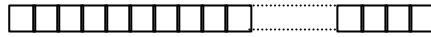


Gestione dei file

Un file e' definito come un oggetto, presente nel file system, costituito da **una sequenza continua di bytes**



Un file, per poter essere utilizzato, deve essere **aperto**
Successivamente e' possibile leggere il file e/o scrivere nuovi contenuti

Dopo l'apertura, il **puntatore di lettura/scrittura** e' posizionato all'inizio del file



Il puntatore di lettura/scrittura e' spostato in avanti a seguito di operazioni di lettura e scrittura. Lo spostamento e' funzione del **numero di byte** letti o scritti.

1

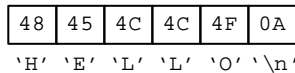
Corrado Santoro - Laboratorio di Informatica - Lezione 10 - CdS Ing. Informatica - Universita' di Catania

File di testo e binari

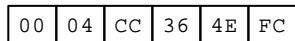
In funzione del tipo di contenuto di un file, distinguiamo:

- **file di testo** contengono caratteri stampabili e informazioni interpretabili da un umano
- **file binari** contengono dati di vario genere (mix di caratteri stampabili e non)

File di testo



File di binario



2

Corrado Santoro - Laboratorio di Informatica - Lezione 10 - CdS Ing. Informatica - Universita' di Catania

Uso dei file (apertura/chiusura)

Apertura:

```
FILE * fopen (char * pathname, char * mode)
```

pathname: percorso completo del file da aprire

mode: modalita' di apertura

r lettura

w creazione e scrittura

r+ lettura e scrittura (file gia' esistente)

w+ creazione, lettura e scrittura

a scrittura in "append" (alla fine del file)

b file binario (se non specificato e' un file di testo)

Chiusura:

```
int fclose (FILE * f)
```

3

Corrado Santoro - Laboratorio di Informatica - Lezione 10 - CdS Ing. Informatica - Universita' di Catania

Uso dei file di testo (lettura/scrittura)

Scrittura:

```
int fprintf (FILE *f, char * format, ...)
```

Lettura:

```
int fscanf (FILE *f, char * format, ...)
```

Stessa modalita' di funzionamento di printf e scanf

4

Corrado Santoro - Laboratorio di Informatica - Lezione 10 - CdS Ing. Informatica - Universita' di Catania

Uso dei file di testo, esempio

```
#include <stdio.h>
...
...
#define VECTOR_SIZE 10
int dati[VECTOR_SIZE];
...
...
    int i;
    FILE * f;
    f = fopen ("dati.txt","w"); // creazione e apertura in scrittura
    if (f == NULL)
        printf ("Errore di apertura\n");
    else {
        for (i = 0;i < VECTOR_SIZE;i++)
            fprintf (f, "%d\n", dati [i]); // scrittura dei dati
        fclose (f); // chiusura del file a conclusione
    }
...

```

5

Corrado Santoro – Laboratorio di Informatica – Lezione 10 – CdS Ing. Informatica – Università di Catania

Uso dei file di testo, esempio

```
#include <stdio.h>
...
...
#define VECTOR_SIZE 10
int dati[VECTOR_SIZE];
...
...
    int i;
    FILE * f;
    f = fopen ("dati.txt","r"); // apertura in lettura
    if (f == NULL)
        printf ("Errore di apertura\n");
    else {
        for (i = 0;i < VECTOR_SIZE;i++)
            fscanf (f, "%d", &dati [i]); // lettura dei dati
        fclose (f); // chiusura del file a conclusione
    }
...

```

6

Corrado Santoro – Laboratorio di Informatica – Lezione 10 – CdS Ing. Informatica – Università di Catania

File di testo

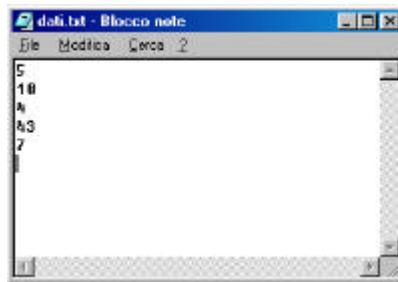
```
dati [0] = 5
dati [1] = 10
dati [2] = 4
dati [3] = 43
dati [4] = 7
```

File "dati.txt"

35	0A	31	30	0A	34	0A	34	33	0A	38	0A
----	----	----	----	----	----	----	----	----	----	----	----

'5' '\n' '1' '0' '\n' '4' '\n' '4' '3' '\n' '8' '\n'

File "dati.txt" aperto con il
"blocco note"



7

Corrado Santoro - Laboratorio di Informatica - Lezione 10 - CdS Ing. Informatica - Università di Catania

Uso dei file di binari (lettura/scrittura)

La scrittura/lettura avviene specificando il **puntatore** al blocco di dati da leggere/scrivere e la **dimensione** dei dati da leggere/scrivere

