

Informe Técnico CAY 1984-3.

LENGUAJE INTERACTIVO PARA EL ANALISIS
DE OBSERVACIONES RADIO ESPECTRALES.
('ANAE' versión noviembre 1984)

Luis A. Delgado y Pere Planesas

<841207.1754>

OBSERVATORIO ASTRONOMICO NACIONAL	
-CENTRO ASTRONOMICO DE YEBES-	
BIBLIOTECA	
REGISTRO	NUMERO CATALOGO _____
N.º _____	ARMARIO _____ TABLA _____

INDICE

=====

1. INTRODUCCION	2
2. GENERALIDADES ACERCA DEL PROGRAMA 'ANAE'	4
2.1 Los datos	5
2.2 Los comandos.	5
2.3 Las variables	7
2.4 Los vectores.	8
3. INICIACION EN EL USO DE ANAE	11
3.1 Modo interactivo.	11
3.2 Utilizando un fichero de comandos	16
3.3 Ayudas diversas	19
4. DESCRIPCION DE LOS COMANDOS DE USO MAS FRECUENTE	23
4.1 Acceso a ficheros	23
4.2 Lectura y transferencia de datos.	25
4.3 Listados e información.	32
4.4 Realización de gráficas	33
4.5 Promedizado de espectros.	40
4.6 Ajuste de líneas de base.	45
4.7 Ajuste de gaussianas	48
4.8 Suavizado y otras alteraciones del espectro	52
4.9 Funciones matemáticas	54
4.10 Ficheros de comandos	56
4.11 Otras funciones.	62
4.12 Otras variables.	64
4.13 Fichero de comandos GRAF15	67
5. DESCRIPCION DE TODOS LOS COMANDOS.	71
ANEXOS:	
A. Iniciación al manejo del editor (EDIT)	87
B. Iniciación al manejo de ficheros almacenados en disco magnético	99
C. Utilización de las funciones TPL	125

1. INTRODUCCION

En el presente informe se describe la utilización del programa de análisis de observaciones espectrales denominado ANAE que hemos venido desarrollando paulatinamente a lo largo de prácticamente todo el presente año 1984. Se han tomado ideas de programas similares utilizados en distintos observatorios radioastronómicos (National Radio Astronomical Observatory en EEUU, Max-Planck Institut fur Radioastronomie en la República Federal de Alemania, Observatoire de Nancay en Francia y Onsala Space Observatory en Suecia) y tenido en cuenta las sugerencias de los demás radioastrónomos del Centro Astronómico de Yebes (CAY) y demás personas que han utilizado el programa hasta este momento.

Se ha tratado de hacer un programa flexible y fácil de usar, lo que ha repercutido en la programación, resultando bastante compleja. En particular, hemos querido usar una partición de sólo 32 kpalabras de memoria central del ordenador, lo cual ha exigido dividir el programa en varios segmentos.

El programa está confeccionado en una forma tan versátil que en pocos minutos permite cambiar o añadir cualquier comando, con lo cual no será posible hablar de una versión "definitiva" hasta que lo hayan utilizado muchas personas durante mucho tiempo.

No se trata en este informe de describir el funcionamiento interno del programa. Ello dará lugar más adelante a otro informe de interés casi exclusivo para los programadores en el momento en que deban cambiar o añadir algún comando.

Este informe es un intento de dotar a los observadores que deseen analizar sus datos de un pequeño libro que contenga toda la información que puedan necesitar para el aprendizaje y la utilización de un potente programa de análisis de espectros.

Con esta perspectiva, hemos dividido el presente informe en tres partes. En la primera se describe la filosofía del lenguaje creado y de su utilización como herramienta de análisis de observaciones espectrales en Radioastronomía. Esta parte es de

lectura imprescindible para cualquiera que vaya a usar el programa, conozca o no otros similares. (Vamos a hacer notar aquí, de entrada, que no hace falta conocer ningún lenguaje de programación para manejar este programa pues él mismo constituye un lenguaje, con un número limitado de instrucciones, pero muy potentes.)

En la segunda parte se inicia al utilizador en los distintos modos de manejar el programa y se le enseña a solicitar ayudas. En la tercera parte se describe la utilización de los comandos de mayor interés y uso, ilustrada con numerosos ejemplos. En la cuarta parte se describen todos los comandos de ANAE. Esta información está contenida en un fichero de ayudas accesible desde el propio programa mediante una llamada HELP. Tal fichero está pensado para que cualquier usuario de ANAE un poco experimentado pueda resolver, desde el propio programa, las dudas que se le presenten relativas al manejo o función de cualquiera de los comandos.

Este lenguaje permite trabajar de dos maneras: interactivamente o bien lanzando programas escritos en este lenguaje. Por ello hemos decidido escribir en un anexo la información más necesaria para el manejo del editor EDIT-1000 con el que está dotado el ordenador HP-1000F, para el que está escrito el programa de análisis. Se pretende con ello que cualquier usuario no iniciado pueda editar sus propios programas escritos en el lenguaje compilable por ANAE.

No vamos a describir en este informe los pasos previos a la obtención de un fichero con espectros. Los programas que reducen los datos de las observaciones espectrales a un fichero conteniendo espectros individuales calibrados analizables con ANAE se pueden encontrar descritos en el Informe Técnico CAY 1984-6. En futuros informes podrán encontrarse aquellos que se hagan más adelante.

Además, vamos a considerar que ANAE analiza única y exclusivamente observaciones espectrales (con espectros de hasta 256 puntos en su versión actual). Sin embargo es posible generar ficheros analizables con ANAE a partir de barridos realizados en continuo: la transformación pertinente la dejamos al cuidado del usuario interesado. Si en el futuro se revela necesario analizar muchas observaciones en el continuo, construiremos un programa específico para tal tarea.

2. GENERALIDADES ACERCA DEL PROGRAMA 'ANAE'

Hemos escrito ANAE en Fortran-77 pues permite realizar cambios en el programa de una manera rápida y por distintas personas, y porque los requisitos en cuanto a velocidad de cálculo no son muy grandes, por lo menos cuando se utiliza el programa interactivamente. Por otra parte, el Fortran-77 del HP-1000F es muy potente en lo relacionado con el manejo de ficheros y operaciones con caracteres, lo que nos ha facilitado bastante el poder escribir un lenguaje simple de usar. Es decir, con pocos comandos pero muy potentes y bien definidos.

El programa ANAE actúa como un compilador (o intérprete). Es decir, cada comando que se le da desde el teclado es compilado (se ve si existe, se analiza la posible existencia de errores, se comprueba que su ejecución es factible) y, si no hay ningún problema, se ejecuta su acción. Hay que esperar a que termine su ejecución antes de introducir el siguiente comando. Este método de trabajo (interactivo) es el que se utiliza más a menudo y con el que el usuario se inicia.

El otro modo de usar ANAE consiste en lanzar un programa escrito en el lenguaje que ANAE es capaz de compilar. Tal programa se encontrará en un fichero fuente (tipo 4 o 3, véase el anexo B) que tendrá el control de ANAE hasta que haya sido ejecutado en su totalidad, o encuentre algún error, o sea parado por el usuario. Tal fichero es compilado a medida que va siendo ejecutado.

Tal forma de trabajo permite ejecutar, dando un solo comando, un conjunto de instrucciones que el usuario ve que tiene que dar a menudo. Tal fichero puede contener incluso bucles, así como tomar decisiones y bifurcarse por distintos caminos.

Al acabar la ejecución de un fichero de comandos, el control vuelve al usuario, el cual puede ejecutar comandos interactivamente o bien dar el control a otro fichero de comandos.

El lenguaje que ANAE compila es, en algunos aspectos, similar al Fortran, con la diferencia que permite un tratamiento simple de

ficheros y de operaciones entre vectores. Además incluye, en forma de comandos, operaciones típicas en el tratamiento de datos espectrales (ajuste de líneas de base o de gaussianas, suavizado) y realización de gráficas. Algunas de las operaciones almacenan su resultado en variables accesibles por el usuario, con las que es posible realizar operaciones simples.

2.1 Los datos

Una de las primeras operaciones a realizar al poner en marcha ANAE es abrir un fichero de datos. Es decir, hacer accesible al programa los espectros que, en un formato adecuado, se encuentran almacenados en los registros de un fichero en algún disco magnético. Después de tal operación estos registros podrán ser leídos (READ), o sea copiados en un vector del programa.

En tales registros puede haber uno o dos espectros. Este segundo caso se da cuando se han realizado observaciones en conmutación de frecuencia o simultáneamente con dos receptores. En su estado actual, ANAE sólo acepta espectros con a lo sumo 256 valores (canales) que se reducen a 128 si hay dos espectros por registro (en este caso hablamos de dos vías).

En cada registro (que ocupa 640 palabras) se encuentra una cabecera (128 palabras) con información relativa a la observación, que es completada a medida que se realiza el análisis. En la continuación se encuentra uno o dos espectros, dependiendo del número de vías (1 o 2) con que se observó.

2.2 Los comandos

El usuario se comunica con el programa a través del teclado del terminal en que se encuentre, mandando instrucciones al programa. Una instrucción (comando) cualquiera consta de una sucesión de caracteres que se teclean como en una máquina de escribir y que se da por terminada pulsando la tecla 'return' situada a la derecha del teclado.

Las instrucciones pueden tener muy distintas funciones:

a) Efectuar operaciones.

Ejemplo: HANN

efectúa un suavizado (por convolución del espectro con la función de von Hann; a tal operación se la suele denominar suavizado "hanning").

b) Dar información al usuario.

Ejemplo: CABECERA

lista información útil contenida en la cabecera del espectro que se está analizando (coordenadas, tiempo de integración, etc.).

c) Controlar ANAE.

Ejemplo: PAUSE

detiene ANAE momentáneamente (sin pérdida de datos) mientras el usuario efectúa otras operaciones, tales como editar un fichero de comandos.

d) Realizar operaciones entre variables.

Ejemplo: B=A*0.35

El contenido de la variable A es multiplicado por 0.35 y guardado en la variable B.

e) Algunas operaciones necesitan ir acompañadas de parámetros. En otros casos tales parámetros son opcionales.

Ejemplo: READ 337

indica que debe ser leído en el fichero de datos el espectro cuyo número de identificación sea el 337.

f) Otras operaciones necesitan que alguna variable tenga un valor definido con anterioridad.

Ejemplo: GRADO=3

BASE

Para poder usar la instrucción BASE (ajuste de una línea de base al espectro) es necesario definir antes el grado del polinomio que se desea ajustar. Una vez dado un valor a la variable GRADO, este valor será usado en todos los ajustes subsiguientes mientras no sea cambiado.

g) Algunas operaciones modifican el valor de alguna variable.

Ejemplo: SIGMA

escribe en pantalla la desviación típica de los valores del espectro. Además guarda en la variable SIG1 dicho valor. Si hay dos vías, guardará los valores correspondientes de la desviación en cada vía en las variables SIG1 y SIG2.

2.3 Las variables

Hay varios tipos de variables:

a) Algunas variables son alteradas por el propio programa al ejecutar ciertas operaciones.

Ejemplo: SIGMA

coloca en las variables SIG1 y SIG2 los valores de la desviación típica correspondiente a las vías 1 y 2.

b) Otras variables son utilizadas por el programa, pero no altera su valor.

Ejemplo: GRADO

El que tenga un valor definido es imprescindible para ajustar una línea de base (operación BASE).

c) Un conjunto amplio de variables están a la disposición del usuario para su utilización arbitraria.

Corresponden a las letras del alfabeto y siguen el convenio típico en Fortran:

- las variables I hasta N pueden contener números enteros, desde -32768 hasta +32767,

- las variables A hasta H y O hasta Z pueden contener números reales con 6 o 7 cifras significativas y un exponente comprendido entre -38 y +38.

Los comandos de variables permiten:

1) Adjudicar un valor a una variable.

Ejemplos: A=33

X0=B

donde X0 es una de las variables utilizadas por el programa.

2) Examinar el contenido de una variable, que es escrito en la consola.

Ejemplo: LAST?
nos dice cuál es el número de identificación del último espectro leído.

3) Realizar operaciones simples entre variables (sólo se permite una operación por comando).

Ejemplo: R=SIG1/SIG2

2.4 Los vectores

El programa ANAE opera con 5 vectores accesibles por el usuario, que están numerados del 1 al 5.

Los comandos suelen actuar sobre uno o varios vectores, en ocasiones modificándolos.

a) vector 1 : es el vector donde se almacena el último espectro leído mediante la operación de lectura READ.

b) vector 2 : es un vector auxiliar, donde el usuario puede guardar algún espectro de interés o colocar algún espectro que vaya a utilizar como referencia (mediante, p. ej., la lectura READ R).

c) vector 3 : en este vector se efectúa la integración (o sea, el promedio de diversos espectros). Este es el vector que contiene el espectro final que es posible guardar (SAVE) en un fichero de espectros resultantes.

d) vector 4 : en este vector se almacena el resultado del último ajuste realizado, sea de una línea de base o de una o varias gaussianas. (Es, pues, alterado por los comandos BASE y GAUSS.)

e) vector 5 : en este vector se almacena el residuo del último ajuste de gaussianas. Al ser alterado sólo por el comando GAUSS, puede utilizarse como vector auxiliar adicional siempre y cuando no se ajusten gaussianas.

Normalmente el usuario maneja solamente los vectores 1 a 3, y en muy pocas ocasiones tiene que hacer referencia a tal número: los comandos suelen saber sobre qué vector deben actuar en cada momento. Sin embargo, algunos comandos sólo actúan sobre uno de ellos. Por ejemplo: GAUSS y SAVE sólo actúan sobre el vector 3.

Los comandos de operación actúan, en general, sobre el vector que señala un puntero. Tal puntero es cambiado por alguno de los comandos de la manera que parece más natural. El usuario puede cambiarlo a su vez, pues no es más que la variable PUNTERO (a la que puede referirse con el nombre completo o simplemente con PUNT).

Pueden realizarse diversas operaciones con los vectores:

a) Se les puede sumar, restar, multiplicar y dividir por un número (o el contenido de una variable). Tales operaciones se denominan: SUMA, RESTA, MULTI y DIVIDE.

Ejemplos: MULTI 0.75,3

multiplica el contenido del vector 3 por 0.75;

DIVIDE Z

divide el vector señalado por el puntero por el valor contenido en la variable Z.

b) Se pueden promediar dos vectores. El comando ACUM hace que sean promediados los vectores 1 y 3, guardando el resultado en 3.

Como peso para el promedio se utiliza, en principio, el tiempo de integración, aunque es posible promediar utilizando como peso el inverso de la varianza del espectro.

c) Se puede restar dos vectores.

Ejemplo: RESID 2,3,4

coloca en el vector 2 el residuo de restar del vector 3 el 4.

d) Se puede desplazar el contenido de un vector a otro.

Ejemplo: MOVE 2,3

guarda en el vector 2 el contenido del 3, quizás resultado del promedio de varios vectores.

-10

Algunas de las operaciones anteriores pueden realizarse sin necesidad de entrar tantos parámetros, pues el programa es capaz de tomar valores por defecto (aquellos que resulta lógico tomar o que están predefinidos).

Ejemplo: MOVE 2
guarda en el vector 2 el contenido del vector señalado por el puntero.

3. INICIACION EN EL USO DE 'ANAE'

El proceso típico de análisis de observaciones espectrales consta de varios pasos. Vamos a desarrollarlo en forma de ejemplo simple que explicaremos con detalle.

3.1 Modo interactivo

1) Puesta en marcha del programa:

- Si la pantalla está en blanco, pulsar la tecla 'return'.

- Si en la pantalla está (o ha aparecido) el símbolo dos puntos, ':', pulsar:

ANAE

y a continuación 'return'. El programa se pondrá en marcha.

- Si en la pantalla está (o ha aparecido) el símbolo "mayor que", '>', nos encontramos en ANAE.

- Si en la pantalla está (o ha aparecido) el símbolo asterisco, '*', preguntar a los demás usuarios si hay algún programa funcionando en tal terminal. Si no es así, pulsar:

KU, FMR y 'return'

y aparecerán los dos puntos, ':', con lo que nos hallamos en un caso ya explicado.

Tras la puesta en marcha del programa aparece en pantalla el símbolo "mayor que", '>', y a continuación de él el cursor. Ello indica que el programa está dispuesto a recibir comandos.

2) Apertura de un fichero de datos:

Supongamos que el fichero de datos que nos interesa se llama DORI y que se encuentra en el cartridge número 13. La apertura de tal fichero se hace mediante:

DATOS DORI::13

(La unidad de discos magnéticos que actualmente tenemos en el CAY

consta de dos discos, uno fijo y otro cambiable, que están divididos en partes que denominamos "cartridges". Los datos a analizar suelen encontrarse en los cartridges número 13 y 14.)

En caso de desconocer el número del cartridge en que se encuentra un cierto fichero de datos, se comanda:

DATOS DORI

con lo que será abierto el primer fichero, que con este nombre, encuentre el ordenador en algún cartridge. (Dos ficheros pueden tener el mismo nombre sólo si se encuentran en distintos cartridges.)

3) A continuación hay que leer y promediar los espectros contenidos en tal fichero.

La lectura se hace a partir del número de identificación que posee cada espectro. Podemos conocer los números que tienen los espectros del fichero abierto, mediante el comando "listar":

LI DATOS

Supongamos que queremos leer y promediar los tres primeros, cuyos números de identificación son, respectivamente, 337, 7324 y 210. Los leeremos y acumularemos mediante el conjunto de instrucciones siguiente:

```
READ 337
ACUM
READ 7324
ACUM
READ 210
ACUM
```

Tras la primera acumulación aparecerá el mensaje:

"inicio de la integración"

Su no aparición indica que el vector 3 donde se procede al promediado estaba ocupado. Es necesario en este caso borrarlo, mediante:

```
CLEAR 3
```

y reiniciar el proceso.

Naturalmente, no es necesario que los espectros que se acumulan estén colocados consecutivamente en el fichero de datos, pero se gana tiempo si se leen en orden de aparición en tal fichero.

4) Podemos obtener una gráfica del espectro promedio en el monitor mediante el comando:

GRAF

5) Ajuste de una línea de base.

Antes de ajustar una línea de base, y a la vista del espectro, hay que definir el grado del ajuste. Por ejemplo:

GRADO=2

En el ajuste hay que evitar el intervalo donde se encuentra (o se espera que se encuentre) la raya espectral de interés. Si ésta ocupa, p. ej., aproximadamente los canales 50 a 68, escribiremos:

INTERV 50-68

para que tal intervalo de canales no sea tenido en cuenta en el ajuste. Es posible prescindir, en un ajuste, de más de un intervalo. P. ej.:

INTERV 50-68 100-115

Al entrar el comando:

BASE

se efectúa tal ajuste y la línea de base es representada sobreimpresa a la gráfica precedente. (Para realizar la gráfica de la línea de base por separado, comandar: GRAF 4.)

6) Si el ajuste lo consideramos bueno, hay que restar tal línea de base al espectro inicial. Para ello se dispone del comando:

RESID

Se puede visualizar el resultado de tal resta simplemente mediante:

GRAF

7) Dispersión del ajuste: el comando

SIGMA

hace que sea impresa en la pantalla la desviación típica del ajuste. Si se están utilizando dos vías, aparece el valor correspondiente a cada una de ellas.

8) En el caso de estar trabajando con dos vías puede interesar promediarlas. Ello se efectúa mediante el comando:

PROM

El resultado de tal promedio queda almacenado en la via 1. Se visualiza el resultado mediante GRAF. La desviación correspondiente se obtiene mediante SIGMA. Tras este comando es conveniente hacer VIA=1.

9) Algunas rayas espectrales pueden ser caracterizadas por su intensidad, su anchura y su velocidad relativa al sistema de referencia LSR (Local Standard of Rest). En algunos de estos casos se suele ajustar la raya observada mediante una función gaussiana, que se caracteriza fácilmente con su altura, anchura y posición.

El ajuste de una gaussiana a un espectro, al que se le debe haber deducido previamente una línea de base, se efectúa mediante el comando:

GAUSS

En la pantalla aparecen los valores de la intensidad, la anchura (en km/s) y la velocidad correspondiente al pico (en km/s). Si hay dos vias, es ajustada una gaussiana a cada uno de los espectros, apareciendo en pantalla los valores que definen cada una de las gaussianas.

10) El área integrada de una raya puede obtenerse de una forma aproximada mediante el comando:

AREA

Este comando calcula el área bajo el espectro: si se desea obtener sólo el área de la raya, debe haber sido deducida antes una línea de base al espectro. En la pantalla aparecerán impresos los valores del área para cada una de las vias accesibles.

11) Impresión de los valores numéricos del espectro en la pantalla:

PRINT

Si se desean imprimir en la impresora, es necesaria la secuencia:

LL=6

PRINT

LL=LU

La primera instrucción define a la impresora (unidad 6) como unidad de listados. La última vuelve a definir la pantalla en que el usuario trabaja (LU) como unidad de listados. Si no se reasignara este valor, todos los listados aparecerían en la impresora.

Se puede imprimir también los resultados del ajuste de la gaussiana, mediante la secuencia:

```
LL=6  
GAUSS  
LL=LU
```

pues los resultados de GAUSS salen siempre impresos en la unidad indicada por LL. (Naturalmente, inicialmente es LL=LU.)

12) Cambio de escala: para pasar de grados Kelvin a Janskys o cualquier otro cambio de escala se utiliza el comando MULTI. Por ejemplo, si hay que multiplicar todos los valores del espectro por 130.5, se comanda:

```
MULT 130.5
```

13) Gráfica del espectro, con indicación de la escala de velocidades y la escala vertical, nombre de la fuente, frecuencia en reposo de la transición y fecha de observación. El comando:

```
PLOT
```

permite realizar 4 de tales gráficas en una misma hoja de papel. Pero sólo representa el espectro contenido en la vis 1.

14) Es posible guardar el resultado del análisis anterior en un fichero de resultados.

Para ello hay que abrir (hacer accesible) un fichero donde guardar los resultados. Si tal fichero se genera (o va a denominarse) RRESS, su código de seguridad es XX y se encuentra (o deseamos que se encuentre, en el caso en que vamos a crearlo) en el cartridge 13, lo abriremos (o lo crearemos) mediante:

```
RESOL RRESS:XX:13
```

Para grabar el espectro en dicho fichero, se comanda:

```
SAVE
```

15) Para despedirse de ANAE se comanda:

```
END
```

Este comando cierra los ficheros de datos y de resultados que estén abiertos, borra la gráfica que esté representada en el monitor y despide el programa.

