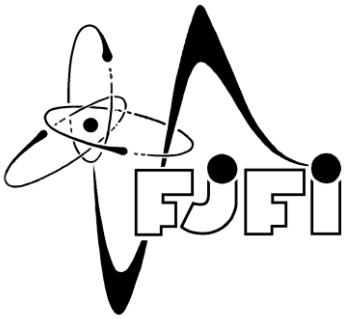
České vysoké učení technické v Praze

Fakulta jaderná a fyzikálně inženýrská



Vědeckotechnické výpočty 12VTV

Zápočtová úloha

Přehled hlavních změn v jednotlivých verzích jazyka Fortran(77,90,95,…)

Martin Suchánek

LPT

2012/2013

OBSAH

Přehled hlavních změn v jednotlivých verzích jazyka Fortran(77,90,95,…)

Úvod

Fortran je zkratka pro FORmula a TRANslator. Fortran je v informatice imperativní programovací jazyk, který v 50.letech 20.století navrhla firma IBM pro vědecké výpočty a numerické aplikace. Brzy se stal jedničkou mezi programovacími jazyky a více než půl století se využíval například pro výpočty předpovědi počasí, analýzu částic a další fyzikální a chemické výpočty. V novějších verzích fortranu se postupně objevovaly různé nové vlastnosti, jako například podpora pro datová pole (Fortran 90/95), objektově orientované programování a generické programování(Fortran 2003) a souběžné programování(Fortran 2008).

Jednotlivé verze programu Fortran, kterými se budeme zabývat

Fortran 77, Fortran 90, Fortran 95, Fortran 2003, Fortran 2008.

V této úloze se budeme zabývat, jaké jsou změny nebo rozšíření od verze od předchozí verze.

Rozšíření ve Fortranech(77;90;95) (podrobně):

Fortran 77

Fortran 77přidal řadu významných funkcí, které řeší mnoho z nedostatků verze Fortran 66. Změny a rozšíření provedené ve Fortranu 77:

- Blokový příkaz IF

  - Jedním z nejdůležitějších rozšíření ve Fortranu 77, které snížilo problémy se zápisem programu ve strukturovaném tvaru. Příkaz se skládá z několika dílčích příkazů, které se provedou po splnění podmínky a jsou obvykle následovány blokem příkazů.

  Příklad: (blokový varianta)

```
IF(podmínka_1)THEN
   Blok_příkazů_1
ELSE IF(podmínka_2) THEN
   Blok_příkazů_2
ELSE IF(podmínka_3)THEN
        .
        .
ELSE
  Poslední_blok_příkazů
END IF
```

  Příklad: (zkrácená varianta):

```
IF(podmínka) příkaz
```

- Příkaz DO

  - Má napohled stejný tvar jako ve Fortranu 66, ale jeho možnosti jsou podstatně rozšířeny.

  Příklad(obecný tvar):

  DO návěstí, proměnná = počát.hodnota, konečná hodnota, krok

- Typ CHARACTER

  - ve Fortranu 77 je nově zaveden a zlepšuje práci s řetězci a znaky. Základní operací pro práci s řetězci je zřetězení reprezentovány operátorem //. Přiřazovací příkazem můžeme hodnoty znakových výrazů přiřazovat proměnným a prvkům polí(typu character) nebo podřetězců.

- POLE

  - Deklarace polí je stejná jako u předchozí verze, ale navíc lze zadat dolní mez(pokud není zadána, považuje se za rovnou jedné). Meze se dají psát jako celočíselné konstantní výrazy.

- Vstup a výstup

  Vstup a výstup má největší změny a rozšíření oproti Fortranu 66.

  - Příkazy READ , WRITE, PRINT
    - READ a WRITE mohou mít stejný tvar jako v předchozí verzi , ale mohou obsahovat další syntaktické části(specifikátory).
    - Díky příkazům READ a WRITE k interním souborům přistupujeme

    Příklad:

    ```
    WRITE(*,*) ‘Ahoj‘

    READ(*,*) CISLO
    ```

  - Popis FORMAT
    - Zachovány všechny možnosti Fortranu 66, díky rozšířeným schopnostem edičních popisovačů existuje nyní řada možností umožňují specifikovat tvar zápisu nebo čtení.

- OPEN a CLOSE
  - Ve Fortranu 77 zcela nové.
  - Zajišťuje vytváření a rušení souborů, jejich připojování k fortranským V/V jednotkám(vstupní a výstupní jednotky) – ve Fortranu 66 se musela tato akce řešit na úrovni jazyka pro řízení úloh nebo prostřednictvím implementačně závislých podprogramů.

