

Ficheros en MVS (II)

Continuando con el tema iniciado el pasado mes, hoy le toca el turno a los métodos de almacenamiento y recuperación de la información en ficheros con organización directa y particionada. Este artículo se complementa con una utilidad que permite intercambiar la información entre los distintos tipos de organización que se estudian.

1.- Introducción:

En la entrega anterior fueron enumerados los métodos de almacenamiento y acceso de información bajo el S.O. MVS de IBM. Se vio la organización Secuencial (SAM) y la Secuencial-Indexada (ISAM). El presente artículo estudia el resto de métodos usados para el almacenamiento de la información. La organización directa, método muy usado por los SGDB (Sistemas Gestores de Bases de Datos; o DBMS Data Base Management System); la organización particionada, posiblemente la mas usada en las grandes instalaciones; y los Ficheros VSAM muy usados en todas las instalaciones antes de que se implantara algún gestor como DB2 y/o ADABAS, y que siguen manteniendo totalmente su vigencia.

2.- Organización Directa

Este tipo de organización obtiene la dirección física del registro en el dispositivo a partir de la información contenida en el propio registro, para lo cual basta aplicar un algoritmo matemático a la clave para obtener dicha dirección.

Las direcciones obtenidas pueden ser de tres tipos:

- Direcciones relativas de registros (DRR).
- Direcciones relativas de pistas (DRP).
- Direcciones absolutas.

Es decir, que suponiendo que el campo clave de un fichero fuera el DNI, aplicando un algoritmo al valor concreto de cada DNI del fichero, se obtendría la dirección física en la que almacenar cada registro dentro del espacio físico asignado al fichero.

Este tipo de organización se puede definir por la existencia de una correspondencia entre la clave de cada registro y su dirección en el dispositivo, por eso es usado habitualmente por los SGBD para el almacenamiento de los datos.

Función de direccionamiento: Se denomina así a la función que establece la correspondencia entre las claves y sus direcciones en el dispositivo.

2.1.- DRR : Direcciones relativas de registro

En este tipo de direccionamiento, se considera el espacio asignado al fichero dividido en registros, estando estos numerados consecutivamente, tal y como muestra el ejemplo de la **figura 1**.

Así, Si el archivo tiene n registros, y en cada pista pueden caber 5 registros, la dirección relativa de registro dentro de la pista irá desde la 0 hasta la 4.

Reg Pista	0	1	2	3	4
0	0	1	2	3	4
1	5	6	7	8	9
2	10	11	12	13	14
3	15	16	17	18	19
...					

Figura 1: Determinar la DRR

2.2.- DRP : Direcciones relativas de pista

Para este direccionamiento, solo se consideran numeradas consecutivamente las pistas del espacio asignado al fichero, por lo que la función de direccionamiento proporciona solo el número de pista en el que ubicar el registro.

Existen dos posibilidades para almacenar los datos cuando se usa este tipo de direccionamiento:

- 1.- Escribir el registro en la primera posición libre de la pista. En este caso, para poder recuperarlo hay que proporcionar la clave además de la DRP.
- 2.- Proporcionar además de la DRP el número de registro dentro de la pista. En este caso, conocidos ambos datos, se puede obtener directamente la DRR, bastando para ello con una multiplicación y una suma, tal y como muestra el siguiente ejemplo: Se desea conocer la DRR de un registro sabiendo que la DRP es 2 y el número de registro es el 4 y que se pueden almacenar 5 registros por pista. El resultado sería: $2 \cdot 5 = 10$ y $10 + 4 = 14$ por lo que la DRR pedida es 14, pudiéndose comprobar este resultado en la tabla de la figura 1.

2.3.- Direcciones absolutas:

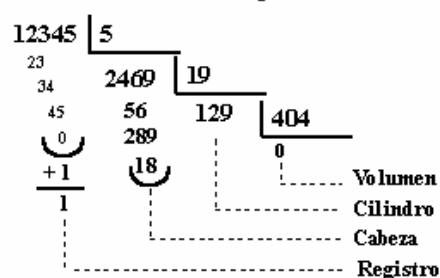
Este direccionamiento necesita un campo de 8 octetos en binario puro para poder recoger la siguiente información:

- 1 octeto para el número de volumen
- 2 octetos a ceros binarios
- 2 octetos para el número de cilindro
- 2 octetos para el número de cabeza
- 1 octeto para el número de registro.

La dirección absoluta se obtiene a partir de la DRR por divisiones sucesivas.

