

Topic: The JADE BOS System

Derek Stork

5th August 1978

Following the decision to adopt the BOS (Bank Organisation System) method of storing output data on tape and disc (see JADE Notes 9(a) and 24), a version of the BOS program has been prepared for the JADE experiment and is now implemented on the three computers. (DESY IBM 370/168; RL IBM 360/195; NORD 10).

This note first describes the philosophy followed in implementation. Details of the FORTRAN calls available are then given and some hints on optimisation set out. The details of the LOAD modules used and how to link with them are finally given.

At all stages more detailed discussion is left to the BOS reference work:

Internal Report

DESY F14-77/01

August 1977

'BOS - Bank Organisation System - Dynamic

Storage Organisation with FORTRAN', by V. Blobel

We will refer to this as 'the BOS report'.

### Philosophy

The following guidelines have been followed in adapting the BOS for JADE:-

- 1) All routines involving IBM FORTRAN syntax in their calls have been omitted and the existing alternative standard - FORTRAN - callable routines have been used instead. This means that the same calls are available on the IBM + NORD machines. Program differences are therefore transparent to the user.
- 2) The 'low priority' applications subroutines, which involve mainly histogramming and counting statistics, have been omitted entirely. (see section 13 of the BOS report). This was done to save core and because perfectly adequate histogram routines are available and widely-used already by members of the collaboration.
- 3) Apart from the above items a policy of 'minimum interference' with the source subroutines has been followed.

## FORTRAN Implementation

The BOS package is entirely FORTRAN, although one or two assembler subroutines are used internally. Anyone wanting to use BOS must have in his main program the following statements:

COMMON/BCS/TW (NSPACE)

REAL RW (NSPACE)

EQUIVALENCE (IW(1), RW(1))

where NSPACE = an integer constant defining the length of the storage work space IW.

Note that on the NORD 10 the statement

DOUBLE INTEGER IW (NSPACE)

must also be included to ensure that storage space is in 4 byte words as on the IBM's.

On the next pages is an alphabetical list of the FORTRAN-callable routines available in the JADE BOS package. More details of each routine will be found in the BOS Report.

In the table

- \* = input variable to the subroutine
- + = output variable from the subroutine

Routine Call	Purpose	Notes
CALL BDAR (NAME*,N+,INDA+,NLIM*)	Locate all banks with name NAME INDA contains the N output indices (N.LE.NLIM)	NAME must be DOUBLE INTEGER for the NORD 10
CALL BDEF (N*,TEXT*)	Set up a list of N bank names in array 'TEXT'	TEXT must be DOUBLE INTEGER for NORD 10. see BOS Report section 12.
CALL BDLM	Delete a pre-defined set of banks	Banks defined in BMLT are deleted.
CALL BDLS (NA*,NR*)	Delete bank name NA number NR	NA is DOUBLE INTEGER for NORD 10.
CALL BDMP	Print a dump	
CALL BGAR (IGA+)	Perform garbage collection	IGA = 0 No garbage collection done IGA = 1 Garbage collection done
CALL BGAC (IGA+,NW*)	Perform garbage collection <u>if</u> there is no room to add a bank of length NW to the store.	See above.
CALL BINT (NSPACE*,NREC*,NDMP*,NADD*)	Initialise the BOS system.	NSPACE as before NREC = Max No. of words in a bank NDMP = No. of words printed in a dump NADD = 0 <u>Must be</u> your first call to BOS.

Routine Call	Purpose	Notes
CALL BMLT (N*, LIST*)	Define a set of N bank names	LIST must be DOUBLE INTEGER for the NORD 10.
CALL BPRM	Print a set of banks	Banks defined in BMLT are printed.
CALL BPRS (NA*, NR*)	Print contents of bank name NA number NR	NA must be DOUBLE INTEGER on NORD 10.
CALL BSAT (N*, LIST*) CALL BSAW (N*, LIST*)	Two routines to update the special list defined on EMLT.	
CALL BSLT CALL BSLW	More bank definition routines (with markers).	See BOS Note.
CALL BSTA	Printing of final statistics at end of program.	
CALL BSTR (IND*, JW*, NW*)	Store NW words from array JW into work array IW starting at location IND.	JW must be DOUBLE INTEGER on the NORD.
CALL BWRITE (IUN*)	Output a set of banks previously defined to unit IUN.	Garbage collection is done if necessary.

