Введение

Ильин Андрей Леонидович E-mail: alilyin@mephi.ru

Организация работы.

Работать будем на ферме lxfarm. Вход ssh –X user@ui02.lxfarm.mephi.ru .
Если Windows – эмулятор LINUX MobaXterm. Или PuttY.
Компиляция программы gfortran namefile.f -o namefile.
Редактор emacs namefile.f ; pico или pine.
Скопировать файлы с lxfarm в windows - программа PSFTP (под windows).
Запуск программы на выполнение . / имя_файла.exe

Литература.

1. Брич З.С. И др. Фортран 77 для ПЭВМ ЕС. Москва «Финансы и статистика» 1991 г. -288 стр.
2. С. Немнюгин, О. Стесик. Современный Фортран (самоучитель). Санкт-Петербург «БХВ-Петербург» 2005г.

FORTRAN.

Предназначен для решения сложных вычислительных задач.
Достаточно древний язык. Разработкой руководил Джон Бэкус. Именно он в IBM возглавил работы по разработке первого языка высокого уровня Фортран (FORTRAN, от FORmula TRANslator - «переводчик формул» на машинный язык. Первая коммерческая версия языка была выпущена в 1957 году.

Эволюция стандартов языка

FORTRAN 66 (1972)

На базе стандарта фирмы IBM **FORTRAN IV**

FORTRAN 77 (1980)

1. Введены операторы открытия и закрытия файла (`OPEN`, `CLOSE`) и вывода на стандартное устройство — `PRINT`.
2. Добавлены строковый тип данных и функции для его обработки.
3. Введён блочный оператор `IF` и конструкция `IF THEN` — `ELSE IF THEN` — `END IF`, а также оператор включения фрагмента программы `INCLUDE`.
4. Введена возможность работы с файлами прямого доступа.
5. Увеличена максимальная размерность массива с 3 до 7. Сняты ограничения на индексы массива.

Fortran 90 (1991)

1. Введён свободный формат написания кода. Появились дополнительные описания `IMPLICIT NONE`, `TYPE`, `ALLOCATABLE`, `POINTER`, `TARGET`, `NAMELIST`.
2. Введены управляющие операторы и конструкции. Добавлены `DO … END DO` (вместо завершения цикла меткой), `DO WHILE`, оператор передачи управления на начало цикла `CYCLE`, конструкция выбора `SELECT CASE` (для замены громоздких конструкций `IF` и операторов `GOTO`), а также заключительный оператор программной единицы, модульной или внутренней процедуры `END`[7].
3. Введён инструментарий указателей и функции для работы с оперативной памятью (по аналогии с языком C).
4. Введены операторы работы с динамической памятью (`ALLOCATE`, `DEALLOCATE`, `NULLIFY`).
5. Добавлены программные компоненты `MODULE`, `PRIVATE`, `PUBLIC`, `CONTAINS`, `INTERFACE`, `USE`, `INTENT`.
6. Введено маскирование присваивания массивов (присваивание при выполнении наложенного на элементы массива логического условия без использования операторов условия), а также работа с сечениями массивов. Введён оператор и конструкция `WHERE` для частичной замены циклов (правая часть оператора присваивания не изменяется). Маскирование присваивания распространяется практически на все операторы, конструкции и функции, оперирующие с массивами.
7. Стандартные операции присваивания, сложения, вычитания, а также деления и умножения на число распространены на массивы и их секции, определяемые сечениями. В этом случае осуществляется поэлементное присваивание.
8. В языке появились элементы ООП. Введены производные типы данных.

Fortran 95 (1997)

Коррекция предыдущего стандарта. Введён оператор и конструкция `FORALL`, позволяющие более гибко, чем оператор и конструкция `WHERE`, присваивать массивы и заменять громоздкие циклы. `FORALL` позволяет заменить любое присваивание сечений или оператор и конструкцию `WHERE`, в частности, обеспечивает доступ к диагонали матрицы. Данный оператор считается перспективным в параллельных вычислениях, способствуя более эффективному, чем циклы, осуществлению распараллеливания

Fortran 2003 (2004)

Дальнейшее развитие поддержки ООП в языке.

1. Асинхронный ввод-вывод данных.
2. Средства взаимодействия с языком **C**.
3. Усовершенствование динамического размещения данных[12].

Fortran 2008 (2010)

Стандартом предполагается поддержка средствами языка параллельных вычислений (Co-Arrays Fortran). Также предполагается увеличить максимальную размерность массивов до 15, добавить встроенные специальные математические функции и др.

