



MS 105

**Introducción
al Sistema Operativo**



Micro Sistemas S.A.

MICRO SISTEMAS S.A.

NUMERO DE CONTROL 16

INTROD. S.O.105

..../..../..

- 2
- 4
- 5
- 6
- 10
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- . 19
- . 20
- . 22
- . 25
- . 26
- . 27
- . 28
- . 36

- . 40
- . 51

Características Principales	2
Instalación y Puesta en Funcionamiento	4
Manejo del Sistema Operativo MS-DOS	5
Manejo de los Archivos en Disco/Diskette	6
Comandos Específicos	10
COMANDO UTILITARIAS	12

M S - 1 0 5	13
Controlador de Archivos (CMA)	14
Controlador de Archivos (CMA)	15
Controlador/Interprete CMA-10 (CMA10)	16

MONOTERMINAL - MONO PROGRAMACION	17
Controlador (CMA)	18
Controlador (CMA)	19
Controlador de Archivos en Disco/Diskette (CMA)	20

G U I A D E L U S U A R I O	22
Controlador de Archivos (CMA)	25
Controlador de Archivos en Disco (CMA)	26
Controlador de Archivos (CMA)	27
Controlador de Archivos (CMA)	28
Controlador de Archivos (CMA)	36
Controlador de Archivos para Terminal (CMA)	40
Controlador de Archivos para Terminal (CMA)	51

I N D I C E

Características Principales	2
Encendido y Puesta en Funcionamiento	4
Comandos del Sistema Operativo MS-105	5
Característica de los Archivos en Disco/Diskette	6
Descripción Comandos Internos	10
RUTINAS UTILITARIAS	12
Editor de Textos (EDT105)	13
Compilador Assembler (ASM105)	14
Monitor Assembler (CAR105)	15
Compilador/Intérprete CBASIC (BAS105)	16
Monitor CBASIC (RUN105)	17
Compilador COBOL (COB105)	18
Linkedición (LNK105)	19
Determina Espacio Libre en Disco/Diskette (ESP105)	20
Copia/Listado de Archivos (COP105)	22
Listador Archivo en Hexadecimal (HEX105)	25
Listado de Archivo de Datos (LST105)	26
Listado Programas en Assembler (LPG105)	27
Clasificación de Archivos (SOR105)	28
Ejecución Archivos de Comandos (CMD105)	36
Paquete de Subrutinas para Lectura/Grabación de Diskettes a modo MS-101	40
Subrutina "VIDEO" para Terminal DIALOGUE'80	51

INTRODUCCION AL SISTEMA OPERATIVO

MS - 105

MONOTERMINAL - MONOPROGRAMACION

Se describen aquí, las características principales del Sistema Operativo MS-105.-

Configuración de Hardware para operar con el SØ-105.-

MS - 105

Procesador Central Z-80.-

Memoria interna de hasta 64 Kb.-

Impresora de 120 cps., 180 cps. o 300 lpm.-

Terminal de 1920 caracteres "DIALOGUE'80" (c/teclado).-

Discos flexibles de simple densidad.-

Disco fijo de 14 Megabytes o 28 Megabytes.-

Sistema Operativo batch.-

INTRODUCCION AL SISTEMA
MS-104
SISTEMA OPERATIVO

Se describen aquí, las características principales del Sistema Operativo MS-104. (Características de hardware para operar con el MS-104)

Procesador Central MS-104
Memoria interna de hasta 64 KB
Impresora de 120 cps, 180 cps, o 300 cps
Terminal de 1024 caracteres (MS-104) (colores)
Discos flexibles de 8 pulgadas
Banco fijo de 16 registros a 22 caracteres
Sistema Operativo batch

INFORMACION

GENERAL

PRELIMINAR

La nueva línea de equipos lanzados al mercado por Micro Sistemas S.A., está basada, en general, sobre equipos MS-104 y su Sistema Operativo, es decir que desde el punto de vista del Software, operación de la máquina, constitución y manejo de archivos, son completamente compatibles con los equipos MS-104, más una serie de posibilidades incorporadas por tratarse de un Sistema Operativo más poderoso.-

ENCENDIDO Y PUESTA EN FUNCIONAMIENTO

DEL EQUIPO MS-105

El equipo se pondrá en funcionamiento al deprimir una llave interruptora ubicada en la parte posterior del Gabinete de la Unidad Central de Proceso (CPU).-
En este instante se escucha una señal (BEEP) sonora que nos indica que la consola también fue activada.-

Colocamos en la Unidad "A" un diskette conteniendo el SØ-105 y, deprimimos el botón RESET.-

Esto provoca la lectura y carga del Sistema Operativo MS-105, el cual nos envía a pantalla el siguiente mensaje:

MICRO SISTEMAS S.A. MS-105

A>

Esto nos indica que el equipo está a la espera de comandos.-

NOTA: Si el diskette que se colocó en "A" no es el correcto, o sea, no contiene el SØ-105, se emitirá el mensaje:

"Error de carga BOOT"

Debiendo en tal circunstancia, reemplazar el diskette de la Unidad "A" por uno correcto y nuevamente dar RESET.-

COMANDOS DEL SO-105

Los comandos del SO-105, son de dos tipos:

- Comandos internos.-
- Comandos externos.-

Genéricamente, los comandos tienen el siguiente formato:

- Comando
- Comando Nombre.de.Archivo
- Comando Nombre.de.Archivo Tipo.de.Archivo

donde "Comando" puede ser un comando interno del SO-105, del tipo DIR o TYPE, o el nombre de un comando externo o programa.-

Si el comando es interno, éste es ejecutado inmediatamente.-

Si el comando es externo, el SO-105 busca en el disco actualmente vigente, el archivo con el nombre:

Comando.COM

y si el SO-105 asume que se trata de un archivo a imagen de memoria listo para / ser ejecutado. El mismo es cargado en memoria y ejecutado.-

ARCHIVOS EN EL SO-105

Las características principales de los archivos utilizados en el SO-105, son las siguientes:

- No existe diferencia alguna, desde el punto de vista del programador, / entre archivos en diskette y archivos en disco fijo.-
- Usuario: es un número que identifica a los archivos como pertenecientes a una determinada persona, unidad o proceso.-
El sistema soporta hasta 16 números de usuario (0-15).-
- Tipo de acceso: Designa características operativas particulares para cada archivo. Pueden ser:
 - a) Archivos de Sistema: No aparecen en los listados de Directorio y no pueden ser copiados sin que se den instrucciones específicas.-
 - b) Archivos de lectura solamente(R/O): solo pueden ser accedidos por el Usuario para ser leídos y, no pueden ser regrabados, modificados ni eliminados del Directorio sin instrucciones específicas.-
 - c) Archivos de lectura y grabación: Pueden ser grabados, modificados o eliminados.-

ARCHIVOS EN DISKETTE:

- Hasta 64 archivos por diskette.-
- Registros de longitud variable.-
- La organización lógica de sectores de 128 bytes en forma secuencial, si bien los sectores físicamente pueden no estar uno a continuación del otro.-
- Se pueden acceder en forma secuencial o al azar a cada registro.-

ARCHIVOS EN DISCO FIJO

Las características principales de los archivos en disco fijo son las siguientes:
- No existe diferencia alguna, desde el punto de vista del programador, entre archivos en disquete y archivos en disco fijo.
- Los archivos en un mismo que idéntica a los archivos como pertenecientes a una determinada persona, entidad o proceso.
- El sistema operativo hace el número de sectores (0-127).
- Tipo de acceso: Designa características operativas particulares para cada archivo. Pueden ser:
a) Archivos de sistema: No aparecen en los listados de directorios y no pueden ser copiados sino que se los trata como archivos especiales.
b) Archivos de lectura solamente (RO): Solo pueden ser accedidos por el usuario para ser leídos y no pueden ser modificados, reemplazados ni eliminados del directorio sin intervención especial.
c) Archivos de lectura y escritura: Pueden ser creados, modificados y eliminados.

ARCHIVOS EN DISCO FIJO

- Hasta el archivo por defecto.
- Registro de longitud variable.
- La organización lógica de sectores de 128 bytes en forma secuencial, al igual que los sectores físicos pueden no estar en la organización del disco.
- Se pueden acceder en forma secuencial o al estar a disposición.

ARCHIVOS EN DISCO FIJO

El/los disco fijo está dividido lógicamente en DISK-DRIVES. Cada uno de ellos están identificados con una letra (C,D,E, etc.).-

Por ejemplo:

- Un disco fijo de 14 megabytes está dividido en 2 Disk-drives, C y D con las siguientes características:

C	D
C: CARACTERISTICAS UNIDAD	D: CARACTERISTICAS UNIDAD
65536: REGISTROS DE 128 bytes	31360: REGISTROS DE 128 bytes
8192: KILOBYTE DE CAPACIDAD	3920: KILOBYTE DE CAPACIDAD
512: ENTRADAS DE DIRECTORIO	512: ENTRADAS DE DIRECTORIO
0: ENTR. DIRECT. CONTROLADAS	0: ENT.DIRECT.CONTROLADAS
1024: REGIST./EXTEN.	2048: REGIST./EXTEN.
128: REGIST./BLOCK.	128: REGIST./BLOCK
52: SECTORES / TRACK	52: SECTORES / TRACK
0: TRACKS RESER	1261: TRACKS RESER

///... Cont.

