

学校以来だ。今時のフォーっ！

Echigo BSD Users Group 19th meeting
at Chuo Citizens' Public Hall of Niigata city.
23th, September, 2006
INOUE Mikio mikio@ebug.jp
Rev. 20070418

おしながき

- FORTRAN の歴史
- Fortran 90/95
- Fortran を使ってる人向けの新機能
 - 一般のプログラマには、どうでもいいこと。
- Fortran を使わない人向けの新機能
 - 一般のプログラマには、すごいこと。
- その他の話題

FORTRAN の概要

- FORMula TRANslation
 - アセンブラではめんどい数式処理を簡単に記述できる。
 - 豊富な組み込み算術関数。
- コンピュータ史上初の高水準記述言語
 - 逐次型、手続き型
 - 僅かなリソースを有効に活用。

FORTRAN の歴史 1/3

- FORTRAN I
 - 1954 年 IBM704 向け。
 - IBM の John Backus (1924-) による。
- FORTRAN II / III
 - 副プログラムの実装など。
- FORTRAN IV
 - データ型の多様化に伴う宣言文の導入。
 - 条件文の追加など。

FORTRAN の歴史 2/3

- FORTRAN66
 - JIS 3000 水準
 - 名前は 5 文字以内。論理 IF 文、3 次元配列などが無い。
 - JIS 5000 水準
 - 名前は 6 文字以内。
 - JIS 7000 水準
 - FORTRAN66 と同じ。

FORTRAN の歴史 3/3

- FORTRAN77
 - FORTRAN66 の拡張。
 - IMPLICIT 文や PARAMETER 文などのプログラムの明確化を行う機能の追加。
 - 配列に寸法宣言子を導入。
 - 文字列処理機能の強化。
 - IF THEN~ELSE~END の導入。
 - ファイル入出力の強化。

Fortran 90 / 95

- いまさらながら言語仕様の大幅な近代化。
 - 構造化を支援する機能強化。
 - 古典的な記述法の廃止。
 - 固定プログラム形式からの脱却。
 - 動的メモリ管理を行う機能強化。
- 配列処理のための機能強化。
 - 新しい配列処理メカニズム。(すんごい)

構造化を支援する機能

- 制御構造の拡張
 - 汎用 DO ~ END 文と EXIT、SKIP、CYCLE。
 - SELECT CASE ~ CASE ~ END。
- データ構造の拡張
 - 構造体を定義する TYPE 文。
- 再帰呼び出しの導入
 - 関数を再帰可能であると宣言する RECURSIVE パラメタの導入。

古典的な記述法の廃止

- 算術 IF 文、割当型 GOTO 文、ASSIGN 文とかは将来廃止されるのが確定。
- 暗黙の変数型
 - 暗黙型を無効にする IMPLICIT NONE 文。
- 固定プログラム形式からの脱却
 - 自由プログラム形式 (文脈自由文法)
 - マルチステートメントの導入

動的メモリ管理

- 動的な記憶を管理する機能
 - 動的な配列や変数を定義する ALLOCATABLE パラメタの導入。
 - 動的に記憶を操作する ALLOCATE 文と DEALLOCATE 文。

ポインタ管理

- ポインタの管理
 - ポインタを操作する POINTER 文と NULLIFY 文、TARGET 文。
 - だれでも LISP が書けるようになりました。;-P

配列定数と代入

```
integer, dimension(10) :: A
!  
! としておくと、次の 3 行は同じ結果。  
!  
A = (/ 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 /)  
A(:) = (/ (2*I, I = 1, 10) /)  
A(1:10) = (/ 2, 4, (I, I = 6, 18, 2), 20 /)  
!  
! 因みに、全ての要素に同じ数を代入するなら一発。  
!  
A(1:10) = 123
```

配列式

```
integer, dimension(4) :: A = (/ 1, 2, 3, 4 /)
integer, dimension(0:3) :: B = (/ -5, 6, -7, 8 /)
integer, dimension(-1:2) :: C
logical, dimension(4) :: D
!
! としておく、こんなことができます。
!
C = A * 2 + abs(B)
D = (B > 0) .and. (mod(A, 2) == 1)
```

