INTERNAL USE ONLY

INTERNATIONAL COMPUTERS AND TABULATORS LIMITED

Scientific Programming Department
I.C.T. 1900 Series

FORTRAN NOTE 2
26.3.65

A Proposed Method of Describing a FØRTRAN Program to the FORTRAN IV (ICT) Compiler

Comment on ASA Specifications

The ASA Fortran language enables programs to be written in a machine independent form, but in itself does not allow enough information to enable programs to be completely described and successfully run on the 1900 Series computers.

In particular, the language does not allow a complete description of object time peripheral operations in that no explicit definition of the type of peripheral device to be used can be given; nor is there any way within the language of specifying auxiliary information which may be required for the successful use of some types of peripheral device (e.g. a File name for magnetic tape).

Also if a compiled program is too large for a particular machine, the language does not indicate a way of generating or referencing Overlay procedures to enable the program to be run in the space available.

In order to complete the information required by the compiler the statements of the first segment of program to be compiled must be preceded by a "Program Description". This program description is only necessary if a complete program is being compiled and consolidated, not if a segment or series of segments not comparing a complete program is being compiled. It is immaterial whether this 'complete program' actually contains any FØRTRAN source segments or contains only semi-compiled segments.

Program Description

A Program Description consists of a series of statements from the following family of statements (written in FØRTRAN Format but with no continuation lines).

INPUT

ØUTPUT

INPUT/ØUTPUT

ØUTPUT/INPUT

ØVERLAY

introduced by the statement PRØGRAM

PRØGRAM Statement

This statement is written in the form

PRØGRAM Name, priority, retention cycle
or PRØGRAM Name, priority
or PRØGRAM Name

"Name" is the name under which the program will run at object time.

The first four characters of Name (including spaces) are taken to be the program name.

"Priority" is an integer between 1 and 99 which is the priority of the object program. If this is missing from the PRØGRAM statement the priority is taken to be 50.

"Retention Cycle" is the period (in days) for which the object program is protected. No retention cycle is required if compilation is to punched tape or punched cards. If the retention cycle is required but not stated it is taken as 0.

Peripheral Statements

These are the INPUT, ØUTPUT, INPUT/ØUTPUT and ØUTPUT/INPUT statements. They describe the various methods of operation of the peripheral devices and all have the same general format.

The format of a peripheral statement is as follows

Type n_1, ... n_k = peripheral information.

"Type" is INPUT, ØUTPUT, INPUT/ØUTPUT or ØUTPUT/INPUT.

The "n_i" are the values by which the peripheral device is known within the object program. They are small integer (<4096).

"peripheral information" designates the peripheral device which is known by the values "n_i" and may give auxiliary information such as a "file name" for that device. The format for "peripheral information" is as follows.

peripheral designation (auxiliary information)
or peripheral designation.

The "peripheral designation" is a 3 character code which defines the device. The first two characters define the type of device, and the third character specifies the device number (0-7).

Initially the types of device are as follows

TR - tape reader
TP - tape punch

CR - card reader

CP - card punch

MT - magnetic tape

LP - line printer

The "auxiliary information" if applicable consists of a File name and a retention cycle separated by a comma. If it is not necessary to have a retention cycle the auxiliary information consists of a File name alone.

INPUT Statement

This statement is used to describe a peripheral device which is only going to be used for input operations.

e.g. INPUT 3,5 = TR0 indicates that within the program, the value 3 or the value 5 in an input operation describes the primary tape readers as the current peripheral device.

ØUTPUT Statement

This statement is used to describe a peripheral device which is only going to be used for output operations.

e.g. OUTPUT 10 = MT0 (JACK/SYSTEMS,3) indicates that within the program, the value 10 in an output operation describes the first magnetic tape deck as the peripheral device. Further the magnetic tape will be opened with a File name "JACK/SYSTEMS" and will have a retention cycle of 3 days.

INPUT/ØUTPUT Statement

This statement is used to describe a peripheral device (initially only a magnetic tape unit) which is going to be used for both input and output operations. The essential feature about the use of this statement is that the object program expects to input information from the device before it uses the device in an output operation.

e.g. INPUT/ØUTPUT 15 = MT1 (FORTRAN DATA) indicates that within the the program, the value 15 in a peripheral operation describes the second magnetic tape unit. Further, the first peripheral operation using this tape unit will be an input operation from a tape with a File name "FØRTRAN DATA".

ØUTPUT/INPUT Statement

This statement is used to describe a peripheral device (initially only a magnetic tape unit) which is going to be used for both input and output operations. The essential feature about the use of this statement is that the object program expects to write to the device before reading from it.

e.g. ØUTPUT/INPUT 20 = MT2 indicates that within the program the value 20 in a peripheral operation describes the third magnetic tape unit. Further the first peripheral operation using this tape unit will be a "write" operation to a "Scratch" tape. (no File name).

ØVERLAY Statement

This statement is written in the following form

ØVERLAY n, a = b = --- = p

n is a small integer called the "Overlay Area Number" ($n \geq 1$)

a,b---p are the names of segments in the program which are obeyed from the Overlay Area numbered "n". No two segments in this list are in the main store at the same time.

The only segments which may lie in an overlay area are SUBRØUTINE segments with no formal arguments.

e.g. a subroutine called by the statement CALL JØE may lie in an overlay area, but a subroutine called by the statement CALL JACK (A) may not lie in an overlay area.

Monitor Peripherals

A FØRTRAN program may have a monitor routine associated with it. If this routine is to be used, it is necessary to indicate to the Compiler from what device monitor steering information will be read, and to what device the monitor results are to be sent. This is done by means of the peripheral statements INPUT and ØUTPUT where the peripheral value is given by the word "MØNITØR".

Thus the statement INPUT 3, MØNITØR = TR0 indicates that monitor information will be read from the primary tape reader. The value 3 in a standard input operation also indicates that reader.

The statement ØUTPUT MØNITØR = LP0 indicates that monitor results will be expected on the primary Lineprinter.

Summary

A Program Description consists of a selection of statements of the types described which will fully define all relationships among object time peripheral values and devices, and define all overlay areas and their contents according to the rules given above. This Program Description is introduced by a PROGRAM statement. This statement must occur at the head of a Program Description and may not occur anywhere else.

The Program Description to the Compiler

The Compiler treats the Program Description in the same manner as it treats a Program Segment. For the various statements it generates constants and lists for use at object time in the form of Semi-compiled program. This psendo segment is given the name %%%F and is compiled as a "subroutine" rather than as a "master". The leader generated for this psendo segment contains all the cue information about peripheral devices and I/Ø object time packages and about overlays and overlay packages that is required by the Consolidator.

All Program Descriptions compile into the same name psendo segment (%%%F) so that a semi-compiled version of the Program Description can be superceded by a new version if a segment of the original program has to be recompiled, followed by the original semi-compiled version of the complete program.

V.K. Taylor
K.F. James