From: Andreas Muenchow <Muenchow@udel.edu>
Subject: **Command Line Arguments (modern fortran)**
Date: June 21, 2012 2:19:17 PM EDT
To: Allison Einolf <aeinolf@macalester.edu>
Cc: Julie Jones <jkjones@UDel.Edu>, Ryan Pat <patriciaryan2@gmail.com>, Huntley Helga <helgah@udel.edu>

Hi all:

The following snipet of Fortran code is the way to incorporate command-line arguments into fortran

```
!
! This is a comment line in modern Fortran
!
!      Declaring variables used
!
       character (len=30) :: argument
       integer idev,iarea,nbytes250
!
! Read in command-line argument #1 into variable idev
!
       call get_command_argument(1,argument)
       read(argument,*)idev
!
! Read in command-line argument #2 and #3 into variables iarea and nbytes250
!
       call get_command_argument(2,argument)
       read(argument,*)iarea
       call get_command_argument(3,argument)
       read(argument,*)nbytes250
```

The executable containing this code snipet could be called from a shell environment like

```
        ls -la dump250 | nawk '{print $5}' >out.dat
         set out = `cat out.dat`
         ./xMODIS $idev $iarea $out
```

where I pass the number of bytes of a file from a directory listing (ls -la) into a file (out.dat) that is subsequently put into an environmental variable (set out) that is passed as a command-line argument ($out) into executable (xMODIS) that contains the fortran snipet above ... along with other parameters set within the shell-environment ($dev, $iarea).

The Fortran command "get_command" is a built-in Fortran function the same way that "read" or "write" are.

andreas

P.S.: This is about as advanced as I ever got with Fortran or csh-programming ...

Andreas Muenchow mail: muenchow@udel.edu
Associate Professor web: http://muenchow.cms.udel.edu
University of Delaware blog: http://IcySeas.org
302-831-0742