Un disco fijo de 28 Megabytes estaría dividido en C,D y E, de los cuales C y D comparten las mismas condiciones de lo especificado en punto anterior.-
Las diferencias de capacidad que se observan, son debido a la reserva por parte del SO-105 de sectores alternativos, entradas de Directorio y marcas de formateo.-
Un disco "lógico" soporta hasta 512 entradas de directorio.-
Cada archivo del Sistema Operativo 105 se caracteriza por tener un nombre y opcionalmente un tipo de archivo.-
El nombre es conjunto de caracteres (letras y números) de un largo máximo de 8.-
El tipo de archivo se designa con tres caracteres. Una referencia a un archivo del SO-105, se hará de la siguiente manera:

NOMBRE.TIPO

Ejemplo:

FACTURAS.DAT

SORNUM

INVERSOR.BAS

A continuación se describen los tipos de archivos empleados por el SO-105

- COM : Designa archivos con programas en lenguaje de máquina directamente ejecutables. Las funciones externas del SO-105 residen en archivos tipo COM.-
- ASM : Archivo de programas fuente para el compilador ASSEMBLER.-
- BAS : Archivo de programas fuente para el compilador.-
- COB : Archivo de programas fuentes empleado por el programa COBOL.-

///...

///...Cont.

- PRN : Archivos generados por el compilador ASSEMBLER para su impresión.-
- LST : Archivos generados por el compilador CBASIC para su impresión.-
- HEX : Archivo generado por el compilador ASSEMBLER con los códigos de instrucción en hexadecimal.-
- INT : Archivo generado por el compilador CBASIC en un código propio.-
- BAK : Archivo "resguardo" generado por el programa Editor de Textos (EDT105).-
- \$\$\$: Archivo de trabajo empleado por varias funciones del SO-105.-
- REL : Archivo de trabajo empleado durante la compilación para el COBOL.-
- LIB : Archivo de librería creado para el uso, en el Editor de Textos (EDT105).-
- SUB : Archivo de Comandos creado para usar con la rutina de Procedimientos /// (CMD105).-

Toda referencia a un archivo en disco provoca su búsqueda en el disco actualmente vigente. Para acceder a archivos en el otro eje, se debe anteponer al nombre del archivo, el nombre de la unidad (A,B,C, o D) seguida de dos puntos (:).-

Ejemplo:

B:DATOS

A:XXXXXX.BAK

Cuando se accede a un archivo del SO-105, tanto los comandos del sistema como los programas del Usuario, deben constituir un bloque de Control de Archivo (BCA). El Subsistema de Entrada/Salida (E/S) del SO-105, recibe la dirección del BCA y un comando (lectura, cierre, grabación, etc.), que ejecuta la función deseada retornando una señal que indica si la operación fue correctamente ejecutada o no.-

DESCRIPCION DE LOS COMANDOS DEL SO-105

Los comandos internos del SO-105 son:

- DIR : Emisión del Directorio.-
- TYPE : Emisión en pantalla del contenido de un archivo.-
- ERA : Eliminación de archivos del usuario activo.-
- REN : Modificación del Nombre de Archivo.-
- SAVE : Rescate de área de memoria.-
- USER : Cambio de Usuario.-

Los comandos DIR y ERA aceptan referencias "ambiguas".
Ejemplo:

- DIR o DIR x.x.
provoca la emisión de todo el Directorio del disco vigente.-
- DIR x. COM
emite todos los archivos tipo COM.-
- DIR DATO. x
emite todos los archivos de nombre DATO, cualquiera sea su tipo.-
- ERA x. BAK
provoca la eliminación de todos los archivos tipo BAK del directorio del disco vigente.-
- ERA P???x

Elimina los archivos cuya primera letra sea P, cualquiera sea su tipo. El comando TYPE debe estar seguido por una referencia no ambigua de un archivo, con su nombre y su tipo de archivo. El contenido del mismo será desplegado en la pantalla.-

Ejemplo:

TYPE DATOS

TYPE CALCULO.BAS

El comando REN debe estar seguido por dos referencias completas separadas por un signo igual (=).-

Ejemplo:

REN CALCULO.ASM=CALCULO.BAK.

el archivo CALCULO.BAK, pasa a llamarse CALCULO.ASM.-

El comando SAVE se utiliza para trasladar el contenido de la memoria libre del sistema a un archivo en disco.-

Ejemplo:

SAVE 4 MEMORIA.BAK

Se crea un archivo de nombre MEMORIA.BAK, que contendrá 8 sectores, éstos tendrán el contenido del primer K de la memoria del Usuario.-

Cualquier comando erróneo o, referencia a un archivo inexistente dará lugar al / mensaje:

XXXXX NO EXISTE

donde XXXXX es el conjunto de caracteres que provocó el error.-

Ejemplo:

TYPE DATOS

TYPE CALCULO.BAS

El comando REM hace estos comentarios por los comentarios completos separados por un signo igual (=).

Ejemplo:

REM CALCULO.ASM-CALCULO.BAS

El archivo CALCULO.BAS, hace a las mismas cosas que el comando SAVE en un archivo pero creando el contenido de la memoria libre del sistema a un archivo en disco.

Ejemplo:

SAVE & MEMORIA.BAS

Se crea un archivo de nombre MEMORIA.BAS, que contiene a memoria, desde la hora al contenido del primer y de la memoria del sistema. Cualquier comando escrito o almacenado a un archivo con este nombre se mantendrá.

XXXXX NO EXISTE

Como XXXX no existe en el archivo de comandos que se ejecuta al escribir.

MS - 105

MONOTERMINAL - MONOPROGRAMACION

RUTINAS UTILITARIAS

EDT105 EDITOR DE TEXTOS

El programa Editor de Textos se utiliza para crear o modificar archivos del MS-105. Los archivos podrán ser tanto de datos como archivos de programas / fuente para los compiladores/intérpretes del SO-105.-

Se convoca digitando:

EDT105 A: NOMBRE.TIPO-cr-

Si el archivo NOMBRE.TIPO ya existe, puede ser alterado agregando, supri- / miendo o modificando líneas.-

Si el archivo especificado no existe, el programa editor lo crea en el dis- / co vigente, o en el disco especificado en el parámetro.-

Después de una edición el programa EDT105 deja los archivos en el diskette: / el NOMBRE.TIPO y el NOMBRE.BAK; este último como respaldo o "back-up" del // primero y su contenido es el del archivo antes de la última edición.-

Para mayor información sobre el Editor de Texto, recurrir al manual corres- / pondiente.-

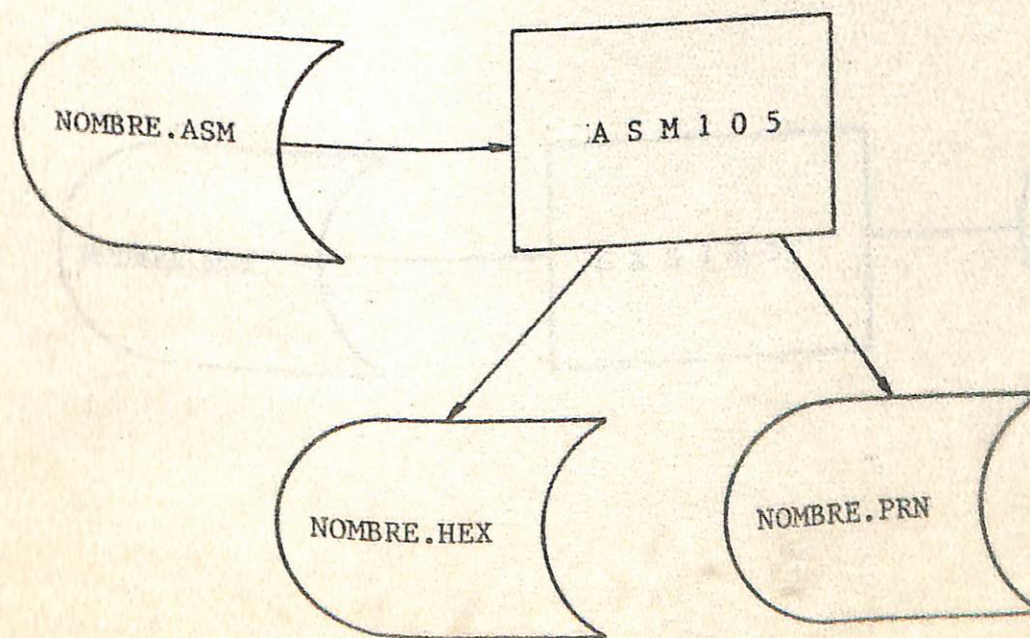
Traduce las instrucciones del lenguaje Assembler de MS-105, contenidos en un archivo de tipo ASM, a su equivalente hexadecimal, capaz de ser directamente ejecutada por el procesador.-

La compilación se convoca con el comando:

ASM105 NOMBRE-cr-

Después de la compilación aparecen los archivos tipo HEX, con las instrucciones de máquinas en hexadecimal y tipo PRN con las sentencias originales en // lenguaje Assembler y su equivalente en hexadecimal.-

Para una información completa sobre funciones y operaciones, ver Manual de // Assembler para MS-105.-

