

Punteros

Introducción a la programación II

Tipos de asignación en memoria

Estática:

- **Se asigna memoria cuando se define la variable.**
- No se puede usar más espacio de lo definido.
- Es necesario declarar los límites.
- Ejemplo: arreglos, matrices, variable de tipo integer, registro.

Dinámica:

- **Se asigna memoria en ejecución (por demanda), y el mismo usuario es responsable de liberarla.**
- Solo se asigna el espacio necesario.
- Su uso es un poco más engorroso:
 - El usuario es el que se encarga de asignar memoria y liberarla.
 - La asignación en variables de tipo punteros genera un resultado diferente a la asignación de variables de otros tipos.
 - El pasaje de parámetros también genera un resultado diferente.

Puntero

Una variable de tipo puntero es una **variable cuyo valor es una dirección de memoria**.

Una variable que no es de tipo puntero puede suponerse que su nombre representa una dirección de memoria, en cuyo lugar hay un valor ocupando un espacio según el tipo definido.

Ejemplo:

```
var
    variableentero:integer;
    variablepuntero:tipopuntero;
begin
    ...
    variableentero:=5;
    ...
end;
```

Variable	Dirección de memoria	Contenido
variableentero	100	5
...	101..150	...
variablepuntero	151	?
...

Puntero

Declarar una variable de tipo puntero es especificar el nombre de una variable y el tipo de valor que se almacenará en la dirección de memoria a la que el puntero se refiere.

tipopuntero = \wedge <tipo de valor>;

Ejemplo:

type

tipopunteroentero = \wedge integer;

var

variablepunteroentero:tipopunteroentero;

Variable	Dirección de memoria	Contenido
variablepunteroentero	151	Dirección de memoria donde habrá un entero
...

Operaciones con punteros

Antes de que programa utilice una variable puntero se requiere hacer un espacio en memoria para el tipo de dato al que va hacer referencia el puntero:

new(variabletipopuntero);

Durante la resolución del problema se pueden realizar asignaciones de valores a lo apuntado por la variable tipo puntero:

variabletipopuntero^:=valor o variable del tipo que hace referencia el puntero;

O asignaciones entre punteros:

variable1tipopuntero:=variableNtipopuntero;

Las comparaciones entre punteros solo puede darse con expresiones de igualdad; = y <>.

Una vez finalizado el uso del puntero deberá liberar la memoria asignada a la variable puntero:

dispose(variabletipopuntero);

Operaciones con punteros

Ejemplo:

type

tipopunteroentero = ^integer;

var

varpument1, varpument2:tipopunteroentero;

variableentero:integer;

begin

new(varpument1);

variableentero:=8;

varpument1^:=variableentero;

varpument2:=varpument1;

varpument2^:=5;

dispose(varpument1);

end;

Variable	Dirección de memoria	Contenido
varpument1	100	?
...	101..150	...
varpument2	151	?
...	152..199	...
variableentero	200	Un entero sin valor inicial
...

Operaciones con punteros

Ejemplo:

type

tipopunteroentero = ^integer;

var

varpunter1, varpunter2:tipopunteroentero;

variableentero:integer;

begin

new(varpunter1);

variableentero:=8;

varpunter1^:=variableentero;

varpunter2:=varpunter1;

varpunter2^:=5;

dispose(varpunter1);

end;

Variable	Dirección de memoria	Contenido
varpunter1	100	301
...	101..150	...
varpunter2	151	?
...	152..199	...
variableentero	200	8
...	201..300	...
varpunter1^	301	Un entero sin valor inicial
...

Operaciones con punteros

Ejemplo:

type

tipopunteroentero = ^integer;

var

varpuntero1, varpuntero2:tipopunteroentero;

begin

variableentero:integer;

new(varpuntero1);

variableentero:=8;

varpuntero1^:=variableentero;

varpuntero2:=varpuntero1;

varpuntero2^:=5;

dispose(varpuntero1);

end;

Variable	Dirección de memoria	Contenido
varpuntero1	100	301
...	101..150	...
varpuntero2	151	301
...	152..199	...
variableentero	200	8
...	201..300	...
varpuntero1^ y varpuntero2^	301	8
...

Operaciones con punteros

Ejemplo:

type

tipopunteroentero = ^integer;

var

varpuntero1, varpuntero2:tipopunteroentero;

variableentero:integer;

begin

new(varpuntero1);

variableentero:=8;

varpuntero1^:=variableentero;

varpuntero2:=varpuntero1;

varpuntero2^:=5;

dispose(varpuntero1);

end;

Variable	Dirección de memoria	Contenido
varpuntero1	100	301
...	101..150	...
varpuntero2	151	301
...	152..199	...
variableentero	200	8
...	201..300	...
varpuntero1^ y varpuntero2^	301	5
...

