Legac-E Education

Using MODESET in COBOL

Most application programs executing under z/OS do so in Problem State and hence have no privileges. The question was raised in LinkedIn as to how to enter Supervisor state in COBOL with:

```
        MODESET KEY=ZERO,MODE=SUP
```

As **MODESET** is a macro this cannot be done in COBOL directly, but of course it is possible to CALL an Assembler routine which exploits the macro. Having made suggestions it was decided to code and test a routine to prove the premise. In fact two versions were produced one for Static Linkage and the other for Dynamic Linkage, the latter being made re-entrant so that it could be used in multiple address spaces concurrently if desire.

Note that programs using **MODESET** need to be authorized which means:

- They have to be Link-Edited with the **AC=1** parameter
- They have to reside in an Authorised Library
- Any concatenation of that library with a non-Authorised library revokes authorization.

Failure to comply with any of the above requirements will result in an S047 ABEND.

The Dynamically Linked routine was made re-entrant to allow it to be placed in the Link Pack Area (LPA) or Extended LPA (ELPA). This facilitates a single copy of the module being shared across multiple address spaces concurrently. If the module is not in LPA/ELPA a fresh copy will be loaded into the Job Pack Area (JPA) of each address spaces that invokes the module.

In order to prove that **MODESET** had been issued successfully the **MGCRE** macro was included to issue a z/OS Command. The code associate with **MGCRE** is highlighted in red and is not relevant to the original question.

The relevant code follows and includes the **MODESET** routine and its caller.

Using MODESET in COBOL

Statically Linked version

The MODESET routine

The original question implied tha the routine should be **AMODE(31)** and **RMODE(24),** but rather than change the COBOL defaults this module was compiled with both **AMODE** and **RMODE** set to **31**. This is probably best in any event as it avoids potential storage constraint below the 16MB boundary.

```
MODESUB  CSECT
MODESUB  AMODE 31
MODESUB  RMODE 31
         SAVE  (14,12)               SAVE REGISTERS EXCEPT R13
         BASR  12,0                  SET UP MY
         USING *,12                         BASE REGISTER
         ST    13,MYSAVE+4           PERFORM SAVE
         LA    13,MYSAVE             R13 = MY SAVE AREA ADDR.
         L     3,0(1)                R3  = PARAM ADDRESS
         L     4,4(1)                R4  = COMMAND ADDRESS
         CLC   =C'PROB',0(3)         IS IS PROBLEM REQUEST
         BE    SETPROB               YEP - SET PROBLEM STATE
         MODESET KEY=ZERO            SET SUPERVISOR STATE
         MGCRE TEXT=(4),CONSID=THISCON,MF=(E,KCOM)
         B     EXIT                  GO TO FINISH
SETPROB  MODESET KEY=NZERO           SET PROBLEM STATE
EXIT     L     13,MYSAVE+4           R13 = PREV SAVE ADDR
         RETURN (14,12),RC=0         RETURN TO CALLER
         LTORG
MYSAVE   DS    18F                   MY SAVE AREA
THISCON  DC    AL4(0)
KCOM     MGCRE MF=L
         END
```

Using MODESET in COBOL

The Caller

```
 IDENTIFICATION DIVISION.
 PROGRAM-ID.                MAINRTN.
 AUTHOR.                    T.R.SAMBROOKS.
     INSTALLATION.
     DATE-WRITTEN.          30TH AUG 2018.
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 77  THIS-MODE              PIC X(4) VALUE 'SUP '.
 01  CMND-AREA.
     03  COMM-LEN           PIC 9(4) COMP VALUE 126.
     03  THE-COMMAND        PIC X(125) VALUE
         'D IPLINFO'.
     03                     PIC X VALUE SPACE.
 PROCEDURE DIVISION.
     CALL 'MODESUB'         USING THIS-MODE CMND-AREA.
     MOVE 'PROB'            TO THIS-MODE.
     CALL 'MODESUB'         USING THIS-MODE CMND-AREA.
     GOBACK.
*------------------------------------------------------*
* The physical end of the program - MAINRTN            *
*------------------------------------------------------*
```

