



Puntatori e Array

Luca Abeni



- Il linguaggio C permette di sommare / sottrarre interi al valore di un puntatore
 - ◆ Qual'è il significato di queste espressioni aritmetiche?
 - ◆ Se p è un puntatore ad intero (`int *p;`), cosa indica $p + 3$?
- $p + n$ (con p puntatore ed n intero) è l'indirizzo dell' n -esima variabile dopo quella puntata da p
 - ◆ Se p è un puntatore ad intero, $p + 3$ è l'indirizzo della terza variabile di tipo intero dopo $*p$
 - ◆ Nota: non terza locazione di memoria dopo p (dimensione di una variabile di tipo intero...)

```
int n;
```

```
int *np;
```

```
np = &n;
```

```
printf("p = %p; p + 3 = %p\n", p, p + 3);
```

- Se p vale 1000, $p + 3$ non vale 1003...
- L'indirizzo $p + 3$ è calcolato tenendo conto del numero di byte usati per memorizzare un intero
 - ◆ Se un intero è memorizzato su 4 byte, $p + 3$ vale $1000 + 3 * 4 = 1012$

Sottrazioni fra Puntatori

- Se $p1$ e $p2$ sono puntatori allo stesso tipo di dato, ed n è un intero, allora è lecito scrivere $p2 = p1 + n$
 - ◆ $p2$ punta alla n -esima variabile dopo a $*p1$
- Allora, è anche lecito scrivere sottrazioni del tipo $p2 - p1$
 - ◆ Risultato: variabili del tipo base comprese fra i due puntatori
 - ◆ Nota: $p2$ e $p1$ devono avere lo stesso tipo base
- Un risultato negativo indica che $p2$ punta ad una locazione di memoria precedente a quella puntata da $p1$

Array e Puntatori

- Esiste una relazione stretta fra array e puntatori...
 - ◆ Quando si definisce un array...
 - ◆ Il compilatore crea un alias fra il nome dell'array ed un puntatore al suo primo elemento!!!

- Gli elementi di un array possono essere quindi acceduti in 2 modi:
 1. Tramite il nome dell'array, l'operatore [] e l'indice dell'elemento
 2. Usando il nome dell'array come un puntatore (e l'aritmetica dei puntatori)

Accesso ad Array Tramite Puntatori

- Aritmetica dei puntatori:
 - ◆ Se p è un puntatore al tipo base T e v è un array con tipo base T ($T *p$; e $T v[10]$)
 - ◆ Se $p = \&v[0]$, allora $p + n = \&v[n]$
- Il nome di un array corrisponde ad un puntatore al suo primo elemento:
 - ◆ Se v è un array con tipo base T ($T v[10]$)
 - ◆ Allora scrivere v è equivalente a scrivere $\&v[0]$
- Si deduce che $v[n]$ è un sinonimo di $*(v + n)$
 - ◆ Ancora, zucchero sintattico!!! Le due notazioni sono perfettamente equivalenti, ma $v[n]$ è più compatta e semplice...