



POLITECNICO  
DI MILANO

# INFORMATICA

Lettura e scrittura su e  
da file

## Lettura e scrittura di caratteri su file

Lettura e scrittura di caratteri su file

scrive un carattere sul file aperto

legge un carattere dal file aperto

leggere un carattere da tastiera

scrivere un carattere a video

*f*

leggere un carattere da file

*f*

scrivere un carattere su un file

*f*

lettura di un carattere da un file

*aperto*

```
FILE *fp;
```

```
char c;
```

```
c = fgetc(fp);
```

```
/* c == EOF se fine file raggiunta */
```

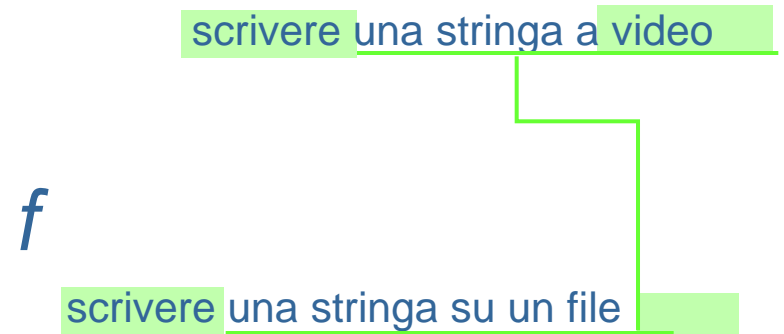
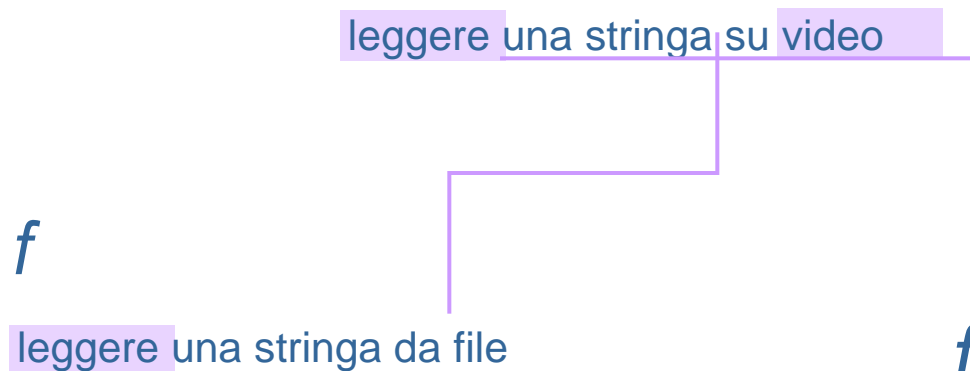
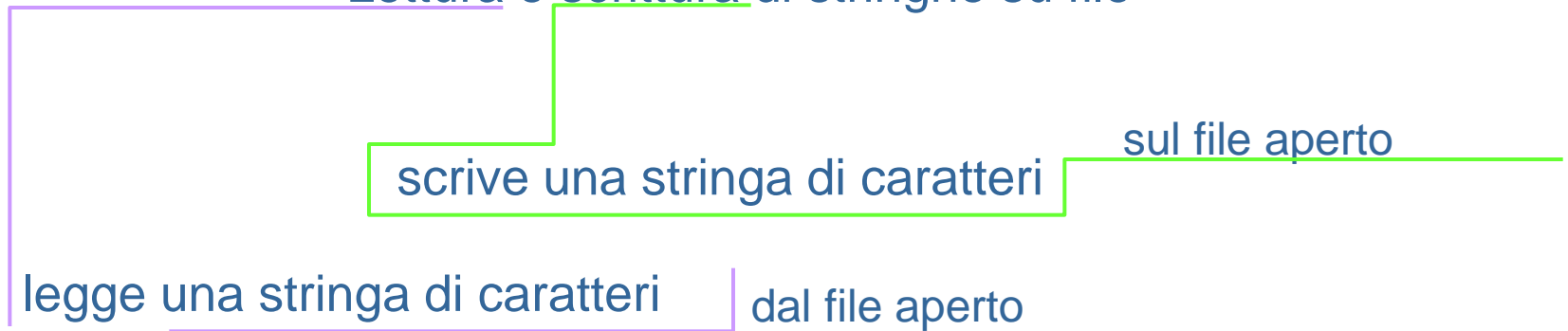
*f*

scrittura di un carattere su un file *aperto*

```
FILE *fp;  
int result;  
char c;  
  
result = fputc(fp, c);  
/* result != 0 se errore di scrittura */
```

## Lettura e scrittura di stringhe su file

Letture e scrittura di stringhe su file



*f*

*filegetstring*

lettura di una stringa da un file

aperto

di *n* caratteri

```
FILE *fp;
char *s;
int n;

if (fgets(s, n, fp) == NULL)
    {errore di lettura};
```

dove vengono memorizzati i caratteri letti

numero di byte letti

handle file

la stringa in memoria è terminata da

`\0`

meno di *n* caratteri se incontra un

`\n`

eof i caratteri letti sono assegnati a *s* che termina con

`\0`



*f*

scrittura di una stringa su un file *precedentemente aperto*

```
FILE *fp;  
char *s;  
int result;
```

```
result = fputs(s, fp);  
/* result = 0 se errore in scrittura */
```

puntatore e stringa

handle file

scrive tutti i caratteri fino a raggiungere

`\0`

## Lettura e scrittura di caratteri *struct*

## Letture e scrittura di caratteri *struct*

l'operazione prevede l'accesso a file per blocchi

lettura di un blocco di dati da un file

*precedentemente aperto*

*f*

scrittura di un blocco di dati su un file

*precedentemente aperto*

*f*

Lettura e scrittura di caratteri *struct*

classe C++

```
typedef struct {
```

```
    char cognome[20];
```

```
    char nome[20];
```

```
    int matricola;
```

```
} studente;
```

attributi

*possono essere letti senza utilizzare*

metodi specifici

*f*

lettura di un blocco di dati da un file *aperto*

```
FILE *fp;
studente stud;
int result, dimelem, numelem;

result = fread(&stud, dimelem, numelem, fp);
```

dimensione dell'elemento

*sizeof(stud)*

numero di elementi da leggere

Indirizzo di studente

esito dell'operazione

se **!=0** indica un errore nell'operazione

*f*

scrittura di un blocco di dati da un file *aperto*

```
FILE *fp;  
studente stud;  
int result, dimelem, numelem;
```

```
result = fwrite(&stud, dimelem, numelem, fp);
```

dimensione dell'elemento

*sizeof(stud)*

numero di elementi da scrivere

Indirizzo di studente

esito dell'operazione

se **!=0** indica un errore nell'operazione