Internal Report
DESY F14-77/01
August 1977

B O S

Bank Organisation System,

Dynamic Storage Organisation with FORTRAN

by

V. Blobel

```
 BBBB     OOO    SSS
 B   B   O   O  S
 B   B   O   O  S
 BBBB    O   O   SSS
 B   B   O   O      S
 B   B   O   O      S
 BBBB     OOO   SSSS
```

BANK ORGANISATION SYSTEM - BOS -

DYNAMIC STORAGE ORGANISATION WITH FORTRAN

AUTHOR - V.BLOBEL
II.INSTITUTE F. EXP.PHYSIK
UNIVERSITY OF HAMBURG AND DESY

TABLE OF CONTENT — ENTRY NAMES

1. INTRODUCTION

LARGE AMOUNT OF DATA ARE TO BE ANALYSED IN HIGH-ENERGY PHYSICS EXPERIMENTS. DATA ON ONE EVENT CONSIST OF DATA FROM SEVERAL SOURCES, E.G. TRACK CHAMBER, SHOWER COUNTER AND SCALER. DURING THE ANALYSIS NEW DATA MAY BE ADDED BY COMPUTATION, E.G. DATA ON SINGLE TRACKS. ALSO DATA MAY BE COMPRESSED BY DATA REDUCTION PROGRAMS.
THE COMPLEXITY OF ANALYSIS DEPENDS VERY MUCH ON THE ORGANISATION OF THE DATA IN STORAGE AND ON MASS STORAGE DEVICES, E.G. TAPES. A FLEXIBLE DATA ORGANISATION SIMPLIFIES THE ANALYSIS.
THE SYSTEM DESCRIBED HERE IS DESIGNED ACCORDING TO THE NEEDS OF HIGH-ENERGY PHYSICS EXPERIMENTS. ALL DATA ARE STRUCTURED IN SO CALLED BANKS OF VARIABLE LENGTH. DATA IN ONE BANK REFER E.G. TO A SINGLE PARTICLE TRACK. A BANK HAS TWO IDENTIFIERS NA AND NR, CALLED NAME AND NUMBER. NORMALLY THE IDENTIFIER NA CONSISTS OF UP TO FOUR CHARACTERS (LEFT ADJUSTED), WHILE IDENTIFIER NR IS AN INTEGER.
ALL BANKS ARE STORED IN A SINGLE COMMON/BCS/IW(NSPACE). AN INDEX IND IS ASSIGNED TO EACH BANK, WHERE IW(IND)=NW CONTAINS THE NUMBER OF DATA WORDS IN THAT BANK. DATA ARE STORED IN IW(IND+1) TO IW(IND+NW). IN ADDITION THERE ARE THREE WORDS CONTAINING NAME, NUMBER AND A POINTER ACCORDING TO THE FOLLOWING SCHEMA.

I	IW(I)
IND - 3	NA (NAME)
IND - 2	NR (NUMBER)
IND - 1	POINTER TO NEXT BANK OF SAME NAME
IND	NW = NUMBER OF DATA WORDS IN THE BANK
IND + 1	1.DATA WORD
IND + 2	2.DATA WORD
...	...
IND + NW	LAST DATA WORD

NO LIMITATIONS EXIST FOR THE LENGTH OF A SINGLE BANK OR THE NUMBER OF BANKS, EXCEPT FOR THE TOTAL LENGTH OF ALL BANKS.
THE NUMBER OF DIFFERENT NAMES MAY NEVER EXCEED 200.
OUTPUT OF SETS OF BANKS IS SIMPLY DONE BY SPECIFYING ALL NAMES OF BANKS, BELONGING TO THE OUTPUT RECORD.
BANK RECORDS MAY CONTAIN MANY BANKS, BANK AFTER BANK WITHOUT GAPS.
THE MAIN ADVANTAGES OF THE SYSTEM ARE -

- EFFICIENT USE OF STORAGE
- EASY ADDITION OF NEW BANKS
- FLEXIBLE INPUT/OUTPUT
- PRINTOUT OF BANK STRUCTURE FOR TEST PURPOSES

LIBRARIES. THE SUBROUTINES OF THE BANK-SYSTEM ARE ON THE DESY-NEWLIB LIBRARIES
DSN=FIEBLO.DAHEP (LOAD)
AND DSN=FIEBLO.DAHEPS (SOURCE)
A ORDERED LIST OF ALL SUBROUTINES FOLLOWS.

