del sistema no tiene que tener en cuenta las características del terminal desde el que se va a correr la aplicación, punto este muy importante en la época inicial, aunque hoy, por evidente, casi impensable.

2.- Procesos Batch y Procesos On-line

Es curioso ver como evolucionan los medios. Tal y como se ha comentado en los puntos anteriores, en sus comienzos solo había procesos BATCH o procesos por lotes, donde el ordenador se dedicaba íntegramente a un único programa, el cual trataba los datos, que solían ser ficheros voluminosos. y generaban listados también voluminosos que debían ser distribuidos a todas las oficinas, pues era el medio de consulta habitual de los resultados de estos procesos.

Hoy, evidentemente siguen existiendo los procesos anteriores, pero los resultados ya no se presentan en papel, a no ser por petición expresa, sino que son consultados desde el propio terminal.

Por otra parte, las grandes instalaciones tienden a planificar la ejecución de los procesos batch fuera del horario de oficinas para que no compitan en el consumo de recursos de ordenador con los procesos ON-LINE en los que se busca procesar un único detalle.

Esto quiere decir que los procesos on-line permiten procesar programas muy cortos que tratan un único registro y que por tanto generan una contestación casi inmediata hacia el terminal que ha solicitado su ejecución. Un claro ejemplo de este tipo de procesos puede ser la reserva de un vuelo, o la disposición de dinero en un cajero automático.

En una primera aproximación, se podrían clasificar los distintos tipos de procesos ON-LINE en los siguientes:

- Sistemas Interactivos para desarrollo y prueba de programas, tales como el TSO, ICCF.

- Sistemas DB/DC (Data Base/Data Communication). Estos sistemas permiten, con herramientas propias, construir programas y aplicaciones, y a su vez, proporciona facilidades para el control de estas aplicaciones. Entre estos sistemas se encuentra el CICS y el IMS.

- Sistemas de Control: Son aquellos procesos que tienen por finalidad controlar procesos de fabricación específicos.

3.- Entorno del CICS:

La figura 4 muestra el entorno de un sistema CICS. En ella se puede apreciar cómo este subsistema cuenta con:

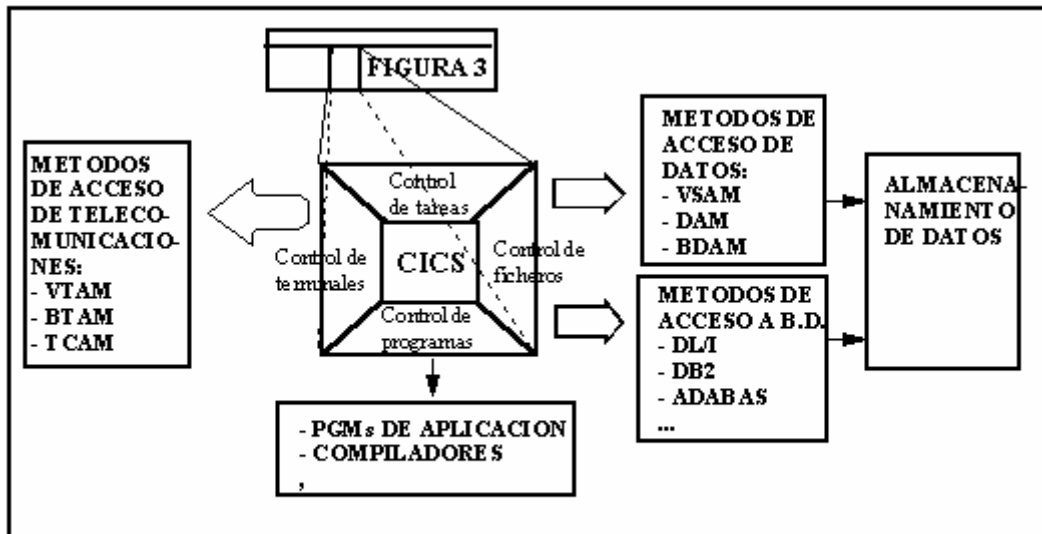


Figura 4 : Entorno general del CICS

- Servicios de control propios que le permiten estar unido al sistema operativo para la utilización de los recursos que no son propiamente suyos.

- Funciones de manejo de datos, lo cual le permite aceptar, desde los programas de aplicación, peticiones de escritura o lectura sobre cualquier tipo de fichero, preparando las peticiones para que sean procesadas por el método de acceso apropiado.

- Funciones de comunicación que le permiten realizar el control de terminales. Esta parte del CICS trata el sondeo y direccionamiento de todos los terminales y la transferencia de datos 'a' y 'desde' ellos.

- Servicios de programas de aplicación, los cuales le permiten gobernar los programas asociados a cada aplicación dentro de la configuración propia del CICS.

