Section Preview

- **I/O and AMODE 31**
  - **AMODE 31 I/O Issues**
  - **AMODE 31, RMODE 24**
  - **AMODE 31, RMODE ANY**

AMODE 31 I/O Issues

- **In general, programs that run in AMODE 31 give you the most flexibility**
  - **Historically, the biggest obstacle to doing this in Assembler has been I/O**
  - **DCB's contain 24-bit addresses and so must reside below the line (still true, will always be true)**
  - **At one time, if you were running AMODE 31 you had to switch to AMODE 24 to do your I/O, then switch back**
    - ✗ This is no longer the case
  - **Other issues include: do you want the buffers to be above the line (or below), do you want the record areas to be above or below the line**

- **Some secondary issues arise if an AMODE 31 program doing I/O is also to be reentrant:**
  - **Which control blocks must be in gotten storage**
    - ✗ In other words, which control blocks are normally modified by processing
      - ➤ These macros need to be coded using list and execute forms

- **Finally, since AMODE 31 is a given in this discussion, we need to examine the impact of RMODE 24 vs RMODE ANY**

AMODE 31 I/O Issues, continued

❒ **So we do a survey of the related macros, control blocks, and data areas for working with I/O in AMODE 31:**

♦ **DCB - must be below the line**

✗ If the program is RMODE 24, no problem, this is automatic

➤ However, if the program is to be reentrant, you must do a getmain and copy the DCB(s) into the getmained storage (DCBs are modified by OPEN and CLOSE at least)

✗ If the program is RMODE ANY, you must do a getmain and copy the DCB(s) into the getmained storage, whether the program is reentrant or not

➤ Furthermore, if a DCB contains an EODAD or SYNAD address, the DCB must reference a **DCBE** (Data Control Block Extension)

⇛ DCBE's may reside above or below the line: they contain 31-bit addresses and other parameters; generally they are not modified, so they do not have to be in a getmained area

⇛ A DCBE may contain the parameters EODAD= and / or SYNAD= (in which case the DCB should not have them), and possibly RMODE31=BUFF which requests OPEN obtain buffers above the line

⇛ To use a DCBE, a DCB must have the parameter DCBE=*label_on_DCBE*

⇛ Note that a DCBE may not be shared: each DCB that needs a DCBE must have its own DCBE

AMODE 31 I/O Issues, continued

- **So we do a survey of the related macros, control blocks, and data areas for working with I/O in AMODE 31, continued:**
  - **Buffers - default to below the line unless you use a DCBE with RMODE31=BUFF**
  - **Record areas - if in program, location is where program is**
    - ✗ If program is reentrant or you want record areas outside your program, use GETMAIN with LOC=({ANY|24|31|RES}); save the R1 value returned to you in a different register; can tie that register to a DSECT with a record layout if desired
  - **OPEN and CLOSE macros**
    - ✗ If program is reentrant or if running RMODE ANY, need to use list and execute forms and point to DCBs that were copied to getmained storage
    - ✗ Furthermore, if a program is running RMODE ANY, these macros must either be copied below the line, or if they are issued from above the line, they must also include MODE=31
  - **GET and PUT macros - no special concerns**

- **On the following pages are two sample programs with various mixes of these characterstics**
  - **Always assuming AMODE 31**

AMODE 31 / RMODE 24

- **Non-reentrant code, buffers and data records below the line - the classic style**
  - **Note that all the examples are a simple file to file copy for 100 byte records**

```
F2F      CSECT
F2F      AMODE 31
F2F      RMODE 24
         STM   14,12,12(13)
         LR    12,15
         USING F2F,12
         ST    13,SAVE+4
         LA    13,SAVE
         OPEN  (INDCB,,OUTDCB,(OUTPUT))
LOOP     DS    0H
         GET   INDCB,IN_REC
         PUT   OUTDCB,IN_REC
         B     LOOP
DONE     DS    0H
         CLOSE (INDCB,,OUTDCB)
         L     13,SAVE+4
         LM    14,12,12(13)
         SR    15,15
         BR    14
SAVE     DC    18F'0'
IN_REC   DS    CL100
INDCB    DCB   DDNAME=INDD,DSORG=PS,MACRF=GM,      X
               EODAD=DONE
OUTDCB   DCB   DDNAME=OUTDCB,DSORG=PS,MACRF=PM,    X
               LRECL=100,RECFM=FB
         END   F2F
```

AMODE 31 / RMODE ANY

❒ **With reentrant code, buffers and records above the line**

- ♦ **Need to GETMAIN all areas that may be changed (save area, DCBs, record area, and list form of OPEN / CLOSE parameters) and we put DCBs and OPEN / CLOSE macros lists below the line, everything else above the line**

```
F2F2    CSECT
F2F2    AMODE 31
F2F2    RMODE ANY
        STM   14,12,12(13)
        LR    12,15
        USING F2F2,12
* get save area and add to chain
        GETMAIN R,LV=72,LOC=(31)
        ST    13,4(1)
        LR    13,1
* get storage below the line and copy DCBs, etc.
        GETMAIN R,LV=SIZE_T,LOC=(24)
        LR    11,1
        USING WORKAREA,11
        MVC   IN(SIZE_INDCB),INDCB
        MVC   OUT(SIZE_OUTDCB),OUTDCB
        MVC   OPENWK(SIZE_OPN),OPENS
        MVC   CLOSWK(SIZE_CLO),CLOSES
* get storage for records
        GETMAIN R,LV=100,LOC=(31)
        LR    3,1
* OPEN; note MF parameter
        OPEN  (IN,,OUT,(OUTPUT)),MF=(E,OPENWK)
* main logic
LOOP    DS    0H
        GET   IN,(3)
        PUT   OUT,(3)
        B     LOOP
```

- more -

AMODE 31 / RMODE ANY

❒ **Reentrant code, buffers and records above the line, continued**

```
DONE     DS    0H
         CLOSE (IN,,OUT),MF=(E,CLOSEWK)
         LR    1,3
         FREEMAIN R,LV=100,A=(1)
         LR    1,11
         FREEMAIN R,LV=SIZE_T,A=(1)
         LR    1,13
         L     13,4(1)
         FREEMAIN R,LV=72,A=(1)
         LM    14,12,12(13)
         SR    15,15
         BR    14
* work areas and control blocks
INDCB    DCB   DDNAME=INDD,DSORG=PS,MACRF=GM,      X
               DCBE=IN_DCBE
SIZE_INDCB  EQU   *-INDCB
OUTDCB   DCB   DDNAME=OUTDCB,DSORG=PS,MACRF=PM,    X
               LRECL=100,RECFM=FB,DCBE=OUT_DCBE
SIZE_OUTDCB EQU   *-OUTDCB
OPENS    OPEN  (,,),MF=L
SIZE_OPN EQU    *-OPENS
CLOSES   CLOSE (,,),MF=L
SIZE_CLO EQU    *-CLOSES
SIZE_TOT EQU    *-INDCB
IN_DCBE  DCBE  RMODE31=BUFF,EODAD=DONE
OUT_DCBE DCBE  RMODE31=BUFF
WORKAREA DSECT
IN       DS    XL(SIZE_INDCB)
OUT      DS    XL(SIZE_OUTCDB)
OPENWK   DS    XL(SIZE_OPN)
CLOSWK   DS    XL(SIZE_CLO)
F2F2     CSECT
         END   F2F2
```

- ♦ **Notice how we capture the size of control blocks, parameter lists, and the whole gotten area using EQUs**