

SEMINARIO DE ORDENADORES EN LA ENSEÑANZA SECUNDARIA

Participantes: P. Campón Rodillo, M. Cibantos, I. Fernández Flórez, E. García Camarero, M^a T. Molina, A. Sánchez, M^a L. Zabala.

Coordinador: S. Montero.

Sesiones: días 16, 23 y 30 de enero.

TRADUCTOR 70/13

Por I. Fernández Flórez e I. Ramos

Sección I - Generalidades

El Traductor 70/13 es un programa escrito en FORTRAN IV y MAP que simula un ordenador ideal en la IBM 7090 del CCUM. Este ordenador ideal consta de una Memoria de 100 palabras direccionables, un Contador de Dirección y un Registro Acumulador donde se realizan todas las operaciones aritméticas; como Unidad de Entrada posee una lectora de fichas perforadas y como Unidad de Salida una impresora de líneas; ambos dispositivos están conectados on-line con la 7090.

Su lenguaje fuente tiene las siguientes características generales: es un lenguaje de nivel máquina que trabaja con direcciones absolutas, está escrito en castellano y posee casi todas las operaciones fundamentales de un ordenador digital real. Las instrucciones posibles son:

LEER EN LA POSICION (XX) EL CONTENIDO DE LA UNIDAD DE ENTRADA.
DEPOSITAR EN LA POSICION (XX) EL CONTENIDO DEL ACUMULADOR.
ESCRIBIR EL CONTENIDO DE LA POSICION (XX) EN LA UNIDAD DE SA-
LIDA.

TRASLADAR EL CONTENIDO DE LA POSICION (XX) AL ACUMULADOR.
TRASLADAR AL ACUMULADOR EL NUMERO = XXXXXX.

SUMAR AL ACUMULADOR EL CONTENIDO DE LA POSICION (XX).

SUMAR AL ACUMULADOR EL NUMERO = XXXXXX.

RESTAR DEL ACUMULADOR EL CONTENIDO DE LA POSICION (XX).

RESTAR DEL ACUMULADOR EL NUMERO = XXXXXX.

MULTIPLICAR EL ACUMULADOR POR EL CONTENIDO DE LA POSICION (XX).

MULTIPLICAR EL ACUMULADOR POR EL NUMERO = XXXXXX.

DIVIDIR EL ACUMULADOR POR EL CONTENIDO DE LA POSICION (XX).

DIVIDIR EL ACUMULADOR POR EL NUMERO = XXXXXX.

HALLAR LA RAIZ CUADRADA DEL ACUMULADOR.

PONER EN EL CD LA DIRECCION (XX).

SI EL ACUMULADOR ES MAYOR QUE CERO PONER (XX) EN EL CD.

SI LA POSICION (XX) ES MENOR QUE EL ACUMULADOR PONER (YY) EN
EL CD.

SI LA POSICION (XX) ES IGUAL QUE EL ACUMULADOR PONER (YY) EN
EL CD.

SI LA POSICION (XX) ES MAYOR QUE EL ACUMULADOR PONER (YY) EN
EL CD.

HACER MM VECES LAS NN INSTRUCCIONES SIGUIENTES.

PARAR

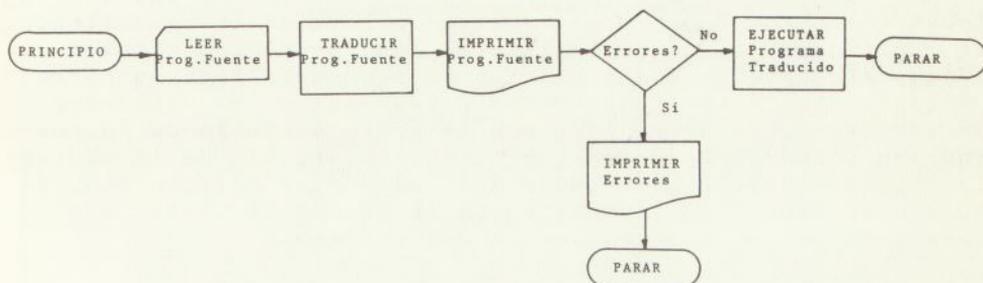
FIN

DATOS

FINAL

Estas instrucciones se traducen a sentencias escritas en Fortran IV en una correspondencia una a una; este programa Fortran, escrito por el Traductor en una cinta magnética, es ejecutado bajo control del Sistema Operativo IBSYS y da los resultados del problema en la impresora on-line de la 7090.