STANDARD SUBROUTINES

BINT	BFMT					
BCRE	BDAR	BDEF	BDLM	BDLM	BDMP	BGAR
BLOC	BPOS	BPRM	BPUT	BREADC	BSAC	BWRITE
FSWRIT	IBLN	MOVERX	READFF	UCOPY2	UWP	

ALTERNATIVE SUBROUTINES

CCRE	CLOC	CREAD	CPOS	CCHL

SUBROUTINES FROM DESYLIB

ITRACE	NTIME	NDARG

APPLICATION SUBROUTINES

CCOR	DEFST	IAF	ITODA	PCOND	PCORR	PHIST
PTABL	PSTOR	PVERT	SORT4	UCOND	UHIST	UTABL
USTOS	VALLPTEXT					

COMPATIBILITY WITH NON-IBM FORTRAN. FOR ALL SUBROUTINES WITH NUMBERED RETURN STATEMENTS (RETURN 1, RETURN 2, STATEMENTNRS AS ARGUMENTS) THERE ARE ALTERNATIVES WITH FIRST LETTER C, WHICH AVOID THIS POSSIBILITY OF IBM-FORTRAN.

INITIALIZATION IS DONE BY THE FOLLOWING STATEMENTS.

```
COMMON/BCS/IW(NSPACE)
REAL RW(NSPACE)
EQUIVALENCE (IW(1),RW(1))
   WHERE NSPACE = INTEGER CONSTANT

          ------ ---- ---- ----
CALL BINT(NSPACE,NREC,NDMP,NADD)
```

WHERE NREC = MAX.NO OF WORDS IN A RECORD (1000)
NDMP = NR OF WORDS, PRINTED IN A DUMP (500)
NADD = NR OF WORDS FOR LOW PRICRITY BANKS (0)

NOTE. HERE AND IN THE FOLLOWING INPUT ARGUMENTS HAVE A LINE ABOVE, OUTPUT ARGUMENTS HAVE A LINE BELOW. ARGUMENTS, WHICH ARE CHANGED BY THE PROGRAM, HAVE A LINE ABOVE AND BELOW.

IF ILLEGAL POINTER ARE DETECTED BY THE BANK PROGRAM,

```
A DUMP CONTAINING RELEVANT STATUS INFORMATION AND
NDMP WORDS OF COMMON/BCS/ IS PRINTED.
NAMES WITH A CHARACTER '*' AS THE FOURTH CHARACTER
(E.G. HST* ) ARE CONSIDERED AS SPECIAL. THE TOTAL
SPACE ATTRIBUTED TO THEM NEVER EXCEEDS NADD WORDS.

----------+----------+----------2
```

2. CREATION OF BANKS

2.1 CREATE A BANK

```
                -- -- --
CALL BCRE(IND,NA,NR,NW,&S1,IER)
          ---           --- ---
```

WHERE NA = NAME OF THE BANK
NR = NUMBER OF THE BANK
NW = NUMBER OF WORDS OF THE BANK (NW.GE.0)
IND = INDEX OF BANK (IER=1, IF BANK ALREADY EXISTING)
ALL WORDS OF THE BANK ARE SET = 0
RETURN 1 (IER=2) IF NOT ENOUGH SPACE

```
ALTERNATIVE
CALL CCRE(IND,NA,NR,NW,IER)
```

TEST IER=2 EQUIVALENT TO RETURN 1

2.2 CHANGE LENGTH OF THE LAST CREATED BANK

```
          --
CALL BCHL(NW,&S1)
          ---
```

WHERE NW = CHANGE IN LENGTH OF THE BANK (NW.LT.0 VALID)

RETURN 1, IF NOT ENOUGH SPACE

```
ALTERNATIVE
CALL CCHL(NW,IER)
```

TEST IER=2 EQUIVALENT TO RETURN 1

NOTE, THAT ONLY THE LENGTH OF THE LAST CREATED BANK CAN
BE CHANGED DIRECTLY (SEE CHAPTER 2.4).

2.3 STORE DATA IN A BANK

```
          --- -- --
CALL BSTR(IND,JW,NW)
```

WHERE IND = INDEX
JW = ARRAY TO BE STORED
NW = LENGTH OF JW
JW(1) IS STORED IN IW(IND+1) ETC

2.4 MOVING A BANK


```
          ---
CALL BMVE(IND,&S1)
          --- ---
```

WHERE IND = INDEX OF BANK TO BE MOVED
THE BANK IS MOVED TO THE END OF THE USED STORAGE, IND
IS CHANGED. AFTER BMVE THE LENGTH OF THE BANK CAN BE CHANGED
BY BCHL.
RETURN 1, IF NOT ENOUGH SPACE

```
ALTERNATIVE
CALL CMVE(IND,IER)
```

