



UNIVERSIDADE DA CORUÑA

FACULTADE DE INFORMÁTICA Departamento de Computación Estructura de Datos e da Información

Práctica 0a: Compilador y Punteros

1. Entono de trabajo:

- El sistema operativo: Linux (Fedora, Ubuntu) o WindowsXP Pro.
- Los editores:
 - Fedora: NEdit, Vimp, Emacs, XEmacs.
 - Ubuntu: Gedit, Emacs, XEmacs.
- El compilador: FreePascal 2.0.0 (local).
- Modo de compilación en línea:
 - Linux: apertura de terminal (En Fedora Menu->Herramientas del sistema, en Ubuntu Menu->Accesorios)
 - WinXP: apertura de ventana de comandos (Inicio->Accesorios->Simbolo de Sistema)
 - Teclear: `fpc <fuente>.pas`.
- ¿Entornos de desarrollo: IDE FreePascal (WinXP), KDevelop (Linux), etc...?

2. Errores de compilación vs ejecución con punteros. Asumir las siguientes declaraciones:

```
var
  x: integer;
  P1, P2: ^integer;
  Q1, Q2: ^real;
```

¿Qué es incorrecto (si lo hay) en cada una de las sentencias?

- a) `writeln(P1);` Error de compilación: las variables de tipo puntero no se “leen” o “escriben” en sentencias de este tipo.
Error: Can't read or write variables of this type
- b) `P1:= Q1;` Error de compilación: las variables son de tipos distintos.
Error: Incompatible types: got ^S64REAL expected ^SMALLINT
- c) `if P1^ = nil then Q1:= Q2;` P1 es una variable de tipo puntero; no así P1^, la variable dinámica asociada, de tipo entero. La comparación no se puede realizar.
- d) `readln(P1^)` Correcto.
- e)

```
  readln(P1^);
  writeln('El valor es ', P1^);
```

¿Correcto?
- f) `new(X)` X no es una variable de tipo puntero.
Error: pointer type expected, but got SMALLINT
- g) `Q1^:=P1^` Correcto.
- h)

```
begin
  P1^:= 17;
  new(P1);
end
```

No hay error de compilación, sino de ejecución puesto que no se ha hecho reserva alguna para el puntero P1, y su valor es indefinido.

3. Acceso al puntero después de liberado.

```
new(p1);
p1^:= 123;
writeln ('p1 es ', p1^);
dispose(p1);
writeln ('p1 es ', p1^);
```

Se produce un error de ejecución.

4. Considerar un tipo de dato `tEstudiante` como un registro con dos campos: nombre y teléfono, y un tipo de dato `pNodo` puntero a registro. Construir un programa para:

- Pedir por teclado los valores y almacenarlos a través del puntero.
- Comparar los dos registros (campo a campo) y determinar si son o no iguales.

Para ello crearemos las operaciones siguientes: `CrearNodo(PNodo)`, `MeterDatos(PNodo)->PNodo` y `Comparar(PNodo, PNodo)->Booleano`.

```
program prueba;

type
  pNodo = ^tEstudiante;
  tEstudiante = record
    nombre: string;
    tlfno: integer;
  end;
var
  P, Q: ^tEstudiante;

Procedure CrearNodo (var P: PNodo);
begin
  New(P);
end;

Procedure MeterDatos (var D: tEstudiante);
begin
  with d do begin
    write('Intro nombre '); readln (nombre);
    write('Intro tlfno '); readln (tlfno);
  end;
end;

Procedure Comparar (P1, P2: pNodo);
begin
  if (p1^.nombre = p2^.nombre) and (p1^.tlfno = p2^.tlfno)
  then writeln ('son iguales')
  else writeln ('son diferentes');
end;

begin
  CrearNodo(P); CrearNodo(Q);
  MeterDatos(P^); MeterDatos(Q^);
  Comparar (P,Q);
  dispose(P); dispose(Q);
end.
```