El organigrama general del Programa Traductor es el siguiente:



Como puede verse en la Fig. 1, las Fases principales del programa son:

1.- Lectura de las Instrucciones Fuente. Estas instrucciones van perforadas una en cada tarjeta, empezando siempre en columna 1. El número entre paréntesis puede variar de 1 a 99, indicando una de las posibles direcciones de la Memoria de la máquina; en caso de que tenga una sola cifra, se puede poner (0X) o simplemente (X), donde X varía de 1 a 9. El número que puede aparecer a la derecha del signo = puede ser entero o real con un máximo de 6 cifras (incluyendo el punto si es real).

En el caso particular de la instrucción HACER, se puede omitir el segundo número y entonces se supone que vale 1; es decir, se puede redactar así:

HACER MM VECES LA INSTRUCCION SIGUIENTE

La abreviatura CD significa Contador de Dirección, es decir, se refiere al registro que contiene la dirección de la instrucción que se está ejecutando.

La última instrucción de cada programa debe ser la instrucción FIN y debe tener al menos una instrucción PARAR.

Los datos fuente deben ir precedidos de la instrucción DATOS e irán perforados en las 6 primeras columnas de cada ficha, a razón de uno por tarjeta; pueden ser números enteros o reales, los primeros sin punto, los segundos con él, en cualquier posición de las 6 previstas para ello.

Después del último dato se pondrá una instrucción FINAL, que marca el fin físico del paquete de fichas.

2.- Traducción del Programa Fuente. Cada una de las instrucciones fuente da lugar a una sentencia Fortran IV equivalente. Las 6 primeras letras de la instrucción fuente constituyen el código de operación y especifican la sentencia Fortran correspondiente. Las direcciones entre paréntesis y los literales que van detrás de un signo = son la parte variable de la instrucción y son introducidas en la parte variable de la sentencia Fortran equivalente. Todas las sentencias Fortran generadas llevan número de sentencia con objeto de facilitar las transferencias de control dentro del programa.

3.- Impresión o listado del Programa Fuente. En todos los casos se imprime el programa fuente tal como fue escrito por el programador, precedido de unas cabeceras o títulos generales (Véase el Ejemplo del final).

4.- Búsqueda de Errores en el Programa Fuente. En el proceso de traducción está prevista la detección de algunos errores sintácticos (Véase colección de mensajes de error). Esta colección se irá ampliando a medida que la experiencia demuestre la necesidad de nuevos mensajes de error. Si se detecta algún error de los previstos en el programa traductor, se imprime el mensaje correspondiente debajo de la instrucción errónea.