Routine Call	Purpose	Notes
CALL CCHL (NW*, IERR <sup>+</sup> )	Change length of LAST CREATED BANK to NW words.	IERR = 2 if not enough space. See BOS report.
CALL CCRE (IND <sup>+</sup> , NA*, NR*, NW*, IERR <sup>+</sup> )	Create bank name NA, number NR, length NW.	IERR is error flag = 1 bank already exists = 2 not enough space NA must be DOUBLE INTEGER for NORD 10
CALL CLOC (IND <sup>+</sup> , NA*, NR*)	Locate bank name NA, number NR.	Test IND = 0 to see if bank exists NA must be DOUBLE INTEGER for NORD 10.
CALL CMVE (IND <sup>+</sup> , IERR <sup>+</sup> )	Move bank at index IND to end of storage. IND is then returned as new position.	IERR = 2 if not enough space.
CALL CNXT (IND <sup>+</sup> ) CALL CPOS (NA*)	Pair of routines to locate all banks of a given name NA.	NA must be DOUBLE INTEGER for the NORD 10.
CALL CREAD (IUN*, IERR <sup>+</sup> )	Read in data to IW from unit IUN.	IERR = 1 read error occurred. IERR = 2 end of record hit.
ILOC = IBLN (NA*)	IW (ILOC) is set to the index of the first bank with name NA.	NA must be DOUBLE INTEGER on the NORD. IW (ILOC) contains 0 if the bank does not exist.

In addition calls to subroutines BINP, ROUTP and BREADC are available on the IBM machines only. Their use is not recommended.

## Notes

### 1. Bank names

Bank names have 4 alphanumeric characters. The last character must not be a '2' as this designates a low priority bank and these are no longer used.

### 2. Optimisation

The user is advised to read section 12 of the BOS report. Since BOS is a sophisticated program a price has to be paid in the time taken to execute various functions. Section 12 tells of one or two ways in which execution may be speeded up.

In addition it is useful to note that the only action performed by SUBROUTINE BSTR is to call an assembler routine to copy the words from the source array to the destination work space. To avoid this double-call the user may use the following code instead of CALL BSTR (IND,JW,NW).

```
IBM's:-      CALL UCOPY2 (JW,IW(IND+1),NW)
NORD :-      DOUBLE INTEGER IW
              COMMON /BCS / IW (1000)
              INTEGER HW (2000)
              EQUIVALENCE (HW(1),IW(1))
              .
              .
              .
              .
              NWH = 2 * NW
              NSTART = 2*(IND+1)-1
              CALL ZCOPY (JW,HW(NSTART),NWH)
              .
              .
              .
```

This will save time when frequent storage of banks occurs.

3. If at any time the BOS source is recompiled remember that PARM.FORT = XL must be specified in the compilation as BOS uses the FORTRAN H inline logical functions.
4. It is assumed that at some later time a subroutine will be added to the package to allow updating of the fixed-pointer table being used by JADE (see JADE Note 24).

## Linking the BOS module with your program

### 1. NORD

Use the following command (once the NORD relocating loader NRL has been invoked).

```
LOAD <YOURPROG>, JADEBOS,FTNLIB.
```

This loads the required parts of BRF file (SYSTEM)JADEBOS-780721:BRF.

This BOS NORD file is compiled in LIBRARY MODE. This means that only the parts of the library actually called by <YOURPROG> will be loaded. In this way memory is economised in the NORD.

### 2. IBM 370/168 DESY

The BOS exists as two NEWLIB libraries

F11HUG.DS.JADEBOS.S	Fortran source
F11HUG.DS.JADEBOS.L	Load library

To link with the load library the following JCL statement is required  
//LKED.SYSLIB DD DSN = F11HUG.DS.JADEBOS.L,DISP=SHR

The NEWLIB structure ensures once again that only those parts of the BOS library invoked by your program will be loaded.

### 3. IBM 360/195 RUTHERFORD

The BOS exists at Rutherford as an 'Automatic Call' library

```
USER.XM65.JADEBOS
```

To link this load library with your program the following JCL statement is required

- a) If you are only linking with the BOS library and no others (apart from the standard libraries) you may use the symbolic parameters in the FORTRAN H procedure.

eg:

```
// EXEC FHCLG, <some parameters>, SYSLIB = 'USER.XM65.JADEBOS'
```

b) If you have more than 1 library to link with the L step must contain

```
//L.SYSLIB DD DSN = USER.XM65.JADEBOS,DISP=SHR
//          DD
//          DD
//          DD
//          DD DSN = OTHER.LIBRARY,DISP=SOMETHING
//          DD DSN = ANOTHER.LIBRARY,DISP=SOMETHING
etc.
```

The automatic call structure acts similarly to NEWLIB and ensures that only the required parts of the BOS library are loaded.