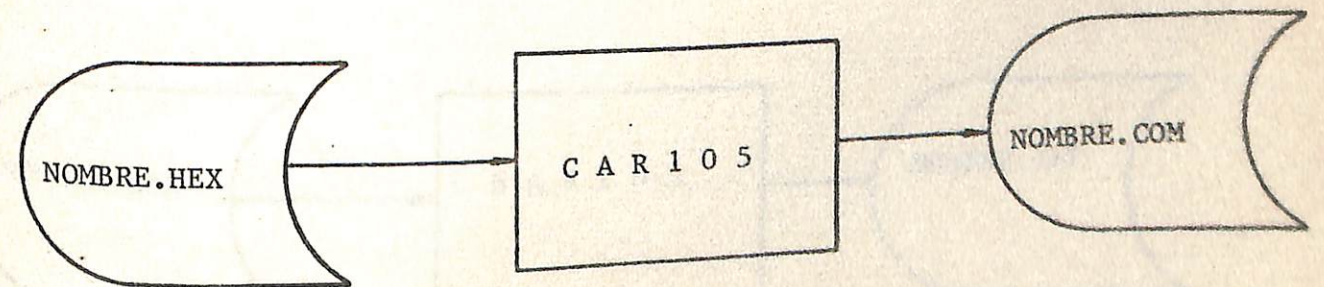


CAR105 MONITOR ASSEMBLER

El comando es convocado digitando:

CAR105 NOMBRE-cr-

El comando CAR105 crea un nuevo archivo NOMBRE.COM, y se debe ejecutar después de una compilación Assembler. Los archivos tipo COM son programas a imagen de memoria y pueden ser ejecutados tipeando directamente NOMBRE.-cr-, al igual // que los programas de Sistema. Esto equivale a crear una nueva función externa del SO-105. Los programas Assembler deberán escribirse a partir de la dirección 100H para no interferir con el Sistema Operativo.-



B A S 1 0 5

C O M P I L A D O R

I N T E R P R E T E

C B A S I C

A partir de un archivo fuente del tipo BAS, generado con rutina utilitaria de edición (EDT105), con sentencias en lenguaje CBASIC (Basic de Orientación Comercial), el compilador genera un archivo tipo INT.-

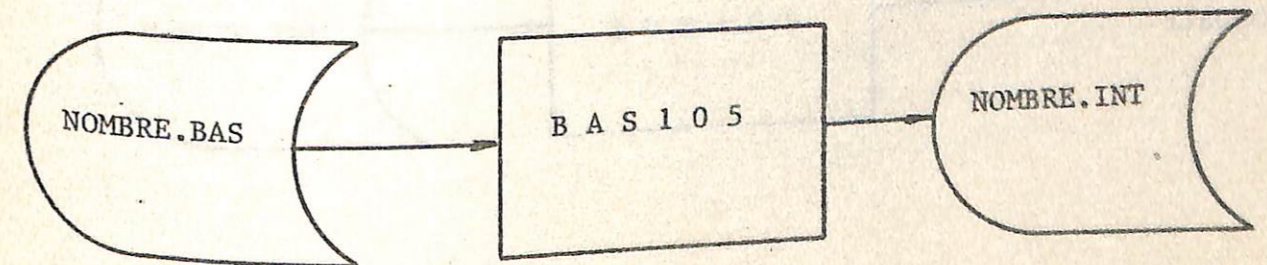
La compilación se ejecuta con el comando:

BAS105 NOMBRE-cr-

El archivo generado por el compilador contiene una serie de códigos que serán interpretados por el Monitor en tiempo de ejecución. Puede optarse por suprimir el listado en pantalla del programa, o su impresión; también puede obtenerse un archivo LST con el compilador.-

Los errores que pudiera contener el archivo fuente, serán mostrados en la posición en que fueron detectados por el compilador, exhibiéndose un Código de Error, cuyo significado deberá consultarse en el manual respectivo.-

Para información completa sobre funciones y operación ver el Manual de Programación Basic para MS-105.-



R U N 1 0 5 M O N I T O R C B A S I C

Después de una compilación, sin errores de sintáxis, esta rutina permite la ejecución de un programa escrito en lenguaje CBASIC.-

Dicha ejecución se convoca digitando:

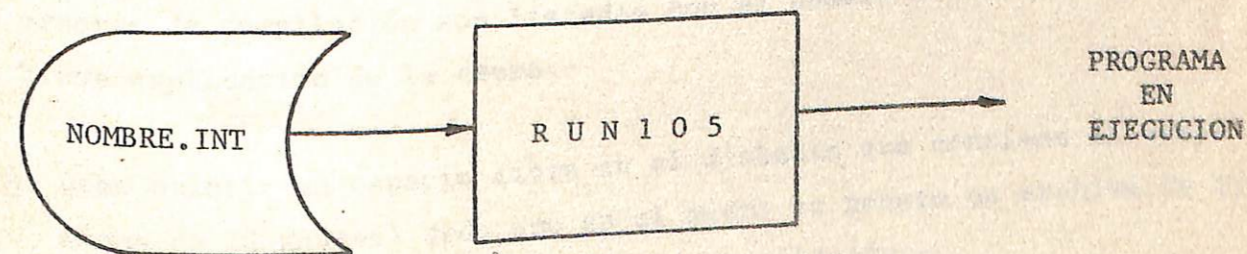
RUN105 NOMBRE-cr-

Sobra recalcar que debe existir previo a este comando el programa NOMBRE.INT.-

Los códigos del archivo NOMBRE.INT son interpretados por el programa Monitor y ejecutados.-

En caso de provocarse un error en tiempo de ejecución el monitor CBASIC re-/ torna un código de error con el número de línea del programa fuente donde ocu- rrió la falla .-

Por una correcta definición del significado de cada uno de estos códigos con-/ sultar Manual de Programación CBASIC.-



A partir de un archivo del tipo COB, que contiene sentencia en lenguaje COBOL, el compilador genera un archivo tipo REL.-

Este archivo tipo REL constituye un intermedio entre el programa fuente y el ejecutable.-

Contiene códigos que luego serán interpretados por el programa LINK-EDITOR para generar un programa del tipo COM (ejecutable).-

La compilación se ejecuta digitando:

COB105 X:NOMBRE, ^{TTY}LST :=Z:NOMBRE-cr-

Donde X = Unidad Destino y Z = Unidad de Origen o donde esté el programa fuente.-

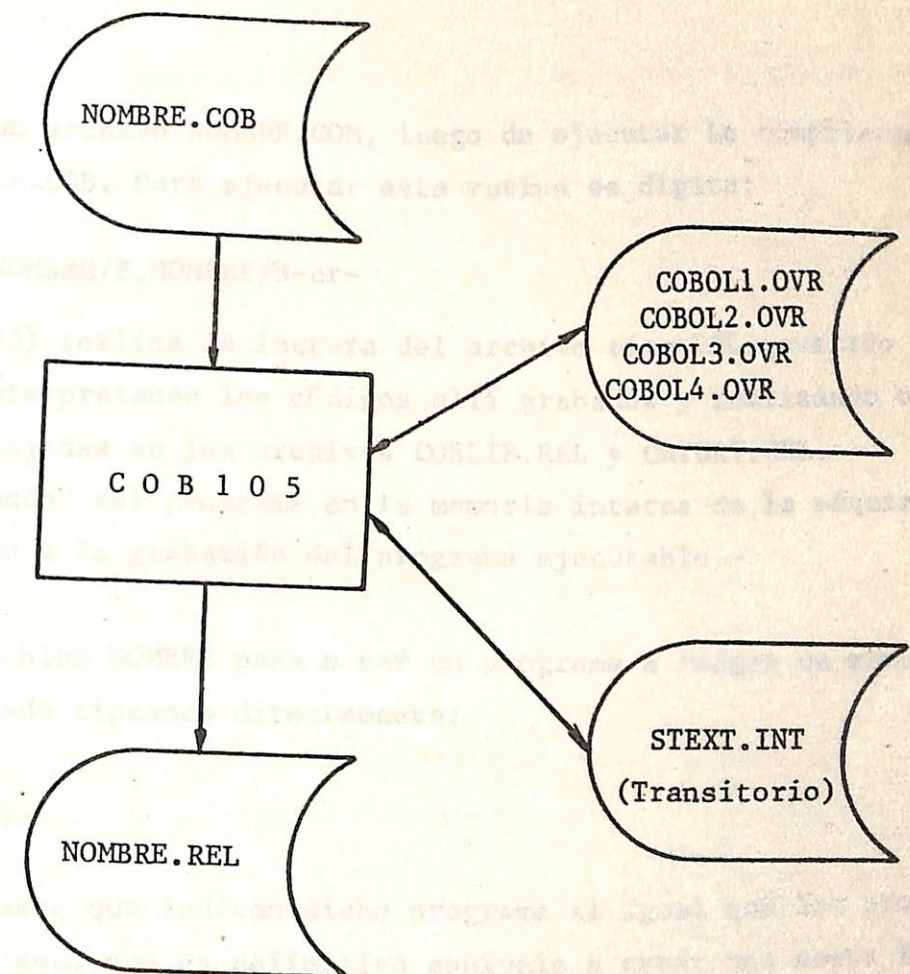
Las opciones TTY y LST son excluyentes y nos indican por cual dispositivo se emitirá el listado de compilación.- (TTY = pantalla LST=Impresora).-

