INTERNAL USE ONLY

INTERNATIONAL COMPUTERS AND TABULATORS LIMITED

Programming Languages Division

FORTRAN NOTE 27

August 1967

The Fortran Batch Monitor System

This note is intended to describe the interface between the Fortran compiler XFAS and the monitor program XFBM from the point of view of the compiler. There is also a description of the systems segments which have been changed and some new common areas. Finally there is a brief description of the loader XP08 and of the new peripheral routines and their interface both with %FINOUT and the system.

1. Introduction

XFBM is a 'trusted' program which monitors the compilation and execution of a series of Fortran source programs, each one being termed a job. To do this it must know what is happening all the time, i.e. whether the object program is being compiled, loaded or run. There are 4 main phases to each job:

1) Job found and compilation started.

2) Compilation completed, object program ready to be loaded.

3) Loading completed, object program ready to run.

4) Run complete, go on to find the next job.

Other phases or combinations thereof may occur, especially under error conditions. For example, if there are errors in compilation phases 2) and 3) are omitted and the compiler goes straight on to the next job.

2. Interface

The interface between XFAS and XFBM consists of a series of DISP instructions. DISP was chosen in preference to SUSWT simply because it is easier to test XFAS apart from XFBM if it does not halt whenever a message has to be given to XFBM. The DISPs fall into two categories:

a) those which give information to XFBM about the progress of a job in the batch.

b) those which require some action from the operator, such as slow peripheral required.

The first character of each DISP is ! and the second ranges from 0 to #14. XFBM uses the ! as identifier and the 2nd character as a modifier to a branch table. Their meanings are as follows with the equivalent HALT in brackets afterwards (if appropriate):

a)	! 5	End of successful compilation	(EC)
	! 6	Object program loaded	(LD)
	! 7	*FOR read	
	! 8	***** read	
	! 9	//// read	
	! :	End of batch	(PF)
	! ;	Overlays loaded	
b)	! 0	CR wanted	(CR)
	! 1	TR "	(TR)
	! 2	LP "	(LP)
	! 3	Incorrect batch parameters	
	! 4	More store required	(ST)
	! <	No SRF6 block	(NL)

DISPs of type a) cause XFBM to test and set various markers, and if directives appear in the wrong order, e.g. two lots of *'s running, messages will be printed on the output device to indicate possible errors in the batch or jobs incorrectly set up. XFBM will normally re-enter the compiler at the next instruction.

DISPs of type b) cause XFBM to halt with a message to the operator. For ease of testing XFAS alone, the next instruction in the compiler is a SUSWT e.g. HALTED CR. XFBM thus re-enters the compiler, when the operator has rectified the condition, at the next instruction after the SUSWT. The only exceptions are ! 3 and ! < when the batch must be restarted from the beginning.

The system is designed to use the system tape as efficiently as possible. The compiler remains in store until there is an object program to be run. The order on the tape

XFAS
SRF6 library block
XPOS loader

is such that there is at most one scan for each job to be run.

If a job is not to be run, i.e. errors in compilation or NORUN (not a published facility, possibly redundant) then the tape is not moved at all and the compiler goes straight on to skip to the next job. Eventually it is hoped to reorder the compiler so that permanent area comes before overlays on the tape. This will save the initial backspace to find the overlays, and will involve writing a special overlay package.

XFAS is never deleted, as far as executive knows, until the end of a batch. XFBH overwrites the area containing XFAS first with the compiler and then with the loader and object program. Consequently, after the first entry to the compiler, peripherals remain allocated and, in the case of magnetic tape, positioned, throughout the execution of a batch.

3. Description of Segments

Overlay structure

Area 1	Unit 1	FBCT	slow input
Area 1	Unit 2	FBMT	magnetic tape input
Area 2	Unit 1	FG2A, FK2A, FM2A, FN2A, FR2A	Main part of compiler
Area 2	Unit 2	FW2S	Consolidation phase

The segments that have been altered are FW2S, FA2S, FB2S, FD2S, FL2S. The two segments FZ2A and FC2B have been deleted and FBCT, FBMT are new segments. Instead of \$TROL, the overlay package is the same as the one in XFOH. The changes to the segments are now described in more detail. Various markers are mentioned in the following descriptions. They are more fully described in the section on common area LC21.

Segment A

This is the supervisor segment as it is all compilers and most sections are the same as in XFAM. All the parts concerned with the final stages of consolidation have been moved to Segment W, some intersegment statements have been removed altogether and minor changes have been made to FINIS and STOP. There is a new section for PAUSE which generates the instructions required to print the message on the system output.

The main change is to the initial entry to the compiler. The first time through for each batch it reads and analyses the batch parameters, and sets markers MARK1 and MARK2 (see section on common area LC21) indicating system input and output devices. Subsequently it discovers system input and output by a series of dummy allot instructions, in fact this is also how it discovers whether it is the first time through or not.