Si lo que se desea es continuar el análisis, puede ser necesario volver a poner VIA=0 para hacer accesibles ambas vías, si ello es necesario. Si sólo se desea trabajar con una vía, se mantiene VIA=1 o bien VIA=2, según cual sea la vía que interesa.

3.2 Utilizando un fichero de comandos

Al poco de utilizar ANAE, cualquier usuario se da cuenta de que tiene que introducir a menudo unas mismas series de comandos. Ello resulta un tanto tedioso y sujeto a frecuentes errores, sobretodo cuando se lleva bastante tiempo analizando datos. Por ello pensamos en la posibilidad de que ANAE fuera capaz de interpretar comandos no procedentes del terminal sino de un fichero editado previamente.

De ello no sólo deben resultar seguridad y comodidad, sino además una considerable disminución del tiempo de análisis. Por ello recomendamos el uso de tales ficheros de comandos siempre que sea posible, tan pronto como se tenga un poco de práctica. Vamos a hacer notar, además, que con tales ficheros pueden realizarse operaciones considerablemente complejas, como podrá verse en el capítulo 4 y en el anexo C.

"Edición de un fichero" significa la escritura en él de una serie de caracteres mediante un programa a tal efecto que se denomina, en nuestro caso, EDIT y cuya utilización puede verse descrita, de manera somera, en el anexo A.

Mediante el programa EDIT se escriben en líneas sucesivas los comandos que, normalmente, el usuario introduciría consecutivamente desde la consola. La instrucción final de un fichero de comandos tiene que ser CLOSE: tal comando hace que el control de ANAE vuelva al usuario y cierra, o sea despide, tal fichero.

Ejemplo 1:

Consideremos un fichero de comandos de nombre COMCOM cuyo contenido sea el siguiente:

DATOS DORI::13

VIA=0

CLEAR 3

READ 337

ACUM

READ 7324

ACUM

READ 210

ACUM

GRAF

GRADO=2

INTERV 50-68

BASE

RESID

SIGMA

PROM

VIA=1

MULT 130.5

GRAF

SIGMA

AREA

GAUSE

PLOT

CLOSE

Este fichero ejecuta, a grandes rasgos, casi todo lo que se ha descrito en el apartado 3.1, salvo lo relativo a guardar el resultado final en un fichero de resultados.

Inicialmente abre un fichero de datos, establece que se quiere analizar ambas vias (VIA=0), por si antes se hubiera hecho, p. ej., VIA=1, y se pone a cero el vector acumulador (el vector 3). Estas precauciones conviene tenerlas siempre muy en cuenta.

Se leen y acumulan 3 espectros, tras lo cual se realiza la gráfica y se ajusta una línea de base, para lo cual se sabía de antemano en qué zona se encuentra la raya y su anchura aproximada (quizás antes se había visualizado uno de los espectros). Deducida la línea de base, se imprime la sigma (dispersión) del ajuste, se promedian las dos vias, se cambia la escala (quizás se pasa de grados Kelvin a Janskys) y se realiza la gráfica final. A continuación se calculan e imprimen la sigma, el área bajo la raya y los resultados del ajuste de la gaussiana. Por fin, PLOT realiza

la gráfica en papel mediante el plotter.

La sentencia CLOSE devuelve el control de manera adecuada al programa principal, o sea al usuario.

El comando que permite pasar el control al fichero de comandos, llamado COMCOM en este ejemplo, es:

```
GO COMCOM
```

Una vez ha sido ejecutado, si se desea guardar el espectro resultante, el usuario puede realizar ahora lo descrito en el punto 14 del apartado 3.1 relativo a la instrucción SAVE.

Como va a verse en el apartado 4.10 los ficheros de comandos pueden ser instrumentos muy versátiles y potentes, pues puede hacerse que se bifurquen por distintos caminos.

Ejemplo 2:

El siguiente fichero permite la lectura y acumulación de un grupo de espectros cuyos identificadores inicial y final están contenidos en las variables I y J. Termina realizando la gráfica del espectro resultante. No es ajustada ninguna línea de base. Se supone que ello lo hará a posteriori el usuario definiendo los intervalos y el grado del polinomio que considere adecuados a la vista del espectro final. Se supone que los identificadores de los sucesivos espectros tienen números estrictamente crecientes y que difieren en 3 unidades.

```
IF I EQ 0 PAUSE
IF I GT J PAUSE
A: READ I
LAST?
ACUM
I=I+3
IF I LE J GO TO A
GRAF
CLOSE
```

Supongamos que este fichero se denomina BUCLE. Antes de ponerlo en marcha hay que abrir un fichero de datos y dar los números inicial y final a analizar (no tienen porqué ser todos los del fichero), para lo cual quizás habrá que listar el contenido del fichero en la pantalla. Sea DDD el fichero de datos:

```
OPEN DDD
LI DATOS
```

A continuación asignamos a I y J los identificadores primero y último:

I=3734

J=4216

y ponemos en marcha el programa contenido en BUCLE:

GO BUCLE

Este programa analiza si I es distinto de cero. Si fuese 0, seguramente se trataría de que el usuario ha olvidado el darle un valor. A continuación compara los valores de I y J; debe ser J>I. En caso de que falle cualquiera de tales condiciones, el programa BUCLE se detiene, con lo que el usuario puede asignar valores adecuados a I y/o J. Para que continúe su ejecución se comanda:

GO

simplemente.

A continuación hay un bucle en el que es leído un registro (de identificador I). Tal número es colocado dentro de la variable LAST (=último) y es escrita en la pantalla cuando es pedido el valor de tal variable mediante:

LAST?

de forma que se tiene información de lo que va siendo leído. Tras ACUM, I es incrementado en 3 unidades

I=I+3

y, si I todavía no excede el valor de J, el control pasa (GO TO) a la sentencia etiquetada con la letra A, que es:

A: READ I

con lo que es leído un nuevo espectro.

Por fin, cuando I>J es hecha la gráfica del espectro resultante (GRAF) y el control vuelve al usuario (CLOSE).

3.3 Ayudas diversas

Todo lo descrito en los apartados 3.1 y 3.2 debe servir para que cualquier persona interesada sea capaz de iniciarse en el manejo de ANAE. Sin duda le van a surgir dudas y cometerá errores que, con la práctica aprenderá a evitar. Descubrirá asimismo las diversas sutilezas del programa.

Creemos que es interesante empezar a manejar tal programa antes de leer el capítulo 4, donde se describen, ya con más detalle y con

numerosos ejemplos, la mayoría de los comandos de ANAE. Sin embargo, hay un par de comandos más que creemos pueden ser de utilidad durante tal iniciación.

A) El primero de ellos es HELP. Tiene varias posibilidades que vamos a examinar someramente.

- Si se comanda simplemente:

HELP

se obtiene una lista de todas las instrucciones y variables disponibles en la versión de ANAE que se esté utilizando.

- Si se desea información adicional sobre alguno de los comandos, p. ej. PROM, se pide en la forma:

HELP PROM

y aparecerá en la pantalla información muy completa, pero en forma compacta, sobre el comando deseado.

- Puede pedirse información sobre el uso que se les da a los distintos vectores mediante:

HELP VECTORES

e información sobre el uso de cada una de las variables mediante:

HELP VARIABLES

La información a la que accede HELP es la contenida en el capítulo 5 del presente informe.

B) Otro comando de naturaleza absolutamente distinta pero que es de gran ayuda para corregir de forma inmediata ciertos errores es:

RECUP

Su función es deshacer la última alteración hecha.

Ejemplo 1:

Hacemos un suavizado del espectro mediante el comando HANN, que efectúa una convolución del mismo:

READ 317

GRAF

HANN

GRAF

Vista la última gráfica (que contiene el espectro suavizado), si

concluimos que tal suavizado no nos mejora apreciablemente su aspecto y queremos recuperar el vector original, comandamos:

RECUP
GRAF

La gráfica permite cerciorarnos de que efectivamente ha sido recuperado. (Podría haber sido leído de nuevo el espectro original, READ 317, pero suele tardar más.)

Ejemplo 2:

Acumulamos, por error, 2 veces un mismo espectro y nos damos cuenta inmediatamente:

READ 33
ACUM
READ 212
ACUM
READ 45
ACUM
ACUM
RECUP

Con el comando RECUP hemos reintegrado al vector de acumulación (3) los valores que tenía antes de la última acumulación, con lo que su contenido será el promedio de los espectros 33, 212 y 45, éste sólo una vez.

Ejemplo 3:

Restamos una línea de base a un espectro, pero luego no estamos de acuerdo con el espectro resultante y queremos recuperar el espectro original:

READ 12
GRAF
INTERV 60-80
GRADO=2
BASE
RESID
GRAF
RECUP
GRAF
INTERV 60-85
etc.

Atención: RECUP sólo deshace el ULTIMO cambio realizado.

C) La variable LAST, que ha aparecido fugazmente al final del apartado anterior, puede ser utilizada para releer un espectro que hemos alterado considerablemente y que ya no podemos recuperar mediante RECUP. Para releer el último espectro leído se comanda:

READ LAST

4. DESCRIPCION DE LOS COMANDOS DE USO MAS FRECUENTE

Se supone que el lector de este capítulo conoce todo lo que se ha descrito en el capítulo 3.

La descripción de cada comando que aquí aparece no siempre es completa. En ocasiones tienen opciones especiales de poco uso o que consideramos no es conveniente que se utilicen hasta poseer un buen dominio del programa.

4.1 Acceso a ficheros

El usuario puede manejar, desde ANAE, datos de varios ficheros, aunque con distintas funciones.

En principio, el análisis de observaciones espectrales pasa por la lectura de datos almacenados en un fichero, se trabaja sobre ellos y el resultado se saca en forma gráfica en el plotter, en forma numérica en la impresora y/o se almacena en otro fichero en disco, un fichero en el que el usuario almacena sus resultados. Naturalmente, los ficheros de datos y de resultados tienen un formato idéntico, de manera que un fichero de resultados puede ser utilizado posteriormente como fichero de datos.

DATOS

La apertura de un fichero de datos de nombre MANIP, a fin de hacerlo accesible al programa, se consigue mediante:

DATOS MANIP

o bien

DATOS MANIP::13

si sabemos que se encuentra en el cartridge 13.

RESUL

La apertura de un fichero de resultados de nombre RRESS y con código de seguridad AA será:

RESUL RRESS:AA

Si no tiene código de seguridad, basta dar:

RESUL RRESS

pero si lo tiene es imprescindible darlo en el caso del fichero de resultados.

LI

Los espectros contenidos en el fichero de resultados no pueden ser leídos. Sólo pueden ser sustituidos por otros o bien puede añadirse otros nuevos. El contenido de tal fichero puede obtenerse mediante:

LI RESULT

con lo que aparecerán listados número de identificación, fuente, frecuencia en reposo, coordenadas y fecha.

Asimismo se puede hacer un listado análogo del fichero de datos mediante:

LI DATOS

El comando LI es mucho más potente, pues permite listar cualquier fichero escrito con caracteres ASCII que se encuentre en cualquier cartridge. Así, por ejemplo, si se desea utilizar un fichero de comandos que haga una cierta función y se está dudando entre dos, denominados ANA y EVA, pueden listarse en pantalla mediante:

LI ANA

y a continuación, si es necesario:

LI EVA

OPEN

Un fichero de comandos puede tener varias posibles funciones (digamos que puede contener varios programas). Para correr un programa que se encuentra en un fichero de nombre, p. ej., GOLF, debemos primero hacerlo accesible:

OPEN GOLF

A partir de este momento se pueden correr programas de este fichero.

Para listar un fichero de comandos abierto (mediante OPEN)

basta dar:

LI

CLOSE

Los ficheros abiertos pueden ser cerrados mediante la función CLOSE. Hay, por lo tanto, tres posibilidades:

CLOSE DATOS

CLOSE RESULT

CLOSE

La última instrucción cierra el fichero de comandos que esté abierto.

Cuando el usuario de ANAE se despide de tal programa, mediante END, son automáticamente cerrados los ficheros, de cualquier tipo, que estén abiertos. La existencia de la función CLOSE es para permitir mayor versatilidad.

Ejemplo:

Analizaremos el fichero MANIP y almacenaremos los resultados en RRESS. A continuación queremos analizar los espectros que están en RRESS, almacenando el resultado final en el mismo fichero RRESS. Procederemos como sigue:

```

DATOS MANIP
RESUL RRESS:AA
análisis
CLOSE DATOS
CLOSE RESULT
DATOS RRESS
análisis
CLOSE DATOS
RESUL RRESS:AA
se almacena el resultado final
END

```

Obsérvese que al final hemos tenido que cerrar DATOS pues el fichero RRESS no puede ser simultáneamente fichero de datos y de resultados.

4.2 Lectura y transferencia de datos

Una vez abierto un fichero de datos es posible leer el contenido de un cierto registro (un espectro o dos) y almacenarlo en uno de los vectores de que dispone el usuario. Normalmente este vector es el número 1 (vector de lectura), aunque es posible almacenarlo también en el vector número 2 (vector de referencia).

READ

La lectura puede comandarse de distintas maneras. Los espectros pueden ser leídos individualmente mediante:

a) lectura de un espectro (registro) cuyo número de identificación es 43:

READ 43

b) lectura de un espectro cuyo número de identificación está almacenado en la variable I:

READ I

Sin embargo, si pretendemos analizar un grupo de espectros hay otra manera más cómoda, combinando varios comandos:

a) comando SCAN : permite definir identificadores a analizar, aislados o en intervalos.

b) comando OFFSCAN : permite definir identificadores aislados que se consideran no válidos pero que se hallan inmersos en alguno de los intervalos antes definidos.

Ejemplo 1:

SCAN 33-89 99-143 95

OFFSCAN 77 100 107

Al comandar:

READ

será leído el espectro número 33. Al comandar de nuevo READ será leído el siguiente en el fichero (sea o no correlativo) y así sucesivamente hasta leer el espectro 89. Si el espectro 77, 100 o 107 se encontraba entre ellos, no habrá sido leído. (Recordar que estos números son números de identificación, no necesariamente números de orden ni siquiera números de registro.) Cuando haya sido

leído el 89, al siguiente comando READ será leído el 99 y así sucesivamente hasta llegar al 143. Finalmente leerá el 95. Si se intenta leer más allá, aparecerá en pantalla un mensaje de error y la variable:

ICODE

asociada a la lectura tendrá un valor nulo, indicando lectura no realizada.

Ejemplo 2:

Hemos comentado antes que los espectros que se encuentran entre el 33 y el 89 del ejemplo anterior no tienen por qué tener números correlativos. Pueden ser:

33, 34, 35, ..., 89

o bien:

33, 36, 39, 42, ..., 89

o incluso:

33, 77, 125, 12, 78, 9324, 3, 89

En cualquier caso-serán leídos todos los que se encuentren entre el 33 y el 89 (ambos inclusive) salvo aquellos que estén en la lista de OFFSCAN.

Variables asociadas a READ:

a) ICODE : ya citada. Vale 0 cuando la lectura ha tenido lugar sin problemas. Vale -1 cuando no ha sido posible tal lectura por cualquier causa. Ello permite tomar decisiones en bucles en ficheros de comandos.

Ejemplo:

```

DATOS MANIP
SCAN 33-89 99-143
OFFSCAN 77 100 107
DO 1000
  READ
  IF ICODE LT 0 GO TO 33
  ACUM
END DO
33: PUNT=3
GRAF
etc

```

b) LAST : contiene el valor del último identificador leído correctamente. Para saber cuál es, se comanda:

LAST?

Para leer de nuevo el último espectro leído:

READ LAST

c) PUNT : es modificado al efectuar una lectura, de manera que tras READ es PUNT=1 automáticamente.

READ R

Lectura y almacenamiento en el vector número 2. Sirve todo lo dicho anteriormente, pero se especifica que el vector es el 2 escribiendo una R tras READ.

Ejemplo:

Leer el registro 77 y almacenarlo en el vector 2 pues vamos a utilizarlo como referencia:

READ R 77

Tras READ R el puntero queda en 2: PUNT=2.

MOVE

El contenido de un vector puede ser trasladado a otro merced a este comando, que actúa de la manera que ilustra los siguientes ejemplos:

MOVE 2 3

almacena en el vector 2 el contenido del vector 3. En cambio

MOVE 2

almacena en el vector 2 el contenido del vector señalado con el puntero.

Ejemplo:

Se analizan y promedian 10 espectros, se almacena el resultado en 2, se analizan y promedian otros espectros correspondientes a la misma fuente y se comparan:

DATOS MANIP1

SCAN 1-10

```

DO 10
  READ
  ACUM
END DO
GRAF
MOVE 2
DATOS MANIP2
SCAN 1331-1350
CLEA 3
DO 20
  READ
  ACUM
END DO
GRAF

```

Tenemos en el vector 2 el primer promedio y en el vector 3 el segundo. Se pueden visualizar mediante GRAF 2 y GRAF 3 (o simplemente GRAF, pues el puntero está en 3 tras el comando ACUM).

La instrucción MOVE no cambia el valor del puntero.

SAVE

Si se desea almacenar el resultado de un análisis en un fichero de resultados abierto, hay que colocar, si no lo está ya, el resultado de tal análisis en el vector 3 y comandar:

```
SAVE
```

En el fichero de resultados será almacenado un espectro con el mismo número de identificación.

ATENCIÓN: recordar que SAVE guarda el vector 3 !!!.

Si se desea cambiar el identificador por otro, para tener, p. ej., números correlativos en el fichero de resultados, se coloca el nuevo valor del número identificador tras SAVE. Por ejemplo:

```
SAVE 17
```

Si además se desea cambiar el nombre del espectro, se entra el nuevo nombre tras el identificador separado por una coma. Ejemplo:

```
SAVE 17,Orion J:3-2 v=1
```

o bien simplemente:

```
SAVE ,Orion J:3-2 v=1
```

si no se desea cambiar el número de identificación.

Al ejecutar SAVE, si en el fichero de resultados es encontrado un espectro con el mismo identificador que el que se desea guardar, va a ser sustituido por el nuevo. Para evitar sustituciones no deseadas, el programa avisa de la existencia del otro y pide confirmación. Si se desea la sustitución, se contesta YES o SI. Si no, se contesta cualquier otra cosa. De este modo se puede, por ejemplo, cambiar el identificador y ejecutar de nuevo SAVE.

Ejemplo:

Tenemos en el vector 3 un espectro con identificador 98 y deseamos guardarlo en un cierto fichero de resultados:

```
SAVE
    Identificador ya existente. Será sustituido.
OK? NO
    Comando no ejecutado
SAVE 99
```

En el último comando le hemos cambiado al espectro su número de identificación antes de guardarlo.

RECUP

Permite intercambiar datos entre un vector de reserva (el vector 6) y el último vector modificado. Es decir, permite "volverse atrás" de una modificación hecha a cualquier vector.

Ejemplo 1:

Suavizamos un vector y, tras realizar su gráfica, decidimos que no nos convence tal suavizado:

```
READ 33
GRAF
HANN
GRAF
RECUP
```

Ejemplo 2:

Integramos en el vector 3 el contenido del vector 1 por error:

```
ACUM
```

RECU

La acción de RECUP (o RECU) es ahora dejar el vector 3 tal cual estaba antes de realizar la acción de ACUM.

Ejemplo 3:

Leemos un nuevo espectro cuando el anterior no estaba analizado:

```

READ
LAST?
      33

```

```

READ
LAST?
      55

```

RECU

hace que volvamos a tener en el vector 1 el espectro 33.

Atención: Naturalmente RECUP no actúa sobre el fichero de resultados RESULT. Actúa solamente sobre vectores.

RECU sólo permite recuperar lo alterado por la última modificación. En particular, se considera que RECU (como MOVE) es una modificación. Es decir, en el ejemplo siguiente:

```

N=N+3
READ N
HANN
RECU

```

tras RECU lo que tenemos en el vector 1 (que ha sido alterado por HANN) es el espectro sin suavizar. Si comandamos de nuevo:

RECU

no obtendremos en 1 el vector leído en la operación precedente, sino que obtendremos el vector suavizado (pues al aplicar RECU la primera vez tal vector había sido trasladado al vector 6).

Sin embargo, y a pesar del ejemplo precedente, el vector 6 no puede ser considerado como un vector auxiliar pues son muchos los comandos que alteran su contenido. (El vector auxiliar no puede ser referenciado explícitamente en ningún comando. P. ej., el puntero nunca puede valer 6.)

4.3 Listados e información

LI

Ya hemos explicado parte de las posibilidades de este comando en el apartado 4.1. Vamos a exponer aquí otra más. Los comandos que se ejecutan desde el teclado y los procedentes de comandos son almacenados en un fichero 'scratch' (que es borrado al despedirse de ANAE). Si se desean ver los últimos comandos entrados basta dar:

LI INPUT

CABECERA

Lista toda la información de la cabecera asociada un vector.

PRINT

Lista los valores numéricos de una o ambas vías de un vector.

En ambos casos, si se comanda:

CABECERA o bien CABE
PRINT o bien PRIN

es listada la información relativa al espectro indicado por el PUNTERO. Si se desea obtener información de otro vector, por ejemplo el 2 cuando PUNT=3, se debe comandar, respectivamente:

CABE 2
PRIN 2

Los listados anteriores son hechos en la pantalla. Para conseguir tenerlos en la impresora hay que utilizar una variable asociada, LL, en la que se indica cual es la unidad lógica para listar. A la impresora le corresponde el número 6.

La unidad lógica en que el usuario está trabajando está contenida en la variable LU.

Ejemplo:

READ 12
LL=6

CABE
PRIN
LL=LU

En este ejemplo, un usuario ha leído un espectro y ha listado su cabecera y sus valores numéricos en la impresora. Ha continuación ha vuelto a colocar la unidad lógica de listar en LU, pues en caso contrario todos los demás listados seguirían apareciendo en la impresora, lo que no resulta interesante ni económico.

STATUS

Permite obtener información sobre diversos grupos de variables. Tiene varias opciones:

- para listar los valores de las variables relativas a la realización de gráficas es:

STAT GRAF

- para listar valores relativos al ajuste de gaussianas:

STAT GAUSS

- para listar valores relativos al ajuste de líneas de base:

STAT BASE

- para listar el nombre de los ficheros abiertos:

STAT FICH

HELP

Permite obtener información sobre los diversos comandos.

