

Puntatori in C

Il linguaggio **C** permette al programmatore di gestire la memoria virtuale dello spazio di indirizzamento del processo in esecuzione ... *fintantoché siano indirizzi di memoria validi.*

Per farlo la sintassi mette a disposizione degli operatori speciali (* e &) utili al fine di dichiarare, accedere al contenuto di una variabile puntatore od ottenere l'indirizzo cui risiede il contenuto di una variabile.

Un puntatore è quindi una variabile che non mantiene direttamente il valore del tipo per cui è dichiarata, ma l'indirizzo di memoria virtuale dove risiede il valore.

Puntatori in C

/ dichiarazione ed inizializzazione
al valore 12 di una variabile intera */*

```
int a = 12;
```

/ dichiarazione di una variabile
puntatore a intero ed inizializzazione
con l'indirizzo di memoria cui risiede
la variabile intera */*

```
int * a_ptr = &a;
```

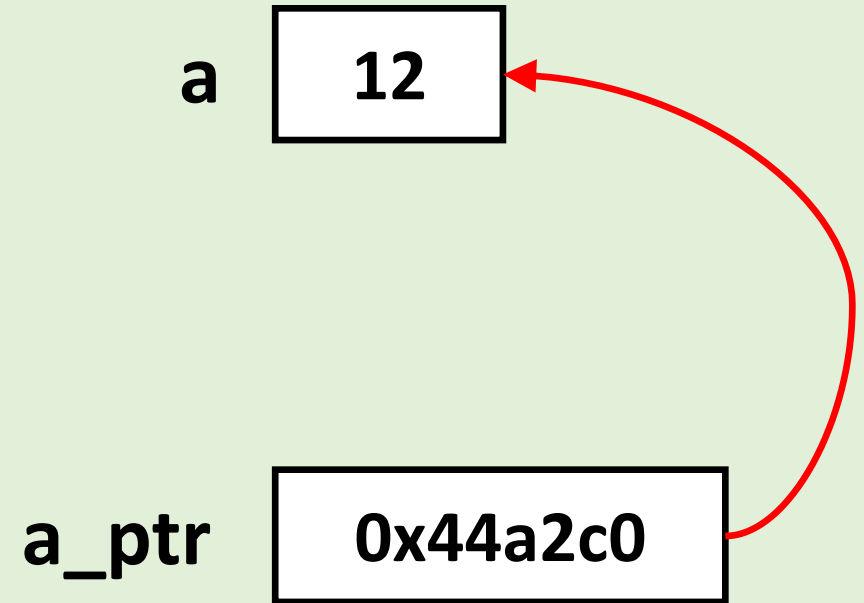
Puntatori in C

/ dichiarazione ed inizializzazione
al valore 12 di una variabile intera */*

```
int a = 12;
```

/ dichiarazione di una variabile
puntatore a intero ed inizializzazione
con l'indirizzo di memoria cui risiede
la variabile intera */*

```
int *a_ptr = &a;
```



Puntatori in C

/ dichiarazione ed inizializzazione
al valore 12 di una variabile intera */*

```
int a = 12;
```

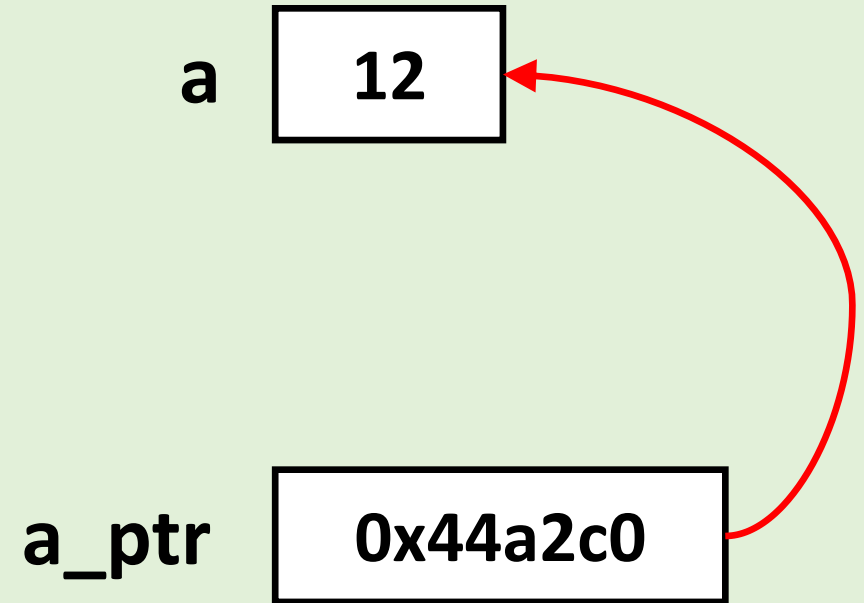
/ dichiarazione di una variabile
puntatore a intero ed inizializzazione
con l'indirizzo di memoria cui risiede
la variabile intera */*

```
int *a_ptr = &a;
```

```
printf("Il valore in a è: %d\n", a); // stampa 12
```

```
printf("L'indirizzo di a è: %p\n", a_ptr); // stampa 44a2c0
```

```
printf("Il valore puntato da a è: %d\n", *a_ptr); // stampa 12
```