TEST IER=2 EQUIVALENT TO RETURN 1

```
----------3
```

3. LOCATING BANKS

3.1 LOCATE A BANK

```
                -- --
CALL BLOC(IND,NA,NR,&S1)
          ---       ---
```

WHERE NA = NAME OF THE BANK
NR = NUMBER OF THE BANK
IND = INDEX OF THE BANK (NORMAL RETURN)
RETURN 1, IF BANK NOT EXISTING (IND=0)

```
ALTERNATIVE
CALL CLOC(IND,NA,NR)
```

TEST IND=0 EQUIVALENT TO RETURN 1

3.2 LOCATING ALL BANKS OF A GIVEN NAME

LOCATING OF ALL BANKS OF A GIVEN NAME IN ASCENDING ORDER
OF NR IS DONE IN THE FOLLOWING WAY.

```
          ----
CALL BPOS(NAME)
```

WHERE NAME = NAME OF THE BANKS

```
          ---
CALL BNXT(IND,&S1)
          --- ---
```

IND = INDEX OF NEXT BANK OF THE SAME NAME
RETURN 1 AFTER LAST BANK

```
ALTERNATIVE
CALL CPOS(NAME)
CALL CNXT(IND)
```

TEST IND=0 EQUIVALENT TO RETURN 1

```
EXAMPLE FOR A LOOP ON ALL BANKS OF THE SAME NAME
   CALL BPOS(NAME)
10 CALL BNXT(IND,&20)
   . . .
   GOTO 10
20 CONTINUE
```

BPOS/BNXT-LOOPS MAY BE NESTED. BNXT MAY ALSO BE USED WITHOUT BPOS TO OBTAIN THE INDEX OF THE NEXT BANK OF THE SAME NAME. THE INDEX IND IS A INPUT- AND OUTPUT-ARGUMENT IN BNXT, EXCEPT FOR THE FIRST CALL AFTER THE CALL OF BPOS, WHERE IT IS AN OUTPUT-ARGUMENT.
SEE ALSO REMARKS IN CHAPTER 12.

3.3 ALTERNATIVE TO BPOS/BNXT

LOCATING ALL BANKS OF A GIVEN NAME CAN ALSO BE DONE WITH BDAR.

```
CALL  BDAR(NAME,N,INDA,NLIM)
```

WHERE NAME = NAME OF THE BANKS
NLIM = NR.OF WORDS OF ARRAY INDA( )
INDA( ) = ARRAY CONTAING N INDICES (N.LE.NLIM)

```
EXAMPLE, EQUIVALENT TO THE ONE IN 3.2
   INTEGER INDA(20)
   CALL BDAR(NAME,N,INDA,20)
   DO 20 I=1,N
   IND=INDA(I)
   . . .
20 CONTINUE
```

4. INPUT FROM CARDS

DATA CAN BE READ FROM CARDS IN FREE FORMAT OR IN A FORMAT GIVEN ON THE CARD ITSELF, AND STORED IN BANKS. THE LAST 8 COLS OF THE CARDS ARE NOT USED IN FREE FORMAT MODE.

```
CALL BREADC
```

DATA ARE READ FROM CARDS (UNIT 5) UNTIL DATA END OR A CARD WITH ENDQ IS READ.

4.1 FREE FORMAT

READING A CARD OF THE TYPE
NAME A B C . . .
CREATES A BANK WITH NA = NAME, NR = 0. A SECOND CARD WITH THE SAME NAME CREATES A BANK WITH NA = NAME, NR = 1 ETC.

$NAM N A B C . . .
CREATES A BANK WITH NA = $NAM, NR = N. A SECOND CARD WITH THE SAME $NAM AND N IS NOT STORED.

THE BANK NAME NA CONSISTS OF THE FIRST FOUR CHARACTERS OF NAME OR $NAM, RESP. THE FIRST CHARACTER OF $NAM IS THE DOLLAR-SIGN. N, A, B, C, . . . ARE CONSTANTS, INTEGER CONSTANTS WRITTEN WITHOUT A DOT (.), REAL CONSTANTS WITH DOT (.) AND OPTIONALLY WITH EXPONENT. CONTINUATION CARDS STARTING IMMEDIATELY WITH CONSTANTS ARE ALLOWED. NON-NUMERIC CHARACTERS BETWEEN CONSTANTS ARE IGNORED.

4.2 FORMATED

AFTER READING A CARD
$NAM N NW 'FORMAT
NW WORDS ARE READ ACCORDING TO THE GIVEN FORMAT FROM CONSECUTIVE CARDS AND STORED IN A BANK WITH NA = $NAM, NR = N. THE FORMAT SHOULD START WITH 1X (SEE EXAMPLE), TO ALLOW ECHO-PRINTOUT WITHOUT LINE SKIPPING.