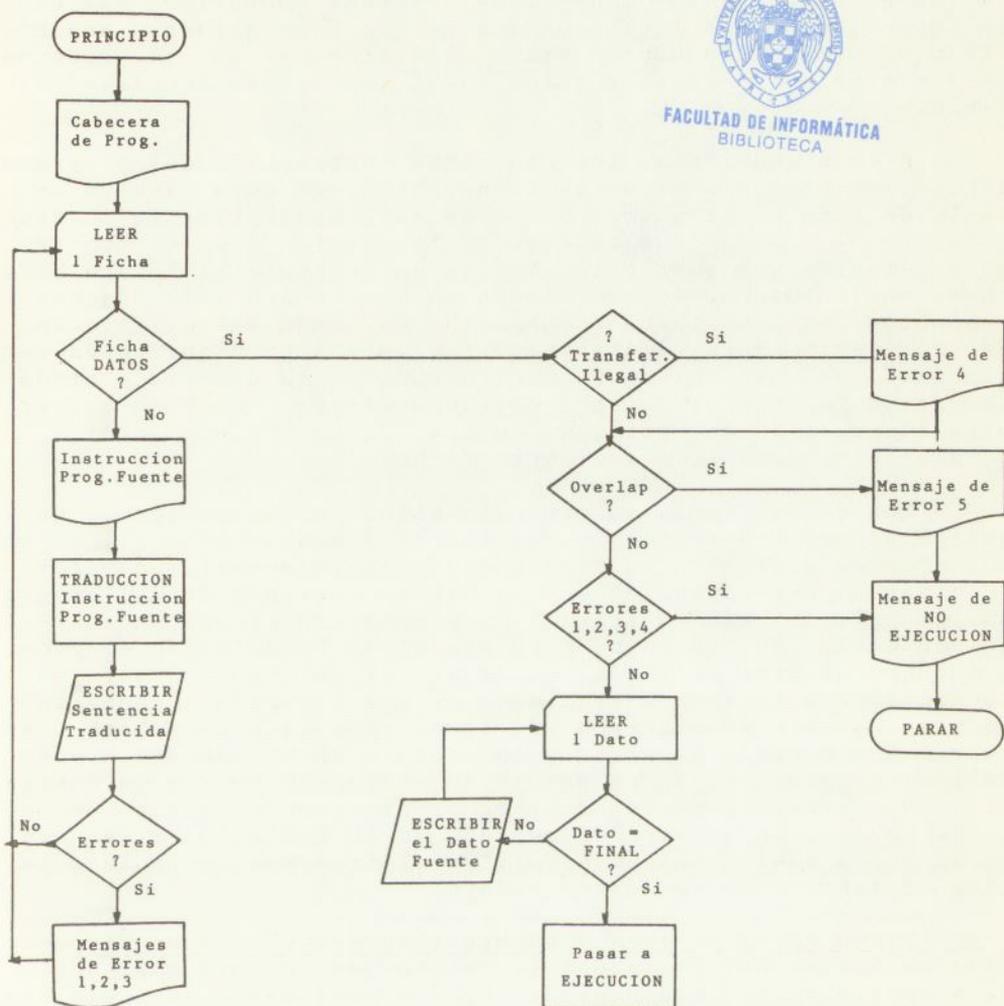
5.- Ejecución del Programa Traducido. Si no se han detectado errores, se ejecuta el programa Fortran y se imprimen los resultados.

Sección II

En la Fig. 2 puede verse un organigrama algo más detallado que el anterior, que examinaremos con más detenimiento.



FACULTAD DE INFORMÁTICA
BIBLIOTECA



Primero se escriben en la impresora unas cabeceras como las del Ejemplo final; conviene a nuestro juicio suprimir en la impresión cualquier mensaje del Sistema Operativo, así como la impresión de las fichas de control de éste para que sea más efectivo desde el punto de vista didáctico, ya que se pretende que la simulación de la máquina ideal sea lo más completa posible.

A continuación se lee una ficha instrucción del programa fuente; se pregunta si es la ficha DATOS, en cuyo caso ya se ha leído todo el programa. Si no es así, se imprime la instrucción fuente y se pasa a la fase de Traducción que, en nuestro caso, efectúa una subrutina escrita en lenguaje simbólico MAP, cuyo funcionamiento se describirá en la Sección III. Una vez traducida, esta sentencia se escribe en algún dispositivo de Salida (perforadora, cinta magnética, etc.) para que pueda ser procesada después. Si en la instrucción se ha detectado algún error, se imprime el mensaje correspondiente y se pasa a leer otra instrucción del programa fuente. Si no hubiese errores, se pasa directamente a leer otra ficha.

Cuando se lee la instrucción DATOS, se investiga si ha habido alguna transferencia de control ilegal, es decir, si el contador de dirección ha alcanzado en algún momento un valor superior al correspondiente a la última instrucción del programa. Si ha sido así, se imprime un mensaje de error; en cualquiera de los dos casos se pregunta si se ha solapado el programa con el área de datos, es decir, si en algún momento se ha depositado un dato o resultado en una posición de memoria correspondiente al programa. Si se ha producido este hecho, se imprime un mensaje de error y un mensaje de NO EJECUCION y se detiene el proceso; si no ha habido solapamiento, se investiga si se ha detectado algún error en el proceso de traducción; si lo ha habido, se imprime el mensaje de NO EJECUCION y se para y, si no ha habido ningún error, en el programa, se pasa a la fase de lectura de los datos fuente.

Estos datos se leen y se escriben después en un dispositivo de Salida del ordenador (el mismo que se haya empleado para escribir las instrucciones traducidas) hasta que se alcance la instrucción FINAL que marca el fin físico del paquete fuente.

