

Gestione dei File C

Generalità

- Il linguaggio C non contiene alcuna istruzione di Input/Output.
- Tali operazioni vengono svolte mediante chiamate a funzioni definite nella libreria standard contenute nel file *stdio.h*.
- Tali funzioni rendono possibile la lettura/scrittura in modo indipendente dalle caratteristiche proprie dei dispositivi di Input/Output.
- Le stesse funzioni possono essere utilizzate, ad esempio, sia per leggere un valore dalla tastiera sia per leggere un valore da un dispositivo di memoria di massa. Lo stesso vale per le funzioni di scrittura: le stesse operazioni possono essere utilizzate sia per la visualizzazione sullo schermo sia per scrivere un valore su un disco o una stampante.
- Ciò è possibile poiché il sistema di I/O C è caratterizzato da un'interfaccia indipendente dal dispositivo effettivo che si interpone tra il programmatore e il dispositivo. Tale interfaccia è chiamata flusso, mentre il dispositivo effettivo è chiamato file.

Flusso

Prof. G. Ascia

- Il sistema di I/O C associa ad ogni dispositivo fisico un dispositivo logico chiamato flusso.
- Poichè tutti i flussi si comportano alla stessa maniera, possono essere utilizzate le stesse funzioni per la loro gestione.
- Esistono due tipi di flussi: flussi binari e di testo.
 - Un flusso binario è formato da una sequenza di byte con una corrispondenza uno ad uno con i byte presenti sul dispositivo fisico.
 - Un flusso di testo è una sequenza di caratteri generalmente suddivisa in linee terminate da un carattere di *newline* (*\n*). In un flusso di testo alcuni caratteri potrebbero essere tradotti. Ad esempio il *newline* può essere convertito sulla stampante, che costituisce il dispositivo fisico, in un ritorno a capo e salto a una nuova linea della testina.

File

Prof. G. Ascia

- Un File è un qualsiasi dispositivo, da un disco a un terminale a una stampante.
- Per associare un flusso a un file è necessario un'operazione di apertura. Una volta aperto un file sarà possibile scambiare informazioni tra il file e il programma.
- Per eliminare l'associazione tra flusso e file è necessaria un'operazione di chiusura. Nel caso un file aperto in scrittura, l'eventuale contenuto del flusso viene scritto sul dispositivo fisico.

File

Prof. G. Ascia

- Ogni flusso ha associato una struttura chiamata FILE contenente i seguenti campi:
 - modalità di utilizzo del file (lettura, scrittura o lettura e scrittura);
 - posizione corrente su file (indicante il prossimo byte da leggere o scrivere su file);
 - un indicatore di errore di lettura/scrittura;
 - un indicatore di end-of-file, indicante il raggiungimento della fine del file.
- Ogni operazione di apertura a file restituisce un puntatore a una variabile di tipo FILE. Per potere leggere e scrivere i file è necessario usare delle variabili di tipo puntatore a file, dichiarate nel seguente modo:

```
FILE *pf;
```

Fondamenti di Informatica-Ingegneria Telematica

5

fopen

Prof. G. Ascia

- L'apertura di un file viene realizzata mediante la funzione fopen avente il seguente prototipo:

```
FILE *fopen(const char *nomefile, const char *modalità);
```
- dove *nomefile* è una stringa di caratteri indicante il nome del file da aprire e *modalità* indica il modo in cui il file deve essere aperto.
- Se si verifica un errore in apertura del file, la fopen() restituisce un puntatore nullo.
- Volendo, ad esempio, aprire in scrittura il file *test.txt* scriveremo:

```
FILE *pf;  
pf=fopen(``test.txt`,`w`);  
if(pf==NULL)  
    printf(``Impossibile aprire il file``);  
else
```
- In generale, prima di accedere ad un file occorre assicurarsi che la chiamata a fopen() sia stata eseguita con successo.