Dynamically Linked Version

The MODESET Routine

```
         PRINT NOGEN
MODESUB  CSECT
MODESUB  AMODE 31
MODESUB  RMODE 31
         BAKR  14,0                    SAVE REGISTERS EXCEPT R13
         BASR  12,0                    SET UP MY
         USING *,12                           BASE REGISTER
         B     PASSNAME
         DC    C'MODESUB '
PASSNAME L     2,0(1)                  R2  = PARAM ADDRESS
         L     4,4(1)                  R4  = COMMAND ADDR
         LA    3,AREASIZE              R3  = WORK AREA SIZE
         STORAGE OBTAIN,LENGTH=(3),ADDR=(6),SP=127
         USING SAVEAREA,6
         ST    13,MYSAVE+4             PERFORM SAVE
         LR    13,6                    R13 = MY SAVE AREA ADDR.
         MVC   KCOM(KCOMLEN),KCOMDEF
         CLC   =C'PROB',0(2)           IS IS PROBLEM REQUEST
         BE    SETPROB                 YEP - SET PROBLEM STATE
         MODESET KEY=ZERO              SET SUPERVISOR STATE
         MGCRE TEXT=(4),CONSID=THISCON,MF=(E,KCOM)
         B     EXIT                    GO TO FINISH
SETPROB  MODESET KEY=NZERO             SET PROBLEM STATE
EXIT     LA    3,AREASIZE              R3  = WORK AREA SIZE
         STORAGE RELEASE,LENGTH=(3),ADDR=(6),SP=127
         LA    15,0                    SET COND CODE 0000
         PR                            RETURN TO CALLER
         LTORG
THISCON  DC    AL4(0)
KCOMDEF  MGCRE MF=L
KCOMDEFE DS    0C
KCOMLEN  EQU   KCOMDEFE-KCOMDEF
```

Using MODESET in COBOL

```
*------------------------------------------------------------*
SAVEAREA DSECT                                               *
MYSAVE   DS    0F                     MY SAVE AREA           *
         DS    AL4(0)                 RESERVED               *
PRESAVE  DS    AL4(*-*)               CALLER'S SAVE ADDR     *
NEXTSAVE DS    AL4(*-*)               OUR SAVE AREA ADDR     *
         DS    15AL4(*-*)             GPRS 14 THRU 12        *
SAVEEND  DS    0C                     END OF SAVE AREA       *
KCOM     MGCRE MF=L                                          *
KCOME    DS    0C                    END OF MGCRE AREA       *
AREASIZE EQU   KCOME-SAVEAREA        SIZE OF SAVE AREA.      *
*------------------------------------------------------------*
MODESET  RSECT
         END
```

The Caller

```
 IDENTIFICATION DIVISION.
 PROGRAM-ID.                MAINRTN.
 AUTHOR.                    T.R.SAMBROOKS.
     INSTALLATION.
     DATE-WRITTEN.          30TH AUG 2018.
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 77  THIS-MODE              PIC X(4) VALUE 'SUP '.
 77  MODE-PGM               PIC X(8) VALUE 'MODESUB '.
 01  COMMAND-AREA.
     03  COMM-LEN           PIC 9(4) COMP VALUE 0.
     03  THE-COMMAND        PIC X(125) VALUE
         'D IPLINFO'.
     03                     PIC X VALUE SPACE.
 PROCEDURE DIVISION.
     MOVE LENGTH OF THE-COMMAND TO COMM-LEN.
     CALL MODE-PGM          USING THIS-MODE
                                  COMMAND-AREA.
     MOVE 'PROB'            TO THIS-MODE.
     CALL MODE-PGM          USING THIS-MODE
                                  COMMAND-AREA.
     GOBACK.
*------------------------------------------------------*
* The physical end of the program - MAINRTN            *
*------------------------------------------------------*
```