```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

/*|Matheus Martins Batista (2019005687) - Sistemas Operacionais (COM120) |
  |Ciências da Computação (CCO) - EP01 - Exercício01 - 16/09/2021        |*/

int main(int argc, char *argv[]){

    int pid = 0, i;
    int pidpai, pidfilho;
    pidpai = getpid();
    pid = fork();
/*Conferir se o fork conseguiu criar um novo processo*/
    if(pid<0){
        perror("Fork");
        return 1;
    }
/*Verificar se o processo rodando é o pai ou filho com base no pid(fork do filho retorna 0)*/
/*Usar laço for para printar o status repetidamente do pai e do filho*/

    if (pid == 0){
        pidfilho = getpid();
        for(i=0;i<50;i++){
        printf("Sou processo filho!!! Meu PID é %d e o PID do meu pai é %d\n", pidfilho, pidpai);
        }
    }
    else{
        for(i=0;i<30;i++){
        printf("Sou o processo pai!!!Meu PID é %d\n", pidpai);
        }
    }
    return 0;
}
/*Rodando o código para loops de valores maiores e menores o comportamento se manteve:
o terminal imprime os PIDs de forma intercalada, iniciando com o pai e intercalando
com o filho. Aparentemente,ambos os processos estão printando ao mesmo tempo
(de forma simultânea) e a forma como o terminal expôs isso foi intercalando as impressões.
Essa percepção é devido ao momento em que o laço do processo pai termina, o processo filho
continua imprimindo o PID do filho e a recíproca também é válida.*/
```