

UNIVERSITAT
DE
BARCELONA

| | | | |
|-------|----|----------|-------|
| W | VV | MM | MM |
| W | VV | MMM | MMM |
| W | VV | MM MM | MM MM |
| VV | VV | MM MM MM | MM |
| VV VV | | MM MMM | MM |
| VVV | | MM M | MM |
| V | | MM | MM |

VM/SP

CENTRE DE CALCUL

guia de iniciación

al uso del sistema VM/CMS

.....



UNIVERSIDAD DE BARCELONA
CENTRO DE CALCULO



UNIVERSIDAD DE BARCELONA
CENTRO DE CALCULO

GUIA DE INICIACION

AL USO DEL SISTEMA VM/CMS

Sección de Aplicaciones Científicas

Barcelona, Abril de 1983

Esta Guía de Iniciación constituye un elemento de base en el contexto de las publicaciones del Centro de Cálculo orientadas a la formación y documentación de sus usuarios. Se ha diseñado y escrito, paralelamente a los trabajos de programación de los cursos intensivos de iniciación de usuarios, con dos objetivos complementarios: constituir un texto de apoyo en la impartición de dichos cursos y también una guía que permita empezar a trabajar a las personas desconocedoras de las características del sistema actual del Centro de Cálculo.

Se ha pretendido conseguir una publicación eminentemente pedagógica, por lo que en ningún caso debe considerarse como un manual completo sobre el sistema VM/SP. Tan sólo se han elegido los elementos mínimos imprescindibles que permitan a un usuario el inicio de trabajos sencillos de edición, compilación y ejecución de programas, así como la utilización de los paquetes de aplicaciones científicas disponibles en estos momentos. Todo ello, incluso, sin haber asistido a un curso de iniciación, simplemente sentándose ante una terminal y realizando las operaciones que va indicando el texto.

Dada la variabilidad en el tiempo de algunos de los componentes de la instalación del Centro de Cálculo, es posible que se hayan producido modificaciones en alguno de los elementos o instrucciones contenidas en esta Guía en el momento de utilizarla. Para subsanar este problema se cuenta por una parte con la previsión de versiones revisadas con una cierta periodicidad y el acceso inmediato a todas las posibles novedades a través de terminal, mediante el comando HELP GUIA INICIO que proporciona el texto actualizado de esta Guía.



Confiamos en que sea de utilidad para los fines previstos y agradeceremos cualquier sugerencia y aportación que nos ayude a mejorarla, teniendo siempre presente que la mejor valoración de sus aciertos y defectos está en manos de aquellos que realmente la utilicen.

Barcelona, Abril de 1983.



INDICE

| | |
|---|----|
| INTRODUCCION | 1 |
| COMO UTILIZAR LAS TERMINALES DE TIPO IBM 3278 ? | 5 |
| Puesta en marcha de las terminales | 5 |
| Control de contraste e intensidad | 5 |
| Interruptores 'Normal/Test' y 'A,a/A' | 5 |
| El cursor | 5 |
| Teclado | 5 |
| Modos, estados y areas de la pantalla | 11 |
| Introducción de datos en modalidad de línea a línea | 11 |
| Borrado de la pantalla | 12 |
| COMO ES EL SISTEMA OPERATIVO ? | 13 |
| Acceso al ordenador: LOGON | 13 |
| Estructura del sistema operativo | 13 |
| Comandos | 14 |
| Estados del sistema | 15 |
| COMO PEDIRLE INFORMACION AL SISTEMA ? | 17 |
| COMO UTILIZAR FICHEROS ? | 20 |
| Concepto de fichero | 20 |
| Ubicación de un fichero | 20 |
| Concepto de minidisco | 20 |
| Tipos de disco | 20 |
| Identificación de un fichero en disco | 22 |
| Orden de búsqueda estandar de CMS | 23 |
| Directorios - FLIST | 23 |
| Visualización de ficheros - BROWSE | 26 |
| Comandos básicos para el manejo de ficheros | 28 |
| Utilización de la barra ('/') en FLIST | 29 |
| MOVIMIENTO, IMPRESION E INTERCAMBIO DE FICHEROS. EL SPOOL | 31 |
| Disco-disco: copia de ficheros entre áreas de disco del usuario | 31 |
| Disco-cinta: utilización de cintas | 31 |
| Disco-impresora: impresión de ficheros | 34 |
| Usuario-usuario: transmisión de ficheros entre usuarios | 38 |
| Envío de ficheros a un usuario | 38 |
| Recepción de ficheros: la lectora virtual | 40 |
| Registro de las actividades de consola: la 'consola virtual' | 42 |
| Observación del estado de su consola y sus comunicaciones | 43 |



| | |
|---|----|
| COMO CREAR Y MODIFICAR FICHEROS: EL EDITOR | 44 |
| Algunos subcomandos del Editor aplicados a dos ejemplos | 44 |
| COMO MODIFICAR LOS COMPONENTES DE SU MAQUINA VIRTUAL ?.. | 50 |
| COMO COMPILAR Y EJECUTAR UN PROGRAMA ? | 51 |
| Compiladores | 51 |
| Ejecución: LOAD, START, RUN. El comando FILEDEF.Las | |
| TXTLIBs | 52 |
| Proceso en Batch | 56 |
| COMO UTILIZAR PROGRAMAS Y PAQUETES DE APLICACIONES ? ... | 63 |
| Paquetes de subrutinas | 63 |
| Programas y paquetes de programas | 64 |
| COMO ELIMINAR TAREAS REPETITIVAS MEDIANTE LA CREACION DE | |
| MODULOS Y DE FICHEROS DE COMANDOS DEL SISTEMA (EXEC'S)?. | 71 |
| Módulos | 71 |
| EXEC's | 71 |
| MANUALES Y REFERENCIAS PARA EL ESTUDIO POSTERIOR | 74 |
| APENDICE A: LISTA DE LOS COMANDOS MAS UTILIZADOS | 78 |
| Comandos de CMS | 78 |
| Comandos de CP | 80 |
| APENDICE B: LISTA DE LOS SUBCOMANDOS DEL EDITOR (XEDIT) | |
| MAS UTILIZADOS | 82 |
| Subcomandos | 82 |
| Operandos del subcomando SET | 84 |
| Subcomandos utilizables desde el área de prefijo .. | 86 |
| Significado inicial de las teclas FPr | 86 |
| APENDICE C: SESIONES DE EJEMPLO | 88 |



INDICE DE TABLAS Y FIGURAS

| | |
|---|----|
| * Figura 1. Sistema VM/SP | 3 |
| * Figura 2. Máquinas virtuales de los usuarios de un Departamento | 4 |
| * Figura 3. Interruptores y controles de la pantalla | 6 |
| * Figura 4.1. Anagrama de la instalación que aparece al conectar la pantalla | 8 |
| * Figura 4.2. Areas de la pantalla | 8 |
| * Figura 5. Teclado de terminal IBM 3278 en su modalidad 'máquina de escribir' | 9 |
| * Figura 6. Componentes de una máquina virtual | 9 |
| * Tabla de significados de las teclas FP al utilizar el comando HELP | 18 |
| * Figura 7.1. Información que aparece al introducir HELP CMS MENU | 19 |
| * Figura 7.2. Información obtenida mediante HELP COPY | 19 |
| * Figura 8. Disco real subdividido en minidisos | 21 |
| * Figura 9. Características de los discos utilizados por una máquina virtual obtenida mediante el comando CP QUERY DISK | 25 |
| * Figura 10. Directorio del disco A proporcionado por FLIST | 25 |
| * Tabla de significados de las teclas FP al utilizar el comando FLIST | 26 |
| * Figura 11. Movimiento, impresión e intercambio de ficheros ... | 32 |
| * Figura 12. Copia de ficheros de cinta a disco y viceversa | 35 |
| * Figura 13. Intercambio de ficheros entre usuarios | 39 |
| * Figura 14.1. Cola de ficheros enviados a la impresora virtual antes de ser imprimidos por la impresora real | 41 |
| * Figura 14.2. Cola de ficheros situados en la lectora virtual de un usuario | 41 |
| * Figura 15. Creación de un fichero | 46 |
| * Figura 16. Creación de un fichero | 46 |



| | |
|---|----|
| * Figura 17. Modificación de un fichero | 49 |
| * Figura 18. Modificación de un fichero | 49 |
| * Tabla de comandos GLOBAL según el lenguaje de programación utilizado | 53 |
| * Tabla de opciones RECFM para el comando FILEDEF | 55 |
| * Figura 19. Esquema de utilización de la posibilidad BATCH | 58 |
| * Figura 20. Panel de opciones de BATCH | 59 |
| * Figura 21. El trabajo se envía a BATCH al teclear ENVIO como respuesta a la instrucción que aparece en pantalla | 59 |
| * Figura 22.1. El comando QQ proporciona información, actualizada cada 5 minutos, de las colas de BATCH | 61 |
| * Figura 22.2. Estado de las colas del BATCH 1 | 61 |
| * Figura 23.1. Fichero de datos preparado para ser procesado por el programa BMDP 6D | 65 |
| * Figura 23.2. Listado de consola enviado por la máquina BATCH .. | 65 |
| * Lista de procedures utilizables desde el paquete SCSS | 69 |
| * Tabla de significados de las teclas FP al utilizar el editor XEDIT | 87 |

—oOo—



e los
ones y
eder a
arios

NOTA: Esta Guía ha sido elaborada utilizando las facilidades de edición del sistema VM/CMS y una impresora de caracteres que no dispone de acentos. Apelamos a la benevolencia del lector para que disculpe esta anomalía.



INTRODUCCION

=====

La Universidad de Barcelona pone a disposicion de los usuarios del Centro de Calculo una serie de instalaciones y recursos informaticos a cuya utilizacion pueden acceder a traves de las terminales instaladas en la sala de usuarios asignada a la Facultad, Escuela o Servicio correpondiente.

En las 'Normas provisionales de utilizacion de los recursos informaticos del Centro de Calculo de la Universidad de Barcelona' (vease la referencia [NORM]), que pueden consultarse en cada una de las salas de usuarios y en el Centro de Calculo, se proporciona una descripcion detallada de dichas instalaciones asi como una serie de normas y recomendaciones para su utilizacion. Asi se especifica como cada usuario debe solicitar un codigo individual que le permitira acceder a una serie de recursos propios y compartidos, los horarios de servicio, los protocolos para obtener los listados de impresora que produzca o para utilizar cintas magneticas, el regimen de proceso interactivo o en 'batch', etc.

Descrito de forma muy general, el equipo informatico mencionado consta de un procesador IBM 4341/M02 con 8 Mbytes de memoria real, tres unidades de disco 3370 con un total de 1710 Mbytes y su correspondiente unidad de control 3880/1, dos unidades de cinta magnetica 3420/3 (800,1600 bpi) y su unidad de control 3803, una impresora 3203/5 (1200 lpm) y una impresora de puntos 3287 (80 cps), una unidad de control de lineas locales 3274 que puede soportar hasta 32 terminales y una unidad de control de de estaciones remotas 3704/A01 que controla la conexion, a traves de linea telefonica, con estaciones conectadas a su vez con varias terminales situadas en salas de usuarios distantes del Centro de Calculo. Las terminales instaladas en las salas de usuarios son del tipo IBM 3278. Tambien dispone de una pantalla grafica Tektronix 4113 y un 'plotter' Benson 1333.

El 'software' disponible en la actualidad consta, entre otros elementos, del sistema operativo VM/SP (Virtual Machine/System Product) y los productos CMS (con editor XEDIT y lenguaje de control EXEC2), VM/370 RSCS, IPF, VMAP, VSAM/AMS, DMS/CMS, IFS, DOS/VS SORT/MERGE. Tambien se pueden utilizar los compiladores y librerias de OS/VS COBOL, PL/I OPTIMIZING, FORTRAN IV G1, FORTRAN VS, PASCAL, EP/VS. Los principales programas y paquetes de aplicaciones disponibles son: SSP (Scientific Subroutine Package) en Fortran y PL/I (coleccion de subrutinas numericas y estadisticas), NAG (Numerical Algorithms Library Mark 9) en Fortran (Coleccion de 450 subrutinas numericas y estadisticas), BMDP (Biomedical Statistical Software) (paquete de 45 programas con la mayor parte de metodos



estadísticos), SCCS (Conversational System Data Analysis) (versión conversacional del SPSS; permite el tratamiento estadístico de ficheros de datos de forma interactiva y autoexplicativa), CLUSTAN (Cluster Analysis Package), CSMP (Continuous System Modelings Program) (Lenguaje de simulación de sistemas continuos).

Todas estas instalaciones y recursos se van incrementando progresivamente a medida que crecen las necesidades de los usuarios. Por ello existen una serie de programas y paquetes en fase de implementación cuya disponibilidad se irá anunciando oportunamente.

El sistema operativo VM/SP trabaja con una filosofía denominada de 'maquina virtual', consistente en dedicar a cada usuario parte de los recursos y dispositivos reales enumerados antes, de manera que en la práctica cada usuario individual accede a ellos como si dispusiera de un ordenador (con los correspondientes periféricos) para el solo. El usuario no tiene que preocuparse de como el sistema gestiona el reparto de recursos reales a partir de los cuales simula las 'maquinas virtuales' de todos los usuarios que están conectados y trabajando en un momento dado (vease fig. 1).

Actualmente cada usuario dispone de una maquina virtual con 512k de memoria central, un disco temporal (accedido con modo A; vease mas adelante la descripción del modo de acceso a discos) y una consola (la propia terminal), lectora, perforadora e impresora virtuales. También tiene acceso a un disco permanente (accedido con modo D), compartido por todos los usuarios de su departamento, donde se pueden almacenar ficheros y guardarlos de una sesión a otra (vease fig. 2). Esta configuración es modificable tal como se especifica mas adelante.

A continuación se describe de una forma sistemática y progresiva como empezar a utilizar estos recursos y se dan las referencias adecuadas (vease 'Manuales y referencias para el estudio posterior') para poder profundizar en el conocimiento del sistema y usar todas las opciones de sus comandos, consiguiendo así la máxima eficacia de los procesos.



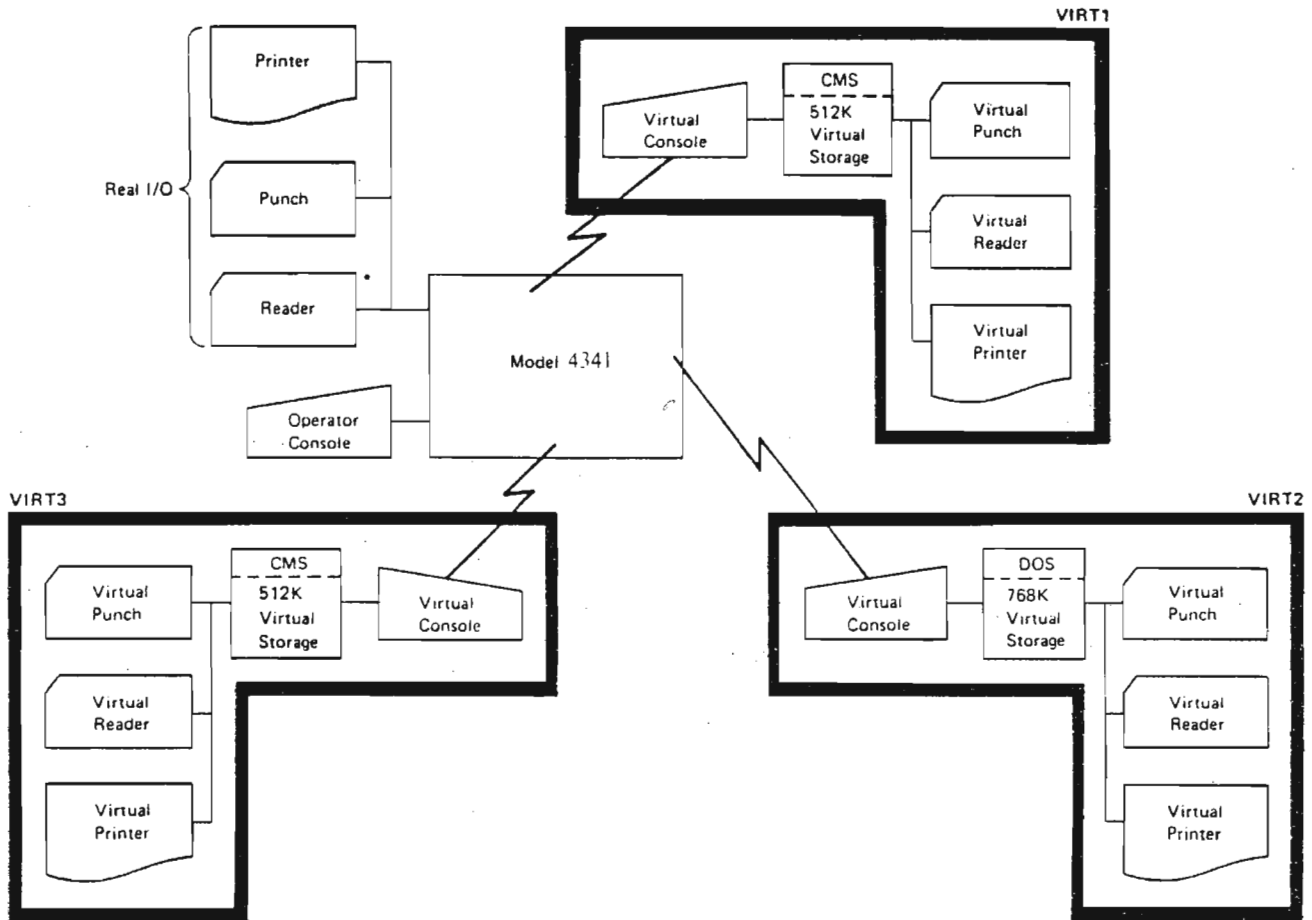


Fig. 1. Sistema VM/SP. (obsérvese que la máquina virtual Virt 2 trabaja bajo sistema operativo DOS mientras las otras dos lo hacen bajo CMS. En esta instalación todas las máquinas virtuales de los usuarios científicos trabajan bajo CMS).



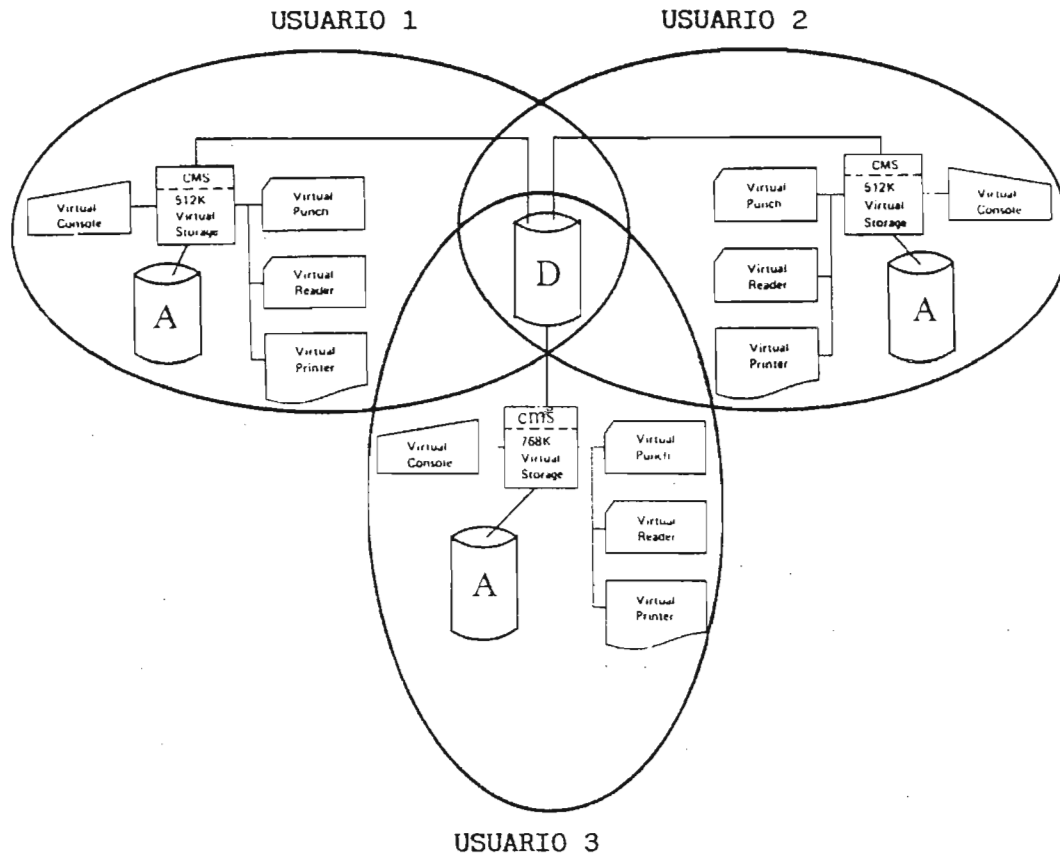


Fig. 2. Máquinas virtuales de los usuarios de un Departamento.



COMO UTILIZAR LAS TERMINALES DE TIPO IBM 3278 ?

Puesta en marcha de las terminales

Las terminales IBM 3278 tienen una serie de interruptores y diales cuyo efecto se especifica en la figura 3.

Para poner en marcha la terminal, accione el interruptor rojo situado a la izquierda de la pantalla, y espere hasta que salga el mensaje 'VM/370 ONLINE' y el anagrama de la instalacion (fig. 4).

Control de contraste e intensidad

Los dos diales situados a la derecha de la pantalla (fig. 3) sirven para controlar el contraste (dial superior) y la intensidad de la pantalla (dial inferior).

Interruptores 'Normal/Test' y 'A,a/A'

El interruptor 'Normal/Test' (vease fig. 3) sirve para verificar el buen funcionamiento de la pantalla. Debe encontrarse siempre en la posicion 'Normal'.

El interruptor 'A,a/A' (vease fig. 3) sirve para controlar la representacion en pantalla de las minusculas. Si se encuentra en la posicion 'A,a', se representan en pantalla mayusculas y minusculas. Si se encuentra en la posicion 'A', las letras minusculas se representan como mayusculas (aunque internamente se tratan como minusculas).

El cursor

El cursor es una señal que indica la siguiente posicion en la que se escribirán los datos. Cada vez que se introduce algun caracter (símbolo) desde el teclado, aparece en la pantalla en la posicion que ocupaba el cursor, y el cursor se desplaza hasta la siguiente posicion. (La forma del cursor puede variarse mediante las teclas 'PARPA CURSR' y 'ALT CURSR' - vease mas adelante).

Teclado

El teclado de las terminales es similar al de una



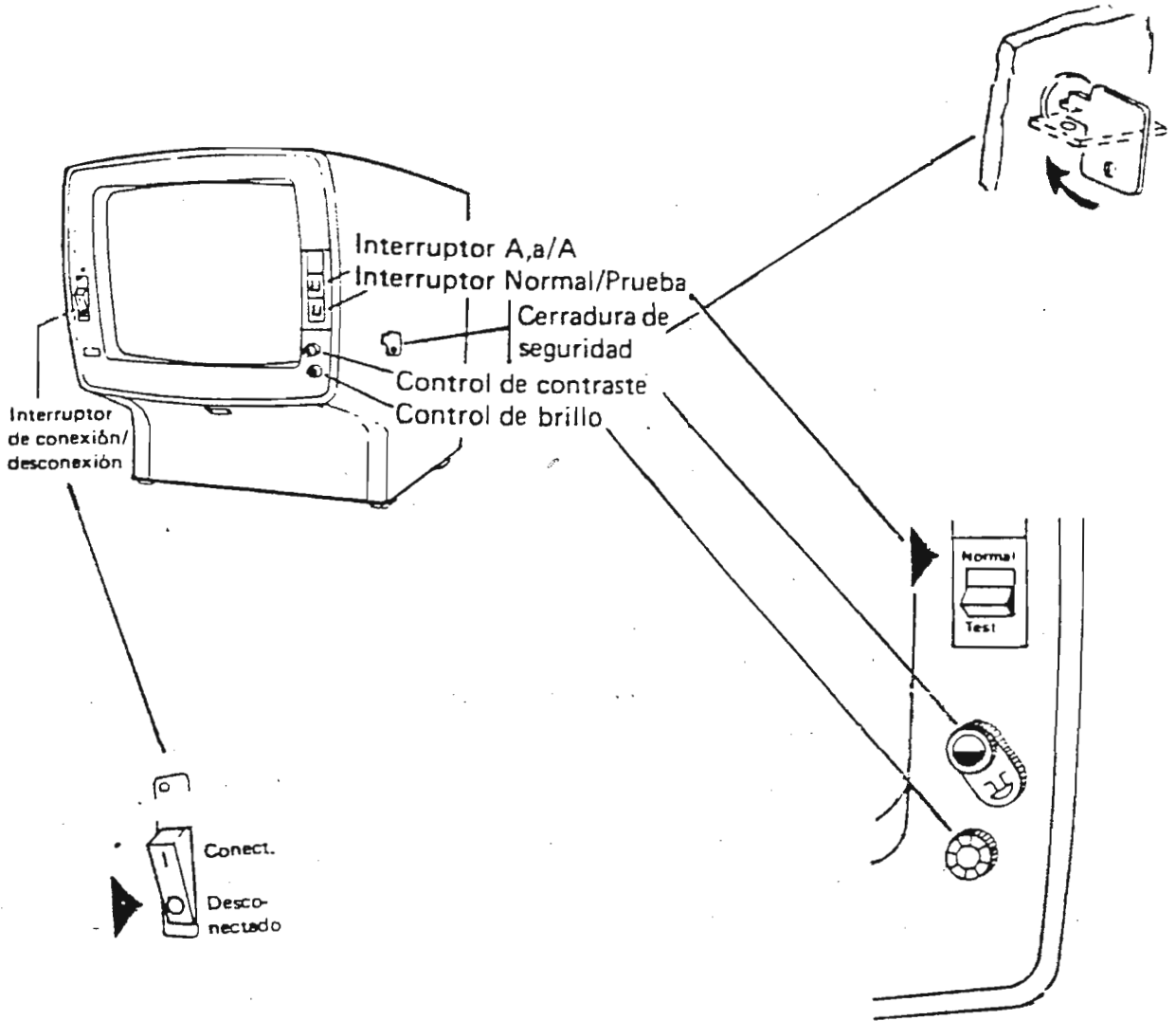


Fig. 3. Interruptores y controles de la pantalla.



maquina de escribir (vease fig. 5). Algunas teclas adicionales permiten realizar funciones que no tendrian sentido en una maquina de escribir (como desplazar el cursor) o comunicarse directamente con el ordenador:

La tecla 'RESTAUR' desbloquea el teclado cuando se produce alguna condicion de error señalada por el bloqueo del teclado y la aparicion de simbolos de error bajo la linea continua (que separa el area de estado de la pantalla que se explica mas adelante [vease fig. 4]).

La tecla 'ALTVA' permite utilizar las funciones indicadas por las inscripciones laterales que tienen algunas teclas, pulsandola simultaneamente, como en el caso de la tecla 'BORRA'.

La tecla 'PRBA' (pulsada simultaneamente con la 'ALTVA') permite realizar pruebas con el teclado. Como en el caso del interruptor 'Normal/Test', explicado mas arriba, no debe utilizarse habitualmente. Al pulsar esta tecla, aparece el indicador 'TEST' bajo la linea continua, la pantalla se borra y el cursor se posiciona en el angulo superior izquierdo de la pantalla. Puede escribirse sin restriccion sobre toda la pantalla (como se explicara mas adelante, esto no sucede normalmente). [Pueden probarse las teclas descritas a continuacion despues de pulsar la tecla 'PRBA'. Al acabar la prueba, devuelvase la pantalla a su estado normal pulsabndo de nuevo la tecla 'PRBA']

La tecla de eliminacion de ruido (parte superior de la tecla 'PRBA') permite controlar el ruido producido al pulsar las teclas. Por atencion al resto de los usuarios, se recomienda trabajar sin ruido.

La tecla 'BORRA' (pulsada simultaneamente con la 'ALTVA') borra el contenido de la pantalla.

La tecla 'BORR FDC' borra todos los caracteres desde el cursor hasta el final del area de entrada (vease mas adelante).

La tecla de supresion (marcada con una a tachada) suprime el caracter sobre el cual se encuentra posicionado el cursor, desplazando los siguientes caracteres un lugar a la izquierda.