Operaciones con punteros

Ejemplo:

type

tipopunteroentero = ^integer;

var

varpuntero1, varpuntero2:tipopunteroentero;

begin

variableentero:integer;

new(varpuntero1);

variableentero:=8;

varpuntero1^:=variableentero;

varpuntero2:=varpuntero1;

varpuntero2^:=5;

dispose(varpuntero1);

end;

Variable	Dirección de memoria	Contenido
varpuntero1	100	?
...	101..150	...
varpuntero2	151	?
...	152..199	...
variableentero	200	8
...	201..300	...
...	301	...
...

Ejemplo moodle

//Ejemplo: Simulacion de asignaciones entre dos punteros.

program Punteros;

type

tipopunteroentero = ^integer;

var

varpument1, varpument2:tipopunteroentero;

variableentero:integer;

begin

new(varpument1);

variableentero:=8;

writeln('En este punto varpument1 es un puntero cuyo valor es una direccion de memoria');

writeln('y varpument1^ tiene espacio asignado con cualquier valor: ',varpument1^);

varpument1^:=variableentero;

writeln('A varpument1^ se le asigna un valor: ',varpument1^);

varpument2:=varpument1;

writeln('varpument2 y varpument1 tienen la misma direccion de memoria');

writeln('y por lo tanto varpument2^ es: ',varpument2^);

varpument2^:=5;

writeln('Cambiar el contenido de los apuntado por medio de varpument2');

writeln('impacta en de igual forma en varpument1 y en el contenido de varpument1^: ',varpument1^);

dispose(varpument1);

writeln('Liberar la memoria de varpument1 hace que varpument2 apunte a una direccion');

writeln('con datos desconocidos, asi varpument2^ es : ',varpument2^);

end.

Inicialización de punteros

NIL es una constante que se utiliza para dar un valor a una variable de tipo puntero que no apunta a ninguna dirección de memoria.

variabletipopuntero:=NIL;

Observación: la asignación indebida de punteros puede provocar la generación de memoria que es inaccesible.

type

tipopunteroentero = ^integer;

var

varpunter:tipopunteroentero;

begin

new(varpunter);

varpunter^:=8;

varpunter:=NIL;

varpunter^:=8;

...;

end;

Variable	Dirección de memoria	Contenido
varpunter	100	?
...
...
...

Inicialización de punteros

NIL es una constante que se utiliza para dar un valor a una variable de tipo puntero que no apunta a ninguna dirección de memoria.

variabletipopuntero:=NIL;

Observación: la asignación indebida de punteros puede provocar la generación de memoria que es inaccesible.

type

tipopunteroentero = ^integer;

var

varpunter:tipopunteroentero;

begin

new(varpunter);

varpunter^:=8;

varpunter:=NIL;

varpunter^:=8;

...;

end;

Variable	Dirección de memoria	Contenido
varpunter	100	301
...
varpunter^	301	?
...

Inicialización de punteros

NIL es una constante que se utiliza para dar un valor a una variable de tipo puntero que no apunta a ninguna dirección de memoria.

variabletipopuntero:=NIL;

Observación: la asignación indebida de punteros puede provocar la generación de memoria que es inaccesible.

type

tipopunteroentero = ^integer;

var

varpunter:tipopunteroentero;

begin

new(varpunter);

varpunter^:=8;

varpunter:=NIL;

varpunter^:=8;

...;

end;

Variable	Dirección de memoria	Contenido
varpunter	100	301
...
varpunter^	301	8
...

Inicialización de punteros

NIL es una constante que se utiliza para dar un valor a una variable de tipo puntero que no apunta a ninguna dirección de memoria.

variabletipopuntero:=NIL;

Observación: la asignación indebida de punteros puede provocar la generación de memoria que es inaccesible.

type

tipopunteroentero = ^integer;

var

varpunter:tipopunteroentero;

begin

new(varpunter);

varpunter^:=8;

varpunter:=NIL;

varpunter^:=8;

...;

end;

Variable	Dirección de memoria	Contenido
varpunter	100	NIL
...
...	301	8
...

Inicialización de punteros

NIL es una constante que se utiliza para dar un valor a una variable de tipo puntero que no apunta a ninguna dirección de memoria.

variabletipopuntero:=NIL;

Observación: la asignación indebida de punteros puede provocar la generación de memoria que es inaccesible.

type

tipopunteroentero = ^integer;

var

varpunter:tipopunteroentero;

begin

new(varpunter);

varpunter^:=8;

varpunter:=NIL;

varpunter^:=5;

ERROR!!!

...;

end;

Variable	Dirección de memoria	Contenido
varpunter	100	NIL
...
...	301	8
...

Punteros a registros

Es uno de los tipos de punteros más utilizados y es la base para armar estructuras más complejas.

type
nombre_reg = record
campo1;
...
campoN;
end;
tipopunteroregistro = ^nombre_reg;

Ejemplo

type
cuenta=record
numcliente:integer;
saldocliente:real
end;
puncta=^cuenta;
var
varpuncta:puncta;
begin
new(varpuncta);
//varpuncta es un puntero, varpuncta^ es un registro, varpuncta^.numcliente es un campo del registro
...
varpuncta^.numcliente:=100;
varpuncta^.saldocliente:=0.0;
...
end.

La asignación de valores a los campos del registro se hace uno por uno.