Ejemplos:

HELP PAUSE

HELP BASE

HELP STATUS

Para obtener la lista de todos los posibles comandos y variables, basta comandar:

HELP

4.4 Realización de gráficas

Mediante ANAE pueden realizarse gráficas de espectros en un

monitor (17) y en un plotter (15). Los comandos son GRAF y PLOT. Hay además algún comando auxiliar y un gran número de variables asociadas.

GRAF

Permite realizar gráficas en el monitor y en el plotter. Tales gráficas puede ser realizadas con tamaños establecidos por el usuario.

Si el espectro tiene 2 vías, aparecerá la gráfica de ambas si VIA=0. Si se desea la gráfica de sólo una de las vías, hay que establecer antes de comandar GRAF un valor no nulo de VIA. Es decir, hacer VIA=1 o VIA=2.

Al poner en marcha ANAE está establecido que las gráficas con este comando sean realizadas en el monitor (unidad 17) y con un tamaño preestablecido, que es posible cambiar (naturalmente permanecerá cambiado mientras el usuario no se despida del programa).

Con este comando puede hacerse la gráfica de cualquiera de los 5 vectores accesibles por el usuario. Si se comanda:

GRAF

es realizada la gráfica del vector indicado con el PUNTERO. Si se comanda:

GRAF 4

es realizada la gráfica del espectro o la línea de base o la gaussiana contenida en el vector 4.

La escala vertical de la gráfica se autoajusta a fin de que el usuario no tenga que preocuparse de dar límites. Sin embargo, si el usuario desea poner unos ciertos límites fijos (p. ej., a fin de poder comparar cómodamente dos gráficas) puede hacerlo mediante las variables auxiliares:

YI e YF

Ejemplo:

Queremos realizar las gráficas de los espectros almacenados en los vectores 2 y 3 con idéntica escala. Podemos determinar unos

limites convenientes por simple inspección visual de las gráficas autoajustadas:

GRAF 2

GRAF 3

A continuación establecemos unos limites adecuados, p. ej.:

YI=-0.05

YF=0.2

y repetimos las gráficas:

GRAF 2

GRAF 3

que aparecerán sucesivamente (no superpuestas) con igual escala.

Para tener de nuevo una escala autoajustable es necesario hacer que YI e YF sean iguales:

YI=YF

Si queremos variar la posición de la gráfica en el monitor, se pueden cambiar los valores del origen inferior izquierdo de la gráfica:

XOTV

YOTV

También puede cambiarse el tamaño horizontal y el vertical de la gráfica:

DXTV

DYTV

Los valores introducidos para las 4 variables anteriores vienen dados en puntos de pantalla (la pantalla tiene 256x256 puntos). Se aconseja no cambiar los valores dados por defecto.

Si se desea realizar con GRAF una gráfica en el plotter el proceso resulta bastante más laborioso, pues no basta dar simplemente GRAF. Si se trata de tener una gráfica meramente informativa, lo más práctico es recurrir a la instrucción PLOT (ver más abajo). Se aconseja usar GRAF para el plotter únicamente cuando se pretende hacer una gráfica grande o casi definitiva.

En lo que sigue vamos a describir todos los pasos y posibilidades para realizar una gráfica en el plotter. En primer lugar hay que establecer que la unidad de gráficos sea el plotter (unidad 15). Para ello se cambia el valor de la variable GRAF:

GRAF=15

(Obsérvese que hay una variable que se llama GRAF y una instrucción con el mismo nombre. El programa distingue cual es cual por la existencia del signo '=' justo detrás de la última letra del comando, la F en este caso. Esta filosofía la hemos adoptado en diversas ocasiones a lo largo del programa, como se verá más abajo.)

A continuación es necesario definir valores para el origen de la gráfica en el papel, dado por:

XOPL

YOPL

y el tamaño de dicha gráfica:

DXPL

DYPL

En este caso las unidades se dan en milímetros. El tamaño útil del papel es de 257.5x190 mm. Atención: no sobrepasar los extremos de esta zona, pues las consecuencias son difíciles de prever y pueden resultar desagradables.

Puede ser conveniente reducir la velocidad de dibujo. Inicialmente esta velocidad está establecida en 38 cm/s, la máxima. Sin embargo para ciertos tipos de gráfica puede ser conveniente utilizar una velocidad menor; p. ej., si se dibuja con Rotring es conveniente usar una velocidad:

VS=2

o incluso: VS=1

Si se desea volver a la velocidad rápida, hacer:

VS=0

Ejemplo:

Realizamos en la pantalla la gráfica de un espectro. Escogemos unos límites para la gráfica y repetimos la gráfica en la pantalla para ver como queda:

READ 888

GRAF

YI=-.02

YF=.78

GRAF

Si estamos de acuerdo con esta escala pasamos a establecer la

posición y tamaño de la gráfica en el plotter:

XOPL=70

YOPL=50

DXPL=150

DYPL=100

y realizamos tal gráfica en una hoja de papel:

GRAF=15

GRAF

Si nos convence, podemos cambiar el rotulador por un Rotring, pasar a velocidad más lenta y repetir la gráfica en otra hoja de papel:

VS=2

GRAF

Realizado todo el proceso anterior, pasamos la velocidad a su valor normal y hacemos que la unidad gráfica vuelva a ser el monitor (17) antes de que se nos olvide:

VS=0

GRAF=17

Quizás sea conveniente también dejar la escala autoajustable:

YI=YF

PLOT

Permite hacer una gráfica en el plotter de la vía 1 de un espectro de una manera cómoda.

Además de realizar la gráfica añade un poco de información, tal como la escala horizontal en velocidades, la escala vertical (en las unidades de trabajo, Kelvins o Janskys), la fuente, la fecha de observación y la frecuencia en reposo.

En cada hoja de papel dibuja hasta 4 gráficas. La posición de la gráfica viene dada por la variable asociada PLOT. Nuevamente, no confundir la instrucción PLOT con la variable del mismo nombre.

La instrucción se comanda mediante:

PLOT

con lo cual ANAE realizará la gráfica en el plotter de aquel vector que esté indicado por el PUNtero (pero sólo de la vía 1, en el caso de que tenga dos vías). Si se desea realizar la gráfica del vector 4, será:

PLOT 4

La posición de la gráfica viene dada por la variable PLOT de la siguiente manera (antihoraria):

PLOT=1	gráfica en la parte sup. izq. del papel
=2	" " " " inf. izq. " "
=3	" " " " inf. der. " "
=4	" " " " sup. der. " "

Cada vez que se comanda PLOT, la variable PLOT es incrementada en 1, de manera que no se van a solapar gráficas por descuido. Cuando ha realizado la gráfica correspondiente a PLOT=4, avisa que es necesario cambiar la hoja de papel.

Ejemplo:

```

SCAN 11-20
PLOT=1
READ
PLOT
READ
PLOT
READ
PLOT
READ
PLOT
    Cambiar la hoja de papel
etc

```

Es posible superponer 2 gráficas mediante PLOT, pero para ello es aconsejable definir una escala, pues en caso contrario cada gráfica se autoajustará en el mismo espacio, con escalas distintas.

Ejemplo:

Hemos ajustado una línea de base a un espectro y queremos hacer la gráfica, mediante PLOT, de tal espectro y, superpuesta a él, de la línea de base ajustada:

```

READ 50
GRAF

```

```

BASE
YI=-.035
YF=0.115
PLOT
PLOT=PLOT-1
PLOT 4

```

pues en el vector 4 se encuentra la línea de base ajustada al comandar BASE.

CURSOR *funcion tambien con velocidad: CURS [v] parametro*

Este comando permite indicar la posición de un cierto canal en una gráfica anteriormente realizada en el monitor.

Si queremos que aparezca una raya vertical en al pantalla indicando la posición del canal 37, p. ej., se comanda:

```
CURSOR 37
```

o bien:

```
CURS 37
```

Si están representadas 2 vias, aparecerá tal raya indicativa en ambas gráficas.

Este comando resulta de gran utilidad en la determinación de los intervalos para el ajuste de líneas de base.

Ejemplo:

En el ajuste de una línea de base queremos evitar la zona en que se encuentra la raya espectral. Para fijar aproximadamente los extremos de ésta se pueden ensayar distintos canales. Por ejemplo:

```
CURS 37
```

```
CURS 65
```

Si el primero de ellos no nos convence, probamos, p. ej.:

```
CURS 39
```

En tal caso el intervalo podría ser:

```
INTERV 39-65
```

También es útil este comando cuando, a causa de alguna interferencia o mal funcionamiento, vemos que uno o varios canales tienen valores claramente anormales. El precisar su posición se realiza cómodamente con ensayos sucesivos mediante CURS.

Ejemplo:

Si vemos un canal "malo" en las cercanías del canal 150, probamos:

CURS 150

Si vemos que está más hacia la derecha, podemos probar:

CURS 148

y así sucesivamente hasta dar con él. A continuación es posible eliminarlo (véanse las sentencias DUM1 y DUM2 más abajo).

TPL

Es otro programa conversacional que se encuentra dentro de ANAE y al que se puede acceder tanto interactivamente como desde un fichero de comandos.

Se utiliza para rotular (escribir caracteres) una gráfica realizada previamente con el plotter (desde ANAE o desde cualquier otro programa).

La descripción detallada de TPL puede verse en el anexo C.

Existe asimismo un programa denominado TPL independiente de ANAE. Su inclusión dentro de ANAE responde a dos hechos:

- el hecho de que GRAF se limita a dibujar la gráfica en el plotter, con las escalas de abscisas y ordenadas, pero ninguna información adicional;
- el hecho de que, si hay que rotular de manera parecida distintas gráficas, ANAE permite el uso de ficheros de comandos que facilitan enormemente tal tarea.

4.5 Promediado de espectros

A menudo el proceso a que deben ser sometidos los datos es el que describimos a continuación. Se tienen almacenados en un fichero una serie de espectros correspondientes a una misma fuente tomados sucesivamente en una misma sesión de observaciones o correspondientes a observaciones realizadas en distintos días. Se trata de leer cada uno de los espectros y promediarlo con los

demás.

Tal promedio se hace utilizando un ponderado. Inicialmente como peso para el ponderado se toma el tiempo de integración para cada espectro, que puede ser distinto de uno a otro. Es decir, si un espectro corresponde al doble de tiempo de observación de otro, pesará el doble.

ACUM

Tal promediado (o acumulación) de espectros se realiza mediante el comando ACUM que automáticamente toma el espectro contenido en el vector 1 y lo promedia con el contenido en el vector 3. Sólo es posible esta configuración de vectores.

Antes de iniciar una acumulación de espectros es conveniente limpiar (poner a cero) el vector 3, para lo que se utiliza el comando:

```
CLEAR 3
```

Ejemplo:

```
DATOS MANIP
SCAN 33-79
CLEA 3
DO 1000
  READ
  IF ICODE EQ 0 GO TO 10
  ACUM
END DO
10: PUNT=3
GRAF
```

Al ejecutar ACUM el puntero es cambiado y toma el valor 3.

Si dos espectros corresponden a intervalos en velocidades ligeramente distintos, son desplazados uno respecto del otro y sólo se promedian los intervalos comunes, siendo el resto no tenido en cuenta.

Una variable asociada a ACUM es VIA. Con esta variable se

indica qué vía o vías se desean analizar o acumular. Veamos algunos ejemplos de su uso:

a) Si los espectros que analizamos sólo tienen datos en la vía 1, al comandar:

READ

aparecerá el mensaje:

identificador solo con vía 1

Es conveniente en tal caso, para ahorrar tiempo, hacer:

VIA=1

b) Si se van a acumular espectros algunos de los cuales sólo tienen vía 1 y otros ambas vías, hacer:

VIA=0

que indica que se van a utilizar ambas vías o las disponibles.

c) Si sólo se va a analizar la vía 2, hacer:

VIA=2

d) Si leemos un espectro con dos vías pero sólo nos interesa la segunda, pues la primera es, p. ej., de muy mala calidad, se procede como sigue:

READ 33

GRAF

VIA=2

ACUM

VIA=0

Con el último comando reestablecemos el hecho de que deseamos analizar las dos vías en los espectros que sigan.

PROM

Cuando se han estado acumulando espectros, algunos de ellos o todos con dos vías, en el vector 3 tendremos un espectro resultante en cada vía. Podemos, a su vez, promediar estos dos espectros, quizás tras comprobar que efectivamente son muy parecidos, mediante el comando PROM. El resultado se almacena en la vía 1 y la vía 2 deja de ser accesible.

Ejemplo:

```

VIA=0
SCAN 33-79
CLEAR 3
DO 1000
  READ
  IF ICODE EQ 0 GO TO 10
  ACUM
END DO
10: PUNT=3
GRAF
PROM
VIA=1
GRAF

```

El comando PROM, sin embargo, puede actuar sobre cualquiera de los 5 vectores. Por defecto actúa sobre el indicado por el puntero.

Ejemplo:

El puntero está señalando el vector 3 y queremos promediar las dos vías del vector 2:

```
PROM 2
```

PESO

Este comando permite escoger entre dos formas de ponderado: en tiempo de integración (tal como se ha descrito) o en inverso de la varianza (véase Memoria de 1981 relativa al proyecto de la CAICYT, pag. 39). Para unas condiciones uniformes del sistema, ambos ponderados son equivalentes; si el sistema ha variado, el ponderado según la varianza pesa más los espectros de mejor "calidad".

En el segundo caso se toma como medida de la varianza el cuadrado de la dispersión del ajuste (σ) de una línea de base al espectro. Es decir, se considera que tal dispersión es una medida de la calidad del espectro. La acumulación se hará según este criterio (que es el óptimo en algún sentido) si antes se establece:

```
PESO 5
```

Para volver al criterio de ponderado según el tiempo de integración, es:

o simplemente: PESO T
PESO

Naturalmente, para la utilización de un promediado con PESO S es necesario que cada espectro tenga una "sigma" asociada. Ello se consigue simplemente ajustando una línea de base.

El establecer un modo u otro de ponderado afecta tanto a ACUM como a PROM.

Ejemplo:

Promediado de 10 espectros, tras ajustarles una línea de base que no les es deducida, sino que se utiliza simplemente para estimar la calidad de cada uno de ellos. Se utiliza como ponderado el inverso de la varianza:

```
DATOS MANIP
SCAN 1-10
CLEA 3
VIA=0
GRAF=0
GRADO=3
INTE 50-80
PESO S
DO 10
  READ
  BASE
  ACUM
END DO
PROM
VIA=1
GRAF=17
GRAF
```

En este ejemplo se ha hecho GRAF=0 durante el bucle para evitar que aparezcan en pantalla las gráficas de las líneas de base, que son hechas automáticamente. Asimismo se ha definido un intervalo de no ajuste bastante amplio alrededor de donde se espera que esté la raya espectral, a fin de estimar el valor del "ruido".

4.6 Ajuste de líneas de base

Muy a menudo (especialmente cuando se trabaja en conmutación de frecuencia) el espectro obtenido no tiene el mismo valor para cada canal (salvo ruido y salvo aquellos canales en que aparece la raya espectral), sino que muestra una tendencia a crecer, decrecer o formas más complejas todavía.

Por ello a los espectros se les debe deducir una línea de base, que haga que tal tendencia desaparezca y el espectro (salvo la raya) sea plano y con valor medio nulo. Como función para ajustar tales valores se suele utilizar un polinomio cuyo grado (GRADO) depende de lo compleja que sea la forma del espectro y debe ser escogido por la persona que lo está analizando. Asimismo debe escoger qué intervalos (INTERV) evita en el ajuste (aquellos que contengan la o las rayas). En ANAE se utilizan polinomios de Chebyshev.

El grado del polinomio puede escogerse entre 0 y 7, y se coloca en la variable GRADO. Ejemplo típico:

GRADO=3

El o los intervalos a evitar en el ajuste se escogen de manera que se evite la raya espectral. Hay que tener en cuenta que la raya puede tener alas bastante prolongadas o tener una forma compleja, con varias componentes. No se puede dar una receta de cómo escoger el mejor intervalo: se aprende con la experiencia.

Ejemplos:

INTERV 50-83

INTE 30-40 50-65

Para definir bien el intervalo es útil la instrucción CURSOR descrita en el apartado 4.4.

BASE

La instrucción BASE efectúa el ajuste y, si GRAF=17, realiza la gráfica de la línea de base en la pantalla, sobreimpuesta a la gráfica precedente. Por defecto, ajusta una línea de base al vector

indicado con el puntero:

BASE

pero es posible ajustarla a otro, p. ej.:

BASE 2

Atención: BASE no resta la línea de base al espectro, sólo la calcula.

La línea de base calculada es almacenada en el vector 4. Su gráfica por separado (no superpuesta a la anterior) puede obtenerse mediante:

GRAF 4

RESI

Esta instrucción efectúa la resta del espectro y la línea de base que se la ha ajustado. (Es decir, resta del vector señalado por el puntero el contenido del vector 4.)

Ejemplo 1:

Ajuste de una línea de base a un espectro recién leído:

```
READ 471
GRAF
CURS 33
CURS 55
GRAD=2
INTE 35-58
BASE
RESI
GRAF
```

Ejemplo 2:

Ajuste de una línea de base al espectro acumulado tras la integración de varios:

```
SCAN 1-10
DO 10
  READ
  ACUM
END DO
GRAF
```

```

GRAD=3
INTE 60-90
BASE
RESI
GRAF

```

Obsérvese que la línea de base de grado 3 no pareció buena y se ajustó a continuación una de grado 4, que se consideró adecuada y fue restada al espectro inicial.

Ejemplo 3:

Véase el ejemplo correspondiente al comando PESO en el apartado 4.5.

SIGMA

Permite conocer la bondad del ajuste (o sea, la dispersión de los valores originales respecto de la función ajustada en los intervalos donde tal ajuste se ha llevado a cabo). Tal valor es escrito en la pantalla.

No hay por qué comandar tal instrucción inmediatamente después de realizar el ajuste, pues el valor de "sigma" es almacenado en la cabecera del vector cuando se ejecuta BASE.

Si el vector consta de dos vías, aparecerá el valor correspondiente a cada una de ellas. Por defecto SIGMA actúa sobre el vector señalado por el puntero, pero se admiten también:

```

SIGMA 1
etc.

```

La respuesta es:

```

sigma ajustada en via 1 = ....
sigma ajustada en via 2 = ....

```

Si no ha sido ajustada ninguna línea de base a tal espectro, es calculada la dispersión de los valores del espectro respecto de su valor medio y aparece el mensaje:

```

sigma sin ajuste en via 1: ....
sigma sin ajuste en via 2: ....

```

Los valores ajustados o calculados son almacenados en las

variables SIG1 y SIG2 cuando se comanda SIGMA. Ello permite utilizar estos valores en cálculos posteriores.

Valores nulos

En ocasiones el valor del espectro para ciertos canales es claramente erróneo. Las causas pueden ser varias: un filtro que no funciona bien (autooscila o se ha muerto), una interferencia, deformación del espectro en los primeros o en los últimos canales del banco de filtros, ...

En tal caso es conveniente eliminar el o los canales que tienen valores anormales. Para ello se utilizan las instrucciones DUM1 y DUM2, que actúan sobre la vía 1 y la 2 respectivamente.

Los comandos son de la forma:

DUM1 1-15 33 55

indicando que queremos que sean anulados los canales del 1 al 15, el 33 y el 55 en el espectro de la vía 1.

Observación: Al decir anulado no se quiere indicar que se les asocie un valor cero. Se quiere decir que no van a ser tenidos en cuenta en ningún caso (ni al realizar gráficas, ni en los ajustes, ...). Si se lista el espectro (mediante PRINT) se observará que se les ha asignado un valor completamente anormal (-33333.).

CLIP

Véase el apartado 4.8.

4.7 Ajuste de gaussianas

Ciertos tipos de rayas espectrales (como, p. ej., las rayas de recombinación a frecuencias radio) tienen una forma tal que no resulta descabellado deducir sus características a partir de su ajuste con una gaussiana. Si su forma es más compleja, quizás sirva el ajustar varias gaussianas simultáneamente.

Una gaussiana la consideramos definida por su altura (o flujo

máximo), su anchura a media altura (expresada generalmente en km/s) y por la velocidad (posición) que corresponde a su pico (en km/s).

En ANAE la función GAUSS efectúa un ajuste por aproximaciones sucesivas de 1 hasta 4 gaussianas al espectro contenido en el vector 3, en las vías definidas por la variable VIA.

MODELO

Una gaussiana no es una función lineal, luego el procedimiento de ajuste es completamente distinto al caso de funciones polinómicas. En el caso de una gaussiana hay que partir de un modelo inicial, que se define desde ANAE con la instrucción:

MODELO

Se pide, para las vías definidas por VIA, información relativa a la altura, anchura y posición de las gaussianas que deseemos ajustar. Cada parámetro introducido debe ir acompañado de una bandera (que toma los valores 0 y 1) para indicar si el parámetro que la antecede debe ser reajustado (bandera = 0) o si debe ser tomado como un valor fijo y definitivo (bandera = 1).

Ejemplo:

```
>MODELO
Via 1
Linea 1: 0.12 0 1.0 0 -12.4 1
Linea 2: 'return'
```

Ello indica que sólo vamos a ajustar una raya espectral y que para ella estimamos que la altura de su pico es del orden de 0.12 unidades y su anchura a media altura es del orden de 1 km/s. Sabemos además que su velocidad (o sea, la posición del pico) es exactamente -12.4 km/s, valor que no va a ser ajustado. Obsérvese que sólo ha pedido datos para la vía 1: ello es consecuencia de que VIA=1.

Para anular un modelo preexistente hacer (caso VIA=0):

```
MODELO
Via 1
Linea 1: 'return'
Via 2
```

Linea 2: 'return'

La utilidad de tal anulación se verá más abajo.

RANGO

Permite definir en qué rango de velocidades deseamos que sea ajustada la(s) gaussiana(s). Inicialmente se toma todo el rango disponible de velocidades.

Ejemplo:

Nos interesa ajustar solamente desde -100 km/s a +200 km/s:

RANGO -100 200

GAUSS

Si deseamos ahora ajustar todo el rango posible:

RANGO

GAUSS

GAUSS

Lleva a cabo el ajuste de la(s) gaussiana(s). Sólo actúa sobre el vector 3. Supone que al espectro se le ha deducido antes una línea de base.

