ESERCIZIO 3 (Testo)

Implementare un programma in cui un processo padre genera N processi figli.

Il processo padre assegna ad ogni figlio un indirizzo (intero > 1) e a se stesso 1.

Ciclicamente (per X volte) il processo padre interroga ogni figlio richiedendo un valore indicante il numero di richieste e il figlio risponde al padre che lo stampa a video.

Dopo X richieste il padre invia un comando di terminazione ai figli e ne attende la terminazione.

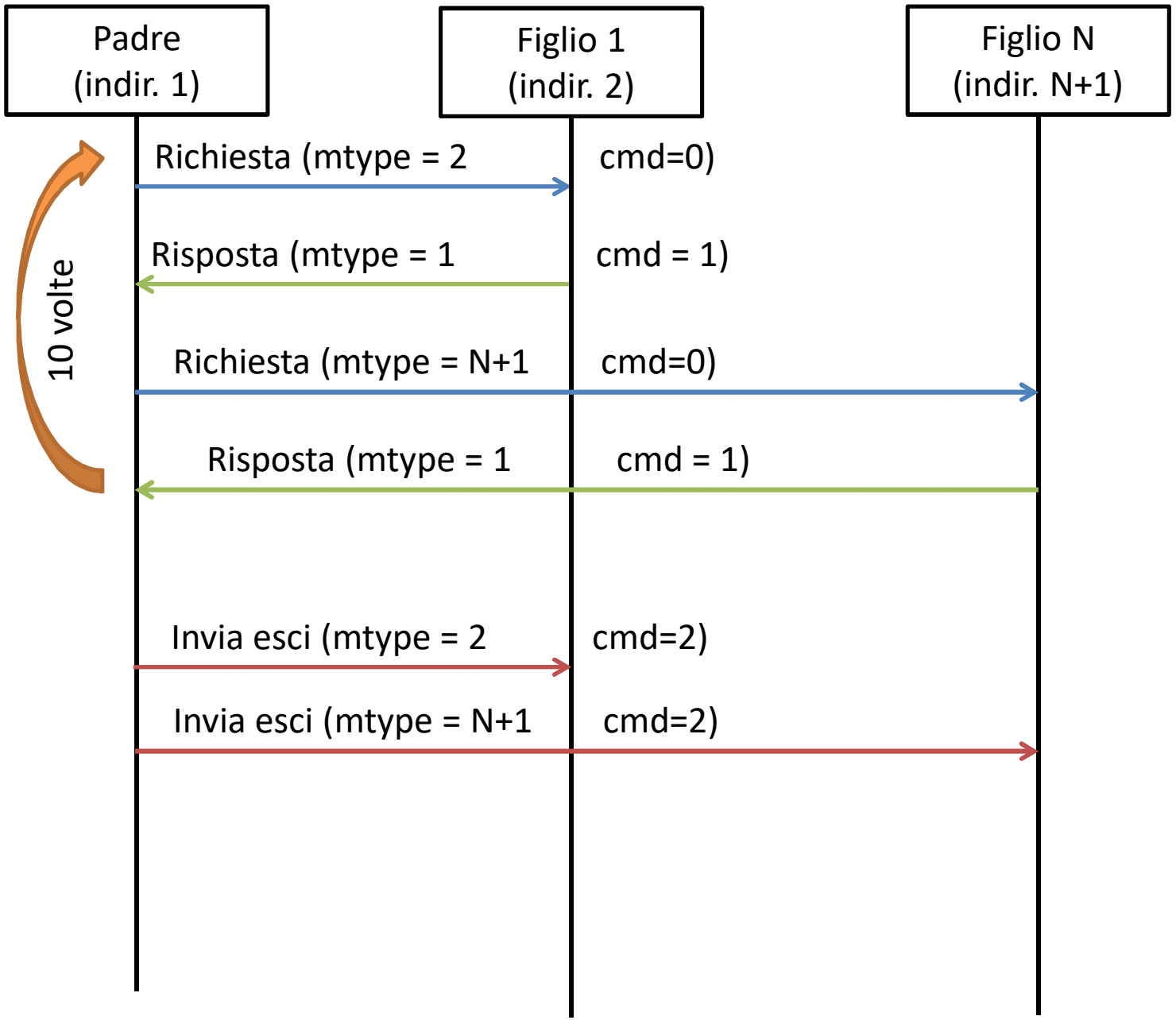
Utilizzare una coda di messaggi per la comunicazione tra padre e figli.