La tecla de insercion (marcada con una a con acento circunflejo) coloca la pantalla en modo de insercion; los caracteres que se introducen a continuacion no se superponen a los ya existentes, sino que los desplazan un lugar a la derecha.



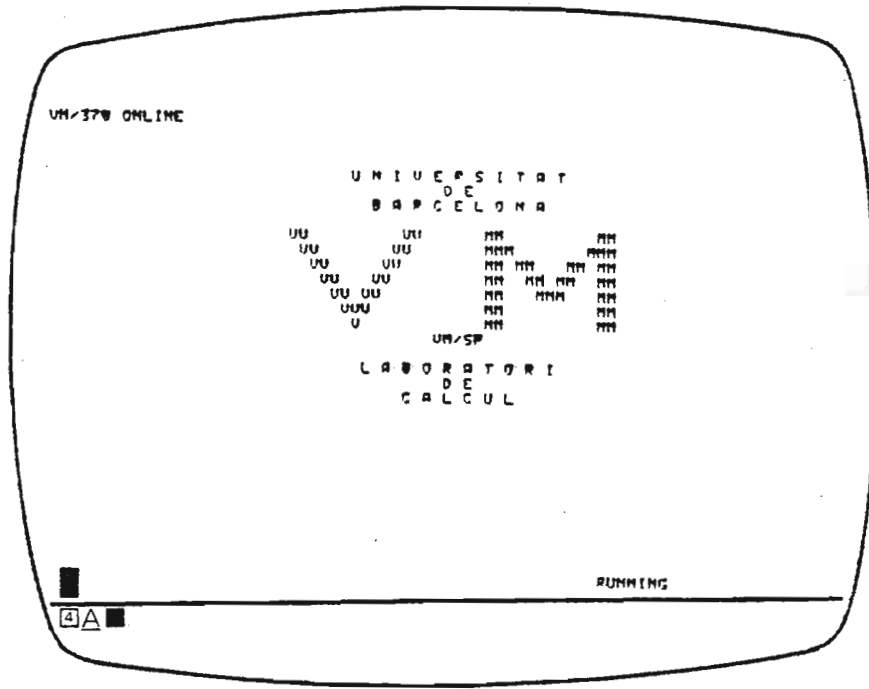


Fig. 4.1. Anagrama de la instalación que aparece al conectar la pantalla.

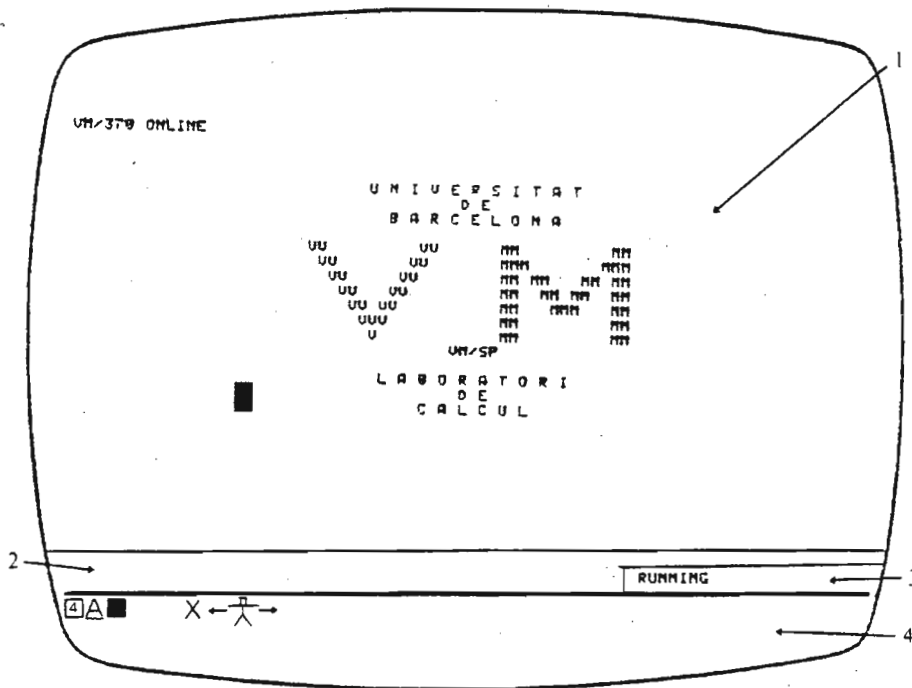


Fig. 4.2. Areas de la pantalla: 1. Área de Salida; 2. área de entrada de datos; 3. área de estado de la máquina virtual; 4. área de estado de la pantalla. Obsérvese los símbolos que aparecen en esta última área al intentar introducir datos en el área de salida en el punto indicado por el cursor en su modalidad rectangular.



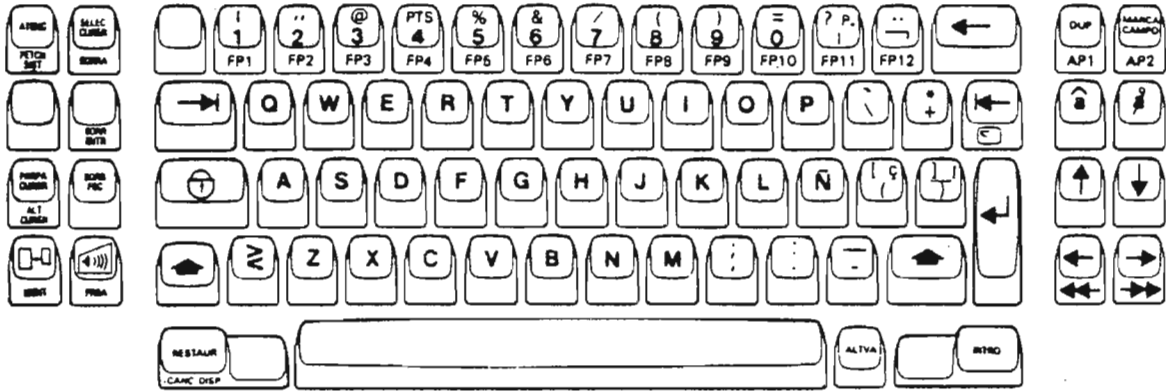


Fig. 5. Teclado de terminal IBM 3278 en su modalidad "máquina de escribir".

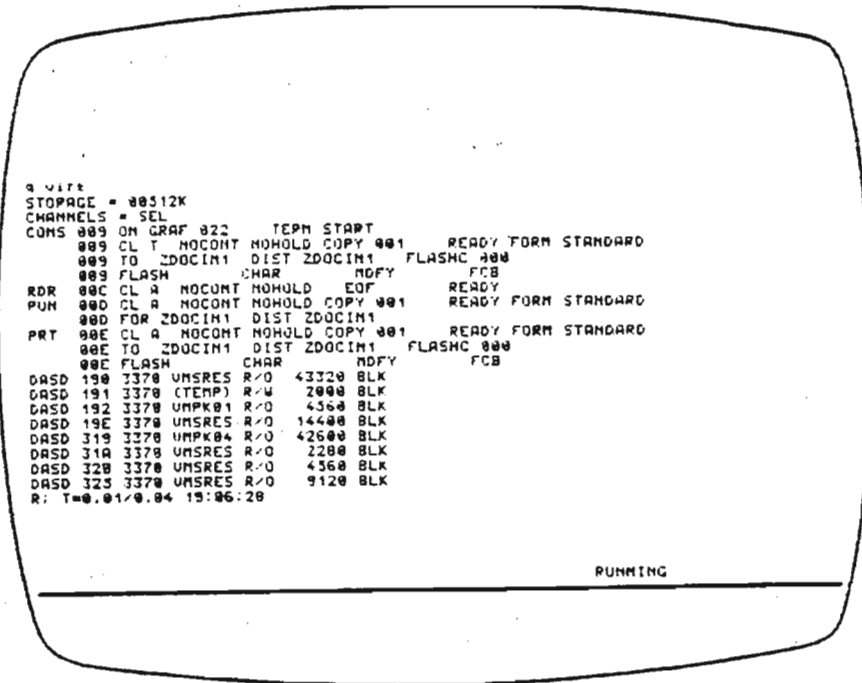


Fig. 6. Componentes de una máquina virtual.



NOTAS: se distingue entre 'espacios blancos' y 'espacios vacios'. En general, los espacios de un area de entrada estan 'vacios' y pueden 'llenarse' mediante caracteres cualesquiera (en particular, mediante el caracter 'espacio en blanco', que ocupan un lugar en la pantalla). Las teclas de desplazamiento del cursor lo mueven sin emitir ningun caracter; la barra de espaciado emite un caracter blanco. Los espacios vacios son ignorados completamente por el ordenador, mientras que los espacios en blanco pueden tener alguna utilidad.

Para que el desplazamiento de los caracteres en el modo de insercion pueda tener lugar, es necesario que haya posiciones 'vacias' al final de la linea en la que se esta insertando. Si esto no sucede, el teclado se bloquea - puede desbloquearlo utilizando la tecla 'RESTAUR', aunque se pierde el modo de insercion.

El modo de insercion queda indicado en la pantalla por la aparicion bajo la linea continua de un acento circunflejo. Puede suprimirse el modo de insercion mediante la tecla 'RESTAUR'.

Las cuatro teclas marcadas con flechas sirven para desplazar el cursor. Son utiles principalmente al utilizar el editor en modo 'full screen' (vease mas adelante).

La tecla 'INTRO' sirve para señalar la introduccion de informacion; su utilizacion se discute en el apartado 'introduccion de datos en la modalidad de linea a linea', mas adelante en esta misma seccion.

Las doce teclas FP n (n entre 1 y 12) no tienen un significado fijo. Cada programa puede definir su funcion: asi, el editor XEDIT o el BROWSE (vease mas adelante) tienen asignadas las teclas FP7 y FP8 a 'paso de pagina hacia atras' y 'paso de pagina hacia adelante', respectivamente. Tambien es posible asignar a estas teclas funciones especificas del sistema operativo. (Vease mas adelante, en la seccion de 'comandos', y utilicese el comando CP SET).

La tecla 'AP1' permite acceder directamente a la componente CP del sistema operativo (vease mas abajo).

La tecla 'AP2' realiza diversas funciones segun el programa que se esta utilizando (en cierto sentido, es como una de las teclas FP, aunque generalmente no se puede definir directamente su efecto).



Modos, estados y areas de la pantalla

La pantalla contiene 24 líneas de 80 caracteres sobre una línea continua, y una línea o área de 'estado de la pantalla' bajo la línea continua (veanse figs. 4.1 y 4.2). La línea de estado de la pantalla puede contener diversos indicadores. Algunos de ellos reflejan el modo de utilización de la pantalla (por ejemplo, si se utiliza el modo de inserción [vease mas adelante] aparece un acento circunflejo en el centro de esta línea; si el teclado esta utilizando mayusculas, aparece una flecha hacia arriba); otros indican la presencia de diversas condiciones de error, que suelen bloquear el teclado (por ejemplo, al intentar escribir sobre un área protegida aparece un hombrecillo enmarcado por flechas; debe pulsarse 'RESTAUR' para desbloquear el teclado: vease la figura 4.2).

La pantalla puede funcionar de dos modos: línea a línea o en modalidad de pantalla completa (full screen). La modalidad de pantalla completa se utiliza, por ejemplo, en el editor (vease mas adelante), pero el modo habitual de funcionamiento es línea a línea.

En esta modalidad, la pantalla se divide conceptualmente en tres áreas (por encima de la línea continua que separa el área de estado de la pantalla):

- * El área de entrada de datos, situada en las líneas inferiores de la pantalla, sirve para introducir información, y puede modificarse tantas veces como se quiera.
- * El área de salida (toda la pantalla excepto las dos últimas líneas) contiene la información introducida por el programador y la producida por el ordenador, en orden de aparición. El contenido de este área no es modificable directamente.
- * El área de estado de la máquina virtual, situada a la derecha de la última línea (inmediatamente ENCIMA de la línea continua: a no confundir con el área de 'estado de la pantalla' explicada mas arriba), indica al programador cual es el estado de su máquina virtual. El contenido de este área no es modificable directamente; su significado se discute mas adelante.

Introduccion de datos en modalidad de línea a línea

Las instrucciones y los datos que se proporcionan al ordenador se introducen línea a línea (a diferencia de



otras terminales, en las que se introducen caracter a caracter). Esto significa que la línea que se está escribiendo puede modificarse o reescribirse sobre el área de entrada tantas veces como sea necesario. Las modificaciones pueden efectuarse desplazando el cursor mediante las cuatro teclas con flechas pequeñas que se encuentran a la derecha del teclado, y modificando directamente los caracteres erróneos. También pueden utilizarse las teclas de inserción y supresión de caracteres (situadas encima de las anteriores). Para hacer efectiva la transmisión de datos hacia el ordenador, debe pulsarse la tecla 'INTRO'. Una vez introducida, la línea aparece en el área de salida.

Borrado de la pantalla.

Cuando el número de líneas introducidas desde la terminal o producidas por programa supera el tamaño del área de salida, aparece 'MORE...' en el área de estado, indicando que es necesario borrar la pantalla para que pueda aparecer más información. La pantalla puede borrarse apretando la tecla 'BORRA' o la 'AP2'; si no se aprieta nada, la pantalla se borra automáticamente pasado un minuto.

Si desea que se mantenga la pantalla (para examinar con más detalle su contenido), apriete la tecla 'INTRO'. El área de estado cambiará a 'HOLDING' (aguantando), y el contenido de la pantalla no desaparecerá hasta que sea borrada o vuelva a apretarse la tecla 'INTRO' (que devuelve la pantalla al estado 'MORE...').

La aparición de 'NOT ACCEPTED' en el área de estado significa que el ordenador no puede atender su demanda. Esta señal se borra automáticamente pasados unos segundos - intente de nuevo.

Otros contenidos del área de estado ('CP READ', 'VM READ', 'RUNNING') se discuten más adelante.



COMO ES EL SISTEMA OPERATIVO ?

=====

El sistema operativo es el conjunto de facilidades de que dispone el ordenador para gestionar el acceso de los diversos usuarios a sus recursos. Es el responsable, por ejemplo, de los mecanismos de seguridad que impiden que un usuario observe o modifique la informacion perteneciente a otro usuario; tambien se encarga de tareas demasiado complicadas o delicadas, como el control a nivel fisico de las terminales.

Acceso al ordenador: LOGON

Cada usuario tiene asignados unos recursos (areas de disco, tiempo de CPU, etc.), que vienen identificados por un codigo de usuario (userid) y una palabra clave (password). Para acceder a estos recursos, siga el siguiente procedimiento:

Despues de poner en marcha la terminal, aparece el mensaje 'VM/370 ONLINE' y el logotipo de la instalacion en la pantalla (fig. 4).

Apriete la tecla 'INTRO'. El mensaje y el logotipo desaparecen y aparece 'CP READ' en el area de estado.

Introduzca 'LOGON', un espacio en blanco y su codigo de usuario. Apriete 'INTRO'.

La linea que introdujo se encuentra ahora en el area de salida. Si debajo aparece un mensaje distinto de 'ENTER PASSWORD:', su linea LOGON no ha sido aceptada -- verifique si es correcta; si no lo es, vuelva al paso anterior. Si aparece 'ENTER PASSWORD:', el ordenador le ha reconocido. Teclee su palabra reservada (password). El cursor se desplaza, pero los caracteres no aparecen en la pantalla, para evitar que otro usuario pueda verlos. Apriete la tecla 'INTRO'.

Espere unos segundos hasta que la pantalla se borre. Ya puede utilizar el ordenador.

Estructura del sistema operativo

El sistema operativo esta estructurado de tal modo que cada usuario accede a una serie de dispositivos y recursos como si dispusiera de un ordenador completo para el solo:

512 K de memoria central,



una impresora de lineas,
una lectora de fichas,
una perforadora de fichas,
un disco con mil bloques de 1024 bytes cada uno, que desaparece al terminar la sesion (disco del usuario),
un disco permanente a compartir por todos los usuarios del departamento.

Estos recursos son VIRTUALES - no existen separadamente, sino que son simulados por el sistema a partir de los recursos reales (veanse figs. 1 y 2).

El conjunto de recursos virtuales de que dispone un usuario se conoce como 'Maquina Virtual' (Virtual Machine, VM - el sistema operativo se llama VM/SF, Virtual Machine/System Product). Dado que a nivel logico estos recursos se comportan como un ordenador independiente, cada maquina virtual necesita su propio sistema operativo (el que se utiliza en esta instalacion se llama CMS, Conversational Monitor System: Sistema Monitor Conversacional).

Ademas es necesario un sistema operativo que coordine las actividades de las maquinas virtuales. Este sistema operativo se llama CP (Control Program: Programa de Control).

Comandos

El usuario se comunica con el sistema tecleando instrucciones llamadas comandos que introduce desde el area de entrada al pulsar la tecla 'INTRO'. Segun a que componente del sistema operativo vayan dirigidos, se habla de 'comandos de CP' o de 'comandos de CMS'. Cada comando especifica la ejecucion de una accion con efectos bien definidos; si se teclea el comando incorrectamente o se produce algun error al ejecutarlo, suelen imprimirse en pantalla, en el area de salida, mensajes de error explicando que ha sucedido. (Vease el Apendice A para una lista de los comandos mas frecuentes; para un estudio mas detallado, veanse las referencias [CMSCMR] y [CPCR]).

En esta guia se muestran los comandos tal como deben introducirse en la terminal, excepto en dos casos:

- si un comando o parte de un comando aparece formando parte de una linea del texto, se encierra entre



comillas ('); estas comillas NO forman parte del comando, y no deben teclearse al utilizarlo.

- Los signos de puntuacion que aparecen despues de algunos comandos forman parte del texto y no del comando; lo mismo sucede con algunos comentarios o simbolos que suelen aparecer bastante separados del comando y en minusculas.

Por ejemplo, el comando de CP

```
CP QUERY VIRT
```

proporciona informacion sobre los componentes de su maquina virtual. Si se equivoca y teclea 'QUERY BIRT', aparece el mensaje de error

```
DMKCQG045E BIRT NOT LOGGED ON
```

que significa "no hay ningun usuario activo llamado BIRT". Esto sucede porque el comando 'QUERY' tambien sirve para averiguar si un usuario determinado esta utilizando el ordenador.

Los comandos de CP empiezan con las letras 'CP', mientras que los de CMS se teclean sin ningun prefijo. De todas formas, si un comando de CP se introduce sin las letras 'CP' tambien es ejecutado, ya que CMS, si no recibe un comando que le es propio, lo envia a CP. En esta Guia se colocara en general el prefijo 'CP' para distinguir entre unos y otros comandos.

Estados del sistema

El area de estado contiene normalmente 'RUNNING' (funcionando) tanto si se esta ejecutando algun programa o comando como si el ordenador esta esperando la introduccion de algun comando. Cuando su programa necesita datos, el area de estado cambia a 'VM READ' (lectura de la maquina virtual), indicando que deben introducirse datos.

En determinados momentos en los que su maquina virtual no esta funcionando (por ejemplo, al hacer 'LOGON') aparece 'CP READ' (lectura del Programa de Control) en el area de estado. Esto indica que pueden introducirse EXCLUSIVAMENTE comandos de CP.

Puede forzarse una transicion del area de estado apretando 'INTRO' en cualquier momento. Si se estaba ejecutando un programa o comando de CMS, aparece 'VM READ' rapidamente: pueden introducirse mas datos o



comandos, que quedan esperando en la memoria del ordenador hasta el momento en que son solicitados por su maquina virtual. Si aparece 'CP READ', recuerde que solo puede introducir comandos de CP (puede encontrarse en este estado debido a un error grave de su maquina virtual); puede intentar volver a su maquina virtual (estado 'VM READ') mediante el comando

BEGIN

Si no lo consigue, teclee

IPL CMS
DTRIFF NOPAN



COMO PEDIRLE INFORMACION AL SISTEMA ?

=====

El sistema operativo proporciona dos metodos para facilitar la utilizacion del ordenador y la obtencion de informacion:

El comando 'IPF' muestra por pantalla listas de alternativas ('menus') entre las cuales el usuario debe escoger. Siguiendo las instrucciones del sistema puede realizarse cualquier tarea, como invocar al editor o compilar y ejecutar un programa.

El comando 'HELP' proporciona por pantalla explicaciones sobre los mensajes de error e informacion referente al formato y utilizacion de los comandos. La informacion proporcionada es la misma que se encuentra en los manuales. Para utilizarlo, introduzcase

'HELP nombre-de-comando', si se trata de un comando de CMS

'HELP CP nombre-de-comando', si se trata de un comando de CP (si duda, pruebelo de las dos formas)

'HELP numero-de-mensaje' si se trata de un mensaje ('numero-de-mensaje' se obtiene a partir del identificador del mensaje - DMKCQG045E en el ejemplo del comando 'QUERY' - suprimiendo los caracteres cuarto, quinto y sexto, como en 'HELP DMK045E').

'HELP CMS MENU' o 'HELP CP MENU' si se desea elegir entre la lista de los comandos de CMS (de CP).

Al utilizar el comando 'HELP', la pantalla cambia; la linea de comandos pasa a encontrarse en la parte superior de la pantalla, precedida por el simbolo '==>', y las teclas FP tienen asignadas funciones especiales, resumidas en las dos ultimas lineas de la pantalla, de acuerdo con la siguiente tabla:



| Tecla | Nombre | Funcion |
|-------|--------------|---|
| FP1 | HELP MENU | elige un miembro de un 'menu' vuelve al 'menu' correspondiente |
| FP2 | TOP | vuelve al principio del fichero |
| FP3 | QUIT | 'sale' de HELP |
| FP4 | PRINT | imprime la informacion visualizada por pantalla |
| FP5 | LOCATE | introduciendo al lado del simbolo '==>' la tira de caracteres que se desee encontrar en el fichero de informacion 'HELP' y pulsando repetidamente la tecla FP5 el cursor se posiciona sobre todas las apariciones de esa tira |
| FP6 | PREV.CMD | coloca al lado del simbolo '==>' el ultimo comando introducido |
| FP7 | UP | 'pasa pagina' hacia atras |
| FP8 | DOWN | 'pasa pagina' hacia adelante |
| FP9 | PFKEYS | proporciona informacion sobre el uso de las teclas FP bajo 'HELP' |
| FP10 | UP 1/2 | pasa media pagina hacia atras |
| FP11 | DOWN 1/2 | pasa media pagina hacia adelante |
| FP12 | CANCEL | 'sale' de HELP inmediatamente (sin volver al nivel de HELP anterior, como hace FP3) |



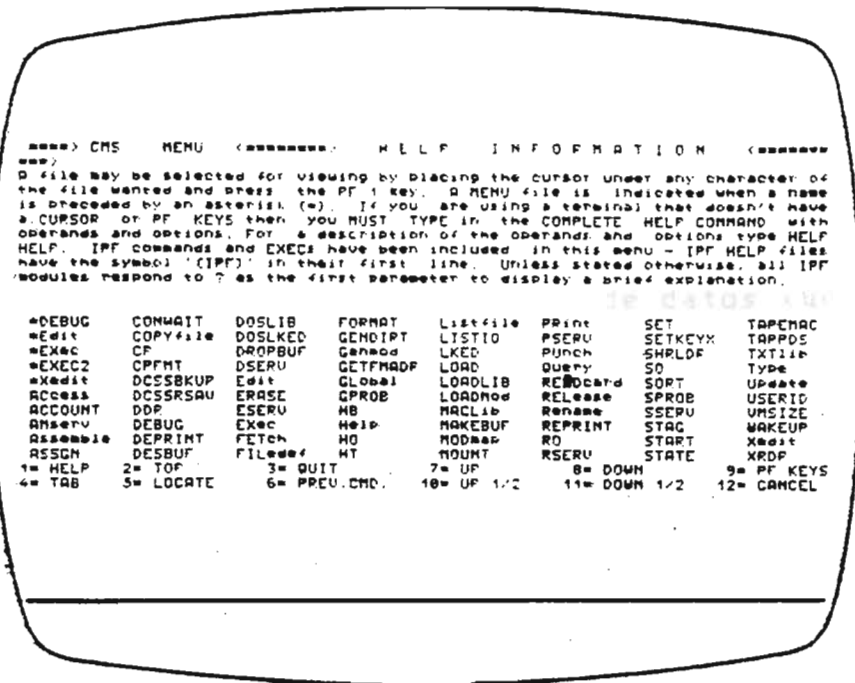


Fig. 7.1. Información que aparece al introducir HELP CMS MENU. Colocando el cursor encima de cualquiera de los comandos y apretando la tecla FP1 se obtendrá información sobre el mismo. En este caso, del comando READcard.

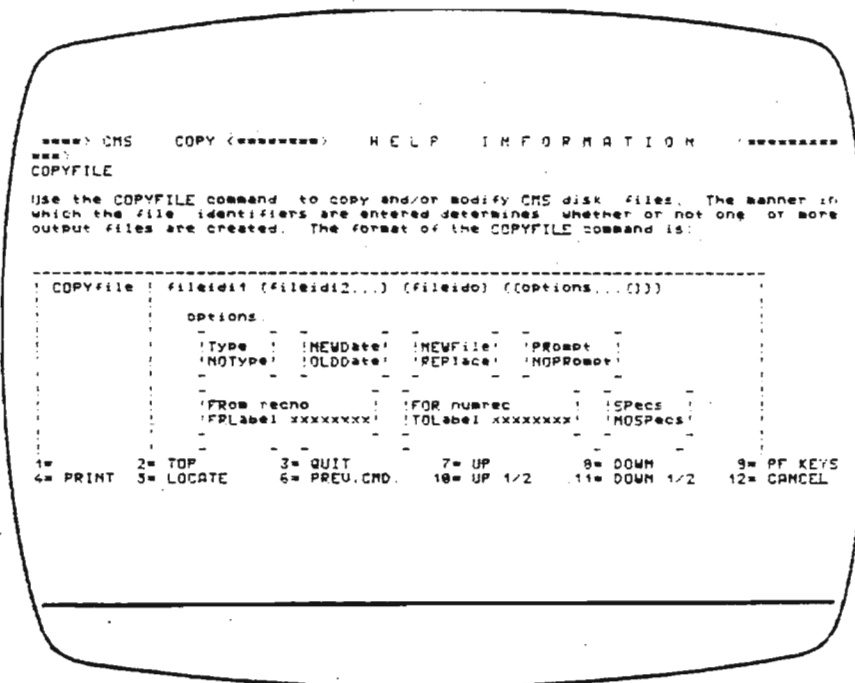


Fig. 7.2. Información obtenida mediante HELP COPY. Se puede visualizar la información que sigue mediante la tecla FP8.

COMO UTILIZAR FICHEROS ?

=====

Concepto de fichero

Un fichero es cualquier coleccion de datos (una matriz de numeros, o un texto medieval antiguo, un programa en lenguaje FORTRAN, un programa en 'lenguaje maquina', etc.). Estos datos suelen dividirse en lineas llamadas registros. Un fichero es de formato fijo cuando todos sus registros tienen la misma longitud, y de formato variable en caso contrario (Son frecuentes los ficheros de formato fijo con registros de 80 caracteres, conocidos a veces como ficheros en imagen de ficha).

Ubicacion de un fichero

La mayoría de los ficheros residen físicamente en áreas de disco magnético llamadas minidisos (vease mas abajo), aunque algunos de ellos (por ejemplo, los destinados a comunicar datos entre distintos ordenadores) pueden residir en otros medios, como cintas, fichas perforadas o papel impreso.

Concepto de minidisco

En el sistema operativo VM/SP, los ficheros se encuentran en unidades lógicas llamadas minidisos, que ocupan distintas zonas de un disco magnético real (vease fig. 8). En la práctica, un minidisco es un 'disco virtual' que el usuario puede utilizar como si dispusiera de un disco real (posiblemente de reducidas dimensiones). En el resto de esta guía, así como en la mayoría de los manuales, se habla de 'discos' directamente, sobreentendiéndose que en realidad se está hablando de minidisos. Cada disco se identifica mediante una letra llamada modo de acceso al disco (que coincide con el filemode 'fm' de los ficheros contenidos en ese disco [vease 'identificación de un fichero', mas adelante]). El disco temporal asignado por defecto a cada usuario está accedido con modo A, el disco del departamento lo está con modo D.

