I file

Laboratorio di Linguaggi di Programmazione
a.a. 2001/2002

dott.ssa Francesca A. Lisi
lisi@di.uniba.it

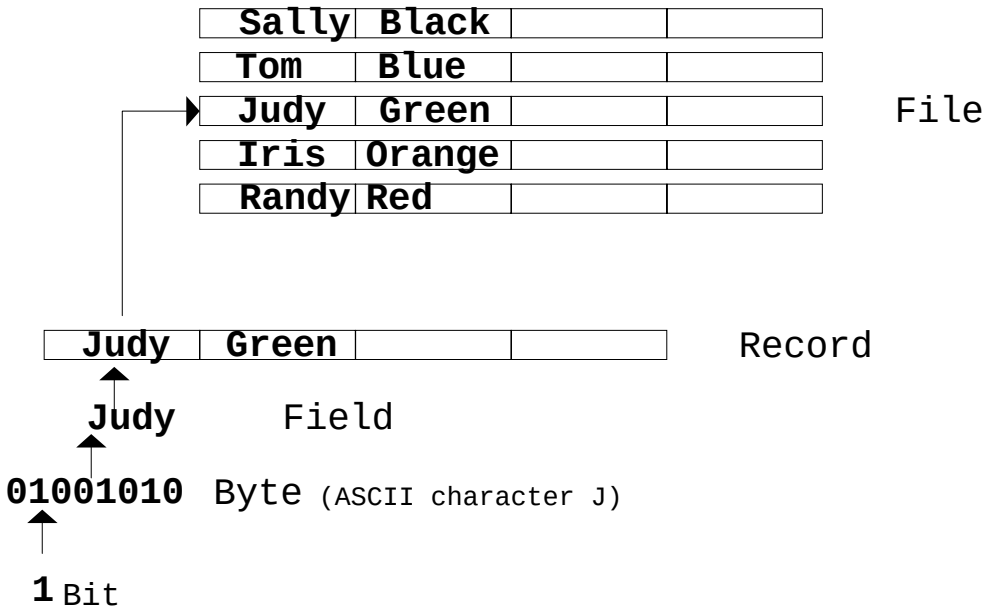
Sommario

- Generalità sui file in C
- I file ad accesso sequenziale
- I file di testo

Riferimenti

- cap. 11 di Deitel & Deitel
- par. 7.5 di Kernighan & Ritchie

La gerarchia dei dati



- La *chiave record* identifica un record per facilitarne il ritrovamento all'interno del file
- Nei *file sequenziali*, i record sono tipicamente ordinati per chiave

File e Stream

- Un *file* è una sequenza di byte terminante con un marcatore *end-of-file*
  - Nessuna struttura (assenza della nozione di *record*)
  - Il programmatore deve fornire una struttura al file
- Uno *stream* fornisce un canale di comunicazione fra file e programmi
  - Viene creato all'apertura di un file
  - L'apertura di un file restituisce un puntatore ad una struttura **`FILE`**
    - **`stdin`** - standard input (tastiera)
    - **`stdout`** - standard output (schermo)
    - **`stderr`** - standard error (schermo)

Lettura/scrittura su file

- **`fgetc`** `/* legge un carattere da file */`
  - Prende un puntatore a **`FILE`** come argomento
  - **`fgetc(stdin)`** equivalente a **`getchar()`**
- **`fputc`** `/* scrive un carattere su file */`
  - Prende un puntatore **`FILE`** ed un carattere da scrivere come argomento
  - **`fputc('a', stdout)`** equivalente a **`putchar('a')`**
- **`fgets`** `/* legge una riga da file */`
- **`fputs`** `/* scrive una riga su file */`
- **`fscanf / fprintf`**
  - Equivalenti di **`scanf`** e **`printf`** per le elaborazioni su file