Scrittura:

```
int fwrite (void * ptr, int size, int num, FILE *f)
```

Lettura:

```
int fread (void * ptr, int size, int num, FILE *f)
```

ptr puntatore al blocco di dati da trasferire
size dimensione di ogni blocco dati
num numero di blocchi da trasferire
f file su cui leggere/scrivere (precedentemente aperto)

valore di ritorno: **numero dei blocchi trasferiti**

8

Corrado Santoro - Laboratorio di Informatica - Lezione 10 - CdS Ing. Informatica - Università di Catania

Uso dei file binari, esempio

```
#include <stdio.h>
...
...
#define VECTOR_SIZE 10
int dati[VECTOR_SIZE];
...
...
int i;
FILE * f;
f = fopen ("dati.bin","wb"); // creazione e apertura in scrittura
                             // file binario

if (f == NULL)
    printf ("Errore di apertura\n");
else {
    for (i = 0;i < VECTOR_SIZE;i++)
        fwrite (& dati[i], sizeof (int), 1, f); // scrittura dei dati
    fclose (f); // chiusura del file a conclusione
}
...
```

9

Corrado Santoro – Laboratorio di Informatica – Lezione 10 – CdS Ing. Informatica – Università di Catania

File binari - fwrite

```
fwrite (& dati[i], sizeof (int), 1, f);
```

Puntatore all'area di memoria
che contiene i dati da trasferire

Dimensione (in byte) dei dati da
trasferire
(*trasferiamo un elemento del
vettore, quindi un intero*)

Numero di elementi da trasferire
(*trasferiamo un solo elemento del vettore*)

File su cui scrivere

Scrivere l'intero vettore

```
fwrite (& dati[0], sizeof (int), VECTOR_SIZE, f);
```

10

Corrado Santoro – Laboratorio di Informatica – Lezione 10 – CdS Ing. Informatica – Università di Catania

Uso dei file binari, esempio

```
#include <stdio.h>
...
...
#define VECTOR_SIZE 10
int dati[VECTOR_SIZE];
...
...
int i;
FILE * f;
f = fopen ("dati.bin","rb"); // apertura in lettura
                             // file binario

if (f == NULL)
    printf ("Errore di apertura\n");
else {
    for (i = 0;i < VECTOR_SIZE;i++)
        fread (& dati[i], sizeof (int), 1, f); // lettura dei dati
    fclose (f); // chiusura del file a conclusione
}
...
```

11

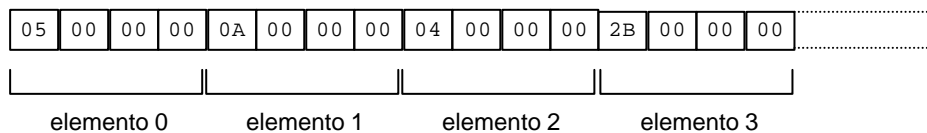
Corrado Santoro – Laboratorio di Informatica – Lezione 10 – CdS Ing. Informatica – Università di Catania

File binari

```
dati [0] = 5
dati [1] = 10
dati [2] = 4
dati [3] = 43
dati [4] = 7
```

Ogni elemento (numero intero) occupa 4 byte

File "dati.bin"



12

Corrado Santoro – Laboratorio di Informatica – Lezione 10 – CdS Ing. Informatica – Università di Catania

Gestione dell'end-of-file

```
for (i = 0; i < VECTOR_SIZE; i++)  
    fread (& dati[i], sizeof (int), 1, f);
```

```
for (i = 0; i < VECTOR_SIZE; i++)  
    fscanf (f, "%d", & dati[i]);
```

C'e' un loop che dice quanti elementi occorre leggere
Stiamo dunque supponendo che il numero di elementi del file sia noto a priori

Se questo numero non e' noto, occorre leggere il file fino alla sua fine
Occorre dunque sapere quanto abbiamo raggiunto la **fine del file**.

La funzione:

```
int feof (FILE * f)
```

restituisce "1" se abbiamo raggiunto la fine del file f, "0" altrimenti

13

Corrado Santoro - Laboratorio di Informatica - Lezione 10 - CdS Ing. Informatica - Universita' di Catania

File binari: gestione dell'eof

```
#include <stdio.h>  
...  
...  
#define VECTOR_SIZE 10  
int dati[VECTOR_SIZE];  
...  
int i;  
FILE * f;  
f = fopen ("dati.bin", "rb"); // apertura in lettura  
                             // file binario  
  
if (f == NULL)  
    printf ("Errore di apertura\n");  
else {  
    i = 0;  
    while (!feof (f)) {  
        fread (& dati[i], sizeof (int), 1, f); // lettura dei dati  
        i++;  
    }  
    fclose (f); // chiusura del file a conclusione  
}  
...
```