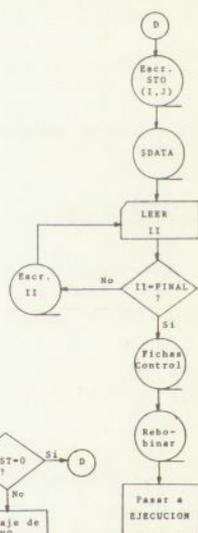
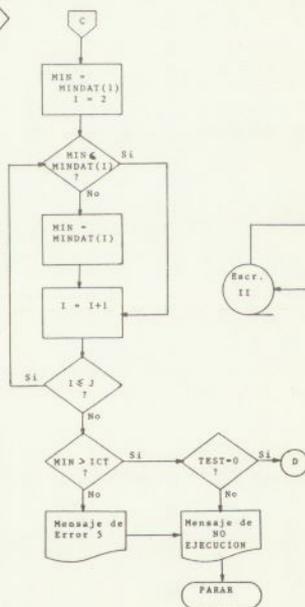
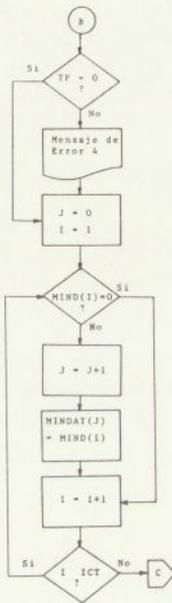
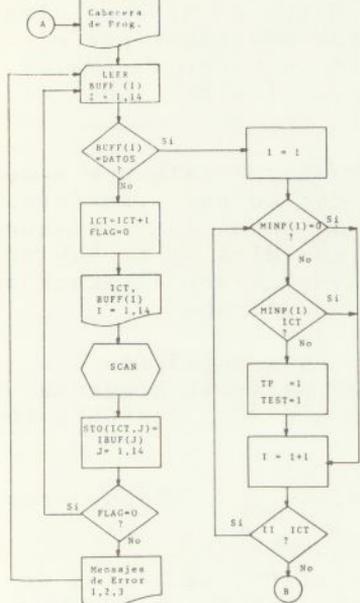
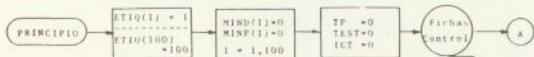
Una vez que se ha llegado a FINAL, se pasa a ejecutar el programa traducido, que imprimirá en la impresora los resultados del problema planteado.

Sección III

En esta sección vamos a describir el programa traductor tal como funciona en la IBM 7090 del CCUM. Adjuntamos el organigrama detallado y el listado del programa, así como la colección de mensajes de error.



FACULTAD DE INFORMÁTICA
BIBLIOTECA



La variable ETIQ es una variable alfanumérica cuyos valores van de 1 a 100, los cuales serán las etiquetas o números de sentencia de las sentencias Fortran generadas.

Los 100 valores de la variable MIND se ponen inicialmente a cero, junto con los 100 valores de MINP. En MIND se almacenarán más tarde las direcciones de datos empleadas por el programa fuente y en MINP las direcciones de programa (valor del contador de dirección) empleadas en las instrucciones de transferencia de control.

Las variables TP, TEST e ICT se ponen inicialmente a cero; el valor de TP indicará si se ha producido una transferencia de control ilegal; el valor de TEST al final de la traducción de todas las instrucciones fuente, nos dirá si ha habido algún error de cualquier tipo de los previstos por el programa e ICT marcará el orden de las instrucciones fuente.

A continuación se escriben en una cinta magnética las tarjetas de control necesarias para poder pasar el programa traducido a Fortran IV en la 7090 del CCUM (\$JOB, \$EXECUTE IB-JOB, \$IBJOB e \$IBFTC).