- Příkaz INQUIRE
  - Slouží k zjišťování různých údajů o souboru nebo o V/V jednotce.

- I/O jednotka

  - virtuální kanál, označený celým číslem od nuly do limitu daného systémem(zpravidla minimálně 99)

  - před použitím musí být externí soubor připojen příkazem OPEN k I/O jednotce , pomocí jejíhož čísla je pak jednoznačně identifikován ostatními příkazy

  Příklad:

  OPEN (UNIT = 1, FILE = `cesta k souboru` , STATUS= `OLD`)

  OPEN(UNIT = 9, FILE = 'PRINTOUT' , STATUS = 'OLD')

- Jiná rozšíření

  - Zejména je zde k dispozici řada funkcí pro práci s texty(nalezení podřetězce, výpočet délky znakového výrazu, lexikografické porovnání dvou znakových výrazů prováděné na rozdíl od relační operátorů nezávisle na vnitřním zobrazení znakových veličin), pro zaokrouhlení na nejbližší celé číslo, další matematické funkce.

V logických operátorech lze použít operátory .EQV. a .NEQV. (ekvivalence a nonekvivalence) ad.

Fortran 90

Zpožděný následovník Fortranu 77. Před názvem Fortran 90 byl znám jako Fortran 8X. V této verzi bylo přidáno mnoho nových funkcí.

V novějším Fortranu se nacházejí tři druhy prvků z předchozí verze:

a) zrušené

b) zastaralé

c) neperspektivní

a) prvky z předchozí verze nejsou součástí jazyka

b) prvky, které jsou nahrazeny modernějšími a v jazyce zůstávají kvůli návaznosti

c) prvky, které se nebudou využívat po té, co se vžijí nové možnosti verze

- Tvar zápisu

  - v této verzi byla zavedena volná forma zápisu, ve které se program zapisuje do řádků libovolné délky

  - komentáře lze psát na samostatný řádek nebo na konec libovolného významového řádku(musí, ale začínat vykřičníkem)

  - dlouhé příkazy lze rozdělit na více řádků znakem &, lze napsat několik příkazů na řádek a oddělit středníkem

  - identifikátoru mohou délku až 31 znaků

- Symboly pro relační operátory

  - dvě možnosti jak zapsat symboly:

    - .LT. .LE. .EQ. .GT. .GE. .NE.
    - < <= == > >= /=

- Deklarace

  - Na jeden řádek lze psát několik deklarací

- Např.:

REAL, DIMENSION (3), PARAMETER : :
[název] = (/0.0, 0.0, 0.0/), [název] = (/1.0, 1.0, 1.0/)

-> vytvoření reálné konstantní vektory délky 3

- NAMELIST (seznam názvů) = vstupní seznam
  - nástroj pro seskupování proměnných pro I/O
  - použití - pro výstup může být užitečné pro testování a ladění proramu
  - také se používá pro definování skupiny proměnných

  - Např.:

    Namelist/ TÝDEN/ Po, Út, St, Čt, Pá

  - Specifikace NAMELIST umožňuje používání při volání READ seznam proměnných

    - Deklarace na začátku:

      NAMELIST /SEZNAM/ X,Y,Z,I,J

      Zavolání během programu

      READ(99,NML=SEZNAM)

      Bude počítač očekávat na vstupu připojeném k jednotce 99 data ve formátu např.

      SEZNAM J = 35, Z = 5.0, X = 1.E6

- Přepínač SELECT CASE
  - Dá se použít jako náhrada ve větvi, kde by se použilo mnoho větví ENDIF

    -> použití příkazu:

```
SELECT CASE (výraz)
  CASE(první_hodnota)
    první_blok_příkazů
  CASE(druhá_hodnota)
    Druhý_blok_příkazů
 CASE DEFAULT
   poslední_blok_příkazů
END SELECT
```

- Nové řídící struktury

  - ve Fortranu77 bylo zavedeno IF – THEN – ELSE -> Fortran 90 zavedl další rozšíření
    -> možnost zapisovat cykly v rozšířeném tvaru:

    [název:]DO[navěští][řídící parametr]

    Blok příkazů

    [návěští]ENDDO[název]

  - místo ENDDO může být CONTINUE DO

  - zavedena i konstrukce CASE -> vícenásobné větvení programu podle – celočíselné, logické nebo znakové hodnoty výrazu

Fortran 95

- další rozšíření pro manipulaci s poli (dynamické ukládání polí)
- dále Fortran 95 obsahuje nové řídící konstrukce SELECT CASE a nové formy příkazu DO,příkazy FORALL a lepší inicializace ukazatelů

