Facultad de Ciencias Exactas y Tecnologías

Universidad Nacional de Santiago del Estero / UNSE

VARIABLES PUNTERO

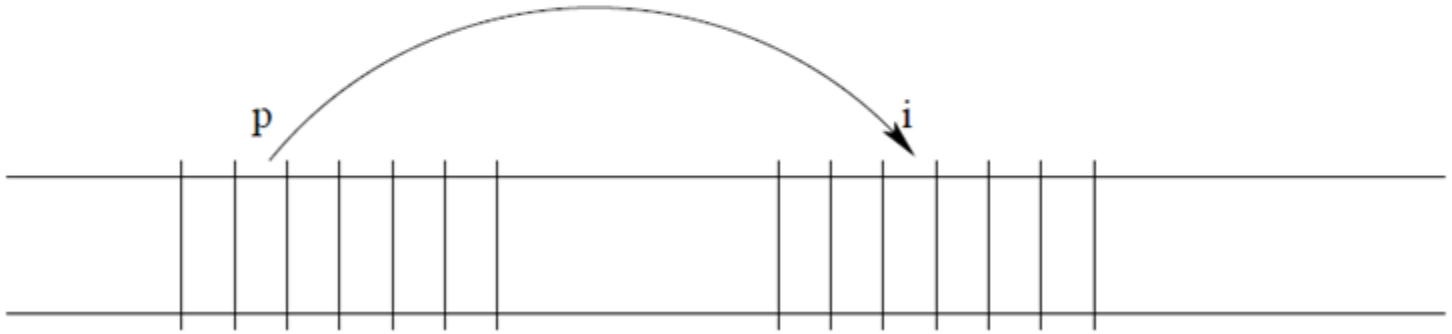
Qué es un puntero?

Consideraciones sobre los punteros

Ing. Analía Méndez

QUÉ ES UN PUNTERO?

Un *puntero* es una variable que contiene la dirección de una variable. El uso de punteros es de más bajo nivel, y ofrece más expresividad. A veces conduce a un código más compacto y en algunos casos más eficiente; la desventaja es que sin cuidado, es fácil cometer errores.



- El operador unario & retorna la dirección de memoria de una variable,

```
p = &i;
```

i es una variable
p es un puntero
el operador & devuelve la dirección de i

CONSIDERACIONES SOBRE LOS PUNTEROS

- El operador unario * es el operador de *indirección* o de *dereferenciación*; cuando se aplica a un puntero, accede al contenido de la memoria apuntada por ese puntero.

```
j = *p; /* j tiene ahora el valor de i */
```

- El lenguaje C nos permite comprobar los tipos de los objetos apuntados por punteros. Esto se logra definiendo para cada tipo T (por ejemplo `int`), un tipo para los puntores asociados (por ejemplo, puntero a `int`). Para definir una variable que apunte a celdas con valores de tipo T, se declara esta como si fuera de tipo T, pero anteponiendo el caracter * al nombre de la variable. Por ejemplo:

```
int *p;
```

Esta notación intenta ser mnemónica, nos dice que la expresión `*p` es un `int`. Aunque cada puntero apunta a un tipo de dato específico, tambien hay un tipo llamado `void *` (sería un "puntero a void" que puede contener cualquier tipo de punteros.

CONSIDERACIONES SOBRE LOS PUNTEROS

- Como cualquier otra variable, es conveniente inicializar un puntero en su declaración. Si el valor no se conocerá hasta más adelante en el código, una buena convención es usar `NULL`. `NULL` es un valor especial de puntero que no referencia a ninguna celda particular.

```
int *p = NULL;
```

- Como los punteros son variables, pueden ser usados sin ser dereferenciados. Por ejemplo, si `p` y `q` son punteros a `int`,

```
p = q;
```

copia el contenido de `q` en `p`, o sea, copia la dirección a la que apunta `q` en `p`, haciendo que `p` apunte a lo mismo que apunta `q`.

CONSIDERACIONES SOBRE LOS PUNTEROS

- Los operadores `&` y `*` asocian con mayor precedencia que los operadores aritméticos. Por lo tanto, cualquiera de las siguientes instrucciones incrementa en 1 el contenido de memoria apuntada por `p`.

```
*p = *p + 1;
*p += 1;
++*p;
(*p)++; /* Notar el uso de paréntesis. */
```

¡Cuidado! Los operadores unarios `*` y `++` asocian de derecha a izquierda, por lo tanto, el uso de paréntesis suele ser necesario.

- El operador `++` aplicado a un puntero hace que apunte a la celda siguiente. Esto suele ser útil al manipular arrays, ya que C nos garantiza que las celdas de un array son contiguas.

BIBLIOGRAFÍA

Introducción a la programación en C

Andrés Marzal - Isabel Gracia

Departamento de Lenguajes y Sistemas Informáticos

Universitat Jaume II