Fortran 2018

Последняя версия языка Fortran 2018 (ранее известная как Fortran 2015) была выпущена 28 ноября 2018 года.

Fortran 2018 включает в себя:

- ISO / IEC TS 29113: 2012 Дальнейшая совместимость с C
- ISO / IEC TS 18508: 2015 Дополнительные параллельные функции в Fortran
- Поддержка ISO / IEC / IEEE 60559: 2011, шестнадцатеричный ввод / вывод, усовершенствования IMPLICIT NONE и другие изменения.

Типы данных

INTEGER*1	От -127 до +127
INTEGER*2	От -32767 до +32767
INTEGER*4	От -2147483647 до +2147483647
INTEGER	Тот же, что для INTEGER*2 или INTEGER*4
REAL*4	От -3.4028235E+38 до -1.1754944E-38, число 0
REAL	и от +1.1754944E-38 до +3.4028235E+38
REAL*8	От -1.797693134862316D+308
DOUBLE PRECISION	до -2.225073858507201D-308, число 0
	и от +2.225073858507201D-308
	до +1.797693134862316D+308
COMPLEX*8	Для действительной и мнимой частей такой же,
COMPLEX	как для целых и вещественных чисел
COMPLEX*16	
LOGICAL*1	.TRUE. или .FALSE.
LOGICAL*2	
LOGICAL*4	
LOGICAL	
CHARACTER	Любые символы ПЭВМ
CHARACTER*длина	
CHARACTER*(*)	

Производные типы данных.
Оператор описания TYPE

- Логические операции и константы

Отношения
.GT.
.LT.
.GE.
.LE.
.NE.
.EQ.

Операции
.NOT.
.OR.
.AND.
.EQV.
.NEQV.

Константы
.FALSE.
.TRUE.

Специальные символы
=, **, *, / , +, -
Отношения <, >, ==, /=, <=, >=
( ) параметры подпрограммы, индексы массива
; разделитель операторов в свободном формате
// объединение строк
& признак переноса оператора на след. Строку
:: разделитель в предложениях описания
! начало комментария
Для компилятора g77
C или * в первой позиции строки .

Создание исходной программы

Описание переменных. Определение типа.

а) неявным связыванием имени переменной с её типом

I, J, K, L, M, N --- по умолчанию INTEGER , все другие --- REAL

б) оператор IMPLICIT NONE изменяет правила неявного определения типа. Нужно всё описывать.

Массивы.

Переменная с индексом является элементом массива A(I,j). Описание DIMENSION A(-5:10,100)

(Фортран 90 - REAL, DIMENSION (10,10) :: A)

Некоторые операторы.

1) Присваивания A=C*B+3.5

2) Условия IF блочный (IF THEN) : IF(логическое -выражение) THEN

IF логический : IF(логическое-выражение) оператор

3) Ветвление SELECT CASE (выражение) (нет в Фортране 77)

4) Цикл. [Имя:] DO параметр цикла= начальное(выр.1), конечное(выр.2), приращение(выр.3)

блок_действий (операторы)

END DO [Имя]

Бесконечный цикл. DO без параметра. Выход из цикла EXIT [имя]. Следующая итерация CYCLE [имя]

5) Цикл с предусловием. DO WHILE (логическое –выражние)

блок_действий (операторы)

END DO

6) Операторы I/O. Достаточно сложный оператор - FORMAT. Можно использовать по умолчанию

READ (*,*) имя_переменной ; WRITE (*,*) имя_переменной ; WRITE(*,’(1x,A$)’) ‘Input x=‘

7) Безусловный оператор перехода. GO TO метка

8) DATA задает начальное значение переменным. DATA список_объектов/список_значений/[,]…

