Introducción a los punteros

Declaración de un puntero

Un puntero, en C, se declara como sigue:

```
TIPO * nombre_puntero ;
```

Donde TIPO es cualquier tipo definido. Asi, un puntero a caracter se declararia de la siguiente forma:

```
char *pchar;
```

Un puntero a una estructura, se declara como sigue:

```
Typedef struct estructura * nombre_puntero ;
```

Diferencia entre "*" y "&"

En C, al contrario que en otros lenguajes de programacion, se puede obtener directamente la direccion de memoria de cualquier variable. Esto es posible hacerlo con el operador unario "&"; asi:

```
char a;         /* Variable 'a' de tipo char */
char *pchar;    /* Variable 'pchar' de tipo puntero a char */
pchar=&a;   /* Coloco en pchar la dirección de a.
```

Como acceder a la información apuntada por un puntero

Si tengo a y pchar declarados como arriba, supongamos asignamos un valor a a

como sigue:

a=’b’;

podemos asignar el valor utilizando el puntero del siguiente modo:

*pchar = ‘b’;

Si quiero acceder a un valor es lo mismo si ponemos a o *pchar, por ejemplo:

x=a;

hace lo mismo que

x=*ptr;

Acceso a miembros de un struct usando punteros

```
        Typedef struct punto *punto_ptr;
```

Supongamos punto es:

```
        Struct punto{
                int x;
                int y;};
```

y declaramos una variable de tipo punto y otra de tipo puntoptr que apunta a la primera

```
        punto pt;
        punto_ptr pt_ptr=&pt;
```

para acceder a (o guardar un valor en) el valor x de una estructura apuntada por p, puedo utilizar:

```
       (*pt_ptr).x
```

o

```
        pt_ptr->x
```

Por ejemplo:

```
      (*pt_ptr).x = 4;

      printf("valor de 1er coordenada es %d", (*pt_ptr).x);
```

o con la otra notación:

```
      pt_ptr->x = 4;

      printf("valor de 1er coordenada es %d", pt_ptr->x);
```

Inicialización de un puntero, asignación de memoria

Si declare una variable ptr de tipo puntero a TIPO, puedo inicializar la variable del siguiente modo: la declaración del puntero es:

```
TIPO *ptr;
```

La inicialización de ptr y asignación de memoria para el elemento de tipo TIPO La realizamos con:

```
ptr = new TIPO;
```

Otra forma de hacer lo mismo es : asignar memoria a t de tipo TIPO declarando la variable:

```
TIPO t;
```

Y luego inicializamos ptr con el puntero haciendo

```
ptr=&t;
```