4.3 SPECIAL CARDS

ENDQ
BREADC RETURNS.

POFF
PRINTING OF CARDS IS SWITCHED OFF. IT MAY BE SWITCHED ON AGAIN BY THE CARD
PON

UNIT N
THE NEXT CARDS ARE READ FROM UNIT N UNTIL DATA END, THEN AGAIN FROM UNIT 5.

4.4 EXAMPLE FOR CARDS AND BANKS STORED

```
CONSTANTS 1.7 2.3 7 1.23E-4
$RUN 2769 1.520 16.0
0.7 2.0
*TIT 2000 5 '(1X,5A4)
 PI+ PI+ PI- PI- PI0
ENDQ
```

BANKS STORED

NAME	CONS	$RUN	*TIT
NUMBER	0	2769	2000
NR.OF WORDS	4	4	5
1.DATA WORD	1.7	1.520	PI+
2.DATA WORD	2.3	16.0	PI+
3.DATA WORD	7	0.7	PI-
4.DATA WORD	1.23E-4	2.0	PI-
5.DATA WORD			PI0

INDICES ARE OBTAINED BY
CALL BLOC(IND,'CONS',0,&S1)

```
CALL BLOC(IND,'$RUN',2769,&S1)
CALL BLOC(IND,'#TIT',2000,&S1)   RESP.
```

---5

5. DEFINING SETS OF BANKS

DEFINITION. THE SET B(LIST) IS THE SET OF ALL BANKS WITH ALL NAMES, APPEARING IN THE ARRAY LIST.

A SET B(LIST) IS DEFINED BY THE FOLLOWING STATEMENT.

```
      - ----
CALL BMLT(N,LIST)
```

WHERE LIST = ARRAY OF N NAMES

EXAMPLE

```
CALL BMLT(2,'HEADGEOM')
```

---6

6. PRINTING OF BANKS

6.1 PRINT A SINGLE BANK

```
      -- --
CALL BPRS(NA,NR)
```

WHERE NA = NAME
NR = NUMBER OF BANK TO BE PRINTED

6.2 PRINT A SET OF BANKS

```
CALL BPRM
```

ALL BANKS OF THE PREVIOUS DEFINED SET B(LIST) ARE PRINTED.

THE PRINTING FORMAT OF EACH WORD IS CHOSEN AUTOMATICALLY (TEXT, INTEGER, REAL), NOT ALWAYS CORRECT.

---7

7. DELETING BANKS

7.1 DELETE A SINGLE BANK

```
      -- --
CALL BDLS(NA,NR)
```

WHERE NA = NAME
NR = NUMBER OF BANK TO BE DELETED

7.2 DELETE A SET OF BANKS


```
CALL BDLM
```

ALL BANKS OF THE PREVIOUS DEFINED SET B(LIST) ARE DELETED.

A DELETED BANK IS NO LONGER ACCESSIBLE BY A CALL BLOC, ALTHOUGH IT IS STILL IN STORAGE UNTIL THE NEXT GARBAGE COLLECTION.

---8

8. GARBAGE COLLECTION

IN A GARBAGE COLLECTION THE ACTIVE (NOT DELETED) BANKS ARE SHIFTED IN STORAGE, IF THERE ARE BANKS DELETED.

```
CALL BGAR(IGA)
      ---
```

IGA = 0 NO GARBAGE COLLECTION DONE (IF NO BANKS WERE DELETED)
IGA = 1 GARBAGE COLLECTION DONE

```
      --
CALL BGAC(IGA,NW)
      ---
```

WHERE NW = NR.OF WORDS OF A NEW BANK
NO GARBAGE COLLECTION IS DONE, IF THERE IS SPACE ENOUGH FOR AN ADDITIONAL BANK WITH NW WORDS

WARNING. BECAUSE IN A GARBAGE COLLECTION, SOME BANKS (ALL BANKS BEHIND DELETED BANKS) ARE SHIFTED, THE INDICES OF THE BANKS CHANGE. AUTOMATIC GARBAGE COLLECTIONS ARE DONE IN THE INPUT-OUTPUT ROUTINES (SEE 9.)

---9

9. UNFORMATED INPUT/OUTPUT OF BANKS

9.1 WRITING SETS OF BANKS

```
      ---
CALL BWRITE(IUN)
```

ALL BANKS OF THE PREVIOUS DEFINED SET B(LIST) ARE WRITTEN ON UNIT IUN. BECAUSE THE BANKS HAVE TO BE STORED CONSEC. IN STORAGE BEFORE WRITING, THEY HAVE TO BE SHIFTED IN STORAGE. IF NECESSARY, A GARBAGE COLLECTION IS DONE.
THE BANKS ARE NOT WRITTEN, IF THE TOTAL RECORD LENGTH IS LARGER THAN NREC (SEE INTRODUCTION).

