INTERNAL USE ONLY

INTERNATIONAL COMPUTERS AND TABULATORS LIMITED

Programming Languages Division

Fortran Note 24

28th March, 1966

Utility Programs and the Production of Fortran Compilers

This note gives some general information concerning the production of the compilers and describes briefly the various utility programs that are used in this connection.

The Production of Fortran Compilers

Compiler Segments

The Fortran compiler is based on the 'compiler kit' philosophy. That is, in addition to the segments forming the 'heart' of the compiler there are a number of alternative 'system' segments. Different compilers may thus be produced by choice of different 'system' segments. There are currently 24 different types of segments, the majority having a name of the form:-

F x 2 y

where x is a letter of the alphabet

y is A - general
P - paper tape
C - cards
M - magnetic tape
L - limited version (Basic Fortran)

Details of each segment are given in appendix 1. In addition, the letter T may be appended to a segment to denote that it is under test.

Compilation

The segments are held in PLAN souce form on magnetic tape, amendments and compilations being made by using the COSY system (≠XPMA + ≠XPLJ) on a four tape cyclic system.

At present three source files exist: FORTSOURCE which contains those segments needed to generate XFAM, XFAP, and XFAC; ZFOMSOURCE which holds the segments for the Basic Fortran compiler XFOM; and TESTSOURCE which basically contains XFAM segments, and is used to develop language enhancements and other off-line tests which, if included in the FORTSOURCE file, would make this too unwieldy to handle.

Fortran Library Tape (FLT)

Fortran Library Tapes are similar to PRE-MLT and MLT library tapes but contain only ASA Fortran compilers, the ASA Fortran library and utility programs that are commonly used by the fortran team. The contents of the FLT is shown in appendix 11. To guard against tape failures four tapes are

used in a cyclic system, being updated by the library maintenance program #XPMU.

Whenever proved versions of compilers or subroutines are produced they are incorporated into the FLT. There is, however, one complication. Since both XFAP and XFAC contain a segment giving library leader information, these compilers would normally be one generation behind any modified library routines. This is overcome by doing a 'dummy' FLT update run using both XPMU and LIBN to produce only the library leader segment, which is then compiled into XFAP and XFAC. It is then possible to include up-to-date versions of XFAP, XFAC and the modified library routines in the next genuine FLT update. This method ensures that FLT's do not exist containing compilers and library routines which are incompatible.

Testing and Issue

There is no guarantee that amendments made which correct a particular fault in a compiler do not have repercussions and causes other faults to appear. For this reason a diagnostic test program, #DIAG, has been written. It consists of many modular phases which are entered in turn at run time, each phase testing certain aspects of the compiler (not input/output), and, if any discrepancies are found, diagnostic messages are printed out. There are two versions of this program, one for Basic Fortran (XFOM) the other for ASA Fortran (XFAM, XFAP, XFAC). Since the size of the latter version is in excess of 18K, only XFAM is used to compile it. (both input and output being magnetic tape). XFAP and XFAC, which differ only in the system aspect, are then tested in this respect.

When a compiler and/or library subroutine have been amended and tested they are ready for issue. Immediately after issue the mark number is updated.

Description of Utility Programs

#PF01

This routine is used to correct a general purpose loader held in binary on paper tape into a psuedo subroutine %F01 which is held in ≤ 20 word blocks at the end of the magnetic tape library. The subroutine looks like an ordinary semi-compiled program but in fact contains blocks of binary programs. The Fortran 11 compiler #XFLM requires the loader to be held in this manner since there is insufficient space available to either hold the loader in store or read the loader in from magnetic tape where it is held as an ordinary binary program in a type 5 block (≤ 512 words)

#GPLC and #GPLL

These routines #GPLC card orientated and #GPLL paper tape oriented, convert a general purpose loader held in binary to an octal representation which is inserted as data (lower preset) into the output segment of the Fortran compiler. This is necessary for any version of the compiler which holds the loader in core store. Note that the loader cannot be in source form as binary output is required.

#COND

The purpose of this routine is to eliminate all unnecessary run out, E blocks and pause blocks present in a semi-compiled library tape. This routine is now obselescent since #FLPM will produce a condensed semi-compiled library direct from magnetic tape.

