

13.2 Lettura e scrittura su file

Dopo aver imparato come si apre e si chiude un file, vediamo come utilizzare le funzioni che consentono di leggere il contenuto di un file e di modificarlo. Partiamo al solito con un esempio:

```
int elementi, dimensione;
char buf[100];
FILE *fp;
int n;
...
elementi = 100;
dimensione = 1;

n = fread(buf, dimensione, elementi, fp);
```

La funzione `fread` legge 100 caratteri dal file `fp` e li trasferisce nel vettore `buf`. I parametri della funzione `fread` sono:

<code>buf</code>	è il vettore dove devono essere trasferite le informazioni lette dal file
<code>dimensione</code>	rappresenta la dimensione in byte di un elemento del vettore
<code>elementi</code>	indica il numero di elementi del vettore
<code>fp</code>	è il file da leggere

Il valore di ritorno della `fread` indica al programmatore il numero di elementi letti dal file; tale numero può non coincidere con il numero di elementi del vettore `buf`, come nel caso in cui il file sia vuoto e contenga un numero di elementi inferiore a quello di `buf`. Se il valore di ritorno assume valore negativo significa che è stato commesso qualche errore, per esempio il file non è stato aperto. Poiché in generale un file può avere notevoli dimensioni, sarebbe impensabile poterlo leggere con una sola chiamata a `fread`: quest'ultima può essere ripetuta più di una volta, leggendo così a ogni chiamata soltanto una limitata porzione del file.

Le operazioni di lettura accedono al file in maniera sequenziale e mantengono traccia del punto in cui si è arrivati nella lettura. Dopo l'apertura in lettura il puntatore si trova posizionato all'inizio del file; a ogni chiamata a `fread` il

puntatore si sposta in avanti di un numero di byte pari a quelli che sono stati letti e trasferiti nella memoria centrale. Quando tutto il contenuto del file è stato letto la funzione `fread` ritorna il valore 0 per indicare che il puntatore è ormai posizionato a fine file; ogni ulteriore tentativo di lettura fallirà e `fread` continuerà a restituire il valore di ritorno 0.

Scriviamo ora un semplice programma che conta il numero di caratteri contenuti nel file di nome `clienti` (Listato 13.1).

```
/* Determina il numero di caratteri di un file esistente */

#include <stdio.h>

main()
{
    char buf[100];    /* Buffer per la lettura */
    FILE *fp;         /* File pointer */
    long nc;          /* Contatore caratteri */
    int n;             /* Numero caratteri letti con fread() */
    int fine_file = 0; /* Indica la fine della lettura del file */

    fp = fopen("clienti", "r"); /* Apertura del file clienti */

    if( fp == NULL )
        /* Si è verificato un errore: il file non esiste */
        printf("Errore : il file ordini non esiste\n");
    else {
        nc = 0L;                /* Inizializza il contatore */
        do {                    /* Ciclo di lettura */
            /* Legge 100 caratteri dal file ordini */
            n = fread(buf, 1, 100, fp);
            if(n==0)             /* Controllo di fine file */
                fine_file = 1;
            nc += n;             /* Incremento del contatore */
        }
        while(fine_file==0);

        fclose(fp);             /* Chiusura del file clienti */
        printf("Il file clienti contiene %ld caratteri\n", nc);
    }
}
```

Listato 13.1 Conta il numero di caratteri nel file `clienti`

Probabilmente i programmatori C avrebbero preferito scrivere il ciclo di lettura in questo modo:

```
for(;;) {
    n = fread(buf, 1, 100, fp);
    if(n==0) break;
    nc += n;
}
```

dove non viene utilizzata la variabile `fine_file`, ma l'uscita dal ciclo di lettura è effettuata per mezzo dell'istruzione `break`, che passa il controllo alla prima istruzione successiva al ciclo stesso.

La funzione per scrivere su di un file, `fwrite`, è analoga a `fread`:

```
int elementi, dimensione;
char buf[100];
FILE *fp;
int n;
...
elementi = 100;
```

```
dimensione = 1;
```

```
n = fwrite(buf, dimensione, elementi, fp);
```

`fwrite` scrive i 100 caratteri del vettore `buf` nel file `fp`. I parametri della funzione sono i seguenti:

<code>buf</code>	è il vettore che contiene i dati che devono essere memorizzati nel file <code>fp</code>
<code>dimensione</code>	rappresenta la dimensione in byte di un elemento del vettore
<code>elementi</code>	indica il numero di elementi del vettore
<code>fp</code>	è il file dove devono essere memorizzati i dati

Il valore di ritorno della `fwrite` indica al programmatore il numero di elementi che sono stati memorizzati nel file; tale numero può non coincidere con quello degli elementi del vettore `buf`, per esempio nel caso in cui il file abbia raggiunto la massima dimensione ammessa. Se il valore di ritorno assume valore negativo significa che è stato commesso qualche errore, per esempio il file non è stato aperto.

Scriviamo ora due semplici programmi: il primo acquisisce una stringa da tastiera e la memorizza all'interno del file di nome `fornitori` (Listato 13.2); il secondo copia il contenuto del file `ordini` nel file `ordini.bak` (Listato 13.3) ■

```
/* Scrittura di una stringa in un file */

#include <stdio.h>
#include <string.h>

main()
{
    char buf[100];          /* Buffer */
    FILE *fp;               /* File pointer */
    int len;

    /* Legge da tastiera il nome del fornitore */
    printf("Inserisci un fornitore : ");
    scanf("%s",buf);
    len = strlen(buf);
    fp = fopen("fornitori", "w"); /* Crea il file fornitori */

    /* Memorizza il nome del fornitore nel file */
    fwrite(buf, 1, len, fp);
    fclose(fp);             /* Chiude il file */
}
```

Listato 13.2 Programma per l'acquisizione di una stringa da tastiera e sua scrittura in un file

```
/* Copia di un file su un altro */

#include <stdio.h>

main()
{
    FILE *fpin, *fpout;     /* File pointer */
    char buf[512];          /* Buffer dati */
    int n;

    fpin = fopen("ordini","r"); /* Apre ordini in lettura */
    if(fpin!=NULL) {
        fpout = fopen("ordini.bak", "w"); /* Crea ordini.bak */
        if(fpout!=NULL) { /* ordini.bak creato correttamente? */
            for(;;) {
                /* Copia ordini in ordini.bak */
            }
        }
    }
}
```

```

        n = fread(buf, 1, 512, fpin);      /* Legge ordini */
        if( n == 0 ) break;                /* controllo di fine file */
        fwrite(buf, 1, n, fpout);          /* Scrive in ordini.bak */
    }
    fclose(fpin);                          /* Chiude il file ordini */
    fclose(fpout);                          /* Chiude il file ordini.bak */
}
else {
    printf("Il file ordini.bak non può essere aperto\n");
    fclose(fpin);                          /* Chiude il file ordini */
}
}
else
/* Errore di apertura */
    printf("Il file ordini non esiste\n");
}

```

Listato 13.3 Programma per la copia di un file su un altro