THE ORDER OF THE BANKS IN THE RECORD IS THE SAME AS WITHIN LIST, BANKS WITH THE SAME NAME IN ASCENDING ORDER OF NUMBER.
THE RECORD CONSISTS OF N + 1 WORDS, WHERE
FIRST WORD IN RECORD = N
NEXT WORD = NAME

```
NEXT WORD              = NUMBER OF FIRST BANK
NEXT WORD              = POINTER
NEXT WORD              = NR.OF WORDS OF THE FIRST BANK
ETC.
```

9.2 READING

```
        ---
CALL BREAD(IUN,&S1,&S2)
           --- ---
```

A RECORD IS READ FROM UNIT IUN (&S1 ERROR EXIT, &S2 END EXIT).
THE BANKS CONTAINED IN THE RECORD ARE STORED. IF BANKS WITH THE
SAME IDENTIFIERS ARE ALREADY EXISTING, THEY ARE DELETED.
A GARBAGE COLLECTION IS DONE AUTOMATICALLY.
IF THERE IS NOT ENOUGH SPACE FOR A RECORD OF NREC WORDS (SEE
INTRODUCTION), THE PROGRAM STOPS.

ALTERNATIVE
CALL CREAD(IUN,IER)
   TEST IER=1 EQUIVALENT TO RETURN 1
   TEST IER=2 EQUIVALENT TO RETURN 2

9.3 ALTERNATIVE FOR BWRITE/BREAD

INPUT-OUTPUT IS ALSO POSSIBLE WITH USER-WRITTEN SUBROUTINES IN
USER-CONTROLLED RECORDS.

```
CALL BOUTP(N,INIW)
         - ----
```

   THE N WORDS IW(INIW) TO IW(INIW+N-1) CONTAIN ALL BANKS
   OF THE PREVIOUS DEFINED SET B(LIST).
   N=0, IF RECORD TOO LARGE.

```
        - --
CALL BINP(N,AR)
```

   WHERE AR( ) = ARRAY OF N WORDS, CONTAINING BANKS.
   THE BANKS CONTAINED IN THE ARRAY ARE STORED.

THE FOLLOWING STATEMENTS ARE EQUIVALENT

```
OUTPUT       CALL BWRITE(IUN)

ALTERNATIVE  CALL BOUTP(N,INIW)
             IF(N.NE.0) WRITE(IUN) N,(IW(INIW+I-1),I=1,N)

INPUT        CALL BREAD(IUN,&S1,&S2)

ALTERNATIVE  READ(IUN,ERR=&S1,END=&S2) N,(AR(I),I=1,N)
```



```
             CALL BINP(N,AR)
```

---

10. SPECIAL SET OF BANKS

FOR THE USE IN AN EVENT-BY-EVENT PROCESSING A SPECIAL LIST
OF BANK NAMES IS SUPPORTED BY THE SYSTEM. THIS LIST CONTAINS
ALL NAMES OF BANKS BELONGING TO THE CURRENT EVENT. EACH
NAME HAS ASSOCIATED A MARKER, WITH A VALUE =1, IF BANKS
WITH THIS NAME SHOULD BE WRITTEN IN THE OUTPUT RECORD.
AFTER INPUT (BREAD/BINP) THE SPECIAL LIST CONSISTS OF
ALL NAMES OF BANKS JUST READ IN, WITH MARKER =1.
THE LIST CAN BE UPDATED DURING EVENT PROCESSING.

10.1 UPDATING THE SPECIAL LIST

```
          - ----
CALL BSAT(N,LIST)

          - ----
CALL BSAW(N,LIST)
```

   WHERE LIST( ) = ARRAY OF N NAMES

IN BOTH CASES THE NAMES ARE ADDED TO THE SPECIAL LIST
(IF NOT ALREADY PRESENT), AND THE MARKER IS SET
=0 FOR BSAT AND =1 FOR BSAW.

10.2 DEFINING THE SET OF BANKS

```
CALL BSLT

CALL BSLW
```

THE SET B(LIST) IS DEFINED AS THE SET OF ALL NAMES OF THE
SPECIAL LIST FOR BSLT, AND AS THE SET OF ALL NAMES OF THE
SPECIAL LIST WITH MARKER =1 FOR BSLW.