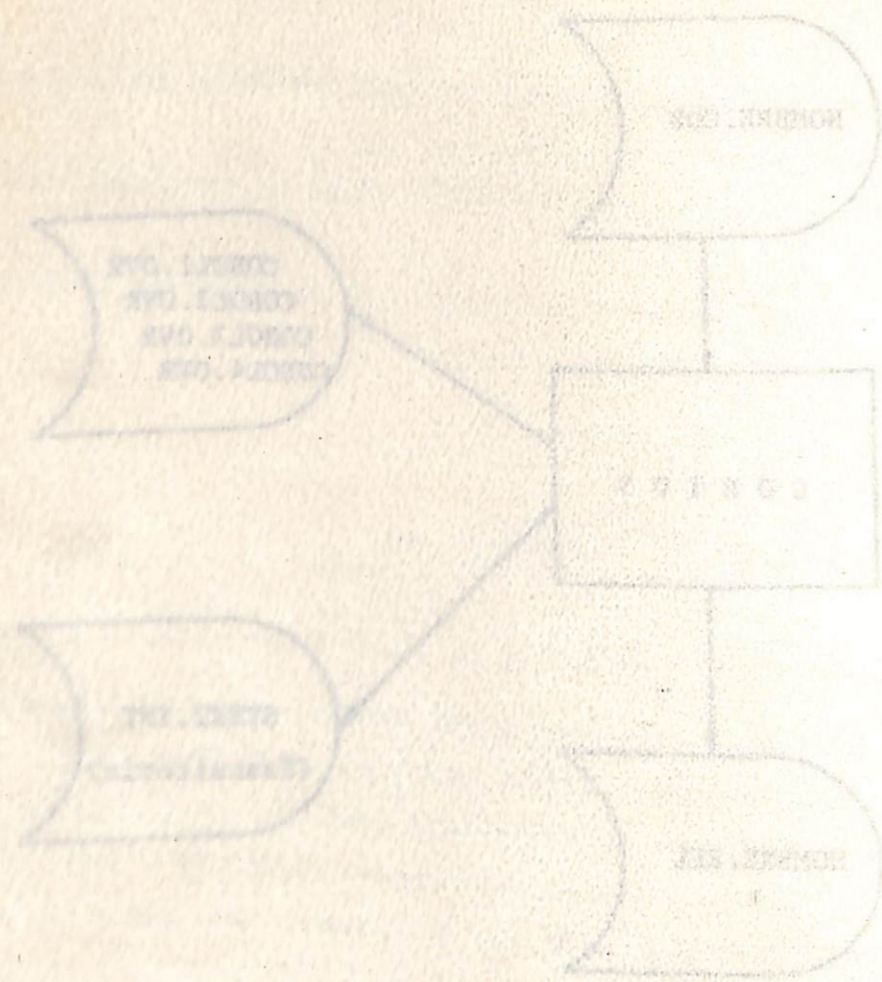
Es condición imprescindible que el compilador (COB105) se encuentre en el Unidad Activa juntamente con cuatro archivos denominados OVERLAYS (COBOL1.OVR/COBOL2.OVR/COBOL3.OVR/COBOL4.OVR), dado que estos son utilizados como áreas de trabajo y consalta del COB105.-

Los errores de compilación son listados con el número de línea donde se produjo y una breve explicación de la causa.-

NOTA: Debe existir un espacio libre en el diskette que contiene el compilador (no menos de 20 Kbytes) dado que en el mismo se genera un archivo de trabajo // STEXT.INT que es deleteado el final de la compilación.-

///...Cont.





L N K 1 0 5

Esta rutina crea un archivo NOMBRE.COM, luego de ejecutar la compilación del programa NOMBRE.COB. Para ejecutar esta rutina se digita:

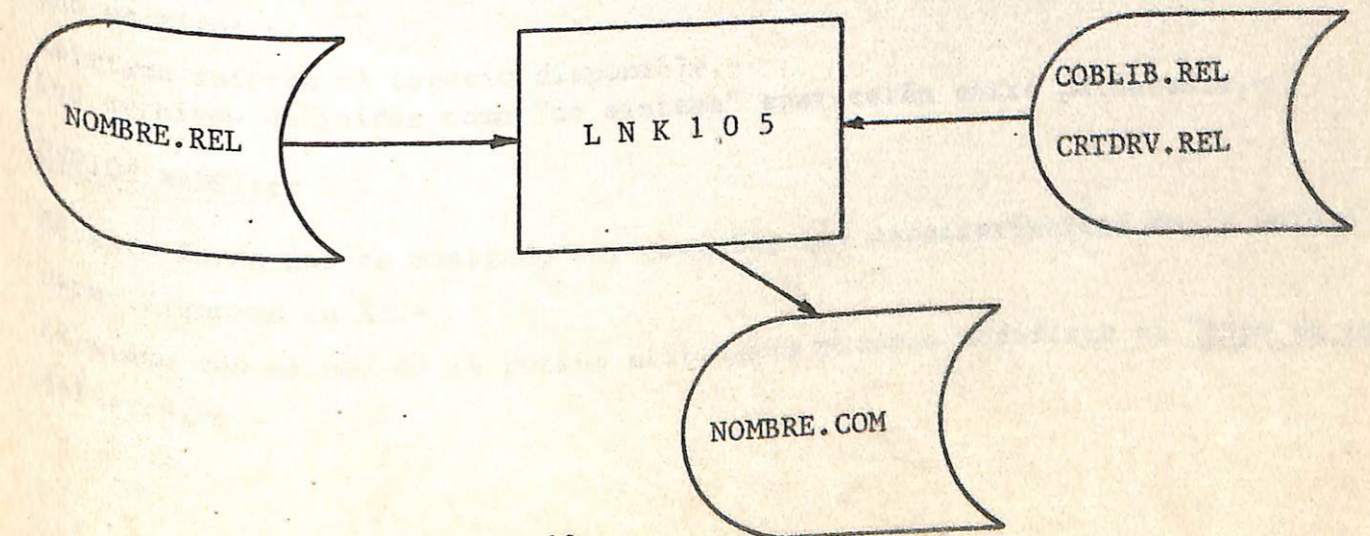
LNK105 NOMBRE/E,NOMBRE/N-cr-

Esta rutina (LNK105) realiza la lectura del archivo tipo REL generado en la compilación, interpretando los códigos allí grabados y realizando o / tomando rutinas alojadas en los archivos COBLIB.REL y CRTDRV.REL.- Se realiza el "armado" del programa en la memoria interna de la máquina y procediendo luego a la grabación del programa ejecutable.-

Paso seguido el archivo NOMBRE pasa a ser un programa a imagen de memoria y puede ser ejecutado tipeando directamente:

NOMBRE-cr-

Ejecutándose los pasos que indican dicho programa al igual que los programas utilitarios; o sea, que en definitiva equivale a crear una nueva función externa del SO-105.-



Se utiliza para conocer el espacio disponible en disco para Almacenamiento de datos o para conocer el espacio ocupado por un determinado archivo.-

Esta rutina utilitaria acepta diferentes comandos:

ESP105 cr

Nos dá como resultado la capacidad total "disponible" para almacenar datos en todas las unidades.-

ESP105 X:cr

Nos dá el espacio disponible en la unidad que le especifiquemos. (x=unidad).-

ESP105 X:NOMBRE cr

Es impreso por pantalla el espacio ocupado por el archivo NOMBRE.-
Esta información consiste en la cantidad de sectores y Kbytes ocupados por el Archivo NOMBRE y el espacio disponible en X:.-

ESP105 X:*. * cr

Nos muestra una tabla en pantalla, ordenada en orden alfabético con todos los archivos existentes en la unidad especificada y el espacio ocupado por cada uno de ellos.-

Asimismo informa el espacio disponible.-
Los archivos definidos como "de sistema" aparecerán entre paréntesis.-

ESP105 X:DSK:cr

De esta forma nos es mostrado por pantalla las características de la unidad que especifiquemos en X:.-

Asimismo con el uso de la rutina utilitaria podemos modificar el "tipo de acceso" del archivo.-

Utilizamos los siguientes comandos:

- 1.- ESP105 X:NOMBRE.TIP \$R/O cr
- 2.- ESP105 X:NOMBRE.TIP \$R/W cr
- 3.- ESP105 X:NOMBRE.TIP \$SYS cr
- 4.- ESP105 X:NOMBRE.TIP \$DIR cr
- 5.- ESP105 X:*. * \$R/W cr

Si optamos por definir un archivo como R/O (READ ONLY) o sea de lectura solamente, // usaremos el comando primero. Este provocará que el archivo indicado, se transforme en uno de lectura solamente, no pudiendo ser modificado ni regrabado por el Usuario.- En ocasión de tratar de realizar alguna de estas dos últimas operaciones, el S0105 // emitirá un mensaje de error advirtiéndolo al operador que el archivo es R/O.-

Bdos ERR ON X: FILE R/O

En tal circunstancia, digitando RETURN se habilitará nuevamente la consola.-
Con el uso del comando número 2, se coloca una señal de que ese archivo puede ser leído y regrabado (R/W - READ/WRITE).-
Si definimos el archivo como de Sistema, (como se ilustra en el ejemplo 3), éste no será mostrado ante la ejecución de un comando DIR, ni será accesible por el Usuario hasta tanto se modifique la señal.-

Si especificamos \$ DIR se resetea el indicador de sistema (ver ejemplo 4).-
Usando un comando como el número 5, se cambiarán todos los tipos de acceso de R/O a R/W, independiente de que el/los archivo/s sean "de sistema".-

C O P I A D E A R C H I V O S

Se puede convocar el programa COP105 y la aparición de un asterisco (*); no indica que el COP105 está a la espera de función.-

Otra forma de convocar el COP105 sería, ingresar a posteriori de COP105, la línea de comandos precedida por un espacio y terminada con un RETURN.-

