Esempio puntatori

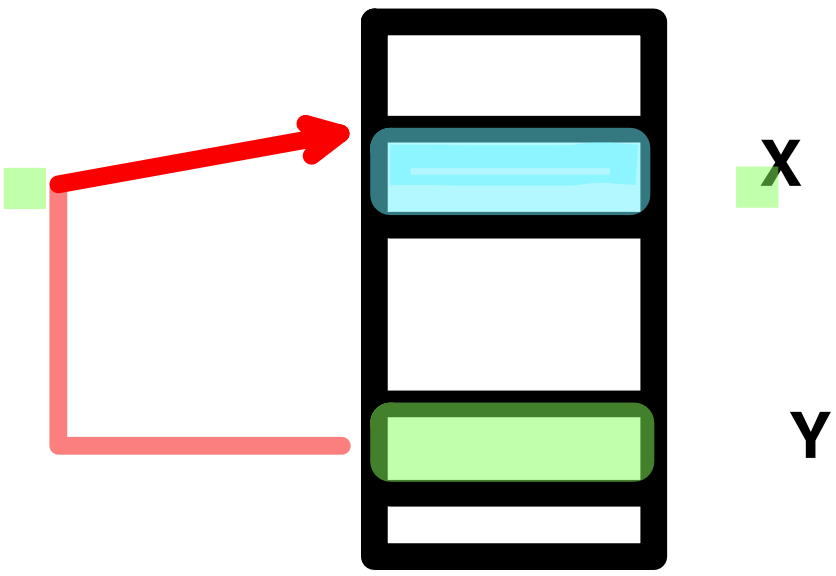
mercoledì 15 marzo 2023 21:54

```c
#include<stdio.h>

int x;
double *y;

void main(void){
    y = &x;
    printf("x address is %p - y is %p | x address+1 is %p - y+1 is %p\n",&x,y,&x+1,y+1);
}
```

```
x address is 0x10b58a000 - y is 0x10b58a000 | x address+1 is 0x10b58a004 - y+1 is 0x10b58a008
```



Un altro esempio di puntatori

```c
int *x;
double *y;

void function(void){
      x = y;
      if ( x+1 == y+1 ) return 1;
      return 0;
}
```

Predicato mai verificato

Int *x;

x+1

(char *) x + 1

Vado avanti di 1 byte per mezzo del Type Cast

Classico utilizzo dei puntatori

- per notificare ad una funzione dove “consegnare” o “leggere” in memoria le informazioni
  - ✓ ad esempio, `scanf()` consegna informazioni in memoria usando puntatori come parametri di input

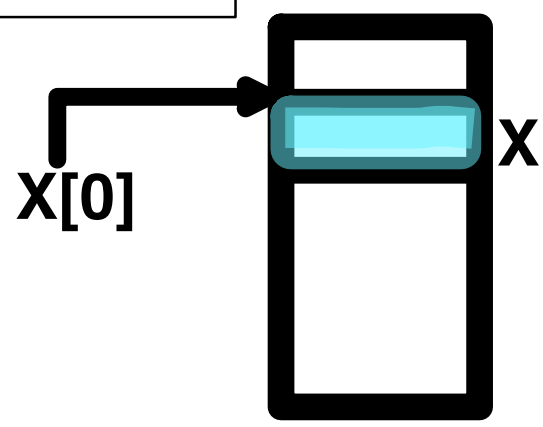
- data un’espressione “indirizzo” gli operatori principali per accedere all’effettivo contenuto di memoria sono 2:
  - * indirezione (prefisso all’espressione)
  - [] spiazzamento ed indirezione (suffisso all’espressione)

- per scandire la memoria e accedere/aggiornare i valori

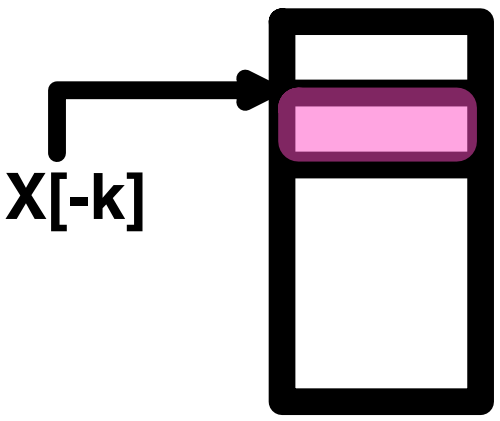
- per accedere ad informazioni nell’heap

Int *x;
x[0] //non mi spiazzo, o meglio, mi spiazzo di //zero, è come leggere *x.

(Mi spiazzo di 0 posizioni int, o in avanti o indietro, e poi accedo)



Int *x;
x[-K] //mi spiazzo di -K (mi sto spiazzando all’indietro.



Alcuni esempi

```
int *x;
int y[128];

 x = y;
 x = y+10;
*x = *(y+10);
*(y+10) = *x;
```

assegnazioni lecite

Y[10]

```
 y = x;
```

assegnazioni non lecite (`y` e’ un `rvalue`!!!)