Tipos de disco

El acceso a la información contenida en un disco puede hacerse en dos modos: en modo de solo lectura (Read/only, R/O), la información puede leerse o copiarse, pero no modificarse; en modo de lectura/grabación (Read/Write, R/W), la información puede leerse, copiarse y



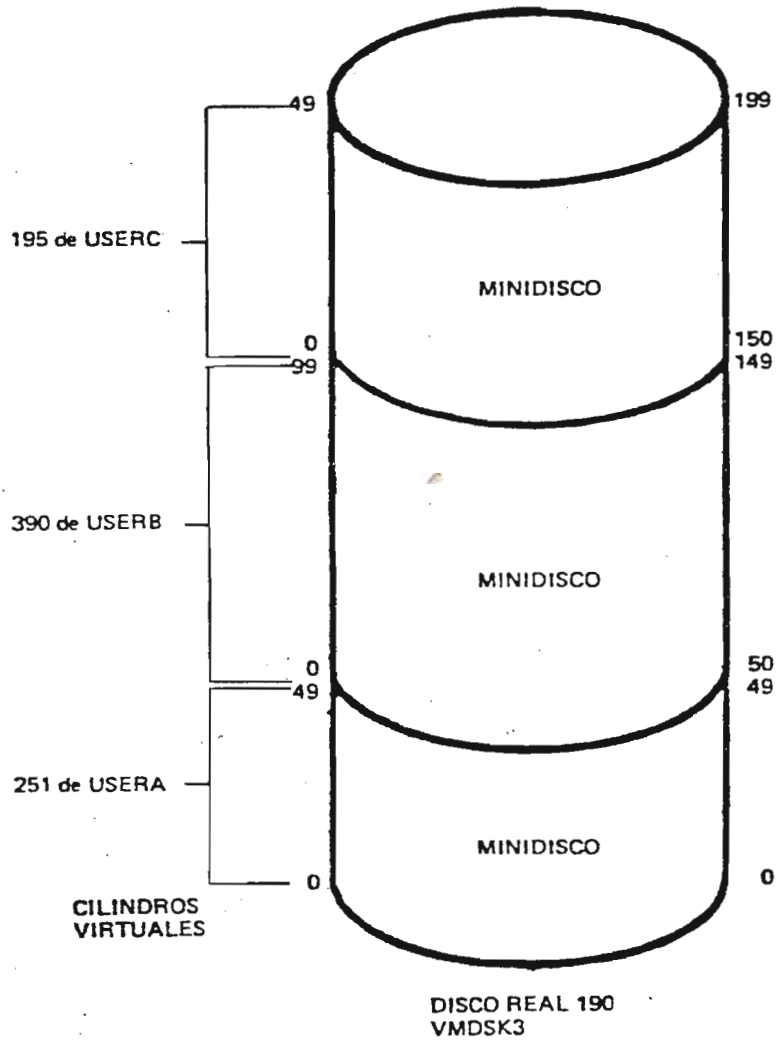


Fig. 8. Disco real subdividido en minidisos.



modificarse.

Algunos discos (en particular, los del sistema) están accedidos permanentemente como Read/Only; otros, como el disco del departamento de cada usuario, están accedidos normalmente como R/O, pero se acceden puntualmente como R/W para grabar información; por último, algunos discos, como el disco A del usuario, están accedidos permanentemente en R/W.

Algunos discos (como el disco del departamento o los discos del sistema) existen permanentemente, prescindiendo de quien los utilice; se conocen como discos fijos. Otros son discos temporales o de trabajo, y desaparecen al terminar la sesión (LOGOFF). El usuario puede definir sus propios discos temporales, pero no puede definir discos fijos.

El comando

QUERY DISK m

proporciona información sobre el estado del disco con modo 'm'; si 'm' es un asterisco ('*'), se proporciona información sobre todos los discos accedidos (vease fig. 9).

Identificación de un fichero en disco

Cada fichero en disco se conoce mediante un identificador (referido globalmente en los manuales como 'fileid') que debe ser único para cada fichero (esto es, no se permite la existencia de dos o más ficheros con el mismo identificador). Consta de tres componentes:

fn ft fm

(también referidos como: filename filetype filemode),
donde:

'fn' (filename: nombre del fichero) puede ser cualquier tira alfanumérica (compuesta de letras, números y/o los caracteres especiales @, R y N) de hasta ocho caracteres de longitud. Junto con 'ft' y 'fm', forma el identificador del fichero; se recomienda la elección de nombres mnemotécnicos.

'ft' (filetype: tipo del fichero) puede ser cualquier tira alfanumérica de hasta ocho caracteres, aunque suele indicar la utilización que se dará al fichero o que tipo de información contiene: así, un fichero que contenga un programa en lenguaje Pascal ha de tener



filetype PASCAL, y un fichero de datos puede tener filetype DATOS.

'fm' (filemode: modo del fichero) consta de una letra y un número entre 0 y 5. La letra identifica el disco sobre el que se encuentra el fichero y el número suele ser 1, 2 o 5 (los demás números son utilizados por el sistema). En la mayoría de los comandos, el uso del número es opcional; el sistema supone por defecto '1' al crear un fichero y el número existente al modificarlo. Además, el número no sirve para distinguir dos identificadores: 'PROB DATA A1' y 'PROB DATA A5' no pueden existir al mismo tiempo (vease [CMSUG], pag. 64 para una explicación completa sobre el sentido de esta numeración).

Orden de búsqueda estándar de CMS

Algunos comandos permiten utilizar un asterisco ('*') en vez de un filemode al identificar un fichero: 'fn ft *'. Esto significa "el primer fichero con filename 'fn' y filemode 'ft' que se encuentre al buscar en todos los discos accedidos por orden alfabético del modo de acceso". El comando 'STATE', seguido de un identificador de fichero, produce mensajes de error si y solo si no existe el fichero nombrado: así, el comando

```
STATE PROB DATA A
```

busca el fichero 'PROB DATA A' sobre el disco 'A', pero

```
STATE PROB DATA *
```

lo busca primero en el disco 'A' temporal del usuario, después en el disco 'D', después en el disco 'H' (es un disco del sistema), etc.

Para saber que discos tiene accedidos y cual es el orden de búsqueda establecido entre ellos, utilice el comando

```
QUERY SEARCH
```

Directorios - FLIST.

Cada disco contiene un directorio con los identificadores de todos los ficheros que contiene. El comando FLIST sirve para visualizar parcial o totalmente uno o más directorios.

El comando FLIST muestra, bajo una cabecera, una lista



de ficheros (vease fig. 10). Para cada fichero se muestra a la izquierda de la pantalla su identificador, y a la derecha su formato (fijo [F] o variable [V]), la longitud maxima de sus lineas (desde 1 hasta 65535), el numero de lineas que contiene, el numero de bloques de 1024 bytes que ocupa en disco, y la fecha y la hora en que se realizo la ultima modificacion (vease fig. 10).

El formato (simplificado) del comando 'FLIST' es

```
FLIST [ fn [ ft [ fm ] ] ]
```

donde las partes encerradas entre corchetes son opcionales. 'fn', 'ft' y 'fm' pueden contener asteriscos, en cuyo caso se listan todos los ficheros cuyo identificador coincida con el dado al reemplazar cero o mas caracteres por asteriscos. Si no se especifica 'fm', se supone 'A'; para los demas argumentos, se supone '*' (todos los filenames, o todos los filetypes). Asi,

'FLIST' equivale a 'FLIST * * A', y lista todos los ficheros del disco A;

'FLIST * * m' lista todos los ficheros del disco 'm';

'FLIST fn' equivale a 'FLIST fn * A', y lista todos los ficheros con filename 'fn' que se encuentren en el disco del usuario;

'FLIST fn * *' lista los ficheros de filename 'fn' (de todos los discos);

'FLIST * ft' equivale a 'FLIST * ft A', y lista los ficheros de filetype 'ft' del disco A (p. ej., 'FLIST * FORTRAN'); y

'FLIST JCF* PLIOPT *' lista todos los ficheros de filetype 'PLIOPT' cuyo filename empiece por 'JCF' en todos los discos accedidos por el usuario.

Las teclas de funcion de programa (FP) sirven para efectuar automaticamente determinadas funciones:



```

R disk
LABEL CUU M STAT CYL TYPE BLKSIZE FILES BLKS USED-(2) BLKS LEFT BLK TOTAL
TEMP-A 191 A R/W FB 3378 1824 5 122-12 878 1888
CMS1 192 D R/O FB 3378 1824 32 377-33 763 1148
IPFPTF 328 M/H R/O FB 3378 1824 168 1348-39 932 2288
FEATRS 319 P/P R/O FB 3378 1824 395 12139-57 9161 21388
MHT198 198 S R/O FB 3378 1824 186 16682-78 4822 21584
B 323 M/W R/O FB 3378 1824 238 1924-42 2636 4568
USERAP 318 X/X R/O FB 3378 1824 67 1861-93 79 1148
IPF 19E Y/S R/O FB 3378 1824 779 4638-64 2362 7288
R: T=0.81/0.83 19:18:45
    
```

RUNNING

Fig. 9. Características de los discos utilizados por una máquina virtual obtenidas mediante el comando CP QUERY DISK.

```

LUL A --- A 191 1888 BLKS 3378 R/W 15 FILES 214 -- FILE 1 OF 15
PRUEBA BATCH A1 F 88 2 1 4/18/83 17:36
PRUEBA CONS A1 F 88 1 1 4/18/83 16:48
PRUEBA DATOS A1 F 88 488 39 4/18/83 16:49
CONSOLA PRUEBA A1 F 133 22 5 4/18/83 18:09
FILE LISTING A1 F 133 145 19 4/18/83 18:09
CONSOLA SESION A1 F 133 389 51 4/18/83 19:15
A A A1 F 133 71 18 4/18/83 19:39
ALTURAS DATOS A1 F 88 474 38 4/18/83 19:39
ALTURAS MASTER A1 COPY / = = b _ F 512 48 20 4/18/83 19:39
B B A1 F 133 8 2 4/18/83 19:39
BMDP BATCH A1 F 88 2 1 4/18/83 19:39
BMDP DATA A1 F 88 4 1 4/18/83 19:39
BMDP NULA A1 F 88 2 1 4/18/83 19:39
BMDP1 BATCH A1 F 88 2 1 4/18/83 19:39
C C A1 F 133 58 7 4/18/83 19:39
    
```

1=HLF 2=BR# 3=END 4=MEQ 5=SPL 6=/SB 7=SCB 8=SCF 9=/SD 10=/ST 11=>I 12=CM

Fig. 10. Directorio del disco A proporcionado por FLIST. El comando COPY introducido copiará el fichero ALTURAS MASTER A en el disco B.



| Tecla | Funcion |
|-------|---|
| FP1 | HELP FLIST describe el comando FLIST |
| FP2 | BROWSE permite examinar un fichero |
| FP3 | QUIT 'sale' de FLIST |
| FP4 | XEDIT edita un fichero |
| FP5 | permite alterar el aspecto de la pantalla al utilizar FLIST's multiples (consultese HELP FLIST) |
| FP6 | /SB (sort by block) ordena los ficheros por su tamaño |
| FP7 | 'pasa pagina' hacia atras |
| FP8 | 'pasa pagina' hacia adelante |
| FP9 | /SD (sort by date) ordena los ficheros por su fecha |
| FP10 | /ST (sort by type) ordena los ficheros segun su filetype |
| FP11 | >I (increase) aumenta el tamaño del area de entrada |
| FP12 | CANCEL 'sale' de FLIST inmediatamente (PF3 ejecuta primero los comandos) |

Hagase 'HELP FLIST' para una discusion mas detallada del comando FLIST.

Visualizacion de ficheros - BROWSE

El comando BROWSE sirve para visualizar un fichero SIN MODIFICARLO. Su formato es



BROWSE identificador-de-fichero

y coloca al usuario en un entorno similar al editor (vease más adelante), en el que puede utilizar diversos subcomandos y teclas FP para controlar la parte del fichero que esta visualizando. El fichero se divide conceptualmente en paginas de 22 lineas; solo se visualizan 80 columnas (normalmente las primeras) de cada linea. Puede controlarse que parte del fichero se visualiza y el formato de la pantalla mediante subcomandos introducidos desde la segunda linea de la pantalla, al lado del simbolo '==>'. Los subcomandos mas importantes son:

'F n', donde 'n' es un numero natural, pasa 'n' paginas hacia adelante. La tecla FP8 tiene el efecto de 'F 1'.

'B n' pasa 'n' paginas hacia atras. La tecla FP7 tiene el efecto de 'B 1'.

'N n' y 'D n' avanzan 'n' lineas.

'U n' retrocede 'n' lineas.

En todos estos comandos, si no se especifica 'n' se supone 1.

'R' permite visualizar las columnas situadas a la derecha de las que se estan visualizando.

'L' permite visualizar las columnas situadas a la izquierda de las que se estan visualizando.

'V n m' indica que se desea visualizar las columnas 'n' a 'm'; si se omite 'm', se supone (n + 80).

'Q' termina la ejecucion del comando BROWSE. La tecla FP3 tiene el mismo efecto.

La tecla FP1 tecleada desde BROWSE proporciona informacion sobre el formato y utilizacion del comando BROWSE; tiene el mismo efecto que el comando 'HELP BROWSE'.

Para buscar una tira de caracteres basta con introducirla en la linea de comandos encerrada entre barras:

/Primera/

busca la tira de caracteres 'primera'. La segunda barra puede omitirse.



Comandos basicos para el manejo de ficheros

Los comandos se muestran aqui en su formato mas sencillo; muchos de ellos pueden utilizarse bajo formas mas potentes o abreviarse. Para conocer el formato completo, utilicese el comando HELP o consultese la publicacion [CMSCMR].

ERASE fn ft fm

borra el fichero (identificado por) 'fn ft fm'.

RENAME fn1 ft1 fm1 fn2 ft2 fm2

cambia el identificador del fichero 'fn1 ft1 fm1' por 'fn2 ft2 fm2'. Si no se desea modificar alguno de los componentes del identificador, puede simplificarse el comando colocando un signo de igualdad ('=') en el lugar correspondiente del segundo identificador. Por ejemplo,

RENAME fn1 ft1 fm1 = ft2 =

cambia solo el filetype del fichero, dejando igual el filename y el filemode.

COPYFILE fn1 ft1 fm1 fn2 ft2 fm2

crea una copia del fichero 'fn1 ft1 fm1' con identificador 'fn2 ft2 fm2' (puede aplicarse la misma simplificacion comentada para el caso del comando 'RENAME').

Para gestionar la modificacion de la informacion contenida en el disco del departamento (disco 'D'), se ponen a disposicion de los usuarios los siguientes comandos, desarrollados en el Centro de Calculo:

ALD fn ft fm

graba en el disco del departamento el fichero 'fn ft fm' si todavia no existia.

ALDR fn ft fm



graba en el disco del departamento el fichero 'fn ft fm', reemplazando cualquier copia que existiera en ese disco.

```
NOUNOMD fn1 ft1 fm1 fn2 ft2 fm2
```

cambia el identificador del fichero 'fn1 ft1 fm1' a 'fn2 ft2 fm2'. El fichero debe encontrarse en el disco del departamento (disco D) por lo que 'fm1' y 'fm2' deberan ser de la forma 'Dn'.

Utilizacion de la barra ('/') en FLIST.

Al utilizar el comando FLIST, el cursor se posiciona al lado del primer fichero en pantalla: pueden teclearse comandos como desde el area de entrada en funcionamiento linea a linea. Cualquier comando que se introduzca se aplica al fichero junto al cual se escribe, ya que FLIST concatena el identificador del fichero al comando; asi, para borrar un fichero basta con escribir 'ERASE' junto al identificador. Notese que FLIST añade el identificador del fichero AL FINAL del comando tecleado por el usuario; si se necesita en otra posicion, puede utilizarse el simbolo '/' (vease fig. 10):

Se desea cambiar el nombre de un fichero llamado 'AITKEN FORTRAN A' por 'RUNGE FORTRAN A'. Si no se utilizase FLIST, esto podria hacerse mediante

```
RENAME AITKEN FORTRAN A RUNGE = =
```

La concatenacion estandar que ofrece el comando 'FLIST' no es util en este caso, ya que el identificador 'AITKEN FORTRAN A' no aparece al final del comando; sin embargo, puede utilizarse el simbolo '/' para representar el identificador, por lo que bastara con teclear

```
RENAME / RUNGE = =
```

Del mismo modo, '/N' representa el filename del fichero, '/T' su filetype, y '/M' su filemode. Si NO se desea que FLIST añada el identificador del fichero al final del comando, puede introducirse precedido por '/O'. Por ultimo, cualquier combinacion de O, N, T, M es valida despues del simbolo '/' para formar comandos complejos.

Asi, para compilar un programa en PASCAL llamado 'A PASCAL D' basta con situarse al lado de su identificador y teclear



/OTN

que se interpreta como sigue: 'O': no añadir nada a lo que introduzca el usuario; T: el tipo del fichero - en este caso, 'PASCAL'; N: el nombre del fichero, 'A': se obtiene 'PASCAL A', que es el comando necesario para compilar el programa. Igualmente, para compilar un programa FORTRAN puede escribirse

FORTGI /N

al lado del identificador del fichero que contiene el programa.



MOVIMIENTO, IMPRESION E INTERCAMBIO DE FICHEROS. EL SPOOL.
=====

En este apartado se describen los mecanismos utilizables para copiar, imprimir e intercambiar ficheros. Tambien se explica como puede obtenerse un registro impreso de las actividades realizadas por pantalla. Los comandos a utilizar dependen del soporte fisico utilizado y de la cantidad de usuarios involucrados en la transmision (vease fig. 11).

Disco-disco: copia de ficheros entre areas de disco del usuario.

Utilicese el comando 'COPYFILE':

COPYFILE fn1 ft1 fm1 fn2 ft2 fm2

crea una copia del fichero en disco con identificador 'fn1 ft1 fm1' sobre un fichero en disco con identificador 'fn2 ft2 fm2'. Este comando no puede utilizarse para copiar ficheros SOBRE el disco del departamento (esto es, 'fm2' no puede ser 'D'). Para copiar un fichero sobre el disco 'D', utilicense los comandos 'ALD' ('ALDR') y 'NOUNOMD'.

Disco-cinta: utilizacion de cintas.

Como el numero de unidades lectoras-grabadoras de cintas suele ser pequeño en comparacion con el numero de usuarios, estos dispositivos deben asignarse por el operador en el momento en que esten disponibles a cada usuario que los solicite. Para ello se envia un mensaje al operador indicandole que cinta se desea y en que unidad (las unicas opciones son 181, 182, 183 y 184) se desea:

MSG OP POR FAVOR, NECESITO LA CINTA
'identificacion' EN 181.

(Veanse las normas de utilizacion [NORM] para conocer en detalle el protocolo establecido para la entrega, identificacion, recogida, etc, de las cintas).

Una vez el operador coloca la cinta y asigna la unidad pedida, aparece en la pantalla el mensaje

TAPE 18n ATTACHED

y la unidad de cinta queda incorporada al conjunto de dispositivos de la maquina virtual del usuario (haga



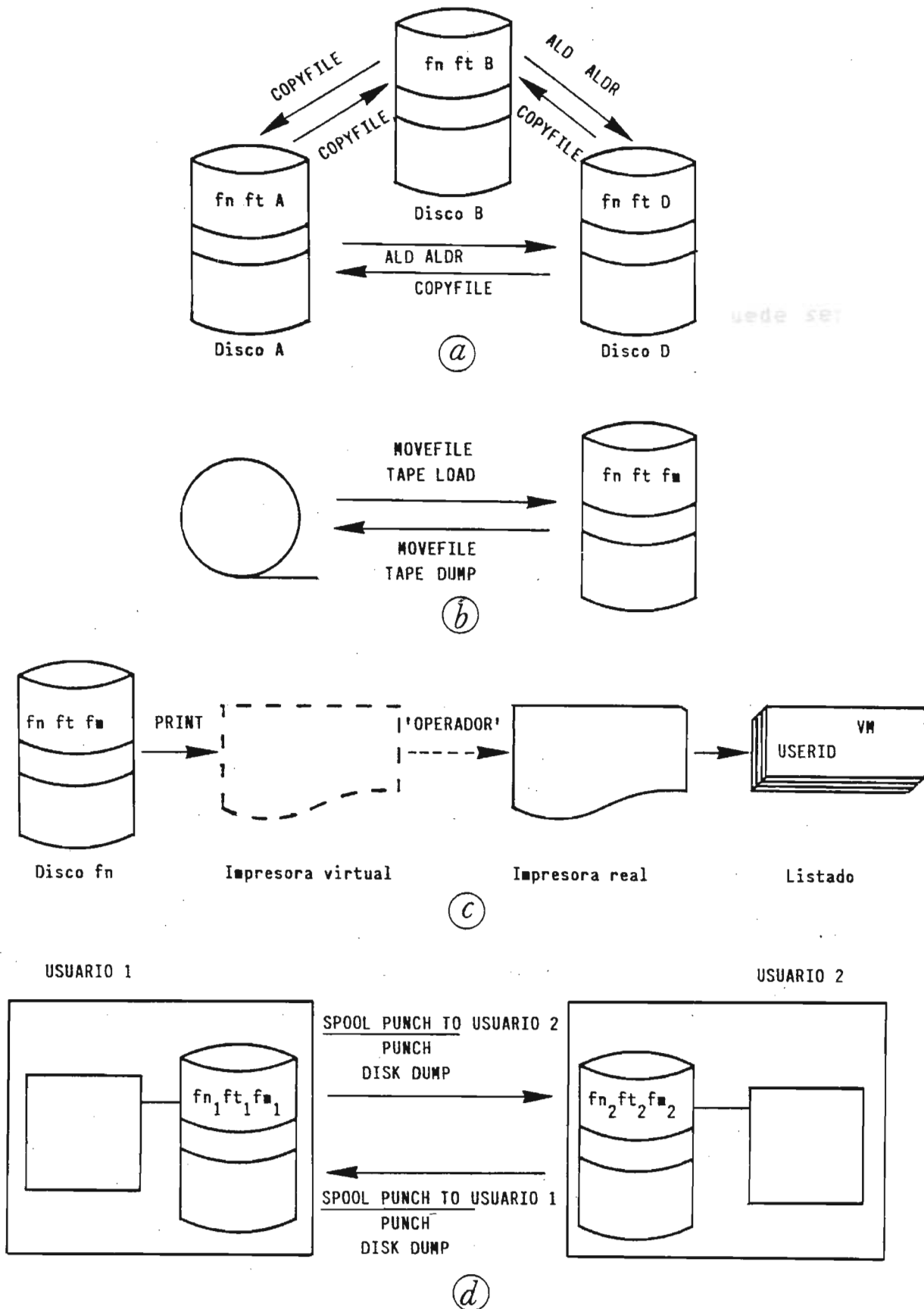


fig. 11. Movimiento, impresión e intercambio de ficheros.
a) Disco-Disco (el disco B sería un temporal definido por el usuario)
b) Cinta-Disco
c) Impresión (Disco-impresora)
d) Usuario-Usuario (Máquina virtual-máquina virtual)



'QUERY VIRT' para comprobarlo).

Mediante el comando 'TAPE' puede controlarse la unidad lectora-grabadora:

TAPE REW (unidad
rebobina la cinta colocada en 'unidad', que puede ser

TAP1 o 181
TAP2 o 182
TAP3 o 183
TAP4 o 184

(si no se coloca se supone 'TAP1' por defecto).

TAPE WTM n (unidad

Escribe 'n' marcas de cinta (Tape Marks) en la unidad 'unidad'.

Los comandos

TAPE FSF n (unidad
TAPE BSF n (unidad

adelantan y retroceden, respectivamente, 'n' marcas de final de fichero sobre la unidad 'unidad'.

Las cintas magneticas pueden utilizarse (vease fig. 12):

- * Para almacenar o utilizar informacion grabada en formato estandar (compatible con otras instalaciones), en cuyo caso puede accederse a ellas mediante el comando 'FILEDEF', explicado mas adelante, en el apartado 'como compilar y ejecutar un programa', y el comando 'MOVEFILE'.

Por ejemplo, los comandos

```
FILEDEF INMOVE TAP1 (RECFM FB LRECL 80 BLOCK
4000
FILEDEF OUTMOVE DISK DATOS CINTA A (RECFM F
LRECL
MOVEFILE
```

copian un fichero de formato fijo bloqueado con registros de 80 bytes y bloques de 4000 de una cinta colocada en la unidad 181 al disco 'A' con identificador 'DATOS CINTA A'.

En cambio, los comandos



```
FILEDEF INMOVE DISK PRUEBA FORTRAN A(RECFM F
LRECL 80
FILEDEF OUTMOVE TAP2 (RECFM F LRECL 80
MOVEFILE
```

copian el fichero en disco 'PRUEBA FORTRAN A' en una cinta situada en la unidad 182.

- * Para almacenar y/o recuperar ficheros en formato CMS (no utilizable en instalaciones que no posean el sistema operativo CMS), en cuyo caso puede utilizarse el comando 'TAPE' de CMS:

```
TAPE DUMP fn ft fm
donde 'fn', 'ft' o 'fm' pueden ser asteriscos (como
en el comando FLIST, vease mas arriba), escribe en
cinta los ficheros 'fn ft fm';
```

```
TAPE LOAD
los vuelve a grabar en el disco 'A', con el mismo
filename y filetype con que fueron grabados en
cinta. Si se desea grabarlos en un disco distinto
del 'A', por ejemplo el disco 'fm', hagase
```

```
TAPE LOAD * * fm
```

Si se desea leer solo uno de los ficheros que se grabaron en cinta, utilicese

```
TAPE LOAD fn ft fm
```

(Si 'fn' o 'ft' son asteriscos, se leeran solo los ficheros correspondientes - vease el comando FLIST).

```
TAPE SKIP [fn ft]
salta ficheros hasta posicionarse despues del
fichero 'fn ft'.
```

```
TAPE SCAN [fn ft]
salta ficheros hasta posicionarse antes del fichero
'fn ft'.
```

Para mas informacion, hagase 'HELP' de los comandos mencionados.

Disco-impresora: impresion de ficheros.

Cuando se desee imprimir un fichero, utilicese el comando

```
PRINT fn ft fm
```



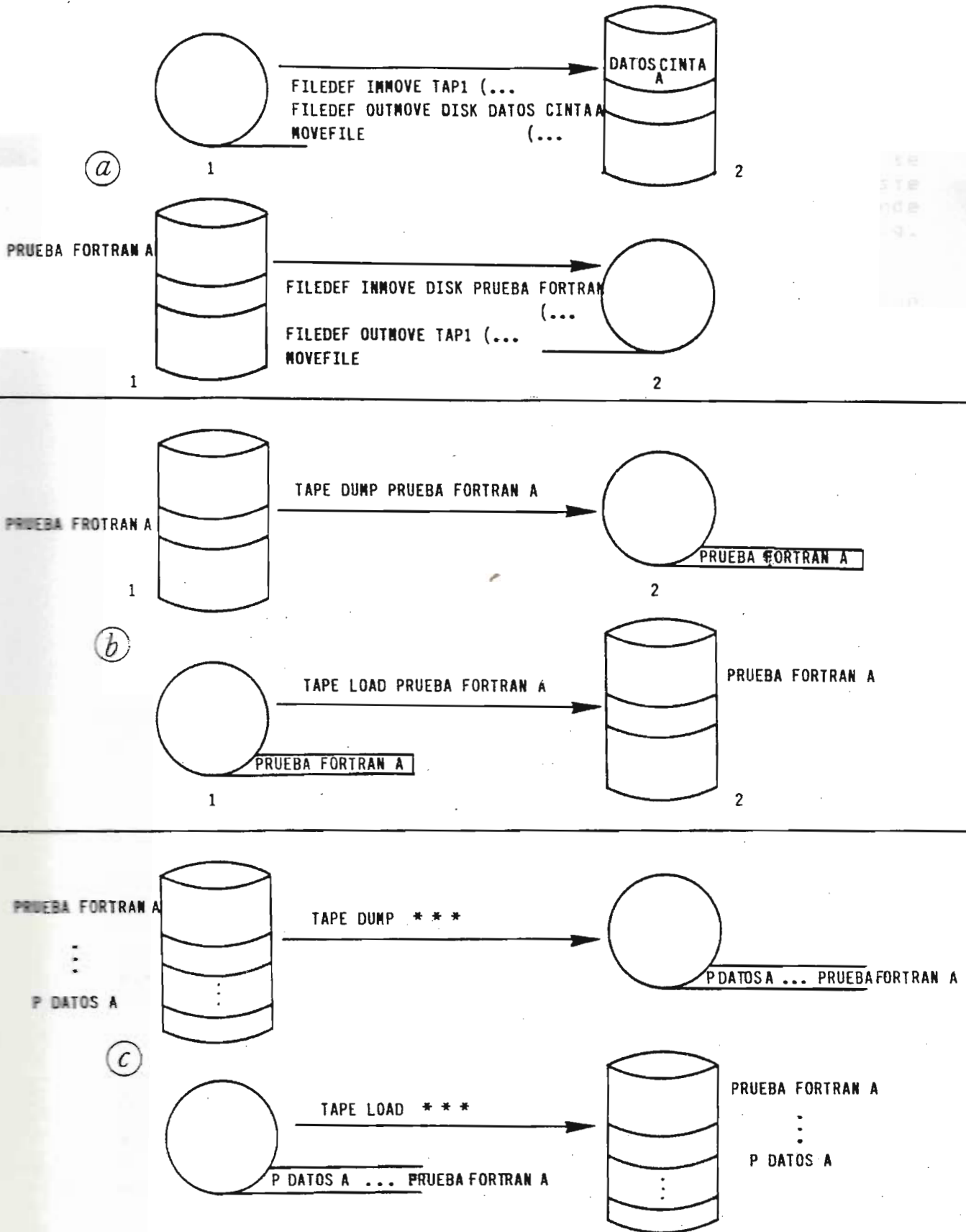


fig. 12. Copia de ficheros de cinta a disco y viceversa.