Ejemplo de algunos comandos más usados:

COP105 B:=A:NOMBRE cr

Copia el archivo NOMBRE de la Unidad A hacia la Unidad B con el mismo nombre.-

COP105 B:=A:*.*

Copia todos los archivos de A hacia la unidad B: con el mismo nombre.-

COP105 B:=A:*.COB

Copia de la unidad A hacia la B todos los archivos COB.-

COP105 B:GAMMA=A:ALFA,B:BETA

Realiza la concatenación de los archivos ALFA y BETA y graba un único archivo en B: llamado GAMMA.-

COP105 TTY:=B:ALFA cr

Realiza una copia desde la unidad B: hacia la pantalla del archivo ALFA.-

De la misma forma podemos realizar la copia hacia el dispositivo de impresión especificando LST:.-

En este último caso podemos especificar la cantidad de líneas a ser impresas en cada hoja, de la siguiente forma:

COP105 LST:=B:ALFA[Pn] cr

donde n = cantidad de líneas por hoja.-

COP105 NOMBRE.LIB=NOMBRE.COB [SWORKING ↑ZPROCEDURE ↑Z] cr

Este comando provocará la copia de los registros (del archivo NOMBRE.COB) comprendidos entre los STRINGS indicados entre corchetes, hacia el archivo NOMBRE.LIB.-

NOTA: S y Q deben preceder los strings y ↑Z es la tecla CTL-Z.-

COP105 A:=B:*.COM [V]

Realizará la copia de todos los archivos tipo COM de la unidad B hacia la unidad A verificando la correcta grabación de los mismos.-

IMPORTANTE

La rutina utilitaria COP105 realiza la copia de un archivo aunque exista previamente una entrada de directorio con idéntico nombre en el disco DESTINO; salvo que el archivo esté marcado como archivo de lectura solamente (R/O); en este caso, solicitará al operador instrucciones para sobrecargar sobre ese archivo.-

Esta interrogación podrá ser obviada si al dar el comando para la rutina COP105, especificamos [W] antes de tipear RETURN.-

Ejemplo:

COP105 B:=A:*.COM[W] cr

Un archivo definido como de sistema solo podrá ser copiado si especificamos [R] .-

Entre corchetes al final del comando para el COP105 podemos agregar lo siguiente:

- Dn Que sean eliminados los caracteres de un registro, ubicados a la derecha de la columna especificada en n.-
- E Muestra por pantalla las operaciones, mientras ellas se están ejecutando.-
- L Convierte letras mayúsculas a minúsculas en el archivo destino.-

N Agrega en el archivo de Destino el número de línea- Comenzando en 1 e incrementando en 1. Se imprimen los ceros a la izquierda, lo / que no ocurre si se especifica NZ.-

Tn Expande la tabulación a n columnas durante la transferencia de caracteres.-

U Convierte minúsculas a mayúsculas.-

NOTA: No se recomienda el uso de la opción de copia con comandos AMBIGUOS
Ejemplo: COP105 C:=D:*.COM.

H E X 1 0 5

EMISION DE LISTADOS EN HEXADECIMAL

Con esta rutina se listan los datos de un archivo de NOMBRE en
Código hexadecimal.-

Para ejecutarla se digita:

HEX105 NOMBRE-cr-

a continuación, por impresora se emite el listado completo del
archivo NOMBRE.-

L S T 1 0 5

EMISION EN LISTADO DE ARCHIVO DE DATOS

Esta rutina lista, por impresora, los datos que contiene el archivo NOMBRE.
El listado emitido contiene todos los datos que en ese momento tuviese el /
archivo NOMBRE. En el caso de haber compaginado y compilado un programa, al
convocarlo a listar con esta rutina, saldrán en el listado todas las senten-
cias del programa con los errores producidos durante la compilación. Para /
convocar esta rutina se digita:

LST105 NOMBRE-cr-

Paso seguido, comenzará a imprimirse el archivo requerido.-

EMISION DE LISTADOS DE PROGRAMAS ASSEMBLER

Esta rutina permite listar programas pero, únicamente aquellos escritos en lenguaje ASSEMBLER, respetando su formato de escritura. La convocación se realiza digitando:

LPG105 NOMBRE.PRN-cr- o LPG105 NOMBRE.PRN (aplicación del programa)

luego, aparece la leyenda:

APLICACION = (si no se especificó en el paraámetro).-

al lado de la cual se digita la descripción de la aplicación del programa a listar; esta descripción tiene una extensión máxima de 35 caracteres. Ejemplo:

APLICACION = RUTINA ELIMINACION ZONA-cr-

Una vez digitada la descripción de la aplicación se comienza a emitir el listado correspondiente. Es importante que el archivo convocado sea del tipo PRN, ya que este tipo de archivo es el destinado para imprimir. El archivo tipo PRN resulta de correr el programa ASSEMBLER con el utilitario ASM105.-

S O R 1 0 5

CLASIFICACION DE ARCHIVOS

Es una rutina de clasificación o intercalación para archivos en el SO-105.-
Se describen aquí, los elementos fundamentales para su utilización, pero se recomienda referirse al manual correspondiente para un completo dominio de esta versátil rutina. Para su ejecución se debe digitar:

SOR105-cr-

Luego de algunos segundos, en la pantalla aparece un asterisco (*), significando que la rutina espera los siguientes parámetros para su ejecución.-

PARAMETROS

- Especificación de largo de registro y tipo de organización del archivo. Deberá indicarse la longitud máxima de registro contenida / en el archivo y el par de letras CR, para indicar que son registros delimitados por: -cr- -lf-. Esta modalidad de archivo es la empleada por el CBASIC, el COBOL y el EDT105.-

Ejemplo:

INPUT = 62, CR-DEL

que se puede abreviar

IN 62 CR o I 62 CR

en este caso la longitud máxima de registro es de 62 caracteres.-

CLASIFICACION DE ARCHIVOS (Cont.)

- Indicación del archivo que va a ser tomado como entrada para la clasificación. Ejemplo:

SORT-FILE = ARCHIVO.DAT

Se va a clasificar un archivo de nombre ARCHIVO.DAT, ubicado en la unidad que está activa en este momento.-

SORT B:DATOS o S=B:DATOS

Aquí se expresa en forma abreviada que el archivo a clasificar es el DATOS, ubicado en la unidad "B".-

- Indicación del archivo de salida que va a contener los datos clasificados. Se puede obtener como salida el mismo archivo especificado como entrada. Ejemplo:

OUTPUT-FILE = ARCHCLAS

El archivo de salida es el ARCHCLAS, ubicado en la unidad de disco actualmente activa. La forma abreviada de la indicación anterior es:

OUT A:ARCHCLAS o O=A:ARCHCLAS

Los datos clasificados estarán en el archivo ARCHCLAS de la unidad "A".-

- Indicación de claves. Se puede indicar la ubicación de la clave de /

CLASIFICACION DE ARCHIVOS (Cont.)

clasificación dentro del registro, de dos formas:

- 1.- Si los registros tienen campos separados por comas, a la modalidad standard del CBASIC, se indicará el número de campo a tomar y la longitud dentro del mismo. Ejempló:

KEY = #2, 10 ASCENDING

la clave está ubicada en el segundo campo y se toma de él como máximo 10 caracteres. La forma abreviada es:

K #2, 10 ASC

- 2.- Si se consideran registros cuyos datos se identifican por su posición relativa dentro del registro, se indicará la posición inicial y final de la clave a considerar. Ejemplo:

KEY = 2, 5 ASCENDING

la clave se toma desde la posición 2 (inclusive) hasta la posición 5 (inclusive). Una aplicación en conjunto de las dos formas es:

KEY 7,19 22,23 ó K 7,12 22,23

indica que primero se ordenará por el contenido de las posiciones 7 a 19 y luego por el contenido de las posiciones 22 y 23. Se asume el orden ascendente (ASC). Si se desea la clasificación en orden

CLASIFICACION DE ARCHIVOS (Cont.)

CLASIFICACION DE ARCHIVOS (Cont.)

descendente se indicará DESCENDING, o su forma abreviada DESC.

3.- Comando de ejecución. Una vez completada la secuencia de parámetros se ordenará que se inicie la ejecución del SOR105, digitando:

GO o G

Al recibir cada comando el programa realiza un chequeo del mismo y emite un mensaje de error, si se ha indicado un comando erróneo. El orden de los comandos indicados anteriormente es indistinto, con la única condición de que el último siempre debe ser GO. Una vez completada la clasificación aparecen en pantalla los mensajes indicando la cantidad de registros clasificados y el tamaño de los archivos en /// Kbytes.-

Un ejemplo completo de la clasificación de un archivo es el siguiente:

IN 80 CR
SORT A:REGISTRO
OUT B:REGCLAS
KEY 1,5 ASC 7,14 ASC
G.

CLASIFICACION DE ARCHIVOS (Cont.)