4.- Características principales:

De entre las muchas características que se podrían enunciar, se pueden resaltar en una primera aproximación las siguientes:

- De cara al MVS representa un solo espacio de direcciones (Address space), tal y como se desprende de la figura 3.

- Se ejecutan todos los programas a la vez, solapándose las interrupciones generadas por las operaciones de E/S de una tareas con el proceso de otra petición, tal y como se puede deducir del estudio de la figura 2. Pero, es el propio CICS quien asigna ciclos de CPU a otra tarea cuando se solicita una lectura por parte de la transacción que se esta ejecutando.

- Así mismo, dispone de su propio *despachador*⁽⁴⁾ de tareas, asignando prioridades a cada una. No utiliza la gestión del *despachador* del MVS en sus tareas, ya que desde el punto de vista del MVS, el propio CICS es una única tarea.

- Realmente se ejecuta un solo programa (DFHSIP)⁽⁵⁾, y todos los programas de usuario y del sistema son considerados subprogramas respecto a él.

- Los programas de usuario no están *linkeditados* ⁽⁶⁾ con el programa principal, ya que el CICS tiene su propio cargador de programas.

- Desde un programa de aplicación, se tiene acceso a cualquier área de otro programa o de módulos básicos de CICS,⁽⁷⁾ lo que da lugar a la posibilidad de corrupción de áreas (violaciones de memoria).

5.- Conceptos importantes:

Transacción: Básicamente cada una de las peticiones que se recibe desde un terminal, constituye una transacción, la cual crea una unidad de trabajo o TAREA para la CPU. Esquemáticamente se podría decir que una transacción consiste en '*hacer una petición al host*'⁽⁸⁾, este ejecuta la petición, y devuelve los resultados al terminal.

Tarea: Es la unidad de proceso interno para realizar una transacción. Siempre estará en uno de los dos estados siguientes:

- ACTIVA: se esta ejecutando.
- SUSPENDIDA: Esta esperando algo del sistema, por ejemplo que el sistema lea un registro de un fichero.

Multitarea: Múltiples aplicaciones se ejecutan al mismo tiempo en la misma región o espacio de memoria (ver figura 3).

Reentrabilidad: Un programa se dice que es:

- **Reentrante:** cuando solo hay un único modulo cargado en memoria ⁽⁹⁾, de modo que cuando una tarea termina de ejecutarle, puede ser usado el mismo modulo por otra. Como es lógico suponer, los módulos reentrantes tiene una única '*working*'⁽¹⁰⁾.

- **Cuasi-reentrante:** Este tipo de programa es '*serialmente reusable entre puntos de entrada y salida*'. es decir, CICS asigna workings distintas a cada tarea que quiera usar dicho modulo concurrentemente, y así utiliza un único ejecutable. Ahora bien, solo es reusable en serie (por varios) entre puntos de E/S, tal y como muestra la figura 5. CICS es cuasi-reentrante, y esto no tiene nada que ver con el concepto conversacional que se trata a continuación.

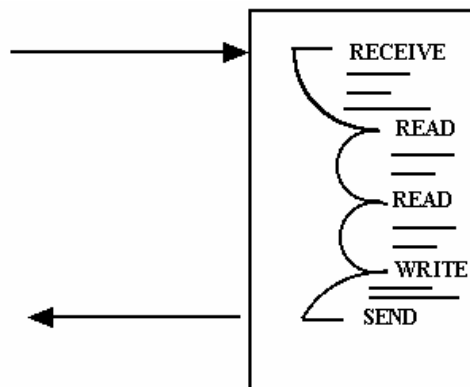


FIGURA 5 : MODULO CUASI-REENTRANTE

Conversacionalidad:

Un programa se dice que es **conversacional** cuando cuenta con instrucciones para aceptar y mostrar datos, realizándose estas operaciones de entrada/salida tantas veces como lo necesite su ejecución para poder llegar al final y presentar los resultados solicitados (ver figura 6). Este tipo de programas como es fácil imaginar inmoviliza áreas de memoria en las que se van almacenando los resultados intermedios.