Al mismo tiempo, coloca la gaussiana (o suma de gaussianas) en el vector 4 y coloca en el vector 5 el residuo del ajuste (vector 3 - vector 4).

No realiza la gráfica de la gaussiana sobrepuesta a la gráfica del espectro. Pero escribe los parámetros calculados para cada gaussiana y para cada día en el terminal (o en la impresora, si LL=6).

Si no se ha definido RANGO alguno, ajusta la(s) gaussiana(s) a todo el espectro.

Si no se ha definido MODELO, o si se ha anulado el modelo utilizado en una ocasión precedente (tal como hemos visto antes), el comando GAUSS ajusta una gaussiana a cada vía accesible (según el valor de VIA) y con un modelo inicial que se genera automáticamente y que en general es una buena aproximación.

Ejemplo 1:

Leemos un espectro, lo colocamos en el vector 3 para poder ajustar gaussianas y no definimos modelo inicial, con lo cual es ajustada una única gaussiana:

```
READ 31
MOVE 3
GRAF 3
GAUS
```

Si además queremos visualizar el residuo del ajuste:

```
GRAF 5
```

Ejemplo 2:

Leemos un espectro, lo desplazamos al vector 3, deducimos una línea de base y ajustamos una gaussiana. Como sólo tiene una vía, ponemos VIA=1.

```
READ 31
31 identificador solo via 1
VIA=1
ACUM
Inicio de integración
GRAF
GRAD=2
INTE 100-132
BASE
RES1
GRAF
GAUS
```

Queremos ahora dibujar en el plotter superpuestas la raya espectral y la gaussiana ajustada. Debemos para ello definir una escala:

```
YI=-.05
YF=.5
GRAF
PLOT
PLOT=PLOT-1
PLOT 4
```

Ejemplo 3:

Queremos ajustar una gaussiana a cada uno de los espectros almacenados en el fichero de datos RRESS y listar los resultados en

la impresora. Los espectros sólo disponen de vía 1.

```

RANGO
VIA=1
LL=6
DATOS RRESS
SCAN 777-789
DO 100
  READ
  IF ICODE EQ OFF GO TO 10
  MOVE 3
  GAUSS
END DO
10: LL=LU

```

La variable OFF equivale a 0. Existe también la variable ON, cuyo valor es -1.

4.8 Suavizado y otras alteraciones del espectro

ANAE permite efectuar distintas operaciones con espectros, tales como comprimirlo, suavizarlo, invertirlo, restar espectros, y otras.

Pasamos a describir a continuación algunas de ellas.

HANN

Realiza un suavizado 'hanning' del espectro indicado con el puntero:

```
HANN
```

Si se desea suavizar el espectro contenido en otro vector, por ejemplo el 5:

```
HANN 5
```

Un tal suavizado reduce el ruido (sigma) aproximadamente al 65%.

COMPRI

Comprime un espectro de manera que la resolución en velocidades disminuye en un factor 2. La disminución del ruido (sigma) se reduce también al 65%.

Esta instrucción ya ha sido descrita en parte en el apartado 4.6. Es más potente que lo descrito en aquel lugar, pues permite restar dos espectros, en la forma:

RESID vector destino, vector minuendo, vector sustraendo

Ejemplo:

Queremos comparar el espectro que hemos acumulado en 3 con el espectro de referencia que se encuentra en 2. Ponemos la diferencia en el vector 5:

RESID 5 3 2
GRAF 5

INVER

Invierte o gira un espectro alrededor del canal donde se encuentra la raya. Es decir, es una inversión en velocidades.

Se utiliza para dar la vuelta a un espectro que ha sido observado por una banda lateral distinta a la supuesta para su cálculo, con lo que ha quedado al girado respecto de su frecuencia central.

CLEAR

Borra un vector.

Para que no pueda haber errores accidentales, este comando NO actúa sobre el vector indicado por el puntero: es imprescindible dar el vector sobre el que debe actuar.

Ejemplo:

CLEA 3 borra el vector 3
CLEA no es aceptado

4.9 Funciones matemáticas

Es posible efectuar una serie de operaciones matemáticas sobre

un espectro.

- a) SUMA: suma una constante o el contenido de una variable al vector indicado con el puntero.
- b) RESTA: resta id. id. id.
- c) MULTI: multiplica un vector por una constante o el contenido de una variable.
- d) DIVIDE: divide id. id. id.

Ejemplo:

Queremos cambiar la escala del vector 3 de Kelvins a Janskys, por lo que debemos multiplicar los valores del espectro por 130.3:

```
PUNT=3
MULT 130.3
```

Sobre estas funciones actúa el comando RECUP.

Es posible realizar también operaciones entre variables. Se admiten solamente operaciones binarias (entre dos variables, entre dos constantes, entre una variable y una constante). Se indican como en Fortran (rige el convenio típico en Fortran de considerar enteras las variables desde I hasta N y reales las demás), aunque las operaciones se efectúan siempre como números reales.

Ejemplos:

```
A=B+C
I=K-N
S=SIG1*SIG1
P=1/S
```

Algunas instrucciones extraen información del espectro contenido en un vector. Tales son:

AREA

Calcula el área del espectro señalado por el puntero y otro vector cualquiera.

Ejemplos:

```
AREA
```

AREA 4

Los valores del área correspondientes a las vías 1 y 2 son almacenadas en las variables ARE1 y ARE2, a fin de poder ser utilizados en cálculos posteriores. No van a ser alteradas asta el próximo comando AREA.

MAX

Calcula el valor máximo de los espectros contenidos en el vector señalado con el puntero u otro vector cualquiera.

Ejemplos:

```
MAX
MAX 2
```

Los valores máximos calculados son almacenados en las variables MAX1 y MAX2 (correspondientes a las vías 1 y 2 respectivamente) para su posterior utilización por el usuario.

SIGMA

Ya ha sido descrita en el apartado 4.6.

Almacene los valores de la dispersión, calculados directamente en un ajuste de línea de base, en las variables SIG1 y SIG2.

Además calcula los valores medios, que son impresos pero no almacenados.

4.10 Ficheros de comandos

A menudo el análisis de datos incluye el uso repetitivo de series de comandos. Para facilitar, aligerar y acelerar el proceso de análisis en tales casos, hemos dotado a ANAE de la posibilidad de que le control pase de las manos del usuario a un fichero editado con EDIT. Un tal fichero debe contener una serie de comandos cuya ejecución no tiene por que ser lineal. Es decir, puede incluir iteraciones (DO, ENDDO), tomar decisiones (IF),

bifurcarse (GOTO), e incluso devolver momentáneamente el control al programa principal (PAUSE) para que el usuario pueda tomar alguna decisión.

Dentro de un mismo fichero puede haber varios grupos de comandos que efectúen acciones diversas. Se puede acceder a ellos por separado.

Por otra parte, si el usuario observa que la ejecución del fichero de comandos no va por buen camino, siempre puede recuperar el control del programa, con sólo pulsar:

```
*BR,ANAE
```

o bien:

```
07>BR,ANA07
```

según en qué terminal se encuentre.

Para editar un nuevo fichero de comandos o corregir uno ya existente es necesario salirse de ANAE, lo que puede hacerse de dos formas:

- despidiéndose de ANAE de manera definitiva (END);
- despidiéndose de manera provisional (PAUSE).

Para mayor información véase el apartado 4.11.

Se supone, en lo que sigue, que el lector está familiarizado con lo descrito en el apartado 3.2.

A) Para pasar el control de ANAE a un fichero de comandos caben dos posibilidades:

- si en el fichero se efectúa un solo tipo de acción, se utiliza el comando GO.

Ejemplo: para poner en marcha el fichero COMCOM se comanda:

```
GO COMCOM
```

- si en el fichero se encuentran varios grupos de comandos con acciones distintas, primero es necesario abrir el fichero y a continuación pasar el control a una sentencia cuya etiqueta sea la deseada.

Ejemplo:

Consideremos el siguiente fichero, de nombre COMCOM, que tiene dos partes. En la primera se ajustan líneas de base y en la segunda se dibujan gráficas con escala fija.

```

BAS: GRAD=4
      BASE
      RESI
      GRAF
      PAUSE
GRF: YI=-0.1
      YF=1
      GRAF
      YI=0
      YF=0
      PAUSE
      CLOSE

```

Para abrir el fichero, o sea hacerlo accesible al usuario, se comanda:

```
OPEN COMCOM
```

Si se desea ajustar una línea de base de grado 4 al vector señalada por el puntero, definiremos un intervalo y a continuación transferiremos el control a la sentencia etiquetada con BAS:

```

READ 44
GRAF
INTE 60-90
GOTO BAS

```

Al llegar a la sentencia PAUSE el control es devuelto al usuario sin que el fichero sea cerrado. Podemos, por lo tanto, comandar este fichero cuantas veces se desee sin necesidad de abrirlo de nuevo. Para realizar una gráfica con escala fija, pero dejándola libre a continuación, basta dar:

```
GOTO GRF
```

Cuando ya no se necesita más el fichero de comandos, se cierra mediante:

```
CLOSE
```

B) Para que el control pase de un fichero de comandos al usuario caben tres posibilidades:

- que haya escrito en tal fichero una sentencia PAUSE: en este caso el usuario puede dar todos los comandos que desee y después puede reanudar la acción del fichero en el mismo punto donde se encontraba, mediante:

GO

o bien en una sentencia etiquetada, mediante:

GOTO etiqueta

- que encuentre una sentencia CLOSE: en este caso el fichero de comandos es cerrado y el usuario tiene el control de ANAE. Para devolver el control al fichero debe abrirlo de nuevo mediante:

OPEN fichero de comandos

o bien:

GO fichero de comandos

- que el usuario interrumpa la acción del fichero mediante un comando 'break'. Para introducir tal comando (del sistema operativo) se pulsa cualquier tecla, con lo que aparecerá en la pantalla el símbolo 'x' o bien '07>'. A continuación se escribe 'ER,' seguido del nombre del programa:

*ER,ANAE

o bien:

07>ER,ANAE07

Aparece en pantalla el símbolo '>' y se permite al usuario introducir un único comando antes de que el fichero de comandos retome el control de ANAE. Si este comando es PAUSE o CLOSE, el control pasa definitivamente al usuario, en el segundo caso cerrando el fichero de comandos. En el primer caso, es posible introducir diversos comandos y, a continuación, devolver el control al fichero mediante GO.

C) Bucles: La combinación de dos comandos (DO y ENDDO) permite (sólo dentro de un fichero de comandos) realizar una serie de cálculos u operaciones repetitivos.

La estructura de un bucle es:

DO k

.
.
.

END DO

donde el bucle se inicia en DO y acaba en ENDDO y se repite 'k' veces, donde k puede ser una constante o una variable entera.

Puede encontrarse diversos ejemplos del uso de DO en los apartados: 4.2, 4.5, 4.6, 4.7 y 4.12. Como puede apreciarse en tales ejemplos, no tiene por que realizarse necesariamente k veces el bucle.

No puede haber bucles encajados.

Hay una variable asociada al bucle: IDO. Su valor inicial viene dado por la variable o constante que sigue a DO. Cada vez que se llega al fin del bucle (ENDDO), IDO es decrementado en 1, hasta que llega a tomar el valor 0, momento en el que el bucle se dará por terminado.

Ejemplos:

En DO 44 es IDO=44 inicialmente.

En I=3

DO I es IDO=3 inicialmente.

Ejemplo:

```
          GRAF=0
          PESO S
          GRAD=3
          N=1000
          DO N
              READ
              IF ICODE EQ OFF GO TO FIN
              BASE
              RESI
              ACUM
          END DO
FIN:     I=N-IDO
          REM Numero de espectros promediados =
          I?
          PROM
          VIA=1
          GRAD=1
```

```

BASE
RESI
SIGM
GRAF=17
GRAF
CLOSE

```

Para utilizar este fichero el usuario debe definir de antemano DATOS, SCAN e INTE. Como salidas durante la ejecución del fichero se tienen: el número de espectros promediados (para cuyo cálculo se ha usado IDO), el nivel de ruido (sigma) y la gráfica en el monitor del espectro resultante.

D) Decisiones: Existe una instrucción condicional (IF) análoga al IF lógico existente en Fortran. Permite comparar dos variables o constantes, o variable y constante, de varias maneras distintas.

Ejemplos:

```

IF K LT 0 CLOSE
IF A EQ 1 GO TO 33
IF SIG1 GT .5 REM sigma muy grande

```

Si se cumple la condición establecida, es ejecutado el comando que viene a continuación.

Hay 6 posibles comparaciones, expresadas siempre con 2 letras:

```

GT mayor que
GE mayor que o igual a
LT menor que
LE menor que o igual a
EQ igual a
NE distinto de

```

Hay diversos ejemplos del uso de la instrucción IF en varios apartados del presente capítulo 4.

E) GOTO o bien GO TO : Como ya se ha visto, tiene dos tipos de funciones:

- dentro de un fichero de comandos, sirve para pasar el control a la sentencia etiquetada del modo adecuado.

Ejemplo:

```

GO TO 17
13: VIA=2
.
.
.
17: VIA=1
.
.
.
IF K EQ 2 GO TO 13

```

- fuera de un fichero de comandos, sirve para pasar el control de ANAE a una sentencia etiquetada de un fichero de comandos ya abierto.

F) REM : Sirve para mandar un mensaje escrito a la unidad indicada por LL (la consola o la impresora) durante la ejecución de un fichero de comandos.

Ejemplos:

```

REM A =
A?
REM Ruido superior al teorico !!!
REM Numero de espectros integrados =
N?

```

4.11 Otras funciones

Hasta aquí hemos descrito la mayoría de instrucciones de que dispone la versión actual de ANAE. Tal descripción en algunos casos ha sido exhaustiva pero en otros ha sido somera, debiendo el usuario remitirse al capítulo 5 para tener una información más completa, aunque a menudo más compacta.

Vamos a describir a continuación el uso de dos comandos que

afectan a la ejecución del propio programa ANAE.

PAUSE

Dado desde la consola, deja que el control del ordenador pase, temporalmente, al usuario. A partir de este momento el usuario está trabajando con el sistema operativo, desde el cual puede correr otros programas.

Tras dar el comando PAUSE, si se pulsa cualquier tecla, aparecerá en pantalla:

* en la consola maestra
07) en la consola número 7

lo que indica que se está trabajando con el sistema operativo (y no con el File Manager).

Ejemplo:

Se está utilizando un fichero de comandos de nombre COMCOM que es necesario alterar, para lo cual hay que utilizar el editor. Sin embargo, no queremos perder los datos que tenemos en estos momentos en los vectores de ANAE, o no queremos tener que reiniciar y redefinir todo, lo que ocurriría si parásemos ANAE.

Tras el comando PAUSE, ponemos en marcha el editor con:

*RU,EDIT,COMCOM

o bien:

07)RU,EDIT,COMCOM

Observación 1:

Recordar que el asterisco '*' o los caracteres '07)' los escribe el ordenador al pulsar cualquier tecla, por ejemplo la de espaciado, cuando está funcionando el sistema operativo o cuando se interrumpe un programa en marcha.

Observación 2:

Dado que se está fuera del File Manager, no se trabaja con una copia de EDIT sino con el propio programa EDIT. Por consiguiente, si el usuario se encuentra en la consola 7 no podrá utilizar el editor a menos que nadie lo esté utilizando en la consola 1. Y, además, cuando el usuario de la consola 7 esté utilizando de tal manera EDIT, no se podrá editar desde la consola maestra: es decir,

sólo puede haber, en estas circunstancias, un usuario del editor EDIT. (Hay un programa editor de línea, llamado EDITR, mucho menos potente, pero que puede ser usado en su defecto.)

Acabada la corrección de un fichero o la edición de uno nuevo, se devuelve el control a ANAE pulsando una tecla cualquiera y diciéndole que siga (GO), de la siguiente manera:

- en la consola maestra: *GO, ANAE
- en la consola 7: 07>GO, ANA07

END

Para despedirse de ANAE se utiliza el comando END. Antes de pasar el programa, son cerrados todos los ficheros, eliminados los ficheros 'scratch' y borrada la pantalla del monitor.

4.12 Otras variables

Hasta aquí han sido descritas la mayoría de variables (enteras o reales) de que dispone la versión actual de ANAE. En algunos casos se han descrito en conexión con algún comando determinado, con el que está asociada la variable. Sin embargo, hay otras variables accesibles al usuario, muchas de las cuales su única función es que puedan ser utilizadas por el usuario para sus propios cálculos.

- a) variables enteras libres: Son
 I, J, K, L, M, N
siguiendo el convenio habitual en Fortran.
- b) variables reales libres: son
 de la A a la H y de la O a la Z
de acuerdo con el anterior convenio.

Es posible realizar operaciones entre ellas, entre ellas y vectores, o entre ellas y variables de las antes estudiadas (las asociadas a comandos).

Ejemplo:

```

DATOS CABL
SCAN 3945-4038
VIA=1
INTE
GRAD=1
A=0
K=0
DO 100
  READ
  IF ICODE EQ OFF GO TO CONT
  BASE
  SIGMA
  A=A+SIG1
  K=K+1
ENDDO
CONT: A=A/K
      REM Valor medio de sigma:
      A?
      REM Numero de espectros:
      K?
      CLOSE

```

En este ejemplo se muestra un fichero de comandos que lee espectros (de una sola via) de un fichero y les ajusta una linea de base recta (quizás se trate de espectros obtenidos en ON-OFF espacial) a fin de determinar el valor medio del ruido. El simbolo "_" sirve para anular el 'carriage return' y 'line feed' (o sea, evita que el cursor pase al inicio de la siguiente linea). Así, p. ej., una posible salida sería:

```

Valor medio de sigma: .390
Numero de espectros: 32

```

BELL.

Esta variable sirve para indicar si se desea que suene o no un pitido cada vez que solicita un nuevo comando. De entrada no está puesto. Para activar el pitido se comanda:

```
BELL=ON
```

y para desactivarlo:

```
BELL=OFF
```

Una utilidad práctica de tal pitido es la siguiente: Si se deja el control de ANAE a un fichero de comandos de muy larga ejecución, puede interesar que al final avise de que va a terminar haciendo que suenen algunos pitidos. Ello se consigue activando el pitido algunos comandos antes del último y desactivándolo luego, p. ej., justo antes de despedirse del fichero.

Ejemplo:

```

DATO MANIP
CLEA 3
VIA=0
PESO S
SCAN 1-100
INTE 70-95
GRAD=3
GRAF=0
DO 100
    READ
    DUM1 15 16
    DUM2 24 42
    BASE
    RESI
    ACUM
ENDD
BELL=ON
FROM
VIA=1
GRAD=1
BASE
RESI
SIGM
GAUS
GRAF=17
GRAF
BELL=OFF
CLOSE

```

Se leen 100 espectros que, tras serles ajustada y deducida una línea de base de grado 3, son acumulados. Acabada su integración, se activa el pitido, se promedian ambas vías y se ajusta una línea

de base recta a fin de estimar la sigma (el ruido). Es ajustada una gaussiana para determinar algunas propiedades de la raya y se realiza la gráfica del espectro resultante. Antes de despedirse del fichero, se desactiva el pitido. Con ello se habrá logrado que suene el pitido cada vez que se efectúe un comando tras BELL=ON. Los 10 pitidos que van a sonar serán el aviso al usuario de que el análisis está finalizando.

ON y OFF

Estas variables pueden ser utilizadas en cualquier operación. Su valor es:

ON = -1

OFF = 0

Su valor inicial nunca debe ser cambiado.

4.13 Fichero de comandos GRAF15

En este apartado presentamos el listado de un fichero de comandos bastante complejo, como ejemplo completo y acabado de uno de tales ficheros. Recomendamos que quien esté interesado en escribir algún fichero bastante complejo lea antes el presente con detalle. En cualquier caso, tal lectura puede ser un ejercicio interesante para conocer el propio programa ANAE.

Este fichero permite la realización de gráficas en el plotter con la escala deseada, tanto horizontal (expresada en mm por km/s) como vertical (expresada en mm por Kelvin o Jansky).

En el listado que aparece a continuación hay gran profusión de comentarios, que están precedidos por un signo de admiración '!'. Tales comentarios dan cuenta de lo que se pretende hacer con los comandos que hay a continuación o bien, si están en la misma línea de un comando, indican lo hace este comando particular.