When the correct overlays for the job are in store it enters XFBM, DISP !; , and returns to skip to the start of a job, i.e. *FORTRAN. The first record presented after the skip entry is listed under the heading ** NEXT DATA RECORD** unless it is *FOR, **** or ////. This is only relevant when the previous job failed at object time without reading all its data, when this becomes equivalent to 'last card read reversed'.

ENTRY 4

FMAS1

Positions scratch tape, i.e. rewinds and skips to tape mark.

Resets core size to 11008 in case previous compilation or object program required more core.

Initiates storage areas and starts compiling complete program.

This entry is used by XFBM if object program reads *FOR parameter. The compiler is reloaded and started at ENTRY0,1,2 until markers are set and overlays loaded. Then ENTRY 3 is missed out and direct entry made to start compiling.

ENTRY 7

STOP

Terminates offlined output with end of file sentinel.

Enters XFBM | DISP ! : |

Halted PF

No peripherals are released, as if the object program reads ////, on entering the compiler at this point no markers are correctly set. It was thought unnecessary to do a complicated system of allots and releases when the next action of XFBM is to delete XFAS and thus release all peripherals.

Segment B

This contains the cue INPUT plus two new cues INPUT1, INPUT2. It uses MARK1 as a modifier to branch to the correct input section, paper tape, cards or magnetic tape in FBCT or FBMT. On exit, the next record is presented in CIMAGE, and various checks are made.

Entry at INPUT1.

This is entry to read batch parameters. Initial blank records are ignored, otherwise no checks made.

Entry at INPUT2.

This is entry for skip mode. Check for *FOR, **** and ////.

Normal entry at INPUT.

This checks for *FOR, ****, ////, *DAT and *STE.

SETSC (check for S/C) and PCH1 (S/C output section) are also in this segment.

Action of Segment B on finding

1) *FOR : If already within job (tests JMK) an error message is printed, then parameters are handed to XFBM with DISP !7. Return is to FMAS1 (Entry 4) to start compiling.

2) **** : If not within a job, enter monitor DISP !8. Return is to look for *FOR record. If within job, unset JMK, set marker for no data, NOIN. In skip mode (enter at INPUT2) terminate listing and enter monitor as above, otherwise treat as FINISH.

3) //// : If within job, print error message, then and otherwise enter XFBM with DISP !9. Return is to STOP.

Segment D

The only change is the removal of CSLC which builds up consolidated leader and forms request slip.

Segment L

References to tape punch have been removed, and magnetic tape output been included.

As it is not possible to test reply word for 'page full' condition on MT, there is now a line count kept, and up to 60 lines are allowed to one page.

Magnetic tape output is in blocks up to 128 words long in XQMP - compatible format.

Segment FBCT

This is paper tape and card input, TRSPL and CRSPL virtually unchanged.

Card input uses CDMK as ATLAS or 1900 code indicator. It also checks for a record *ATL or *190, changes CDMK when found and returns for the next record.

Segment FBMT

This is magnetic tape input, MTSPL but rewritten. The main section presents the next record in CIMAGE, and there are two subsidiary section: one checks sentinels when a tape mark is read, the other skips to the start of the next job when the compiler is in skip mode.

Sentinel checks

Start of job sentinel (name PR0G). If word 9 of the sentinel is negative or second two characters are not FB skip to the next job. Otherwise zeroize skip marker, ENTMK, and return for next record.

Start of Fortran sentinel. Return to read next record.

Start of Semicompiled sentinel. Set S/C marker, return for next record.

Start of Data or Steer sentinel. Set *DATA or *STEER in CIMAGE and return to process.

End of subfile sentinel. Ignore it in skip mode. Otherwise subtract 1 from subfile level count SCT and if non-zero read next block, if zero treat as *'s read.

End of composile file sentinel. Treat as //// read.

Unrecognized sentinels are ignored and the action is to skip to the start of the next job.

Segment W

This is brought into store and entered as soon as a program requires consolidation. It performs the following actions:

1) Outputs program description segment %%%F containing cues for system input (if data or steering information) and output routines. Also outputs 1st word of %FI0LIST which contains count or entries in list. Its value is 2 if no input data, 4 otherwise.

2) Scans the library block, passes each record through CSLA (consolidator) and outputs routines called for, in batched form if the library is batched.

3) Prints blank cues if there are any.

4) Positions library tape at end of SRF6 block ready for XFBM to read in loader.

5) Forms request slip (section CSLC of consolidator contained in this seg.)

6) Lists leaders if switch 10 on, (not a published facility but useful for testing purposes).

7) Terminates listing.

8) Tests core size of object program and if > current core size of compiler does GIVE for more core.

9) Sets up table of offset, address of consolidated cue list and relativisor settings.

Enters XFBM with DISP !5, with address of this table in X1 and the entry point for the object program in X0, (#200 if GO 20, normal entry; #210 if GO 28, trace steering list entry).

4. Common area LC21

8 wds.