Fondamenti di Informatica-Ingegneria Telematica

6

fopen

Prof. G. Ascia

MODALITA'	SIGNIFICATO
r	Apri un file di testo in lettura
w	Crea un file di testo in scrittura
a	Apri un file di testo in modalità append
rb	Apri un file binario in lettura
wb	Crea un file binario in scrittura
ab	Apri un file binario in modalità append
r+	Apri un file di testo in lettura/scrittura
w+	Crea un file di testo in lettura/scrittura
a+	Crea o apri un file di testo in modalità append per lettura/scrittura
r+b	Apri un file binario in lettura/scrittura
w+b	Crea un file binario in lettura/scrittura
a+b	Crea o apri un file binario in modalità append per lettura/scrittura

Fondamenti di Informatica-Ingegneria Telematica

7

fclose

Prof. G. Ascia

- La chiusura di un file viene realizzata mediante la funzione `fclose` avente il seguente prototipo:

```
int fclose(FILE *pf);
```

dove `pf` è il puntatore restituito dalla `fopen()`.

Fondamenti di Informatica-Ingegneria Telematica

8

feof()

Prof. G. Ascia

- La funzione feof() restituisce un valore logico vero nel caso in cui è raggiunta la fine del file e zero in tutti gli altri casi.
- Il prototipo della feof è il seguente:

```
int feof(FILE *pf)
```

fscanf, fprintf

Prof. G. Ascia

- Le funzioni fscanf() e fprintf() vengono utilizzate per la lettura e la scrittura su file. Il loro comportamento è lo stesso delle funzioni scanf() e printf().
- Il loro prototipo è il seguente:

```
int fscanf(FILE *pf, const char *stringa_controllo, ..);
```

```
int fprintf(FILE * pf, const char *stringa_controllo,..);
```

Esempio: Lettura da file e visualizzazione sullo schermo

Prof. G. Ascia

```
#include <stdio.h>
void main(void)
{ FILE *pf;
  int a;
  pf=fopen("numeri.txt","r");
  if(pf)
  {
    while(!feof(pf))
    { fscanf(pf,"%d\t",&a);
      printf("%d\n",a);
    }
    fclose(pf);
  }
  else
  printf("errore");
}
```

Fondamenti di Informatica-Ingegneria Telematica

11

Esempio: Lettura da tastiera e scrittura su file

Prof. G. Ascia

```
#include <stdio.h>
void main(void)
{ int num,a;
  FILE *pf;
  printf("Quanti numeri vuoi inserire ?");
  scanf("%d",&num);
  pf=fopen("dati.txt","w");
  if(pf)
  { for(;num>0;num--)
    { printf("Inserisci un nuovo numero: ");
      scanf("%d",&a);
      fprintf(pf,"%d\t",a);
    }
    fclose(pf);
  }
  else printf("Errore");
}
```

Fondamenti di Informatica-Ingegneria Telematica

12

Scrittura e lettura delle informazioni (cognome, nome, matricola, media) relative a n studenti (1)

Prof. G. Ascia

```
#include <stdio.h>

void main(void)
{ FILE *pf;
  char cognome[20], nome[20];
  long matricola;
  double media;
  int i, nstud;

  printf("Quanti studenti ?\n");
  scanf("%d", &nstud);
  pf=fopen("studenti.txt", "w"); /* Apertura file*/
  if(pf==NULL) printf("Errore in scrittura\n");
  else /* file aperto con successo */
  { for(i=0; i<nstud; i++)
    { printf("Cognome:"); scanf("%s", cognome);
      fprintf(pf, "Cognome: %s\n", cognome);
      printf("Nome: "); scanf("%s", nome);
      fprintf(pf, "Nome: %s\n", nome);
      printf("Matricola: "); scanf("%ld", &matricola);
      fprintf(pf, "Matricola: %ld\n", matricola);
      printf("Media: "); scanf("%lf", &media);
      fprintf(pf, "Media: %lf\n", media);
    }
    fclose(pf);
  }
}
```

Fondamenti di Informatica-Ingegneria Telematica

13

Scrittura e lettura delle informazioni (cognome, nome, matricola, media) relative a n studenti (2)

Prof. G. Ascia

```
/* lettura da file */
pf=fopen("studenti.txt", "r");
if(pf==NULL)
  printf("Errore in lettura\n");
else
  { while (!feof(pf))
    { fscanf(pf, "Cognome: %s\n", cognome);
      printf("Cognome: %s\n", cognome);
      fscanf(pf, "Nome: %s\n", nome);
      printf("Nome: %s\n", nome);
      fscanf(pf, "Matricola: %ld\n", &matricola);
      printf("Matricola: %ld\n", matricola);
      fscanf(pf, "Media: %lf\n", &media);
      printf("Media: %lf\n", media);
    }
    fclose(pf);
  }
}
```

Fondamenti di Informatica-Ingegneria Telematica

14