arm

Report: Joint J3 / WG5 Fortran standards meeting

N J Sircombe

24 September 2019

Fortran standards organizations

National and international bodies working on the Fortran standard

- International standards in IT maintained by the International Organisation for Standardization (ISO) and the International Electrotechnical Commission (IEC) through their Joint Technical Committee (JTC1)
- Work is delegated to sub-committees (SCs), and permanent working groups (WGs)
- **ISO/IEC JTC1/SC22/WG5**, or just '**WG5**', owns the ISO Fortran Standard
- The membership of WG5 is drawn from national member bodies, for example BSI (British Standards Institute) in the UK and the InterNational Committee for Information Technology Standards (INCITS) in the US.
- The Primary Development Body is the US Fortran Committee, **INCITS/PL22.3**, often known simply as '**J3**', a reference to its old ANSI name - X3J3
- *See https://wg5-fortran.org and https://j3-fortran.org for more information*

Joint WG5 – J3 meeting

5 – 9 August 2019, Tokyo University Hongo Campus

Meeting aims

- Arrive at a definitive list of which features to take forward for the next standard
  - Starting from submissions from national bodies:
    - UK - WG5/N2167
    - US - WG5/N2165
    - Japan - WG5/N2168
- Working title **“Fortran 202X”**.
- Aiming for a 2022 release (i.e. **Fortran 2022**)
- Relatively minor revision
- Only one or two significant new features
  - Ease the burden of implementation while still delivering enhancements requested by users

https://wg5-fortran.org/documents.html

arm

Joint WG5 – J3 meeting

5 – 9 August 2019, Tokyo University Hongo Campus

Attendees

Steve Lionel (Self, USA) *convener*

Dan Nagle (NCAR, USA) *PL22.3 chair*

Reuben Budiardja (Oak Ridge, USA)

Kiran Chandramohan (Arm, UK)

Daniel Chen (IBM, Canada)

Tom Clune (NASA, USA)

Malcolm Cohen (NAG, UK)

Magne Haveraaen (University of Bergen, Norway)

Yasuharu Hayashi (NEC, Japan)

Hidetoshi Iwashita (Fujitsu, Japan)

Gary Klimowicz (NVIDIA, USA)

Tom Knox (Kernelyze, USA)

Bill Long (Cray, USA)

John Reid (JKR Associates, UK)

Damian Rouson (Sourcery, USA)

Hiroyuki Sato (University of Tokyo, Japan)

Nathan Sircombe (Arm, UK)

Van Snyder (Caltech/JPL, USA)

Toshihiro Suzuki (Fujitsu, Japan)

Masayuki Takata (Edogawa University, Japan)

Fortran 202X

The small stuff

- ‘SIMPLE’ procedures: PURE procedures with no writing or referencing beyond their arguments (J3/19-201r1)
  - PURE functions do not write to global memory
  - SIMPLE functions will not reference global memory either
- DO CONCURRENT reductions (J3/19-198r2)
  - Introduced in 2008, CONCURRENT loops have no data dependency, the compiler is free to execute in parallel
  - Proposals for 202X plan to add reductions analogous to OpenMP parallel do clauses

```
foo = a(1)
DO CONCURRENT (i = 2:size(a)) REDUCE(MIN: foo)
  foo = MIN(foo, a(i))
END DO
```

References J3/YY-xxxx documents can be found at https://j3-fortran.org/doc/year
WG5/Nxxxx can be found at https://wg5-fortran.org/documents.html

arm

Fortran 202X

The small stuff

- Improvements to string handling
  - Plans to incorporate Part II of the standard into an ISO_FORTRAN_STRINGS module scaled back
  - New SPLIT intrinsic (J3/19-196r3)
  - Improved interoperability with C strings (J3/19-197r3), two new intrinsics
    `C_F_STRPOINTER` - associate a character pointer with a string
    `F_C_STRING` - copy a string with appended NUL
  - Trimmed character format specifier (J3/19-137r2)
  - Improved support for delayed length character strings.

arm

Fortran 202X

The small stuff

- Degree and IEEE circular trigonometry support (J3/19-203r1 and J3/19-204r1)
  - **`COSD`**, **`SIND`**, **`TAND`**, **`ACOSD`**, **`ASIND`**, **`ATAND`**, **`ATAN2D`** – Degree versions of trig intrinsics are a common extension of the standard
  - **`COSPI`**, **`SINPI`**, **`TANPI`**... etc. implementations of IEEE circular trig functions (angles expressed in ½ revolutions rather than radians or degrees)
- Conditional expressions
  - Note, short-circuit operators and lazy evaluation are processor dependent.
  - Separate requests for inclusion of short-circuit logical operators were rejected, as they were effectively subsumed into this proposal