部分配列

```
integer, dimension(10) :: &
  A = (/ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
integer, dimension(5) :: B, &
  C = (/ 7, 4, 5, 2, 8 /)
!
! としておく、こんなことができます。
!
B = A(1:10:2)
A(2:10:2) = (/ I*2, (I = 1, 5) /)
B = A(C) ! ベクトル添え字
```

WHERE 構文

```
integer, dimension(5) :: A = (/ -5, 6, -7, 8, 9 /)
Real*8, dimension(5) :: B
!
! としておく、こんなことができます。
!
where (A < 0)
  B = 0
elsewhere
  B = sqrt(real(A))
end where
```

その他の拡張

- モジュールのサポート
 - MODULE 文、USE 文。
 - INCLUDE 文の導入
 - モジュール使った方がいいかも。
- 利用者定義
 - 明示的引用宣言 INTERFACE 文。
 - 引数の仕様を宣言する INTENT の導入。

gfortran (GNU Fortran)

- GCC4.0 以降なら使えます
 - G77 は、GCC3 系列で終了～。
- 仕様の拡張
 - 古いスタイルのサポート
 - Cray ポインタサポート
 - C との相互運用性の向上
 - 内部副プログラムの追加 (日付・時刻関連等)

gfortran を使ってみよう

- 多様なプラットフォーム
- FreeBSD の ports から入れるなら注意
 - ‘-D WITH_FORTRAN’ を付けて make。
- コンパイルオプション
 - コンパイラの基本的な部分は gcc と一緒。
 - 言語仕様や方言に関するオプション。
 - コーディング形式に関するオプション。
- GNU Fortran は開発途上。

これからの gfortran

- Fortran 95 標準ではない機能
 - でも、既存の多くのコンパイラに実装されてる。
 - コンパイラの拡張
 - 条件付コンパイル (C の #define みたいな)
 - Runtime を含む OpenMP への対応
 - Fortran 200X への取り組み
 - 環境変数を取り扱う各機能

悩ましいかも知れないこと

- Fortran を道具として使っている人の多くが、プログラミングのパラダイムに興味が無い。
- 多くの Fortran が、マルチプロセッサやクラスタリングに十分対応できない。
 - OpenMP – メモリ共有に関する提案
- ハードウェア性能向上の方向性
 - ベクトルプロセッサからクラスタへ。

今後の Fortran

- Fortran 2003
 - フルスペックな実装は、まだ無いっぽい。
 - データ型の拡張や I/O の拡張等、より進んだ近代化。
 - C との親和性の拡張。
- その先は、怪しい...?
 - 要求に対する、ハードウェアへの対応。
 - 利用者のニーズ。

雑談の種

- 66, 77 ときたら 88, 99 だろ。
 - 66, 77, 90, 95, 2003
- 歴史的資産を生かす方法。
- 過去との決別 F。
- BSD 系のクラスタリングって何か知ってる?
- 懇親会では、「おおブレネリ」を歌おう!!

参考

- Fortran90 プログラミング (書籍になってる)
 - <http://ace.phys.h.kyoto-u.ac.jp/~tomita/education/fortran90/sec0.html>
- GCC, the GNU Compiler Collection
 - <http://gcc.gnu.org/>
- The Fortran Company
 - <http://www.fortran.com/>
- Fortran 標準コーディングルール
 - <http://www.mri-jma.go.jp/Project/mrinpd/coderule.html#sec3>

追補: gfortran について

- 最新の状況を Nakata Maho 氏からご指摘いただきました。(2007 年 3 月現在)
 - FreeBSD の ports では、lang/gcc42 及び lang/gcc43 において、-D WITH_FORTRAN の指定は不要です。そのまま、make install するだけで gfortran も導入されます。
 - gfortran のデフォルトの ports は、lang/f77 から lang/gfortran に移行作業中です。作業完了まで、関連する ports が unstable になっています。