Subject: Exposition of command-line argument access
From: Van Snyder

1 Introduction

The discussion in section 11.1.1 of main program arguments depends on examples. I think it would be better if the examples weren't normative. Also, the term "line" isn't appropriate in every environment. If there's no objection to normative text depending on examples, and use of the term "line," then this paper is moot.

The intent is to re-cast the exposition without changing the meaning.

2 Edits

Edits refer to 00-007. Page and line numbers are displayed in the margin. Absent other instructions, a page and line number or line number range implies all of the indicated text is to be replaced by immediately following text, while a page and line number followed by + indicates that immediately following text is to be inserted after the indicated line. Remarks for the editor are noted in the margin, or appear between [and] in the text.

[Editor: Insert "argument" after "program". After "significant" add "The use of each main program argument is determined from its type and rank."] 236:5

11.1.1.1 The command program argument 236:8-38

The command program argument, if present, shall be a default character scalar with assumed length.

The value of the command program argument is a processor-dependent representation of the entire command that invoked the program.

11.1.1.2 The argument text program argument

The argument text array program argument, if present, shall be a default character array of assumed size with an explicit lower bound of zero, and assumed length.

The argument text array argument program contains the text of the command. The zero'th element of the array contains the command name. The i'th element contains the i'th argument after the command name.

11.1.1.3 The argument length program argument

The argument length program argument, if present, shall be a default integer array of assumed size with an explicit lower bound of zero.

The value of each element of the argument length array argument is the length of the significant character string in the corresponding element of the argument text program argument, if the argument text program argument is present, or that would be in the corresponding element of the argument text program argument, if it is absent.

Any characters in the i'th element of the argument text program argument beyond the position given by the i'th element of the argument length program argument shall be blanks.

The value of an element of the argument length program argument shall not be greater than the length of the argument text program argument.

Note 11.3

> The processor is expected to select a suitable length for the elements of the argument text array, such that all characters specified for the command arguments are present in the argument text array. Similarly, the processor is expected to select a suitable length for the command program argument, such that the entire command is present.

If the argument length array and the argument text array are both present, they shall have the same bounds.

A value of zero for the zero'th element of the argument length array indicates that the processor could not obtain the command name. A value of zero for any other element of the argument length array indicates that the corresponding command argument was not present, had zero length, or could not be obtained by the processor.

Note 11.3$\frac{1}{2}$

Example:

```
PROGRAM MAIN ( command_line, argument_text, argument_length )
  USE ISO_FORTRAN_ENV, ONLY: u => error_unit
  CHARACTER(LEN=*) :: command_line
  CHARACTER(LEN=*), DIMENSION(0:) :: argument_text
  INTEGER, DIMENSION(0:) :: argument_length
  INTEGER :: i
  LOGICAL :: test

  do i = 0, ubound(argument_text)
    test = argument_text(i)(argument_length(i)+1:) /= ' '
    if ( test ) exit
  end do
  if ( size(argument_text) /= size(argument_length) .or.      &
       any(argument_length > len(argument_text) ) .or. test ) &
    write (u,*) 'Processor is not standard-conforming'
END  PROGRAM MAIN
```