ARCHIVO DE PROCEDIMIENTO

Si el Usuario prepara con el Editor de Textos (EDT105), un pequeño archivo cuyo contenido sean los comandos para el SOR105, la clasificación puede ejecutarse / con dichos comandos tipeando:

CF Nonbre.de.Archivo-cr-

cuando aparezca el asterisco (*) de espera para recibir comando. Si se desea ver los comandos del SOR105 se debe incluir un comando LIST en el archivo, además, si el mismo contiene un comando GO, la ejecución del SOR105 se inicia inmediatamente.-

COMANDOS EN UNA MISMA LINEA DE LLAMADA

La rutina de clasificación SOR105 puede ser convocada a a continuación en la misma línea tipear los comandos de ejecución propios de la clasificación. Ejemplo:

SOR105 I 62 CR; S ENTRADA; O B: SALIDA; K 5,7 9,10; GO-cr-

que equivale a:

SOR105 CF CLAS-cr-

suponiendo que el archivo CLAS del diskette actualmente activo contiene los siguientes comandos:

///...

CLASIFICACION DE ARCHIVOS (Cont.)

INPUT 62 CR-DEL

SOR ENTRADA

OUTPUT B:SALIDA

KEY 5,7 9,10

GO

DISCO DE TRABAJO

Si se especifica:

WORK DRIVE B:

se indica a la rutina de Clasificación que el área de trabajo está en el disco "B".-

CAMBIO DE DISCO

En la rutina de Clasificación no pueden cambiarse los discos entre los archivos de entrada. Se puede hacerlo antes del comando GO. Si se desea cambiar los discos antes de que sea escrito el archivo de salida se indica, por ejemplo, lo siguiente:

OUTPUT DESTINO/C

Cuando la rutina esté lista para emitir el archivo de salida emitirá un mensaje // por consola, indicando al operador el disco que debe ser reemplazado.-

ARCHIVO DE INTERCALACION

Si se desea juntar dos o más archivos ya clasificados, o se desea incorporar los / registros de un archivo a uno ya clasificado, se emplea el siguiente comando:

CLASIFICACION DE ARCHIVOS (Cont.)

MERGE Nombre.de.Archivo

Este tipo de comando de la rutina de Clasificación, no altera el orden de los registros dentro del archivo. A continuación se dá un ejemplo de clasificar el archivo ALTAS e incorporarlo al archivo MAESTRO.CLS:

```
IN 128 CR
SOR ALTAS
MERGE MAESTRO.CLS
OUT B:MAESTRO.CLS
KEY 10,15 ASC
P 5
GO
```

El comando "P 5" es abreviatura de "PRINT-LEVEL = 5", que ordena a la rutina de Clasificación a emitir el máximo número de mensajes a consola durante la ejecución. El comando "P 0" suprime todos los mensajes a consola. El comando "P 2" es el nivel de impresión asumido automáticamente.-

COMANDOS SELECT Y EXCLÚDE

El comando SELECT indica los registros que se desean que aparezcan en el archivo de salida. En cambio, el comando EXCLUDE indica los registros que se desean suprimir en el archivo de salida. Ejemplo:

```
SELECT FIELD #2 = "285"
```

CLASIFICACION DE ARCHIVOS (Cont.)

El archivo de salida contendrá los registros cuyo segundo campo, o sea entre la primera y segunda coma, sean igual a 285.-

EXCLUDE FIELD 2,7 = "NULO"

Aquí, los registros que tengan el campo que está definido desde la columna 2 a la columna 7, con el valor NULO son ignorados y no participan en la clasificación o intercalación.-

Tampoco son incluidos en el archivo de salida. Otro ejemplos son los siguientes:

SE #5 > "0"

se seleccionan los registros cuyo quinto campo sea mayor que cero.-

SE FI 1,2 < "50"

se seleccionan los registros cuyas dos primeras posiciones sean menor a 50 (abreviatura mínima).-

EJECUCION DE ARCHIVOS COMANDOS

Esta rutina permite convocar a un archivo de comandos en donde se tienen grabadas todas las instrucciones operativas para ejecutar un proceso determinado.-
El archivo de Comandos debe ser creado previamente con el Editor de Textos /// (EDT105), teniendo su nombre el siguiente formato:

XXXXXX.SUB

donde XXXXXX, es el nombre asignado por el Usuario y SUB, indica que es un archivo de Comandos a cumplir. El tipo de archivo SUB es obligatorio. Una aplicación concreta de esta rutina es el caso de un proceso de actualización de cuentas, en donde se tienen los siguientes objetivos:

- Clasificar los movimientos.-
- Actualizar el Maestro de Cuentas.-
- Listar los movimientos.-
- Eliminar los movimientos del archivo original.-

Si ahora se asigna un nombre a cada programa del proceso, como por ejemplo:

- | | |
|---------------|--|
| - MOVIN | (Archivo de Movimientos).- |
| - ACTUAL.INT | (Programa de Actualización).- |
| - LISTA.COM | (Programa que lista los movimientos).- |
| - CUENTAS.FNC | (Archivo de comandos).- |

en donde el Archivo de Comandos se crea una sola vez, con el Editor de Textos /// (EDT105), tipeando:

EDT105 CUENTAS.SUB-cr-

EJECUCION DE ARCHIVOS COMANDOS (Cont.)

a continuación se emite el mensaje:

EDITOR MS105 V/02

NVO ARCH

*

Después de lo cual, haciendo uso del comando "I" del Editor de Textos, insertamos los siguientes datos:

*I-cr-

SOR105 I=80 CR; S=B:MOVIM.CLA; K=1,5; GO-cr-

RUN105 ACTUAL

LISTA

ERA B:MOVIM

*E

terminada la edición, este archivo CUENTAS.FNC, formará parte de la biblioteca del diskette, en cual también residen los programas:

CMD105.COM

RUN105.COM

ACTUAL.INT

LISTA.COM

En la unidad "B" debe existir el archivo de movimiento MOVIM.-
Para ejecutar el archivo CUENTAS.SUB, se convoca al utilitario CMD105, digitando:

CMD105 CUENTAS-cr-

IMPORTANTE: Siempre deberá estar vigente la unidad "A".-

EJECUCION DE ARCHIVOS COMANDOS (Cont.)

Inmediatamente se realizan los pasos operativos que incluye el archivo CUENTAS.SUB;
o sea:

- Clasifica el archivo MOVIM. El archivo clasificado se denomina MOVIM.CLA y será grabado en el diskette de la unidad "A".-
- Ejecuta el programa de Actualización (ACTUAL.INT).-
- Ejecuta el programa LISTA.COM, que lista los movimientos. Este programa convocará el archivo de movimientos MOVIM.CLA, que está en la unidad "B", para listarlo.-
- Elimina el archivo MOVIM.-

Todos los pasos operativos anteriores se ejecutan en forma secuencial, permitiendo de esta manera simplificar la implementación operativa de un sistema.-
El ejemplo descrito anteriormente, es muy simple, ya que esta rutina de ejecución / de comandos cuenta con posibilidades mayores que la hacen más potente.-
Estas posibilidades son las siguientes:

- Definir datos variables en el Archivo de Comandos.-
- Solicitar Cambio de Diskette.-
- Solicitar Cambio de Formulario.-
- Consultar sobre la continuidad de un proceso.-

DEFINIR DATOS VARIABLES EN

EL ARCHIVO DE COMANDOS

Una aplicación de esta posibilidad es que los nombres de los archivos: MOVIM y // MOVIM.CLA, del ejemplo de Archivo de Comandos, varíen en cada ejecución. El proceso, en esencia, es siempre el mismo. Ahora, para producir estos cambios, en vez de asignar un nombre fijo, se asigna un número (1 al 9) a cada nombre del archivo, anteponiéndole el signo "\$", con lo que queda definida la variable. Por lo tanto, el ejemplo de Archivo de Comandos, con los nombres variables, contiene los siguientes datos:

```
SOR105 I=80 CR; S=B:$1; O=A:$2; K=1,5; GO-cr-  
RUN104 ACTUAL-cr-  
LISTA-cr-  
ERA B:$1-cr-
```

Como ahora los nombres de los archivos son variables, éstos se deben especificar a continuación de la convocación del Archivo de Comandos, es decir que para convocarlo se digita:

```
CMD105 CUENTAS MOVIM.MOVIM.CLA-cr-
```

Con lo que la rutina CMD105 procede a reemplazar dentro del archivo de Comandos /// CUENTAS.SUB, lo siguiente:

```
Todos los archivos $1 por MOVIM  
Todos los archivos $2 por MOVIM.CLA
```

Luego, ejecuta todo el proceso normalmente.-

PAQUETES DE SUBROUTINAS PARA EL MANEJO DE ARCHIVOS EN DISKETTE A MODO MS-101

Las cuatro subrutinas, "ABRIR", "LEER", "CREAR", "GRABAR", que constituyen este paquete, consideradas ampliaciones del compilador COB105, permiten que un programa escrito en el lenguaje COBOL pueda leer y grabar archivos en la modalidad Básica de Intercambio de // Datos, empleada por los equipos MS-101 e IBM-3740 y sus similares.-

La lectura, directamente desde un programa COBOL, de archivos generados en equipos de / input u otros computadores, permite simplificar los sistemas suprimiendo un paso previo de conversión de archivos (en el caso de MS-104, con la rutina CNV104), evitándose la / duplicación innecesaria de información y permitiendo la validación de los datos en el / momento mismo de su lectura. Los registros leídos podrán ser listados, acumulados, trans-

formados todo en un solo paso y directamente desde el diskette de origen.-
La grabación desde programas COBOL de archivos en el Modo Básico de Intercambio de Da- / tos, permite llevar información desde el equipo MS-105 a otro equipo de otra marca y // características que soporte esta modalidad de grabación de datos, cosa que ocurre con la mayoría de los computadores y convertidores.-

