Co nového ve Fortranu 2008

Standard byl publikován v říjnu 2008. K hlavním přínosům patří rozšíření pro paralelní běh (coarrays a do concurrent) a submoduly, mezi jinými pak např. specifikace newunit a deskriptory G0/G0.d. V překladačích se standard objevuje pozvolna: v hranatých závorkách stojí symboly pro realizaci v překladačích gfortran 4.7 (G), ifort 14.0 (I) a pgfortran 13.9 (P); g95 nerealizuje Fortran 2003 ani 2008, s výjimkou coarrays.

Kooperativní pole (coarrays) [G I – g95]
Fortranská syntaxe pro spuštění několika obrazů (images) programu, schopných vzájemných datových přenosů (úmyslem je co nejjednodušší použitelná alternativa k MPI). Realizováno v ifort (Linux, Windows) a g95 (Linux), gfortran 4.7 jen akceptuje syntaxi.

```
integer :: s[*]=0
print *,'image ',this_image(),': start ',s
sync all
if (this_image()==1) then
  do i=1,num_images() ; s[i]=1 ; enddo
endif
sync all
print *,'image ',this_image(),': stop ',s
end program
```

Překlad: ifort -coarray file.f90 ; ifort -coarray=distributed file.f90 ; gfortran -fcoarray=single file.f90 ; g95 file.f90
Spouštění: ifort proměnná FOR_COARRAY_NUM_IMAGES ; ./a.out --g95 images=N

Submoduly [– – –]
Cílem je možnost vydělit implementaci modulových procedur z modulu do submodulů, volitelně umístěných v jiných souborech; v modulu musí zůstat jen modulová data a deklarace rozhraní modulových procedur. Lze tak modularizovat moduly, oddělovat veřejné popisy rozhraní od neveřejných implementací a umožnit vývojovým prostředkům typu make omezit kaskádu překladů vyvolaných po změně implementace modulové procedury.

```
module m ; interface ; module subroutine s(x) ; real x ; end subroutine ; end interface ; end module
submodule (m) sm ; contains ; module subroutine s(x) ; real x ; ... ; end subroutine ; end submodule
```

nebo

```
submodule (m) sm ; contains ; module procedure s ; ... ; end procedure ; end submodule
```

Ostatní
datový podtyp integer(8) [G I P]

```
integer(8) :: i=huge(i) ; print *,i ! 9223372036854775807
```

maximální počet rozměrů pole zvýšen ze 7 na 15 [– I –]
atribut contiguous pro pole předpokládaného tvaru (formální argumenty) a ukazatelová pole [G I –]

```
subroutine s(x) ; real,contiguous :: x(:) ; real,pointer,contiguous :: p(:)
```

rekurzivně deklarované alokovatelné položky struktur [– – –]

```
type tlist ; real value ; type(tlist),allocatable :: next ; end type
type(tlist) list ; allocate (list,source=tlist(1)) ; allocate (list%next,source=tlist(2)) ; deallocate (list)
```

položky re a im komplexního typu [– – P]

```
complex c ; c=(0.,1.) ; c%re=c%im ; print *,c
```

jednodušší alokace polí podle vzoru a formy [– I –]

```
real,allocatable :: a(:),b(:),c(:),d(:) ; allocate (a(10)) ; allocate (b,c,source=a) ; allocate (d,mold=a)
```

paralelizovatelný cyklus do concurrent (dříve standardizovaný cyklus forall připouští jen přiřazovací příkazy) [G I –]

```
do concurrent (i=1:n, j=1:n) ; ... ; enddo
```

skok exit ve více konstrukcích (associate, block, if, select case, select type, ne critical a do concurrent) [G – –]
konstrukce block pro omezení oboru platnosti jmen [G – –]

```
do i=1,n ; block ; real x ; ... ; end block ; enddo
```

specifikace newunit pro získání volných kanálů pro vnější soubory [G I P]

```
open (newunit=id,file='file') ; write (id,*) id ; close (id)
```

editovací deskriptory G0, G0.d a opakovač *(..) [G I –]

```
print '(*(G0))',1,',',1.,',',.true.,',','a' ; print '(*(G0,:,","))',1,1.,.true.,'a' ! 1,1.000000,T,a
```

prvkové procedury mohou být nečisté (impure elemental), tj. mohou měnit globální proměnné, vypisovat aj.
interní procedury jako skutečné argumenty nebo cíle ukazatelů (funkční objekty, functors)
volání ukazatelových funkcí na levé straně přiřazovacích příkazů nebo jako výstupní skutečné argumenty (accessors)

standardní funkce bessel_j0, .., bessel_yn, erf, gamma, log_gamma, hypot, norm2, bitové manipulace
funkce find_loc(array,value,..) pro nalezení hodnoty v poli [– – –]

```
print *,findloc([(i,i=1,10)],7)
```

volání vnějšího programu, informace o překladu [G – –]

```
call execute_command_line(..)
use iso_fortran_env ; print *,compiler_version(),compiler_options()
```

Odkazy
Fortran 2008 v překladačích: http://fortranwiki.org/fortran/show/Fortran+2008+status
Coarrays: http://www.g95.org/coarray.shtml, http://www.g95.org/N1747.pdf, http://www.g95.org/compendium.pdf

1. 11. 2013