10.3 EXAMPLE

THIS EXAMPLE EXPLAINS THE TYPICAL INPUT/OUTPUT -
PROCEDURE IN AN EVENT-BY-EVENT PROCESSING. ONE EVENT IS
READ IN BY BREAD. BY COMPUTATION NEW BANKS ARE CREATED.
SOME OF THESES BANKS SHOULD BE WRITTEN , BUT ALL BANKS
HAVE TO BE DELETED AFTER PROCESSING THE EVENT.

```
C      INPUT
    10 CALL BREAD(IN-UNIT,&30,&40)
C
C      COMPUTATION
```

```
      . . .
      . . .
C
C      BANKS, WHICH ARE CREATED AND WHICH SHOULD NOT BE WRITTEN
      CALL BSAT(NT,NAMT)

C      BANKS, WHICH ARE CREATED AND WHICH SHOULLD BE WRITTEN
      CALL BSAW(NW,NAMW)

C     DECIDE ON OUTPUT FOR THIS EVENT
      IF(.NOT.OUTPUT) GOTO 20

C
C     OUTPUT
      CALL BSLW
      CALL BWRITE(OUT-UNIT)
C
C     DELETE ALL EVENT-BANKS
   20 CALL BSLT
      CALL BDLM
      GOTO 10
C     READ ERROR
   30 WRITE(. . .)
      GOTO 10
C     END OF DS
   40 WRITE(. . .)
      . . .
```

---11

11. PRINTING STATISTIC AND A DUMP

A FINAL STATISTIC ON THE SYSTEM PERFORMANCE IS PRINTED BY

```
CALL BSTA
```

A DUMP CONTAINING RELEVANT STATUS INFORMATION AND NDMP WORDS OF THE COMMON/BCS/IW(NSPACE) IS PRINTED BY

```
CALL BDMP
```

AND THE PROGRAM IS STOPPED.

---12

12. OPTIMIZATION

LOCATING BANKS BY CALLS TO BLOC AND BPOS/BNXT IS DONE BY FAST ALGORITHMS, BUT OF COURSE SOME TIME IS SPENT IN THESE SUBROUTINES. HOWEVER, CALLS TO BLOC AND BPOS/BNXT CAN BE REPLACED BY SOME FORTRAN-STATEMENTS, AVOIDING THE SAVING AND RESTORING OF REGISTERS ETC IN THE CALL/RETURN PROCEDURE.

THE METHOD USED FOR THE POINTERS TO BANKS IS AS FOLLOWS. AT ANY TIME THE FIRST 200 WORDS OF THE COMMON /BCS/ CONTAIN THE INDICES FOR THE FIRST BANK OF EACH NAME. THE ASSIGNMENT OF A NAME TO A LOCATION IS NEVER CHANGED DURING PROGRAM EXECUTION. NORMALLY THE ASSIGNMENT IS DONE IN THE ORDER OF APPEARENCE OF NAMES, BUT THE ASSIGNMENT CAN ALSO BE FIXED AFTER INITIALIZATION OF THE SYSTEM (AFTER CALL BINT) BY THE CALL

```
   CALL BDEF(N,LIST)
     WHERE LIST( ) = ARRAY OF N NAMES
```

THE FIRST NAME OF THE ARRAY (LIST(1)) IS ASSIGNED TO IW(1), THE SECOND TO IW(2) ETC.

EXAMPLE

```
   CALL BDEF(3,'HEADTITLDATA')
   CALL BDEF(2,'GEOMSHOW')
```

ASSUME THAT THE INDEX OF THE FIRST BANK WITH NAME 'GEOM' IS NEEDED. THIS INDEX IS STORED IN IW(4).

```
      IND=IW(4)
      IF(IND.EQ.0) GOTO 10
C     USE IND
      . . .
      GOTO 20
C     NO BANK WITH NAME 'GEOM' EXISTS
   10 CONTINUE
```

THE LOCATION ASSIGNED TO A NAME CAN ALSO BE OBTAINED USING THE FUNCTION IBLN.

```
   I=IBLN(NAME)
   -
   IW(I) CONTAINS THE INDEX OF THE FIRST BANK WITH THE
   THE GIVEN NAME OR 0, IF NO BANK IS EXISTING.
```

EXAMPLE

```
   I=IBLN('GEOM')
```

WOULD YIELD I=4 IN THE PREVIOUS EXAMPLE.

A FAST (HASH) ALGORITHM IS USED IN THE FUNCTION IBLN, WITH EXECUTION TIME NEARLY INDEPENDENT FROM THE TOTAL NUMBER OF EXISTING NAMES.

