Utilizing a program loaded into the user program area to load another module in the same user program area

Carlo Bianchi and Giorgio Calzolari and Paolo Corsi

IBM Scientific Center, Pisa, Italy

September 1976

Online at https://mpra.ub.uni-muenchen.de/23062/
MPRA Paper No. 23062, posted 5. June 2010 18:57 UTC
It is known [*] that a program loaded into the User Program Area can load, via SVC 202, only programs to be allocated in the Transient Program Area and not programs to be allocated in the same User Program Area.

For example, a user program can call system functions like RENAME, LISTFILE, ERASE, etc. (hold in files MODULE loaded, one by one, in the Transient Program Area) but cannot load other functions like EDIT, COPYFILE, SORT, etc. (hold in files MODULE which are loaded in the User Program Area and therefore overlap the calling program, so that a correct reentry is impossible). To allow any program to use also this second type function, the following procedure can be adopted:

1. The main program (MAIN) is loaded not at the HEX 20000 but, at a higher address, for example:

   LOAD MAIN (ORIGIN 50000)

2. The program (or a subroutine, at any level) saves the content of the following addresses of the NUCON table of CMS:

   MAINHIGH
   MAINSTRT
   MAINLIST
   FRELOWE

3. Two addresses are modified. The load address of the main program, eventually rounded at integer page address, is stored at FRELOWE location, and an address of a few pages lower (say 5 for example) is stored at MAINHIGH location.

4. An SVC 202 is issued to load the required MODULE (for example, EDIT of a file); the loading is executed starting from the beginning of the User Program Area (20000 hex). The loaded MODULE does not overlap the calling program. If there is not enough space, the system issues the usual error message, and control is correctly returned to the calling user program.

5. At the exit of the MODULE (FILE or QUIT subcommands for EDIT), the user program takes the control again and restores in NUCON the four addresses previously saved, in order to allow the regular development of eventual GETMAIN or FREMAIN operations. In one example, the main program is in FORTRAN, and must be loaded at Hexadecimal address 50000.
The operations at point 2 are performed by the subroutine SAVNUC. The operations at point 3 are performed by the subroutine CNGNUC. Subroutine CMS (not printed here) simply performs the SVC 202 operation (point 4). The operations at point 5 are performed by the subroutine RESNUC.

C FORTRAN MAIN PROGRAM (OR SUBROUTINE AT ANY LEVEL)

DATA LOADAD /Z00050000/

CALL SAVNUC
CALL CNGNUC (LOADAD)
CALL CMS('EDIT ','FILE ','FT01F001','A1 ')
CALL RESNUC

END

* ASSEMBLER MAIN STORAGE MANAGEMENT SUBROUTINES

CSECT
ENTRY SAVNUC
ENTRY CNGNUC
ENTRY RESNUC
USING *, 12
USING NUEN, 0

SAVNUC STM 14,12,12(13)
LR 12,15
PRINT NOGEN
MVC OLFREELW(4),FREELOWE
MVC OLMAINHI(4),MAINHIGH
MVC OLMAINST(4),MAINSTR
MVC OLMAINLS(4),MAINLIST
LM 14,12,12(13)
BR 14
DS OD

CNGNUC STM 14,12,12(13)
LA 12,CNGNUC-SAVNUC
SR 15,12
LR 12,15
L 2,0(1)
L 2,0(2)
N 2,PAGE
ST 2,FREELOWE
S 2,FIVEPAGE
ST 2,MAINHIGH
UTILIZING A PROGRAM LOADED INTO THE USER PROGRAM AREA TO LOAD ANOTHER
MODULE IN THE SAME USER PROGRAM AREA — Continued

LM 14,12,12(13)
BR 14
DS OD
RESNUC STM 14,12,12(13)
LA 12,RESNUC-SAVNUC
SR 15,12
LR 12,15
MVC MAINHIGH(4),OLMAINHI
MVC MAINSTRT(4),OLMAINST
MVC MAINLIST(4),OLMAINLS
MVC FREELWE(4),OLFREELW
LM 14,12,12(13)
BR 14
DS OD
OLFREELW DS 1F
OLMAINHI DS 1F
OLMAINST DS 1F
OLMAINLS DS 1F
FIVEPAGE DC XL4'000005000'
PAGE DC XL4'FFFFF000'
NUCON
END