HEWLETT-PACKARD GENERAL SYSTEMS USERS GROUP

FEBRUARY, 1980 MEETING

RESOURCE OPTIMIZATION SERIES

F A S T I O

Mike Vislosky

Mandate Corp.
1717 East 9th
Cleveland, Ohio 44114

IF YOU WOULD LIKE TO READ OR WRITE SEQUENTIAL FILES ON THE HP/3000 AT A RATE OF 60,000 TO 100,000 RECORDS PER MINUTE RATHER THAN A SLOW 8,500 RECORDS PER MINUTE, THEN THIS IS FOR YOU.

THIS SUBROUTINE CAN BE USED BY ANY COBOL PROGRAM WITH MINOR CHANGES. IT WAS DESIGNED TO READ BLOCKS OF DATA AND DO DEBLOCKING. SINCE MPE DOES DEBLOCKING INEFFICIENTLY, THE USE OF 'FASTIO' IN PROGRAMS WHICH DO ANY SIGNIFICANT AMOUNT OF I/O WILL RESULT IN SUBSTANTIAL SAVINGS. YOUR PROGRAMS RUN FASTER AND THE LOAD ON THE SYSTEM WILL BE REDUCED.

IT IS VERY EASY TO IMPLEMENT 'FASTIO' IN COBOL PROGRAMS. THE ONLY CHANGES NEEDED ARE BUFFER AREAS IN WORKING-STORAGE AND REPLACEMENT OF ALL READS AND WRITES WITH "CALLS" TO 'FASTIO'. IN ADDITION YOU MUST CHECK FOR END-OF-FILE WHEN READING OR SET END-OF-FILE WHEN WRITING OUTPUT FILES. IT IS NOT NECESSARY TO USE 'FASTIO' ON EVERY FILE IN THE PROGRAM.

THE THINGS THAT DO NOT CHANGE IN YOUR COBOL PROGRAM ARE:

1. SELECT STATEMENT
2. FD DECLARATION
3. OPEN STATEMENT
4. CLOSE STATEMENT

THE ONLY OTHER CHANGE THAT IS NECESSARY IS TO THE JCL. ALL FILE STATEMENTS FOR THOSE USING 'FASTIO' MUST HAVE 'NOBUF' ADDED TO THEM. IF YOU FAIL TO MODIFY THE JCL YOUR PROGRAMS WILL STILL RUN. HOWEVER, THE RESULTS WILL BE THE SAME AS BEFORE THE USE OF 'FASTIO' -- S L O W -- SO, DON'T FORGET THIS SMALL ITEM.

AFTER MODIFYING YOUR COBOL PROGRAM COMPILE INTO A USL FILE. NEXT, PREP THE USL FILE WITH THE RL FILE CONTAING 'FASTIO'. NOW YOU ARE READY TO RUN.

AS YOU CAN SEE IT IS QUITE EASY TO INSTALL THIS ROUTINE. YOU CAN LEAVE IN THE OLD COBOL READS AND WRITES IN CASE YOU WISH TO REMOVE 'FASTIO', JUST NOTE THEM OUT.