Se puede correr este programa con sólo leer un espectro, ajustarlo y definir las variables requeridas (en cualquier caso, de no estar definidas, el programa las va a pedir). A continuación se comanda:

GO GRAF15

Si se quiere ver el listado de las sentencias que van siendo ejecutadas, antes hay que comandar:

LIST=ON

El listado es el siguiente:

```

!
! Fichero de comandos GRAF15
!
!                                     <841207.1754>
! *****
! Realiza una grafica en el plotter con la escala deseada.
!
! Pueden representarse una o dos vias.
! Actua sobre el vector indicado con el puntero.
!
! Se definen tanto la escala horizontal (mm/km/s)
! como la vertical (mm/K o Jy).
!
! Entradas:      XOPL,YOPL  posicion (mm) del origen de la grafica
!                H          escala horizontal (mm por km/s)
!                V          escala vertical (mm por K o Jy)
! El programa utiliza ademas las variables A, B y C.
!
! Puesta en marcha: hay que dar valores a XOPL, YOPL, H y V
! y a continuacion:  GO GRAF15
!
! *****
! !! comprobacion de que han sido dados valores a XOPL y YOPL
!
C:  IF XOPL LE 0 GO TO A      ! debe ser XOPL > 0
    IF YOPL GT 0 GO TO B      ! debe ser YOPL > 0
! caso en que no han sido definidos o son erroneos
A:  REM Definir origen : XOPL,YOPL (mm)
    PAUSE
    GO TO C      ! regreso a C para comprobacion
! !! determinacion de los valores maximo (A) y minimo (-B)
B:  MAX          ! calculo del valor maximo de ambas vias
    A=MAX1

```

```

IF MAX2 GT MAX1 A=MAX2 ! A = maximo de ambas vias
MULT -1 ! se da media vuelta a la grafica
MAX ! calculo del valor "minimo" de ambas vias
B=MAX1
IF MAX2 GT MAX1 B=MAX2 ! -B = minimo de ambas vias
MULT -1 ! se reintegra la grafica a su posicion original
!!! determinacion escala vertical y extremos grafica
C=A+B
DYPL=1.1*C ! tamaño total aumentado un 10%
C=C/20
YI=B+C
YI=0-YI ! extremo inferior (margen del 5%)
YF=A+C ! extremo superior (margen del 5%)
! comprobacion de que ha sido definido V
IF V GT 0 GO TO D ! debe ser V > 0
REM Definir escala vertical: V (mm/K)
PAUSE
!!! tamaño vertical
D: DYPL=DYPL*V ! tamaño vertical en mm
! comprobacion de que la grafica no se sale del papel:
A=Y0PL+DYPL
IF VIA NE 0 GO TO H
! caso en que hay 2 vias (2 graficas)
B=2*A ! el tamaño total es doble
E=B+30 ! y hay que añadir la distancia entre graficas
IF B GT 190 GO TO I ! altura maxima = 190 mm
! caso en que hay una sola via
H: IF A LT 190 GO TO E
! altura excesiva: reentrar datos
I: DYPL=DYPL/V ! tamaño de nuevo en K o Jy
REM Grafica demasiado alta u origen erroneo
REM Cambiar Y0PL o bien V
PAUSE
GO TO D
!!! tamaño horizontal
E: DXPL=299.7925/43122.027 ! 43122 = frecuencia reposo
DXPL=DXPL*50 ! 50 kHz por canal
DXPL=DXPL*128 ! 128 canales
! comprobacion de que ha sido definido H
IF H GT 0 GO TO F ! debe ser H > 0

```

```
REM Definir escala horizontal: H (mm/km/s)
PAUSE
F: DXPL=DXPL*H      ! tamaño horizontal en mm
! comprobación de que la grafica no se sale del papel
A=XOPL+DXPL
IF A LE 257 GO TO G      ! anchura maxima = 257 mm
! grafica demasiado ancha: reentrar datos
DXPL=DXPL/H      ! tamaño de nuevo en km/s
REM Grafica demasiado ancha u origen erroneo
REM Cambiar XOPL o bien H
PAUSE
GO TO F

!!! ejecucion de la grafica
G: GRAF=15      ! unidad 15
GRAF
GRAF=17      ! unidad 17, como precaucion
! puesta a cero de algunas variables usadas
A=0
B=0
C=0
YI=YF      ! eliminacion de la escala fija
CLOSE      ! despedida y cierre
```

5. DESCRIPCION DE TODOS LOS COMANDOS

(Listado completo del fichero de ayudas)

>CLAVES

ACUM	AREA	BASE	CABECERA	CLEAR	CLIP	CLOSE
COMPRI	CURSOR	DATOS	DIVIDE	DO	DUM1	DUM2
END	ENDDO	GAUSS	GO	GOTO	GRAF	HANN
HELP	IF	INTERV	INVER	LI	MAX	MODELO
MOVE	MULTI	OFFSCAN	OPEN	PAUSE	PESO	PLOT
PRINT	PROM	RANGO	READ	RECUP	RELLENA	REM
RESID	RESTA	RESUL	SAVE	SCAN	SHIFT	SIGMA
STATUS	SUMA	TPL				NOP

>VARIABLES

BELL	GRADO	GRAF	ICODE	IDO	LAST	LIST
LL	LU	OFF	ON	PLOT	PUNTERO	VIA
I ... N						
ARE1	ARE2	CLIP	DXPL	DXTV	DYPL	DYTV
MAX1	MAX2	SIG1	SIG2	VS	XOPL	XOTV
YOPL	YOTV	YI	YF			
A ... H	O ... Z					

++

Este es un programa destinado a hacer analisis espectral, interpretando las instrucciones que se le dan desde consola o desde fichero de comandos.

Las posibles ayudas se solicitaran mediante el comando HELP key, en donde <key> es la llave de acceso a las diferentes secciones del fichero de ayudas <"ANAE>. Estos <keys> son:

- CLAVES: Lista de claves definidas
- VARIABLES: Lista de variables definidas
- VECTORES: Uso de vectores
- FICHEROS: Uso de ficheros

Tambien estan definidas como <keys> cada una de las CLAVES, en donde se da una informacion mas detallada.

Si nos hace la pregunta <Continua (SI o NO)?> es porque la seccion solicitada puede ampliar mas la informacion.

++

Una instruccion consta de una linea de caracteres compuesta de tres elementos posibles: LABEL, COMANDO y COMENTARIO.

- LABEL: Es una etiqueta direccionable y solamente tiene sentido en instrucciones desde fichero de comandos. Admite cualquier numero de caracteres alfanumericos, seguido inmediatamente por dos puntos (<:).>

- COMANDO: Se compone de uno o varios grupos de caracteres, de los cuales el primero sera una CLAVE o una VARIABLE definida. Se diferencian unas de otras por el primer caracter que le sigue. Seran CLAVES si dicho caracter es blanco, y seran VARIABLES si es <=> o <?>. Las CLAVES se utilizan para definir un tipo particular de accion y podran ir seguidas de varios parametros, que en algunos casos podran ser va-

riables. Las variables son utilizadas para ver, modificar o referenciar el contenido de dicha variable.

- COMENTARIO: Es el comentario opcional que puede seguir a cualquier comando. Comenzara con el caracter (!) y el programa despreciara todo lo que siga a este signo. Es util solamente para ficheros de comandos.

++

La ejecucion de un fichero de comandos podra detenerse poniendo el BREAK FLAG en el programa, con lo que la siguiente instruccion sera solicitada en la consola. Si esta no cambiara el flujo, la ejecucion continuaria normalmente en el fichero.

Para introducir el BREAK FLAG en el programa, primero tendremos que quitar el control de la consola al programa pulsando una tecla hasta lograr que nos apareciera un (*). Entonces escribimos el comando BR, ANAE y al pulsar RETURN pasara el control al programa y este nos solicitara la instruccion.

""VECTORES

El programa maneja 6 vectores, por los cuales pasan los datos en las diferentes etapas del analisis. Estos vectores son tipicamente:

- 1 - Vector de trabajo. Donde normalmente entraran los datos.
- 2 - Vector de referencia. Posible entrada ocasional de datos.
- 3 - Vector acumulador. Donde se efectuaran las integraciones.
- 4 - Vector linea de base. Donde se genera la linea de base o gaussianas ajustadas.
- 5 - Vector de residuos. Donde quedan colocados los residuos del ajuste con el modelo de gaussianas.
- 6 - Vector de recuperacion. Donde queda almacenado cualquier vector antes de ser modificado por algunos comandos. El comando RECUP esta expresamente creado para rescatar los datos de este vector. Ningun otro comando podra actuar directamente sobre este vector.

""FICHEROS

El programa puede manejar hasta 5 ficheros simultaneamente:

- 71 - Fichero de datos. Se abre con la instruccion DATOS.
- 72 - Fichero de resultados. Se crea o se abre con la instruccion RESUL. En realidad el programa trabaja sobre un fichero tipo 2 llamado .ANA01:LA:3 o .ANA07:LA:32754 ocupando el resto del cartridge. El fichero inicial solo permanecera abierto durante las instrucciones RESUL o CLOSE RESUL.
- 76 - Fichero de entradas. Donde quedan escritas las ultimas instrucciones entradas desde consola mientras dure la sesion. Este es un fichero tipo 2 con un solo registro de 1024 caracteres llamado .INP01:LA o .INP07:LA. LI INPUT listara su contenido.
- 77 - Fichero de comandos. Es un fichero que contiene un juego de instrucciones para su ejecucion. Se abre y se cierra con los comandos OPEN, GO y CLOSE.
- 75 - Cualquier otro fichero que deseemos listar. LI file lista su contenido.

""CLAVES.

Las CLAVES definidas son:

- 45 - ACUM: Integracion del vector 1 en el 3
- 46 - AREA: Calculo del area de un vector
- 21 - BASE: Calculo de linea de base
- 47 - CABECERA: Muestra cabecera de un vector por LL
- 42 - CLEAR: Borra el contenido de un vector
- 27 - CLIP: Hace recorte de canales malos
- 10 - CLOSE: Cerrar el fichero de comandos
- 48 - COMPRI: Comprime un vector en un factor 2
- 28 - CURSOR: Pone un cursor sobre un canal (TV)
- 13 - DATOS: Abrir el fichero de datos
- 39 - DIVIDE: Divide un vector por una constante o variable
- 7 - DO: Comienzo de bucle
- 22 - DUM1: Pone dummies en via 1
- 23 - DUM2: Pone dummies en via 2
- 3 - END: Despedida del programa
- 8 - ENDDO: (o END DO) Fin de bucle
- 31 - GAUSS: Efectua el ajuste de gaussianas
- 5 - GO: Devuelve control al fichero de comandos
- 2 - GOTO: (o GO TO) Salto incondicional
- 19 - GRAF: Efectua salida grafica

++

- 26 - HANN: Hace filtrado Hanning
- 9 - HELP: Solicitud de ayuda
- 6 - IF: Ejecucion condicional
- 20 - INTERV: Define intervalos que no entran en ajuste
- 52 - INVER: Invierte espectro alrededor de la raya
- 11 - LI: Listar un fichero del FMGR o del programa
- 51 - MAX: Calcula el valor maximo de un vector
- 30 - MODELO: Define gaussianas a ajustar
- 32 - MOVE: Traslada el contenido de un vector a otro
- 38 - MULTI: Multiplica un vector por una constante o variable
- 100 - NOP: No ejecuta. Equivale a CLAVE en blanco
- 17 - OFFSCAN: Definicion de identificadores no validos
- 1 - OPEN: Abrir fichero de comandos
- 4 - PAUSE: Deja control temporalmente
- 34 - PESO: Define tipo de peso a usar
- 50 - PLOT: Grafica automaticamente via 1 en plotter
- 24 - PRINT: Lista un vector por LL
- 35 - PROM: Efectua promedio de las vias de un vector
- 29 - RANGO: Define rango para el ajuste de gaussianas
- 15 - READ: Leer del fichero de datos

++

- 40 - RECUP: Recupera ultimo vector modificado
- 41 - RELLENA: Rellena huecos en un vector
- 12 - REM: Escribir un texto en la consola
- 25 - RESID: Resta de dos vectores
- 37 - RESTA: Resta constante o variable de un vector
- 14 - RESULT: Abrir el fichero de resultados
- 43 - SAVE: Salva un vector al fichero de resultados
- 16 - SCAN: Definicion de intervalos de identificadores validos
- 44 - SHIFT: Desplaza un vector un numero de canales
- 33 - SIGMA: Calcula el sigma de un vector
- 18 - STATUS: Muestra estado de grupos de variables

- 36 - SUMA: Suma constante o variable a un vector
 49 - TPL: Escritura de un texto en el plotter

""VARIABLES

Las VARIABLES definidas como enteras son:

- 1 - BELL: Activa campana
 11 - GRADO: Grado maximo de la linea de base
 9 - GRAF: Unidad logica para graficos
 8 - ICODE: Control de ejecucion de READ
 3 - IDO: Variable de control del DO
 7 - LAST: Ultimo identificador leído correctamente
 2 - LIST: Activa listado de entradas
 4 - LL: Unidad logica para listados
 5 - LU: Unidad logica de la consola que se esta usando
 13 - OFF: =0
 12 - ON: =-1
 14 - PLOT: Posicion PLOT
 6 - PUNTERO: Indicador de vector en uso
 10 - VIA: Via que se esta usando (0=ambas)
- 50 I ... N: Variables libres de usuario enteras

++

Las VARIABLES-definidas como reales son:

- 63 - ARE1: Area calculada por AREA en via 1
 64 - ARE2: Area calculada por AREA en via 2
 51 - CLIP: Factor de recorte para CLIP
 57 - DXPL: Dimension X del cuadro para el plotter
 61 - DXTV: Como DXPL para TV
 58 - DYPL: Dimension Y del cuadro para el plotter
 62 - DYTIV: Como DYPL para TV
 67 - MAX1: Valor dado por MAX en via 1
 68 - MAX2: Valor dado por MAX en via 2
 65 - SIG1: Sigma dado por SIGMA en via 1
 66 - SIG2: Sigma dado por SIGMA en via 2
 52 - VS: Velocidad de dibujo con el plotter
 55 - XOPL: Posicion del cuadro en X para el plotter
 59 - XOTV: Como XOPL para TV
 56 - YOPL: Posicion del cuadro en Y para el plotter
 60 - YOTV: Como YOPL para TV
 53 - YI: Limite inferior (abajo) en el cuadro de la grafica
 54 - YF: Limite superior (arriba) en el cuadro de la grafica
- 100 - A ... H, O ... Z: Variables libres de usuario reales

""OPEN

Cierra el fichero de comandos (de estar abierto) y abre el nuevo fichero. El puntero es colocado en la primera instruccion. Si se ejecuta desde un fichero, el control continua en el fichero abierto.

Formato: [label:] OPEN fname

""GOTO

Como excepcion tambien se acepta GO TO.

Salto incondicional a la etiqueta <label> del fichero de comandos abierto. Podra ejecutarse desde consola o desde fichero de comandos. Si no encuentra la etiqueta, el control pasa a la consola y el puntero queda en el EOF del fichero de comandos.

Formato: [label:] GOTO label o [label:] GO TO label

""END

Este comando despide el programa y podra ser ejecutado desde consola o fichero de comandos. Salva el fichero de resultados y purga el el fichero auxiliar de resultados y de entradas.

Formato: [label:] END

""PAUSE

Ejecutado desde consola efectua un PAUSE en el programa. Ejecutado desde fichero de comandos, pasa el control a la consola, quedando el fichero abierto y el puntero en la instruccion siguiente al PAUSE.

Formato: [label:] PAUSE

""GO

Pasa el control al fichero de comandos abierto en la posicion marcada por el puntero o al comienzo del fichero aqui definido.

Formato: [LABEL:] GO [fname]

""IF

Ejecucion condicional de un COMANDO si el resultado de la operacion logica es TRUE. Los operadores de relacion son los mismos que en FORTRAN pero omitiendo los puntos. A cada lado del operador podra haber una variable definida o constante.

Si alguna variable, operador o comando no estan definidos, el control pasa a la consola dejando el puntero en la instruccion siguiente al IF.

Este comando puede ejecutarse tambien desde consola.

Formato: [label:] IF n1 op n2 comando

""DO

Este comando determina el comienzo de un grupo de sentencias que se ejecutaran una o mas veces. El numero de ejecuciones del grupo es determinado por el parametro <n>, que podra ser una variable entera o una constante. La variable IDO es la que toma el valor de <n> y decrementandose controla el DO. El grupo termina con el comando ENDDO. No se admiten DO's encajados.

Formato: [label:] DO n

""ENDDO

Como excepcion tambien se acepta END DO.

Senala el limite del grupo de instrucciones que componen el DO. Es la salida normal del grupo. Actúa como sigue: Decrementa la variable de control (IDO=IDO-1) y si IDO>0 salta a la primera instruccion del grupo. En caso contrario, la ejecucion continua en el comando siguiente.

Esta instruccion puede llevar <label> pero hay que tener cuidado con su uso, ya que un salto a ella solo tiene sentido desde dentro del propio DO.

Formato: [label:] ENDDO o [label:] END DO

""HELP

Formato: HELP [key]

""CLOSE

Este comando cierra el fichero de comandos, el de datos o el de resultados segun el parametro que lleve:

- Si no lleva ningun parametro, cierra el fichero de comandos, pasando el control a la consola. Es la manera natural de terminar la ejecucion de un fichero de comandos.
- Si lleva como parametro DATOS, cierra el fichero de datos.
- Si lleva como parametro RESULT, salva el fichero de resultados y le cierra.

Formato: [label:] CLOSE [parametro]

""LI

Lista un fichero en la unidad definida por LL. Generalmente se usa desde consola, pero puede usarse desde fichero de comandos.

Tiene cinco formas de uso:

- Para listar cualquier fichero-disco (ASCII), el parametro tomara el nombre de dicho fichero.
- Para listar el fichero de entradas, el parametro sera INPUT.
- Para listar el fichero de comandos, no se dara ningun parametro.
- Para listar el fichero de datos, el parametro sera DATOS.
- Para listar el fichero de resultados, el parametro sera RESULT.

En estos dos ultimos casos, solamente listara parte de la cabecera de cada registro (identificador, nombre, frecuencia, fecha, ascension recta y declinacion).

Formato: [label:] LI [parametro]

""REM

Escribe un texto en la unidad LL.

Formato: [label:] REM texto

""DATOS

Abre el fichero de datos, que debera existir previamente. De estar abierto otro fichero de datos, le cierra para abrir el nuevo.

Formato: [label:] DATOS fname

""RESUL

Abre el fichero de resultados. Dice si existe o no, y en este caso pide confirmacion para crearle. Atencion al codigo de seguridad. A continuacion pasa los datos al fichero auxiliar, que es donde el programa trabaja directamente.

De estar abierto previamente el fichero auxiliar de resultados, seran salvados estos datos al fichero de origen, antes de proceder a tomar los datos del nuevo fichero.

Formato: [label:] RESUL fname

""READ

Hace lectura del registro del fichero de datos caracterizado por un determinado numero de identificacion. Podra llevar 0, 1 o 2 parametros:

- Si no lleva ningun parametro, intentara leer el registro cuyo identificador corresponda, siguiendo las tablas establecidas por SCAN y OFFSCAN. Hara PUNTERO=1 y el contenido lo pondra en el vector de trabajo si la lectura se efectua. De no conseguir efectuar la lectura, daria un mensaje, pasando el control a la consola, a no ser que fuera por agotamiento de la tabla SCAN (este caso debera ser previsto a la hora de editar el fichero de comandos).

- Si lleva un solo parametro, y este es la letra F, la funcion es similar a la anterior, pero haciendo PUNTERO=2 y poniendo los datos leidos en el vector de referencia. Si no se produce la lectura actuaria como en el caso anterior.

- Los dos casos anteriores podran llevar un parametro opcional que podra ser una constante o variable definida, siendo su valor el que tomara como identificador a leer. De no encontrar dicho identificador, el control quedaria en la consola.

A la salida de READ podremos saber si la lectura se ha efectuado testeando el valor de ICODE que tomara los valores ON o OFF segun se haya efectuado o no la lectura. Es de especial utilidad en ejecucion desde fichero de comandos para controlar el final de la tabla SCAN. LAST contendra el ultimo identificador leido correctamente.

Formato: [label:] READ [R] [ident]

""SCAN

Genera la tabla de intervalos de identificadores validos.

Cada intervalo viene definido por: identificador inicial, un guion (-) y el identificador final del intervalo (sin ningun blanco entre medias). Un intervalo de un solo identificador podria definirse por un simple numero de identificacion.

El comando SCAN inicializa la tabla OFFSCAN de identificadores no validos.

Tambien puede usarse el comando SCAN para acumular nuevos inter-

valos a los existentes anteriormente, dando como primer parametro la letra (A). En este caso no inicializa la tabla OFFSCAN.

El numero maximo de intervalos que se pueden acumular es de 20, dando un aviso si se trata de sobrepasar.

Formato: [label:] SCAN id1-id2 id3 id4-id5 ... 0,
 [label:] SCAN A id1-id2 id3 id4-id5 ...

""OFFSCAN

Genera la tabla OFFSCAN de identificadores no validos. Los parametros son los identificadores individuales que queremos excluir, almacenandose a continuacion de los definidos anteriormente. Esta tabla puede contener un maximo de 40 identificadores, dando un aviso si se trata de sobrepasar. Se inicializa conjuntamente con el comando SCAN.

Formato: [label:] OFFSCAN id1 id2 id3 ...

""STATUS

Muestra el estado de grupos de variables que afectan a algunos comandos.

Podra llevar como parametro el nombre de un grupo reconocido, segun la siguiente relacion de grupos:

- GRAF: Tamano, dimensiones, limites, etc.
 - SCAN: Tablas SCAN y OFFSCAN y variable LAST.
 - BASE: Variables GRADO, PUNTERO, VIA y tabla INTERV.
 - GAUSS: Modelo y variables propias del ajuste de gaussianas.
 - FICHEROS: Caracteristicas de los ficheros abiertos.
- Si no lleva ningun parametro mostrara todos los grupos.

Formato: [label:] STATUS [grupo]

""GRAF

Hace la grafica de los datos existentes en el vector definido por PUNTERO en la unidad grafica definida por GRAF. Podemos dar uno o dos parametros, que temporalmente tomara como valores de PUNTERO y GRAF respectivamente. Inicialmente GRAF=17.

El tamano, posicion, etc. sera el definido por las variables YI, YF, XOPL, YOPL, DXPL, DYPL, XOTV, YOTV, DXTV y DYTV que, en principio estan inicializadas a 0, menos las particulares de TV que respectivamente valdran 50, 15, 190 y 180. Las unidades se entienden en milímetros para el plotter y unidades de pantalla para el monitor de TV.

YI e YF son los limites de inferior y superior de la grafica en unidades de trabajo. Si YI=YF entonces calcula temporalmente los limites, ajustados al maximo y minimo de los datos con un margen del 10%.

Formato: [label:] GRAF [puntero [graf]]

""INTERV

Genera la tabla de intervalos de puntos que no van a intervenir en el ajuste de base.

Cada intervalo viene definido por: punto inicial, un guion (-) y el punto final del intervalo (sin ningun blanco entre medias). Un intervalo de un solo punto podria definirse por un simple numero de punto.

Tambien puede usarse el comando INTERV para acumular nuevos intervalos a los existentes anteriormente, dando como primer parametro la letra <A>.

El numero maximo de intervalos que se pueden acumular es de 40, dando un aviso si se trata de sobrepasar.

Formato: [label:] INTERV p1-p2 p3 p4-p5 ... 0,
 [label:] INTERV A p1-p2 p3 p4-p5 ...

""BASE

Calcula la linea de base de un vector y grado determinados, sobre la via(s) definida(s) por VIA, y sera puesto en el vector 4. El sigma del ajuste sera puesto en la cabecera del vector ajustado.

Podra llevar 0, 1 o 2 parametros:

- Si no lleva ningun parametro, el vector y grado seran los definidos por PUNTERO y GRADO.

- Si lleva uno o dos parametros (numeros enteros), estos seran el vector y el grado del ajuste ($0 \leq \text{grado} \leq 7$).

Si la ultima grafica (GRAF) ha sido efectuada en la TV, dibujara la linea de base encima de la grafica existente en pantalla. En caso contrario no efectuara la grafica.

Formato: [label:] BASE [vector [gradol]]

""DUM1

Pone dummies en los canales especificados de la via 1 y del vector senalado por PUNTERO. Se pueden definir intervalos poniendo un guion (-) entre los limites.