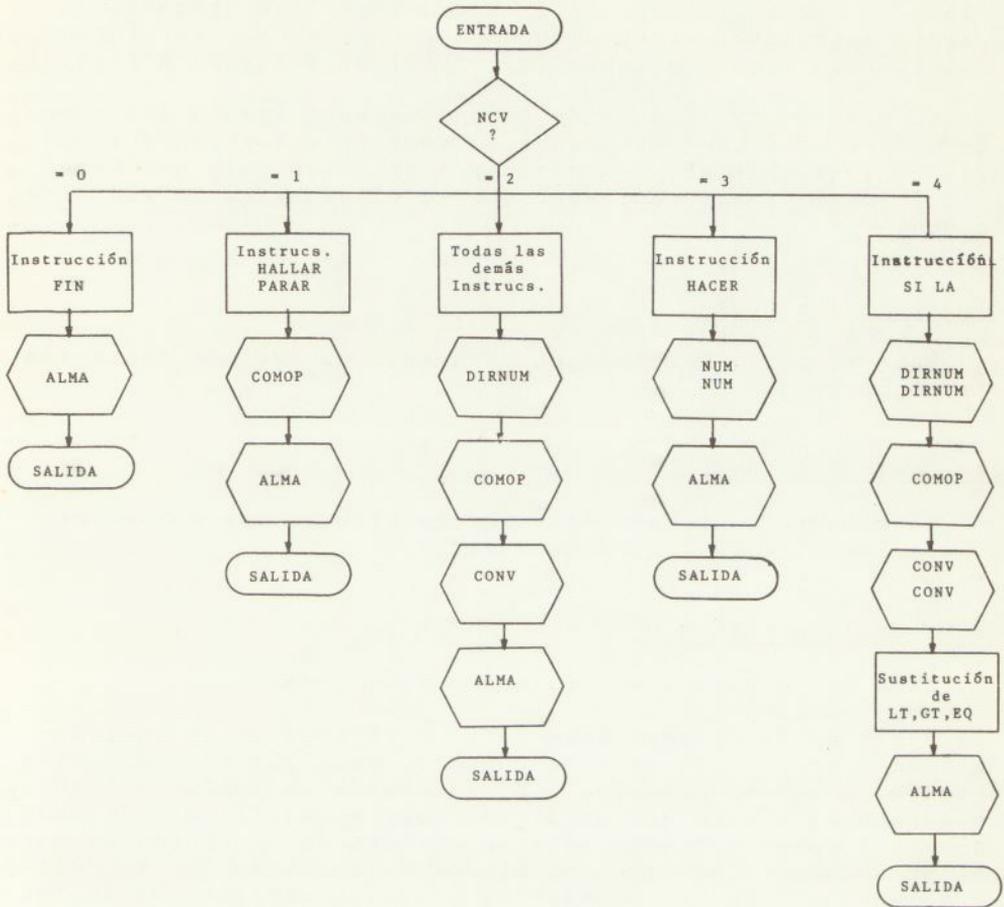
Las cabeceras de programa impresas on-line pueden verse en el Ejemplo del final.

La variable BUFF contiene la instrucción fuente, es decir, el contenido de una de las tarjetas del paquete fuente. La variable FLAG se pone a cero cada vez que se lee una instrucción fuente y sirve para detectar, según su valor, el tipo de error encontrado en el proceso de traducción.

Subrutina SCAN

Esta subrutina es la encargada del proceso de traducción y del análisis sintáctico de las instrucciones fuente. Sus argumentos de Entrada son las variables BUFF y ETIQ correspondientes a la instrucción que va a ser traducida. Sus argumentos de Salida son IBUF (sentencia Fortran equivalente), FLAG (señalizador de error) y los valores de MIND y MINP correspondientes.

Su organigrama es el siguiente:



Al entrar en la subrutina lo primero que hace es comparar las primeras 6 letras de la instrucción que está en BUFF con los elementos de la Tabla de Operaciones incorporados en la propia subrutina; si no encuentra concordancia con ninguno de ellos, hace FLAG=1 y devuelve control al programa principal.

El proceso de traducción depende esencialmente del número de campos variables que tenga la sentencia Fortran correspondiente (variable NCV del organigrama). Veamos lo que hacen las macro-operaciones representadas en el organigrama mediante exágonos.

Macro-operación ALMA

Esta macro lo que hace es almacenar en IBUF la sentencia Fortran generada.

Macro-operación COMOP

Esta macro sustituye el valor de ETIQ en las dos primeras posiciones de la sentencia Fortran.

