

File binari

Introduzione

- Supponiamo di voler creare un file che memorizza un numero intero (ad esempio, il valore 999.000)
- Usando un file di testo posso scrivere:

```
...  
int num = 999000;  
FILE * file_txt;  
file_txt = fopen( "test.txt" , "w" );  
fprintf( file_txt, "%d", num );  
...
```

Introduzione

- **DOMANDA:** Quanti byte occupa il file che ho creato?
- **RISPOSTA:**

Il file avrà una dimensione di **6 byte**, perchè il numero 999000 è composto da 6 caratteri, e ogni carattere viene memorizzato su file tramite un byte contenente il codice ASCII corrispondente.

00 111 001	00 111 001	00 111 001	00 11 0000	00 11 0000	00 11 0000
-------------------	-------------------	-------------------	-------------------	-------------------	-------------------

carattere: '9'
ASCII: 57

carattere: '9'
ASCII: 57:

carattere: '9'
ASCII: 57

carattere: '0'
ASCII: 48

carattere: '0'
ASCII: 48

carattere: '0'
ASCII: 48

Introduzione

- **DOMANDA:** Possiamo usare meno di 6 byte per memorizzare lo stesso numero ?

- **RISPOSTA:**

Sì, usando un **file binario** posso usare solo 4 byte anzichè 6.

- **MOTIVO:**

Qualsiasi numero compreso tra -2^{31} e $(2^{31}-1)$ può essere rappresentato in complemento a due su 32 bit (4 byte).

Quindi posso scrivere su file i 4 byte corrispondenti alla rappresentazione binaria (in complemento a due) del numero!

file binario e file di testo a confronto

File di testo contenente la stringa "999000"

00 111 001	00 111 001	00 111 001	00 11 0000	00 11 0000	00 11 0000
-------------------	-------------------	-------------------	-------------------	-------------------	-------------------

carattere: '9'	carattere: '9'	carattere: '9'	carattere: '0'	carattere: '0'	carattere: '0'
ASCII: 57	ASCII: 57:	ASCII: 57	ASCII: 48	ASCII: 48	ASCII: 48

File binario contenente il numero 999000 memorizzato su 4 byte

0 1011 000	00 1111 10	0000 1111	00000000
-------------------	-------------------	------------------	----------

file binario e file di testo a confronto

File di testo contenente la stringa "999000"

00 111 001	00 111 001	00 111 001	00 11 0000	00 11 0000	00 11 0000
-------------------	-------------------	-------------------	-------------------	-------------------	-------------------

carattere: '9'	carattere: '9'	carattere: '9'	carattere: '0'	carattere: '0'	carattere: '0'
ASCII: 57	ASCII: 57:	ASCII: 57	ASCII: 48	ASCII: 48	ASCII: 48

File binario contenente il numero 999000 memorizzato su 4 byte

0 1011 000	00 1111 10	0000 1111	00000000
-------------------	-------------------	------------------	----------

Introduzione

Cosa vedo se apro il file test.bin in un editor di testo?

The screenshot shows a Notepad++ window titled "C:\Users\Me\Desktop\test.bin - Notepad++". The menu bar includes File, Modifica, Cerca, Visualizza, Formato, Linguaggio, Configurazione, Macro, Esegui, and Plugin. The toolbar contains various icons for file operations and editing. The text area shows the file "test.bin" with the text "X>SINUL" on line 1. The status bar at the bottom indicates "length: 4 Ln: 1 Col: 1 Sel: 4|0", "Dos\Windows", "ANSI", and "INS".

Arrows point from the characters in the text to their binary representations:

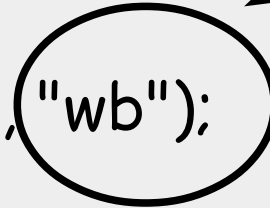
- X → 01011000
- > → 00111110
- S → 00001111
- I → 00000000

Esempio 1: scrittura su file binario

SCRITTURA DI UNA VARIABILE SINGOLA

```
#include <stdio.h>
int main() {
    int num = 999000;
    FILE * file_bin;
    file_bin = fopen("test.bin", "wb");
    if ( file_bin != NULL ) {
        fwrite( &num, sizeof(int), 1 , file_bin );
        fclose( file_bin );
    }
    else
        printf("Impossibile scrivere il file");
}
```

b = binary



La funzione fwrite

La funzione fwrite ricopia un'area di principale centrale su file

```
fwrite( indirizzo, dim_elemento, num_elementi, file_pointer );
```

*indirizzo del
primo byte
dell'area di
memoria che si
vuole scrivere
su file*

*dimensione (in
byte) degli
elementi
(blocchi) da
scrivere sul
file*

*Numero di
elementi
(blocchi) da
scrivere su
file*

*File su cui
effettuare
la scrittura*

Esempio 2: scrittura su file binario

SCRITTURA DI UN ARRAY CON UN'UNICA *fwrite*

```
#include <stdio.h>

int main() {
    int arr[ ] = { 1,2,3,4 } ;
    FILE * file_bin = fopen("array.bin", "wb");
    if ( file_bin != NULL ) {
        fwrite( arr, sizeof(int) , 4 , file_bin );
        fclose( file_bin );
    }
    else
        printf("Impossibile scrivere il file");
}
```

Esempio 3: scrittura su file binario

SCRITTURA DI UN RECORD (variabili di tipo "struct")

```
struct punto { float x; float y; };
int main() {
    struct punto p1 = { 3.5, 4.5 };
    FILE * file_bin = fopen("punti.bin", "wb");
    if ( file_bin != NULL ) {
        fwrite( &p1, sizeof(p1) , 1 , file_bin );
        fclose( file_bin );
    }
    else
        printf("Impossibile scrivere il file");
}
```

Esempio 4: Lettura di un file binario

```
#include <stdio.h>

int main() {
    int num;
    FILE * file_bin = fopen( "array.bin", "rb" );
    if ( file_bin != NULL ) {
        fread( &num, sizeof(int) , 1, file_bin );
        while( !feof(file_bin) ) {
            printf("ho letto: %d \n", num );
            fread( &num, sizeof(int) , 1, file_bin );
        }
        fclose( file_bin );
    }
    else
        printf("Impossibile leggere il file");
}
```