THE FOLLOWING IS A GUIDELINE FOR CHANGING A COBOL PROGRAM:

```
    SELECT IN-FILE  ASSIGN TO "DSKIN".
    SELECT OUT-FILE ASSIGN TO "DSKOUT".

FD  IN-FILE
    RECORD CONTAINS 80 CHARACTERS DATA RECORD IS IN-REC.
01  IN-REC          PIC X(80).

FD  OUT-FILE
    RECORD CONTAINS 80 CHARACTERS DATA RECORD IS OUT-REC.
01  OUT-REC         PIC X(80).

WORKING-STORAGE SECTION.
01  INBUF.                                                          (WORDS)
    05  I-REC-LNTH-IN-WORDS     PIC S9(4)  COMP  VALUE 40.          IBUF(0)
    05  I-EOF-FLAG              PIC S9(4)  COMP  VALUE  0.          IBUF(1)
    05  I-REC-DISP              PIC S9(4)  COMP  VALUE  0.          IBUF(2)
    05  FILLER                  PIC X(6).                           IBUF(3-4-5)
    05  FILLER                  PIC X(1280).       80 BLOCKED 16    IBUF(6-?)
    05  I-REC-CNT               PIC S9(8)  COMP.   NUM RECS IN FILE


01  OUTBUF.                                                         (WORDS)
    05  O-REC-LNTH-IN-WORDS     PIC S9(4)  COMP  VALUE 40.          OBUF(0)
    05  O-EOF-FLAG              PIC S9(4)  COMP  VALUE  0.          OBUF(1)
    05  O-REC-DISP              PIC S9(4)  COMP  VALUE  0.          OBUF(2)
    05  FILLER                  PIC X(6).                           OBUF(3-4-5)
    05  FILLER                  PIC X(1280).       80 BLOCKED 16    OBUF(6-?)

PROCEDURE DIVISION.
    MOVE           40 TO I-REC-LNTH-IN-WORDS.
    MOVE           40 TO O-REC-LNTH-IN-WORDS.
    MOVE         ZERO TO I-EOF-FLAG  I-REC-DISP.
    MOVE         ZERO TO O-EOF-FLAG  O-REC-DISP.
*   N O T E: ABOVE FIELDS MUST BE INITIALIZED EVERYTIME
*            A FILE IS OPENED.

    OPEN INPUT IN-FILE  OUTPUT OUT-FILE.
READ-LOOP.
*   READ IN-FILE    AT END GO TO EOJ.
*   N O T E: FOLLOWING SUBROUTINE CALL REPLACES COBOL READ.
    CALL "CREADU"  USING IN-FILE  IN-REC  INBUF  I-REC-CNT.
    IF I-EOF-FLAG = 1
       GO TO EOJ.
WRITE-RECORDS.
*   WRITE OUT-FILE.
*   N O T E: FOLLOWING SUBROUTINE CALL REPLACES COBOL WRITE.
    CALL "CWRITEU"  USING OUT-FILE  OUT-REC OUTBUF.
    GO TO READ-LOOP.
EOJ.
```

```
     COMPUTE O-EOF-FLAG = 1.
     CALL "CWRITEU"  USING OUT-FILE  OUT-REC OUTBUF.
     CLOSE IN-FILE  OUT-FILE.
     STOP RUN.

*****************************************************************
    ** THE FOLLOWING COMMENTS ILLUSTRATE USING
    ** CREADU WITH AN INPUT FILE CONTAINING
    ** 80 BYTE FIXED-LENGTH RECORDS, BLOCKED 16.
    **
    **IN-REC
    **  A. IS THE TARGET AREA FOR THE LOGICAL RECORD
    **       BEING READ.
    **  B. SHOULD BE 1 LOGICAL RECORD LONG.
    **  C. MAY BE IN THE FILE SECTION OR IN
    **       WORKING STORAGE.
    **
    **LAYOUT OF INBUF
    **  IBUF(0)=LOGICAL RECORD LENGTH (+, IN WORDS)
    **  IBUF(1)=EOF FLAG (1=EOF ENCOUNTERED)
    **  IBUF(2)=CURRENT RECORD DISPLACEMENT IN BLOCK
    **  IBUF(3)=BLOCK LENGTH READ BY LAST FREAD
    **  IBUF(4)=MAXIMUM EXPECTED FREAD TRANSFER LENGTH
    **  IBUF(5)=MPE FILE #
    **  IBUF(6) THRU ? IS THE BLOCK HOLD AREA
    **  I-REC-CNT IS USED TO STORE COUNT FROM DISC LABEL
    **
    **  IBUF(0) MUST BE INITIALIZED BY THE CALLING
    **    PROGRAM.
    **  IBUF(1) AND IBUF(2) MUST BE INITIALIZED TO 0
    **    BY THE CALLING PROGRAM.
    **
    ** * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
    ** THE FOLLOWING COMMENTS ILLUSTRATE USING
    ** CWRITEU WITH AN OUTPUT FILE CONTAINING
    ** 80 BYTE FIXED-LENGTH RECORDS, BLOCKED 16.
    ** THE LAST CALL TO CWRITEU, WITH EOF-FLAG = 1,
    ** POSTS THE LAST BLOCK (IF PARTLY FULL).
    ** NOTE THAT THE OUTFILE-REC PARAMETER IS IGNORED
    ** DURING THIS LAST CALL.
    **OUT-REC
    **  A. IS THE SOURCE AREA FROM WHICH THE LOGICAL
    **       RECORD WILL BE WRITTEN.
    **  B. SHOULD BE 1 LOGICAL RECORD LONG.
    **  C. MAY BE IN THE FILE SECTION OR IN
    **       WORKING STORAGE.
    **
```

```
**LAYOUT OF OUTBUF
**  OBUF(0)=LOGICAL RECORD LENGTH (+, IN WORDS)
**  OBUF(1)=EOF FLAG (1=POST LAST BLOCK)
**  OBUF(2)=CURRENT RECORD DISPLACEMENT IN BLOCK
**  OBUF(3)=(UNUSED BY CWRITEU)
**  OBUF(4)=MAXIMUM EXPECTED FWRITE TRANSFER LENGTH
**  OBUF(5)=.(0:15) IS MPE FILE #
**          .(15:1) IS CONTROL FOR FWRITE
**  OBUF(6) THRU ? IS THE BLOCK HOLD AREA
**
**  OBUF(0) MUST BE INITIALIZED BY THE CALLING
**    PROGRAM.
**  OBUF(1) AND OBUF(2) MUST BE INITIALIZED TO 0
**    BY THE CALLING PROGRAM.
**
```