ESERCIZIO 3 (Suggerimenti)

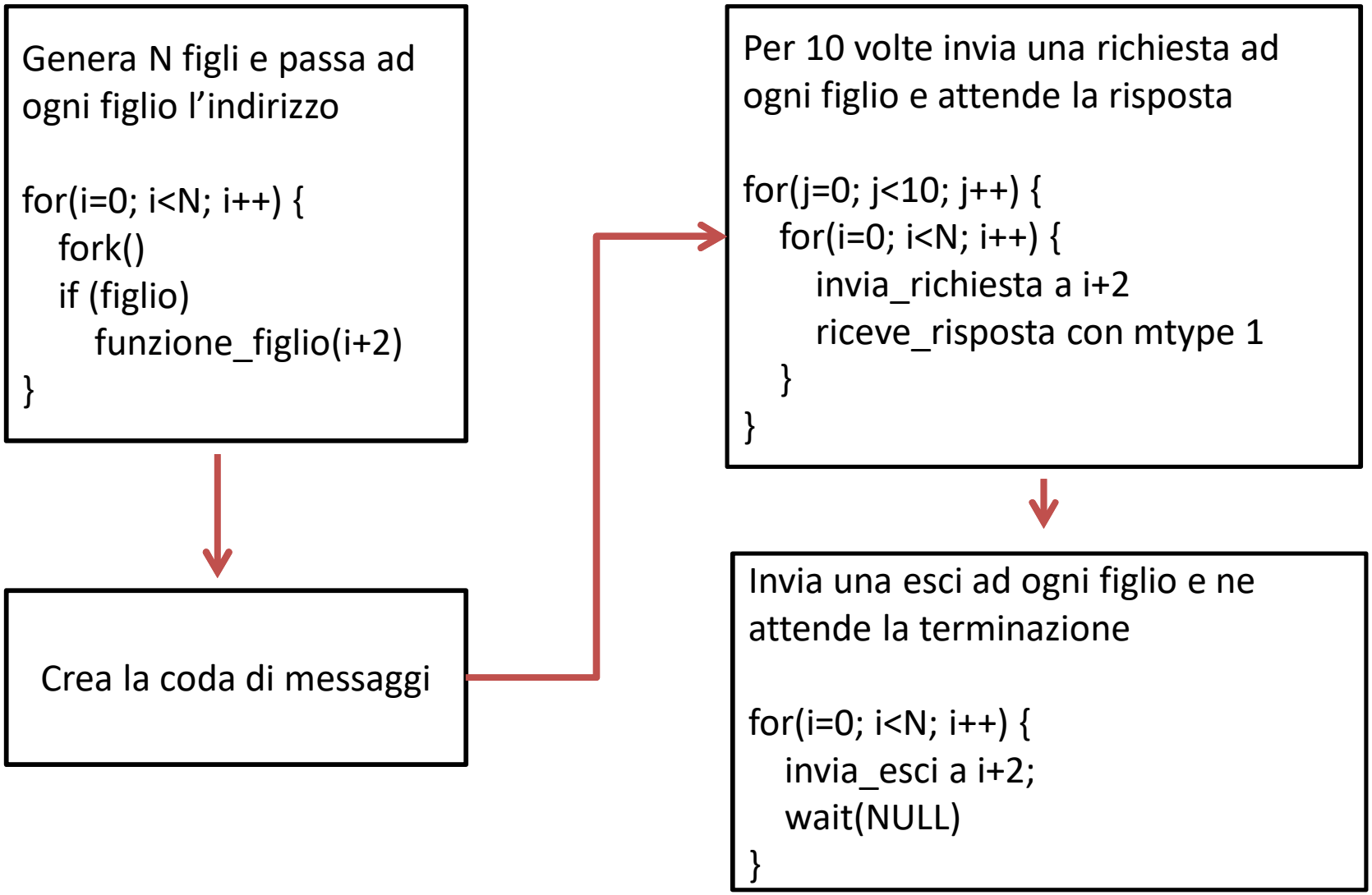
Struttura del messaggio:

```
struct message {
    long int message_type;
    int command;
    int value;
}
```

- Utilizzare il campo *message_type* per indirizzare un messaggio ad uno specifico processo.
- Utilizzare il campo *command* per specificare il tipo di comando (es. 0=Richiesta, 1=Risposta, 2=Esci).
- Il padre deve avere indirizzo 1, quindi tutti i messaggi indirizzati al padre devono avere tipo 1.


Padre
(indir. 1)
Figlio 1
(indir. 2)
Figlio N
(indir. N+1)
10 volte
Richiesta (mtype = 2
cmd=0)
Risposta (mtype = 1
cmd = 1)
Richiesta (mtype = N+1
cmd=0)
Risposta (mtype = 1
cmd = 1)
Invia esci (mtype = 2
cmd=2)
Invia esci (mtype = N+1
cmd=2)

Processo Padre



Generico Processo Figlio (indir. X)