




File

Francesco Isgrò

File

- Un file è una sequenza di byte, memorizzata su una unità di memoria di massa, al quale si accede tramite un nome.
- I file vengono gestiti dal sistema operativo, quindi il programma deve invocare il SO per accedere ai file.
- Principali operazioni sui file
 - Apertura: richiede al SO l'accesso ad un file
 - Lettura
 - Scrittura
 - Chiusura: comunica al SO che il programma rilascia il file

- 
- Per ogni programma vengono aperti automaticamente
 - stdin (sola lettura): lettura è da tastiera
 - stdout (sola scrittura): scrittura su video
 - stderr (sola scrittura): per messaggi d'errore su video

Apertura file


- Apertura è tramite la funzione `fopen()`
- Deve essere effettuata prima di poter operare su di un file
- Esempio


```
FILE *fp;
```

```
fp = fopen("pippo.dat", "r");
```

- `FILE` è una struttura definita in `stdio.h`
- Parametri di `fopen`
 - Nome del file
 - Modalità di apertura

`"r" "r+" "w" "w+" "a" "a+"`

- 
- “r” Open text file for reading. The stream is positioned at the beginning of the file
 - “w” Truncate file to zero length or create text file for writing. The stream is positioned at the beginning of the file
 - “a” Open for appending (writing at end of file). The file is created if it does not exist. The stream is positioned at the end of the file.

- 
- fopen restituisce un puntatore a FILE
 - La struttura mantiene le informazioni necessarie alla gestione del file
 - Il puntatore viene utilizzato per accedere al file
 - Se si verifica un errore fopen restituisce NULL

```
FILE *fp;  
if ((fp=fopen("pippo.dat","r"))==NULL) {  
    fprintf(stderr,"Errore!\n");  
    exit(-1);  
}
```




Chiusura

- Il file viene chiuso tramite `fclose()`
`fclose(fp);`
- Se si è verificato un errore `fclose()` ritorna EOF
- La chiusura va sempre effettuata appena le operazioni di lettura/scrittura sono terminate



Scrittura

- L'accesso ai file in lettura e scrittura avviene in maniera sequenziale
- La scrittura è possibile tramite la funzione `fprintf`
 - Scrive sul file `fp` a partire dalla posizione corrente
 - Il file deve essere stato correttamente aperto
 - In caso di errore restituisce EOF, altrimenti il numero di byte scritti



```
int ris1, ris2;
FILE *fp;

if ((fp=fopen("pippo.dat", "w"))!=NULL) {
    ris1 = ...;
    ris2= ...;

    fprintf(fp, "%d %d\n", ris1, ris2);
    fclose(fp);
}
```

Lettura

- La lettura avviene tramite la funzione `fscanf()`
- Il prototipo è

```
int fscanf(FILE *fp, char *formato, ....)
```

 - Legge dal file identificato da `fp` a partire dalla posizione corrente
 - Il file deve essere stato aperto in lettura o update
 - Restituisce il numero di conversioni effettuate con successo
 - Se vi è un errore (ad es., il file termina) restituisce EOF.


End of file


- La funzione
`int feof(FILE *fp);`
verifica la fine del file
- Restituisce 1
 - Quando si tenta di leggere oltre la fine del file
 - Per lo standard input, quando viene digitato
 - Ctrl-d (Linux)
 - Ctrl-z (Windows)

- Calcolare la somma degli interi in un file

```
int somma=0, n;
FILE *fp;


if
((fp=fopen("pippo.txt","r"))
!= NULL) {
    while
    (fscanf(fp,"%d",&n)==1) {
        somma += n;
    }
    fclose(fp);
}
```

- 
- Attenzione all'uso di feof e fscanf
 - Input file contiene la sola riga
 pippo pluto e paperino
 - Proviamo a leggerlo e a stamparne il contenuto
 - Prima usiamo una sola variabile char e poi 2 variabili char (il file è di 22 caratteri)



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

```
int main () {  
    FILE *fp;  
    char c;  
  
    fp = fopen("file.txt", "r");  
    if(fp == NULL) {  
        perror("Error in opening file");  
        return(-1);  
    }  
  
    while(!feof(fp)) {  
        fscanf(fp, "%c", &c);  
  
        printf("%c", c);  
    }  
    fclose(fp);  
  
    return(0);  
}
```




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

```
int main () {  
    FILE *fp;  
    char c;  
  
    fp = fopen("file.txt", "r");  
    if(fp == NULL) {  
        perror("Error in opening file");  
        return(-1);  
    }  
  
    while(!feof(fp)) {  
        fscanf(fp, "%c", &c);  
  
        printf("%c", c);  
    }  
    fclose(fp);  
  
    return(0);  
}
```


```
>programma  
pippo pluto e paperino
```

```
>
```



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


```
int main () {  
    FILE *fp;  
    char c;  
  
    fp = fopen("file.txt","r");  
    if(fp == NULL) {  
        perror("Error in opening file");  
        return(-1);  
    }  
  
    while(1) {  
        fscanf(fp,"%c",&c);  
        if( feof(fp) ) {  
            break ;  
        }  
        printf("%c", c);  
    }  
    fclose(fp);  
  
    return(0);  
}
```

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


```
int main () {  
    FILE *fp;  
    char c;  
  
    fp = fopen("file.txt","r");  
    if(fp == NULL) {  
        perror("Error in opening file");  
        return(-1);  
    }  
  
    while(1) {  
        fscanf(fp,"%c",&c);  
        if( feof(fp) ) {  
            break ;  
        }  
        printf("%c", c);  
    }  
    fclose(fp);  
  
    return(0);  
}
```

```
>programma  
pippo pluto e paperino  
>
```



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

```
int main () {  
    FILE *fp;  
    char c;  
  
    fp = fopen("file.txt","r");  
    if(fp == NULL) {  
        perror("Error in opening file");  
        return(-1);  
    }  
  
    while(fscanf(fp,"%c",&c)) {  
        printf("%c", c);  
    }  
    fclose(fp);  
  
    return(0);  
}
```




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

```
int main () {  
    FILE *fp;  
    char c;  
  
    fp = fopen("file.txt","r");  
    if(fp == NULL) {  
        perror("Error in opening file");  
        return(-1);  
    }  
  
    while(fscanf(fp,"%c",&c)) {  
        printf("%c", c);  
    }  
    fclose(fp);  
  
    return(0);  
}
```


>programma
pippo pluto e paperino

(loop infinito)



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

```
int main () {  
    FILE *fp;  
    char c;  
  
    fp = fopen("file.txt","r");  
    if(fp == NULL) {  
        perror("Error in opening file");  
        return(-1);  
    }  
  
    while(fscanf(fp,"%c",&c)==1) {  
        printf("%c", c);  
    }  
    fclose(fp);  
  
    return(0);  
}
```



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

```
int main () {  
    FILE *fp;  
    char c;  
  
    fp = fopen("file.txt", "r");  
    if(fp == NULL) {  
        perror("Error in opening file");  
        return(-1);  
    }  
  
    while(fscanf(fp, "%c", &c) == 1) {  
        printf("%c", c);  
    }  
    fclose(fp);  
  
    return(0);  
}
```

```
>programma  
pippo pluto e paperino  
>
```