Puntatori in C

*int ***a_pptr



0x44a2d0

*int **a_ptr



0x44a2c8

int a



12

0x44a2c0

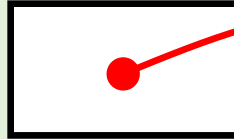
Puntatori in C

*int ***a_pptr



0x44a2d0

*int **a_ptr

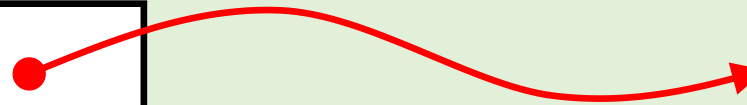


0x44a2c8

int a

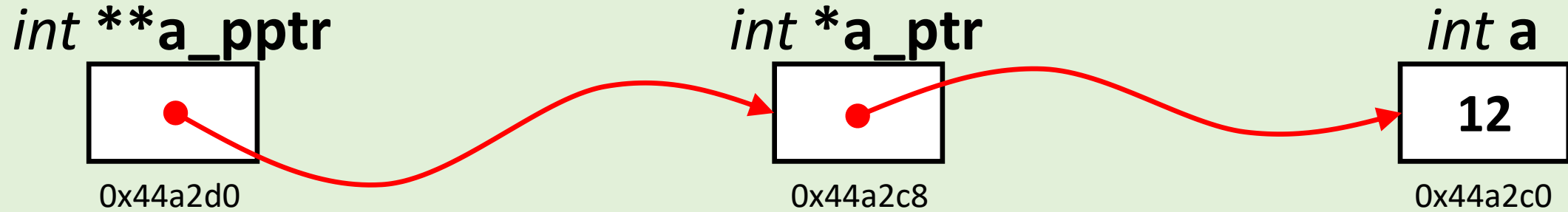


0x44a2c0



a_ptr = &a;

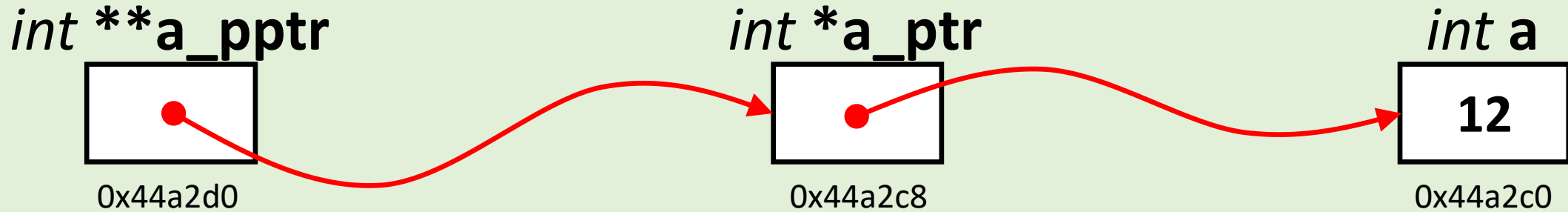
Puntatori in C



`a_pptr = &a_ptr;`

`a_ptr = &a;`

Puntatori in C

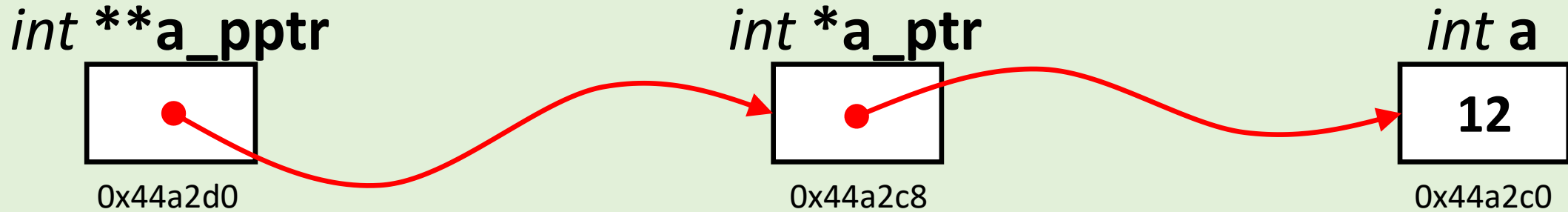


```
a_pptr = &a_ptr;
```

```
a_ptr = &a;
```

```
printf("%d\n", a);           // stampa 12  
printf("%d\n", *a_ptr);     // stampa 12  
printf("%d\n", **a_pptr);   // stampa 12
```


Puntatori in C

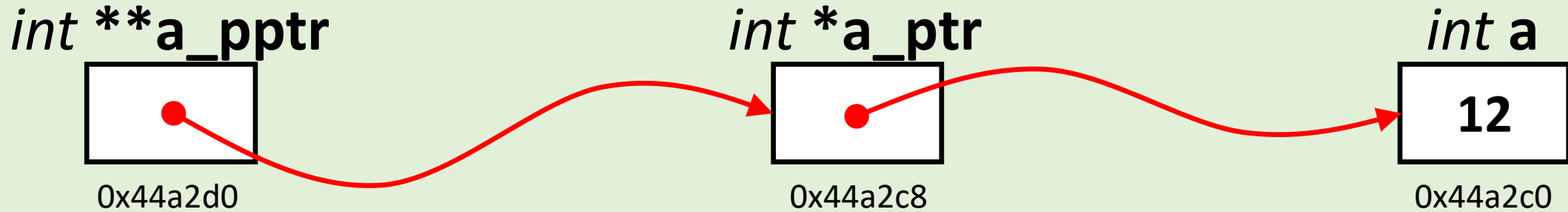


`a_pptr = &a_ptr;`

`a_ptr = &a;`

```
printf("%p\n", &a);           // stampa 0x44a2c0
printf("%p\n", a_ptr);        // stampa 0x44a2c0
printf("%p\n", *a_pptr);      // stampa 0x44a2c0
```

Puntatori in C



`a_pptr = &a_ptr;`

`a_ptr = &a;`

```
printf("%p\n", &(&a)); // errore!  
printf("%p\n", &a_ptr); // stampa 0x44a2c8  
printf("%p\n", a_pptr); // stampa 0x44a2c8
```