

Lezione 6

I file terminano attraverso l'utilizzo del carattere EOF (End Of File), (* il ritorno a capo viene fatto attraverso l'utilizzo di "\n") per quanto riguarda il file, esso non verrà "toccato" direttamente ma attraverso l'uso di un puntatore, inoltre prima di poter lavorare con un file è necessario, il sistema deve preparare il file per effettuare operazioni di I/O e per fare ciò si utilizzano determinate funzioni che si trovano nella libreria "stdio.h", se l'operazione di creazione/apertura del file il puntatore punta a NULL.

La funzione fopen consente di aprire un file, il primo argomento indica il nome del file da elaborare (path), il secondo indica quale operazione si intende compiere sul file¹, ritorna un puntatore ad un file e se l'operazione non va a buon fine il puntatore assume il valore NULL.

¹ Le opzioni di apertura di un file:

r → Lettura

w → Scrittura (In caso di file già esistente il tutto verrà sovrascritto)

a → Appendere in coda (In caso di file già esistente il tutto verrà inserito dopo il testo già scritto)

r+ → Creazione lettura/scrittura

w+ → Appende o crea un file

a+ → Lettura/Scrittura

Infine bisogna chiudere il file attraverso l'operazione fclose().

Per leggere un carattere alla volta nel file di testo si utilizza la funzione getc() (* Si può leggere tutto il file, bisogna semplicemente controllare fino a quando il carattere non è uguale a EOF *), mentre per scrivere un carattere per volta si utilizza la funzione putc().

Per inserire i dati in un file sequenziale possono essere utilizzate le funzioni fprintf() e fscanf(), per quanto riguarda la fscanf(), il primo parametro deve essere il puntatore del file.

* sizeof(tipo_variabale) si utilizza per conoscere la dimensione della variabile in memoria (in byte)*

In C è possibile creare dei file ad accesso casuale (viene utilizzato per le struct) attraverso le funzioni fwrite(), fseek(), fread().

fwrite(indirizzo_della_struct_da_passare, sizeof(variabale_struct), n° di elementi alla volta che vanno scritti, puntatore_del_file)

Fseek(puntatore_file, sizeof(struttura che si vuole intende scrivere), posizione_di_partenza).

La posizione di partenza può contenere i seguenti valori:

SEEK_SET → indica l'inizio del file

SEEK_CUR → indica la posizione corrente

SEEK_END → indica la fine del file

fread(indirizzo_della_struct_da_passare, sizeof(variabale_struct), elementi_da_leggere_dal_file, puntatore_del_file)

Il file sorgente servono a inserire l'implementazione delle funzione e invece i prototipi e l'implementazione delle funzioni vengono utilizzate nel header file, in questo modo è possibile utilizzare più volte le stesse funzioni in source file diversi, permette una maggiore modularità e astrazione, i file sorgente sono indicati da ".c" mentre i file header attraverso l'utilizzo di ".h"

L'header deve includere le direttive che definiscono le costanti, le eventuali definizioni di tipi, i prototipi delle funzioni.

Il bubble sort o ordinamento a bolle confronta i valori in coppia e il più pesante (nel caso di ordinamento crescente) va in fondo.

Esistono due tipi principali di ricerca, la ricerca sequenziale e la ricerca dicotomica (binaria).

La ricerca sequenziale effettua una ricerca dell'array in modo sequenziale e controlla in sequenza gli elementi dell'array e verifica ogni elemento controllato se è uguale all'elemento, l'algoritmo termina se viene trovato l'elemento cercato, oppure se si è controllato tutto l'array.

Nella ricerca dicotomica richiede come requisito che l'array sia ordinato, mediamente si utilizzano meno confronti. Nella ricerca dicotomica si "spezza l'array"