Author(s) : R. Matthews, A. Cass
Submitter :
Language : Assembler
Library: KERNLIB, IBM only
Submitted: 01.02.1983
Revised: 19.07.1988

Intercept a Fortran Abend on IBM

ABUSER enables a user-supplied subroutine to receive control when the user's program abends. A call to ABUSER identifies the user-supplied subroutine which is to receive control. The identified subroutine will be called if the user's program abends and can perform pre-termination processing such as printing summaries or plotting histograms.

Structure:

SUBROUTINE subprogram
User Entry Names: ABUSER

Usage:

```
CALL ABUSER(NAME)
```

NAME Name of a user-supplied SUBROUTINE subprogram declared EXTERNAL in the calling program.

This subprogram receives control via a call of the form

```
CALL NAME(KODES,KODEU)
```

KODES A 4-byte integer containing, if available, the system completion code as hexadecimal number (use Z format for printing).

KODEU A 4-byte integer containing, if available, the user completion code as integer number (use I format for printing).

Restrictions:

This subprogram is compiler and system dependent.

MVS:

The Fortran 4 version relies on modifications to the IBM H-extended compiler library and is therefore not portable. The Fortran 77 version uses a standard interface into the FACOM compiler library.

CMS:

The subprogram is compiler independent but KODES and KODEU are not available and so are set to zero. Note that the routine uses storage in the CMS nucleus – the NUSERFWD field and also 8-bytes at NCCOPYR – which must not be overwritten. (No other CERN Library routine uses these locations.)

Notes:

ABUSER can be called at any time during normal processing, (i.e. before an abend occurs), to re-specify the name of the user-supplied subroutine. Alternatively, the effect of previous calls can be cancelled by CALL ABUSER(0). A call to ABUSER after an abend will have no effect.

A secondary abend which occurs while the user is processing the primary abend will cause program termination.

Under MVS the user-supplied subroutine will not receive control for the following completion codes:

`122`	– job cancelled with dump
`222`	– job cancelled
`322`	– cpu time exceeded
`522`	– wait time limit exceeded

Examples:

In the following example, `ABUSER` is called to identify a subroutine called `FATAL` as the subroutine which is to receive control when the user's program abends. If an abend occurs, subroutine `FATAL` will be called and will print the completion codes and then call `HISTDO` to plot histograms.

```
EXTERNAL FATAL
...
CALL ABUSER(FATAL)
...
END
SUBROUTINE FATAL(KODES,KODEU)
WRITE(6,'(1X,''PROGRAM ABENDING WITH CODES '',Z3,I5)') KODES,KODEU
CALL HISTDO
RETURN
END
```

•