MARK1	Input marker :	Paper tape - 0 Cards (1900) - 1 Cards (Atlas) - 2 Magnetic tape - 3
MARK2	Output marker :	Line printer - 0 Magnetic tape - 1

Above two words set up at start of Segment A as each job is begun. Used in FB2S, FL2S.

GOMK — Entry point of object program.

Normal entry #200 i.e. GO 20
Trace steer #210 i.e. GO 28

Set up in Segment A when *DATA, *STEER or **** read.
Used in Segment W to pass across to monitor.

JMK — Job mark.

Set zero in Segment A before start of each job.
Set non-zero when *FOR read.
Zeroized again in reading *DAT, *STE, ****, ////.

CORE — Current core size, normally 11008.

Set up in Segment A and increased for list size if required.
Used in Segment W to check if room for object program.

NOIN — No input data for object program.

Set zero at start of each job.
Set non-zero when **** read at end of compilation indicating no object program data. No input routine will be incorporated in the object program.

RDMK — Double buffer mark for cards.

Used only in FBCT, but zeroized initially in Segment A.

CDMK — Card code marker.

Set up at start of each job to value of MARK1.
Used in FBCT as card code indicator. Needs to be distinct from MARK1 as it must be re-set at start of each job even if compiler not reloaded. Re-set whenever *ATL or *190 card read (detected in FBCT).

Common area MTUSE — preset.
2 wds

LSTRE Pointer to message **NEXT DATA RECORD**

Used in Segment A and FBMT.

ENTMK Set zero for normal mode input in FB2S
Set non-zero for skip mode.

Used in FB2S and FBMT.

5. XP08

Loader for Fortran Batch Monitor System.

This loader is based on XP01 with only minor modifications. It is always run under the name XFAS, being read into store by XFBM using a CONT.

On entry, X0 contains the address of a table set up by the compiler as follows:

Wd 0	SHIFT i.e. offset
Wd 1	Address of consolidated cue list as set up by the compiler.
Wds 2-11	Relativisor settings.

Thus relativisors and cue list are set up directly from store instead of being read in from MT.

Semi-compiled is input from MT3 which is already allocated and in a rewound position. It positions this tape initially by skipping to tape mark. S/c segments are in both batched and unbatched form: o/p from the compiler being unbatched, library s/rs being batched.

When the program has been loaded, XFBM is re-entered by a DISP ! 6 message in place of the HALTED LD.

6. Peripheral routines

There are 6 new peripheral routines and these are the only ones used by the system. The 4 input ones are %FIBTR, %FIBCR, %FIBCRA and %FIBMTI corresponding to paper tape, cards (1900), cards (ATLAS) and magnetic tape. The 2 output routines are %FIBLP and %FIBMTO, line printer and magnetic tape. All the routines are more straightforward than their counterparts in XFAH. This is for a number of reasons:

a) The peripheral is already allocated and has a fixed unit number.

b) The magnetic tape routines are essentially equivalent to slow input and so process the tape only in one direction, no backspace or rewinding is allowed.

c) Initial blank lines or cards are not ignored. There is an extra check built into the input routines for records *FOR, **** and ////. When these are found, common area %FBHK is set up with two chars ! 7, ! 8 or ! 9 and execution error 6 reported. %FERROR, after giving trace printout, enters XFBH with a DISP of the value in %FBHK.

A consequence of a) is that the interface with %FINOUT is of a fixed form; thus %FIOLIST and %FIOINF are preset within each routine. For a standard job, with input and output statements, %FIOINF and %FIOLIST contain the following values:

%FIOLIST

Wd 0	4	No. of entries in list. Set up by compiler	
Wd 1	2	Peripheral no.	Set up in output peripheral routine
Wd 2	Addr of Wd 0 of %FIOINF		
Wd 3	6	Peripheral no.	
Wd 4	as Wd 2		
Wd 5	1	Peripheral no.	Set up by input peripheral routine
Wd 6	Addr of wd 2 of %FIOINF		
Wd 7	5	Peripheral no.	
Wd 8	as Wd 6		

If there is no data, then wd 0 is 2 and wds 5-8 are omitted.

%FIOINF

Wd 0	Addr of system output routine		Set up by output routine
Wd 1	2	(indicates output)	
Wd 2	Addr of system input routine		Set up by input routine.
Wd 3	1	(indicates input)	

This interface was worked out without considering the possibility of using magnetic tapes at run time and there may be some difficulty in fitting that in to the existing structure. A possible solution would be to revert to compiling %FIOLIST in Segment V.

The slow peripheral routines are straightforward.

%FIBMT0. This outputs single record blocks in XQMP-compatible format.

%FIBMTI. This assumes input tape positioned just before first record: either data or steer. Each record is presented to %FIN0UT in turn, those of less than 80 characters being space-filled up to 80 characters. When a sentinel is read
'start of subfile' is assumed to be DATA subfile and is ignored, level is set to zero
'end of subfile' at level zero is ignored and level set to 1; at level 1 is treated as **** read.
Other sentinels should not occur.

A.S. FINCH.