СТРУКТУРА ПРОГРАММЫ

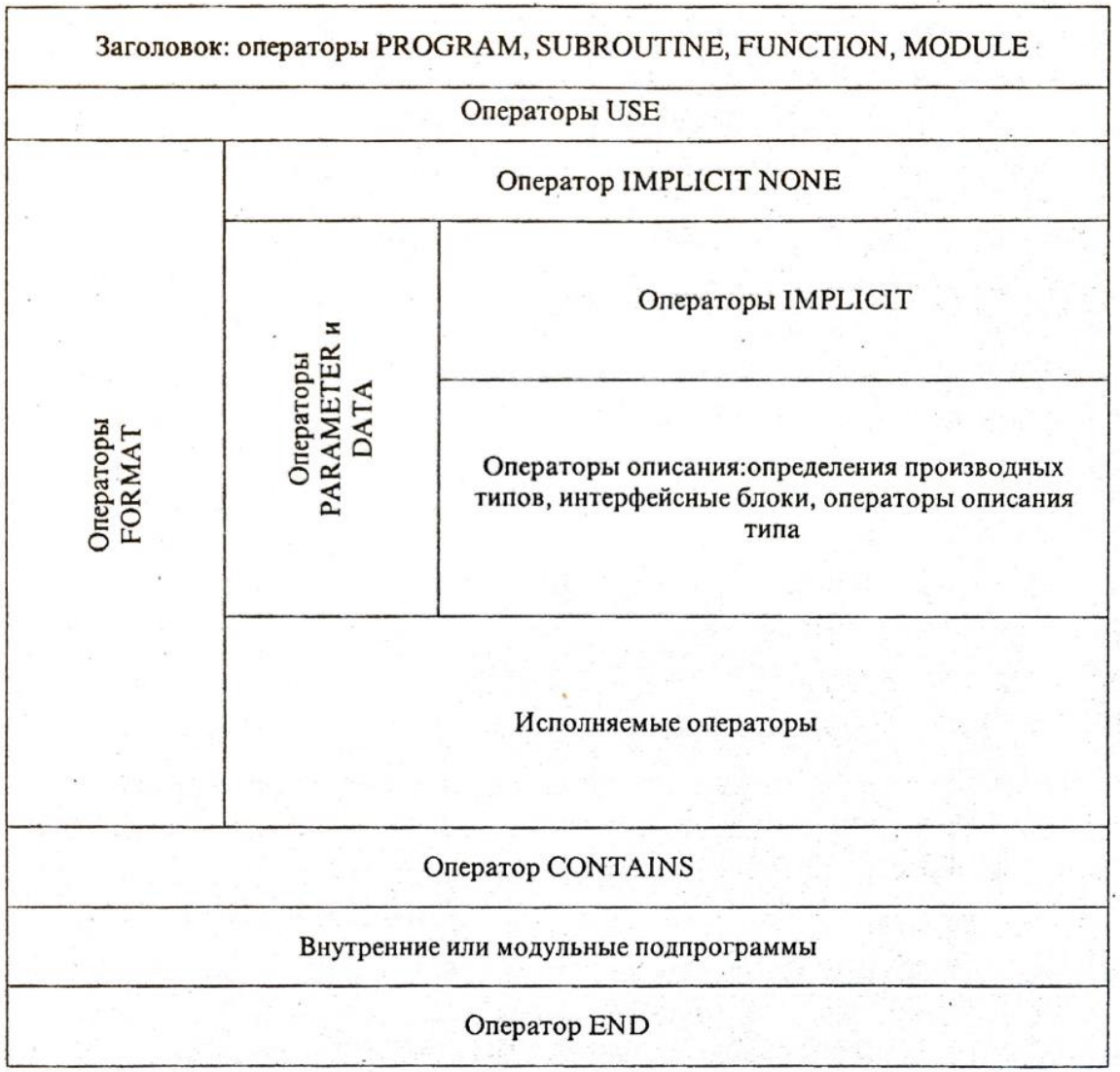
Программа на Фортране состоит из главной программы main и, возможно, нескольких подпрограмм и модулей (Фортран 90).
В Фортране выделены три вида независимых программных компонент: *главная программа, внешняя подпрограмма и модуль.*

Главная PROGRAM имя_программы
Внешние подпрограммы:
а) функциями типа FUNCTION имя
б) процедурами типа SUBROUTINE имя
Модуль : MODULE имя

Пример написания программы в фиксированном формате на Фортран 77.

```
C       Simple example
        PROGRAM SUMMING
        SUM=0
        DO I=1,20
                SUM=SUM+I
        END DO
        WRITE(*,*) 'SUM=', SUM
        END
```

Порядок операторов



Правила записи арифметических выражений

1) Знак умножения опускать нельзя AB A*B
2) Запись двух следующих друг за другом операций A/-B A/(-B)
3) Вычисления не выходят за область значений целых и вещественных величин.
4)Результат деления двух операндов целого типа является целое частное 17/5=3 ; 1/5=0 (выход - 1.0/5=0.2)
5)Операция возведения в степень выполняется по-разному :

Если показатель целый – то возведение в степень выполняется как многократное умножение;

Если показатель вещественного типа – то возведение в степень выполняется по эквивалентной формуле $x^y = e^{ylnx}$.

