

9.2 Array e puntatori

Gli array e i puntatori in C sono strettamente correlati. Il nome di un array può essere usato come un puntatore al suo primo elemento. Considerando, per esempio:

```
char buf[100];
char *s;

s = &buf[0];
s = buf;
```

si ha che le due assegnazioni `s=&buf[0]` e `s=buf` sono perfettamente equivalenti. Infatti in C il nome di un array, come nel nostro caso `buf`, è una costante – si noti bene: una costante, non una variabile! – il cui valore è l'indirizzo del primo elemento dell'array. Allora, come gli elementi di un array vengono scanditi per mezzo dell'indice, equivalentemente si può avere accesso agli stessi elementi per mezzo di un puntatore. Per esempio consideriamo il seguente codice:

```
char buf[100];
char *s;
...
s = buf;
buf[7] = 'a';
printf("buf[7] = %c\n", buf[7]);

*(s + 7) = 'b';
printf("buf[7] = %c\n", buf[7]);
```

Si ha che `s` e `buf` sono due sinonimi, con la differenza che `s` è una variabile puntatore a carattere, mentre `buf` è una costante. Incrementato di 7 il valore di `s` si ottiene un valore corrispondente all'indirizzo dell'ottavo elemento, al quale si accede per mezzo dell'operatore di indirezione. Il valore del puntatore `s` è incrementato di 7 unità, cioè di 7 volte la dimensione dell'oggetto corrispondente al tipo base del puntatore (Figura 9.1).



Figura 9.1 Si può far riferimento a `buf[7]` scrivendo `*(s+7)`

Per le stesse ragioni avremmo potuto riscrivere il frammento di programma

```
char buf[2];

for (i = 0; i < 2; i++)
    buf[i] = 'K';
```

per l'inizializzazione degli elementi di un array come

```
char *s;
char buf[2];

s = buf;
for (i = 0; i < 2; i++)
    *s++ = 'K';
```

L'istruzione `*s++='K'` opera nel modo seguente:

- copia `K` nella locazione di memoria puntata da `s`
- poi incrementa `s` di un elemento.

✓ **NOTA**

Se per errore avessimo scritto:

```
char *s;  
char buf[2];
```

```
s = buf;  
for (i = 0; i < 100; i++) *s++ = 'K';
```

il compilatore non avrebbe segnalato alcun errore, ma avremmo avuto problemi in esecuzione perché `s` sarebbe inizializzata con `'K'` una regione di memoria al di là del limite allocato con `buf`. Che si usi una variabile puntatore o si faccia riferimento alla notazione con indice, è sempre dovere del programmatore assicurarsi che le dimensioni di un array vengano rispettate.