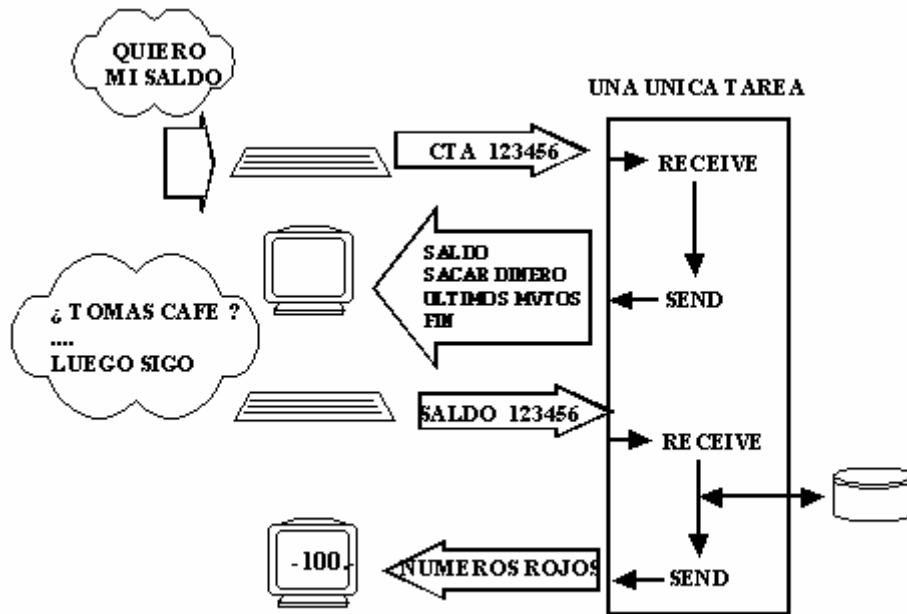


Figura 6 Transaccion conversacional

Un programa se dice que es Pseudoconversacional cuando consta de una única instrucción para aceptar datos y una única instrucción para presentar datos, cediendo el control al CICS, de forma que la siguiente entrada de operador arranque una nueva tarea que ejecute otra transacción. Este tipo de programas inmoviliza menos áreas de memoria que el tipo conversacional, y para su ejecución total, requiere tantas tareas CICS como entradas haya que considerar. La ventaja es que mientras el operador 'se piensa la respuesta', en el Ordenador central no existe tarea, pudiendo dedicarse esos recursos a otras tareas. La figura 7 muestra de una forma esquemática estos extremos.

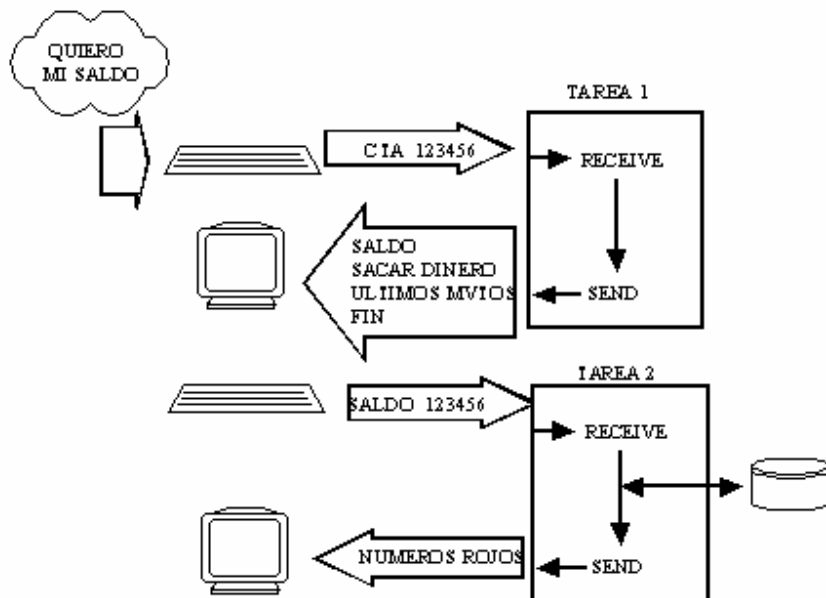


Figura 7 : Transaccion Pseudo-conversacional

6.- NATURAL:

En artículos anteriores, esta sección ha hablado entre otras cosas de NATURAL. El hecho de hacer mención a este producto hoy, solo es para proporcionar un claro ejemplo de lo que realmente es CICS. En aquellos artículos se decía que NATURAL era un entorno, es decir un *MACROPROGRAMA* ⁽¹¹⁾ que permitía editar, crear comandos de usuario (programas), etc. Pues bien, en el caso de que NATURAL corra bajo CICS, NATURAL no es mas que una transacción CICS.

Así, NATURAL, como tal transacción CICS, se compone de una serie de rutinas que son invocadas por los servicios de control de programas del CICS, y que sirven para cargar y ejecutar los distintos ejecutables de los comandos NATURAL creados por el programador, exactamente igual que la transacción que solicita el saldo desde un cajero, y que se encuentra escrita en COBOL.

7.- Historia:

Y ya que este artículo comenzó con algo de historia, se va a cerrar también con una pequeña comparativa de su evolución:

- Tiene algo mas de 20 años.
- Fue diseñado en sus orígenes por una compañía eléctrica, (aunque el autor no ha podido confirmar este extremo), si bien IBM tuteló este proyecto casi desde sus comienzos.
- Al principio era configurable, es decir cargaba o seleccionaba solo los módulos con los que podía trabajar.
- contaba con una tabla de recursos
- era excesivamente débil, (se caía muy fácilmente)
- La recuperación era deficiente.

