```c
/*
   Esempio super minimale di fork. Una sola fork, e sia il figlio che
   il padre non fanno altro che scrivere i due ID. Il padre non
   aspetta il figlio e quindi, se il figlio "ritarda" diventa uno
   'zombie' e viene 'adottato' dal processo 'init', per cui il parent
   ID che stampa non è quello del padre originale. Le due opzioni -c e
   -p permettono di scegliere di quanto "ritardare" l'esecuzione della
   printf nel figlio e nel padre; il default è zero per entrambi.
*/

#include <stdio.h>              /* printf */
#include <stdlib.h>             /* exit */
#include <unistd.h>             /* fork, getopt */
#include "fun.h"

int main(int argc, char *argv[])
{
  pid_t pid;
  int sleep_child = 0;
  int sleep_parent = 0;
  int opt;

  get_opt(argc, argv,
          &sleep_child, &sleep_parent,
          NULL);

  pid = fork();
  if (pid == 0) {

    sleep(sleep_child);
    printf("child:  %d with parent: %d\n",
           getpid(), getppid());
    return 0;
  } else {

    sleep(sleep_parent);
    printf("parent: %d with child:  %d\n",
           getpid(), pid);
    return 0;
  }
}
```