- a) Formato estandar
- b) Formato CMS-un sólo fichero
- c) Formato CMS-varios ficheros



donde 'fn ft fm' es el identificador del fichero que se pretende imprimir (si se omite 'fm', se supone 'A'). Este comando coloca su fichero en su impresora virtual, donde espera para ser impreso en la impresora real (vease fig. 11).

Si el filetype del fichero es 'LISTING', se supone que el primer caracter de cada registro es un 'caracter de control', y se imprime el fichero sin formateo alguno (estos caracteres, que regulan el espaciado entre líneas y el salto de página, son generados automáticamente por los compiladores y por muchos componentes del sistema); si el fichero no contiene caracteres de control, debe imprimirse con la opción 'NOCC' del comando 'PRINT':

```
PRINT fn LISTING fm (NOCC
```

Al contrario, si el filetype del fichero no es 'LISTING', el sistema supone por defecto que no hay caracteres de control, y formatea el fichero, separando las líneas en páginas numeradas con una cabecera que contiene el identificador del fichero. Si desea imprimir un fichero que, sin tener filetype LISTING, si contiene caracteres de control, utilice la opción 'CC' del comando 'PRINT':

```
PRINT fn ft fm (CC
```

Normalmente, cada ejecución del comando 'PRINT' provoca la impresión de un fichero separado. Los ficheros se separan mediante dos hojas de cabecera que contienen el código del usuario y una hoja de finalización con la palabra 'END'. Si se desean imprimir varios ficheros sin que aparezcan los separadores intercalados, puede utilizarse el comando

```
CP SPOOL PRINTER CONT
```

que coloca su impresora virtual en 'modo continuo'. Todos los ficheros que se impriman a partir de este momento se acumulan en un solo fichero de impresión hasta el momento en que se reestablezca el modo normal de operación mediante la secuencia de comandos

```
CP SPOOL PRINTER NOCONT  
CP CLOSE PRINTER
```

Los ficheros que se envían a la impresora virtual (mediante el comando 'PRINT') pueden ser impresos por el operador en cualquier momento. Si desea mantener un fichero bajo su control momentáneamente (por ejemplo,



para cambiar algunas de sus características), utilice el comando

CP SPOOL PRINTER HOLD

A partir de la ejecución de este comando, todos los ficheros que se envían a imprimir mediante el comando 'PRINT' tienen 'estado HOLD', y están bajo su control.

El comando

CP SPOOL PRINTER NOHOLD

devuelve su impresora virtual a su estado normal; sin embargo, todos los ficheros que se enviaron en estado 'HOLD' siguen estando bajo su control. Utilice el comando 'CHANGE' (explicado a continuación) para devolver sus ficheros de impresión al estado normal.

NOTA: Las Normas de Explotación del Centro de Cálculo pueden establecer convenciones relativas al estado ('HOLD' o 'NOHOLD') de los listados con características especiales.

Cada fichero de impresión tiene asignado un número entre 1 y 9999 que le identifica ('spoolid'). Este número puede utilizarse para cambiar las características del fichero, suprimirlo de las colas de impresión, etc. Así, el comando

CP CHANGE PRINTER nnnn [nnnn ...] NOHOLD

devuelve al estado normal ('NOHOLD') los ficheros con números de identificación ('spoolids') nnnn [nnnn ...]; y el comando

CP PURGE PRINTER nnnn [nnnn ...]

suprime los ficheros con números de identificación nnnn [nnnn ...] de las colas de impresión (los ficheros se pierden, y no pueden volver a recuperarse). En estos comandos, puede substituirse 'nnnn' por 'ALL', significando 'todos los ficheros de la cola de impresión'.

Puede visualizarse la cola de impresión mediante el comando

CP QUERY PRINTER ALL

que proporciona información sobre los ficheros que se enviaron, sus números de identificación ('spoolid'), su estado ('HOLD' o 'NOHOLD'), etc. (Véase fig. 14.f)



Usuario-usuario: transmision de ficheros entre usuarios.

Envio de ficheros a un usuario

Especifique a que usuario desea enviar ficheros mediante el comando

CP SPOOL PUNCH TO userid

donde 'userid' es el codigo del usuario al que se desea enviar ficheros - no es necesario que el usuario este conectado. A continuacion, puede enviar sus ficheros mediante uno de los dos metodos siguientes (vease fig. 13. Se describen comandos para enviar uno o mas ficheros - todos los que envie seran recibidos por el mismo usuario hasta que se especifique un nuevo 'userid' mediante el comando 'CP SPOOL PUNCH'):

* El comando 'PUNCH' envia ficheros cuya longitud maxima de registro no exceda 80 caracteres. Su formato es

PUNCH fn ft fm [(NOHEADER)]

El comando 'PUNCH' envia el fichero precedido de una cabecera o linea de control que simplifica su recepcion mediante el comando 'READCARD' (explicado mas adelante, en el apartado 'Recepcion de ficheros'); si no desea que se envie esta cabecera, utilice la opcion 'NOHEADER'.

Al igual que en el caso del comando 'PRINT', puede interesar que varios ficheros sean recibidos juntos. Para ello, utilice el comando

CP SPOOL PUNCH CONT

que establece que el envio de ficheros se realizara en 'modo continuo'. Los ficheros se reciben juntos, pero siguen teniendo entidad propia. Si desea enviar varios ficheros de modo que se reciban como un solo fichero, utilice ademas la opcion 'NOHEADER' del comando 'PUNCH' para cada fichero que envie.

Para devolver la comunicacion a su estado normal (discontinuo: cada fichero enviado se recibe como un fichero independiente) utilice

CP SPOOL PUNCH NOCONT CLOSE



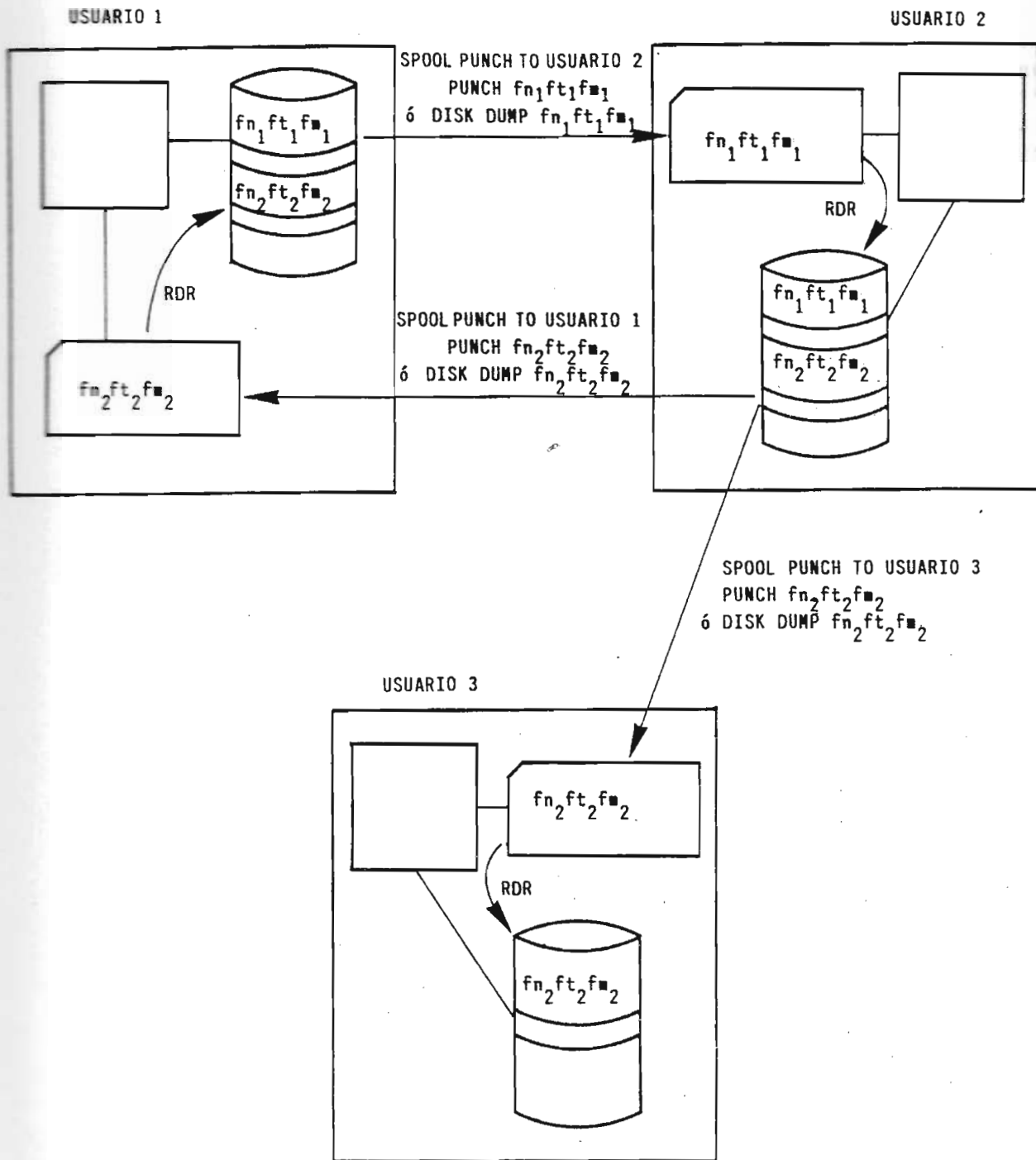


fig. 13. Intercambio de ficheros entre usuarios



* El comando 'DISK DUMP' envia ficheros de cualquier formato y tamaño de registro. Su formato es

DISK DUMP fn ft fm

donde 'fm ft fm' es el identificador del fichero que se esta enviando. Como en el comando 'FLIST' (y otros), 'fn', 'ft' o 'fm' pueden contener asteriscos, en cuyo caso se envian todos los ficheros cuyo identificador pueda transformarse en el dado cambiando caracteres por asteriscos.

Segun la filosofia del sistema VM, el envio de ficheros a otro usuario se realiza mediante la 'perforadora virtual', que 'perfora virtualmente' el fichero enviado y lo coloca en la 'lectora virtual' de la 'maquina virtual' del usuario receptor.

Recepcion de ficheros: la lectora virtual.

Los ficheros recibidos quedan en cola situados en la denominada 'lectora virtual'. Esta cola puede observarse (vease fig. 14) mediante el comando

CP QUERY READER ALL

* Para leer un fichero que haya sido enviado mediante el comando 'PUNCH', utilice el comando

READCARD fn ft fm

que lee el primer fichero de la cola y lo graba en disco con identificador 'fn ft fm'. Si se omite 'fm', se supone 'A'. Si 'fn ft' es '*' (o '* *'), se asigna al fichero el mismo identificador que tenia cuando fue enviado (excepto si se utilizo la opcion 'NOHEADER' del comando PUNCH, en cuyo caso 'fn' y 'ft' no deben ser asteriscos).

* Para leer uno o mas ficheros que hayan sido enviados mediante el comando 'DISK DUMP', utilice el comando 'DISK LOAD':

DISK LOAD

que lee los ficheros que han sido enviados y los graba en el disco 'A' del usuario con los mismos filename y filemode que tenian cuando fueron enviados.

Si no desea ocuparse del modo en que fueron enviados sus ficheros, utilice el comando



```

R: T=0.03/0.13 14:20:31
q print all
ORIGINID FILE CLASS RECORDS CPY HOLD DATE TIME NAME TYPE DIST
ZDOCIM1 6335 A PRT 00000070 002 USER 04/20 14:19:10 A LISTING ZDOCIM1
ZDOCIM1 6336 A PRT 00000082 002 USER 04/20 14:19:21 ALTURAS DATOS ZDOCIM1
ZDOCIM1 6338 A PRT 00000049 002 USER 04/20 14:19:29 C LISTING ZDOCIM1
ZDOCIM1 6360 A PRT 00000196 002 USER 04/20 14:19:47 EXEM DADES ZDOCIM1
ZDOCIM1 6364 A PRT 00000009 001 USER 04/20 14:20:30 COM EXEC ZDOCIM1
ZDOCIM1 6363 A PRT 00000023 001 USER 04/20 14:20:48 EXM FORTRAN ZDOCIM1
P: T=0.01/0.02 14:20:33

```

Fig. 14.1. Cola de ficheros enviados a la impresora virtual antes de ser imprimidos por la impresora real. Obsérvese que de los cuatro primeros se efectuarán dos copias.

```

q reader all
ORIGINID FILE CLASS RECORDS CPY HOLD DATE TIME NAME TYPE DIST
BATCH3 2926 A PRT 00000143 001 NONE 04/18 19:24:00 CMSBATCH ZDOCIM1 ZDOCIM1
BATCH3 2923 T COM 00000023 001 NONE 04/18 19:23:42 BATCH CONSOLE ZDOCIM1
BATCH3 2928 A PRT 00000143 001 NONE 04/18 19:24:40 CMSBATCH ZDOCIM1 ZDOCIM1
BATCH3 2927 T COM 00000022 001 NONE 04/18 19:24:21 BATCH CONSOLE ZDOCIM1
ZDOCIM2 2943 A PUN 00000473 001 NONE 04/18 19:29:44 ALTURAS DATOS ZDOCIM2
ZDOCIM2 2946 A PUN 00000011 001 NONE 04/18 19:30:00 CONTAAS PASCAL ZDOCIM2
ZDOCIM2 2947 A PUN 00000023 001 NONE 04/18 19:30:10 EXM FORTRAN ZDOCIM2
ZDOCIM2 2930 A PUN 00000136 001 NONE 04/18 19:30:27 SUMA PLIOPT ZDOCIM2
P: T=0.01/0.02 19:31:07

```

RUNNING

Fig. 14.2. Cola de ficheros situados en la lectora virtual de un usuario .



RDR

que es automatico y no requiere saber nada sobre como fue enviado el fichero. El comando pregunta interactivamente si se desea almacenar cada uno de los ficheros que se encuentran en su lectora; puede decidirse entre grabar un fichero, cambiarle el nombre, dejarlo en la lectora o borrarlo.

Al igual que en la cola de impresion, cada fichero viene identificado por un numero ('spoolid'). Si desea suprimir un fichero de su cola de lectura (por ejemplo, porque se lo han enviado pero no le interesa) utilice el comando

```
CP PURGE READER nnnn [nnnn ...]
```

Los ficheros de la cola de lectura se leen (obtienen) en el orden en el que se observan mediante el comando 'QUERY READER ALL'. Si le interesa leerlos en otro orden, puede reordenar la cola mediante el comando

```
CP ORDER READER nnnn [nnnn ...]
```

que coloca a la cabeza de la cola los ficheros identificados por nnnn [nnnn ...].

Registro de las actividades de consola: la 'consola virtual'.

'Consola Virtual' es otro de los nombres dados a su terminal (se refiere a la idea de que su terminal actua como 'consola de operacion' de su maquina virtual). Si lo desea, cada linea aparecida en la pantalla puede registrarse en un fichero que luego puede imprimirse o tratarse. El comando

```
CP SPOOL CONSOLE START
```

empieza a registrar las lineas que aparecen en su consola;

```
CP SPOOL CONSOLE STOP
```

detiene este registro. Para terminar la creacion del fichero, utilice el comando

```
CP CLOSE CONSOLE
```

que envia el fichero a su cola de impresion (impresora virtual). Si desea que el fichero se envie a su cola de



lectura (lectora virtual) en vez de a su impresora (para poder visualizarlo y/o modificarlo) puede hacerlo sin más que añadir la opción 'TO *' al primer comando 'SPOOL CONSOLE':

CP SPOOL CONSOLE TO * START

Veanse ejemplos de listados de consola en el Apéndice C.

Observación del estado de su consola y sus comunicaciones.

Puede ser que, avanzada la sesión, no recuerde a quien van a dirigirse los ficheros que envía, o si está imprimiendo en estado continuo, o si el registro de actividades de su consola está funcionando. El comando

CP QUERY CONS

proporciona información sobre el estado del mecanismo de registro de las actividades de su consola; el comando

CP QUERY UR

proporciona información sobre el envío, recepción y lectura de ficheros.



COMO CREAR Y MODIFICAR FICHEROS: EL EDITOR

=====

El editor es un programa del sistema operativo que ayuda a crear y modificar ficheros desde la terminal. El proceso de creacion/modificacion de un fichero se llama EDICION del fichero.

Mientras se esta editando un fichero, la terminal funciona de un modo distinto (modo 'full screen', de pantalla entera: veanse las figuras 15, 16, 17 y 18) para facilitar la edicion. La linea de comandos pasa a encontrarse en la segunda linea de la pantalla; empieza con el simbolo '===>', y se conoce como 'linea de subcomandos'. Los comandos validos son ahora los llamados 'subcomandos del editor', que afectan al fichero que se esta editando (generalmente en su primera linea - 'current line') y se introducen desde la linea de subcomandos, y los 'subcomandos de prefijo', que se explican mas abajo. Parte del fichero se visualiza en la pantalla, y puede modificarse directamente.

Mediante los subcomandos del editor, es posible visualizar cualquier parte del fichero y eventualmente realizar en ella modificaciones y/o adiciones.

Se invoca al editor mediante el comando XEDIT. El formato del comando XEDIT es:

```
XEDIT fn ft fm
```

donde 'fn ft fm' es el identificador del fichero que se quiere editar. Si se omite 'fm', se supone A1.

Vease el Apendice B para una lista de los subcomandos de editor mas frecuentes; para un estudio mas detallado, vease [XEDCMR] y [XEDUG].

Algunos subcomandos del Editor aplicados a dos ejemplos:

Ejemplo 1: creacion de un fichero.

Se pretende crear un fichero de datos numericos llamado MATRIZ DATOS A que contendra una matriz de numeros reales. Para invocar al editor, utilicese

```
XEDIT MATRIZ DATOS A
```

La pantalla se borra, y aparece una linea con el mensaje 'CREATING NEW FILE'. Para empezar a introducir datos (fig. 15), teclee a la derecha del simbolo '===>' (el cursor se encuentra alli automaticamente)



el subcomando

INPUT

Se encuentra ahora en modo de entrada de datos ('input mode': vease la figura 16). La cuarta línea de la pantalla contiene la llamada 'línea de escala', que le permite saber en cada momento en qué columna se encuentra. Esta línea es fija y no forma parte del fichero. Puede llenar la pantalla en el orden que desee, tal como escribiría sobre una hoja de papel. Para desplazar el cursor puede utilizar las teclas marcadas con flechas. Si llega a la última línea de la pantalla y todavía tiene que introducir más datos, pulse la tecla 'INTRO': la pantalla se borra de nuevo; la última línea que introdujo pasa a ser la primera de la pantalla - a continuación puede escribir más líneas (las anteriores líneas se almacenan, pero no se visualizan).

Para terminar la introducción de datos, pulse 'INTRO' dos veces. Puede ser interesante repasar el fichero en su totalidad antes de escribirlo definitivamente en disco. Para volver al principio, utilice el subcomando

TOP

(Recuerde que debe teclearse en la línea de subcomandos. Si el cursor no está en ella a la derecha del símbolo '==>', coloquelo allí pulsando la tecla 'INTRO' o mediante las teclas de desplazamiento del cursor).

Observará que junto a las líneas que introdujo, a la izquierda de la pantalla, aparecen unos números que identifican la posición de cada línea en el fichero (estos números son para facilitarle la edición, y no forman parte del fichero - vease la figura 18). Sobre estos números pueden introducirse comandos, conocidos como 'subcomandos de prefijo' (prefix subcommands); el área que contiene los números se llama 'área de prefijo' (prefix area). Su utilización se discute en el ejemplo 2.

Si observa algún error, simplemente corrijalo desplazando el cursor sobre los datos erróneos y substituyéndolos por los correctos. Si su fichero contiene más de 21 líneas, necesitará 'pasar página' hacia adelante y/o hacia atrás para poder ver todo el fichero. Para pasar página hacia adelante, utilice la tecla FP8; para pasar página hacia atrás, FP7. Si desea cambiar las líneas visualizadas en su pantalla en unidades menores de una página, puede utilizar los



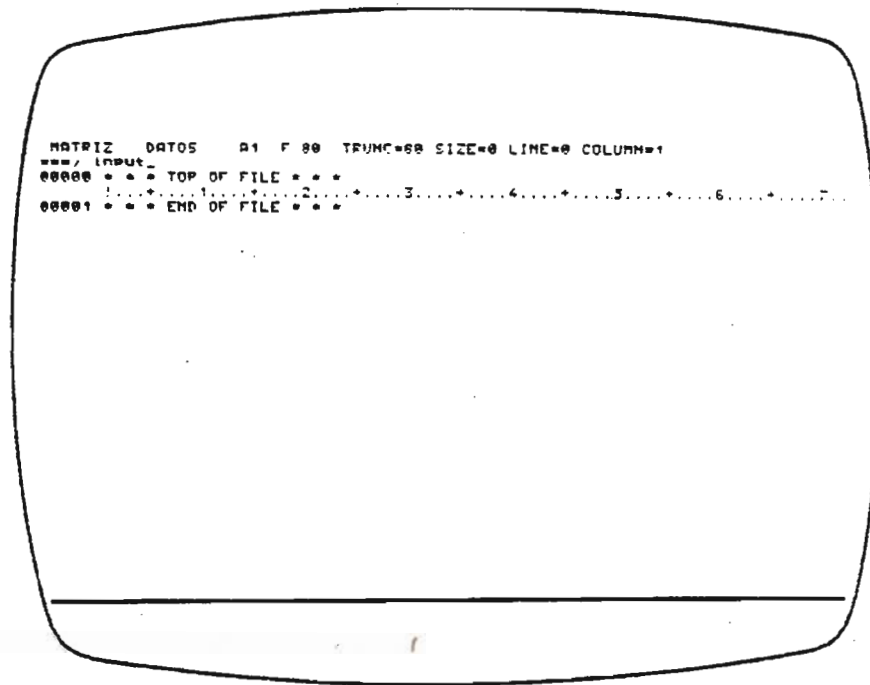


Fig. 15. Creación de un fichero. Al introducir el subcomando de editor INPUT toda la pantalla quedará disponible para introducir datos.

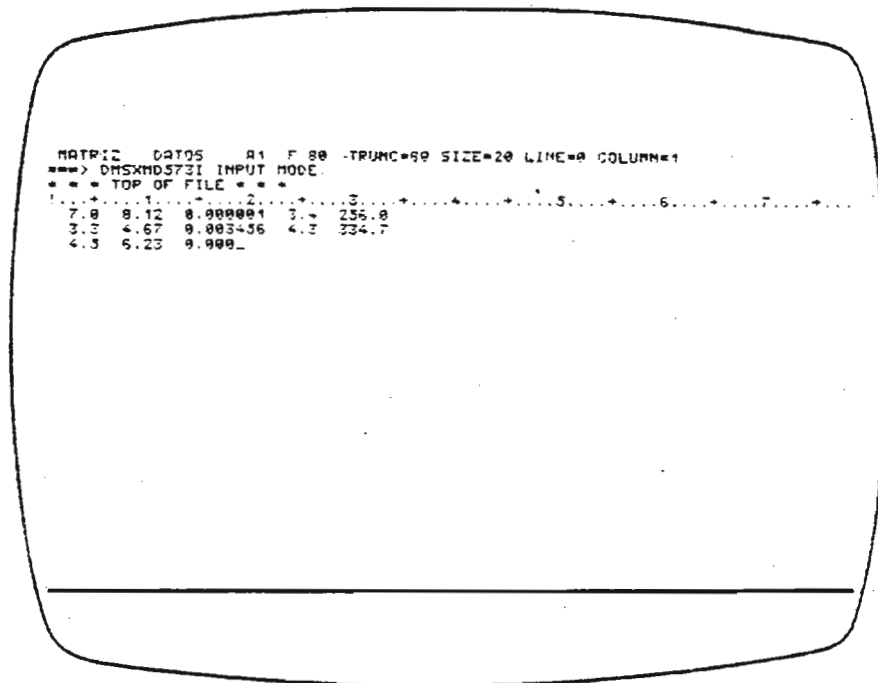


Fig. 16. Creación de un fichero. El último dato introducido, 6.28 aparece inmediatamente antes del cursor.

comandos

UP n
DOWN n (o su forma alternativa, NEXT n)

para 'subir' o 'bajar' la pantalla n lineas, respectivamente.

Una vez corregidos los posibles fallos, es necesario grabar el fichero en disco para que sea definitivamente conocido por el sistema. Esto se hace mediante el subcomando

FILE

que termina la sesion de edicion. En el caso de que NO desee grabar el fichero en disco, puede utilizar el comando

QQ

Ejemplo 2: modificacion de un fichero

Ha compilado un programa, y el listado le ha revelado la existencia de numerosos errores que desea corregir. El programa se encuentra en un fichero con identificador 'P ADA A1' - editelo mediante el comando

XEDIT P ADA A

Para localizar las lineas erroneas puede utilizar cualquiera de las tecnicas explicadas en el ejemplo 1; ademas, algunos problemas, como la reordenacion o adiccion de lineas, requieren el uso de comandos mas potentes:

Adiccion de una o mas lineas blancas debajo de una linea dada: desplace el cursor sobre el area de prefijo junto a la linea deseada y teclee 'nA' o 'An', donde 'n' es el numero de lineas que desea añadir; apriete 'INTRO', y apareceran 'n' lineas blancas con el area de prefijo intensificada. Cuando las haya llenado de informacion, el area de prefijo vuelve a su intensidad habitual (tambien puede dejarlas en blanco; si 'n' es 1, puede omitirse).

Supresion de una linea y de un bloque de lineas: tecleando una letra 'D' sobre el area de prefijo correspondiente a una linea, esta desaparece al pulsar 'INTRO'; si desea suprimir varias lineas,



identifique la primera y la ultima tecleando 'DD' sobre ellas, y pulse luego 'INTRO'.

Reordenacion, copia y duplicacion de las lineas: para mover una linea (un bloque de lineas) marquela con una 'M' (marque la primera y ultima lineas con 'MM' en el caso de un bloque) y marque la linea DESPUES de la cual desea ponerla con una 'F' (si utiliza una 'P', se colocaran ANTES - veanse las figuras 17 y 18 para un ejemplo de movimiento de un bloque de lineas). Para copiar lineas de un lugar a otro, utilice la misma tecnica, pero marcando las lineas con 'C' y 'CC' en vez de 'M' y 'MM'. Para duplicar lineas, marquelas con '' o ''''; no se necesita marcar ninguna linea con 'F' o 'P' en este caso, porque se duplican a continuacion de donde estaban.

Supresion e insercion de caracteres: si desea suprimir un caracter sobrante, utilice la tecla de supresion (fig. 5) despues de posicionar el cursor sobre ese caracter; si desea insertar uno o mas caracteres, utilice la tecla de insercion (recuerde la distincion entre caracteres blancos y caracteres vacios explicada con la tecla de insercion).

Localizacion de una linea que contenga una tira de caracteres dada: introduzca en el area de subcomandos la tira deseada encerrandola entre barras ('/'), como en

/PROCEDURE/

despues pulse 'INTRO'. La primera linea del fichero visible en la pantalla (llamada 'linea actual' o 'current line') pasara a ser la primera linea del fichero que contenga la tira a partir de la actual.

