

NAME

UNCTLD – Reestablish unit 5 as standard-in after ^D.

SYNOPSIS

CALL UNCTLD(RC)

RC is the INTEGER*4 return code; see table below.

DESCRIPTION

When Fortran unit 5 is defaulted to the keyboard, it is actually attached to a device having a name like /dev/pts/0. Sometimes it is desired to take an END= exit from READ when the user sends ^D from the keyboard, perform some action, and eventually resume reading input from unit 5. Unfortunately, if a ^D is sent the unit remains attached to the keyboard device but subsequent attempts to read from it might elicit the Fortran runtime error message "Sequential READ not allowed after EOF." To reenale unit 5 for input from the keyboard, it is necessary to close the unit and reopen it (presumably after doing something in response to the EOF). This routine begins by finding the name of the file or device to which unit 5 is attached.

If unit 5 is redirected or piped to by the shell, this routine closes the unit and returns RC=3.

If unit 5 is already closed, this routine does nothing but return RC=2.

If unit 5 is attached to a file and the end of the file has been reached, this routine closes the unit and returns RC=1.

Otherwise the routine closes unit 5, reopens it on the device to which it was previously attached, and returns RC=0.

UNITS and DEVICES

5 unit to be closed and reopened
 /dev/ prefix of a device name associated with the keyboard
 stdin device name associated with a redirection or pipe

DIAGNOSTICS

On output these are the possible RC values:

0 all went well; unit 5 was closed and reopened on the keyboard
 1 unit 5 had been attached to a file; the unit was closed
 2 unit 5 had been already closed; nothing was done
 3 unit 5 had been redirected or piped; the unit was closed

LINKAGE

gfortran source.f -L\${HOME}/lib -lmisc

AUTHOR

Michael Kupferschmid

EXAMPLE

```

INTEGER*4 RC
2 READ (5,*,END=1) I
  PRINT *, I
  IF(I.EQ.0) STOP
1 PRINT *, 'doing the EOF processing...done'
  CALL UNCTLD(RC)
  IF(RC.EQ.0) GO TO 2
  PRINT *, 'RC=',RC
  STOP
END

```

This program produced the following exchange:

```

unix[1] a.out
1
          1
2
          2
^D
doing the EOF processing...done
3
          3
0
          0
unix[2] echo 4 | a.out
          4
doing the EOF processing...done
RC=          3
unix[3]

```

The user entered 1, which was printed; then the user entered 2, which was printed. Then the user entered ^D, causing the END=1 branch to be taken, so the program printed the message and called UNCTLD to reset unit 5 to the keyboard. Unit 5 had been attached to the keyboard, so RC returned as 0 and the program resumed prompting for inputs. The user entered a 3, which was printed, and then 0 to stop the program. Then the program was rerun with its input piped from echo. It read the input number 4 and printed it, then detected the end of the input stream and took the END=1 branch. The program printed the message and called UNCTLD to reset unit 5, but because unit 5 was attached to a pipe rather than the keyboard RC returned as 3. The program wrote out that value of RC and stopped.