



## Laboratorio 2 A

**ESERCITAZIONE:** *Gestione file e directory in C (libreria), file binari e posizionamento su file.*

Prof. Patrizio Dazzi • a.y. 2025/26

<https://pages.di.unipi.it/dazzi/>

[patrizio.dazzi@unipi.it](mailto:patrizio.dazzi@unipi.it)

Copia File



# Obiettivi dell'esercizio

**Copia** – Scrivere `mycp <src> <dst>` che copia un file in un altro:

- Versione A: un byte alla volta con `fgetc/fputc`.
- Versione B: a blocchi da `BLOCK_SIZE` con `fread/fwrite`.

# Possibile Soluzione per A

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(int argc, char *argv[]) {
5     if (argc != 3) {
6         printf("Uso:%s <src> <dst>", argv[0]);
7         return 1;
8     }
9
10    FILE *src = fopen(argv[1], "rb");
11    if (!src) {
12        perror("Errore apertura sorg.");
13        return 1;
14    }
15
16    FILE *dst = fopen(argv[2], "wb");
17    if (!dst) {
18        perror("Errore apertura dest.");
19        fclose(src);
20        return 1;
21    }
```

```
1     int ch;
2     while ((ch = fgetc(src)) != EOF) {
3         if (fputc(ch, dst) == EOF) {
4             perror("Errore scrittura");
5             fclose(src);
6             fclose(dst);
7             return 1;
8         }
9     }
10
11    fclose(src);
12    fclose(dst);
13    return 0;
14 }
```

# Possibile Soluzione per B

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #define B_SIZE 4096
4
5 int main(int argc, char *argv[]) {
6     if (argc != 3) {
7         printf("Uso:%s <src> <dst>", argv[0]);
8         return 1;
9     }
10    FILE *src = fopen(argv[1], "rb");
11    if (!src) {
12        perror("Errore apertura sorg.");
13        return 1;
14    }
15    FILE *dst = fopen(argv[2], "wb");
16    if (!dst) {
17        perror("Errore apertura dest.");
18        fclose(src);
19        return 1;
20    }
21
```

```
1     char buf[B_SIZE];
2     size_t bytesRead;
3
4     while ((bytesRead =
5         fread(buf, 1, B_SIZE, src))>0) {
6         if (fwrite(buf, 1, bytesRead, dst)
7             != bytesRead) {
8
9             perror("Errore scrittura");
10            fclose(src);
11            fclose(dst);
12            return 1;
13        }
14    }
15
16    fclose(src);
17    fclose(dst);
18    return 0;
19 }
20
```

Conta parole



# Obiettivi dell'esercizio

**Conta** – Implementare `wcline <nomefile>`: conta righe, parole, byte.

- Parola: massima sequenza di non-spazi
- Output: linee, parole, bytes.

# Possibile Soluzione per Conta parole

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <ctype.h>
4
5 int main(int argc, char *argv[]) {
6     if (argc != 2) {
7         fprintf(stderr, "Uso: %s <nomefile>\n",
8             argv[0]);
9         return 1;
10    }
11    FILE *fp = fopen(argv[1], "r");
12    if (!fp) {
13        perror("Errore apertura file");
14        return 1;
15    }
16
17    long bytes = 0;
18    long lines = 0;
19    long words = 0;
20    int in_word = 0;
21    int ch;
```

```
1     while ((ch = fgetc(fp)) != EOF) {
2         bytes++;
3         if (ch == '\n') {
4             lines++;
5         }
6         if (isspace(ch)) {
7             in_word = 0;
8         } else {
9             if (!in_word) {
10                words++;
11                in_word = 1;
12            }
13        }
14    }
15    fclose(fp);
16    printf("Linee: %ld\n", lines);
17    printf("Parole: %ld\n", words);
18    printf("Byte: %ld\n", bytes);
19    return 0;
20 }
21
```



Copia con filtro



# Obiettivi dell'esercizio

**Copia con filtro** – Realizzare `mycp <src> <dst>` che copia le righe non vuote di un file in un altro:

- Riga vuota: Contiene solo `\n`

# Possibile Soluzione per copia con filtro

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #define MAX_LINE 1024
5
6 int main(int argc, char *argv[]) {
7     if (argc != 3) {
8         printf("Uso:%s <src> <dst>", argv[0]);
9         return 1;
10    }
11    FILE *src = fopen(argv[1], "r");
12    if (!src) {
13        perror("Errore apertura sorgente");
14        return 1;
15    }
16    FILE *dst = fopen(argv[2], "w");
17    if (!dst) {
18        perror("Errore apertura dest.");
19        fclose(src);
20        return 1;
21    }
```

```
1     char line[MAX_LINE];
2
3     while (fgets(line, sizeof(line), src)) {
4         // Controlla se la riga è solo "\n"
5         if (strcmp(line, "\n") != 0) {
6             if (fputs(line, dst) == EOF) {
7                 perror("Errore scrittura");
8                 fclose(src);
9                 fclose(dst);
10                return 1;
11            }
12        }
13    }
14
15    fclose(src);
16    fclose(dst);
17    return 0;
18 }
19
```

Somma per utente



# Obiettivi dell'esercizio

**Somma** – Da un CSV semplice, di sole due colonne user e amount (un record per riga, niente virgole nei valori), sommare gli amount per utente e produrre un report ordinato alfabeticamente. Lettura con fgets, scrittura con fprintf.

Input:

```
1 marco, 50
2 pino, 20
3 gino, 30
4 pino, 10
5 marco, 30
```

Output:

```
1 gino, 30
2 marco, 80
3 pino, 30
```

# Q & A

*Thank you for your attention!*  
*Any Questions?*

patrizio.dazzi@unipi.it