Cambio de una tira de caracteres por otra: puede efectuarse mediante el subcomando CHANGE (abreviadamente, C) -

C/FUNCION/FUNCTION/

cambia la primera ocurrencia de la tira 'FUNCION' por 'FUNCTION' en la linea actual. Tambien pueden cambiarse varias o todas las ocurrencias de una tira en una o varias lineas; hagase 'HELP CHANGE' desde el editor o 'HELP XEDIT CHANGE' desde CMS para conocer mas detalles sobre su uso.



```
DERIVADA PASCAL C1 F 88 TRUNC=88 SIZE=248 LINE=8 COLUMN=1
===>
00000 * * * TOP OF FILE * * *
00001 (asd-e)
00002 PROGRAM Derivador formal (INPUT,OUTPUT);
00003
00004 TYPE
00005
00006     sonomi = RECORD
00007         coef: REAL;
00008         exp: INTEGER
00009     END;
00010
00011     operacio = (suma,resta,mult,divi);
00012
00013     formula = 3 caja;
00014
00015     caja = RECORD CASE son: BOOLEAN OF
00016         TRUE: (m: sonomi);
00017         FALSE: (op: operacio;
00018                 s.h: formula);
00019     END;
00020
```

Fig. 17. Modificación de un fichero. Utilización de subcomandos de prefijo para colocar las instrucciones 6 a 10 después de la 12.

```
DERIVADA PASCAL C1 F 88 TRUNC=88 SIZE=248 LINE=8 COLUMN=1
===>
00000 * * * TOP OF FILE * * *
00001 (asd-e)
00002 PROGRAM Derivador formal (INPUT,OUTPUT);
00003
00004 TYPE
00005
00006     operacio = (suma,resta,mult,divi);
00007
00008     sonomi = RECORD
00009         coef: REAL;
00010         exp: INTEGER
00011     END;
00012
00013     formula = 3 caja;
00014
00015     caja = RECORD CASE son: BOOLEAN OF
00016         TRUE: (m: sonomi);
00017         FALSE: (op: operacio;
00018                 s.h: formula);
00019     END;
00020
```

Fig. 18. Modificación de un fichero. Resultado del movimiento descrito en la figura 17 una vez apretada la tecla INTRO.



COMO MODIFICAR LOS COMPONENTES DE SU MAQUINA VIRTUAL ?

=====

Si necesita mas memoria central (por ejemplo, para editar un fichero muy grande o para utilizar determinados paquetes), utilice el comando de CP

```
CP DEFINE STORAGE nnnnK
```

donde 'nnnn' es el numero de Kbytes que desea (<= 4096). Siga despues las instrucciones del sistema.

Si necesita mas espacio en disco, puede obtenerlo creando un minidisco temporal mediante la siguiente secuencia de comandos:

```
CP DEFINE T3370 193 nnnnn
```

```
FORMAT 193 m
```

donde 'nnnnn' es el numero de bloques (un bloque = 1/2 Megabyte) de disco que necesita y 'm' es el modo con el que desea acceder el disco que ha creado, es decir, es la letra que aparecera en el filemode de la identificacion de los ficheros que se coloquen en ese minidisco. El comando 'FORMAT' pregunta si realmente quiere formatear su disco (por si se hubiese equivocado) y pide una etiqueta de disco; hay que contestarle, respectivamente, 'YES' y cualquier tira no vacia de caracteres. Recuerde que el disco es temporal, por lo que la informacion en el almacenada desaparece al terminar la sesion.



COMO COMPILAR Y EJECUTAR UN PROGRAMA ?

=====

Tal como se ha descrito, mediante el Editor puede crearse un fichero que contenga un programa en lenguaje de alto nivel que puede compilarse y ejecutarse. Esto puede hacerse de forma interactiva desde la terminal o utilizando la facilidad de 'BATCH'. El Apendice C contiene ejemplos practicos de todo lo explicado en este apartado.

Compiladores

Se compila un programa mediante un comando que identifica el compilador y el filename del fichero que contiene el programa (fichero cuyo filetype debe ser caracteristico para cada lenguaje utilizado). En el caso de que existan varios ficheros con el mismo filename y filemode se determina cual es el que debe contener el programa siguiendo el orden de busqueda estandar de CMS.

PASCAL:

PASCAL filename

compila el programa escrito en PASCAL ubicado en el fichero (de identificador) 'filename PASCAL *'.

ASSEMBLER:

ASSEMBLE filename

compila el programa escrito en VM ASSEMBLER ubicado en el fichero 'filename ASSEMBLE *'.

PL/I:

PLIOPT filename [filetype [filemode]]

compila el programa escrito en PLI ubicado en el fichero 'filename filetype filemode'. Si se omite 'filemode', se busca en todos los discos accedidos. Si se omite tambien 'filetype', se supone PLIOPT o PLI.

FORTRAN:

FORTGI filename

compila el programa escrito en FORTRAN IV ubicado en el fichero 'filename FORTRAN *'.



COBOL:

COBOL filename

compila el programa escrito en COBOL ubicado en el fichero 'filename COBOL *'.

Los compiladores, al traducir un programa escrito en lenguaje de alto nivel a lenguaje maquina, producen:

un fichero con filetype 'LISTING' y el mismo filename que el del fichero fuente, que puede contener un listado del programa y de los errores que contiene, tablas de referencias cruzadas, listados del codigo maquina generado, etc;

un fichero con filetype 'TEXT' y el mismo filename que el del fichero fuente, que contiene el codigo maquina generado por el compilador; y

posiblemente otros ficheros, dependiendo del compilador y de las opciones de compilacion elegidas.

Ejecucion: LOAD, START, RUN. El comando FILEDEF. Las
TXTLIBs.

El fichero de filetype 'TEXT' producido en la compilacion de un programa no puede utilizarse directamente. Muchas de las rutinas predefinidas (como las rutinas que evaluan el seno de una expresion matematica, o las que realizan la entrada y salida) no se encuentran fisicamente en su fichero TEXT, sino que necesitan ser incorporadas a su programa para que este pueda ejecutarse. Esta incorporacion se llama 'carga del programa' y se realiza mediante el comando 'LOAD', explicado mas abajo.

Las rutinas predefinidas y algunas otras de utilidad general se encuentran en ficheros llamados librerias de rutinas o librerias de texto (TXTLIB: Text LIBrary). Estas rutinas son diferentes para cada lenguaje, por lo que es necesario definir cuales van a utilizarse mediante el comando GLOBAL antes de cargar el programa. Utilice las siguientes formas del comando GLOBAL:



| Lenguaje | Comando Global |
|-----------|--------------------------------|
| PASCAL | GLOBAL TXTLIB PASLIB |
| FORTTRAN | GLOBAL TXTLIB FORTMOD2 MOD2EEH |
| COBOL | GLOBAL TXTLIB COBLIBVS COBOLVS |
| PLI | GLOBAL TXTLIB PLILIB |
| ASSEMBLER | GLOBAL TXTLIB CMSLIB |

Solo necesita utilizar el comando 'GLOBAL' una vez por sesion (a menos que cambie de lenguaje o deba utilizar el comando 'IPL' - por ejemplo, si redefine el tamaño de su memoria). Utilice el comando

QUERY TXTLIB

para saber que TXTLIB's esta utilizando en un momento dado.

Antes de ejecutar un programa debe especificarse la ubicacion fisica de los distintos ficheros logicos de entrada y salida que utiliza (Logical Unit Numbers, como 3 o 5 en READ(12,3) X, en FORTRAN; files en PL/I, como SYSIN o SYSPRINT; External Files en PASCAL, como INPUT o OUTPUT). Por ejemplo, puede decidirse que la entrada de datos se realizara a partir del fichero en disco 'PRUEBA DATOS A1', y que la salida se dirigira al fichero en disco 'PRUEBA LISTING A1'. Esto se realiza en algunos casos automaticamente (SYSIN o INPUT, por ejemplo, estan asignados por defecto a la terminal) y en la mayoria de los casos mediante el comando 'FILEDEF'.

Formato del comando 'FILEDEF' (simplificado):

FILEDEF archivo-logico ubicacion-fisica
[(opciones)]

- * 'archivo-logico' puede ser un numero de unidad logica (para FORTRAN) o un nombre de fichero logico (como OUTPUT) para otros lenguajes.
- * 'ubicacion-fisica' puede ser:



Terminal - su terminal
PRinter - su impresora virtual
PUnch - su perforadora virtual
Reader - su lectora virtual

DISK fn ft fm - el archivo en disco con
identificador 'fn ft fm'. Si se omite 'fm', se supone
A1.

TAFn - con $1 \leq n \leq 4$: la cinta
magnetica que se encuentra en la direccion fisica 18n.
(Vease mas atras, en la descripcion de la utilizacion
de cintas).

En general, seran dos ficheros en disco. Por ejemplo:

```
FILEDEF 1 DISK PRUEBA DATOS A (RECFM F LRECL  
80 BLOCK 80  
FILEDEF 3 DISK PRUEBA LISTING A (RECFM FA  
LRECL 121
```

asignan dichos ficheros a la unidad logica '1' que
aparece en una lectura del tipo

```
READ(1,100) ...
```

y la unidad logica '3' que aparece en una escritura
del tipo

```
WRITE(3,190) ...
```

tipicos de un programa FORTRAN (para un programa PLI o
PASCAL, bastaria con cambiar '1' y '3' por 'SYSIN' y
'SYSPRINT' o 'INPUT' y 'OUTPUT' respectivamente).
Veanse las referencias [FORTUG] y [PLICMSUG].

- * 'opciones' permiten controlar el tamaño y formato de
los registros y otros detalles dependientes de
dispositivo:

RECFM a - donde 'a' es el formato de los registros:



| Formato | Significado |
|---------|---|
| F | Registros de formato Fijo |
| V | Registros de formato Variable |
| FA | Registros de formato Fijo, con caracter de control de carro. |
| FB | Registros Fijos Bloqueados |
| VB | Registros Variables Bloqueados |

LRECL nnnnn - especifica, 'nnnnn' como la longitud maxima en caracteres de los registros del fichero.

BLOCK nnnnn - especifica 'nnnnn' como la longitud maxima en caracteres de un bloque (solo para ficheros con registros bloqueados)

PERM - especifica que el comando 'FILEDEF' no necesita introducirse cada vez que se utiliza un programa, sino que es permanente, y solo puede ser anulado mediante la accion explicita del comando

FILEDEF archivo-logico CLEAR

(Ademas de por este comando, un 'FILEDEF' con la opcion 'PERM' queda tambien anulado por el comando 'HX' - explicado a continuacion - o si debe utilizarse el comando 'IPL' - por ejemplo, si se altera la cantidad de memoria central disponible).

Una vez introducidos los comandos 'GLOBAL' y 'FILEDEF' que sean necesarios, puede ejecutarse el programa mediante

LOAD filename

START

o, abreviadamente,

LOAD filename (START

Tambien puede utilizarse el comando RUN:



RUN filename

Si le interesa cancelar la ejecución de su programa antes de que este haya terminado, utilice el 'comando inmediato'

HX

Este comando se ejecuta inmediatamente (a diferencia de los demás, que esperan para ejecutarse que cualquier otro comando haya terminado). Después de utilizarlo, debe volver a definir cualquier 'FILEDEF' que estuviera utilizando, incluidos los que tenían la opción 'PERM'.

Igualmente, el comando inmediato

HT

suprime inmediatamente la impresión de líneas por pantalla SIN DETENER LA EJECUCION de su programa. El modo de impresión normal se reestablece con el comando

RT

o automáticamente al finalizar la tarea que se estaba realizando.

Para una secuencia típica de compilación, carga y ejecución, vease más adelante, en el apartado 'Como eliminar tareas repetitivas...' y el Apéndice C.

Proceso en Batch

Para tareas que requieran un cierto tiempo de proceso (veanse las normas de utilización [NORM]), se recomienda la facilidad de Batch: el usuario crea un fichero conteniendo comandos del sistema (tal como los introduciría desde la terminal) con algunas modificaciones; este fichero es transmitido por el comando 'BATCH' a unas colas de espera donde se procesa su contenido secuencialmente por una máquina virtual especial denominada 'máquina BATCH'; los resultados de la ejecución de los comandos contenidos en el fichero pueden ser impresos directamente o devueltos al usuario en su lectora virtual. La máquina 'BATCH' tiene acceso automático al disco del departamento del usuario (con modo 'D'), aunque NO pueden ejecutarse los comandos 'ALD', 'ALDR', 'ERASED' y 'NOUNOMD'. El disco 'A' del usuario NO es accedido por la facilidad de Batch, aunque existen comandos especiales para transmitir los ficheros que sean necesarios (vease el ejemplo más adelante y la



figura 19).

El formato del comando 'BATCH' es el siguiente:

```
BATCH fn [ft [fm] ]
```

donde 'fn ft fm' identifica el fichero que contiene los comandos que deben ejecutarse. Si se omite 'fm', se busca el fichero en todos los discos accedidos por el usuario secuencialmente; si tambien se omite 'ft', se supone por defecto 'BATCH'.

Los compiladores se utilizan con nombres distintos en esta modalidad:

```
ASSEMBLE debe substituirse por ASM;  
PLIOPT debe substituirse por PLI o por PL1;  
PASCAL debe substituirse por PAS;  
FORTGI debe substituirse por FOR.
```

Haga 'HELP BATCH' para conocer el formato exacto y la utilizacion de esta facilidad.

Ejemplo 1:

se desea compilar y ejecutar un programa PASCAL contenido en un fichero 'G PASCAL A1' con datos contenidos en un fichero 'PR DATA D1'. Se crea un fichero llamado 'EX BATCH A1' que contiene (los numeros son para referencia y no deben introducirse):

```
(1) PAS G  
(2) GLOBAL TXTLIB PASLIB  
(3) FILEDEF INPUT DISK PR DATA D (LRECL 80 RECFM F  
(4) LOAD G (START
```

y, despues de la terminar la edicion ('FILE') se introduce el comando

```
BATCH EX
```

El sistema pregunta interactivamente (veanse las figuras 20 y 21) sobre las caracteristicas del trabajo (por ejemplo, duracion, lineas a imprimir, Kbytes a utilizar) y decide en funcion de ellas la posicion que ocupara en las diversas colas. El usuario recibe mensajes (si esta conectado) en su pantalla cuando la facilidad de 'BATCH' recibe su programa, cuando empieza a ejecutarlo y cuando termina la ejecucion.

Cuando la maquina BATCH empieza a ejecutar este trabajo, (1) - llama al compilador de PASCAL; (2) - indica que se utilizaran las librerias de PASCAL; (3)

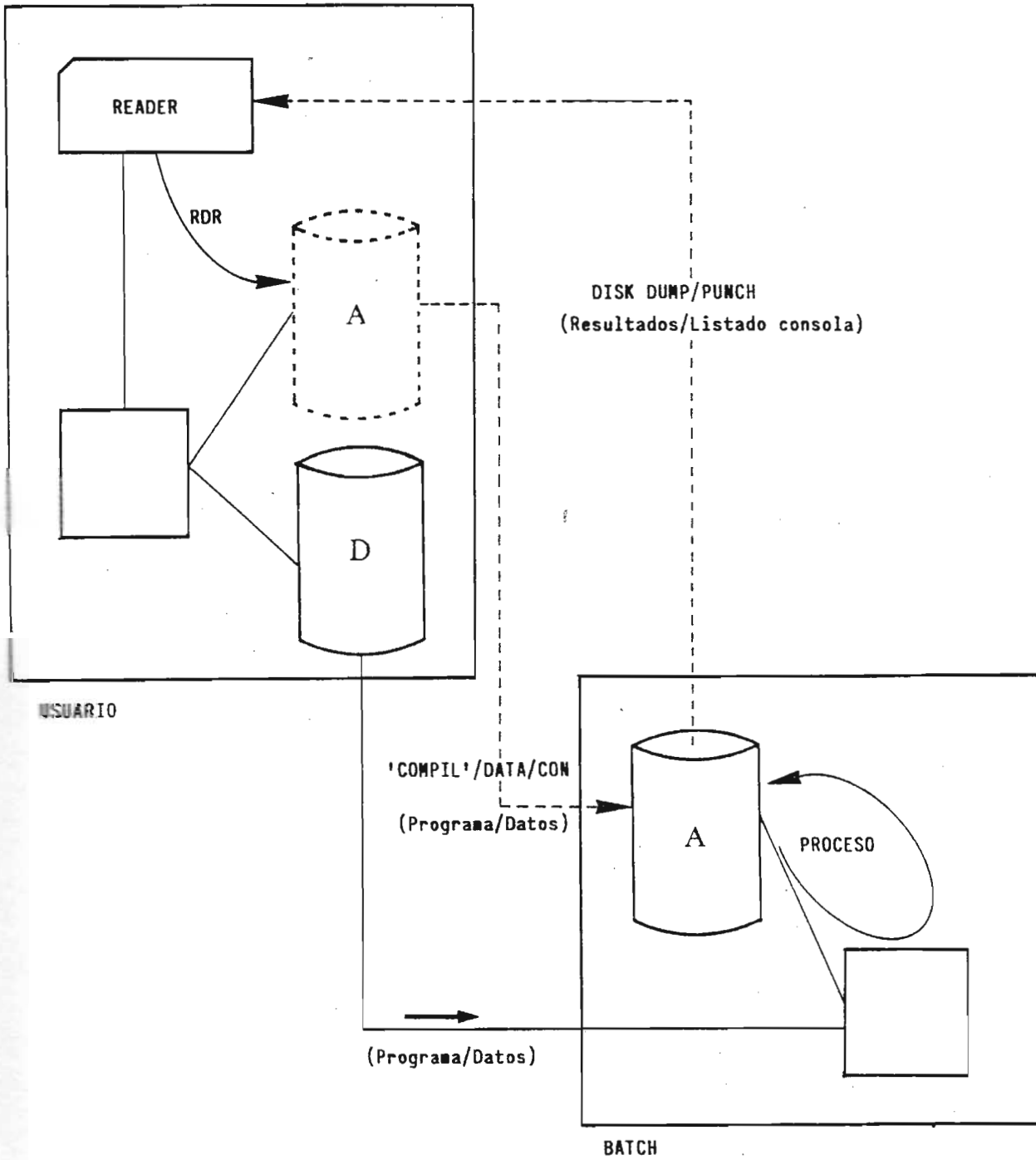


fig. 19. Esquema de utilización de la facilidad BATCH

| CONCEPTO | VALOR | RANGOS | RUNNING |
|-------------------|-------------|-----------------------------------|---------|
| NOMBRE DEL JOB | ==> ZDOCIH1 | | |
| TIEMPO DE PROCESO | ==> 300 | 300:5400 SEGUNDOS 5401:32000 " | |
| LINEAS A IMPRIMIR | ==> 1200 | | |
| K'S DEL PROCESO | ==> 312 | 312:4096 | |
| CINTAS A USAR | ==> 8 | 8:2 | |
| IMPRESORA | ==> R | R,P | |
| CONSOLA | ==> R | R,P | |

R: READER DEL USUARIO
P: IMPRESORA DEL SISTEMA

Fig. 20. Panel de opciones del BATCH. Se debe especificar las características del trabajo que se envía al lado de la correspondiente flecha.

EMPIEZA A PREPARAR ENVIO
INTRO: 'ENVIO' PARA ENVIAR AL BATCH

envio_

UN READ

Fig. 21. El trabajo se envía a BATCH al teclear ENVIO como respuesta a la instrucción que aparece en pantalla.



- el disco 'D' del usuario esta accedido tambien como disco 'D' por la maquina 'BATCH', por lo que puede hacerse el 'FILEDEF' directamente; la salida, al no haber 'FILEDEF' para el fichero 'OUTPUT', se dirige a la terminal de la maquina 'BATCH'. (4) - el programa se carga y ejecuta. Al terminar el trabajo, la maquina 'BATCH' envia al usuario su listado de consola, que en este caso (vease el punto (3)) contiene los resultados del programa. Mediante el comando RDR puede obtenerse este fichero para examinarlo (vease la seccion 'Recepcion de ficheros').

El comando 'QQ' proporciona informacion sobre el estado de las colas de BATCH, actualizada cada 15 minutos (vease la figura 22). Introduzcase simplemente

QQ

y contestese a las preguntas interactivas del sistema. El comando coloca al usuario en 'BROWSE' de un fichero con la informacion deseada (vease la explicacion del comando 'BROWSE').

Ejemplo 2:

Se desea ejecutar en BATCH un programa en FORTRAN llamado RAX que se encuentra en forma de fichero TEXT en el disco 'D' del departamento. Este programa debe leer datos contenidos en el fichero 'RAX INPUT A' a traves de la unidad logica 3, y, de ser ejecutado desde consola, necesitaria responder 'SI' y '1983' a dos preguntas interactivas; los resultados se graban en la unidad logica 11. Se edita un fichero llamado 'RAX BATCH A', que contiene:

- (1) DATA RAX INPUT A1 (LRECL 80 RECFM F
- (2) GLOBAL TXTLIB FORTMOD2 MOD2EEH
- (3) FILEDEF 3 DISK RAX INPUT A (LRECL 80 RECFM F
- (4) FILEDEF 11 DISK RESUL LISTING A (RECFM FA LRECL
- (5) LOAD RAX (START
- (6) SI
- (7) 1983
- (8) DISK DUMP RESUL LISTING A

[(1) - Copia mediante el comando 'DATA', especial de 'BATCH', el fichero 'RAX INPUT A1' del disco 'A' del usuario en el disco 'A' de la maquina BATCH, ya que de otro modo no seria accesible. (2) - Se utilizaran las librerias de FORTRAN. (3) - Una vez copiado el fichero en el disco 'A' de la maquina BATCH puede hacerse el FILEDEF adecuado con



```
R: T=0.06-0.14 14:29:02
00
ESTAT DE LES CUES DE BATCH. LA INFORMACIO S'ACTUALITZA CADA 15 M.
DASD 200 LINKED R/O; R/W BY SPOOL
QUE VOLS ?

1 : LLISTA LES CUES DEL BATCH1
2 : LLISTA LES CUES DEL BATCH2
3 : LLISTA LES CUES DEL BATCH3
SEM : LLISTA LES CUES DEL BATCHSEM
TOT : LLISTA LES CUES DE TOTS ELS BATCH'S
RES : LLISTA ELS TOTALS DE TOTS ELS BATCH'S

1- UM READ
```

Fig. 22.1. El comando QQ proporciona información, actualizada cada 5 minutos, de las colas de BATCH. Al introducir un 1 se obtiene la información que aparece en la figura 22.2.

```
CUES BATCH1 R1 P 50 3 BLKS 03/04/20 LINE 1 OF 94
==== BROWSE
ESTAT DE LES CUES DEL BATCH1
-----
CECFLLP CECFLLP 00000102 04/19 17:04:03 00003600
-----
TOTAL CECFLLP BATCH1 01 00003600
-----
CECFNGG CECFNGG 00000107 04/11 12:07:00 00003600
CECFNGG CECFNGG 00000107 04/11 13:32:32 00003600
CECFNGG CECFNGG 00000107 04/11 13:34:23 00003600
CECFNGG CECFNGG 00000107 04/11 13:35:46 00003600
CECFNGG CECFNGG 00000107 04/11 13:37:00 00003600
CECFNGG CECFNGG 00000107 04/11 13:38:43 00003600
CECFNGG CECFNGG 00000107 04/11 13:40:26 00003600
CECFNGG CECFNGG 00000107 04/11 13:41:42 00003600
CECFNGG CECFNGG 00000107 04/11 13:43:01 00003600
CECFNGG CECFNGG 00000107 04/11 13:44:04 00003600
CECFNGG CECFNGG 00000107 04/11 13:45:23 00003600
-----
TOTAL CECFNGG BATCH1 11 00039600
-----
FFTCEBG ENAPROA1 00002419 04/19 21:03:13 00003600
```

Fig. 22.2. Estado de las colas del BATCH 1.



normalidad. (4) - Asi como el FILEDEF del fichero de resultados. (5) - Carga y ejecuta el programa. (6) y (7) - Las lineas, tal y como serian introducidas desde la pantalla (7) - Envia al usuario el resultado de la ejecucion.]

y se envia al BATCH, como en el ejemplo anterior, mediante 'BATCH RAX'.



COMO UTILIZAR PROGRAMAS Y PAQUETES DE APLICACIONES ?

Tal como se menciona en la introduccion, el Centro de Calculo ha implementado una serie de programas y paquetes de aplicaciones que permiten realizar procesos complejos sin tener que programarlos.

Se puede obtener informacion de todos ellos introduciendo su nombre o tecleando

HELP nombre-de-paquete

Por ejemplo:

BMPF

proporciona informacion sobre el comando BMPF que permite invocar cada uno de los 45 programas de este paquete estadistico, asi como sobre las caracteristicas principales de cada uno de ellos.

Paquetes de subrutinas

Antes de ejecutar un programa que invoque a alguna de las subrutinas del paquete SSP, debe introducirse el comando

SSP lenguaje opcion

donde 'lenguaje' puede ser FOR o PL/I, segun que se invoquen desde un programa escrito en FORTRAN o en PL/I, y 'opcion' puede ser DOCU o CARGA, segun que se desee informacion sobre una subrutina concreta o bien que se efectue automaticamente el 'GLOBAL' necesario para que al cargar el programa ('LOAD') se encuentre la subrutina y se incorpore al programa, que ya se podra ejecutar ('START').

Para una mayor informacion sobre las subrutinas SSP consultese las referencias [SSPF] y [SSPP].

Analogamente se invocan las subrutinas del paquete NAG mediante el comando

NAG opcion

donde opcion puede ser DOCU o CARGA con el mismo significado que en el caso anterior (en este caso solo estan en version FORTRAN).

Para una mayor informacion sobre las subrutinas NAG



consultense las referencias [NAGM] y [NAGLM].

Programas y paquetes de programas

BMDP (Biomedical Statistical Software)

Se puede ejecutar cada uno de los programas del paquete BMDP mediante el comando

```
BMDP fn ft fm (id opciones
```

donde 'fn ft fm' es la identificación del fichero con las instrucciones propias del BMDP y los datos a tratar (veanse las referencias [BMDPG] y [BMDPC]) 'id' es la identificación del programa (por ejemplo bmdp2t) las distintas opciones permiten obtener los resultados en disco, listado, etc.; efectuar transformaciones de los datos; crear un fichero BMDP; etc. (introduzca el comando BMDP para obtener información mas detallada).

El comando BMDP prepara los elementos para ejecutar el programa 'id' con los datos del fichero 'fn ft fm', preguntando si el usuario desea utilizar alguna TXTLIB propia (Para el caso en que se deseen efectuar transformaciones en los datos o invocar a una función); si no se necesita, no hay mas que contestar pulsando la tecla 'INTRO', tras lo cual se ejecuta automáticamente el programa y se obtienen los resultados que se colocan en un fichero con identificador 'fn LISTING A' (es decir, queda en el disco A del usuario), donde 'fn' es el mismo filename del fichero de datos.

Dado que alguno de los programas BMDP consumen mucho tiempo de CPU conviene utilizar la facilidad BATCH explicada mas arriba.