Ejemplo: La **figura 2** muestra el cálculo de la DRP y la dirección absoluta en un disco 3330 de IBM suponiendo que se haya definido 5 registros por pista y sabiendo que la DRR del registro a calcular es 12345

Calculo del DRP correspondiente al DRR 12345



DIRECCION ABSOLUTA:

0-0	0-0,0-0	0-0,8-1	0-0,1-2	0-1
Volumen	Cilindro	Cabeza	Reg.	

FIGURA - 2

Claves

Las claves en este direccionamiento deberán ser siempre numéricas, y, en el caso de que una clave fuera alfanumérica, será necesario aplicarla una transformación, siendo el método más usual el de las tablas de conversión.

Rango de claves: Es el máximo número de claves que se pueden tener. Se obtiene por diferencia entre la clave más alta y la más baja, siendo un caso habitual que parte de las claves que están incluidas en el rango no se utilicen. Esto quiere decir que el rango es mucho mayor que los números de claves utilizados.

Otro concepto importante es la distribución de las claves que se van a utilizar. Esta distribución es uniforme si la proporción de claves utilizadas es igual en cualquier parte del rango; en caso contrario se dice que no es uniforme, siendo ésta la más normal.

Sinónimos:

Son registros de claves diferentes pero que al aplicarles la función de direccionamiento resulta la misma dirección. Este problema no surge únicamente con el método de división-resto visto anteriormente, sino que surge con cualquier método que se use.

Los sinónimos en sí mismos pueden no ser un problema, dependiendo del tipo de direcciones que se generen. Así, si se esta generando DRR, todos los registros que tengan la misma dirección serán sinónimos, y a la vez excedentes, porque todos los registros sinónimos no se pueden grabar en la misma dirección.

Pero, si se estuviera generando DRP, todos los registros que pertenezcan a la misma pista serán sinónimos, pero no habrá excedentes hasta que se llene la pista.

Una función de direccionamiento será peor o mejor, dependiendo de que genere mayor o menor numero de excedentes. En el caso de que se generen muchos excedentes, lo que se suele hacer es ampliar el número de direcciones de un fichero, dando mas espacio del que realmente se necesita. Llevando al límite esta posibilidad se podría reservar espacio para todas las direcciones del rango, lo cual conduciría a una infrautilización del fichero.

Factor de empaquetamiento: Una medida de esta técnica es lo que se denomina factor de empaquetamiento, definiéndose como el cociente que resulta al dividir el número de registros utilizados por el número de registros reservados. Lo normal es usar factores de empaquetamiento entre el 80 y 85 %.

Formato de los registros: La organización directa admite todos los tipos de formatos, sin embargo, los registros son considerados siempre como desbloqueados. Por tanto, si se desea bloquear registros, será el propio informático el que se plantee y resuelva el problema, generando bloques y asignándoles una clave a la que aplicar el algoritmo de direccionamiento. Esto último es lo que realizan los distintos gestores de bases de datos que utilizan este tipo de organización.

En este sentido, cabe recordar el hecho comentado al tratar el tema de DB2, en el que el espacio físico asignado al fichero (tabla) se encontraba dividido en páginas, pudiendo contener cada una de estas paginas varios registros lógicos. Pues bien, el hecho que se desea resaltar es que si bien el registro lógico, desde el punto de vista del usuario , es el registro de *clientes*, por ejemplo, desde el punto de vista del SGDB, el registro lógico esta constituido por una página, no pudiendo aplicarse el concepto de bloqueo visto en el articulo anterior, por ser cada pagina un único registro lógico para el gestor.

Procesos:

Carga: Es siempre al azar. Se produce progresivamente en el tiempo.

Adiciones: Son siempre al azar, e iguales al proceso de la carga. Al aumentar el número de registros se incrementa el *factor de bloqueo*

Recuperación: también se realiza mediante proceso al azar. Al aumentar el número de excedentes disminuye la eficacia.

3.- Organización Particionada:

Este tipo de organización nace con el sistema operativo DOS/VS, siendo junto a la secuencial las organizaciones mas usadas en una gran instalación de MVS en las labores cotidianas.

El nombre de esta organización se debe a que tiene la información dividida en dos partes o áreas:

- Área de directorio
- Área de miembros.

Donde: el Área de directorio hace las veces de un diccionario en el que se encuentran referenciados cada uno de los miembros del fichero, pues contiene los nombres y direcciones de cada uno de ellos; y, el área de miembros recoge la información de interés para el usuario.

Un miembro es el equivalente de un fichero secuencial, con la salvedad de que el conjunto formado por todos estos ficheros secuenciales configuran un único fichero. Pero, como las Características se definen a nivel de fichero, todos los miembros tendrán la misma longitud de registro, formato, etc.