I. Nové prvky

- Forall

  - alternativa ke smyčce DO

  - je třeba dodržovat určitá pravidla, aby nedocházelo ke kolizím

  - forma zápisu:

    FORALL (rozsah_prvního_indexu,…,rozsah_posledního_indexu, maska) přiřazení

    Rozsah má tvar –

    Řídící_proměnná = počáteční_hodnota : konečná_hodnota : krok

- uvnitř smyčky FORALL lze použít konstrukt WHERE

- Rozšíření ELSEWHERE

  - v konstrukci WHERE lze použít masku i klíčového slova ELSEWHERE a to pak opakovat:

```
WHERE (podmínka_1)
        Blok_příkazů_1
ELSEWHERE(podmínka_2)
        Blok_příkazů_2
ELSEWHERE(podmínka_3)
        .
        .
ELSEWHERE
        Poslední_blok_příkazů
END WHERE
```

- Čisté funkce a procedury

  - čistá = nemá postranní účinky -> kromě proměnných lokálních(platné pouze v jejím těle) a předávaných(v hlavičce) nemění už žádné jiné

  -výhodné v paralelním programování

  -občas označovány jako PURE

- Elementární funkce a procedury

  - procedura je tzv.elementární = lokální proměnné jsou skaláry(ne ukazatele nebo další procedury)

  - elem. funkce musí mít skalární návratovou hodnotu

- elem. funkce a procedury se označují jako ELEMENTAL -> předpokládá se PURE

- Prázdné ukazovatele

  - funkce NULL umožňuje deklarovat prázdný ukazatel

- Automatická dealokace paměti

  - při zapomenutí použití příkazu DEALLOCATE -> systém sám uklidí za čas

- Aritmetika s nulou

  - dvě reprezentace nuly: ''plus nula'' a ''mínus nula'' -> zavedení funkce SIGN

II. Zrušené konstrukce

- řídící proměnné typu REAL a DOUBLE PRECISION ve smyčkách Do
- skok na ENDIF zvnějšku
- příkaz PAUSE
- ASSIGN, přiřazené GOTO a aritmetická hodnota návěští
- Nejstarší tvar příkazu FORMAT
- Mnoho prvků jsou doporučeny nepoužívat v nových programech -> na odstranění

Rozdíly ve Fortranu(2003;2008) – obecný přehled:

Fortran 2003

- přehled nejvýznamnějších změn ve Fortranu 2003:

  - Objektově orientované programování
  - Spolupráce s C

- drobné doplňky:

  - Ukazatele procedury
  - Parametrické odvozené typy
  - Přenos přidělení
    - zobecnění často požadované schopnosti přerozdělit
  - Přístup k příkazové řádce a proměnných prostředí
  - Zobecnění výrazů pro rozměry pole a počátečních hodnot
  - Uživatelsky definované odvození typu vstupu/výstup
  - Polymorfismus
    - ''je'' vztah, který nám pomáhá představit, jak polymorfní proměnné komunikují s typem rozšíření
    - polymorfní proměnná je taková, jejíž datový typ je dynamický při běhu
    - musí to být ukazatele proměnné allocatable proměnná nebo fiktivní spor
    - např.:

      třída(tvar), ukazatel : : sh

    - sh – může být objekt ukazatel do tvaru nebo jakékoliv typ rozšíření
    - dva základní typy polymorfismu – postup polymorfismu
      - polymorfismus údaje

      a) postup polymorfismu – zabývá se postupy, které mohou pracovat na různých datových typech a hodnot

      b) údaje polymorfismu – zabývá se programovými proměnnými, které lze uložit a provozovat na různých datových typech a hodnot

Fortran 2008

- zatím poslední verze

- od verze Fortran 95 došlo jen k drobným úpravám -> zjednodušení kódu a odladění verze Fortran 2003 a přibyly některé módy kompatibility

- Co-array Fortran

  - model paralelního výpočtu

- Datový typ BIT

REFERENCE

- www.fi.muni.cz/usr/jkucera/pv109/2000/cpilat.rozdily.il2.html
- www-troja.fjfi.cvut.cz/~vachal/fortran/node47.html
- www.fortran.com/fcd_announce.html
- www.pgroup.com/lit/articles/insider/v3n1a3.htm
- en.wikipedia.org/wiki/Fortran#Fortran_2008
- cs.wikipedia.org/wiki/Fortran
- http://www.adt.unipd.it/corsi/Bianco/www.pcc.qub.ac.uk/tec/courses/f90/stu-notes/F90_notesMIF_9.html