Ejemplo: se desea tratar los datos contenidos en el fichero 'PRUEBA DATOS D', cuyo contenido se refleja en la figura 23, mediante el programa BMDP6D, que realiza diagramas bivariantes de puntos (scatter). Para enviarlo a ejecutar mediante la facilidad BATCH, debe prepararse un fichero con filetype BATCH, PRUEBA BATCH A, que contenga las siguientes instrucciones (los numeros son para referencia y no deben reproducirse):

- (1) BMDP PRUEBA DATOS D (BMDP6D PRINT
- (2) CON PRUEBA CONS A (LRECL 80 RECFM F



```
PRUEBA DATOS D1 F 88 39 BLKS 83/04/18 LINE 1 OF 488 BROWSE
***
/BMDP6D
/PROBLEM TITLE IS 'ESTUDIO ALTURAS ALUMNOS-PADRES-PAREJAS. REGRESIONES'.
/INPUT VARIABLES ARE 14.
/FORMAT = '(1X,A4,F2.0,2F3.0,F5.1,F3.1,2(F3.0,F5.1),2F3.0,F6.1,F2.0)'.
/VARIABLE NAMES ARE ID,SEXO,PROU,EDAD,ALTURA,PESO,
'PROU.P','ALTURA.F','PROU.M','ALTURA.M',
'PROU.PAR','EDAD.PAR','ALT.PAR',
ELECCION.
/LABEL IS ID.
/PLOT XUAR = ALTURA.
YUAR = 'ALTURA.P','ALTURA.M'.
COMMON.
STATISTICS.
/END
00001 1 00 18 164.0 51.0 00 178.0 00 154.0 00 18 179.0 1
2 1 00 19 139.0 45. 00 164. 03 163. 41 22 174. 1
3 1 00 19 160.1 51. 36 154. 36 167. 08 19 170. 1
4 1 29 19 151. 53. 51 156. 42 159. 08 21 183. 1
5 1 00 20 169. 74. 20 176. 08 166. 08 21 179. 1
6 1 00 18 175. 69. 99 184. 08 164. 08 19 180. 1
7 1 00 19 163.5 57.5 08 189. 08 138. 08 20 173. 1
8 1 93 21 156. 54. 52 163. 38 154. 08 23 178. 1
```

Fig. 23.1. Fichero de datos preparado para ser procesado por el programa BMDP 6D. Las primeras líneas son instrucciones de BMDP. Se visualizan los datos correspondientes a los 8 primeros individuos del total de 473. El resultado aparece en el ejemplo 3 del Apéndice C.

```
CONSOLA PRUEBA A1 F 133 3 BLKS 83/04/18 LINE 1 OF 22 BROWSE
***
1'19E' REPLACES 'Y (19E)'
Y (19E) R/O
P (319) R/O
X (31A) R/O
H (328) R/O
/SET TIME 300 PRINT 1200
EXEC BATCHINI ZDOCINI TO TO ZDOCINI T300
DEV 991 DOES NOT EXIST
R: T=0.15/0.38 18:06:30
BMDP PRUEBA DATOS D ( BMDP6D PRINT
SI DESER UTILIZAR ALGUNA TXTLIB PROPIA INTRODUCZA LOS CORRESPON-
DIENTES NOMBRES (HASTA UN MAXIMO DE 3). EN CASO CONTARIO INTRODUCZA
UNA LINEA NULA (PRESIONE LA TECLA INTRO).
Loading BMDP6D ...
EXECUTION BEGINS...
DASD 198 DETACHED.
R: T=7.72/9.54 18:07:02
/*
CMSBATCH ENDED.
PRT FILE 2826 TO ZDOCINI COPY 001 NOHOLD
```

Fig. 23.2. Listado de consola enviado por la máquina BATCH tras ejecutar el programa de la figura anterior. Obsérvese la línea en blanco contestando a la petición de TXTLIB.



(donde 'PRUEBA CONS A' es un fichero que contiene una línea en blanco y representa la respuesta nula al comando 'BMDF' - hagase 'HELP BATCH' para obtener información sobre el comando 'CON'). Después (tal como se ha expuesto en el apartado 'proceso en Batch') se introduce desde la terminal el comando

BATCH PRUEBA

se rellenan las opciones del panel que aparece a continuación, se aprieta 'INTRO', se contesta a la pregunta sobre el envío tecleando 'ENVIO' e 'INTRO' y se esperan los mensajes de la máquina BATCH (mientras tanto, se pueden realizar por terminal otros trabajos, en particular enviar otros trabajos a BATCH).

Cuando la máquina BATCH empiece este trabajo, efectuará:

- (1) ejecuta el comando 'BMDF', invocando al programa 'BMDF' y localizando el fichero de datos en el disco D del usuario (ATENCIÓN: si el fichero de datos se encuentra en el disco A del usuario, la máquina BATCH no lo encontrará y se producirá un error). Notese la opción 'PRINT' del comando 'BMDF', que provoca el envío de información impresa al usuario.
- (2) contesta con una línea en blanco a la pregunta sobre TXTLIB's propias (NOTA 1: es necesario el uso del comando de BATCH 'CON', ya que las líneas en blanco escritas directamente en un fichero de tipo BATCH se ignoran. NOTA 2: en caso de que se necesiten TXTLIB's, no hay más que ponerlas en la única línea del fichero 'PRUEBA CONS A').

La máquina BATCH ejecuta el programa y produce unos resultados que envía directamente al usuario (debido a la opción 'PRINT' del comando 'BMDF'). El usuario los recibe en su lectora virtual, y puede copiarlos en disco tal como se describe en el apartado 'recepción de ficheros'.

A continuación no hay más que leer los ficheros de consola y resultados desde la propia lectora virtual.

Si se desea que la máquina BATCH imprima directamente los resultados, debe prepararse el mismo fichero 'PRUEBA BATCH D' descrito y colocar una P en la línea del panel de opciones



(vease fig. 20) que aparece al introducir el comando 'BATCH PRUEBA'.

SCSS Conversational System Data Analysis

El SCSS es en realidad un gran programa que se ejecuta interactivamente y va pidiendo informacion al usuario sobre los procesos estadisticos que desea realizar con los datos que se pueden introducir tambien de forma interactiva.

Para empezar a ejecutar el programa tan solo debe introducirse el comando

SCSS START

y seguir las instrucciones que el mismo programa va proporcionando.

El programa SCSS va haciendo preguntas e informa de las posibles respuestas. Estas preguntas se denominan 'prompts' y son distintas segun en que fase del proceso se este. tambien pueden introducirse instrucciones propias de scss denominadas 'preempts', que son unos comandos que siempre van precedidos de /:

/preemt especificaciones

donde especificaciones son las distintas opciones del comando, por ejemplo:

/LIST VOCABULARY

proporciona una lista de 'prompts' y 'preempts' en el contexto en que se encuentre en ese momento el programa.

Los datos a tratar por SCSS deben colocarse en un fichero especial denominado MASTER, que se puede construir introduciendo los datos desde la terminal o bien trasladandolos ('transposition') desde un fichero normal ubicado en alguno de los discos del usuario. Una vez construido se mantiene inalterado de una sesion a otra. Para efectuar modificaciones y transformaciones de los datos durante la sesion se utiliza un fichero de trabajo denominado WORKFILE que puede mantenerse de una sesion a otra o eliminarse si no interesa conservar las alteraciones



efectuadas a los datos del MASTER, que se conserva intacto.

Por ello, lo primero que pregunta SCSS, despues de ofrecer informacion, es si existen ya un fichero WORKFILE y un MASTER y, si se contesta negativamente, se dispone a crearlos.

Una vez identificados o creados los ficheros mencionados el programa pregunta sobre que proceso (PROCEDURE) estadistico de los disponibles se quiere realizar con los datos. Una vez contestada esta pregunta se entra el correspondiente 'procedure' y sigue preguntando sobre las distintas opciones del mismo hasta que esta en condiciones de efectuar el tratamiento solicitado, cuyo resultado aparecera en la terminal. Si se quiere guardar en un fichero, para su posterior examen e impresion, los comandos y resultados de una sesion SCSS se debe introducir el 'preempt':

```
/LOGFILE BOTH, filename
```

Lo que provocara que toda la sesion SCSS a partir de ese momento, hasta el momento en que se teclee

```
/LOGFILE OFF
```

o bien se abandone SCSS, quedara registrado en un fichero con el filename especificado.

En cualquier momento se puede pasar de un 'procedure' a otro simplemente introduciendo el 'preempt'

```
/PROCEDURE nombre_de_procedure
```

y se puede seguir entonces contestando a las preguntas que se haga desde el nuevo 'procedure'. La lista de 'procedures' es:



| procedure | funcion |
|--------------|---|
| UNIVARIATE | estadística descriptiva |
| CROSSTABS | tablas de frecuencias |
| TTEST | test t Student |
| BREAKDOWN | análisis de la varianza |
| SCATTERPLOT | diagramas bivariantes de puntos regresión lineal |
| CORRELATIONS | Pearson |
| PARTIALS | correlación parcial |
| REGRESSION | regresión lineal múltiple |
| FACTOR | análisis factorial |

El programa SCSS proporciona diversas ayudas para orientar al usuario en su utilización. De entrada pregunta si se desea una explicación general mediante el 'prompt'

EXPLAIN?(Enter yes or no)

explicación que se proporciona si se contesta 'yes'. A continuación pregunta sobre como se desean las preguntas o 'prompts':

STYLE OF PROMPTING?

a lo que puede contestarse 'verbose', 'normal' o 'terse' según que se desee en cada pregunta una referencia sobre las posibles respuestas, ninguna referencia o la pregunta abreviada.

También puede pedirse información sobre cualquier 'preempt' o 'prompt' mediante el comando EXPLAIN. Por ejemplo:

/EXPLAIN PROCEDURE

proporciona información sobre el 'prompt' PROCEDURE



(puede hacerse /LIST VOCABULARY FOR CONTEXT GLOBAL para obtener una lista de 'prompts'. Tambien puede contestarse EXPLAIN a cualquier pregunta y se proporciona informacion extensa. Si se desea una informacion mas reducida puede responderse a cualquier pregunta con /HELP o bien ?.

Para finalizar la sesion SCSS se introduce el 'preempt'

/STOP

A continuacion puede estudiarse la sesion y resultados en el fichero fn SCSSLIST A generado a partir del 'preempt' /LOGFILE explicado antes.

En el apendice D puede verse una sesion tipica de SCSS.

Para obtener mas informacion sobre las caracteristicas del SCSS consultese las referencias [SCSS1] y [SCSSG].

CLUSTAN Cluster Analysis Package

Este paquete se invoca mediante el comando

CLUSTAN fn ft fm1 fm2

donde 'fn ft fm1' es la identificacion del fichero que contiene las instrucciones propias del CLUSTAN (vease la referencia [CLUS]) y los datos a tratar, mientras que 'fm2' es el modo del disco donde se quiere que se graven los resultados del proceso (si se omite se colocaran en el disco A).



COMO ELIMINAR TAREAS REPETITIVAS MEDIANTE LA CREACION DE
=====

MODULOS Y DE FICHEROS DE COMANDOS DEL SISTEMA (EXEC'S) ?
=====

Modulos

Si desea utilizar un mismo programa frecuentemente, puede crear un comando que especifique la ejecucion de ese programa. Para ello, despues de cargar el programa mediante el comando 'LOAD', utilice el comando

GENMOD nombre

donde 'nombre' es el nombre del comando que esta generando. En su disco 'A' aparecera un nuevo fichero con identificador 'nombre MODULE A1' que contiene el codigo maquina necesario para ejecutar el programa. A partir de la ejecucion del comando 'GENMOD', 'nombre' pasa a incorporarse a la lista de comandos utilizables: sin mas que introducir 'nombre' desde la terminal, el programa que contiene empieza a ejecutarse.

Ejemplo:

Se ha compilado correctamente un programa escrito en cualquier lenguaje y con filename 'PRUEBA', y, por tanto, existe en alguno de los discos del usuario el fichero 'PRUEBA TEXT *'. Ademas, suponemos que se ha emitido el comando 'GLOBAL' pertinente. Entonces, la secuencia

```
LOAD PRUEBA
GENMOD COM1
```

establece 'COM1' como el nombre de un nuevo comando que especifica la ejecucion del programa 'PRUEBA'.

NOTA: Si un programa del usuario se coloca como MODULE en el disco 'D', es utilizable desde el BATCH sin compilacion ni carga alguna.

EXEC'S

Despues de utilizar unas cuantas veces el sistema, notara que hay secuencias de comandos que utiliza a menudo. Cualquier secuencia de comandos puede almacenarse en un fichero (con filetype EXEC) y dirigir luego al sistema operativo para que la ejecute. Por ejemplo, una secuencia tipica de compilacion, impresion



del listado, carga y ejecucion es (los numeros solo sirven para referencia posterior y no deben introducirse):

- (1) PLIOPT X
- (2) PRINT X LISTING
- (3) GLOBAL TXTLIB PLILIB
- (4) FILEDEF SYSIN TAP1
- (5) FILEDEF SYSPRINT DISK RESULT DATOS A1
- (6) LOAD X (START

[Comentarios: (1) - compila el programa 'X PLIOPT'. (2) - Imprime el listado de la compilacion. (3) - Accede a las librerias de PL/I. (4) - Los datos de entrada se leen desde cinta. (5) - Los datos de salida se graban en un fichero en disco con identificador 'RESULT DATOS A1'. (6) - Carga y ejecuta el programa; equivalente a 'RUN X'.]

Si se crea un fichero llamado 'P EXEC A' que contenga estos comandos, puede utilizarse 'P' (el filename del fichero) como si fuese un comando. Al introducir 'P' en la terminal, pasan a ejecutarse los comandos contenidos en 'P EXEC A'.

NOTA: al utilizarlos dentro de un EXEC, los comandos de CP DEBEN llevar el prefijo 'CP', y los EXECs pueden invocarse solo si se preceden del prefijo 'EXEC'. Asi, el comando de CP 'QUERY VIRT', si aparece dentro de un fichero EXEC, debe escribirse como

```
CP QUERY VIRT
```

y el comando de CMS 'RUN', que esta contenido en un fichero EXEC, puede utilizarse escribiendo

```
EXEC RUN fn
```

Este mecanismo puede extenderse mediante la parametrizacion y la introduccion de sentencias de control: los ficheros de comandos pueden utilizar comandos y sentencias similares a los de otros lenguajes de programacion para programar tareas complicadas que utilicen funciones del sistema. Por ejemplo, si en el ejemplo anterior quiere condicionarse la ejecucion de los comandos al exito de la compilacion, puede añadirse una linea

```
(1 bis) - &IF &RC > 4 &EXIT
```

que indica: si el resultado de la compilacion produjo errores (expresados mediante el codigo de retorno [RETURN CODE, R. C.] que aparece entre parentesis despues de la



letra 'R' al ejecutarse un comando) de gravedad mayor que 4 (esto es, errores de tipo E, S o U), terminese (EXIT: salir) la ejecución del programa.

Otro ejemplo: el comando 'RUN' del sistema es en realidad un fichero de comandos: hagase 'BROWSE RUN EXEC *' para comprobarlo.

Existen dos procesadores de ficheros de comandos, llamados EXEC y EXEC2. EXEC permite acceder directamente a determinadas variables del sistema, y esta documentado en el manual [CMSCMR].

EXEC2 carece de variables del sistema, pero es mas abstracto, general y potente - ademas, puede utilizarse para crear ficheros de comandos de editor (con filetype XEDIT). Esta documentado en el manual [EX2R]. Los ficheros de comandos de editor (XEDIT) se explican tambien en los manuales del editor (consultese [XEDUG] y [XEDCMR]).



MANUALES Y REFERENCIAS PARA EL ESTUDIO POSTERIOR.

=====

[NORM] - Normes provisionals d'utilització dels recursos informàtics del Centre de Càlcul de la Universitat de Barcelona.

[VMSPIN] - IBM Virtual Machine/System Product: Introduction

Contiene una introducción a VM/SP, y describe brevemente CP, CMS, IPCS y RSCS.

[CPCR] - IBM Virtual Machine/System Product: CP Command Reference for General Users. (SC19-6211-0)

Es el manual de referencia para los comandos de CP.

[VMSPTUG] - IBM Virtual Machine/System Product: Terminal User's Guide. (GC19-6196-0).

Explica el manejo de la terminal, la utilización de las teclas de función de programa (o teclas de programa, PF keys), el proceso de LOGON y LOGOFF, las convenciones de escritura a seguir al utilizar las terminales, y la utilización de los conjuntos de caracteres extendidos para APL y TEXT.

[CMSCMR] - IBM Virtual Machine/System Product: CMS Command and Macro Reference (SC19-6209-0).

Es el manual de referencia de los comandos y macro instrucciones de ensamblador de CMS, así como del editor EDIT y del lenguaje de control EXEC.

[CMSUG] - IBM Virtual Machine/System Product: CMS User's Guide (SC19-6210-0).

Contiene ejemplos detallados de utilización de todas las facilidades del sistema CMS: Editor, Exec.

[PAS] - Kathleen Jensen & Niklaus Wirth: 'PASCAL User Manual and Report' Second Corrected Reprint of the Second Edition. Springer-Verlag, Berlin, 1978.

Es el manual del lenguaje PASCAL. Orientado a lectores con una cierta experiencia en programación, se divide en



dos partes: La primera parte introduce los conceptos del lenguaje de un modo estructurado y a base de ejemplos. La segunda parte es el manual de referencia del lenguaje. Contiene listados de los errores de compilacion.

[XEDUG] - IBM Virtual Machine/System Product: System Product Editor User's Guide. (SC24-5219-0).

Describe el editor XEDIT para usuarios sin experiencia. No es un manual de referencia, sino una guia para el usuario, que puede utilizarse como libro de estudio. Contiene numerosos ejemplos.

[XEDCMR] - IBM Virtual Machine/System Product: System Product Editor Command and Macro Reference. (SC24-5221-0)

Es el manual de referencia para los comandos del editor. Describe los 'return codes' y mensajes de error para todos los comandos. Contiene algunos ejemplos.

[EX2R1] - IBM Virtual Machine/System Product: EXEC2 Reference. (SC24-5219-0).

Describe el lenguaje EXEC2 a todos los niveles. Contiene capitulos introductorios para usuarios sin experiencia, asi como referencias completas de todos los aspectos del lenguaje. Se compara el lenguaje con su homonimo EXEC. Contiene explicaciones detalladas sobre la utilizacion de ficheros EXEC2 desde el editor XEDIT (ficheros de filetype XEDIT), y, en general, desde cualquier programa mediante la utilizacion de rutinas en lenguaje ensamblador (ASSEMBLE).

[PLIMES] - OS PL/I Optimizing Compiler: Messages (SC33-0027-5).

Contiene explicaciones y descripciones detalladas de los mensajes de error producidos por el compilador de PL/I, tanto en tiempo de compilacion (mensajes IEL0nnnX) como de ejecucion (mensajes IBMnnnX). Con ejemplos.

[PLICMSUG] - OS PL/I Optimizing Compiler: CMS User's Guide. (SC33-0037-3).

Explica como utilizar el compilador PLIOPT bajo CMS. Esta redactado para ser comprensible por usuarios con muy pocos conocimientos del sistema operativo. Explica y discute a fondo el formato y las opciones del comando



PLIOPT, así como las opciones en tiempo de ejecución.
Contiene ejemplos.

[FORLAN] - IBM System/360 and System/370 FORTRAN IV
Language. (GC28-6515-10).

Es el manual de referencia para el lenguaje FORTRAN.
Discute simultáneamente todos los niveles del lenguaje.
Contiene ejemplos.

[FORTUG] - IBM VM/370 (CMS) Terminal User's Guide for
FORTRAN IV Program Products. (SC28-6891-1).

Contiene toda la información necesaria para utilizar
FORTRAN bajo CMS en compilación y en ejecución. Discute
el comando FILEDEF para FORTRAN, y estudia los formatos
de ficheros de entrada y salida. Contiene ejemplos.

[FORMAT] - IBM System/360 Operating System FORTRAN IV
Library - Mathematical and Service Subprograms.
(GC28-6818-1)

Explica detalladamente el funcionamiento de las rutinas
matemáticas y de servicio utilizadas por el compilador de
FORTRAN (como las rutinas trigonométricas).

[COBFG] - IBM OS/VS COBOL Compiler and Library Programmer's
Guide (SC28-6483-2).

Explica la utilización y ejecución de programas COBOL
bajo OS. Contiene ejemplos y figuras.

[COBCMSUG] - IBM VM/370 CMS User's Guide for COBOL

Explica como utilizar el compilador de COBOL.

[COBLAN] - IBM VS COBOL for OS/VS

Es el manual del lenguaje COBOL.

[SSPF]- System/360 SCIENTIFIC SUBROUTINE PACKAGE
(360-CM-03x) Version III. Programmer's manual.

Es el manual de las SSP en FORTRAN.



[SSPP1]- SYSTEM/360 SCIENTIFIC SUBROUTINE PACKAGE (PL/I).
Program description and operation manual (360A-CM-07X).

Es el manual de las SSP en PL/I.

[NAGM]- NAG FORTRAN mini manual. Mark 9

[NAGLM]- NAG FORTRAN Library manual. Mark 9, vols. 1 a 6.

[BMDPG]- DIXON, W.J. (Ed.). BMDP Statistical Software 1981.
Univ. California Press.

[BMDPD]- HILL, M.A. (1982). BMDP User's digest.

Es una guía condensada de la anterior.

[SCSSII]- SOURS, K.J. (1982). SCSS short guide. An introduction
to the SCSS conversational system. Mc Graw-hill Co.

[SCCSG]- NIE, N.H. et al. (1980). SCSS a user's guide to the
SCSS conversational system. Mc Graw-hill Co.

[CLUS]- WISHART, D. (1982). CLUSTAN User manual. Program
Library Unit. Edinburgh Univ.



APENDICE A: LISTA DE LOS COMANDOS MAS UTILIZADOS

=====

Se describen primero los comandos de CMS, y luego los de CP. Puede obtenerse mas informacion mediante el comando HELP o consultando los manuales [CMSCMR] y [CFCR].

Comandos de CMS:

| | |
|----------|--|
| ACcess | accede un disco con un modo determinado |
| Assemble | compila un programa en lenguaje ensamblador |
| BROWSE | permite visualizar un fichero |
| COMpare | compara dos ficheros en disco |
| COPYfile | copia ficheros en disco |
| CP | coloca al usuario en el entorno CP |
| DISK | envia y recupera ficheros de cualquier formato mediante la lectora y perforadora virtuales |
| ERASE | borra un fichero en disco |
| EXec | ejecuta un fichero de comandos |
| FILEdef | define la ubicacion fisica de un fichero |
| FLIST | permite visualizar uno o varios (fragmentos de) directorios de ficheros |
| FORMAT | formatea un disco para su utilizacion por CMS |
| Genmod | genera un modulo ejecutable |
| GLobal | define las librerias a las que accede el usuario |
| Help | proporciona informacion sobre la utilizacion del sistema |
| HT | detiene la impresion de informacion en pantalla sin detener la ejecucion del programa |



| | |
|---------------|--|
| HX | termina incondicionalmente la ejecución de un programa |
| Listfile | lista por pantalla conjuntos de ficheros |
| LOAD | carga en memoria un programa, preparandolo para la ejecución |
| LOADMod | carga en memoria un modulo ejecutable |
| MOVEfile | copia ficheros residentes en cualquier medio fisico |
| PRint | imprime un fichero (Listado) en su impresora virtual |
| PUnch | perfora un fichero en su perforadora virtual o coloca un fichero en la lectora virtual de otro usuario |
| Query | pregunta sobre el estado del sistema |
| READcard | lee un fichero que esta en su lectora virtual |
| RELease | termina la asociacion de un modo con un disco logico conseguida mediante el comando ACCESS |
| Rename | cambia el identificador de un fichero en disco |
| RT | Libera la impresion de informacion en pantalla previamente retenida mediante el comando HT |
| RUN | condicionalmente compila, carga y ejecuta un programa dependiendo de las opciones |
| SET | da valor a diferentes variables y opciones del sistema |
| SORT | ordena un fichero en disco |
| START | empieza la ejecución de un programa previamente cargado mediante el comando LOAD o LOADMOD |
| STATE, STATEW | verifican la existencia de un fichero |
| TAPE | vuelca en cinta o recupera ficheros en |



formato CMS y controla la unidad de cinta

Type permite visualizar (fragmentos de) un fichero por pantalla

Xedit invoca al editor del sistema

Consultese el manual [CMSCMR] para una lista completa de los comandos de CMS, sus formatos y abreviaciones.

Comandos de CP:

Begin devuelve el control desde CP a CMS

CHange permite cambiar las características de sus ficheros de SPOOL

CLOse termina la impresion de un fichero

CP no hace nada

DEFine permite redefinir la memoria o los minidisos

DETach devuelve un minidisco al sistema

Ipl pone en marcha el CMS

LOGoff termina la sesion

Logon comienza la sesion

Message envia mensajes a otros usuarios

NCP permite dar comandos DE CP mientras se ejecuta un programa (hagase HELP NUMCP)

ORDer reordena sus archivos de SPOOL (e.g., ficheros de lectora, impresora o perforadora virtual).

PURge suprime la impresion de archivos de SPOOL (e.g., listados) que todavia se encuantan bajo su control

QUERY obtiene informacion sobre el estado de elementos del sistema

SET da valor a diferentes variables y opciones del sistema



SPOOL controla el funcionamiento de su lectura, perforadora, impresora, y consola virtuales.

TERMINAL cambia algunas características del funcionamiento de su terminal

Consultese el manual [CPCR] para una lista completa de los comandos de CP, sus formatos y abreviaciones.



APENDICE B: LISTA DE LOS SUBCOMANDOS DEL EDITOR (XEDIT) MAS
=====

UTILIZADOS
=====

Se describen los subcomandos, los operandos del subcomando SET, los comandos utilizables desde el area de prefijo y las funciones asignadas a la mayoria de las teclas FFn. Puede obtenerse mas informacion mediante el comando 'HELP XEDIT subcomando' o consultando el manual [XEDCMR].

Subcomandos:

| | |
|----------|---|
| Add | añade cualquier numero de lineas despues de la primera ('current line') |
| BACKward | pasa pagina(s) hacia atras |
| Bottom | coloca la ultima linea del fichero en la 'current line'. |
| CANCEL | cancela la ejecucion del Editor |
| Change | cambia una tira de caracteres por otra |
| CMS | permite utilizar comandos de CMS sin abandonar el Editor |
| COunt | cuenta las apariciones de una tira de caracteres |
| CP | permite utilizar comandos de CP sin abandonar el Editor |
| DELeTe | borra una o mas lineas |
| Down | la linea actual baja (la pantalla sube) |
| DUPlicat | permite duplicar una o mas lineas |
| FILE | graba el fichero en disco terminando la edicion |
| Find | busca una linea que empiece con determinada tira de caracteres. |
| FINDUp | busca hacia arriba una linea que empiece con determinada tira de caracteres |



| | |
|----------|---|
| FOrward | pasa pagina hacia atras |
| GET | copia (un trozo de) un fichero sobre el que se esta editando |
| Help | obtiene informacion sobre la utilizacion del Editor |
| Input | pasa a modo de entrada (input mode). |
| Locate | busca una tira de caracteres |
| LOWercas | pasa a minusculas una o mas lineas |
| Next | la 'current line' baja (la pantalla sube) |
| NFind | busca una linea que no empiece con determinada tira de caracteres |
| NFindUp | busca hacia arriba una linea que no empiece con una determinada tira de caracteres |
| Overlay | superpone una tira de caracteres a la linea actual |
| POWerinp | permite entrar datos sin preocuparse por la longitud de linea |
| PUT | copia (parte de) el fichero que se esta editando en un fichero en disco |
| PUTD | vuelca (parte de) el fichero que se esta editando en disco, de modo que su recuperacion sea automatica mediante el subcomando 'GET' |
| Query | permite visualizar el estado de diferentes variables y opciones del editor |
| QUIT | termina la edicion sin grabar en disco el fichero |
| RECOVer | recupera las ultimas lineas suprimidas |
| REPEAt | repite en comando un cada linea hasta un tope |
| ReplacE | reemplaza la linea actual |



| | |
|----------|--|
| RESet | cancela el area de prefijo |
| SAVE | graba el fichero en disco sin terminar la edicion |
| SET | permite modificar el estado del Editor y de sus variables internas |
| SORT | permite ordenar un numero de lineas alfabeticamente |
| TOP | La primera linea del fichero pasa a ser la 'current line' |
| Up | la linea actual sube (la pantalla baja) |
| UPPerCas | traduce a mayusculas una o mas lineas |
| Xedit | permite editar simultaneamente mas de un fichero |
| & | fija un comando en la linea de comandos |
| = | vuelve a ejecutar el ultimo comando ejecutado |
| ? | muestra el ultimo comando ejecutado |

Operandos del subcomando SET

| | |
|----------|---|
| ARBchar | selecciona un caracter como 'caracter arbitrario', para su utilizacion en comandos LOCATE Y xFINDxx |
| Autosave | determina el limite de alteraciones tras las cuales el fichero se graba automaticamente en disco |
| CASE | decide entre mayusculas o mayusculas y minusculas |
| CMDline | permite cambiar la posicion de la linea de comandos |
| CURLine | permite cambiar la posicion en pantalla de la linea actual |
| FMode | Cambia el filemode del fichero que se esta editando |



| | |
|----------|---|
| FName | cambia el filename del fichero que se esta editando |
| FType | cambia el filetype del fichero que se esta editando |
| HEX | permite que algunos comandos admitan operandos hexadecimales |
| IMPcmscp | decide si los comandos no reconocidos seran interpretados como comandos de CP o CMS. |
| LINEND | selecciona un caracter como final logico de linea |
| LRecl | define la longitud logica (maxima) del fichero que se esta editando |
| NONDisp | define el caracter que se mostrara por pantalla si en el fichero se encuentra un caracter extraño |
| NULIs | decide si el final de las lineas contendra blancos o caracteres nulos |
| NUMBER | decide si se desea que la numeracion de las lineas aparezca en el area de prefijo o no |
| PACK | empaqueta un fichero al escribirlo en disco |
| PFn | asigna significado a las teclas PFn |
| PREFix | decide la existencia y posicion del area de prefijo |
| RECFm | permite cambiar el tipo de registro (fijo o variable, empaquetado o no) |
| SCALE | permite suprimir o decidir la posicion de la escala |
| SCREen | permite configurar la pantalla |
| SERIAL | permite controlar la numeracion automatica de las lineas |
| TABS | cambia las tabulaciones |
| TRunc | decide cual es la columna de truncacion |



| | |
|----------|---|
| VARblank | permite que el numero de blancos no sea relevante en un subcomando LOCATE o similares |
| Verify | decide que columnas se visualizan, y si se muestran por pantalla las lineas alteradas |
| Zone | decide que columnas intervendran en comandos como LOCATE o similares |

Subcomandos utilizables desde el area de prefijo:

| | |
|----------------|--|
| A (add) | añade lineas |
| C (copy) | copia lineas |
| D (delete) | suprime lineas |
| E (extend) | extiende una linea al editar ficheros con longitudes de registro grandes |
| F (following) | marca la linea DESPUES DE la cual se copiaran o moveran otras lineas |
| I (insert) | añade lineas |
| M (move) | mueve lineas |
| P (preceding) | marca la linea ANTES DE la cual se copiaran o moveran otras lineas |
| SCALE | decide la posicion de la linea de escala |
| TABL (tabline) | decide la posicion de la linea de tabulacion |
| * | duplica lineas |
| . | da un nombre simbolico a una linea |
| / | cambia la linea actual |

Significado inicial de las teclas FFn:



| Tecla | Efecto |
|-------|--|
| FP1 | HELP XEDIT MENU Lista de los comandos del Editor |
| FP2 | añade una línea A CONTINUACION de aquella sobre la que se encuentra el cursor |
| FP3 | 'sale' del Editor si no hay alteraciones en el fichero |
| FP4 | es la tecla de tabulación |
| FP5 | permite hacer 'cambios selectivos' (hagase 'HELP XEDIT SCHANGE') |
| FP6 | muestra en la línea de subcomandos el último subcomando introducido |
| FP7 | 'pasa página' hacia atrás |
| FP8 | 'pasa página' hacia adelante |
| FP9 | vuelve a ejecutar el último comando |
| FP10 | parte la línea sobre la que está el cursor en dos líneas separadas, a partir de la posición del cursor |
| FP11 | concatena la línea siguiente a la que contiene el cursor a continuación de la posición del cursor |
| FP12 | coloca el cursor en la posición de la 'current line' identificada por el símbolo ' ' de la línea de escala |



APENDICE C: SESIONES DE EJEMPLO

=====

Se incluyen a continuacion varias sesiones de ejemplo, tomadas del listado de consola de una terminal. El texto que aparece en la parte izquierda de los listados corresponde a la salida por terminal. El texto que aparece en la parte derecha se a~nadio luego; su unica funcion es la de comentar la parte izquierda.