Se podría equiparar esta organización a la de una biblioteca en la que el área de directorio representa el índice, y el área de miembros estaría representando a cada uno de los libros físicos de la biblioteca, eso si, todos los libros deberían tener el mismo formato o longitud de registro, aunque como es lógico, todos tendrían distinto contenido, autor, numero de paginas, etc.,

Directorio:

Es un conjunto de registros lógicos agrupados en bloques de 256 bytes, lo cual quiere decir que en un bloque deberán grabarse el máximo numero de registros lógicos enteros que quepan.

En todo bloque, el primer octeto contiene el número de octetos ocupados del bloque.

Los registros lógicos pueden contener información opcional, por lo que su longitud varía desde 12 octetos como mínimo, hasta 74 octetos como máximo. En los primeros doce octetos viene el nombre del miembro, y la dirección del mismo en el área de miembros.

Todos los registros en el área de directorio se encuentran clasificados alfabéticamente, tal y como muestra la **figura 3**.

Miembros:

Los miembros son ficheros que no tienen relación alguna con el resto de miembros del fichero, es decir son independientes entre sí. Se encuentran organizados consecutivamente, lo cual quiere decir que la organización es compacta y que por tanto no hay huecos libres entre los miembros.

Admite cualquier tipo de formato, de pistas y de registros.

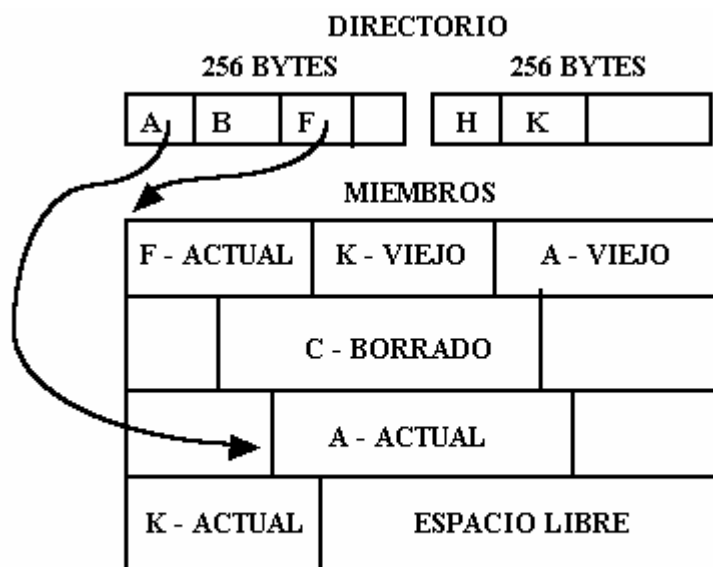


FIGURA 3 : Areas de la organizacion particionada

Para recuperar un miembro, se debe buscar el nombre del mismo en el área de directorio del fichero; y una vez obtenida la dirección del mismo, recuperar secuencialmente la información contenida en el mismo.

Para adicionar miembros, el propio sistema genera áreas adicionales y reorganiza el directorio para colocar alfabéticamente el nombre y la dirección de los nuevos miembros en el lugar que les corresponda. .

Los miembros que se dan de baja, no desaparecen físicamente del directorio en el momento de darle de baja, sino que el miembro permanece hasta que se reorganice físicamente el área de miembros.

Cuando el área del directorio se llena, no se pueden añadir nuevos miembros por falta de sitio en el directorio, siendo necesario reorganizar toda la librería, tanto de miembros como directorio.

Particularidades de esta organización:

- Como ya se ha dicho, el fichero está formado por el conjunto de las dos áreas citadas, por lo que a la hora de asignar recursos para un trabajo (ALOCAR), si ya existe el fichero, se debe poner disposición OLD o SHARE, aunque lo que se desee sea crear un miembro, tal y como se puede ver en el JCL que se encuentra en el disquete que acompaña a la revista.

- El concepto de reorganización citado anteriormente consiste en eliminar físicamente los miembros dados de baja del área de miembros, ya que como se muestra en la figura 3, los miembros dados de baja o modificados se van almacenando consecutivamente en el área de datos, pudiendo llegar a llenar el área que tiene asignada.

4.- Organización VSAM (Virtual Storage Access Method)

El concepto de almacenamiento virtual consiste en almacenar en disco los datos o programas que se encuentran en memoria real, tratando este almacenamiento en disco como si se tratara de memoria real, con lo que se obtienen muchas mas direcciones de memoria. Ahora bien, como desde esta memoria auxiliar no se puede ejecutar programas, el sistema dispone de un sistema de paginación para poder pasar el contenido desde memoria virtual a real y así ejecutar el programa.

