Emulation of ISO_C_BINDING in Fortran 95

Aleksandar Donev

June 25, 2002

1 Support for module ISO_C_BINDING in C

1.1 Pointer conversion

"WEAVE.c" 1.1 ≡

```
#include <stdbool.h>
  void *_MangledProcedure(F_C_POINTER_C)(void **f_pointer)
    {
    return (*f_pointer);
    }
  void *_MangledProcedure(F_C_PROCEDURE_C)(void *f_pointer)
    {
    return (f_pointer);
    }
  void _MangledProcedure(C_F_POINTER_C)(void **c_pointer, void **f_pointer)
    {
    *f_pointer = *c_pointer;
    }
  ⟨STRIDE_C 1.2⟩
```

1.2 Finding strides between array elements

Since Fortran arrays can be strided, it is useful to have a mechanism to find the stride (distance in the length of the unit type) between two array elements. C's pointer arithmetic gives us exactly this with just a simple subtraction.

"WEAVE.c" 1.2 ≡

```
@m _STRIDE_C_Body(type_, kind_)

    int _MangledProcedure(STRIDE_C_@&kind_)(type_ * var1, type_ * var2)
      {
      return (var2 - var1);
      }

@m _STRIDE_C_Body_T(number_)
    _STRIDE_C_Body(_AllCTypes(number_), _AllCKinds(number_))
```

⟨ STRIDE_C 1.2 ⟩ ≡

```
_GenerateBody_TK(N_TC_ALL, _STRIDE_C_Body_T)
```

This code is used in section 1.1.

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

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

§. `@I "Common.macros.hweb"` *Section(s) skipped…*

§. `@I "C.macros.hweb"` *Section(s) skipped…*