Las subrutinas están contenidas en el archivo CRTDRV.REL, el programa accede a ellas // con el mecanismo CALL provisto en el COB105. El programa Link-Editor LNK105 automática- mente incorpora las subrutinas, cuando estas sean empleadas, en el programa objeto. La / transferencia de datos desde el programa principal COBOL y cada una de las subrutinas y viceversa, se realiza a través de áreas fijas definidas en el programa COBOL, en la //// WORKING-STORAGE SECTION.-

MODO DE EMPLEO DE LAS SUBROUTINAS "ABRIR" Y "LEER"

Se dá a continuación un ejemplo de programa en COBOL que lee los datos contenidos en un archivo modo MS-101 y los lista en la impresora.-

```
IDENTIFICATION DIVISION.
PROGRAM-ID LISTA-101,
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES, PRINTER IS IMPRESORA.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 SENAL PIC 99 COMP.
01 AREA-DAT.
    02 NOMBRE PIC x(13) VALUE SPACES.
    02 FILLER PIC x(115).
PROCEDURE DIVISION.
UNO.
    MOVE "DATOS-101" TO NOMBRE.
    CALL "ABRIR",
    IF SENAL NOT = 0 GO TO NO-EXISTE.
PROX-LECTURA.
    CALL "LEER".
    IF SENAL NOT = 1 GO TO FIN-ARCHIVO.
PROCESAR.
    DISPLAY AREA-DAT UPON IMPRESORA.
    GO TO PROX-LECTURA.
FIN-ARCHIVO.
    DISPLAY "FIN ARCHIVO 101".
    STOP-RUN.
NO-EXISTE.
    DISPLAY "NO EXISTE ARCHIVO 'DATOS 101' EN UNIDAD 'B'".
    STOP RUN.
```

El programa tiene las siguientes características, que son fundamentalmente para poder procesar los archivos en diskettes modo 101 colocados en unidad "B":

- a)- Los items de datos "SENAL" y "AREA DAT", deben ser los primeros que se escriben en la WORKING-STORAGE SECTION.-
- b)- "SENAL" es un item binario (PIC 99 COMP) que se emplea para comunicar, a y desde el programa y las subrutinas los sucesivos pasos del proceso.-
- c)- "AREA-DAT" es un item de 128 caracteres de longitud y sirve para transferir la / información, sector a sector, a medida que se los va leyendo.-
El item subordinado "NOMBRE" permite indicar a la subrutina ABRIR, el nombre del archivo que se desea leer.-
- d)- Los pasos a seguir en el proceso, son:
 - I) ABRIR el archivo
 - II) LEER cada uno de los sectores de datos y procesarlos.-
- e)- Para ABRIR un archivo, movemos el nombre del mismo al campo "NOMBRE" y hacemos / un CALL "ABRIR".-
El valor que se encuentra en "SENAL" nos indicará si la apertura fue exitosa o / no.-
SENAL = 00 Todo normal, se puede proceder a LEER la información.-
SENAL = 02 No existe el archivo con el nombre propuesto en el diskette colocado en unidad 'B' (la unidad "B" debe estar vigente o activa en el momento de la ejecución, es decir que al momento de iniciar la ejecución debe aparecer en pantalla la indicación "B ").-
- f)- Si la apertura fue existosa, se podrá proceder a LEER cada uno de los regis- / tros. El programa no deberá alterar los datos presentes en AREA-DAT desde la // apertura correcta de un archivo y la primera lectura, pues los mismos son vitales para el correcto funcionamiento de la subrutina LEER.-
El valor de SENAL debe también permanecer inalterado (o sea = 00).-

- g)- Después de LEER cada registro, el programa deberá consultar el valor en SENAL los valores posibles son:
- 01 - Lectura correcta, datos disponibles en AREA-DAT.
 - 05 - Fin del archivo, no hay más datos (el último registro ya fué procesado).-
- h)- En el programa ejemplo, la parte indicada como PROCESAR se puede reemplazar por el grupo de sentencias que realicen la tarea que se desea ejecutar con los datos según la aplicación o problema que se intente resolver.-
- Los datos podrán grabarse en un archivo, en otra unidad de disco, realizando, / por ejemplo, validaciones, empaquetamiento o compactación de campos numéricos, etc....
- El archivo destino, si existe, podrá ser secuencial, indexado o relativo.-
- i)- Entre cada lectura, el programa podrá utilizar el AREA-DAT sin limitaciones, pero el valor en SENAL debe permanecer siempre igual a 01.-
- j)- La rutina LEER procesa 26 sectores de diskette por cada acceso, es decir, si // bien el programador dispone un nuevo sector de datos luego de cada CALL "LEER", solo 1 de cada 26 veces se realiza la lectura física del disco en "B".-
- k)- El código de lectura (ASCII-EBCDIC) es automáticamente determinado, los datos / en AREA-DAT aparecen siempre en el código ASCII, que es el propio del equipo // MS-105.-
- l)- Sólo se podrá LEER 1 archivo por vez.-
- m)- El tiempo de proceso, para 1800 registros (un diskette completo) es de aproxi-/ madamente 3 minutos, dependiendo de la complejidad del tratamiento de los datos de cada sector.-

MODO DE EMPLEO DE LAS SUBRUTINAS "CREAR y GRABAR"

Se dá a continuación un ejemplo de programa COBOL que graba 50 registros (conteniendo solo asteriscos) en un archivo MS-101.-

IDENTIFICATION DIVISION.

PROGRAM-ID. GRABA 101.

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 SENAL PIC 99 COMP.

01 AREA-DAT.

02 NOMBRE.

03 FILLER PIC x (13) VALUE SPACES.

02 FILLER REDEFINES NOMBRE.

03 LIBRES PIC 99 COMP.

03 FILLER PIC x(11).

02 FILLER PIC x(115).

PROCEDURE DIVISION.

UNO.

MOVE 2 TO SENAL.

MOVE "SALIDA-101" TO NOMBRE.

CALL "CREAR".

IF SENAL NOT=3 GO TO NO-ABRIO.

IF LIBRES < 50 GO TO NO-HAY-LUGAR.

PROCESO.

PERFORM PONE-DATOS 50 TIMES.

CIERRE.

MOVE 7 TO SENAL.

CALL "GRABAR".

STOP RUN.

PONE-DATOS.

MOVE ALL "*" TO AREA-DAT.

CALL "GRABAR".

IF SENAL NOT = 3 GO TO FIN-ANORMAL.

NO-ABRIO.

DISPLAY "NO SE PUDO ABRIR ARCHIVO DESTINO".

DISPLAY "DISKETTE LLENO ???".

STOP RUN.

NO-HAY-LUGAR

DISPLAY "EL ESPACIO DISPONIBLE EN EL DISKETTE".

DISPLAY "ES INSUFICIENTE PARA GRABAR 50 REGISTROS".

STOP RUN.

FIN-ANORMAL.

DISPLAY "NO SE COMPLETO LA GRABACION DEL ARCHIVO".

DISPLAY "CIERRE AUTOMATICO ???".

El proceso que realiza este programa se puede reasumir en el siguiente esquema;

SENAL = 2 NOMBRE = (Nombre del archivo a grabar) CALL "CREAR"

en este momento el programador dispone en LIBRES la cantidad de sectores que es posible grabar en el diskette en unidad "B".

SENAL = 3 AREA-DAT = (Datos a Grabar) CALL "GRABAR".

y, para cerrar el archivo una vez transferidos los datos, con sucesivos CALL "GRABAR" para cada sector a grabar, se debe realizar:

SENAL = 7 CALL "GRABAR"

El programa tiene las siguientes características que son fundamentales para poder grabar archivos a modo MS-101 en diskettes colocados en unidad "B".-

- a)- Los items de datos SENAL y AREA-DAT y los subordinados deben estar al principio y de la WORKING-STORAGE SECTION y respetar exactamente las descripciones y características dadas en este ejemplo.-
- b)- SENAL es un item binario (PIC 99 COMP) que se emplea para comunicar desde y al programa los sucesivos pasos del proceso.-
- c)- AREA-DAT es un item de 128 caracteres de longitud, que se emplea para transferir los datos, sector a sector, para que éstos sean grabados en el diskette por la / rutina GRABAR.-

El item subordinado NOMBRE, permite indicar a la subrutina CREAR el nombre del archivo que se va a grabar.-

El item subordinado LIBRES, contiene despues de una creación exitosa, la cantidad de sectores disponibles en el diskette. Si se va a emplear esta cantidad, / deberá moverse la misma a otro items numérico, puesto que su valor se pierde // en el momento de ubicar información en AREA-DAT. En LIBRES se podrá indicar la longitud de registro al momento del cierre del archivo, (ver mas adelante).-

- d)- Los pasos a seguir en el proceso son:
 - I) CREAR el archivo
 - II) Ubicar los datos en AREA-DAT y GRABAR cada uno de los sectores
 - III) Cerrar al archivo. (señal = 7 y un último "CALL GRABAR")
- e)- Para crear el archivo, el programa ubicará el nombre del archivo en el campo /// NOMBRE, y en SENAL un valor 02 y luego ejecutará la subrutina con la instrucción CALL "CREAR".-

El valor presente en SENAL nos indicará si la creación fué exitosa o no:

