ITS TECNOLOGIE DELL'INFORMAZIONE E DELLA COMUNICAZIONE

Tecnico superiore per i metodi e le tecnologie per lo sviluppo di sistemi software

- BACKEND SYSTEM INTEGRATOR

Unità Formativa (UF) **FONDAMENTI DI PROGRAMMAZIONE**

Docente: **VALENTINO ARMANDO**

Titolo argomento: **PUNTATORI A PUNTATORI**

Puntatori a puntatori

1. i puntatori sono variabili come tutte le altre
2. quindi, si può determinare il loro indirizzo
3. differenza fra indirizzo e valore

Abbiamo già visto come l'indirizzo di una variabile sia un numero, che è quindi possibile memorizzare in una variabile. Si è anche visto come il tipo di un indirizzo dipende dal tipo dell'oggetto puntato. Una variabile di tipo puntatore è anche essa una variabile, memorizzata in una zona di memoria, per cui si può trovare il suo indirizzo usando l'operatore `&`.

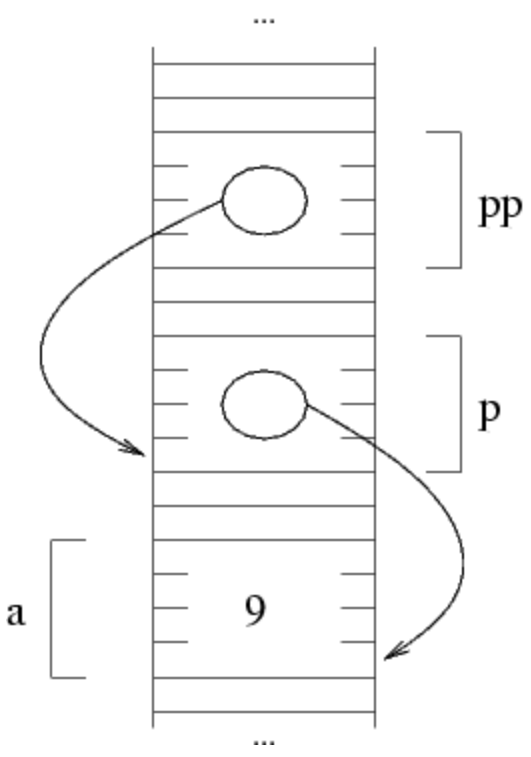
Per essere precisi, il puntatore è una variabile con un tipo, ed anch'esso ha un indirizzo di memori, e se conserviamo il suo indirizzo in un altro puntatore, viene creato quello che si chiama un puntatore a un puntatore. Per memorizzare per esempio l'indirizzo di un puntatore a intero in una variabile, questa deve essere si tipo puntatore a puntatore a intero. Questo tipo si scrive `int **`. Il seguente programma memorizza in una variabile puntatore `p` l'indirizzo della variabile intera `a`, e poi trova l'indirizzo del puntatore `p` e lo memorizza nella variabile puntatore `pp`.

```
/*
  Un esempio di puntatore a puntatore.
*/
int main() {
  int a;
  int *p;
  int **p1;
  a=9;
  p=&a;
  p1=&p;
  printf("Indirizzo di p1=%x, valore=%x\n", &p1, p1);
  printf("Indirizzo di  p=%x, valore=%x\n", &p, p);
  printf("Indirizzo di  a=%x, valore=%x\n", &a, a);
  return 0;
}
```

Quello che si ottiene eseguendo il programma è una stampa del genere:

```
Indirizzo di pp=bffff444, valore=bffff448
Indirizzo di  p=bffff448, valore=bffff44c
Indirizzo di  a=bffff44c, valore=9
```

A parte il valore di **a**, tutti gli altri sono indirizzi, per cui il loro valore non è noto a priori. In una successiva esecuzione del programma si potrebbero ottenere numeri diversi. Quello che però vale sempre è che il valore della variabile **p1** coincide con l'indirizzo della variabile **p**, e che il valore di **p** coincide con l'indirizzo di **a**.



La figura qui accanto mostra una rappresentazione della memoria: **p** contiene l'indirizzo di **a**, e a sua volta **p1** contiene l'indirizzo di **p**.

```
int a=9;
int *p= &a;
Int *p1 =&p;
```

Dato il valore di **p1** è chiaro che è possibile accedere al valore di **a**: basta seguire i puntatori, ossia prima trovare il valore di ***p1**, che è l'indirizzo di **a**, e questo permette di trovare il valore di **a** usando ancora l'operatore *****. Quindi, dato **p1**, il valore di **a** si può trovare con ****p1**.

In questo modo si può anche assegnare un valore alla variabile **a** usando **p1**: basta usare una istruzione del tipo ****p** = ...

Per assegnare un valore alla variabile a posso usare il puntatore **p** o il puntatore **p1.**

Es:
assegno alla variabile a il valore 10

```
*p=10;
**p1=10;
```

prelevo il valore di a con i puntatori e lo metto in una variabile b

```
int b;
b = *p;
b = **p1;
```