Dado que los listados de consola (obtenidos mediante el comando 'CP SPOOL CONSOLE START') no muestran las pantallas del editor ni del BROWSE, para poder visualizar el contenido de los ficheros se utiliza el comando 'TYPE' (citado en el Apendice A; vease [CMSCMR]), antigua version del BROWSE.

En el primer ejemplo, un programa sencillo (contar las apariciones de la letra 'A' en una frase acabada por un punto) sirve para demostrar todas las posibilidades de ejecucion de un programa: por pantalla, con o sin FILEDEFS a disco, mediante modulos y EXECs, interactivamente y en BATCH.

En el segundo ejemplo, se compila y ejecuta un programa sencillo en FORTRAN (calcular la media de varios valores reales) utilizando diversas posibilidades de entrada y salida de datos, asi como la opcion BATCH y otras facilidades.

El tercer ejemplo es un listado producido por el programa BMDF6D aplicado a los datos que aparecen en la figura 23.

Finalmente, el cuarto ejemplo muestra una sesion de trabajo producida al utilizar el paquete SCSS.



```

type contaas pascal
PROGRAM CONTAAS(INPUT,OUTPUT);
VAR N: INTEGER; C: CHAR;
BEGIN N:= 0; READ(C);
  WHILE C ( ) ' ' DO BEGIN
    IF (C = 'a') OR (C = 'A') THEN N := N + 1;
    READ(C)
  END;
  WRITELN(' EL NUMERO DE LETRAS 'A' QUE APARECEN EN ESTA FRASE ES: ',
    N:1);
END.

```

Vamos a ejecutar el siguiente programa PASCAL

```

R: T=0.01/0.04 12:47:49
pascal contaas
R: T=0.93/1.12 12:47:57
global txtlib paslib
R: T=0.01/0.02 12:48:05
run contaas
DMSLI0740I EXECUTION BEGINS...
Cuántas a's tiene esta frase antes de que aparezca el punto?
EL NUMERO DE LETRAS 'A' QUE APARECEN EN ESTA FRASE ES: 9
R: T=0.55/0.70 12:48:45
fi output disk
R: T=0.01/0.01 12:49:02
run contaas
DMSLI0740I EXECUTION BEGINS...
El número de a's que ahora se cuenta se graba en un fichero en disco
llamado FILE OUTPUT A - notese que, al faltar el punto, podemos
seguir escribiendo en diversas líneas.
R: T=0.57/0.74 12:50:07
type file output

```

Para ello, lo compilamos,

emitimos el comando GLOBAL adecuado (que servira durante toda la sesion) y introducimos el comando RUN. Esto indica que el programa empieza a ejecutarse. Esta es la respuesta del programa a la línea que hemos introducido. Probemos de ejecutar el mismo programa, pero escribiendo los resultados en un fichero en disco:

EL NUMERO DE LETRAS 'A' QUE APARECEN EN ESTA FRASE ES: 14

Observese que el resultado NO aparece por pantalla, sino que se graba en un fichero en disco:

```

R: T=0.01/0.02 12:50:16
x file input
R: T=0.22/0.36 12:51:33
type file input

```

Del mismo modo, si editamos un fichero que contiene

La palabra AAAAA no tiene sentido.

```

R: T=0.01/0.02 12:51:37
filedef input disk
R: T=0.01/0.01 12:51:58
run contaas
DMSLI0740I EXECUTION BEGINS...
EL NUMERO DE LETRAS 'A' QUE APARECEN EN ESTA FRASE ES: 9
R: T=0.55/0.69 12:52:03
filedef input disk
R: T=0.01/0.01 12:52:12
filedef output disk
R: T=0.01/0.01 12:52:17
run contaas
DMSLI0740I EXECUTION BEGINS...
R: T=0.55/0.71 12:52:26
type file output

```

y especificamos que a partir de ese fichero se lean los datos.

el programa se ejecuta y produce resultados leyendo los datos desde disco (no por pantalla). Combinando los dos ejemplos anteriores, definimos la entrada desde disco y la salida tambien.

Naturalmente, la ejecución se produce sin leer ni escribir nada en la consola, aunque los datos de salida se graban en el fichero en disco:

EL NUMERO DE LETRAS 'A' QUE APARECEN EN ESTA FRASE ES: 9

```

R: T=0.01/0.02 12:52:34
load contaas
R: T=0.00/0.14 12:52:58
genmod contaas
R: T=0.01/0.03 12:53:03
contaas
Cuántas a's hay aqui?
EL NUMERO DE LETRAS 'A' QUE APARECEN EN ESTA FRASE ES: 5
R: T=0.40/0.48 12:53:19
x contaas exec
R: T=0.17/0.28 12:54:35
type contaas exec

```

Si el programa va a ejecutarse muy a menudo, podemos generar un MODULO: a partir de ahora, solo con utilizar el nombre del programa se ejecuta este:

Si queremos que cada vez se compile y cargue el programa, podemos crear un fichero EXEC que contenga

```

PASCAL CONTAAS;
GLOBAL TXTLIB PASLIB
LOAD CONTAAS (START

```

Despues de lo cual el nombre del fichero EXEC equivale a la ejecución de los comandos que contiene:

```

R: T=0.01/0.02 12:54:41
contaas
PASCAL CDNTAAS
GLOBAL TXTLIB PASLIB
LOAD CDNTAAS ( START
DMSLI0740I EXECUTION BEGINS...
Este EXEC realiza todas las funciones automaticamente;
podriamos haber incluido aqui los FILEDEFS;
notese que habiendo un modulo y un EXEC se utiliza primero el EXEC.
EL NUMERO DE LETRAS 'A' QUE APARECEN EN ESTA FRASE ES: 12
R: T=1.08/1.41 13:00:54
x contaas batch
R: T=0.19/0.34 13:10:23
type contaas batch

```

Por ultimo, ejecutaremos este programa utilizando el BATCH. editamos un fichero que contiene:

Ejemplo 1 (parte 1 de 3): variaciones sobre un programa PASCAL.



PAS CONTAAS
GLOBAL TXLIB PASLIB
LOAD CONTAAS (START
Cuenta cuantas as hay.

R; T=0.01/0.02 13:10:28
batch contaas
DMSCPY601K ENTER SPECIFICATION LIST:

EMPIEZA A PREPARAR ENVIO
INTRO: 'ENVIO' PARA ENVIAR AL BATCH
PUN FILE 4215 TO BATCH COPY 001 NOHOLD
R; T=0.35/0.63 13:10:38
13:10:38

MSG FROM BATCH JOB ARRIBAT AL BATCH...
13:10:38

MSG FROM BATCH3 JOB 'ZDOCINI' STARTED.
13:10:51

MSG FROM BATCH3 JOB 'ZDOCINI' ENDED.

CON FILE 4217 FROM BATCH3 COPY 001 NOHOLD
RDR
0001 FILE CHANGED
PRINTER FILE (4217) ITEM LENGTH= 084;
ENTER NEW FILEID OR PURGE (NULL LINE TO EXIT)
batch console
READER IS EMPTY
R; T=0.10/0.21 13:11:09
type batch console

'19E' REPLACES ' Y (19E) '
Y (19E) R/D
P (319) R/D
X (31A) R/D
H (320) R/D
/SET TIME 300 PRINT 1290
EXEC BATCHINI ZDOCINI TO TO ZDOCINI T300
DEV 591 DOES NOT EXIST
R; T=0.14/0.37 13:10:43
FI O DISK CONTAAS PASCAL (LRECL 80 RECFM F BLOCK 80
R; T=0.01/0.01 13:10:43
FI I TERM (LRECL 80 RECFM F BLOCK 80
R; T=0.01/0.01 13:10:43
MOVE I O
TERM INPUT -- TYPE NULL LINE FOR END OF DATA.
R; T=0.03/0.08 13:10:43
PASCAL CONTAAS
R; T=1.47/1.82 13:10:48
GLOBAL TXLIB PASLIB
R; T=0.01/0.01 13:10:48
LOAD CONTAAS (START
EXECUTION BEGINS...
CUENTA CUANTAS AS HAY.
EL NUMERO DE LETRAS 'A' QUE APARECEN EN ESTA FRASE ES 5
R; T=0.99/1.20 13:10:50
EXEC BATCHFIN ZDOCINI ZDOCINI T300
DASD 591 DETACHED
R; T=0.02/0.03 13:10:51
/*
CMSBATCH ENDED.

R; T=0.04/0.11 13:11:15
x contaas batch
R; T=0.15/0.25 13:12:38
type contaas batch

PAS CONTAAS
GLOBAL TXLIB PASLIB
FI OUTPUT DISK
LOAD CONTAAS (START
Cuenta cuantas as hay.
DISK DUMP FILE OUTPUT
R; T=0.02/0.08 13:12:43
batch contaas
DMSCPY601R ENTER SPECIFICATION LIST:

EMPIEZA A PREPARAR ENVIO
INTRO: 'ENVIO' PARA ENVIAR AL BATCH
PUN FILE 4221 TO BATCH COPY 001 NOHOLD

A continuacion llamamos al BATCH.
tras responder 'INTRO' al panel que nos
presenta (y que no se ve en este listado).

respondemos 'e' (abreviacion de 'envio')
a esta pregunta.
Este mensaje lo da el sistema para indicar
que la transmision de informacion hacia la
maquina BATCH se ha efectuado.

Este mensaje es enviado por el BATCH al recibir
el trabajo.

este cuando empieza a procesarlo

y este al terminar el proceso.

Señala la llegada de informacion desde el BATCH.
El comando RDR nos permite leer comodamente
la informacion enviada por el BATCH.

Aqui nos pide un identificador de fichero
(fileid). Elegimos batch console.
pero podria ser otro.

Veamos que nos ha enviado el BATCH:

esto es el listado de consola. y lo que aparece
aqui es similar a la informacion irrelevante
que aparece al hacer LOGON. antes
de que se borre la pantalla y podamos
empezar a trabajar.

Aqui se compila nuestro programa.

El comando GLOBAL es necesario para la maquina BATCH.

o bien: RUN CONTAAS

Esta es la frase que contenia el fichero.
y esta la respuesta del programa.
De nuevo informacion irrelevante.

Por ultimo, modificaremos el fichero BATCH

Los resultados se graban ahora en el
disco A DE LA MAQUINA BATCH.

por lo que es necesario que los enviemos
para poder leerlos luego.

Igual que antes.

Ejemplo 1 (parte 2 de 3): variaciones sobre un programa PASCAL.



R, T=0.39/0.68 13:12:46
13:12:46

MSG FROM BATCH JOB ARRIBAT AL BATCH...
13:12:47

MSG FROM BATCH3 JOB 'ZDOCINI' STARTED.
13:13:03

MSG FROM BATCH3 JOB 'ZDOCINI' ENDED.

PUN FILE 4224 FROM BATCH3 COPY 001 NOHOLD
CON FILE 4223 FROM BATCH3 COPY 001 NOHOLD
TOT
0002 FILES CHANGED
-----) FILE OUTPUT (4224) ORIGIN: BATCH3
FILE OUTPUT A EXISTS, OVERLAY EXISTING COPY? (ENTER Y, N, NEW FILEID, OR PURGE)
new output
NEW OUTPUT A1
PRINTER FILE (4223) ITEM LENGTH= 084;
ENTER NEW FILEID OR PURGE (NULL LINE TO EXIT)
batch console2
READER IS EMPTY
R, T=0.16/0.34 13:13:24
type batch console2

Esta vez hemos recibido dos ficheros:
para el fichero FILE OUTPUT, nos avisa de que ya hay uno llamado así en nuestro disco A, por lo que hay que cambiarle el nombre. Elegimos 'new output'. Este es el listado de consola, al que llamaremos BATCH CONSOLE2 para distinguirlo del anterior BATCH CONSOLE.
Veamos que contiene:

1'19E' REPLACES ' Y (19E) '
Y (19E) R/O
P (319) R/O
X (31A) R/O
H (320) R/O
/SET TIME 300 PRINT 1280

EXEC BATCHINI ZDOCINI TQ TO ZDOCINI T300
DEV 591 DOES NOT EXIST
R, T=0.15/0.38 13:12:55
FI O DISK CUNTAAS PASCAL (LRECL 80 RECFM F BLOCK 80
R, T=0.01/0.01 13:12:55
FI I TERM (LRECL 80 RECFM F BLOCK 80
R, T=0.01/0.01 13:12:55
MOVE I O
TERM INPUT -- TYPE NULL LINE FOR END OF DATA.
R, T=0.03/0.08 13:12:56
PASCAL CONTAAS
R, T=1.44/1.79 13:13:00
GLOBAL TXTLIB PASTLIB
R, T=0.01/0.01 13:13:00
FI OUTPUT DISK
R, T=0.01/0.01 13:13:00
LOAD CONTAAS (START
EXECUTION BEGINS...
CUENTA CUANTAS AS HAY.
R, T=0.99/1.20 13:13:03
DISK DUMP FILE OUTPUT
R, T=0.01/0.02 13:13:03
EXEC BATCHFIN ZDOCINI ZDOCINI ZDOCINI T300
DASD 591 DETACHED
R, T=0.02/0.03 13:13:03
/*
CMSBATCH ENDED.
PUN FILE 4224 TO ZDOCINI COPY 001 NOHOLD

R, T=0.05/0.12 13:13:35
type new output

Nada interesante:
los resultados estan aqui:

EL NUMERO DE LETRAS 'A' QUE APARECEN EN ESTA FRASE ES: 5

R, T=0.01/0.01 13:13:52

Ejemplo 1 (parte 3 de 3): variaciones sobre un programa PASCAL.



R: T=0.01/0.01 19:06:22

Se desea ejecutar el siguiente programa escrito en lenguaje FORTRAN:

type media forttran a

```

      DIMENSION A(100)
      WRITE(6,100)
      READ(5,*) N
      WRITE(6,200) N
      READ(5,*)(A(I),I=1,N)
      DO 10 I=1,N
10    SM=SM+A(I)
      SM=SM/N
      WRITE(6,300) SM
100  FORMAT(' ESTE PROGRAMA CALCULA LA MEDIA DE VARIOS NUMEROS REALES'/
           ' CUAL ES EL NUMERO DE VALORES QUE SE VAN A INTRODUCIR?')
200  FORMAT(' INTRODUZCA 5 VALORES REALES SEPARADOS'/
           ' POR BLANCOS (SI NO TIENEN DECIMALES DEBE PONERSE'/
           ' EL PUNTO DECIMAL, P. EJ. 23.0 O 23.)')
300  FORMAT(' LA MEDIA DE LOS VALORES INTRODUCIDOS ES: ',F8.5)
      STOP
      END

```

```

MED00010
MED00020
MED00030
MED00040
MED00050
MED00060
MED00070
MED00080
MED00090
MED00100
MED00110
MED00120
MED00130
MED00140
MED00150
MED00160
MED00170
MED00180

```

R: T=0.03/0.09 19:07:06

Primero se compila el programa.

```

fortran media
G1 COMPILER ENTERED
SOURCE ANALYZED
PROGRAM NAME = MAIN
* NO DIAGNOSTICS GENERATED
R: T=0.33/0.46 19:07:51
global txtlib fortmod2 mod2sch
R: T=0.02/0.02 19:08:11

```

Contesta confirmando que no se ha producido ningun error de compilacion. Se le indica en que librerias estan los modulos de FORTRAN. Este comando GLOBAL servira para toda la sesion hasta que se haya otro GLOBAL.

```

load media(start
DMSLI0740I EXECUTION BEGINS...
ESTE PROGRAMA CALCULA LA MEDIA DE VARIOS NUMEROS REALES
CUAL ES EL NUMERO DE VALORES QUE SE VAN A INTRODUCIR?
?
5
INTRODUZCA 5 VALORES REALES SEPARADOS
POR BLANCOS (SI NO TIENEN DECIMALES DEBE PONERSE
EL PUNTO DECIMAL, P. EJ. 23.0 O 23.)
?
18.9 45.67 33.78 46.9 54.6
LA MEDIA DE LOS VALORES INTRODUCIDOS ES:
39.96799
R: T=0.23/0.44 19:08:52
redit media datos a
R: T=0.20/0.32 19:11:18
type media datos a
9
23.4 33.6 55.7 34.4 33.5 56.8 56.9 98.7 12.4
R: T=0.01/0.02 19:11:37
filedef 5 disk media datos a(recf= f trecl 80 per=
R: T=0.01/0.01 19:12:07
filedef 6 disk media resul a(recf= f trecl 80 per=
R: T=0.01/0.01 19:12:49
load media(start
DMSLI0740I EXECUTION BEGINS...
R: T=0.22/0.39 19:13:00
type media resul a

```

Se carga el programa y se inicia su ejecucion.

El programa pide que se le introduzca por terminal el dato pedido.

Id.

Esta es la respuesta y resultado.

Se edita a continuacion un fichero con datos para el programa.

El fichero contiene los siguientes datos:

Se le indica al sistema que cuando el programa quiera leer datos de la unidad 5, los tomara del fichero especificado. Id. para la unidad 6.

Se vuelve a ejecutar el programa.

Acaba la ejecucion y los mensajes y resultado se han grabado en el fichero media resul a.

```

ESTE PROGRAMA CALCULA LA MEDIA DE VARIOS NUMEROS REALES
CUAL ES EL NUMERO DE VALORES QUE SE VAN A INTRODUCIR?
INTRODUZCA 9 VALORES REALES SEPARADOS
POR BLANCOS (SI NO TIENEN DECIMALES DEBE PONERSE
EL PUNTO DECIMAL, P. EJ. 23.0 O 23.)
LA MEDIA DE LOS VALORES INTRODUCIDOS ES:
45.04440
R: T=0.01/0.04 19:13:38
load media
R: T=0.19/0.32 19:14:37
genmod media
R: T=0.03/0.08 19:14:52
4 filedef
FT05F001 DISK MEDIA DATOS
FT06F001 DISK MEDIA RESULT
R: T=0.02/0.02 19:14:59
media
R: T=0.04/0.09 19:15:05
filedef 5 term
R: T=0.01/0.01 19:15:31

```

Si el programa va a ejecutarse a menudo conviene generar un MODULO, cuyo nombre puede utilizarse directamente para ejecutar el programa tantas veces como se quiera. Antes de ejecutar de nuevo el programa, se comprueba si los filedef's siguen en efecto:

Al introducir el nombre del modulo generado se ejecuta de nuevo el programa produciendo el mismo resultado que en el caso anterior.

Se puede volver a entrar y obtener datos por terminal mediante nuevos filedef's:

Ejemplo 2 (parte 1 de 5): variaciones sobre un programa FORTRAN.




```
filedef 6 term
R; T=0.01/0.01 19:15:37
media
```

ESTE PROGRAMA CALCULA LA MEDIA DE VARIOS NUMEROS REALES CUAL ES EL NUMERO DE VALORES QUE SE VAN A INTRODUCIR?

```
?
2
INTRODUZCA 2 VALORES REALES SEPARADOS
POR BLANCOS (SI NO TIENEN DECIMALES DEBE PONERSE
EL PUNTO DECIMAL, P. EJ. 23.0 O 23.)
?
```

```
728.9 345.6
LA MEDIA DE LOS VALORES INTRODUCIDOS ES:
*****
```

```
R; T=0.03/0.11 19:15:51
x media fortran a
```

```
R; T=0.18/0.26 19:16:50
fortn media
```

```
G1 COMPILER ENTERED
SOURCE ANALYZED
PROGRAM NAME = MAIN
* NO DIAGNOSTICS GENERATED
```

```
R; T=0.28/0.42 19:17:00
load media
R; T=0.19/0.33 19:17:06
genmod media
R; T=0.03/0.08 19:17:16
```

```
media
ESTE PROGRAMA CALCULA LA MEDIA DE VARIOS NUMEROS REALES
CUAL ES EL NUMERO DE VALORES QUE SE VAN A INTRODUCIR?
```

```
?
2
INTRODUZCA 2 VALORES REALES SEPARADOS
POR BLANCOS (SI NO TIENEN DECIMALES DEBE PONERSE
EL PUNTO DECIMAL, P. EJ. 23.0 O 23.)
?
```

```
728.9 345.6
LA MEDIA DE LOS VALORES INTRODUCIDOS ES:
537.25960
R; T=0.03/0.11 19:17:35
```

```
x mediaex exec a
R; T=0.17/0.26 19:46:25
type mediaex exec a
```

```
FILEDEF 5 DISK A1 DATOS A(RECFM F LRECL 80 PERM
MEDIA
FILEDEF 6 DISK A1 RESULT A(RECFM F LRECL 80 PERM
MEDIA
```

```
R; T=0.01/0.03 19:46:33
mediaex media
FILEDEF 5 DISK MEDIA DATOS A ( RECFM F LRECL 80 PERM
FILEDEF 6 DISK MEDIA RESULT A ( RECFM F LRECL 80 PERM
MEDIA
R; T=0.06/0.14 19:46:44
```

```
x med2 datos a
R; T=0.23/0.36 19:47:46
type med2 datos a
```

```
5
34.6 57.8 79.8 56.4 34.7
```

```
R; T=0.01/0.02 19:47:55
mediaex med2
FILEDEF 5 DISK MED2 DATOS A ( RECFM F LRECL 80 PERM
FILEDEF 6 DISK MED2 RESULT A ( RECFM F LRECL 80 PERM
MEDIA
R; T=0.05/0.12 19:48:01
type med2 result a
```

```
ESTE PROGRAMA CALCULA LA MEDIA DE VARIOS NUMEROS REALES
CUAL ES EL NUMERO DE VALORES QUE SE VAN A INTRODUCIR?
INTRODUZCA 5 VALORES REALES SEPARADOS
POR BLANCOS (SI NO TIENEN DECIMALES DEBE PONERSE
EL PUNTO DECIMAL, P. EJ. 23.0 O 23.)
LA MEDIA DE LOS VALORES INTRODUCIDOS ES:
```

52.65996

```
R; T=0.02/0.04 19:48:32
```

```
x media batch a
R; T=0.19/0.30 19:57:52
type media batch a
```

FOR MEDIA

Se vuelve a vincular el programa, esta vez de nuevo a través de terminal.

El resultado obtenido ocupa mas posiciones de las especificadas en el correspondiente FORMAT. Para corregir este problema se edita de nuevo el programa. El comando XEDIT se abrevia con X.

Una vez corregido, debe volverse a compilar y generar el correspondiente modulo:

Ya puede ejecutarse de nuevo.

Esta vez el resultado puede escribirse.

Para no tener que repetir cada vez los comandos filedef's y poder asignar como entrada y salida diferentes ficheros, se puede elaborar un fichero EXEC:

Al introducir el nombre del fichero EXEC anterior se ejecutan los comandos que contiene y el parametro MEDIA que se ha teclado a su lado pasa a sustituir el termino 'a' que aparecia en el fichero EXEC. A continuacion se ejecuta el programa con los datos contenidos en el fichero media datos a y se colocan el resultado en el fichero media resul a.

Se edita ahora un nuevo fichero de datos:

Al ejecutar ahora el EXEC mediaex con el parametro med2, se asignan las unidades 5 y 6 a los nuevos ficheros.

El resultado obtenido es:

Por ultimo ejecutaremos el programa mediante la facilidad BATCH, para lo que se edita el fichero:

Ejemplo 2 (parte 2 de 5): variaciones sobre un programa FORTRAN.



GLOBAL TXLIB FORTMOD2 MOD2EEH
LOAD MEDIA(START)
3
34.5 56.7 77.8

R: T=0.02/0.03 20:00:40
batch media
DMSCF601R ENTER SPECIFICATION LIST.

EMPIEZA A PREPARAR ENVIO
INTRO. 'ENVIO' PARA ENVIAR AL BATCH
envio
PUN FILE 7534 TO BATCH COPY 001 NOHOLD
R: T=0.45/0.74 20:01:17
20:01:17

MSG FROM BATCH : JOB ARRIBAT AL BATCH...
20:08:57

MSG FROM BATCH3 : JOB 'ZDOCINI' STARTED.
20:09:18

MSG FROM BATCH3 : JOB 'ZDOCINI' ENDED.

