

Notas (breves) de FORTRAN 90.

Introducción.

En el presente curso aprenderemos a programar en Fortran 77, pero sabiendo esto, no nos será difícil programar en Fortran 90 o 95. En estas breves notas, introduciremos las principales diferencias entre la programación en Fortran 77 y 90/95. Para una lectura más extensa y detallada se puede consultar la bibliografía sugerida por la cátedra.

Para distinguirlo de un programa fuente escrito en Fortran 77, a nuestro programa fuente en Fortran 90 lo llamaremos *Nombre.f90* en vez de *Nombre.f*. Además, al darle la extensión *.f90*, el editor *emacs* (y otros) reconocerá que se trata de un programa escrito en Fortran 90 y podremos utilizar todas las funcionalidades que brinda el *emacs* para este tipo de archivos (tabulaciones, sangrías, colores de los comandos, etc.).

Al momento de compilar nuestro programa fuente en Fortran 90, lo haremos de la misma forma en que lo haríamos en Fortran 77 con la única diferencia del nombre del archivo a compilar, es decir:

```
> gfortran -Wall -o Nombre-ejecutable Nombre.f90
```

Con lo visto hasta ahora, las diferencias con la programación en Fortran 77, residen en cuestiones de sintaxis.

La primera diferencia de sintaxis que mencionaremos es que no necesitamos empezar a escribir en la séptima columna, sino que podemos hacerlo desde la primera.

Una línea puede tener una longitud de hasta 132 caracteres (hasta la columna 72 en Fortran 77). Para indicar que continúa en el renglón siguiente, usamos el símbolo `&` al final de la misma.

Además para indicar un comentario se usa el signo de admiración `!`, no usándose la `C`.

Otra diferencia es que es necesario al final del programa indicar qué se está finalizando, por ejemplo:

```
PROGRAM PPAL
...
...
END PROGRAM PPAL
```

Al momento de declaración de tipo de variables y de parámetros, hay unas pequeñas diferencias en la sintaxis.

En el siguiente ejemplo mostramos cómo se declaran los distintos tipos de variables. Cabe aclarar que las variables así declaradas están en simple precisión y

más adelante explicaremos cómo se declaran en doble u otras precisiones. Nótese que debemos colocar en todos los casos, los `::` y además la longitud de la cadena de caracteres se coloca entre paréntesis.

```
PROGRAM PPAL
  IMPLICIT NONE
  REAL :: A, B, C
  INTEGER :: I, J
  CHARACTER(19) :: NOM
  COMPLEX :: Z, W
  LOGICAL :: CLAVE
  REAL, PARAMETER :: D = 3.24
  REAL, PARAMETER :: PI = 4.0 * ATAN(1.0)
  REAL :: T = 2.23
  ...
  ...
END PROGRAM PPAL
```

En cuanto al uso del `PARAMETER`, en Fortran 90 podemos realizar operaciones al momento de asignarle un valor, algo que no podíamos hacer en Fortran 77. Otra cosa que en Fortran 77 no podíamos hacer era asignarle un valor a una variable en el momento en que la declarábamos, esto sí es posible en Fortran 90. A esta variable (`T` en nuestro ejemplo), podemos asignarle otro valor en el programa, cosa que no podemos hacer con las variables `PI` y `D` de nuestro ejemplo, asignadas mediante un `PARAMETER`.

A modo de ejemplo, transcribimos a continuación en Fortran 90 los programas fuentes para calcular el volumen de una esfera y el área de un triángulo.

```
PROGRAM VOLUMEN_ESFERA
  ! Declaración de tipo de las variables.
  REAL :: VOLUMEN, RADIO
  REAL, PARAMETER :: PI = 4.0 * ATAN(1.0)
  ! Impresión en pantalla.
  WRITE(*,*) "Este programa calcula"
  WRITE(*,*) "el volumen de una esfera"
  WRITE(*,*)
  ! Ingreso de datos.
  WRITE(*,*) "Ingrese el radio de la esfera"
  READ(*,*) RADIO
  ! Cálculo del volumen de una esfera.
  VOLUMEN = (4.0/3.0) * PI * RADIO**3
  ! Impresión del resultado.
  WRITE(*,*) "Volumen de la esfera:", VOLUMEN
  ! Finalización del programa.
END PROGRAM VOLUMEN_ESFERA
```

```
PROGRAM TRIANGULO
  ! Declaración de tipo de las variables.
  REAL :: LADO1, LADO2, LADO3, S, AREA
  ! Impresión en pantalla.
  WRITE(*,*) "Este programa calcula el area"
  WRITE(*,*) "de un triangulo cualquiera."
  WRITE(*,*)
  WRITE(*,*) "Escriba las longitudes de los"
  WRITE(*,*) "lados separadas por comas."
  ! Lectura de los datos.
  READ(*,*) LADO1, LADO2, LADO3
```

```
  ! Cálculo auxiliar.
  S = 0.5 * (LADO_1 + LADO_2 + LADO_3)
  ! Cálculo del area.
  AREA = SQRT(S*(S-LADO1)*(S-LADO2)*(S-LADO3))
  ! Impresión del resultado por pantalla.
  WRITE(*,*)
  WRITE(*, "(A24,1X,F7.3)")                                &
    "El area del triangulo es ", AREA
  ! Finalización del programa.
END PROGRAM TRIANGULO
```