Chapter 16

Language Bindings

16.1 Existing Fortran Bindings

Arbitrarily picked MPI_SEND, MPI_STARTALL, MPI_TYPE_CREATE_STRUCT to show as samples.

```
MPI_SEND(BUF, COUNT, DATATYPE, DEST, TAG, COMM, IERROR)
    <type> BUF(*)
    INTEGER COUNT, DATATYPE, DEST, TAG, COMM, IERROR

MPI_STARTALL(COUNT, ARRAY_OF_REQUESTS, IERROR)
    INTEGER COUNT, ARRAY_OF_REQUESTS(*), IERROR

MPI_TYPE_CREATE_STRUCT(COUNT, ARRAY_OF_BLOCKLENGTHS,
              ARRAY_OF_DISPLACEMENTS, ARRAY_OF_TYPES, NEWTYPE, IERROR)
    INTEGER COUNT, ARRAY_OF_BLOCKLENGTHS(*), ARRAY_OF_TYPES(*), NEWTYPE,
    IERROR
    INTEGER(KIND=MPI_ADDRESS_KIND) ARRAY_OF_DISPLACEMENTS(*)
```

16.2 Verbatim Fortran Bindings Samples

This is using simple "verbatim" latex style. Note that the `INTEGER` KIND stuff has not been finalized yet; we put in `MPI_INT_KIND` everywhere just to show what it would look like.

It is *possible* (likely?) that the MPI handles will be some other type in the new MPI-3 Fortran interface, not `INTEGER`.

```
subroutine MPI_SEND(BUF, COUNT, DATATYPE, DEST, TAG, COMM, IERROR)
  import
  <type>, dimension(*), intent(IN) :: BUF
  integer(MPI_INT_KIND), intent(IN) :: COUNT
  integer(MPI_INT_KIND), intent(IN) :: DATATYPE
  integer(MPI_INT_KIND), intent(IN) :: DEST
  integer(MPI_INT_KIND), intent(IN) :: TAG
  integer(MPI_INT_KIND), intent(IN) :: COMM
  integer(MPI_INT_KIND), intent(OUT) :: IERROR
end subroutine MPI_SEND

subroutine MPI_STARTALL(COUNT, ARRAY_OF_REQUESTS, IERROR)
  import
  integer(MPI_INT_KIND), intent(IN) :: COUNT
  integer(MPI_INT_KIND), dimension(*), intent(INOUT) :: ARRAY_OF_REQUESTS
  integer(MPI_INT_KIND), intent(OUT) :: IERROR
end subroutine MPI_STARTALL

subroutine MPI_TYPE_CREATE_STRUCT(COUNT, ARRAY_OF_BLOCKLENGTHS, &
     ARRAY_OF_DISPLACEMENTS, ARRAY_OF_TYPES, NEWTYPE, IERROR)
  import
  integer(MPI_INT_KIND), intent(IN) :: COUNT
  integer(MPI_INT_KIND), dimension(*), intent(IN) :: ARRAY_OF_BLOCKLENGTHS
  integer(MPI_ADDRESS_KIND), dimension(*), intent(IN) :: ARRAY_OF_DISPLACEMENTS
  integer(MPI_INT_KIND), dimension(*), intent(IN) :: ARRAY_OF_TYPES
  integer(MPI_INT_KIND), intent(OUT) :: NEWTYPE
  integer(MPI_INT_KIND), intent(OUT) :: IERROR
end subroutine MPI_TYPE_CREATE_STRUCT
```

16.3 Sample 1 Fortran Bindings

This uses similar LaTeXmechanisms as the existing "mpifbind" macros, but slightly modified.

Same disclaimers as above – I used `MPI_INT_KIND` everywhere just to see what it would look like.

```
subroutine MPI_SEND(BUF, COUNT, DATATYPE, DEST, TAG, COMM, IERROR)
    import
    <type>, dimension(*), intent(IN) ::  BUF
    integer(MPI_INT_KIND), intent(IN) ::  COUNT
    integer(MPI_INT_KIND), intent(IN) ::  DATATYPE
    integer(MPI_INT_KIND), intent(IN) ::  DEST
    integer(MPI_INT_KIND), intent(IN) ::  TAG
    integer(MPI_INT_KIND), intent(IN) ::  COMM
    integer(MPI_INT_KIND), intent(OUT) ::  IERROR
end subroutine MPI_SEND

subroutine MPI_STARTALL(COUNT, ARRAY_OF_REQUESTS, IERROR)
    import
    integer(MPI_INT_KIND), intent(IN) ::  COUNT
    integer(MPI_INT_KIND), dimension(*), intent(INOUT) ::
    ARRAY_OF_REQUESTS
    integer(MPI_INT_KIND), intent(OUT) ::  IERROR
end subroutine MPI_STARTALL

subroutine MPI_TYPE_CREATE_STRUCT(COUNT, ARRAY_OF_BLOCKLENGTHS,
              ARRAY_OF_DISPLACEMENTS, ARRAY_OF_TYPES, NEWTYPE, IERROR)
    import
    integer(MPI_INT_KIND), intent(IN) ::  COUNT
    integer(MPI_INT_KIND), dimension(*), intent(IN) ::
    ARRAY_OF_BLOCKLENGTHS
    integer(MPI_ADDRESS_KIND), dimension(*), intent(IN) ::
    ARRAY_OF_DISPLACEMENTS
    integer(MPI_INT_KIND), dimension(*), intent(IN) ::  ARRAY_OF_TYPES
    integer(MPI_INT_KIND), intent(OUT) ::  NEWTYPE
    integer(MPI_INT_KIND), intent(OUT) ::  IERROR
end subroutine MPI_TYPE_CREATE_STRUCT
```