Formato: [label:] DUM1 c1 c2 c3-c4 c5 ...

""DUM2

Pone dummies en los canales especificados de la via 2 y del vector senalado por PUNTERO. Se pueden definir intervalos poniendo un guion (-) entre los limites.

Formato: [label:] DUM2 c1 c2 c3-c4 c5 ...

""PRINT

Lista los valores de los canales de un vector.

Podra llevar 1 o 2 parametros que seran respectivamente el vector y la unidad logica usada para el listado. Como defecto tomara PUNTERO y LL.

Formato: [label:] PRINT [vector [lu]]

""RESID

Efectua la resta entre dos vectores.

Podra llevar 0, 1, 2 o 3 parametros que respectivamente corresponderan con el vector donde sera situado el resultado, el vector tratado como minuendo, y el vector tratado como substraendo. Por defecto tomara como valores: PUNTERO, PUNTERO y el vector linea de base, respectivamente.

En la resta se haran coincidir los canales que contienen la raya tomando la posicion del vector tratado como minuendo. Cualquier dummy en el minuendo o substraendo producira un dummy en el resultado. El numero de canales ha de ser igual en los vectores afectados.

Formato: [label:] RESID [vres [vminu [vsubs]]]

""HANN

Hace filtrado Hanning de un vector. Puede llevar un parametro, que seria el vector a filtrar. En caso contrario tomaria el vector señalado por PUNTERO.

Formato: [label:] HANN [vector]

""CLIP

Recorta canales del vector PUNTERO que sobrepasen un determinado factor del propio sigma. Puede llevar un parametro, que sera el que tome temporalmente como factor de recorte. De no llevar ningun parametro tomara como factor el contenido inicial de la variable definida CLIP, que inicialmente vale 10.

Formato: [label:] CLIP [factor]

""CURSOR

Pone un cursor sobre un canal, dado como parametro, en las vias dibujadas en la TV. *adulta velocidad*

Formato: [label:] CURSOR [N] canal

""RANGO

Definicion del rango de velocidades (Km/s) que intervendran en el ajuste de gaussianas (GAUSS) o en el calculo del valor maximo (MAX). Si no damos ningun parametro tomara como rango los limites del espectro a ajustar (este es el estado inicial).

Formato: [label:] RANGO [r1 r2]

""MODELO

Definicion del modelo de gaussianas a ajustar.

Los parametros son solicitados por el programa interactivamente, para las vias que se esten usando. Pueden definirse hasta 4 gaussianas por via.

Para cada gaussiana a ajustar hay que dar: Temperatura (K), Velocidad central y Anchura a media potencia (KM/s), con sus banderas para indicar si cada parametro queda libre (0) o no (1) en el ajuste. El orden de los parametros sera: Temp b Vel b Anch b.

Si a una gaussiana se responde con un blanco, conservara los datos definidos anteriormente para esa gaussiana. Si se responde pulsando unicamente RETURN, dara por terminada la entrada de datos para esa via, quedando definido el numero de gaussianas para esa via.

Formato: [label:] MODELO

""GAUSS

Ejecuta el ajuste de gaussianas, segun los parametros previamente definidos en RANGO y MODELO, sobre el vector acumulador (3) y colocando el resultado en el vector linea de base (4). Ademas coloca en el vector residuos (5) el resultado de la resta de los anteriores (3-4).

Por la unidad definida como LL lista los resultados del ajuste.

Formato: [label:] GAUSS

""MOVE

Traslada el contenido de un vector (incluida cabecera) a otro.

Puede llevar 1 o 2 parametros que seran los numeros de los vectores de destino y origen respectivamente. De llevar un solo parametro tomara como vector origen el señalado por PUNTERO.

Formato: [label:] MOVE vdest [vorig]

""SIGMA

Escribe en pantalla el sigma y el valor medio de los datos de un vector.

Podra llevar 1 parametro que sera el numero del vector. Si no lo lleva, tomara el valor de PUNTERO.

Si la cabecera contiene el sigma calculado mediante ajuste, tomara este valor. Si no, lo calculara con respecto al valor medio. Este valor quedara almacenado en las variables definidas SIG1 y SIG2, segun las vias en uso.

Formato: [label:] SIGMA [vector]

""PESO

Determina la modalidad de peso en operaciones de promedio, etc.. Podra llevar 1 o ningun parametro. De no llevar parametro tomara como peso el tiempo de integracion (inicialmente esta asi). De llevar un parametro, este sera la letra S y tomara el peso en funcion del sigma.

Formato: [label:] PESO [S]

""PROM

Efectua el promedio de las dos vias de un vector, por tanto, el vector debera tener las dos vias y se estara trabajando con ambas, es decir VIA=0. Si esta seleccionado el peso en funcion del sigma y no ha habido ajuste previo, tampoco podra efectuar el promedio. Podra tener un parametro opcional que sera el numero del vector. Por defecto tomara PUNTERO.

Formato: [label:] PROM [vector]

""SUMA

Suma una constante o variable definida a un vector o parte.

Llevara de 1 a 4 parametros que seran: la constante o variable, el vector y los puntos inicial y final afectados. Por defecto tomara PUNTERO, 1 y el numero total de puntos.

Formato: [label:] SUMA sumando [vector [pini [pfin]]]

""RESTA

Resta una constante o variable definida de un vector o parte.

Llevara de 1 a 4 parametros que seran: la constante o variable, el vector y los puntos inicial y final afectados. Por defecto tomara PUNTERO, 1 y el numero total de puntos.

Formato: [label:] RESTA substraendo [vector [pini [pfin]]]

""MULTI

Multiplica un vector o parte por una constante o variable definida.

Llevara de 1 a 4 parametros que seran: la constante o variable, el vector y los puntos inicial y final afectados. Por defecto tomara PUNTERO, 1 y el numero total de puntos.

Formato: [label:] MULTI factor [vector [pini [pfin]]]

""DIVIDE

Divide un vector o parte por una constante o variable definida

Llevara de 1 a 4 parametros que seran: la constante o variable, el vector y los puntos inicial y final afectados. Por defecto tomara PUNTERO, 1 y el numero total de puntos.

Formato: [label:] DIVIDE divisor [vector [pini [pfin]]]

""RECUP

Recupera el contenido del vector tal y como estaba antes de la ultima modificacion efectuada por algunos comandos.

El vector sera devuelto a su posicion original, a no ser que demos un parametro, indicando otro vector.

Los comandos que salvan el vector antes de modificarle son: READ, RESID, HANN, CLIP, MOVE, PROM, .SUMA, RESTA, MULTI, DIVIDE, RECUP,

RELLENA, CLEAR, ACUM, COMPRI, INVER y GAUSS.

Formato: [label:] RECUP [vector]

""RELLENA

Rellena los huecos de un vector (maximo 11 puntos consecutivos) con valores obtenidos del ajuste de una recta a los puntos del entorno.

Los puntos tomados a cada lado del hueco son $n/2+1$, siendo n el numero de puntos del hueco. No actua en los extremos del vector. Podra llevar un parametro, que sera el vector. De no llevarlo, tomara el señalado por PUNTERO.

Formato: [label:] RELLENA [vector]

""CLEAR

Borra el contenido de un vector, que es necesario especificar.

Formato: [label:] CLEAR vector

""SAVE

Salva el vector acumulador (3) al fichero de resultados.

Podra llevar 1 o 2 parametros, que seran, el numero de identifi- cacion, y el nombre (hasta 20 caracteres) que deseamos tenga en el fichero de resultados. De faltar algun parametro, tomara el valor actual.

Si el identificador del vector a salvar ya existe en el fichero de resultados, solicitara autorizacion para efectuar la sustitucion. Si no existiese, lo escribiria a continuacion del ultimo.

Formato: [label:] SAVE [identificador] [nombre]

""SHIFT

Desplaza la raya de un espectro un numero de canales especifi- cado como parametro.

Formato: [label:] SHIFT ncanales

""ACUM

Efectua la integracion del vector de trabajo (1) sobre el acumu- lador (3).

La primera integracion genera el formato de este vector (nombre, num. identificacion, num. canales, frecuencias, canal-rama, etc.). Los promedios se efectuaran atendiendo al criterio de peso establecido con el comando PESO.

Para comenzar una integracion es necesario limpiar el vector acumulador mediante el comando CLEAR 3.

Formato: [label:] ACUM

""AREA

Calcula el area limitada por el espectro del vector que demos como parametro o el senalado por PUNTERO, en caso de no darlo.

El calculo del area se hace sobre el vector con los huecos rellenados como lo haria el comando RELLENA, pero no modifica los datos originales del vector.

El resultado, ademas de salir en la consola, queda almacenado en las variables definidas ARE1 y ARE2, segun las vias en uso.

Formato: [label:] AREA [vector]

""CABECERA

Lista por LL el contenido completo de la cabecera del vector que demos como parametro o el senalado por PUNTERO, en caso de no llevarlo.

Formato: [label:] CABECERA [vector]

""COMPRI

Comprime el vector que demos como parametro o el senalado por PUNTERO, en caso de no llevarlo, duplicando la anchura de los canales. Con ello, queda reducido el numero de canales aproximadamente a la mitad.

El metodo utilizado es hacer un suavizado hanning sobre el espectro, quedandonos despues, solamente con los canales pares o impares segun sea el canal de la raya.

Formato: [label:] COMPRI [vector]

""TPL

Rotulacion de textos en el plotter.

Este comando es similar al programa tambien llamado TPL, que se ha adaptado para poder usarle en ANAE. Permite ser usado en modo interactivo, y lo que es mas importante, desde fichero de comandos.

Para usarse en modo interactivo, basta con dar el comando TPL desde consola. El mismo nos dara la informacion necesaria.

Para su uso desde fichero de comandos, tendremos que construir en un fichero una serie de comandos TPL, de los cuales el primero sera TPL, y el ultimo TPL EX. Los comandos intermedios definiran el texto, posicion, etc. Cada comando consta de los caracteres TPL seguidos (tras un blanco) de dos caracteres que identifican el comando y sus respectivos parametros.

Informacion completa podemos encontrar en el ANEXO C del Informe Tecnico CAY 1984-3.

Formato: [label:] TPL [parametro]

""PLOT

Permite hacer la grafica de la via 1 del vector que demos como parametro o el senalado por PUNTERO, de un modo estandar y rapido.

El tamaño de la grafica es fijo (100 x 70 mm). La posición en el papel viene dada por el valor de la variable definida PLOT, que puede tomar los siguientes valores:

- PLOT=1 grafica en la parte superior izquierda del papel
- PLOT=2 " " " inferior " "
- PLOT=3 " " " inferior derecha " "
- PLOT=4 " " " superior " "

Cada vez que se ejecuta, la variable PLOT es incrementada en 1, avisando si es necesario cambiar el papel cada vez que se ejecuta con PLOT=4. Podemos dar valor a la variable PLOT para seleccionar la posición.

La escala vertical viene dada por las variables definidas YI e YF, autoajustandose si YI=YF.

Formato: [label:] PLOT [vector]

""MAX

Calcula el valor máximo de los canales que caen dentro del rango definido por el comando RANGO, y para las vias en uso, del vector dado por PUNTERO, en caso de no darlo como parametro.

Por la consola nos da dicho valor máximo y el canal correspondiente. Además, toman valor las variables definidas MAX1 y MAX2.

Formato: [label:] MAX [vector]

""INVER

Invierte alrededor de la raya los datos del vector que demos como parametro o el señalado por PUNTERO.

Formato: [label:] INVER [vector]

INICIACION AL MANEJO DEL EDITOR EDIT-1000

CONTENIDO:

1. Comunicación con el ordenador
2. Puesta en marcha del programa
3. Nombre del fichero
4. Problemas con la puesta en marcha
5. Modos de trabajo con el editor
6. Editor de pantalla
 - 6.1 Puesta en marcha
 - 6.2 Teclas de edición de pantalla
 - 6.3 Teclas de movimiento en pantalla
 - 6.4 Manejo de páginas
 - 6.5 Uso de las teclas de edición de pantalla
7. Comandos
 - 7.1 Comandos de posicionado
 - 7.2 Comandos de búsqueda
 - 7.3 Comandos de cambio
 - 7.4 Deshacer un cambio
8. Despedida del editor
9. Borrado de un fichero

Este editor permite crear o alterar ficheros escritos en caracteres ASCII (fichero tipo 3 o tipo 4) cuyos registros (líneas) tengan menos de 150 caracteres (1 línea de pantalla tiene 80 caracteres como máximo).

1. Comunicación con el ordenador

El usuario se comunica con el ordenador a través del teclado del terminal en que se encuentre, mandándole instrucciones. Una instrucción cualquiera consta de una sucesión de caracteres que se teclean como en una máquina de escribir y que se da por terminada pulsando la tecla 'return', que está situada a la derecha del teclado.

Para hacer correcciones a lo que se está entrando se pueden usar dos teclas:

- La tecla 'backspace' hace retroceder el cursor un lugar cada vez que es pulsada, dejando invalidados todos los caracteres que queden a su derecha. Una vez llegado a la posición donde se cometió el error, se reescribe tal carácter y todos los siguientes.
- La tecla 'del(ete)' invalida toda la entrada: hay que reintroducirla desde el primer carácter.

2. Puesta en marcha del programa

Se pone en marcha mediante la instrucción:

xRU,EDIT,nombre del fichero

Si el usuario está trabajando desde el File Manager (en cuyo caso aparecen dos puntos, ':', en pantalla cada vez que el ordenador está dispuesto a ejecutar una nueva instrucción), basta dar:

:EDIT,nombre del fichero

Tras la puesta en marcha, aparece en pantalla el indicador:

/ (slash)

como aviso de que el programa está listo para aceptar comandos.

3.Nombre del fichero

Los nombres de los ficheros pueden darse en forma casi completa de la siguiente manera:

nombre del fichero : código de seguridad : número del cartridge

donde:

- el nombre del fichero debe tener como máximo 6 caracteres, el primero de los cuales no puede ser un número. No se admiten los caracteres: '+', '-', ',', y ':'. Si se trata de un fichero que contiene un programa fuente escrito en Fortran se aconseja que el primer carácter sea '&' (ampersand).

Ejemplos: son válidos como nombres para ficheros:

&CAY ABCDEF A123JJ \$%&#! P@GH2 \$A FFT

- el código de seguridad debe estar constituido por dos letras que, a ser posible, permitan identificar al usuario. En caso de tratarse de un fichero para pruebas o de escaso interés se ruega no dar ningún código de seguridad. (De este modo, en caso de necesitarse espacio en el disco, va a poder ser borrado sin contemplaciones.)

Ejemplos: son válidos como códigos de seguridad:

XY P2 \$\$ 99 PP 0

- número del cartridge: tradicionalmente utilizamos los cartridges 15 y 16 para que contengan programas fuente o relocatables (compilados), los cartridges 13 y 14 para que contengan ficheros de datos a analizar o analizados, y el cartridge 2 para programas en absoluto (montados).

Ejemplos de nombres completos de ficheros:

a) Para referirse a un fichero de nombre CAYAC de un usuario cuyo código sea PP y que está o se quiere que esté en el cartridge número 16:

CAYAC:PP:16

b) Para referirse a un fichero de nombre FUTBOL de código)(y cuyo número de cartridge desconocemos, basta dar:

FUTBOL:))

c) Para referirse a un fichero de nombre HAY! que se encuentra en el cartridge 15 sin código de seguridad (o sea, código 0):

HAY!::15

ATENCIÓN: Si no se da el número del cartridge toma por defecto el 2. Un fichero tipo 3 o 4 en el cartridge 2 peligra de ser borrado en cualquier momento.

La puesta en marcha del editor para trabajar con un fichero de nombre AYER, con código de seguridad CS y situado (o que queremos que se sitúe al crearlo) en el cartridge 16, será:

EDIT,AYER:CS:16

4. Problemas en la puesta en marcha

Si se ha puesto en marcha el programa simplemente mediante:

*RU,EDIT o bien :EDIT

puede dársele a continuación el nombre del fichero a editar mediante el comando:

/FI,nombre del fichero

con iguales convenios que antes para el nombre del fichero.

En la puesta en marcha puede aparecer un mensaje de la forma:

"Work file...."

Ello ocurre cuando, estando el programa editor en marcha, hubo un apagón o alguien paró el ordenador. En tal caso hay que preguntar a los demás usuarios presentes si es necesario salvar lo editado. Si no lo es (o no hay a quien preguntar) se borra lo editado mediante la instrucción:

/A (abortar)

y a continuación se llama de nuevo al editor.

Puede también faltar espacio en disco para el fichero auxiliar que el programa EDIT maneja. En este caso el mensaje que aparece es de la forma:

"Out of disk work space"

y el programa es abortado. Pedir ayuda a otro usuario.

5. Modos de trabajo con el editor

Este editor permite trabajar de varias maneras distintas:

- a) mediante comandos, que afectan al fichero como un todo
- b) editando línea a línea (no consideraremos este modo)
- c) trabajando en "modo pantalla", o sea trabajando con varias líneas a la vez que se representan simultáneamente en la pantalla y utilizando el teclado como si se tratara del de una máquina de escribir convencional.

El modo pantalla ("screen mode") es el más cómodo cuando se edita por primera vez un programa, cuando se le añaden comentarios o para escribir textos. Aunque vamos a hablar sobretodo del modo pantalla, se describirán algunos comandos de interés.

Para obtener información adicional:

- el comando HELP o H del propio editor da información muy esquemática (que puede verse en parte listada en la carpeta de información que se encuentra junto a la consola maestra).
- en el Quick Reference Guide (librito azul) puede encontrarse la misma información clasificada por áreas (apartado ED1TR).
- en el volumen 4 del conjunto de manuales del HP-1000 se encuentra la información más completa.

6. Editor de pantalla

6.1 Puesta en marcha

Estando el indicador '/' (slash) en la pantalla, el "modo pantalla" se inicia mediante la introducción de la letra ese ('S' o bien 's') y pulsando a continuación la tecla 'return':

/S (screen)

Tal instrucción hace que aparezcan en pantalla unos indicadores de límite de pantalla editable. Tales indicadores son dos líneas que empiezan con los símbolos '>>' y a continuación una serie de asteriscos 'xxxxxx' o guiones '-----'. Asimismo tal instrucción hace que el ordenador admita la función de varias teclas de edición, que hacen ésta muy cómoda.

Cuando se inicia la edición de un fichero no existente, las líneas de inicio y fin de pantalla editable están juntas:

>>xxxxxx etc

>>----- etc

y es necesario crear hueco entre ellas, pues es en este hueco donde hay que editar (escribir).

ATENCIÓN: NO modificar ni borrar ninguna de estas dos líneas. En caso de modificación o borrado accidentales, es imprescindible reconstruirlas reescribiendo sus 8 primeros caracteres (véase más arriba cuales son éstos).

6.2 Teclas de edición de pantalla

Las teclas de edición de pantalla son 4. En la consola maestra (número 1) se encuentran en la parte superior derecha del teclado:

INSERT LINE DELETE LINE
INSERT CHARACTER DELETE CHARACTER

y no sirven para otra función más que ésta.

En la consola número 7 se utilizan las 4 teclas auxiliares (grises sin rótulo) de la parte superior derecha cuando se habilitan a tal efecto. El efecto de cada una de estas teclas pulsada conjuntamente con una de las teclas 'shift' puede verse en la parte baja de la pantalla. La manera de habilitarlas al modo de

edición depende del estado en que se encuentren:

a) Si la tecla superior derecha indica:

DELETE CHARACTER

están ya definidas del modo adecuado.

b) Si la tecla superior derecha indica:

edit

hay que pulsar simultáneamente la tecla 'shift' y la tecla 'edit', con lo cual aparecerá en su lugar:

DELETE CHARACTER

y nos encontramos en el caso anterior.

c) Si la tecla superior derecha no indica ninguna de las dos funciones antes descritas, pulsar la tecla gris cuadrada que se encuentra por encima del 'return' tantas veces como sea necesario hasta que aparezca o bien:

DELETE CHARACTER

o bien:

edit

con lo que nos encontraremos en uno de los dos casos anteriores.

De este modo habremos logrado que las 4 teclas auxiliares de la parte superior derecha del terminal 7 ejerzan las funciones:

INSERT LINE DELETE LINE INSERT CHAR. DELETE CHAR.

cuando se pulsan simultáneamente con la tecla 'shift'.

6.3 Teclas de movimiento en pantalla

Hay otras 4 teclas de gran utilidad para la edición en pantalla, pues sirven para moverse de una manera cómoda a través de ella. Estas son:

1) en la consola maestra: las teclas de la parte inferior derecha del teclado, indicadas con flechas verticales u horizontales. Su función es obvia. Tienen autorrepetición (es decir, si se mantienen pulsadas más de un par de segundos repiten su función hasta que dejen de ser pulsadas).

2) en la consola 7: son las mismas teclas de la parte superior derecha, pero cuando se pulsan sin 'shift', o sea, es su función propia. Tienen también autorrepetición.

6.4 Manejo de páginas

Vamos a describir los 4 comandos más útiles. Se ejecutan pulsando simultáneamente la tecla de control (designada por CNTL en el terminal 1 por CTRL en el terminal. 7, en ambos casos a la izquierda del teclado) y una letra (P, F, S o Q) y a continuación pulsar 'return'. La letra P, F, S o Q, que se pulsa simultáneamente con la tecla 'control', puede ser minúscula.

Si tales comandos se entran dos veces seguidas, el contenido original de la pantalla no es alterado, es decir no se tienen en cuenta los cambios realizados. (Entrar dos veces control/P significa pulsar simultáneamente 'control' y P, volver a pulsar simultáneamente 'control' y P y a continuación pulsar 'return'.)

La entrada de estos comandos debe hacerse con el cursor dentro de los límites de la pantalla editable.

Vamos a describir brevemente la función de los comandos de interés:

- control/P: va a la pantalla (o página) anterior, leyendo todo lo que se haya escrito o modificado en la presente pantalla.
- control/P control/P: va a la pantalla anterior sin leer la presente (o sea, sin tener en cuenta lo que se haya escrito o modificado en ésta).
- control/F: va a la página siguiente, leyendo la presente.
- control/F control/F: va a la página siguiente sin leer la presente.
- control/S: inicia la siguiente página en la línea donde se encuentra el cursor. El contenido de la presente pantalla es leudo.
- control/S control/S: inicia la siguiente página en la línea donde se encuentra el cursor, sin leer la presente página.
- control/Q: se sale del modo pantalla (es lo que hay que hacer cuando se da por terminada la edición), leyendo lo que hubiera en la presente pantalla.
- control/Q control/Q: se sale del modo pantalla sin leer las modificaciones introducidas en esta última.

