Tinkering with Array Descriptors in F95

Aleksandar Donev

July 1, 2002

1 Module *Array_Descriptors*

"WEAVE.f90" 1 ≡

```
MODULE Array_Descriptors
  USE Precision
  USE ISO_C_BINDING
  IMPLICIT NONE
PRIVATE
  PUBLIC :: MakeArrayDescriptor
  ⟨ MakeArrayDescriptor_Generic 1.1 ⟩
END MODULE Array_Descriptors
```

1.1 Making Array Descriptors

In this case there is really no need of a Fortran wrapper around the C routines.

"WEAVE.f90" 1.1 ≡

@m _MakeArrayDescriptor_Declaration(*type_*, *kind_*, *c_kind_*, *rank_*)

```
  SUBROUTINE MakeArrayDescriptor_C_@&c_kind_@&rank_(array_descriptor, base_address,
      lower_bounds, upper_bounds, strides) _BoundProcedure(C,
      MakeArrayDescriptor_C_@&c_kind_@&rank_)
  USE Precision
  USE ISO_C_BINDING
    _BindProcedure(MakeArrayDescriptor_C_@&c_kind_@&rank_)
    type_(KIND = kind_), DIMENSION (_FullExtent(rank_)), POINTER :: array_descriptor
    _PassedByValue (base_address, _C_PTR) :: base_address
    INTEGER (KIND = C_INT), DIMENSION (rank_), INTENT (IN) :: lower_bounds, upper_bounds, strides
  END SUBROUTINE
```

@m _MakeArrayDescriptor_Declaration_TR(*t_number_*, *rank_*)

_MakeArrayDescriptor_Declaration($\_AllCBoundTypes\,(t\_number\_)$, $\_AllCBoundKinds\,(t\_number\_)$, $\_AllCKinds\,(t\_number\_)$, $rank\_$)

⟨ MakeArrayDescriptor_Generic 1.1 ⟩ ≡

$\_GenerateExternalInterface\_TKR\_Array\,(MakeArrayDescriptor$, _MakeArrayDescriptor_Declaration_TR, $N\_TC\_ALL)$

This code is used in section 1.

§. `@I "C_Interop.hweb"` *Section(s) skipped...*
§. `@I "Macros.hweb"` *Section(s) skipped...*

§. `@I "Fortran.formatting.hweb"` *Section(s) skipped...*

§. `@I "Fortran.macros.hweb"` *Section(s) skipped...*
§. `@I "Fortran.F2x.hweb"` *Section(s) skipped...*