Поэтому необходимо учитывать следующее:

а) 0^0 - не определено; $0^{-5} = (\frac{1}{0})^5$ - возникает ошибка деления на ноль

б) отрицательную величину вещественного типа допускается возводить только в целую степень

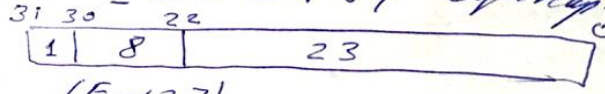
-3.1**5 - можно

-3.1**5.0 – нельзя (т.к. компьютер будет вычислять $e^{5\ln(-3.1)}$ - ошибка «вычисление логарифма отрицательного числа)

в) комплексную константу можно возводить только в целую степень

6) Арифметические операции выполняются над операндами одного и того же типа, который становится так же типом результата операции (при разном типе затрачивается дополнительное время на приведение величин к одному типу)
7) Важный вопрос потеря точности при выполнении арифметических операций.
8) Избегать сравнения на равенство для вещественных чисел A.EQ.B A-B.LE.EPSILON

Формула для получения десятичного числа из числа IEEE754 одинарной точности.



$$F = (-1)^S \, 2^{(E-127)} \left(1 + M/2^{23}\right)$$

(1 — это единица, которую отбросили)

S - бит знака (31-й бит)

E - смещенная экспонента (30-23 биты)

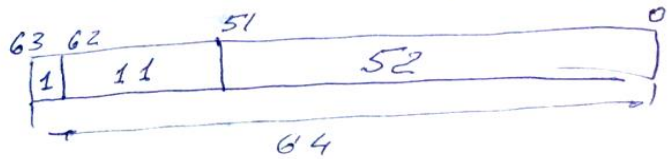
M - остаток от мантиссы (22-0 биты)

F - десятичное число.

(у нормализованной двоичной мантиссы первый бит всегда равен 1, так как число имеет в диапазоне $1 <= M < 2$)
Эту „1" нет смысла записывать в освободившиеся 23 бита, поэтому записывают остаток от мантиссы!

Формат числа двойной точности

$$F = (-1)^S \, 2^{(E-1023)} \left(1 + M/2^{52}\right)$$



Среди форматов IEEE 754, имеются:
(есть и другие)

- формат с простой точностью — 32-разрядный формат, 24 разр. мантиссе и знаку
  8 разрядов для порядка (смещение = 127)

- Формат с двойной точностью — 64-разрядное число,
  53 разряда отводятся мантиссе и знаку
  11 — порядку (смещение = 1023)

Нужно представлять как?

выполняются арифметические операции с вещ. числами

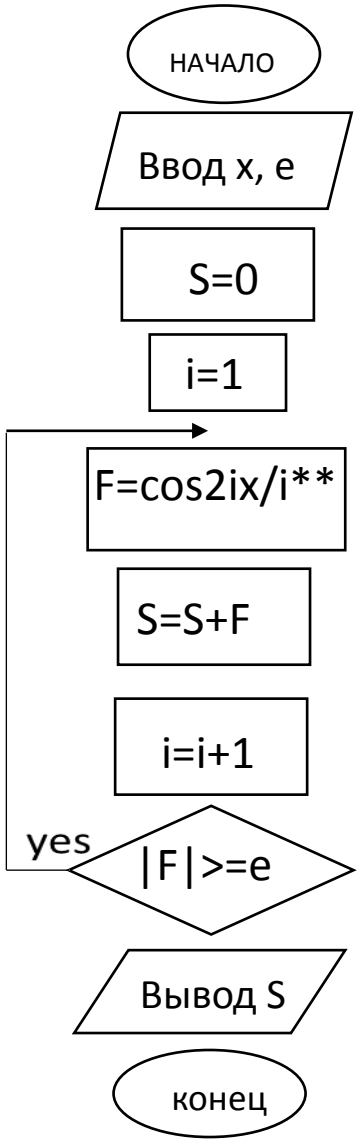
1) числа загружаются в спец регистры Арифм-логич устр (АЛУ) сбс.
2) числа нормализованы.
3) при (+/-) вначале производится выравнивание порядков
   а) у числа с меньшим порядком мантисса сдвигается вправо на кол-во разрядов, равное разности порядков обоих операндов
   б) разряды с равным весом будут расположены в соответствующих разрядах регистра
   в) потом мантиссы +/-, а порядок изменяется только если необходимо нормализовать результат операции.
4) при ($*_{умн}$) — порядки складываются; мантиссы перемножаются.
5) при ($/_{дел}$) — порядки вычитаются мантисса делимого делится на мантиссу делителя