En la actualidad:

- es monobloque
- cuenta con un fichero de recursos, el cual sustituye a las tablas que se verán en el próximo artículo.
- Cuenta con un interface de programación en comandos
- Todavía es posible la corrupción de áreas comentada anteriormente.
- La recuperación esta ya muy perfeccionada
- *Es un producto estratégico de IBM para Grandes Sistemas.*

8.- Próximo mes:

Como el espacio no perdona, se deja para el próximo mes, la descripción de los distintos componentes del CICS, esperando con ello que el lector se haga una clara idea de lo que es este producto tan comentado en el entorno del los Grandes Sistemas..

Notas:

- (1) Este tipo de dispositivo ya se describió en SOLO PROGRAMADORES 19, en el primer artículo dedicado a los ficheros.
- (2) Se denomina '*abrazo de la muerte*' al bucle o estructura repetitiva que no tiene salida o tiene infinitas iteraciones, y por consiguiente, no permite que se pueda abortar su ejecución ya que no cede el control al sistema operativo y se hace dueña de la máquina.
- (3) No entra en el propósito de este artículo una explicación más detallada del VTAM, por eso simplemente el autor quiere dejar constancia de que estos productos se encuentran íntimamente relacionados con el tema tratado, y aplaza para un momento posterior el tratamiento de este tema.
- (4) Este anglicismo proviene del término inglés '*dispatcher*', y que es la denominación que recibe la rutina de MVS encargada de determinar qué trabajo, de entre todos los que están en la cola, se ejecutará a continuación.
- (5) Todos los objetos relacionados con el CICS comienzan por DFH.
- (6) Después de compilar un programa debe linkeditarse a fin de adaptar la traducción realizada por el compilador a las macros que se encargan de invocar a las rutinas del S.O.
- (7) Cualquier programa puede moverse en su región o espacio de direcciones sin problemas. Pero como los programas de usuario son considerados como subprogramas del CICS, pueden tocar direcciones de memoria en las que se encuentran cargados los módulos del CICS dando lugar a las corrupciones de memoria citadas.
- (8) El ordenador central suele recibir la denominación inglesa de HOST u ordenador principal.
- (9) Recordar que como en cualquier sistema solo se ejecutan programas que estén cargados en memoria principal.
- (10) La '*Working*' es un área de memoria principal que se reserva para contener la información correspondiente a todas las variables locales definidas en un módulo. Esta denominación tiene su origen en el hecho de que en COBOL todas estas variables se encuentran definidas en la sección WORKING-STORAGE SECTION.
- (11) Valga esta definición solo para decir que es un programa muy grande, no para crear un nuevo tipo de programa.

¹ Este tipo de dispositivo ya se describió en SOLO PROGRAMADORES 19, en el primer artículo dedicado a los ficheros.

² Se denomina 'abrazo de la muerte' al bucle o estructura repetitiva que no tiene salida o tiene infinitas iteraciones, y por consiguiente, no permite que se pueda abortar su ejecución ya que no cede el control al sistema operativo y se hace dueña de la máquina.

³ No entra en el propósito de este artículo una explicación más detallada del VTAM, por eso simplemente el autor quiere dejar constancia de que estos productos se encuentran íntimamente relacionados con el tema tratado, y aplaza para un momento posterior el tratamiento de este tema.

⁴ Este anglicismo proviene del término inglés 'dispatcher', y que es la denominación que recibe la rutina de MVS encargada de determinar qué trabajo, de entre todos los que están en la cola, se ejecutará a continuación.

⁵ Todos los objetos relacionados con el CICS comienzan por DFH.

⁶ Después de compilar un programa debe linkeditarse a fin de adaptar la traducción realizada por el compilador a las macros que se encargaran de invocar a las rutinas del S.O.

⁷ Cualquier programa puede moverse en su región o espacio de direcciones sin problemas. Pero como los programas de usuario son considerados como subprogramas del CICS, pueden tocar direcciones de memoria en las que se encuentran cargados los módulos del CICS dando lugar a las corrupciones de memoria citadas.

⁸ El ordenador central suele recibir la denominación inglesa de HOST u ordenador principal.

⁹ Recordar que como en cualquier sistema solo se ejecutan programas que estén cargados en memoria principal.

¹⁰ La 'Working' es un área de memoria principal que se reserva para contener la información correspondiente a todas las variables locales definidas en un módulo. Esta denominación tiene su origen en el hecho de que en COBOL todas estas variables se encuentran definidas en la sección WORKING-STORAGE SECTION.

¹¹ Valga esta definición solo para decir que es un programa muy grande, no para crear un nuevo tipo de programa.