ATENCIÓN: Mientras se está ejecutando uno de estos comandos NO

SE DEBE pulsar ninguna tecla: Las consecuencias son imprevisibles.

Una pantalla y la siguiente (o la anterior) tienen dos líneas en común.

6.5 Uso de las teclas de edición de pantalla

INSERT LINE: Al pulsar esta tecla es añadida una línea en blanco justo por encima de la indicada por el cursor.

Pueden ser añadidas varias líneas, pero hay que tener en cuenta que los terminales tienen una memoria limitada a algo menos de dos pantallas (aproximadamente unas 40 líneas de texto). Cuando se han escrito unas 20 líneas de texto en una pantalla es conveniente pasar a la siguiente pantalla (control/P), que quizás estará vacía, a fin de continuar la edición.

DELETE LINE: Al pulsar esta tecla es eliminada enteramente la línea de la pantalla en que se encuentra el cursor. Atención: NO BORRAR las líneas de límite de pantalla.

INSERT CHARACTER: No hay que mantener pulsada esta tecla mientras se insertan caracteres, PERO hay que acordarse de volverla a pulsar (para inhabilitarla) cuando se da por terminada la inserción.

En la consola maestra al pulsar tal tecla se enciende un piloto rojo (inserción habilitada). En la consola 7 aparecen los caracteres 'IC' en la parte baja central de la pantalla.

Al cambiar de pantalla (a la previa o a la siguiente) se apaga el piloto o desaparecen los caracteres IC, es decir: la inserción de caracteres queda inhabilitada.

DELETE CHARACTER: Elimina el carácter sobre el que está situado el cursor, desplazando el resto de la línea hacia la izquierda. Si se mantiene pulsada, borrará los caracteres que sigan.

Para borrar un carácter (o sea, poner en blanco el lugar que ocupa) basta colocar el cursor en este carácter y pulsar la barra de espaciado.

7. Comandos

Los comandos actúan FUERA del modo pantalla.

En este modo, la línea donde uno se encuentra aparece listada por encima del símbolo '/' (slash). Este símbolo indica que el programa está dispuesto a ejecutar el comando que a continuación se escriba.

7.1 Comandos de posicionado

- para posicionarse en la línea 33 basta dar:

/33

- para posicionarse en la última línea:

/\$

- para moverse 12 líneas más abajo:

/+12

- para moverse 15 líneas más atrás:

/-15

- para conocer el número de la línea donde uno se encuentra:

/N

- para mostrar las 10 líneas anteriores y las 10 posteriores a una línea dada, situarse en esta línea y dar:

/W

A continuación se queda posicionado en la misma línea donde se hallaba.

- para listar las 20 líneas que siguen a donde uno se encuentra:

/L

Se queda posicionado en la última línea listada.

7.2 Comandos de búsqueda

- búsqueda de MAMAITA a partir de la línea donde uno se encuentra:

/F/MAMAITA

Cuando la encuentre por primera vez, la búsqueda termina.

- para buscar desde la primera línea:

/1F/MAMAITA

- para seguir buscando lo mismo, basta dar: /F

- para encontrar todas las veces en que MAMAITA aparece en el fichero:

/F/MAMAITA/A (A=all)

7.3 Comandos de cambio

- para cambiar MAMAITA por PAPAITO en la línea donde uno se encuentra, dar:

/G/MAMAITA/PAPAITO

- para realizar este cambio en todas las ocasiones en que MAMAITA aparezca entre las líneas 100 y 200:

/100,200G/MAMAITA/PAPAITO

- para realizar tal cambio en todo el fichero:

/1\$G/MAMAITA/PAPAITO

- si se quiere listar en la pantalla todas aquellas líneas en las que tiene lugar el cambio, utilizar el comando X en lugar de G.

Algunos cambios exigen que se reafirme el comando dado, lo que el editor hace mediante la cuestión:

OK?

a lo que hay que responder: YES o Y, o bien NO o N, según sea el caso.

7.4 Deshacer un cambio

Deshacer un cambio o la acción de cualquier comando ejecutado previamente puede hacerse mediante el simple comando:

/UN

Se utiliza este comando cuando se ha realizado un cambio erróneo.

8. Despedida del editor

Sólo puede hacerse desde FUERA del modo pantalla, naturalmente.

Hay varios comandos posibles:

- abortar dejando el fichero de partida sin cambiar:

/A (abort)

- reemplazar el fichero de partida por el editado:

/ER (end & replace)

- cuando se está escribiendo un nuevo fichero, cuyo nombre se ha dado, el comando ER lo crea.

Si al poner en marcha el editor no se ha dado el código de seguridad (sc) adecuado, hay que darlo a la salida, en la forma:

/ER:sc

(donde en lugar de 'sc' hay que poner el código del usuario).

9. Borrado de un fichero

Un fichero que no tiene ya utilidad o que va a ser repetido entero por contener múltiples errores, puede ser eliminado (purgado).

Ello se lleva a cabo desde el File Manager, o sea desde fuera del editor. El comando a utilizar es:

:PU,nombre del fichero : código : cartridge

En caso de dar algún parámetro erróneo (nombre, código o cartridge) el File Manager se quejará de alguna de las siguientes maneras:

- FMGR-006 indica error en el nombre o en el cartridge
- FMGR-007 indica error en el código de seguridad.

y no se efectuará acción alguna.

<841018.0323>

<841218.1040>

INICIACION AL MANEJO DE FICHEROS
ALMACENADOS EN DISCO MAGNETICO

CONTENIDO:

1. Estructura de los discos
2. Ficheros
 - 2.1 Generalidades
 - 2.2 Tipos de ficheros
 - 2.3 Nombre de un fichero
3. Realización de operaciones con ficheros
4. "File Manager"
 - 4.1 Puesta en marcha
 - 4.2 Estructura de los comandos
 - 4.3 Comandos para manejo de ficheros
 - 4.4 Comandos de información y ayuda
 - 4.5 Comandos relativos a programas
 - 4.6 Comandos de uso extremadamente peligroso
 - 4.7 Pseudocomandos
5. Compilación y carga de un programa en lenguaje FORTRAN

Se pretende dar una información somera sobre el uso de los ficheros en disco magnético y, en particular, sobre la realización de operaciones con ellos desde el programa denominado File Manager.

Para obtener información adicional debe consultarse el volumen 2 del manual del ordenador HP-1000.

1. Estructura de los discos

La unidad de disco actual en el CAY consta de dos discos, uno de los cuales está fijo en la unidad y el otro es cambiabile. Cada uno de ellos tiene una capacidad hábil de casi 10 Mbyte (1 byte = 8 bits).

Cada disco está dividido en varias zonas que denominamos 'cartridges'. A cada una de ellas se accede independientemente, como si se tratara de periféricos distintos. Los cartridges actuales (con el uso que les damos) son:

en el disco móvil:

- # 2 : programas del sistema y programas en absoluto
- # 3 : programas del sistema
- # 32754 : programas del sistema

en el disco fijo:

- # 13 y 14 : ficheros con datos
- # 15 y 16 : ficheros con programas fuente o relocatables

Atención:

El número del cartridge no es igual al de la unidad lógica asociada. Así, por ejemplo, la unidad lógica del plotter tiene el número 15, no teniendo nada que ver con el cartridge 15, cuya unidad lógica es 24.

Los cartridges están divididos en unidades menores, los

'tracks'. Cada track corresponde a 48 bloques. A su vez un bloque corresponde a 128 palabras, siendo una palabra 2 bytes. (Una palabra es la cantidad de memoria que se necesita para almacenar un número entero de precisión sencilla, comprendido entre -32768 y 32767.) En ocasiones se habla de sectores, siendo 1 sector = medio bloque.

Un fichero es un conjunto de información organizada en unidades menores denominadas "registros" (records). Un registro es la menor cantidad de información a la que se puede acceder desde un programa.

Cada cartridge dispone de una amplia zona donde el ordenador almacena los ficheros que crean los usuarios o que él mismo necesita crear al ejecutar ciertos programas. Además tiene un pequeño fichero directorio (que ocupa 1 track) donde asocia el nombre de cada fichero y la posición física donde empieza tal fichero en el cartridge (contado en tracks y bloques).

2. Ficheros

2.1 Generalidades

El usuario del disco trabaja con ficheros, que se identifican con un nombre y, si hay varios con el mismo nombre (lo que no se aconseja) o si se desea acceder a él más rápidamente, con el número del cartridge en que se encuentra.

Un fichero corresponde a una zona de memoria del disco que contiene datos almacenados en un cierto código (en caracteres ASCII o en binario) y al que se accede de una determinada manera (acceso directo o acceso secuencial). Para el ordenador, tal zona viene identificada por número de cartridge, número de track y número de bloque, y tiene un tamaño de una cierta cantidad de bloques.

Al usuario no le interesa la posición exacta de un fichero. Para él un fichero es simplemente una zona del disco con un nombre de hasta 6 caracteres: el ordenador se encarga de encontrar la posición física dentro del cartridge partiendo del nombre dado,

para lo cual cuenta, en cada cartridge, con el correspondiente directorio. Es más, la posición de un fichero en un cartridge puede ser cambiada, sin que sea alterado.

2.2 Tipos de ficheros

Hay ficheros de diversos tipos que van numerados del 1 al 7. Vamos a describir brevemente las características y uso de cada uno de ellos.

Tipo 1 : Son los de acceso más rápido pues sus registros tienen una longitud fija de 128 palabras.

Tipo 2 : Sus registros tienen un tamaño fijo que es definido por el usuario en el momento de crear el fichero.

Tipo 3 : Se trata de ficheros con registros de longitud variable, o sea que puede ser distinta de un registro a otro. Pueden contener datos y programas fuentes, relocatables o absolutos. En cada lectura es transferido un solo registro.

Tipo 4 : Son idénticos a los tipo 3, pero sólo aceptan código ASCII. Los ficheros con programas fuente son de este tipo.

Tipo 5 : Son idénticos a los tipo 3, pero sólo aceptan código binario relocatable. Los ficheros con el resultado de una compilación son de este tipo.

Tipo 6 : Son idénticos a los tipo 3, pero sólo pueden contener un programa en código binario, listo para ser corrido. Son creados con el comando SP del File Manager. (En realidad el FMP accede a ellos como si se tratara de tipo 1, luego su acceso es muy rápido.)

Tipo 7 : Son idénticos a los tipo 3, pero sólo pueden contener código binario absoluto.

También existen los ficheros de tipo 0, utilizados para referirse a periféricos distintos de disco mediante un nombre.

Los ficheros pueden contener programas (fuentes, relocatables o absolutos) o bien datos.

Los datos generalmente se almacenan dentro de un fichero en grupos, constituyendo cada grupo lo que se denomina un "registro". Cada vez que se hace una lectura en un fichero de datos, es leído un registro entero.

En los ficheros de tipos 1 y 2 los registros son de tamaño idéntico. En los de tipo 1 son de 128 palabras y en los de tipo 2, de cualquier número de palabras, pero fijo. Se recomienda usar un múltiplo de 128, pues así la transferencia es más rápida y no se desaprovecha memoria en el disco.

En los ficheros tipo 1 y 2 es fácil, por ser todos los registros del mismo tamaño, contar registros. Es decir, el usuario puede posicionarse en el registro que desea (en el 35, p. ej.) y leer lo que hay en él. Por ello se denominan ficheros de acceso DIRECTO.

En los ficheros tipos 3 y 4 los registros pueden tener distintos tamaños (p. ej., el primer registro puede tener 33 palabras, el segundo 77, el tercero 3, etc.). La única forma de posicionarse en un registro dado es leyendo todos los anteriores. Por ello se denominan ficheros de acceso SECUENCIAL. Por ello mismo, son de lectura más lenta que los de acceso directo.

2.3 Nombre de un fichero

A) En general un fichero se identifica con su nombre. El nombre del fichero debe tener como máximo 6 caracteres, el primero de los cuales no debe ser un número. No se admiten los caracteres: '+', '-', ',', y ':', ni blanco. Si se trata de un fichero que contiene un programa fuente escrito en Fortran se aconseja que el primer carácter sea '&' (ampersand).

Ejemplos: son válidos como nombres para ficheros:

&TRIAL HAY! HELLO A#?!\$ PI3.14 A

Una manera más completa de dar el nombre de un fichero es:

nombre del fichero : código de seguridad .

El código de seguridad o identificación debe estar constituido por 2 letras que, a ser posible, permitan identificar al usuario. Tal código impide la modificación del contenido del fichero si no se ha dado al acceder a él (o sea, al abrirlo). En caso de tratarse de un fichero para pruebas o de escaso interés se ruega no dar ningún código. De este modo, en caso de necesitarse espacio en el disco, podrá ser eliminado sin contemplaciones.

Ejemplos:

RRL:UD &D:PP &LVG:VA

Si se quiere impedir que un fichero sea leído puede darse como código un número negativo. (Sin embargo, tal como está montado el sistema en la actualidad, es posible conocer el código de cualquier fichero y, por lo tanto, leerlo.)

Para una mayor rapidez en la localización del fichero y/o para evitar confusiones dolorosas, el nombre de un fichero se puede dar de una forma todavía más completa mediante:

nombre del fichero : código de seguridad : número del cartridge

Ejemplos:

CISNE:PR:13 JAJAJA:JA:16

B) Para la creación de un fichero de un tamaño determinado se puede dar el nombre completo, que es de la forma:

nombre:código:cartridge:tipo:tamaño fich:tamaño reg

El tamaño de los registros (en palabras) sólo hay que darlo en los ficheros tipo 2. En los demás casos no tiene sentido (y no es tenido en cuenta).

El tamaño del fichero se da en bloques. Para un fichero tipo 2 es:

$$\text{núm bloq} = \text{num reg} * \text{tamaño reg} / 128$$

Si no es un número exacto se redondea al siguiente número entero.

Ejemplos:

1) &BLA:JA:13:4:200

2) &JULIO:::2:200:256

Se trata de un tipo 2 con 100 registros de 256 palabras (2 bloques por registro, lo que implica un total de 200 bloques). No se da código de seguridad ni cartridge. Al no dar el número del cartridge será creado en el primero que esté montado, generalmente el número 2. No es conveniente hacer tal cosa: este cartridge es limpiado a menudo y un fichero sin código va a ser eliminado sin contemplaciones (y no se admitirán reclamaciones!).

3) DATOS:AL:14:1:50

Tiene 50 registros de 128 palabras cada uno de ellos.

4) El usuario XX quiere crear un fichero tipo 4 de nombre DATA en el que quepan 20 registros de 640 palabras (5 bloques). Piensa rellenarlo a posteriori con un programa. Si se tratara de un fichero tipo 2 su tamaño sería:

$$\text{num bloques} = 20 * 640 / 128 = 100$$

Al tratarse de un tipo 4 es:

$$\text{num bloques} = 20 * (640+2) / 128 = 101$$

Luego el nombre completo será:

DATA:XX:14:4:101

(Obsérvese que no hemos dado el tamaño de los registros, pues se trata de un fichero tipo 4.)

3. Realización de operaciones con ficheros

Hay varios modos distintos de acceder a los ficheros. Vamos a describir someramente algunos de ellos.

A) Desde un programa FORTRAN:

Existe una colección de subrutinas en la librería File Management Pakage, conocidas como "llamadas FMP", mediante las cuales se puede acceder a un fichero o a un periférico tratado como si fuera un fichero (p. ej., la unidad de cinta magnética).

Mediante tales subrutinas es posible:

- crear un fichero nuevo
- abrir un fichero ya existente (o sea, hacerlo accesible)
- añadir registros
- cambiar registros
- leer registros
- cerrar un fichero abierto
- purgar (o sea, eliminar) un fichero no abierto
- cambiar de nombre un fichero

Naturalmente esta última acción no afecta al fichero en sí sino que afecta al directorio del cartridge, que es el único lugar donde el nombre del fichero existe como tal.

Puede encontrarse información sobre ellos en el QUICK REFERENCE GUIDE TO HP-1000 (librito azul) y más detallada en el libro de FORTRAN 77.

B) Mediante comandos del File Manager (FMGR).

El File Manager es un programa que admite una serie de comandos introducidos desde el terminal o desde un fichero al que se haya transferido el control. Permite:

- crear y purgar ficheros
- listar un fichero o el directorio de un cartridge
- transferir datos de un fichero a otro
- poner en marcha y detener programas

El File Manager puede actuar sobre ficheros de datos y de programas, sobre periféricos y sobre programas en absoluto. Utiliza muchas de las mismas llamadas FMP que el usuario puede utilizar desde sus programas en Fortran.

Varios usuarios pueden usar el File Manager al mismo tiempo, simplemente haciendo una copia de él para cada terminal. En la consola maestra (número 1) se utiliza una versión llamada FMGR, pero en la consola 07 se utiliza una copia llamada:

FMG07

que, en general, es generada por el propio sistema operativo al poner en marcha el ordenador.

Si por alguna razón tal programa es abortado, se puede regenerar la copia correspondiente al terminal desde la consola maestra.

C) Mediante el editor interactivo EDIT (o una copia de éste) es posible, actuando sobre ficheros de tipo 4 o 3:

- crear ficheros
- unir ficheros
- reordenar el contenido de un fichero

4. File Manager

4.1 Puesta en marcha

Cuando está en marcha aparece en pantalla el símbolo dos puntos, ':'. En caso de no haber en la pantalla ningún símbolo, pulsar la tecla 'return'; si el File Manager está en marcha aparecerá el mensaje:

WAITING FOR INPUT

:

Si no está en marcha (y el terminal no está controlado por ningún programa de algún usuario), al pulsar 'return' aparece en el terminal el símbolo:

* en la consola maestra
07) en la consola 7

y el usuario está trabajando con el sistema operativo. Es necesario poner en marcha el File Manager.

Se suele trabajar con el sistema operativo en pocas ocasiones. Vamos a describir una de ellas. Si se introduce un comando del File Manager y si, mientras se está ejecutando, deseamos interrumpirlo (p. ej., un listado), pulsando una tecla cualquiera (a veces hay que hacerlo insistentemente) aparecerá uno de los símbolos anteriores:

* o bien 07)

y a continuación debemos introducir:

BR,FMGR o bien BR,FMG07

según en qué terminal nos encontremos. (BR = break.)

Otra ocasión en que se utiliza el sistema operativo es la siguiente. Si hemos puesto en marcha un programa bajo el control del File Manager, podemos interrumpirlo mediante el comando de sistema operativo AB (de "abortar"):

*AB o bien 07)AB

El programa es abortado y el control vuelve al File Manager. (Si no había tal programa, AB actúa sobre el File Manager de igual manera que el comando BR.)

Para salirse del control del File Manager hay que comandar:

OF,FMGR o bien OF,FMG07

según en qué consola nos encontremos. Atención : NO ES CONVENIENTE hacer tal cosa cuando el File Manager está ejecutando alguna acción.

Para poner en marcha el File Manager se comanda:

RU,FMGR

si estamos en la consola principal. Desde la consola 7 no se puede hacer una copia (FMG07): hay que pedir que la haga quien se encuentra en la consola principal, simplemente comandando:

RP,FMG07

4.2 Estructura de los comandos

En el caso más general, un comando consta de 3 partes:

- 1) 2 letras indican la acción a realizar
- 2) tras una coma, se le da el tipo de información que necesita conocer el comando para actuar: puede tratarse del número de una unidad lógica, el nombre de un fichero, el número de un cartridge, el nombre de un programa, ...
- 3) tras otra coma, se le introducen uno o varios parámetros que, en la mayoría de los casos son opcionales. Si son necesarios y no se le dan, el File Manager toma algunos de ellos por defecto.

Ejemplos:

LI,&TEST,S,1,30

listará en la pantalla en código ASCII (S) las líneas de la 1 a la 30 del primer fichero que encuentre que tenga el nombre &TEST. En cambio:

LI,&TEST::15

listará en ASCII (si &TEST es de tipo 4) todo el fichero &TEST si se encuentra en el cartridge 15.

4.3 Comandos para el manejo de ficheros

Vamos a dar una breve descripción de los más utilizados. Información más detallada sobre éstos y sobre los demás existentes puede encontrarse en el volumen 2 del manual del ordenador HP-1000.

CR, fichero

crea un fichero en disco si su tipo y su tamaño son mayores que cero. Es necesario dar en este caso el nombre completo del fichero, tal como se ha descrito en el apartado 2.3B. Recordar que si es un fichero de tipo 2 hay que dar la longitud de los registros.

Ejemplo: CR,&TABU:HH:16:4:100

DL, cartridge

lista el contenido del directorio del cartridge indicado. Un usuario cuyo código es XX puede listar los nombres de todos los ficheros que posee comandando:

DL,-----:XX

S

LI, fichero, B, L1, L2

D

lista desde el registro L1 al registro L2 del fichero indicado en ASCII (S) o en binario (B), o sólo lista su directorio (D). Para listar todo el fichero basta dar simplemente:

LI, fichero

y el File Manager escogerá el modo en función del tipo de fichero.

Ejemplos: LI,ESMIO,B,3,5

LI,&TABU

LI,&TABU,D

LL, unidad lógica o nombre del fichero

si se da un número, cambia la unidad lógica para listar. La

unidad lógica de la impresora es el número 6. Acordarse de, a continuación, volver a dar como unidad lógica la correspondiente a la pantalla, pues de lo contrario el siguiente listado volverá a salir en la unidad antes escogida. Para ello dar LL,1 o bien LL,7, según en qué unidad se encuentre el usuario. (O bien, dar en general LL,0G.)

```
Ejemplo:  LL,6
           LI,&TABU
           LL,0G
```

Si se da el nombre de un fichero, todos los listados que se hagan a continuación van a hacerse en este fichero, mientras no se comande lo contrario. Tal fichero debe haber sido creado previamente al comando LL.

```
Ejemplo:  CR,LISTR:13:4:200
           LL,LISTR
           LI,&TABU
           LL,0G
```

PK,cartridge

empaqueta un cartridge. Es decir, traslada el contenido de los ficheros existentes en el cartridge de manera que no queden zonas entre ellos sin ocupar. De esta manera se consigue que el área libre al final del cartridge sea la mayor posible. Es conveniente realizar tal acción cuando han sido purgados varios ficheros. (Se recomienda purgar de vez en cuando los ficheros relocatables que no se están usando normalmente.).