Macro-operación DIRNUM

Esta macro hace las siguientes funciones:

1) Busca un paréntesis abierto o un signo = en la instrucción fuente (BUFF). Si no lo encuentra, hace FLAG=2 y devuelve control al programa principal. Si encuentra un paréntesis abierto almacena el número que va detrás (que es una dirección absoluta) en el campo variable correspondiente de la sentencia Fortran. Si encuentra un signo =, almacena el número que va detrás (que es un valor entero o real) en el campo variable de la forma alternativa de la sentencia Fortran.

2) Además de lo anterior, esta macro guarda en una posición de almacenamiento temporal el valor alfanumérico de la dirección utilizada en la instrucción fuente, posición que más tarde usará la macro CONV.

Macro-operación CONV

Esta macro transforma el valor alfanumérico de la dirección utilizada en la instrucción fuente (almacenado en una palabra por DIRNUM) en su valor numérico equivalente. Este valor queda almacenado en MIND o MINP según que sea una dirección-dato o una dirección que afecta al contador de dirección.

Macro-operación NUM

Esta macro es muy parecida a DIRNUM y se emplea exclusivamente para la instrucción HACER; ahora lo que busca es un campo numérico en la instrucción fuente y lo sustituye en la sentencia Fortran (el número de veces que se va a repetir el ciclo); la segunda vez que es llamada esta macro, se busca el segundo campo numérico de la instrucción; si lo encuentra, le suma este valor a la propia dirección de la instrucción para obtener el número de sentencia de la última sentencia del rango del ciclo. Si no lo encuentra, supone que es la unidad.

Sustitución de LT, GT y EQ

La instrucción SI LA da lugar a cuatro campos variables: el número de sentencia (obtenido por COMOP), la dirección de la posición de memoria que se compara con el acumulador (obtenida por DIRNUM), el valor que hay que poner en el contador de dirección en el caso de que se cumpla la condición impuesta (obtenido en una segunda llamada a DIRNUM) y el operador de relación que liga el valor del acumulador y el valor de la posición de memoria. Las palabras clave para la determinación del operador de relación son: MAYOR, MENOR e IGUAL que dan lugar a los operadores GT, LT y EQ en la sentencia Fortran equivalente.

Una vez que la subrutina SCAN ha devuelto control al programa principal, se almacena en la ICT-ésima fila de la variable STO la sentencia Fortran generada (IBUF). Si FLAG = 0 no ha habido errores y se pasa a leer otra ficha; si FLAG es distinto de cero, es que ha habido algún error en la instrucción fuente, se imprime el mensaje de error correspondiente y se pasa a leer otra ficha.

Cuando se lee la instrucción DATOS, se pasa a la fase de comprobar si ha habido alguna transferencia ilegal en el programa; para ello se eliminan todos los MINP = 0 (correspondientes a las instrucciones que no usan direcciones que afectan al contador de dirección) y con los restantes se comprueba que son menores que el número total de instrucciones del programa (contando la instrucción FIN). Si se encuentra alguna transferencia ilegal, hace TP = 1 y TEST = 1 e imprime un mensaje de error.

Después se comprueba si se ha depositado algún dato o resultado en alguna posición correspondiente al programa. Para ello se eliminan todos los MIND = 0 (correspondientes a las instrucciones que no usan direcciones del área de datos) y se meten en MINDAT todos los MIND distintos de cero. Una vez hecho esto, se obtiene el valor mínimo de MINDAT que se coloca en MIN y se pregunta si este valor es mayor que el número de instrucciones del programa. Si es menor o igual, se imprime un mensaje de error, el mensaje de NO EJECUCIÓN, y se detiene el proceso.

Si no ha habido solapamiento, se pregunta por el valor de TEST; si es distinto de cero es que ha habido algún error y se detiene el proceso después de imprimir el mensaje de NO EJECUCION.

Si TEST = 0, se escribe en la cinta magnética la matriz STO que contiene las sentencias Fortran generadas; luego se escribe la ficha de control \$DATA.

A continuación se pasa a leer los datos fuente (variable II) que son escritos igualmente en la cinta.

Para cerrar el proceso se escriben en la cinta las fichas de control terminales (\$EOF, \$IBSYS, \$JOB, \$STOP), se rebobina la cinta y se pasa a ejecución el programa Fortran contenido en ella.