CON FILE 7553 FROM BATCH3 COPY 001 NOHOLD
q reader all
ORIGINID FILE CLASS RECORDS CPY HOLD DATE TIME NAME TYPE DIST
BATCH3 7553 T CON 00000043 001 NONE 04/20 20:08:57 BATCH CONSOLE ZDOCINI
R: T=0.02/0.03 21:08:09
rdy
0001 FILE CHANGED
PRINTER FILE (7553) ITEM LENGTH= 084;
ENTER NEW FILEID OR PURGE (NULL LINE TO EXIT)
media consola a
READER IS EMPTY
R: T=0.12/0.25 21:08:21
type media consola a

'19E' REPLACES ' Y (19E) '
Y (19E) R/O
P (319) R/O
X (31A) R/O
H (320) R/O
/SET TIME 300 PRINT 1280
EXEC BATCHINI ZDOCINI TO TO ZDOCINI T300
DEV 591 DOES NOT EXIST
R: T=0.15/0.38 20:09:09
FI O DISK MEDIA FORTRAN (LRECL 80 RECFM F BLOCK 80
R: T=0.01/0.01 20:09:10
FI I TERM (LRECL 80 RECFM F BLOCK 80
R: T=0.01/0.01 20:09:10
MOVE I O
TERM INPUT -- TYPE NULL LINE FOR END OF DATA.
R: T=0.04/0.10 20:09:10
FORTGI MEDIA
G1 COMPILER ENTERED
SOURCE ANALYZED
PROGRAM NAME = MAIN
* NO DIAGNOSTICS GENERATED
R: T=0.27/0.41 20:09:15
GLOBAL TXLIB FORTMOD2 MOD2EEH
R: T=0.02/0.02 20:09:15
LOAD MEDIA (START
EXECUTION BEGINS...
ESTE PROGRAMA CALCULA LA MEDIA DE VARIOS NUMEROS REALES
CUAL ES EL NUMERO DE VALORES QUE SE VAN A INTRODUCIR?
?
3
INTRODUZCA 3 VALORES REALES SEPARADOS
POR BLANCOS (SI NO TIENEN DECIMALES DEBE PONERSE
EL PUNTO DECIMAL, P. EJ. 23.0 O 23.)
?
34.5 56.7 77.8
LA MEDIA DE LOS VALORES INTRODUCIDOS ES:
56.33333
R: T=0.22/0.45 20:09:18
EXEC BATCHFIN ZDOCINI ZDOCINI ZDOCINI T300
DAJD 591 DETACHED
R: T=0.03/0.05 20:09:18
/*
CMSBATCH ENDED.

R: T=0.06/0.18 21:08:37

x mediab2 batch
R: T=0.21/0.37 21:17:23
type mediab2 batch

Observese que los datos se colocan en la posicion en que debe encontrarlos la maquina BATCH cuando intente ejecutar el programa. Para enviarlo a la correspondiente maquina BATCH se introduce este comando y se obtiene el panel de especificaciones que aparece en la pag. 59.

Pide la confirmacion del envio mediante la introduccion de la palabra ENVIO.

Este mensaje nos informa de que se le ha enviado el trabajo a la correspondiente maquina BATCH.

A continuacion llega un mensaje del BATCH conforme el trabajo ha llegado a la cola.

Al cabo de un rato avisa de que el trabajo ha empezado a ejecutarse (mientras se puede seguir trabajando en la propia terminal).

Finalmente avisa de que ha finalizado el trabajo.

Se recibe un fichero del BATCH en la lectora. Se puede visualizar mejor:

Se lee el fichero:

Se colocara con esa identificacion en el disco A.

El fichero contiene el listado de consola con las operaciones realizadas por la maquina BATCH

Aqui la maquina BATCH inicio la compilacion del programa.

Aqui inicio la ejecucion tomando los datos.

Aqui escribio el resultado.

Este es el final del listado de consola de las operaciones de la maquina BATCH.

Se edita un nuevo fichero BATCH para que los datos y resultados se obtengan y coloquen en ficheros en disco. Y se ejecute el programa a partir del modulo que esta en el disco D accedido automaticamente por la maquina BATCH.

Ejemplo 2 (parte 3 de 5): variaciones sobre un programa FORTRAN.



```
FOR MEDIA
DATA MEDIARECFM F LRECL 80 BLOCK 80
GLOBAL TXLIB FORTMOD2 MOD2EEH
FILEDEF 5 DISK MEDIA DATA A( RECFM F LRECL 80
FILEDEF 6 DISK MEDIA RESULT A( RECFM F LRECL 121
MEDIA(START
DISK DUMP MEDIA RESULT A
```

Así se le envía el fichero MEDIA DATA A.

Se ejecutará el módulo que está en el disco D. Así se recibirá el fichero de resultados en la lectora.

```
R: T=0.01/0.03 21:17:42
type media data a
```

Los datos enviados en el fichero data son:

```
4
3.2 4.5 6.7 8.9
```

```
R: T=0.01/0.01 21:17:48
```

Se vuelve a enviar al BATCH.

```
batch mediab2
DMSCPY601R ENTER SPECIFICATION LIST:
```

EMPIEZA A PREPARAR ENVIO

```
*
INTRO: 'ENVIO' PARA ENVIAR AL BATCH
PUN FILE 1549 TO BATCH COPY 001 NOHOLD
R: T=0.56/0.96 21:28:07
21:28:07
```

```
MSG FROM BATCH : JOB ARRIBAT AL BATCH...
```

21:30:09

```
MSG FROM BATCH3 : JOB 'ZDOCINI' STARTED.
```

21:30:25

```
MSG FROM BATCH3 : JOB 'ZDOCINI' ENDED.
```

En este caso la maquina BATCH envía dos ficheros: el resultado y el listado de consola.

```
PUN FILE 1549 FROM BATCH3 COPY 001 NOHOLD
CON FILE 1547 FROM BATCH3 COPY 001 NOHOLD
q reader jll
ORIGINID FILE CLASS RECORDS CPY HOLD DATE TIME NAME TYPE DIST
BATCH3 1549 A PUN 00000013 001 NONE 04/20 21:30:25 CMSBATCH ZDOCINI ZDOCINI
BATCH3 1547 T CON 00000032 001 NONE 04/20 21:30:14 BATCH CONSOLE ZDOCINI
R: T=0.01/0.04 21:31:07
```

```
rdr
0002 FILES CHANGED
----> MEDIA RESULT ( 1549 ) ORIGIN: BATCH3
LOAD THE FILE? ( ENTER Y OR N OR NEW FILEID OR PURGE )
media result j
MEDIA RESULT A1
PRINTER FILE ( 1547 ) ITEM LENGTH= 084
ENTER NEW FILEID OR PURGE ( NULL LINE TO EXIT )
media consola a
READER IS EMPTY
R: T=0.17/0.35 21:32:17
flist
R: T=0.07/0.19 21:33:30
type media consola a
```

El listado de consola del BATCH contiene:

```
1'19E' REPLACES ' Y (19E) '
Y (19E) R/O
P (319) R/O
X (31A) R/O
H (320) R/O
/SET TIME 300 PRINT 1200
EXEC BATCHINI ZDOCINI TO TO ZDOCINI T300
DEV 591 DOES NOT EXIST
R: T=0.15/0.37 21:30:23
FI O DISK MEDIA DATA ( RECFM F LRECL 80 BLOCK 80
R: T=0.01/0.01 21:30:23
FI I TERM ( RECFM F LRECL 80 BLOCK 80
R: T=0.01/0.01 21:30:23
```

```
MOVE I 0
TERM INPUT -- TYPE NULL LINE FOR END OF DATA.
R: T=0.02/0.06 21:30:24
GLOBAL TXLIB FORTMOD2 MOD2EEH
R: T=0.01/0.02 21:30:24
FILEDEF 5 DISK MEDIA DATA A ( RECFM F LRECL 80 BLOCK 80
R: T=0.01/0.01 21:30:24
FILEDEF 6 DISK MEDIA RESULT A ( RECFM F LRECL 121 BLOCK 80
R: T=0.01/0.01 21:30:24
MEDIA
R: T=0.03/0.12 21:30:25
DISK DUMP MEDIA RESULT A
R: T=0.01/0.03 21:30:25
EXEC BATCHFIN ZDOCINI ZDOCINI ZDOCINI T300
DASD 591 DETACHED
R: T=0.02/0.03 21:30:25
/*
```

Ejecuta el módulo MEDIA que encuentra en el disco D del usuario al que accede al empezar. Envía el resultado que había grabado en su disco A.

Ejemplo 2 (parte 4 de 5): variaciones sobre un programa FORTRAN.



CMSBATCH ENDED.
FUN FILE 1547 TO ZDOCIN1 COPY 001 NOHOLD

R; T=0.04/0.12 21:34:01
type media result a

El fichero de resultados contiene:

ESTE PROGRAMA CALCULA LA MEDIA DE VARIOS NUMEROS REALES
CUAL ES EL NUMERO DE VALORES QUE SE VAN A INTRODUCIR?
INTRODUZCA 4 VALORES REALES SEPARADOS
POR BLANCOS (SI NO TIENEN DECIMALES DEBE PonerSE
EL PUNTO DECIMAL, P. EJ. 23.0 O 23.)
LA MEDIA DE LOS VALORES INTRODUCIDOS ES:
5.82500

R; T=0.01/0.03 21:34:07
spool console stop
R; T=0.02/0.04 21:34:11
spool console close
R; T=0.02/0.03 21:34:14
logoff

Se para y cierra el listado de consola del
usuario.

Se finaliza la sesion de trabajo.

Ejemplo 2 (parte 5 de 5): variaciones sobre un programa FORTRAN.



PAGE 1

BMP60 - BIVARIATE (SCATTER) PLOTS
 DEPARTMENT OF BIOMATHEMATICS
 UNIVERSITY OF CALIFORNIA, LOS ANGELES, CA 90024
 (213) 825-5940 TWX UCLA LSA
 PROGRAM REVISED JUNE 1981
 MANUAL REVISED -- 1981
 COPYRIGHT (C) 1981 REGENTS OF UNIVERSITY OF CALIFORNIA
 APRIL 26, 1983 AT 13:27:19

TO SEE REMARKS AND A SUMMARY OF NEW FEATURES FOR
 THIS PROGRAM, STATE NEWS, IN THE PRINT PARAGRAPH.

PROGRAM CONTROL INFORMATION

```

/BMP60
/PROBLEM TITLE IS 'ESTUDIO ALTURAS ALUMNOS-PADRES-PAREJAS. REGRESIONES'.
/INPUT VARIABLES ARE 14.
  FORMAT = '(IX,A4,F2.0,2F3.0,F6.1,F5.1,2(F3.0,F6.1),2F3.0,F6.1,F2.0)'.
/VARIABLE NAMES ARE ID,SEXO,PROV,EDAD,ALTURA,PESO,
  'PROV.P','ALTURA.P','PROV.M','ALTURA.M',
  'PROV.PAR','EDAD.PAR','ALT.PAR',
  ELECCION.
  LABEL IS ID.
/PLOT XVAR = ALTURA.
  YVAR = 'ALTURA.P','ALTURA.M'.
  COMMON.
  STATISTICS.
/END
  
```

PROBLEM TITLE IS
 ESTUDIO ALTURAS ALUMNOS-PADRES-PAREJAS. REGRESIONES

```

NUMBER OF VARIABLES TO READ IN. . . . . 14
NUMBER OF VARIABLES ADDED BY TRANSFORMATIONS. . . . . 0
TOTAL NUMBER OF VARIABLES . . . . . 14
NUMBER OF CASES TO READ IN. . . . . TO END
CASE LABELING VARIABLES . . . . . ID
MISSING VALUES CHECKED BEFORE OR AFTER TRANS. . . . . NEITHER
BLANKS ARE. . . . . MISSING
INPUT UNIT NUMBER . . . . . 5
REWIND INPUT UNIT PRIOR TO READING. . . . . NO
NUMBER OF WORDS OF DYNAMIC STORAGE. . . . . 113144
NUMBER OF CASES DESCRIBED BY INPUT FORMAT . . . . . 1
  
```

VARIABLES TO BE USED

| | | | | |
|-------------|------------|-------------|-------------|-------------|
| 2 SEXO | 3 PROV | 4 EDAD | 5 ALTURA | 6 PESO |
| 7 PROV.P | 9 ALTURA.P | 9 PROV.M | 10 ALTURA.M | 11 PROV.PAR |
| 12 EDAD.PAR | 13 ALT.PAR | 14 ELECCION | | |

INPUT FORMAT IS
 (IX,A4,F2.0,2F3.0,F6.1,F5.1,2(F3.0,F6.1),2F3.0,F6.1,F2.0)

MAXIMUM LENGTH DATA RECORD IS 56 CHARACTERS.

| INPUT VARIABLES | | | | | VARIABLE | | | | | | |
|-----------------|--------|------------|---------------|-----------|----------|-------|----------|------------|---------------|-----------|-------|
| VARIABLE INDEX | NAME | RECORD NO. | COLUMNS BEGIN | FIELD END | TYPE | INDEX | NAME | RECORD NO. | COLUMNS BEGIN | FIELD END | TYPE |
| 1 | ID | 1 | 2 | 5 | A | 8 | ALTURA.P | 1 | 28 | 33 | 5.1 F |
| 2 | SEXO | 1 | 6 | 7 | F | 9 | PROV.M | 1 | 34 | 36 | 3 F |
| 3 | PROV | 1 | 8 | 10 | F | 10 | ALTURA.M | 1 | 37 | 42 | 5.1 F |
| 4 | EDAD | 1 | 11 | 13 | F | 11 | PROV.PAR | 1 | 43 | 45 | 3 F |
| 5 | ALTURA | 1 | 14 | 19 | 6.1 F | 12 | EDAD.PAR | 1 | 46 | 48 | 3 F |
| 6 | PESO | 1 | 20 | 24 | 5.1 F | 13 | ALT.PAR | 1 | 49 | 54 | 6.1 F |
| 7 | PROV.P | 1 | 25 | 27 | 3 F | 14 | ELECCION | 1 | 55 | 56 | 2 F |

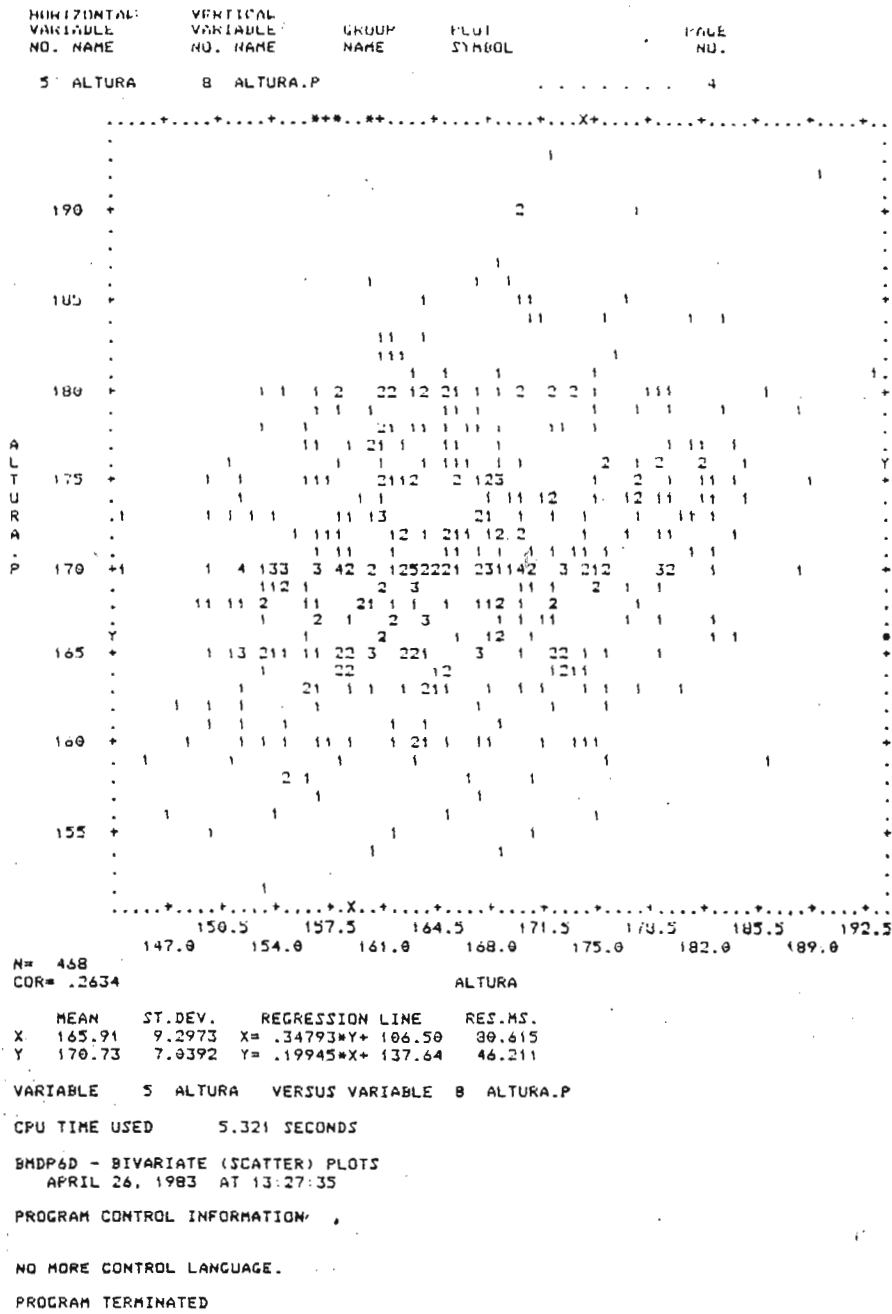
BASED ON INPUT FORMAT SUPPLIED 1 RECORDS READ PER CASE.

NUMBER OF CASES READ. 473

TABLE OF CONTENTS

Ejemplo 3 (parte 1 de 2): resultado de ejecución de programa BMPD.





Ejemplo 3 (parte 2 de 2): resultado de ejecución de programa BMDP.



R. 1 0.01/0.01 11 01 40
scss start

fin de sesion SCSS empieza con este comando.

SCSS RELEASE 94.2

EXPLAIN? (Enter YES or NO)
no
STYLE OF PROMPTING?

Pregunta si se desea una explicacion general sobre el programa.
Pregunta si deseamos preguntas ampliadas (verbose), normales (normal) o abreviadas (terse).

verbose
WORKFILE NAME? (NONE* or filename)
/logfile both,sesion

Pregunta el nombre del fichero de trabajo si existe, pero antes se define un fichero donde se guardara todo lo que se haga en la sesion.

NEW FILE ALLOCATED
LOGFILE OK
WORKFILE NAME? (NONE* or filename)
none
NEW WORKFILE NAME? (Filename)
trabajo
NEW FILE ALLOCATED
MASTERFILE NAME? (NONE or filename)
alturas master d

Vuelve a preguntar por el fichero de trabajo.

Se le ha dicho que no hay y se da un nombre:

PROCEDURE? (Procedure name followed by varlist)
?

Ahora pregunta por el nombre del fichero master y se le da el nombre de un fichero master que se habia preparado en una sesion previa, con datos sobre alturas, etc.
Pregunta cual de los procedimientos se desea y se le pide ayuda sobre las posibilidades que hay:

Enter procedure name optionally followed by variable list.
Current procedures are: UNIVARIATE, CROSSTABS, BREAKDOWN, TTEST, SCATTERPLOT, CORRELATIONS, PARTIALS, REGRESSION, FACTOR, and CASelist. RESIDUALS is entered via REGRESSION.
If you name PARTIALS, REGRESSION, or FACTOR, the first series of prompts is issued by the CORRELATIONS procedure.

PROCEDURE? (Procedure name followed by varlist)
/explain univariate

Vuelve a preguntar sobre un procedure y se le pide mas informacion sobre UNIVARIATE.

UNIVARIATE (Ch. 12) displays frequency distributions, measures of central tendency and dispersion, simple and cumulative histograms in raw and standardized form, and n-tile values.

UNIVARIATE prompts CELLS? and STATISTICS? for display profiles, and DISPLAY? to display output. /TABLE selects a subset of active variables for display, /HISTOGRAM produces histograms, /NTILES displays quantile cutting points, and /PERCENTILE displays percentile cutting points.

PROCEDURE? (Procedure name followed by varlist)
univariate
VARIABLE LIST? (Varlist)
/list var

Se pide entrar en UNIVARIATE.
Pregunta que variables queremos estudiar.
Se le pide una lista de las variables contenidas en el fichero:

ALPHABETICAL LIST OF VARIABLES ON WORKFILE

ALTURA ALTURAM ALTURAP EDAD PESO PROVINCI SEXO SPSS
SPSS0001

VARIABLE LIST? (Varlist)
altura alturap

Se eligen las siguientes variables:

CELLS? (ALL* or VALUE,N,TOTPCT,NMPCT,CUMPCT,LABELS or NONE)

Pregunta diversas caracteristicas del tratamiento de datos en este procedure:

all
STATISTICS? (MEAN*, STDDEV* or list of statistics)
all
DISPLAY? (STATS, CELLS, BOTH and/or item list)
both

Proporciona los resultados pedidos:

ALTURA ALTURA DEL ESTUDIANTE

MEAN = 165.496
S E MEAN = .552
MODE = NA
MEDIAN = 165.066
VARIANCE = 144.110
STD DEV = 12.005
SKEWNESS = -5.415
S E SKEW = .112
KURTOSIS = 75.399
S E KURT = .224
MINIMUM = .000
MAXIMUM = 193.000
RANGE = 193.000
SUM = 78279.375

TOTAL N = 473 VALID N = 473

ALTURAP ALTURA DEL PADRE DEL ESTUDIANTE

MEAN = 169.277
S E MEAN = .788
MODE = NA

Ejemplo 4 (parte 1 de 3): sesion de trabajo con SCSS.



```

MEAN = 173.146
VARIANCE = 73.741
STD DEV = 8.588
SKEWNESS = -8.138
S E SKEW = .112
KURTOSIS = 78.618
S E KURT = .224
MINIMUM = .000
MAXIMUM = 193.000
RANGE = 193.000
SUM = 80667.938

```

TOTAL N = 473 VALID N = 473

```

DISPLAY? (STATS, CELLS, BOTH and/or item list)
/list vocabulary

```

Pide nuevos parametros de tratamiento y para conocer todas las posibilidades del proceso se le pide una lista de palabras clave en este contexto

LOCAL VOCABULARY

CELLS? DISPLAY? HIST? NTILES? PERCENT? STAT? TABLE?

```

DISPLAY? (STATS, CELLS, BOTH and/or item list)
/hist?

```

Se le pide mas informacion sobre /HIST

```

SIMPLE* or CUMULATIVE, RAW* or STANDARDIZED, NORMAL* or
NONNORMAL, COLLAPSED* or EXPANDED, INCREASING* or DECREASING,
(low, high, increment)
STANDARDIZED overrides any value specification.
CUMULATIVE overrides DECREASING.
A specified increment overrides COLLAPSED or EXPANDED.
To request a histogram of 10 values with no normal plot:
/histogram (1,10,1) nonnormal

```

```

DISPLAY? (STATS, CELLS, BOTH and/or item list)
/hist simple

```

En cualquier momento se puede acceder a otra posibilidad del procedure mediante instrucciones iniciadas con /. Ejecuta la instruccion /HIST SIMPLE:

```

ALTURA  ALTURA DEL ESTUDIANTE
PCT      N  LOW BND (* = 3 CASES      . = NORMAL)
.0       0  101.58
.0       0  111.74
.0       0  121.89
.0       0  132.95
7.2     34  142.21 *****
32.3    153 152.37 *****
36.8    174 162.53 *****
19.7     93  172.68 *****
3.8     18  182.94 *****
MAXIMUM VALUE ERROR = .000
TOTAL N = 473  VALID N = 473

```

```

ALTURAP  ALTURA DEL PADRE DEL ESTUDIANTE
PCT      N  LOW BND (* = 4 CASES      . = NORMAL)
.0       0  101.58
.0       0  111.74
.0       0  121.89
.0       0  132.95
.2       1  142.21 *****
11.4     54 152.37 *****
49.5    234 162.53 *****
33.6    159 172.68 *****
4.4      21 182.84 *****
MAXIMUM VALUE ERROR = .000
TOTAL N = 473  VALID N = 473

```

```

DISPLAY? (STATS, CELLS, BOTH and/or item list)
/procedure scatter
VARIABLE LIST? (Varlist WITH variable BY variable)
/list var

```

En cualquier momento puede entrarse en un nuevo procedure simplemente invocandolo asi:

ALPHABETICAL LIST OF VARIABLES ON WORKFILE

```

ALTURA  ALTURAM  ALTURAP  EDAD  PESO  PROVINCI  SEXO  SPSS
SPSS0001

```

```

VARIABLE LIST? (Varlist WITH variable BY variable)
altura with edad
LIMITS? (ALL* or varlist = min, max)
SIZE? (NORMAL* or DOUBLE)

```

Se contesta con INTRO las opciones estandar de cada pregunta que son las señaladas por *:

```

DISPLAY? (PLOT, STAT, BOTH, or BLOWUP)
both

```

Se obtiene un grafico que puede mejorarse:

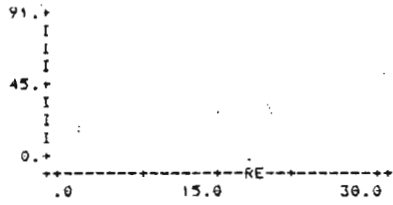
```

DOWN-ALTURA'A'  ACROSS-EDAD'E'
I                RE                SYMBOLS
193.+          ...                . = 1 - 29
I              ...                . = 30 - 57
I              **..... RA        * = 58 - 86
RA
I

```

Ejemplo 4 (parte 2 de 3): sesión de trabajo con SCSS.





| | MEAN | SD | R | RSQ | SIGF | SLOPE | INTCP | SEE | N |
|--------|--------|-------|------|------|------|--------|--------|-------|-----|
| ALTURA | 165.50 | 12.00 | .026 | .001 | .287 | .20 | 161.91 | 12.01 | 473 |
| EDAD | 18.86 | 1.59 | | | | 3.4-03 | 18.29 | 1.59 | 473 |

```

DISPLAY? (PLOT, STAT, BOTH, or BLOWUP)
/procedure scatter
VARIABLE LIST? (Varlist WITH variable BY variable)
altura with edad
LIMITS? (ALL* or varlist = min, max)
altura = 140.,195.
LIMITS? (ALL* or varlist = min, max)
    
```

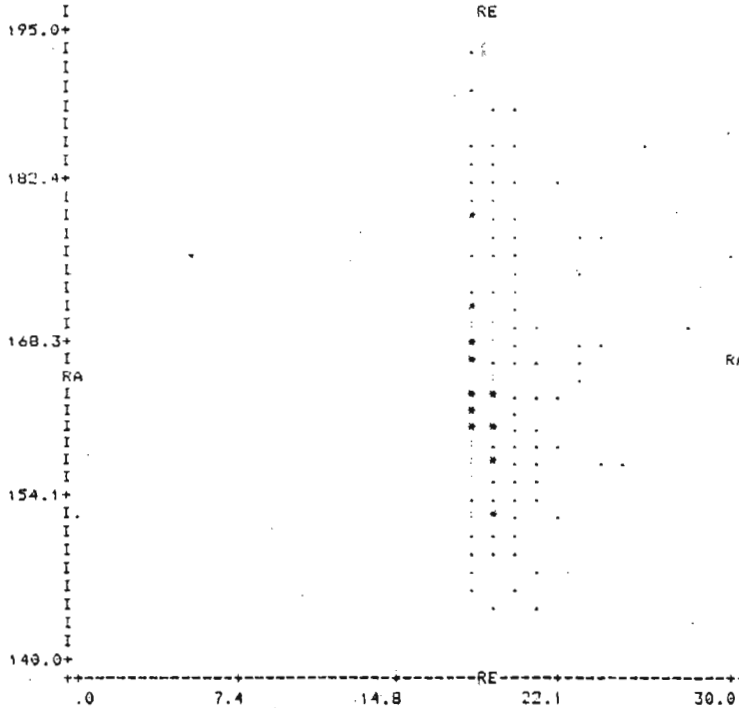
Se reinicia el procedure para mejorar la salida:

```

OUTLIERS? (INCLUDE*, EXCLUDE, REPORT, or STATISTICS)
exclude
SIZE? (NORMAL* or DOUBLE)
double
DISPLAY? (PLOT, STAT, BOTH, or BLOWUP)
both
    
```

```

DOWN-ALTURA'A' ACROSS-EDAD'E'
. = 1 - 6 . = 7 - 11 . = 12 - 17
    
```



| | MEAN | SD | R | RSQ | SIGF | SLOPE | INTCP | SEE | N |
|--------|--------|------|------|------|------|--------|--------|------|-----|
| ALTURA | 165.85 | 9.28 | .013 | .000 | .389 | 7.6-02 | 164.41 | 9.29 | 472 |
| EDAD | 18.86 | 1.59 | | | | 2.2-03 | 18.49 | 1.59 | 472 |
| EDAD | 18.86 | 1.59 | | | | | | | 473 |

Ahora el grafico es mas satisfactorio, observese que las rectas de regresion se dibujan uniendo los puntos señalados por RE y RA.

```

DISPLAY? (PLOT, STAT, BOTH, or BLOWUP)
/stop
    
```

Se finaliza la sesion en cualquier momento mediante /STOP.

Ejemplo 4 (parte 3 de 3): sesión de trabajo con SCSS.