Pues bien, VSAM es un método de acceso apropiado para sistemas operativos con memoria virtual.

Esta organización es parecida a la organización secuencial indexada comentada en el artículo anterior, con la diferencia de que en este tipo de ficheros, no se habla de los términos de cilindro, cabeza o pista para referenciar un registro, sino que se habla de RBA (Relative Byte Address). Tampoco se habla de Bloque, sino de IC (intervalo de control), donde este contiene no solo la información de 'n' registros, sino que además contienen otra información de control llamada RDF.

VSAM es una organización de datos que admite el acceso a los registros de forma secuencial y al azar. Los registros pueden ser de longitud fija o variable, aunque VSAM esta pensado para registros de longitud variable. El bloqueo o desbloqueo de registros es controlado por el propio VSAM aprovechando al máximo el espacio del dispositivo ya que es el propio sistema el que calcula el factor de bloqueo.

Comparando VSAM con los otros métodos de acceso, podremos apreciar las siguientes ventajas:

- Mejores resultados
- Mayor seguridad e integridad de los datos
- Mayor facilidad de uso y control
- Alta velocidad de recuperación y de almacenamiento de datos
- Tres organizaciones en una, pues puede procesar los archivos organizados de tres formas distintas:

Organización de los ficheros VSAM:

Existen tres tipos de ficheros según la secuencia en que se graben los registros:

- ESDS (Entry Sequence Data Set): Equivalentes a ficheros secuenciales
- KSDS (Key Sequence Data Set): equivalentes a ficheros de organización secuencial indexada
- RRDS (Relative Records Data Set): equivalentes a ficheros con organización directa.

Los ficheros KSDS deben llevar clave, pero los ficheros ESDS no necesitan llevar clave,

5.- Fichero GDG

Estos son ficheros secuenciales normales que solo tienen de especial el nombre ya que bajo un mismo nombre se pueden referenciar varios ficheros.

La definición de un fichero GDG consta de dos pasos: En el primero se crea lo que es el nombre del fichero en el catálogo. La **figura 4** muestra un paso de JCL para realizar este cometido. Como puede verse en él, no se especifican características, como ocurre cuando se crea cualquier fichero, ya que solo es una entrada en el catálogo de usuario, no en el VTOC pues todavía no existe físicamente.

Una vez creado el nombre, cuando se desee añadir una versión a ese fichero, se realizará lo siguiente:

- Añadir el sufijo (+1) al nombre
- Especificar las características de este fichero secuencial, que no tienen por que coincidir con las otras versiones,
- Poner como disposición inicial NEW.

En cambio, cuando se desee leer una versión habrá que especificar como nombre, el nombre del GDG y el sufijo (0) para la última versión, (-1) para la penúltima, etc. ; y como disposición inicial hay que especificar SHR (Shared:compartido)

6.- Acceso y recuperación de datos

El acceso a datos en MVS se hace a través del juego de VTOC y catálogos.

6.1.- VTOC (Tabla de contenidos del Volumen o Volume Table of Contents)

En MVS todos los volúmenes tienen una zona en la que se registra toda la información física que contiene, así como la organización de los ficheros que están en ese volumen.

Esta área, que recibe el nombre VTOC se crea en el momento de inicializar el volumen, y esta formada por entradas, una por fichero (DSCB: Data Set Control Block) con una longitud de 148 octetos.

Es en estas DSCBs donde se registra toda la información disponible sobre fichero, acerca de su estructura física (formato de registro, organización, longitud de bloque, etc.)

6.2.- Catálogos:

Los catálogos son el otro componente en el que se apoya MVS para el acceso y localización de ficheros.

Un catálogo no es más que un fichero VSAM, pero con un contenido especial ya que contiene información acerca de la ubicación física de los ficheros, de modo que se pueda localizar cualquier fichero de la instalación solo con esa información.

En MVS un fichero puede o no estar catalogado. En el caso de estarlo, el sistema, a través del catálogo puede determinar el volumen en el que se encuentra, y una vez en el volumen, examinando la VTOC podrá recoger todas las Características de este fichero. En el caso de no estar catalogado, cuando se desee utilizar, habrá que especificar la información del volumen que le contiene, para así, mirando la VTOC de dicho volumen recoger sus Características, y la dirección de comienzo.

Para poder gestionar la catalogación de todos los ficheros de la instalación, MVS mantiene un catálogo maestro y varios catálogos de usuario.

