

File di testo e file binari Directory

Dove salvare i file perché vengano ritrovati dal programma?

CASO 1

Programma eseguito da ambiente di lavoro (MS Visual C++).
La directory in cui viene creato/letto il file è la stessa del progetto.

Es: C:\temp\nomeprogetto

CASO 2

Programma eseguito da prompt.
La directory in cui viene creato/letto il file è la stessa dove c'è l'eseguibile.

Es: C:\temp\nomeprogetto\debug

File di testo e file binari Esempio

Un file binario (persone.dat), contiene una lista di persone. In particolare per ogni persona è specificato:

- Nome (non più di 20 caratteri)
- Cognome (non più di 40 caratteri)
- Codice fiscale (17 caratteri)

Si scriva un programma in C che, dopo aver definito una struttura *persona*:

- apre il file in lettura
- chiede all'utente una posizione nella lista
- visualizza i dati della persona che si trova nella posizione richiesta
- chiude il file

Usare la funzione `fseek()`. (La prima posizione è 0).

Esempio (uso di fseek)

```
#include <stdio.h>
typedef struct {
    char nome[20],
    char cognome[40],
    char cf[17],
} persona;

void main() {
    persona p;
    int pos;
    FILE *file;

    file = fopen("persone.dat", "rb");

    printf("Inserisci la posizione della persona nella lista: ");
    scanf("%d", &pos);

    fseek(file, pos*(sizeof(persona)), SEEK_SET);

    fread(&p, sizeof(persona), 1, file);
    printf("\nLetto:\t%s %s %s\n", p.nome, p.cognome, p.cf);
    fclose(file);
}
```

Crypt-decrypt su file

Scrivere un programma in C che richiede all'utente il nome di un file di testo, apre il file in lettura e scrive sul file *crypt.txt* il cifrato (vedi lezione 6). Quindi apre il file *crypt.txt* e lo decifra scrivendo il risultato nel file *decrypt.txt*.

Confrontare il file di ingresso con il file *decrypt.txt*.

Lettura e conversione file di testo

Un file di testo contiene i dati relativi a film (titolo, regista e anno), separati da spazi, e disposti su più righe (max 50).

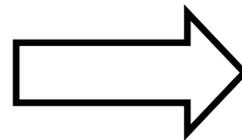
Scrivere un programma in C che legga il file di testo e importi i dati in un opportuno array di strutture.

Si riscriva poi un secondo file (con estensione .csv), in cui gli spazi tra i campi sono sostituiti da “,”.

(Verificare il risultato aprendo il file .csv in MS Excel)

FILM.TXT

```
Serafino Germi 1969  
Fantozzi Salce 1975  
Spartacus Kubrick 1960  
...
```



FILM.CSV

```
Serafino,Germi,1969  
Fantozzi,Salce,1975  
Spartacus,Kubrick,1960  
...
```

Rubrica su file

Sul file binario `rubrica.rub` (scaricabile dal sito) sono scritti una serie di contatti telefonici.

Ogni contatto è scritto secondo il seguente schema (fare uso delle typedef struct):

- cognome (stringa di massimo 30 caratteri)
- nome (stringa di massimo 30 caratteri)
- telefono (stringa di massimo 20 caratteri)
- età (intero)

File di testo e file binari

Esercizio 3 (2)

Rubrica su file

Si scriva un programma in linguaggio C che legge il file e

- visualizza il contenuto del file
- legge da tastiera il nome di una persona
- chiede all'utente il nuovo numero di telefono
- aggiorna il dato corrispondente
- visualizza di nuovo il contenuto del file

Es, se il file conteneva

Rossi	Piero	1234567890	23
Verde	Mario	0987654321	33
Marconi	Giacomo	111222333	12

E l'utente chiede di cambiare "Mario", che ha nuovo numero 00112233, dopo l'aggiornamento il file `rubrica.rub` dovrà contenere

Rossi	Piero	1234567890	23
Verde	Mario	00112233	33
Marconi	Giacomo	111222333	12

Monitoraggio temperatura

Questo esercizio è costituito da due parti principali: il monitor che genera dei dati relativi ad una temperatura, e l'allarme che cataloga i dati rilevando le situazioni di allerta secondo una certa soglia. Le due parti vengono eseguite sequenzialmente: prima il monitor e poi l'allarme.

In particolare:

Il monitor genera un file binario (uso di struct) che contiene dati di rilevamento temperatura (ora, temperatura). I dati sono generati in maniera casuale usando la funzione `rand()`.

Quindi avremo, ad esempio:

Dato 1: ora 00
 temperatura 27.3
Dato 2: ora 01
 temperatura 27.0
Dato 2: ora 02
 temperatura 26.5

ecc... ecc...

Vengono generati un totale di 24 dati (si suppone rilevamento quotidiano).

Una volta che il monitor ha generato tutti i dati, entra in funzione l'allarme. Quest'ultimo richiede all'utente una soglia di sicurezza (una temperatura) oltre la quale i rilevamenti vengono considerati come situazione di allarme. Quindi apre il file generato dal monitor, legge tutti i valori copiando su un file di testo tutti i dati (ora e temperatura) la cui temperatura supera la soglia. In seguito viene stampato il file.

Esempio sull'uso di rand().

```
#include <stdlib.h>
```

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

```
#include <time.h>
```

```
int main( void )
```

```
{
```

```
    int i;
```

```
    /* Display 10 numbers. */
```

```
    for( i = 0; i < 10; i++ )
```

```
        printf( " %6d\n", rand() );
```

```
}
```