#LIBN and #LIBC

These routines will generate a plan source segment consisting primarily of data giving the essential information required by the consolidator about the library routines (i.e. a table of compacted leader information). ASA Fortran compilers require the use of #LIBN which generates a plan source segment FV2A with an internal cue LIBRY. Parameters are read from either paper tape or cards, the library being on magnetic tape in block SRF4. The routine #LIBC is used to create a segment called LIBRY, the library being on paper tape. This routine is used with the Fortran 11 compiler.

MTPR

This is a magnetic tape to printer general listing program which prints the first 20 words of any block read. The advantage over other similar programs is that it will print for each block read only the number of words given in the second character of the block (or 30 words if this is zero or <20). Also it will not halt until it finds two tape marks.

RENB

The purpose of this program is to copy only selected segments of semi-compiled program from an input magnetic tape to an output tape which is opened as a scratch tape but given a specified name and a retention period of 50 days.

This program is used in conjunction with # ZPOT to overcome two difficiencies, in # ZPOT, namely.

a) program may not be renamed
b) segments cannot be deleted.

Since ASA Fortran compilers are at present compiled using the COSY system this program is of use only with Fortran 11 compilers.

SCPR

This routine may be used whenever semi-compiled program held on paper tape, cards, or magnetic tape is to be analysed and printed on the line-printer.

It is in fact identical to # YKBP (see Software Testing Manual for specification) but with the addition that input may be from cards or magnetic tape thus enabling the output from all three compilers, XFAM XFAP and XFAC, to be examined.

CDMT

This routine will transcribe cards in Fortran source language to magnetic tape in subfile format (see PDCC/171 issue 2) suitable for input to # XFAM. Parameter cards mixed with source cards are required to create start of subfile or end of subfile sentinels.

#FLPM

Whenever it is desired to produce the ASA Fortran library on paper tape this routine may be used. It assumes that the library is held on magnetic tape in program library format under group name SRF4 and produces two reels of paper tape in a compact form.

#CSDR

The ASA Fortran compilers are compiled using the COSY system. This system has the advantage of selecting specified segments to be compiled but also has the disadvantage of not being able to duplicate segments without going via a hard copy on cards or paper tape. For other than trivial modifications development of the compilers is achieved by creating duplicates of segments into which modifications are put and tested. When proved, the duplicate segment becomes the up-to-date version and the original is deleted.

The utility program #CSDR is capable of both deleting and duplicating segments, the duplicated segments being renamed in specified. Except for reading parameters from a slow peripheral the process is carried out entirely on magnetic tape, in a manner very similar to #XPMA. In the normal course of development #CSDR is used twice, once to duplicate and rename, once to delete the old segment and name the duplicate as the new segment.

#FLID

Three compiling systems are possible depending on the way in which leader information is supplied to the consolidator for a paper tape (or card) compiler such as #XFAP (or #XFAC)

In order of preference they are:-

(i) Leader information is held in store, (segment FV2A in XFAP or XFAC), semi-compiled information being read from the library supplied at load time.

(ii) As (i) but leader information is read from the library which is also supplied at compile time.

(iii) As (ii) but semi-compiled information is read and punched out (copy switch on) at compile time. Library is not then required at load time.

The utility program #FLID is a means to improve system (ii) by producing a paper tape containing the leaders only, copied from the paper tape library.

#CSRØ

This routine is designed to re-order segments on a COSY source file as specified by a parameter list which may be read from cards or paper tape. Such re-ordering is necessary in order to reduce the core store required by XFOM to 5632 words, the pre-release PLAN compiler ZPLJ is used in place of XPLJ during the COSY run. However at present ZPLJ requires segments to be supplied in overlay unit order within overlay area order with permanent last. Thus, as XPMA is not capable of re-ordering segments CSRØ is used as a preliminary whenever amendments made to XFOM alter the sequence of overlays.

Summary of Utility Programs

Name	Description
//PFOI	Creates Fortran II loader for //XFLM.
//GPLC)	Conversion of binary GPL's to plan format.
//GPLL)	
//COND	Copies semi-compiled program on paper tape removing superfluous run out etc.
//LIBN)	Creates 'LIBRY' segment from the semi-compiled
//LIBC)	library.
//MTPR	Magnetic tape to printer (general).
//RENB	Renames input file to //ZPOT.
//CSDR	Duplicates, renames and deletes segments held on magnetic tape in COSY format.
//SCPR	Prints and analysis semi-compiled programs.
//CDMT	Transcribes cards to magnetic tape in subfile format.
//FLPM	Creates paper tape library from FLT.
//FLID	Creates paper tape with leaders only.
//CSRO	Re-orders segments on a COSY source file.