Para empaquetar es imprescindible que no haya ningún fichero abierto en el cartridge.

```
Ejemplos:  PK,15
           PK   empaquetará todos los cartridges
```

PU,fichero

elimina el fichero. Es imprescindible dar el código de seguridad del fichero, si lo tiene (de ahí su nombre).

```
Ejemplo:  PU,%TABU
           PU,&TABU:HH
```

RN,fichero,nuevo nombre

permite cambiar de nombre un fichero. El nuevo nombre no debe estar repetido en el mismo cartridge. Si el fichero tiene código de seguridad, es necesario introducirlo en el nombre del fichero.

Ejemplo: RN,&TABU:HH,&TUBA

ST,fichero 1,fichero 2

permite almacenar el contenido del primer fichero (1) en el segundo (2), siendo éste creado en tal acción. Con ello se puede cambiar un fichero de cartridge, reducir el número de extensiones, cambiar el código de seguridad.

Ejemplos: a) Queremos trasladar el fichero &TABU que se encuentra en el cartridge 13 al cartridge 16, cambiando, además, su código de seguridad HH por RT y a continuación purgar el que se encuentra en el cartridge 13:

ST,&TABU,&TABU:RT:16

PU,&TABU:HH:13

b) El fichero ESMIO:HH:15 fue creado inicialmente con un tamaño de 5 bloques, pero paulatinamente ha ido creciendo a base de extensiones que genera el File Manager. Actualmente ocupa 18 veces 5 bloques. Queremos crear un nuevo fichero con el mismo nombre y en el mismo cartridge con el mismo contenido pero que no tenga ninguna extensión. Deberemos crearlo pues con un tamaño de 90 bloques. Dado que no puede haber dos ficheros con el mismo nombre simultáneamente en el mismo cartridge deberemos utilizar un nombre auxiliar para uno de los ficheros. Una posible secuencia sería la siguiente:

ST,ESMIO,XXXXX:HH:15:4:90

PU,ESMIO:HH:15

RN,XXXXX:HH:15,ESMIO

y quizás: PK,15

ATENCION !!! : Si se traslada un fichero cuyos registros (algunos o todos) tienen más de 128 palabras van a ser RECORTADOS a esta longitud, con lo que se perderá parte de la información contenida en el fichero inicial.

Para trasladar ficheros cuyos registros superen las 128 palabras, hay que usar un comando más complejo, pero análogo en todo lo demás:

FC,CO,fichero 1,fichero 2

4.4 Comandos de información o ayuda

L

este corto comando muestra en pantalla los comandos que han sido entrados en el File Manager. Cuando un mismo comando ha sido entrado varias veces, sólo aparecerá una vez: la última en que se ha utilizado.

En caso de haber habido un error en la utilización del File Manager, lo que vendrá indicado en la pantalla con un mensaje de la forma:

FMGR-006

se puede solicitar más información de dos maneras distintas:

??

da una información breve. En el ejemplo anterior, obtendríamos:

FMGR-006 FILE NOT FOUND

HE,FMGR-006

es una manera de solicitar una explicación más amplia que la anterior relativa al error, p.ej., 006. Obtendríamos, en este caso:

```
FILE NOT FOUND.
ATTEMPT TO ACCES A FILE THAT
CANNOT BE FOUND. CHECK THE
FILE NAME OR THE CARTRIDGE REFERENCE.
```

4.5 Comandos relativos a programas

AB

aborta el programa actualmente en proceso

OF, programa o bien DE, programa

eliminan del disco un programa temporal. Es necesario dar este comando cuando ha sido cargado una vez un programa y va a ser cargado de nuevo tras, presumiblemente, ser modificado. El comando OF va a hacer desaparecer la versión ejecutable anterior.

Atención:

Si el usuario no se encuentra en la consola principal (1) NO debe usar tal comando si en la consola principal otro usuario está manejando el editor. Es preciso que en este caso utilice el seudocomando DE:

DE, programa

Quizás sea buena costumbre utilizar siempre DE en lugar de OF.

RU, programa

pone en marcha un programa que se encuentra en un fichero tipo 6 o bien que acaba de ser cargado con LOADR o CLOAD.

El File Manager acepta en muchos casos el denominado "RU implícito", que consiste en dar sólo el nombre del programa cuando éste acaba de ser cargado.

Ejemplo: RU, TABU o bien TABU

SP, fichero

cuando el usuario tiene cargado un programa que funciona bien y desea guardarlo en un fichero como programa absoluto (es decir, que pueda ejecutar directamente, sin necesidad de cargarlo de nuevo) utiliza el comando SP. El nombre del fichero debe coincidir con el del programa.

Ejemplo: Hemos cargado y probado el programa TABU que vamos a tener que utilizar repetidas veces en días sucesivos. Nos interesa, por ello, guardarlo en un fichero tipo 6. Normalmente colocamos tales ficheros en el cartridge 2, especialmente cuando son de interés general. Si queremos darle HH como código de seguridad, comandaremos:

SP,TABU:HH:2

WH

Este comando nos informa de qué programas están funcionando en este momento en el ordenador y cuales van a ponerse en marcha a una hora prefijada.

4.6 Comandos de uso muy PELIGROSO

Se trata de comandos que no hay que usar sin el consentimiento de una o, mejor todavía, varias de las personas que ya los han usado varias veces. Estos son:

CO

DC

DU

IN

El comando CO es especialmente peligroso: NO CONFUNDIRLO con el comando FC,CO que si se puede usar.

Se recomienda asimismo el evitar que los programas que utiliza el usuario empiecen con alguno de los pares de letras antes citados (CO, DC, DU, IN).

4.7 Seudocomandos

Existe una serie de programas de uso muy general adquiridos a Hewlett-Packard o confeccionados por usuarios del CAY que actúan sobre programas o ficheros y a los que les hemos dado una forma tal que parecen actuar como si se tratara de comandos del File Manager.

Entre ellos se encuentran algunos ya citados anteriormente:

FC,CO,fichero 1,fichero 2

que sirve para copiar un fichero en otro, que es creado en esta acción, cuando los registros del fichero de partida tienen más de 128 palabras. Se ha descrito en el apartado 4.3.

DE, programa

elimina un programa que ha sido cargado de manera que no aparezca ningún mensaje en la consola maestra relativa a tal eliminación.

Existe un programa que permite listar el contenido de ficheros ASCII (programas en fuente, ciertos ficheros de datos) en la impresora paginando adecuadamente. Tiene tres opciones: LP,fichero,,L1,L2 o simplemente LP,fichero

lista el fichero indicado (desde la línea L1 hasta la L2 o bien todas) en la impresora, paginando y numerando las sentencias.

LP,fichero,1,L1,L2 o simplemente LP,fichero,1

actúa como el anterior pero sin numerar las sentencias.

LP,fichero,2,L1,L2 o simplemente LP,fichero,2

lista el fichero paginando pero ni numera las sentencias ni las páginas.

Existen asimismo dos comandos que actúan sobre ficheros relativos a observaciones espectrales. Estos son:

FE,fichero

indica la fecha en que se inició la observación y el programa de análisis que es adecuado a los datos en tal fichero contenidos.

HD,fichero

lista la cabecera de los registros contenidos en tal fichero indicando además qué programa de análisis es el adecuado a tales datos.

Se puede encontrar información más amplia relativa a estos dos
seudocomandos en el Informe Técnico CAY 1984-6.

5. Compilación y carga de un programa en lenguaje Fortran

EDIT

Con el programa editor EDIT se generan ficheros de tipo 4 (secuenciales, en código ASCII, con registros de longitud variable) que pueden contener datos, destinados a ser leídos desde algún programa, o bien programas escritos en algún lenguaje. En este caso se habla de "programa fuente".

El lenguaje de programación típico en el campo científico y técnico es el FORTRAN. La versión de que disponemos en el HP-1000 es la denominada FORTRAN 77. Sin embargo, en ciertas ocasiones en que un manejo especial de periféricos, el mejor aprovechamiento de la memoria del ordenador o en que la velocidad de cálculo lo exigen, escribimos los programas en lenguaje ASSEMBLER.

En ambos casos, el programa que se edita tiene que ser compilado y, junto con subrutinas y librerías, montado y cargado en el ordenador, a fin de poder utilizarlo. Por compilar se entiende traducir el programa a lenguaje máquina, codificando las instrucciones a realizar y reservando zonas de memoria donde deberán estar contenidos los valores de las variables del programa.

El montaje del programa ("linkage") consiste en asignar a cada variable e instrucción una posición absoluta dentro de la memoria del ordenador y en la interconexión del programa con las distintas subrutinas del usuario o del sistema que resulten necesarias. A éstas, a su vez, se les asignan también posiciones absolutas en la memoria del ordenador.

La carga consiste en colocar el programa montado en una de las particiones de la memoria del ordenador. Ello tiene lugar cuando se da la instrucción RUN.

Observación: Normalmente nosotros denominamos "carga" al montaje por razones "históricas", pues en los sistemas no basados en disco (como el HP-2100) el montaje tenía lugar dentro de la propia memoria del ordenador con lo que montaje y carga eran simultáneos.

Aún cuando parezca más propia, actualmente, la nomenclatura antes descrita, seguimos manteniendo la antigua, en la que se habla sólo de compilación y carga.

Mediante el editor pueden también escribirse programas utilizables mediante programas interpretadores escritos por los propios usuarios. El ejemplo más claro de ello es el programa de análisis espectral ANAE. Tal programa va leyendo el fichero que contiene las instrucciones a realizar: una vez leída una instrucción, examina su validez y la ejecuta, pasando a la siguiente. Este proceso puede parecer muy lento, pero para el tipo de programas para el que está pensado es suficientemente rápido, muy potente y versátil, y exige unos conocimientos prácticamente nulos de programación de alto nivel.

FTN7X

Una vez escrito un programa en Fortran con el editor, se compila mediante el programa FTN7X. Por ejemplo, si el fichero que contiene el programa fuente se llama &FUENT, la compilación se comanda mediante:

```
FTN7X,&FUENT
```

En pantalla aparece el listado del programa con la localización de los errores que el compilador haya encontrado. (Evidentemente, se trata de errores de escritura, bifurcaciones mal encajadas, utilización incorrecta de bucles o de la memoria. Pero puede haber errores que sólo aparezcan cuando se cargue el programa o cuando se ejecute. Por ejemplo: si el programador ha escrito $A=B+C$ en lugar de $A=B-C$, por supuesto el ordenador no va a detectar tal error.)

Si se desea que, al compilar, no aparezca en la pantalla todo el listado del programa, sino solamente el listado de los errores, debe haberse escrito previamente como primera línea en el fichero fuente la sentencia:

```
FTN
```

Cuando haya habido una compilación sin errores, se puede

compilar de nuevo el fichero almacenando en otro el programa traducido a lenguaje máquina. Si se ha tenido la precaución de denominar el fichero fuente de manera que el primer carácter sea '&' ("ampersand"), basta con comandar, siguiendo con el ejemplo anterior:

FTN7X,&FUENT,,-

donde la pequeña raya es simplemente un signo menos. El fichero relocatable, o sea el resultante de la compilación, se denominará, de forma automática:

%FUENT

Denominamos "relocatable" a tal fichero pues contiene el programa ya dispuesto a ser colocado en la memoria del ordenador, pero las posiciones de memoria que programa y variables tienen asignados empiezan en cero.-En el momento de la carga, programa y variables serán recolocados a fin de que ocupen ciertas posiciones absolutas en la memoria del ordenador.

El fichero &FUENT puede contener un programa y varias subrutinas utilizadas (o no) por él. Es aconsejable hacerlo así, pues si se utilizara un fichero para cada subrutina se llenarían con mucha facilidad los directorios de los cartridges.

Evidentemente:

FTN7X,&FUENT,,-

y FTN7X,&FUENT,,%FUENT

son comandos equivalentes. Es posible, sin embargo, dar un nombre distinto al fichero relocatable. Ejemplo:

FTN7X,&FUENT,,%FUEN2

En cualquier caso, mientras no se indique lo contrario, el fichero relocatable es creado en el mismo cartridge que el fuente, siempre que haya espacio suficiente. Para colocarlo en otro cartridge, por ejemplo el 13, puede hacerse:

FTN7X,&FUENT,,%FUENT::13

LOADR

Es el programa que permite cargar el programa que nosotros deseamos más las subrutinas de librería que necesite. Si no vamos a cargar más que el programa y subrutinas que se encuentran en el fichero, p. ej., %FUENT, aparte de las subrutinas propias del sistema, comandaremos:

LOADR,,%FUENT

Si, por el contrario, además del programa contenido en %FUENT debemos cargar las subrutinas contenidas en el fichero (relocatable !) %CHEBY y parte de la librería contenida en %GRLIB, se procede como sigue:

- se comanda: LOADR

con lo que se pone en marcha este programa y solicita datos.

- Si queremos que no se haga el listado de carga, daremos:

NL

- Para cargar el programa principal, más las subrutinas contenidas en el mismo fichero que aquel:

RE,%FUENT

- Para cargar las subrutinas (todas) contenidas en %CHEBY:

RE,%CHEBY

- Para cargar parte de la librería %GRLIB:

SE,%GRLIB

- Para cargar, finalmente, las subrutinas de sistema:

EX

Cuando LOADR se detenga dará avisos de error si los ha habido. En caso contrario dará el mensaje de que el programa está listo para ser ejecutado (p. ej., "FUENT ready").

Si se ha cargado un programa con un nombre idéntico a otro ya cargado (presumiblemente una versión anterior de él), avisará de tal circunstancia ("duplicate ...") y alterará el nombre del programa, sustituyendo los dos primeros caracteres por puntos.

Ejemplo: si habiendo cargado el programa FUENT, lo ejecutamos y encontramos errores, editaremos el fichero fuente (&FUENT) para corregirlos, compilaremos de nuevo el programa (obteniendo %FUENT) y si lo cargamos sin haber eliminado la versión precedente (con un comando DE o OF), el nuevo programa se denominará: ..ENT.

Para tener mayor información sobre el cargador, consultar el volumen 3 del manual del ordenador.

RU

Para ejecutar (correr) un programa cargado de nombre, p. ej., FUENT, se comanda:

RU,FUENT

o simplemente (RU implícito):

FUENT

si nos encontramos dentro del File Manager. Ahora bien, hay nombres de programas que pueden dar lugar a problemas si utilizamos el RU implícito. Ejemplo: Si vamos a ejecutar un programa de nombre TEST (nombre nada aconsejable por lo que vamos a ver y por que muchos usuarios tienen la tendencia a usarlo) comandaremos:

TEST

En su lugar es ejecutado un comando del File Manager llamado TE. Tal comando, en el caso anterior, no ejecuta nada, con lo cual el usuario puede pensar que su programa no funciona. Para ejecutar el programa debe comandarse: RU,TEST . (Si ahora sigue sin funcionar, efectivamente se tratará de un problema en el programa del usuario.)

CLOAD

Este es un programa que permite compilar y cargar un fichero que contenga todas las subrutinas que necesite, aparte de las del sistema, y cuya primera sentencia sea:

FTN

Se pone en marcha mediante: `CLOAD,&FUENT`
Generará el fichero relocatable `%FUENT` y a continuación cargará el programa, siempre y cuando no haya habido errores de compilación.

AB

Para abortar (detener) un programa que está siendo ejecutado y que vemos que no funciona de la manera esperada, pulsar una tecla cualquiera del terminal (en la consola principal hay que insistir bastante) y a continuación escribir: `AB` (y 'return', claro).

En la consola principal, al pulsar una tecla aparece en pantalla un asterisco, tras el cual escribiremos `AB`:

```
*AB
```

En la consola 7, al pulsar una tecla aparece '`07>`', tras lo cual escribiremos `AB`:

```
07>AB
```

OF y DE

Para hacer desaparecer un programa cargado, pues lo vamos a modificar y cargar de nuevo, el File Manager dispone del comando `OF`:

```
OF,FUENT
```

Sin embargo, si se está trabajando en la consola 7, tal comando genera un mensaje que es escrito en la consola principal, lo que puede provocar problemas al usuario de tal consola.

Es por esta causa, principalmente, que se aconseja que quien trabaje en la consola 7 utilice el comando `DE` ("delete"). Ejemplo:

```
DE,FUENT o bien OF,FUENT
```

En realidad DE no es más que un programa escrito en Fortran destinado a evitar las dificultades que el comando OF nos ocasionaba.

<841215.1335>

UTILIZACION DE LAS FUNCIONES 'TPL'

CONTENIDO:

1. Introducción
2. Descripción de los comandos
3. Ejemplos
4. Utilización de TPL desde ANAE
5. Ejemplos

1. Introducción

Con TPL designamos tanto un programa completamente autónomo de este nombre como un comando del programa ANAE que nos permite también utilizarlo.

Se trata, en cualquier caso, de un programa interactivo que se utiliza para rotular textos (T) mediante el plotter (PL), con el fin de rotular gráficas realizadas previamente (con ANAE o con cualquier otro programa a tal efecto) o bien rotular diagramas o esquemas.

Su inclusión dentro de ANAE responde principalmente a que es posible utilizarlo desde un fichero de comandos, lo que permite realizar rotulaciones parecidas de manera rápida y segura (sin errores). Ello resulta especialmente útil, por ejemplo, cuando hay que rotular de manera parecida un conjunto de gráficas.

2. Descripción de los comandos

TPL dispone de 10 comandos de 2 caracteres cada uno de ellos y que pueden llevar ninguno, uno o dos parámetros. Si alguno de los parámetros requeridos no es entrado, se toma el valor que tenía anteriormente, con excepción del comando TE.

Después de la ejecución de cualquier comando aparece en pantalla el estado resultante del conjunto de variables que usa el programa.

??

Solicitud de información. Escribe en la pantalla la lista de los comandos posibles.

TEtexto

Da entrada al texto que deseamos rotular, empezando a contar desde el primer caracter que sigue inmediatamente a TE. El texto puede tener una longitud total de 150 caracteres (sólo 80 en dentro de ANAE). Este comando actúa sobre otras variables del programa:

actualiza la longitud total del texto (LT, en mm) y el número de caracteres (variable NC).

CSx,y

Define la "celda" del carácter a usar (anchura, altura) en milímetros. Un carácter ocupa los 2/3 de la anchura de la celda, dejando el 1/3 restante para la separación entre caracteres. Asimismo, un carácter en mayúscula ocupa sólo 1/2 de la altura de la celda, dejando el resto como espacio entre líneas. Este comando actualiza la longitud total del texto (LT, en mm).

LTx

Define la longitud total del texto en mm, actualizando la anchura del carácter (CSx).

POx,y

Da la posición absoluta, en mm, del lugar, sobre la hoja de papel, donde se desea que empiece la rotulación. La pluma queda colocada en tal posición.

Atención: tener en cuenta que las dimensiones útiles de la hoja de papel en el plotter son 257.5x190 mm.

INx,y

Da incrementos en milímetros a la posición de la pluma, tomando su posición como nueva posición absoluta (es decir, actualiza PO). La pluma queda colocada en la nueva posición.

DIa

Define la dirección de escritura (en grados). El origen de ángulos es el eje X del plotter (la longitud más larga del papel) y el sentido es directo.

VSv

Define la velocidad de trazado. Inicialmente vale 38 cm/s, que es su velocidad máxima. Cuando se utiliza Rotring o un rotulador algo gastado es conveniente reducir la velocidad. Para un Rotring de tamaño .3 es conveniente usar VS2.

SLtan

Define la inclinación de los caracteres, en base al valor de la

tangente del ángulo de caída.

WRn

Da la orden de escritura del texto, definiendo el color (1 o 2; por defecto toma el 1). No modifica ningún parámetro, salvo el color, si se da, por lo que una nueva ejecución de WR escribe exactamente lo mismo y en la misma posición.

EX

Salida (despedida) de TPL.

3. Ejemplos

1) Para escribir el texto "Centro Astronomico de Yebes" con los caracteres (mayúsculas) de 5 mm de alto y ocupando una longitud total de 120 mm, cruzando la hoja en diagonal y comenzando en la posición 100,50, comandaremos sucesivamente:

```
TECentro Astronomico de Yebes
CS ,10
LT 120
DI 45
PD 100,50
WR
```

2) Para centrar un rótulo en un punto, nos centraremos en dicho punto con sucesivos comandos IN. Una vez definidos CS y TE, tendremos la longitud total del rótulo (en LT). Desplazándonos de nuevo mediante IN una distancia igual a la mitad de la longitud total, estaremos posicionados en el inicio del texto. (Hay que tener en cuenta que este desplazamiento ha de hacerse en la dirección opuesta a la definida por DI. Si DI es 0, entonces el desplazamiento será de la forma IN -LT/2. Si DI es distinto de 0, habrá que dar la componente X y la componente Y de tal desplazamiento.) A continuación puede ejecutarse WR.

4. Utilización de TPL desde ANAE

Desde ANAE se puede ejecutar TPL tal como se ha descrito (modo

interactivo) y también de otro modo que permite su ejecución desde un fichero de comandos, sin intervención directa del usuario.

La utilización de TPL desde un fichero de comandos para ANAE exige la escritura de comandos compuestos. Cada comando consta de los caracteres TPL seguidos (tras un blanco) de los dos caracteres que identifican el comando de TPL y de sus respectivos parámetros. No tiene sentido utilizar ???. Desde fichero e comandos, los caracteres TPL seguidos de espacio en blanco sirve para inicializar el plotter.

5. Ejemplos

1) Para ejecutar el ejemplo 1 del apartado 3, podríamos escribir el siguiente fichero de comandos:

```
TPL
TPL TECentro Astronomico de Yebes
TPL CS ,10
TPL LT 120
TPL DI 45
TPL PO 100,50
TPL WR
TPL EX
CLOSE
```

2) Para rotular al pie de una gráfica las unidades, en la forma:

```
          -1
V      (km s )
lsr
```

con caracteres de distintos tamaños y ligeramente inclinados (SL.2). La posición inicial debe fijarla previamente el usuario mediante comandos PO e IN.

```
TPL      ! inicializa TPL
TPL VS2  ! Velocidad baja, pues se va a usar Rotring
TPL CS3 6
TPL SL.2
```

TPL TEV (km s)

TPL WR

TPL TE1sr

TPL IN3 -2

TPL WR

TPL TE-1

TPL CS2,4

TPL IN 24 4

TPL WR

TPL EX

CLOSE