

## COMPILANDO EN GNU/LINUX

### El compilador FreePascal

El compilador FreePascal nació el julio de 1993 impulsado por su autor Florian Klaempfl. Casi 7 años después de su inicio, salió la versión 1.00 del compilador, suficientemente estable como para poder desarrollar aplicaciones, actualmente se encuentra en la versión 2.6.2, aunque nosotros trabajaremos con la versión 2.4.0 por cuestiones de compatibilidad de programas. Se distribuye bajo licencia GNU GPL que permite, básicamente, su distribución gratuita y del código fuente sin ningún coste aunque los autores retienen el copyright.

El compilador FreePascal existe para plataformas MS-DOS, MS-Windows, GNU/Linux, OS/2, AmigaOs, \*BSD, etc. Está limitado a las arquitecturas Intel y Motorola.

El compilador puede ser descargado de <http://www.freepascal.org>

El alumno puede instalar la versión 2.4.0 siguiendo los pasos dados en el tutorial publicado por la cátedra.

Como característica interesante hay que decir que soporta muchas de las características del Pascal de Borland y del ObjectPascal de Delphi y además incluye nuevas posibilidades inexistentes en estos compiladores, como es la sobrecarga de operadores, por poner un ejemplo. También aporta una librería estándar que funciona en todas las plataformas que da soporte el compilador.

Pascal define dos tipos básicos de archivos que se pueden compilar: los programas y las units. Los archivos de código fuente que se escriben con un editor de texto o con un IDE conviene que tengan extensión **.pp** o bien **.pas**. Al compilar programas obtendremos archivos ejecutables que tendrán el mismo nombre que el archivo fuente y carecerán de extensión mientras que si compilamos units obtendremos archivos **.ppu** que no son ejecutables.

Además de un compilador y un linker, FreePascal a partir de su versión 2 incluye un IDE muy similar al IDE de los productos Borland (Turbo Pascal, Borland C, etc.) llamado **fp**. Para acceder a este IDE se debe tipear el siguiente comando:

**fp <programa\_pascal.pp>**

Además es importante que el alumno tenga en cuenta que Freepascal brinda extensa documentación que se instala con el compilador.

Dentro de la documentación podemos resaltar:

- **Free Pascal Users manual (user.pdf):** Describe la instalación y uso del compilador de Freepascal en las diferentes plataformas soportadas.
- **Free Pascal Programmers manual (prog.pdf):** Describe las particularidades del compilador Freepascal, y describe a grandes rasgos el funcionamiento del mismo.
- **Free Pascal Reference guide (ref.pdf):** Describe todas las constantes, tipos, variables, funciones y procedimientos definidas en la unit System. Y en menor medida todas las estructuras soportadas y lista los tipos soportados.
- **Free Pascal Run-Time Library Reference guide (rtl.pdf):** Describe todas las constantes, tipos, variables, funciones y procedimientos definidas en las units que están instaladas con el compilador Freepascal.

La ubicación de esta documentación difiere de un sistema operativo a otro, e inclusive de una distribución a otra. En nuestro caso la documentación se encontrará en el directorio **/usr/share/doc/fpc-2.4.0/** ó **/usr/local/doc/fpc-2.4.0/**, dependiendo de la opción elegida durante la instalación.

## Compilando en GNU/Linux

Para emplear el compilador FreePascal tendrá que abrir una consola de texto. En este ejemplo compilaremos uno de los programas de ejemplo más conocidos. Este ejemplo es el típico Hola mundo:

Ingresa al sistema como usuario no privilegiado. Cree un directorio donde almacenar los programas de este práctico dentro de su directorio home. Luego sitúese en el directorio recién creado. Escriba el siguiente programa usando el IDE de FreePascal o bien con un editor de texto cualquiera\*\* y nómbrelo "**prog01.pp**".

```
program Hello;  
  
begin  
    Writeln('Hola Mundo!!!');  
  
end.
```

Para compilar el archivo escriba:

**fpc prog01.pp**

O si está usando el IDE presione la combinación de teclas ALT+F9.

En unos instantes, si todo ha ido bien durante la compilación y el enlazado, obtendrá un archivo ejecutable con el nombre **prog01**. Si lo ejecuta obtendrá la salida : Hola Mundo!!! En caso que hubiera obtenido el error de Comando o nombre de archivo incorrecto revise que el directorio donde está alojado el compilador de freepascal se encuentre incluido en la variable PATH.

\*\* Respecto de qué editor de texto utilizar, podemos emplear cualquier editor de texto (en modo texto o en modo gráfico) que permita guardar los archivos en formato texto ascii. Es recomendable emplear algún editor con el cual usted se sienta cómodo y que incorpore funciones de resaltado de sintaxis.

Mi recomendación sería: para editar en modo texto **mcedit** (el editor del Midnight Commander), o bien algún otro editor de texto como **vi** o **pico**, si se usa la consola, o bien puede usar algún editor del entorno gráfico y luego compilar por consola

Para ejecutar el programa recién compilado escriba:

**./prog01**

Hace falta el **./** debido a que GNU/Linux no incluye el directorio actual en el PATH. En caso de estar usando el IDE puede ejecutar el programa con las teclas CTRL+F9.