Fortran 202X

The small stuff

- Binary compatibility of C descriptors
  - Currently flexibility to add struct members between the mandated members at the beginning and end of the descriptor - standardising would benefit library developers in particular
  - Concern over burden on vendors and backwards compatibility, combined with weak support lead to this proposal being defered for a future revision of the standard
- Standardized support for BOZ literals (J3/19-211 and J3/19-212r1), proposals for a BITS datatype were abandoned
  - Binary, octal and hex literal constants widely supported
  - This will standardise support

Fortran 202X

The small stuff

- Protected components (J3/19-214r1)
  - Provides an attribute of a type component that specifies that its value cannot be changed outside the module where the type is defined
  - Ensures that derived type components can only be updated via module or type bound procedures provided for the purpose
- Standardized support for long lines (J3/19-138r1)
  - Fortran 2003+ requires support for 132 characters, and 255 continuation lines
  - In practice, the line 'limit' is not enforced by default
  - Compilers support *much* longer lines
  - This standardises behaviour (far beyond current practice!), proposing:
    - Limit of 10,000 characters per line
    - Limit of 1,000,000 characters per statement

Fortran 202X

The small stuff

- Enumerators (J3/216r1)
  - Currently **`ENUM, BIND(C)`** support for interoperability, but no ‘native’ support for enumerators
  - Proposes enumeration types containing ordered enumerators (compatible with DO construct and use of range in a CASE statement)
  - Type-bound procedures to return integer representation and first/last/next/previous enumerators in the type

```
ENUMERATION :: colours
  ENUMERATOR :: red, green, yellow, blue
END ENUMERATION
```

Fortran 202X

The small stuff

- Coarray enhancements
  - Put-with-notify (J3/18-277r1)
  - Arrays of coarrays (J3/208r1)
- New KINDS (J3/19-139r1 and J3/19-147r1)
  - **`REAL16`**, half precision where hardware supports it
  - more LOGICAL kinds and **`SELECTED_LOGICAL_KIND()`**.

Fortran 202X

Exception handling

- See (J3/19-133r2) for details of use cases.

- The committee was unable to produce requirements which
  - Would be achievable without significant impact on 202X delivery timescales
  - Would not have a dramatic adverse impact on performance
  - Would not overly burden vendors at the implementation stage
  - Would deliver functionality worth having and significantly beyond that achievable with existing language tools.
- **The committee has decided not to pursue exception handling for 202X**

Fortran 202X

Generic programming

- MACROs (J3/19-154) were considered as an option to provide some of the key functionality users have requested:
  - Specifically, templated procedures
    (to avoid the potentially error prone process of hand-crafting specific implementations)
  - Some advanced pre-processors already provide this functionality (outside of the standard)
- Originally proposed for Fortran 2008

- **The committee has decided not to pursue ‘intelligent MACROs’ for 202X**

Fortran 202X

Generic programming

- Support for greater array and type generic programming:
  - TYPEOF and CLASSOF attributes (J3/19-142r1)
  - Rank generics: bounds and rank attributes; rank agnostic array allocation; rank agnostic reference and assignment of elements and sections (J3/19-202r2)
  - New notation for assignment and reference will be developed to avoid ambiguity with vector subscripts (J3/19-150)

    Access **`A ( V(1), V(2), ... V(n) )`** using **`A @( V )`**
    e.g. **`A @( MINLOC(A) )`** to access the minimum element.
- In the longer term a response to user requests for ‘templates’ is a necessity:
  - Work towards parametrized modules, or similar, will be this response
  - Considerable work needed now to develop the design, requirements and specs
- **Committee intends to commit to the development of a detailed design targeting the next standard beyond 202X - 202Y (where X < Y ≤ 9)**

WG5 / J3

Future meetings

- **J3:**
  - 14 - 18 Oct 2019 - Las Vegas, US;
  - 24 - 28 Feb 2020 - Las Vegas, US;
  - 22 -26 June 2020 - Joint with WG5 - Minnesota, US.
- **WG5 (all joint with J3):**
  - 2020: 22 - 26th June 2020. Bloomington, Minnesota, US.
    - CRAY hosting - CRAY office at the Mall of America (MoA)
  - 2021: 21 - 25th June 2021. Manchester, UK.
    - Arm hosting
  - 2022: TBD

Links

- WG5: wg5-fortran.org
- J3: j3-fortran.org
- Minutes: WG5/N2169
- Resolutions: WG5/N2170
- J3 Meeting papers: j3-fortran.org/doc/meeting/219

arm

Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
تشكر