Apertura/chiusura di un file ad accesso sequenziale

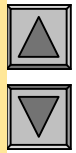
- **`FILE *myPtr;`** `/* Crea un puntatore` **`FILE*/`**
- **`myPtr=fopen("myFile.dat", openmode);`**
  - La funzione **`fopen`** prende il nome del file e la modalità di apertura come argomenti e restituisce un puntatore **`FILE`** al file specificato se l'apertura ha avuto successo, **`NULL`** altrimenti
- **`feof(FILE pointer)`**
  - Restituisce **`true`** se l'indicatore *end-of-file* (niente più dati da processare) è impostato per il file specificato
- **`fclose(FILE pointer)`** `/* chiude il file */`
  - Eseguito automaticamente quando il programma termina
  - Buona pratica chiudere i file esplicitamente

Apertura/chiusura di un file ad accesso sequenziale (II)

- I programmi possono elaborare uno o più file
- Ogni file deve avere un nome unico ma può avere molteplici puntatori
- Tutta l'elaborazione di file deve riferirsi al file utilizzando uno dei suoi puntatori

Mode	Description
`r`	Open a file for reading.
`w`	Create a file for writing. If the file already exists, discard the current contents.
`a`	Append; open or create a file for writing at end of file.
`r+`	Open a file for update (reading and writing).
`w+`	Create a file for update. If the file already exists, discard the current contents.
`a+`	Append; open or create a file for update; writing is done at the end of the file.

```c
/* Fig. 11.3: fig11_03.c
   Create a sequential file */
#include <stdio.h>

int main()
{
   int account;
   char name[ 30 ];
   double balance;
   FILE *cfPtr;   /* cfPtr = clients.dat file pointer */

   if ( ( cfPtr = fopen( "clients.dat", "w" ) ) == NULL )
      printf( "File could not be opened\n" );
   else {
      printf( "Enter the account, name, and balance.\n" );
      printf( "Enter EOF to end input.\n" );
      printf( "? " );
      scanf( "%d%s%lf", &account, name, &balance );

      while ( !feof( stdin ) ) {
         fprintf( cfPtr, "%d %s %.2f\n",
                 account, name, balance );
         printf( "? " );
         scanf( "%d%s%lf", &account, name, &balance );
      }

      fclose( cfPtr );
   }

   return 0;
}
```



Outline

1. Initialize variables and `FILE` pointer

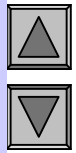
1.1 Link the pointer to a file

2. Input data

2.1 Write to file (`fprintf`)

3. Close file

```
Enter the account, name, and balance.
Enter EOF to end input.
? 100 Jones 24.98
? 200 Doe 345.67
? 300 White 0.00
? 400 Stone -42.16
? 500 Rich 224.62
?
```



Outline

Program Output

Leggere dati da un file ad accesso sequenziale

- Aprire il file in lettura

```
myPtr = fopen("myFile.dat", "r" );
```

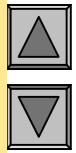
- Usare **`fscanf`** per leggere dal file

```
fscanf(myPtr, "%d%s%f", &myInt, &myString,
  &myFloat );
```

  - I dati vengono letti dall'inizio alla fine del file
  - Il puntatore di posizione nel file (*byte offset*) indica il numero del prossimo byte da leggere/scrivere
- **`rewind(myPtr)`**
  - Riposiziona il puntatore all'inizio del file (byte 0)

```c
/* Fig. 11.7: fig11_07.c
   Reading and printing a sequential file */
#include <stdio.h>

int main()
{
   int account;
   char name[ 30 ];
   double balance;
   FILE *cfPtr;   /* cfPtr = clients.dat file pointer */

   if ( ( cfPtr = fopen( "clients.dat", "r" ) ) == NULL )
      printf( "File could not be opened\n" );
   else {
      printf( "%-10s%-13s%s\n", "Account", "Name", "Balance" );
      fscanf( cfPtr, "%d%s%lf", &account, name, &balance );

      while ( !feof( cfPtr ) ) {
         printf( "%-10d%-13s%7.2f\n", account, name, balance );
         fscanf( cfPtr, "%d%s%lf", &account, name, &balance );
      }

      fclose( cfPtr );
   }

   return 0;
}
```

```
Account   Name         Balance
100       Jones          24.98
200       Doe           345.67
300       White           0.00
400       Stone         -42.16
500       Rich          224.62
```