L.R. FAIRBROTHER.

Appendix I

Compiler Segments

Name	Description
FA2A	Supervisor
FB2M	Input (magnetic tape, paper tape, cards)
FB2A	Input (paper tape or cards only)
FC2M	Output (magnetic tape)
FC2P	Output (paper tape)
FC2C	Output (cards)
FD2M	Consolidator (magnetic tape)
FD2A	Consolidator (paper tape or cards)
FE2A	Segment compiler, statement compiler, END
FF2A	MASTER, SUBROUTINE, FUNCTION, EXTERNAL, BLOCK DATA, RETURN, PAUSE, STOP.
FG2A	EQUIVALENCE, DATA.
FH2A	Arithmetic statements, IF, GO TO, CALL, ASSIGN.
FI2A	DO
FJ2A	Input/output statements.
FK2A	Type statements, DIMENSION, COMMON
FL2A	Listing segment.
FM2A	Expression analysis (Part I).
FN2A	Expression analysis (Part II) and compilation.
FP2A	Input (peripheral independant).
FQ2A	Output (peripheral independant).
FR2A	Instruction generator.
FS2A	List System.
FT2A	TRACE.
FU2A	Error segment.
FV2A	Library leaders (LIBRY).
INTFC	Intrinsic and standard functions.
FZ2A	Program Description.

NOTE: In addition to these segments there is one segment per compiler which supplies the compiler name and mark number.

There is also a set of segments FA2L to FZ2L which comprise the Basic Fortran compiler, XFOM

APPENDIX 11

FORTRAN 4 LIBRARY TAPE (FLT)

Binary Programs

XFAM/1D
XFAP/1D
XFAC/1D
XFCM/1
LIBN/1
MTPR/1
SCPR/1
GPLC/1
CDMT/1
FLPM/1

Library Subroutines (under group name SPF4).

%FAP4/4
%FMN4D/4
%FMN4/5
ALLØT/4
DISENG/4
RLEASE/4
RUNØUT/4
%FINCUT/9
%FIOPT/5
%FIOCARD/5
%FIOLP/5
%FINIL/6
%FIOTA/5
%FIOMT/7
%FIOMTF/6
%FINMT/6
%FIODA/4
%FIODP/6
%FARHD/5
%FEx4/4
REAL/4
AIMAG/4
CMPLX/4
CØNJG/4
CLØG/4
CSIN/4
CCOS/4
CEXP/4
CSQRT/4
%FC2/4
%FCP/4
CABS/4
DMINI/4
DMAXI/4
DSIGN/5
SNGL/4
DBLE/4
DSQRT/5
DLØG10/4
%FDR/4
%FDD/4
DMØD/5
DCØS/5
DSIN/5
DLØG/5
DEXP/4
DATAN/5
DATAN2/5
DABS/5
%FDPØLY/4
%FD2/4
%FDP/5
SQRT/4
COS/4
SIN/4
ABS/4
LABS/4
COT/4
TAN/4
EXIT/4
DATE/4
MØD/5
SWØFF/4
SWØN/4
INT/4
NINT/4
FLØAT/4
ACØS/4
ASIN/4
ACØT/4
ATAN2/4
ATAN/4
SINH/4
CØSH/4
ALØG10/4
EXP10/4
TANH/4
COTH/4
AMØD/5
ACØSH/4
ASINK/4
ATANH/4
ACØTH/4
ALØG2/4
AINT/4
ANINT/4
PLØT/4
SIGN/5
ISIGN/5
XTY/4
EXP/4
ALØG/4
IDATE/4
MAXO/4
AMAXO/4
MAX1/4
AMAX1/4
MINO/4
AMINO/4
MINI/4
AMINI/4
IDINT/4
IFIX/4
DIM/4
IDIM/4
SLITE/4
SLITET/4
SSWTCH/4
DVCHK/4
ØVERFL/4
%FØVL/4
%RØL/4