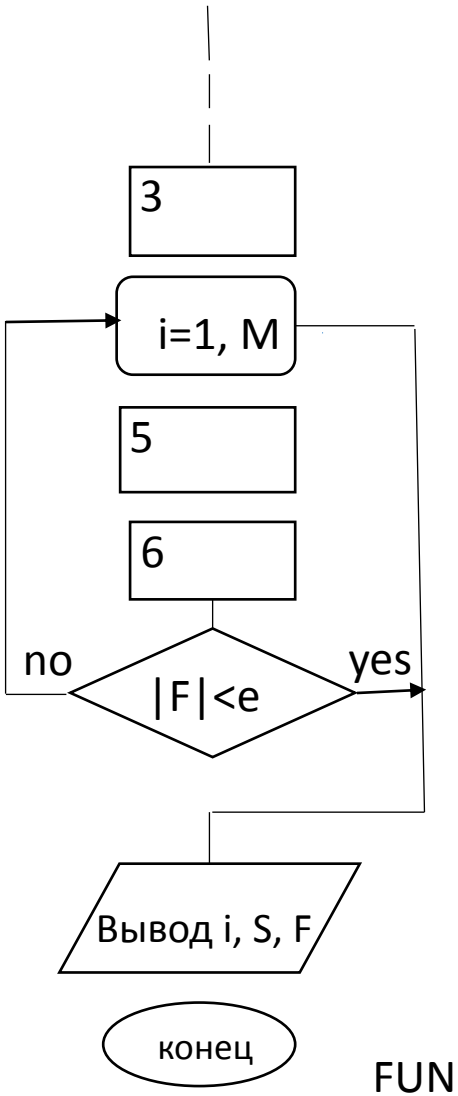
При необходимости результат нормализуется.

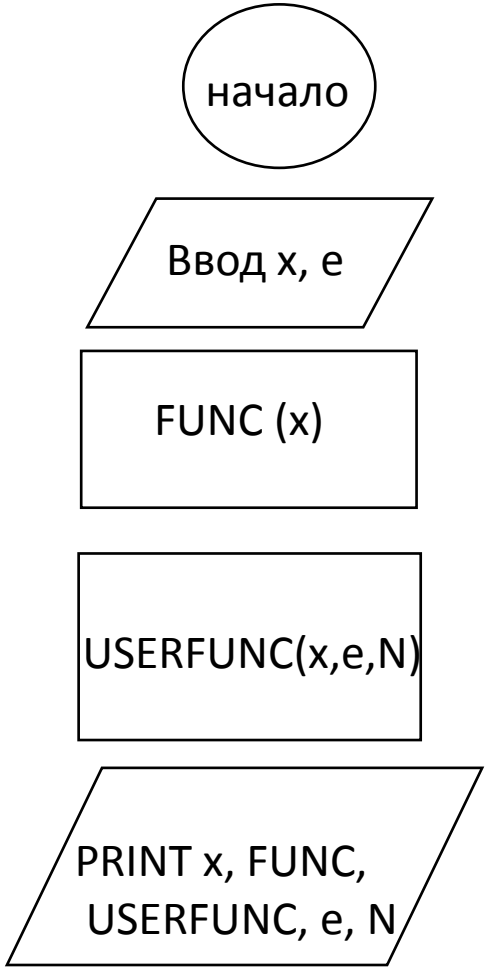
Деление для компьютера → самая сложная арифметическая операция.

Итерационные циклы

S=cos2x +cos4x/4 +cos6x/9 + ……..=∑ cos2ix/i**







FUNCTION имя ф-ции (M, eps)

1)

$$\frac{\pi^4}{96}=\sum_{k=0}^{\infty}\frac{1}{(2k+1)^4}$$

Рекуррентная формула для нахождения π с погрешностью EPS1

2)

Для заданного X вычислить значение функции y=x(π-x) , используя равенство

$$x(\pi - x) = \frac{\pi^2}{6} - \sum_{k=1}^{\infty}\frac{\cos 2kx}{k^2}$$

с погрешностью EPS2

3) exp (x); 4) sin x; 5) cos x; 6) ln(1+x); 7) arctg x; 8) sh x; 9) ch x; 10) ln(1-x); 11) 1/(1+x); 12) 1/(1-x) ; 13) ln(1+x)/(1-x);