ALL BANKS OF THE SAME NAME ARE CONNECTED BY (FORWARD) POINTERS, STORED IN IW(IND-1) FOR A BANK WITH INDEX IND. THE ORDER OF THE BANKS IS IN ASCENDING NUMBER. FROM A GIVEN BANK WITH INDEX IND, THE NEXT BANK OF THE SAME NAME CAN BE OBTAINED BY THE FOLLOWING STATEMENT.

```
     IND=IW(IND-1)
     IF(IND.EQ.0) GOTO 10
C    USE IND
     . . .
     GOTO 20
```

```
C   NO FURTHER BANK OF THE SAME NAME
 10 CONTINUE
```

THUS LOCATING OF THE NEXT BANK OF A GIVEN NAME IS ALWAYS FAST, BUT E.G. LOCATING ONLY THE LAST OF MANY BANKS OF THE SAME NAME IS SLOW.

THE FOLLOWING EXAMPLE SHOWS THE REPLACEMENT OF THE BPOS/BNXT USE BY SOME FORTRAN-STATEMENTS.

EXAMPLE

```
    CALL BPOS(NAME)                  IND=IBLN(NAME)+1
 10 CALL BNXT(IND,&20)            10 IND=IW(IND-1)
    . . .                            IF(IND.EQ.0) GOTO 20
    . . .                            . . .
    . . .                            . . .
    GOTO 10                          GOTO 10
 20 CONTINUE                      20 CONTINUE
    . . .                            . . .
```

NOTE, THAT ALSO THE USE OF IBLN CAN BE AVOIDED BY USING BDEF AS EXPLAINED ABOVE.

---

13. APPLICATION PROGRAMS

THE SUBROUTINE DESCRIBED HERE USE BANK STORAGE OF LOW PRIORITY. THEY CAN BE USED FOR COUNTING, HISTOGRAMMING, STORING OF VECTORS ETC. AND MAY BE USEFUL IN THE TESTING PHASE OF LARGE PROGRAMS.
THE TOTAL NUMBER OF WORDS USED IN THE BANK COMMON /BCS/ CAN NEVER EXCEED NADD WORDS (NADD IS AN ARGUMENT OF BINT).
THE SUBROUTINES START WITH A LETTER U FOR A NEW ENTRY, AND WITH A LETTER P FOR PRINTOUT. PRINTOUT CAN ALSO BE DONE FOR ALL SUBROUTINES BY PALL.

```
CALL PALL
   PRINTOUT FOR ALL SUBROUTINES
```

THE ARGUMENT N IN THE CALLS SHOULD ALWAYS BE A POSITIVE INTEGER WITH LESS THAN 9 DIGITS.

13.1 COUNT IN A 32768 * 32768 ARRAY

```
        - -
CALL UCOND(I,J)
   INCREASE COUNT IN ELEMENT I,J BY ONE.
   0 .LE. I .LE. 32767  AND 0 .LE. J .LE. 32767

CALL PCOND
   PRINT ALL ELEMENTS .NE. 0. ALL BANKS USED ARE DELETED.
        - -
```



```
CALL QCOND(I,J,NR)
        --
   WHERE I,J = INDICES OF ELEMENT
         NR = NUMBER OF COUNTS IN ELEMENT
```

BANKS (CON ,IJ) OF LENGTH 8(+4) ARE USED, WHERE IJ IS CALCULATED FROM I AND J.

13.2 COUNT IN A 256 * 16 ARRAY

```
        - - -
CALL UTABL(N,I,J)
   INCREASE COUNT IN ELEMENT I,J BY ONE.
   0 .LE. I .LE. 255  AND  0 .LE. J .LE. 15.
   MAXIMUM COUNT IN ONE ELEMENT IS 65535.

        -
CALL PTABL(N)
   THE I/J ARRAY NUMBER N IS PRINTED. FOR N=0 ALL ARRAYS
   ARE PRINTED.

        - -
CALL QTABL(N,I,AR)
          --
   WHERE AR( ) = ARRAY OF LENGTH 16 CONTAINING THE COUNTS
                 IN THE I-ROW OF ARRAY NUMBER N.
```

A BANK (TAB ,N 16+I/16) OF LENGTH 128(+4) IS USED FOR EACH FILLED 16 16 SUBARRAY.

