

Soluzioni Esercizi 1 e 2 sui Segnali in C

November 17, 2025

Contents

Esercizio 1: Gestore di SIGINT (CTRL+C)	2
Esercizio 2: Segnali Utente (SIGUSR1 e kill)	2

1. Esercizio 1: Gestore di SIGINT (CTRL+C) | Contatore Statico

Implementazione di un handler per SIGINT che ritarda la terminazione fino alla terza ricezione del segnale, utilizzando un **contatore statico** per mantenere lo stato.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <signal.h>
4  #include <unistd.h>
5
6  //Contatore statico per memorizzare lo stato tra le chiamate
7  static int sigint_count = 0;
8
9  void sigint_handler(int signum) {
10     if (signum == SIGINT) {
11         printf("\nNon mi puoi terminare subito! Ho bisogno di pulire.\n");
12         sigint_count++;
13
14         if (sigint_count >= 3) {
15             printf("-----\n");
16             printf("SIGINT ricevuto 3 volte. Esecuzione pulizia e uscita.\n");
17             printf("-----\n");
18             exit(0); // Terminazione pulita
19         }
20     }
21 }
22
23 int main() {
24     //Configura l'handler per SIGINT
25     if (signal(SIGINT, sigint_handler) == SIG_ERR) {
26         perror("Impossibile impostare il gestore per SIGINT");
27         return 1;
28     }
29
30     printf("Programma in esecuzione. Premi CTRL+C 3 volte per uscire.\n");
31
32     //Ciclo di attesa di segnali
33     while (1) {
34         pause();
35     }
36     return 0;
37 }
```

Listing 1: Gestore di SIGINT con contatore statico

2. Esercizio 2: Segnali Utente (SIGUSR1 e kill) | Comunicazione Inter-Processo

Programma che stampa il proprio **PID** e attende il segnale SIGUSR1 da un processo esterno, utilizzando `pause()` per bloccare l'esecuzione fino alla ricezione di un segnale.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <signal.h>
4  #include <unistd.h>
5
6  void sigusr1_handler(int signum) {
7     if (signum == SIGUSR1) {
8         printf("\nSegnale Utente 1 ricevuto!\n");
9     }
10 }
11
12 int main() {
13     pid_t my_pid = getpid();
14     printf("Il PID di questo processo    : %d\n", my_pid);
15 }
```

```

16 //Imposta l'handler per SIGUSR1
17 if (signal(SIGUSR1, sigusr1_handler) == SIG_ERR) {
18     perror("Impossibile impostare il gestore per SIGUSR1");
19     return 1;
20 }
21
22 printf("In attesa del segnale SIGUSR1...\n");
23 printf("Istruzione: Apri un altro terminale e usa: kill -SIGUSR1 %d\n", my_pid);
24
25 //Ciclo di attesa
26 while (1) {
27     pause(); // Mette il processo in stato di attesa
28 }
29 return 0;
30 }

```

Listing 2: Gestione del segnale SIGUSR1 tramite kill