Outline

1. Initialize variables

1.1 Link pointer to file

2. Read data (`fscanf`)

2.1 Print

3. Close file

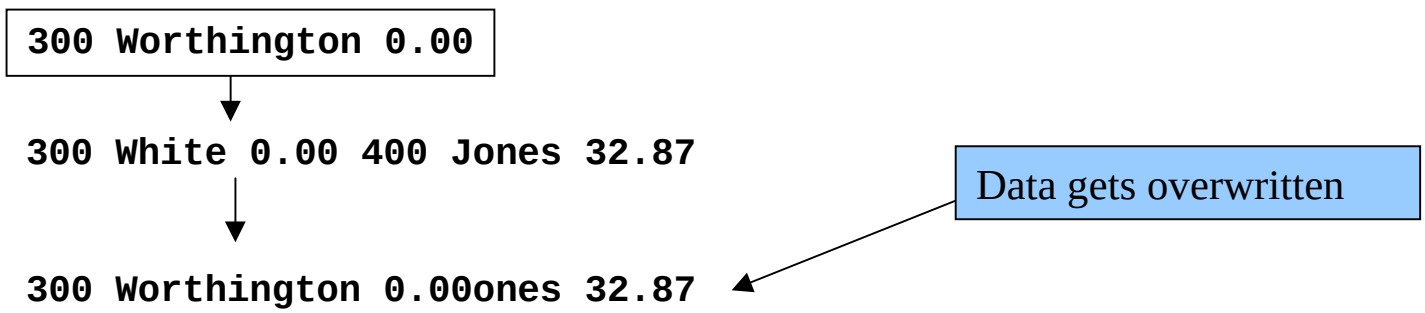
Program Output

Aggiornare i dati in un file ad accesso sequenziale

- I file ad accesso sequenziale non possono essere modificati senza correre il rischio di essere danneggiati
- La rappresentazione nei file e sullo schermo differisce da quella interna (es. **`1`**, **`34`**, **`-890`** sono tutti di tipo **`int`**, ma hanno differenti grandezze su disco)

```
300 White 0.00 400 Jones 32.87   (old data in file)
```

If we want to change White's name to Worthington,



Lettura da file di testo

```
void LeggiTesto(char *nomefile, TipoTesto *t)
{ FILE *fi;

  fi = fopen(nomefile, "r");
  if (fi == NULL) {
    fprintf(stderr, "Errore nell'apertura in
  lettura di %s", nomefile);
    exit 1;}
  while (LeggiParola(fi,word,MAXWORD)!=0)
    t= InserisciCodaLista (&t, word);
  fclose(fi);}
```

Scrittura su file di testo

```
void ScriviTesto(char *nomefile, TipoTesto t)
{  FILE *fi;

fi = fopen(nomefile, "w");
if (fi == NULL) {
  fprintf(stderr, "Errore nell'apertura in
scrittura di %s", nomefile);
  exit 1;}
while (t != NULL) {
  ScriviParola(fi, t->inizio);
  t = t->next_word;}
fclose(fi);}
```

Esercizi

- Definire la funzione `LeggiParola` per la lettura di una parola da file di testo
- Definire la funzione `ScriviParola` per la scrittura di una parola su file di testo
- Scrivere un programma C che copia un testo da file ad un altro parola per parola
- Assunzioni:
  - Una parola è una sequenza di caratteri alfanumerici compresa fra spazi o segni di punteggiatura
  - Il testo letto e/o scritto è memorizzato in una lista concatenata di stringhe