13.3 HISTOGRAM

```
        - -
CALL UHIST(N,X)
   ADD NEW ENTRY X INTO HISTOGRAM N WITH 100 BINS. THE BIN SIZE
   IS CHOOSEN AUTOMATICALLY, UNLESS DHIST IS USED.

        - -- --
CALL DHIST(N,XL,XH)
   FOR HISTOGRAM N THE VALUES XL AND XH ARE DEFINED AS LOWER
   AND UPPER LIMITS, RESP.. (MUST BE CALLED BEFORE UHIST).

        -
CALL PHIST(N)
   THE HISTOGRAM N IS PRINTED, TOGETHER WITH MEAN VALUES ETC.
   FOR N=0 ALL HISTOGRAMS ARE PRINTED.

        -
CALL QHIST(N,AR)
          --

   WHERE AR( ) = ARRAY OF LENGTH 115 FOR CONTENT OF
   HISTOGRAM N (SEE BELOW).

   I              AR(I)
```

1 . . . 100	CONTENT OF HISTOGRAM BINS
101	NR OF ENTRIES (TOTAL)
102	LOWEST BIN LEFT EDGE
103	BIN SIZE
104	NR OF ENTRIES OUTSIDE LOW
105	NR OF ENTRIES OUTSIDE HIGH
106	MINIMUM VALUE OF X
107	MAXIMUM VALUE OF X
108	MEAN VALUE
109	MEDIAN (50 PC VALUE)
110	STANDARD DEVIATION
111	. . . FROM QUANTILES
112	QUANTILES 16 PC
113	. . . 84 PC
114	. . . 2.3 PC
115	. . . 97.7 PC

A BANK (HST*,N) OF LENGTH 120(+4) IS USED FOR EACH HISTOGRAM.

13.4 CORRELATION PLOT

CALL UCORR(N,Y,X)
ADD NEW ENTRY Y,X TO CORRELATION PLOT N WITH
50 . 100 BINS. BIN SIZE IS ALWAYS CHOOSEN AUTOMATICALLY.

CALL PCORR(N)
THE CORRELATION PLOT N IS PRINTED ON ONE PAGE TOGETHER
WITH PROJECTIONS, MEAN VALUES AND THE CORRELATION PARAMETER.
BINS WITH CONTENT .NE. 0 ARE PRINTED AS A X. FOR N=0
ALL CORRELATION PLOTS ARE PRINTED.

A BANK (COR*,N) OF LENGTH 380(+4) IS USED FOR EACH
CORRELATION PLOT.

13.5 STORE VECTORS

CALL USTOR(N,AR,NDIM)
STORE VECTOR AR(1) . . . AR(NDIM) IN STORAGE N. THE
ARGUMENT NDIM SHOULD ALWAYS BE THE SAME FOR A GIVEN
STORAGE N.

NOTE. USTOR IS AN ENTRY OF USTOS. IF USTOS IS NOT CALLED,
USE A STATEMENT
EXTERNAL USTOS

CALL USTOS(N,A1,A2,A3,A4)
STORE VECTOR A1, . . . IN STORAGE N. THE NUMBER OF ELEMENTS
IN A VECTOR MAY BE BETWEEN 1 AND 4 (VARIABLE NR OF ARGUMENTS
ALLOWED), BUT SHOULD ALWAYS BE THE SAME FOR A GIVEN
STORAGE N.

THE NUMBER OF VECTORS STORED IN A STORAGE IS LIMITED TO 100.
THIS LIMIT CAN BE CHANGED BY DSTOR.

CALL DSTOR(LIM)
THE LIMIT ON THE NR OF VECTORS IS SET TO LIM.

CALL PSTOR(N)
THE VECTORS STORED IN STORAGE N ARE PRINTED. FOR N=0
THE VECTORS OF ALL STORAGES ARE PRINTED.

A BANK (STO*,N) IS USED FOR EACH STORAGE. THE STORAGE IS USED
AS FOLLOWS.

INDEX OF THE BANK	CONTENT
+1	NUMBER OF STORED VECTORS
+2	LENGTH OF ONE VECTOR
+3	(USED INTERNAL)
+4	1. ELEMENT OF 1. VECTOR
+5	2. ELEMENT OF 1. VECTOR
+6	ETC

13.6 COMMENT FOR APPLICATION PROGRAMS

CALL PTEXT
COMMENT IS PRINTED FOR ALL APPLICATION PROGRAMS

COMMENT IS DEFINED BY THE FOLLOWING CALLS.

CALL TCOND(N,'COMMENT $')
COMMENT FOR UCOND

CALL TTABL(N,'COMMENT $')
COMMENT FOR UTABL

CALL THIST(N,'COMMENT $')
COMMENT FOR UHIST

CALL TCORR(N,'COMMENT $')
COMMENT FOR UCORR

CALL TSTOR(N,'COMMENT $')
COMMENT FOR USTOR/USTOS

CALL TTEXT(N,'COMMENT $')
GENERAL COMMENT

THE COMMENT-TEXT IS DELIMITED BY A $ SIGN, MAXIMUM NUMBER
OF CHARACTERS IS 60.

A BANK WITH NAME 'TEX*' IS USED FOR EACH COMMENT.