En el catálogo maestro, normalmente figuraran estradas maestras, es decir referencias de los ficheros que son vitales para el funcionamiento de MVS, y sus alias.

En los catálogos de usuario figurará el volumen donde se encuentran los distintos ficheros de usuario.

7.- Próxima entrega ...

El artículo del próximo mes se dedicará a tratar los **ficheros VSAM**, cerrando con él la serie dedicada a este tema.

8.- Ejemplo:

En el disquete que acompaña a la revista se incluye una utilidad completa para poder definir un fichero GDG a la vez que sirve de ejemplo para el manejo de ficheros secuenciales, particionados y generacionales o GDGs. Todos los ficheros que se incluyen se encuentran en ASCII.

La utilidad esta contenida en los siguientes ficheros:

-CLIST.GDG : CLIST que pide el nombre del GDG a generar mediante un PANEL; realiza las sustituciones pertinentes a un esqueleto de JCL encargado de generar dicho GDG; y submite dicho JCL.

Para poder ejecutar esta clist recuerdo lo dicho en el artículo titulado CREACION DE UN MENU DE DESARROLLADOR (*Solo PROGRAMADORES NUM. 13 - SEPTIEMBRE'95*), en el sentido de que previamente se deben alocar las librerías correspondientes a Paneles, Clist y Esqueletos.

-PANEL.GDG : Definición del panel ISPF usado por la clist anterior.

-ESKELETO.GDG: Esqueleto de JCL usado por la clist.

-CREA_GDG.LOG : JCL submitido, copiado directamente desde el spool.

-CREA_GDG.HC : Es un Hard-copy en tiempo de ejecución de la utilidad comentada.

Los siguientes ficheros solo sirven de ejemplo en el manejo de los tipos de ficheros citados:

-PUT_GDG.JCL : contiene un JCL que sirve para cargar 5 versiones del fichero GDG con la información contenida en los ficheros que se comentan en las líneas previas de cada paso. Así mismo, sirve para mostrar que cada una de las versiones de un GDG tiene características distintas.

-IBM_HC.34S : Es un hardcopy del menu de utilidades de ISPF que muestra las Características del fichero JMPDES.IBM.HC copiado en el primer paso en el que se han incluido los hardcopys que se muestran en este artículo. Como se ve, el formato de los registros de este fichero son VB (Variable Bloqueado).

-CAPTURAR.LOG : La salida del SPOOL conteniendo los mensajes generados en la ejecución del JCL puede pasarse a un fichero para ser tratada como cualquier otra información de usuario. Este paso a fichero se hace escribiendo, en la línea correspondiente a la ejecución que se desea pasar a fichero, el comando **XD** con el cual se especifica al sistema que copie todo el log de la ejecución en el fichero que se especifica en el panel que aparecerá a continuación de pulsar INTRO una vez escrito dicho comando. Pues bien, el contenido de **CAPTURAR.LOG** no es el log de la ejecución del JCL, sino los Hardcopys que muestran el procedimiento descrito, y que sirve para mostrar las Características del fichero JMPDES.JMP.SALIDA, entre las que se encuentra la longitud de registro de 133 bytes y formato FBA .

-VERSION.GDG : Hardcopy que muestra los nombres dados por el sistema a las distintas versiones.

-PARTICION.JCL : Este fichero contiene un JCL que sirve de ejemplo para el tratamiento de los ficheros particionados. Así, en el primer paso, se borra el fichero particionado a crear, por si acaso existe. En el segundo paso, se crea un fichero particionado, reservando para ello el espacio correspondiente, y asignando una longitud de registro de 133 octetos. En los siguientes pasos copia las distintas versiones del GDG cargadas en el ejemplo anterior, como miembros de este particionado.

Por ultimo, se ha pretendido copiar todos los miembros de este fichero en un secuencial, para lo cual lo primero que se hace es borrar este, si existe, y copiar mediante la utilidad correspondiente todos los miembros a un fichero secuencial.

Nota: Todos los **programas de utilidad** usados en este ejemplo, son propios del sistema operativo, y por tanto estándar en todas las grandes instalaciones con MVS.

-LIBRO.DOC : El autor de este artículo ha escrito un libro, inédito hasta la fecha, titulado **“TSO para desarrolladores”** en el cual describe los conceptos básicos de MVS y ya mas exhaustivamente, con ejemplos concretos, los tres apartados que debería conocer todo desarrollador en una gran instalación JCL, CLIST e ISPF. El fichero citado contiene una presentación de dicho libro.