### Compilando unos ejemplos (Repaso de lenguaje Pascal)

Escriba el siguiente programa y nómbrelo “**prog02.pp**” y compílelo y ejecútelo.

```
program Calcular_Area_del_triangulo;

var
    base, altura, area : real;

begin
    base := 2.4;
    altura := 5.3;
    area := 0.5*base*altura;
    Writeln(area);
end.
```

Escriba el siguiente programa y nómbrelo “**prog03.pp**” y compílelo y ejecútelo.

```
program Calcular_R_a_la_N;

uses Math;

var
    r, n : Integer;

begin
    Write('R: ');
    Readln(r);
    Write('N: ');
    Readln(n);
    Write('Resultado: ');
    Writeln(r**n);
end.
```

Escriba el siguiente programa y nómbrelo “**prog04.pp**” y compílelo y ejecútelo.

```
program Calcular_R_a_la_N;
{ Este comentario está hecho con llaves }
uses Math;
var
    r, n : Integer;
    (* Este, en cambio, emplea la notación antigua de Pascal *)
begin
    Write('R: '); // Desde las 2 barras hasta el final de línea es un
comentario
    Readln(r);
    Write('N: ');
    Readln(n);
    Write('Resultado: ');
    Writeln(r**n);
end.
```

Escriba el siguiente programa y nómbrelo “**prog05.pp**” y compílelo y ejecútelo.

```
program Calcular_n_potencias;  
  
uses Math;  
  
var  
    r, n : Integer;  
  
begin  
    Write('R: ');  
    Readln(r);  
    if r > 1 then  
        begin  
            Write('N: ');  
            Readln(n);  
            Write('Resultado: ');  
            Writeln((r**n-1) div (r-1));  
        end;  
    end;  
end.
```

Escriba el siguiente programa y nómbrelo “**prog06.pp**” y compílelo y ejecútelo.

```
program Calcular_n_potencias;  
  
uses Math;  
  
var  
    r, n : Integer;  
  
begin  
    Write('R: ');  
    Readln(r);  
    if r > 1 then  
        begin  
            Write('N: ');  
            Readln(n);  
            Write('Resultado: ');  
            Writeln((r**n-1) div (r-1));  
        end  
    else  
        begin  
            Writeln('R tiene que ser un valor superior a 1');  
        end;  
    end;  
end.
```

Escriba el siguiente programa y nómbrelo “**prog07.pp**” y compílelo y ejecútelo.

```
program Calcular_n_potencias;

uses Math;

var
  r, n, i, suma : Integer;

begin
  Write('R: ');
  Readln(r);
  if r > 1 then
    begin
      Write('N: ');
      Readln(n);
      Write('Resultado: ');
      Writeln((r**n-1) div (r-1));
      // Método mecánico
      suma := 0;
      for i := 0 to n-1 do
        begin
          suma := suma + r**i;
        end;
      Write('Resultado Mecánico: ');
      Writeln(suma);
    end
  else
    begin
      Writeln('R tiene que ser un valor mayor que 1');
    end;
  end.
end.
```

Escriba el siguiente programa y nómbrelo “**prog08.pp**” y compílelo y ejecútelo.

```
program Prueba_Mientras;

var
  cadena : string;

begin
  cadena := ''; // Una cadena vacía
  while cadena <> '.' do
    begin
      Write('Escriba algo (Un punto para terminar) : ');
      Readln(cadena);
    end;
  Writeln('Fin del programa');
end.
```

Escriba el siguiente programa y nómbrelo “**prog09.pp**” y compílelo y ejecútelo.

```
program Prueba_Repetir;  
  
var  
    nombre : string;  
  
begin  
    repeat  
        Write('Escriba su nombre: ');  
        Readln(nombre);  
    until nombre<>'';  
end.
```

Escriba el siguiente programa y nómbrelo “**prog10.pp**” y compílelo y ejecútelo.

```
program Dias_Semana;  
  
var  
    diasemana : Integer;  
  
begin  
    Write('Introduzca el día de la semana: ');  
    Readln(diasemana);  
    case diasemana of  
        1 : begin  
            Writeln('Lunes');  
            end;  
        2 : begin  
            Writeln('Martes');  
            end;  
        3 : begin  
            Writeln('Miercoles');  
            end;  
        4 : begin  
            Writeln('Jueves');  
            end;  
        5 : begin  
            Writeln('Viernes');  
            end;  
        6 : begin  
            Writeln('Sábado');  
            end;  
        7 : begin  
            Writeln('Domingo');  
            end;  
        else  
            begin  
                Writeln('Número no válido');  
            end;  
    end;  
end.
```

Escriba el siguiente programa y nómbrelo “**prog11.pp**” y compílelo y ejecútelo.

```
program Funciones;

var
  n, m : integer;

function max(a, b : integer) : integer;
begin
  if a >= b then
    begin
      max := a
    end
  else
    begin
      max := b;
    end;
  end;

begin
  Write('Introduzca el valor entero M: '); readln(m);
  Write('Introduzca el valor entero N: '); readln(n);
  Writeln('El máximo entre ', M, ' y ', N, ' es ', max(m, n));
end.
```

Escriba el siguiente programa y nómbrelo “**prog12.pp**” y compílelo.

```
program Variables;

var
  a : integer;

procedure Asignar;
var
  b, c : integer;
begin
  a := 15;
  b := 10;
  c := b;
  d := b;
end;

p
var
  b, d : integer;

begin
  a := 10;
  b := 5;
  Asignar;
  c := 4;
end.
```

Por qué se producen los errores? Corrija el programa compílelo y ejecútelo para verificar que funciona correctamente.