Fortran: A Dr. Jekyll and Mr. Hyde Story

Craig Rasmussen

Los Alamos National Laboratory

Fortran is the Language of Science

- Fortran is still popular in the scientific community
  - 19 out of 20 SciDAC I applications were in Fortran
  - 85% of ORNL Cray cycles are in Fortran
  - All of the major codes at LANL are in Fortran
  - 58% of DOD applications are in Fortran

MPI Fortran Bindings are Problematic

- Fortran 77 (implicit interfaces)
  - rather frightening warnings in the standard about portability
  - no defined interface so not type safety
    - what happens when error return value is forgotten?
  - requires wrappers to get to C
- Fortran 90 (explicit interfaces)
  - type safety
    - but how safe is void* anyway
    - requires countless TKR (type-kind-rank) interfaces
  - anyone using these?

Proposal: A New Fortran MPI Binding Based on the Fortran 2003 C Interoperability Standard

- Fortran now has a well-defined mechanism for interoperating with C
- Programmer defines a BIND(C) interface in Fortran
  - implementation can be in either Fortran or C
  - can be called from either Fortran or C

- Fortran now has types equivalent to many C primitives
  - integer (kind=C_INT, C_INT_32T...)
  - type(C_PTR)
  - CLOC function
  - value (pass by) attribute

There Are Unresolved Issues

- What type should an MPI handle be?
  - an integer (MPI_HANDLE_KIND)
    - could be of size to hold a C pointer
  - an opaque type (MPI_HANDLE_TYPE)

Example Usage

```
! import C interop and MPI interfaces

    use, intrinsic :: ISO_C_BINDING
    use :: MPI3

    integer(MPI_INT_KIND) :: rank, size, err

! MPI_Init example

    character(len=:, kind=C_CHAR), allocatable :: args(:)
    integer :: n_args

    err = MPI_Init(n_args, args)

    err = MPI_Comm_rank(MPI_COMM_WORLD, rank)
    err = MPI_Comm_size(MPI_COMM_WORLD, size)

! MPI_Send and MPI_Recv examples

    integer(MPI_INT_KIND) :: next, tag

    err = MPI_Send(C_LOC(message), 1, MPI_INTEGER, next, tag, MPI_COMM_WORLD)
    err = MPI_Recv(C_LOC(message), 1, MPI_INTEGER, prev, tag, MPI_COMM_WORLD, MPI_STATUS_IGNORE)

    err = MPI_Finalize()
```