- SENAL=03 Operación correcta, se puede proceder a la grabación del archivo.-
- SENAL=04 Se produjo un error de lectura del directorio del diskette en unidad "B", el archivo no ha sido creado.-
- SENAL=05 Ya hay un archivo abierto y en proceso de grabación , solo se puede operar sobre un archivo por vez.-
- SENAL=06 No hay lugar libre en el diskette (no hay espacio para más datos o el directorio está agotado).-

f)- Si el valor en SENAL es 03, indicando que la creación del archivo fue exitosa, se procederá a GRABAR uno a uno los sectores del archivo mediante sucesivos / CALL "GRABAR".-

Inmediatamente después de una creación exitosa, se dispone en LIBRES de la cantidad de sectores que se han asignado al archivo creado.-

g)- El valor en SENAL debe ser de 03 durante el proceso de transferencia de información. Los valores posibles en SENAL son:

Antes de CALL "GRABAR"	SENAL = 03	Proceso normal de transferencia de datos.
	SENAL = 07	Se va a cerrar el archivo. (Cierre no // automático).
Después de CALL "GRABAR"	SENAL = 03	Listo para aceptar más información.
	SENAL = 07	Se acaba de cerrar el archivo (automático o no).
	SENAL = 08	Se hizo CALL "GRABAR" pero no había archivo creado activo.-

h)- En el programa ejemplo, los datos a grabar surgieron simplemente de una operación MOVE, pero normalmente los datos provendrán de un archivo, y la parte denominada PROCESO en este ejemplo, podrá adquirir una forma mucho más compleja, incluyendo apertura lectura y cierre de archivos, ya sean estos secuenciales, indexados o relativos. El programa puede operar libremente sobre el campo AREA-DAT, pero debe respetar el valor en SENAL.

i)- Consideraciones sobre el cierre del archivo:

I) Automático: Cuando después de sucesivos CALL "GRABAR" se agote el espacio disponible en el diskette, la subrutina "GRABAR" / automáticamente procederá al cierre del archivo e indicará en SENAL = 07 que se ha producido el cierre.-

II) No automático: Cuando el programa principal haya terminado de transferir los datos, para cerrar el archivo colocará el valor SENAL = 07 y hará un último CALL "GRABAR". No / hay transferencia de datos en este último paso.-

- j)- El código (ASCII-EBCDIC) de grabación de los datos, se determina en el momento de la creación del archivo, de modo automático. El programa principal colocará los / datos en AREA-DAT en el código interno del equipo MS-105 (ASCII) y la rutina GRABAR se encargará de traducir la información si corresponde.-
- k)- La determinación del espacio libre se hace de la siguiente manera:
Los archivos activos preexistentes en el diskette con respetados y se ubica el Fin de Archivo más alto.-
El área libre será el que quede entre el próximo sector después del Fin de Ar-/ chivo más alto y la pista 74 sector 26.-
No se puede emplear el espacio que quede libre entre dos archivos.-
Un diskette totalmente vacío se grabará desde la pista 01 hasta la pista 74 sector 26 o sea un total de 1898 sectores.-
- l)- Consideraciones sobre longitud de registro:
Cuando el archivo grabado deba contener en su entrada de directorio el valor real de la longitud de registro, se debe emplear el cierre no automático, ubicando en LIBRES el valor de la longitud de registro deseada. Es responsabilidad del programa principal asegurar la correcta correspondencia entre el valor dado como longitud de registro y la configuración de datos presente en cada uno de los sectores del archivo.-
En especial, será responsable de cumplimentar con el relleno de los caracteres / libres con el valor 00 hexadecimal para respetar la convención IBM (equivalente a SOFT06V4 del equipo MS-101).-
Si se produce el cierre automático, la longitud de registro en el directorio del diskette será de 128.-
En el caso de cierre no automático, se verifica que el valor en LIBRES sea igual

CONSIDERACIONES GENERALES

Los ejemplos dados, de programas COBOL, indican el tipo de datos y procesos que se pueden emplear. El nombre de los items de datos no es de ninguna importancia, pero si lo es para la disposición, (al principio de la WORKING-STORAGE SECTION) orden, / tamaño y tipo.-

Durante el proceso de lectura, con la subrutina LEER, se consideran DELETED aquellos sectores de datos con 00 hexadecimal en la primera posición del registro, convención empleada por los equipos MS-101 y los datos no serán transferidos saltándose el sector.-

En particular, los sectores con "D" en la primera posición, según la convención IBM, no son considerados como DELETED.-

Será conveniente por esta razón, evitar la presencia de campos empaquetados (COMP-3) en el primer byte de cada registro.-

Las subrutinas "ABRIR" y "LEER" se emplean conjuntamente para leer un archivo por vez, y "CREAR" y "GRABAR" se emplean conjuntamente para grabar un archivo por vez, no se puede leer un archivo y simultáneamente grabar otro, tampoco se podrá procesar más de un archivo por vez, tanto en lectura como en grabación.-

Como se procesan siempre 26 registros en bloque en cada acceso físico al diskette, se obtiene un buen rendimiento, para mejorar aún más la performance, conviene que el programa emplee técnicas de buffer (tablas para almacenar registros) tanto para lectura como para grabación de archivos secuenciales, indexados o relativos de MS-105.-

El diskette origen o destino, según se lea o grabe información siempre estará ubicado en la unidad "B". Esta debe ser la unidad vigente al momento de la ejecución del programa (en la pantalla debe aparecer B> al momento de convocar el programa).-

SUBROUTINA "VIDEO" PARA MANEJO DE LA TERMINAL DIALOGUE - 80

Esta subrutina realiza la secuencia de escape según caracteres de control que provee el programa principal, de modo que se puedan utilizar todas las características de la terminal Dialogue'80 de los equipos MS-105, desde programas escritos en lenguaje CO-/BOL.-

Para convocar esta subrutina se empleará la sentencia:

CALL "VIDEO" USING CAMPO-1

CAMPO-1 será un campo cualquiera definido en la WORKING-STORAGE SECTION, de la siguiente manera: 77 CAMPO-1 PIC X.

La subrutina "VIDEO" está contenida dentro del archivo CRTDRV.REL y será automáticamente cargada en el momento de la Link-Edición con el LNK105.-

El siguiente ejemplo muestra un trozo de programa que escribe un mensaje en la modalidad invertida (fondo blanco y caracteres en negro).-

INVERTIDO

```
MOVE "j" TO CAMPO-1
CALL "VIDEO" USING CAMPO-1
DISPLAY "ESTE ES EL MENSAJE"
MOVE "k" TO CAMPO-1
CALL "VIDEO" USING CAMPO-1
```

La letra "j" (minúscula) es el caracter de control que indica a la terminal que debe representar todos los mensajes, que se envíen desde ese momento, a la modalidad invertida. La letra "k" (minúscula) anula o cancela el efecto del anterior comando, volviendo a funcionar en el modo normal. El mensaje, sin embargo, quedará representado al modo invertido.-

En general todos los caracteres de control, que definen una forma de funcionamiento, tienen el caracter de control opuesto, que suspende el efecto.-

Para trabajar con el Editor de Textos en minúscula y mayúscula se procede de la siguiente forma :

Se convoca al Editor de Textos en mayúscula :

A EDT105 PROG.COB

y se dan los comandos del mismo en minúscula, o sea que la tecla CAPS-LOCK debe estar apagada, por ejemplo :

* i

* s

luego se podrán dar letras mayúsculas y minúsculas presionando sólo la misma tecla ó NUM SHIFT.-

T A B L A D E C A R A C T E R E S D E C O N T R O L

#	(Numeral)	Bloquear teclado:	El teclado queda inactivo.-
"	(Comillas)	Desbloquera teclado:	El teclado vuelve a quedar activo.-
&	(Ampersand)	Modo protegido:	Se suspende el rotado de pantalla, los campos escritos a media intensidad quedan protegidos.
'	(Comilla)	Modo desprotegido:	Los campos escritos a media intensidad dejan de estar protegidos, se reinicia el rotado normal de pantalla.-
)	(Paréntesis)	Media intensidad:	Los caracteres son representados a media intensidad.-
((Paréntesis)	Intensidad normal:	Los caracteres son representados a intensidad normal.-
F	(Letra efe mayúscula)	Primera página:	Representar la página 1.-
N	(Letra ene mayúscula)	Próxima página:	Representa la página siguiente.-
+	(Signo suma)	Borrar:	Deja en blanco la pantalla y la memoria.-
:	(Dos puntos)	Borrar desprotegido:	Deja en blanco la parte no protegida de la pantalla.-
Y	(Letra y griega mayúscula)	Borrar página:	Deja en blanco la página.-
j	(Letra j minúscula)	Video invertido:	Representa letras oscuras sobre fondo blanco.-
k	(Letra k minúsc.)	Video Normal:	Representa letras blancas sobre fondo oscuro.-
l	(Letra l minúsc.)	Subrayado:	Cada caracter se representa subrayado.-
n	(Letra ene minúsc)	Intermitente:	Representa los caracteres en modo intermitente.-

PAGINADO: La terminal DIALOGUE'80 tiene dos memorias internas, denominadas páginas, que pueden ser empleadas alternativamente.-
Normalmente, si no se emplea la opción de paginado, la terminal / trabaja solo sobre la página 1.-
Cuando se pasa a trabajar con otra página, la anterior permanece sin cambios hasta que se la vuelve a utilizar. Los comandos son / propios de la terminal y tienen efecto cualquiera sea la página / empleada o los cambios de página que se hagan.-

ENCUADERNACION
RUGGIERI
T. E. 71-5520

ENCUADERNACION
RUGGERI
T. E. 71-5520