14

Corrado Santoro - Laboratorio di Informatica - Lezione 10 - CdS Ing. Informatica - Universita' di Catania

File binari: gestione dell'eof – Implementazione corretta

```
#include <stdio.h>
...
...
#define VECTOR_SIZE 10
int dati[VECTOR_SIZE];
...
int i;
FILE * f;
f = fopen ("dati.bin","rb"); // apertura in lettura
                               // file binario

if (f == NULL)
    printf ("Errore di apertura\n");
else {
    i = 0;
    while (fread (& dati[i], sizeof (int), 1, f) == 1) { // lettura e
                                                            // controllo

        i++;
    }
    fclose (f); // chiusura del file a conclusione
}
...
```

15

Corrado Santoro – Laboratorio di Informatica – Lezione 10 – CdS Ing. Informatica – Università di Catania

File di testo: gestione dell'eof – Implementazione corretta

```
#include <stdio.h>
...
...
#define VECTOR_SIZE 10
int dati[VECTOR_SIZE];
...
int i;
FILE * f;
f = fopen ("dati.bin","r"); // apertura in lettura

if (f == NULL)
    printf ("Errore di apertura\n");
else {
    i = 0;
    while (fscanf (f, "%d",& dati[i]) == 1) { // lettura e
                                                // controllo

        i++;
    }
    fclose (f); // chiusura del file a conclusione
}
...
```

16

Corrado Santoro – Laboratorio di Informatica – Lezione 10 – CdS Ing. Informatica – Università di Catania

Case-Study: una rubrica telefonica

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

#define FILENAME      "rubrica.bin"

typedef struct {
    char cognome [40];
    char nome [40];
    char telefono [30];
} t_elemento_rubrica;

...
```

17

Corrado Santoro – Laboratorio di Informatica – Lezione 10 – CdS Ing. Informatica – Università di Catania

Case-Study: una rubrica telefonica

```
...
int main (int argc, char * argv[])
{
    int choice;
    do {
        printf ("0- Fine\n");
        printf ("1- Inserisci nominativo\n");
        printf ("2- Cerca nominativo\n");
        scanf ("%d", &choice);
        switch (choice) {
            case 1: insert_name (); break;
            case 2: search_name (); break;
        }
    } while (choice != 0);
}
...
```

18

Corrado Santoro – Laboratorio di Informatica – Lezione 10 – CdS Ing. Informatica – Università di Catania

Case-Study: una rubrica telefonica

```
...
void insert_name (void)
{
    t_elemento_rubrica e;
    FILE * f;
    printf ("Cognome :");
    scanf ("%s", e.cognome);
    printf ("Nome      :");
    scanf ("%s", e.nome);
    printf ("Telefono:");
    scanf ("%s", e.telefono);
    f = fopen (FILENAME, "ab"); // append = apertura in scrittura alla
                                // fine del file

    if (f == NULL)
        printf ("Errore di apertura del file.\n");
    else {
        fwrite (&e, sizeof (t_elemento_rubrica), 1, f);
        fclose (f);
    }
}
...
```

19

Corrado Santoro – Laboratorio di Informatica – Lezione 10 – CdS Ing. Informatica – Università di Catania

Case-Study: una rubrica telefonica

```
...
void search_name (void)
{
    char cognome [40];
    t_elemento_rubrica e;
    FILE * f;
    printf ("Inserisci il cognome da cercare :");
    scanf ("%s", cognome);
    f = fopen (FILENAME, "rb");
    if (f == NULL)
        printf ("Errore di apertura del file.\n");
    else {
        while (fread (&e, sizeof (t_elemento_rubrica), 1, f) == 1) {
            if (strcmp (e.cognome, cognome) == 0) {
                printf ("Cognome   : %s\n", e.cognome);
                printf ("Nome     : %s\n", e.nome);
                printf ("Telefono : %s\n", e.telefono);
            }
        }
        fclose (f);
    }
}
```

20

Corrado Santoro – Laboratorio di Informatica – Lezione 10 – CdS Ing. Informatica – Università di Catania

ESERCIZIO SULLE CODE + FILES

Realizzare un sistema informatico che consenta di smistare l'accesso agli sportelli di una "segreteria studenti".

Si supponga la presenza di tre sportelli e si consideri la seguente operativita':

1. Lo studente inserisce i propri dati (cognome, nome e numero di matricola)
2. Il sistema sceglie lo sportello con minor gente in fila e accoda lo studente
3. Lo studente riceve dal sistema l'informazione sul **numero di sportello** sul quale egli e' stato accodato ed il **numero di persone** che si trovano in coda a quello sportello.
4. Ogni addetto allo sportello interroga il sistema per sapere il prossimo studente da servire e per conoscere il numero di studenti in fila davanti al suo sportello.
- 5. All'orario di chiusura, deve essere possibile registrare lo stato delle code in modo da ripristinare i turni nel giorno successivo**

21