

```

#include <stdio.h>
#include <unistd.h>
#include <ctype.h> /*khai báo các hàm isascii(), toupper()..*/

/*Cài đặt hàm điều khiển tiến trình cha nhận nhập liệu từ phía người dùng*/
void user_handler(int input_pipe[], int output_pipe[])
{
    char c; /*Lưu ký tự nhập của người dùng đưa vào*/
    int rc; /*Lưu mã trả về của hàm*/
    close(input_pipe[1]); /*Tiền trình cha không ghi vào pipe đọc*/
    close(output_pipe[0]); /*Tiền trình cha không đọc từ pipe ghi*/

    /*Lập yêu cầu người dùng nhập dữ liệu - nhận dữ liệu và chuyển vào đường ống cho tiến trình con*/
    while((c = getchar())>0)
    {
        /*Ghi vào đường ống*/
        rc = write(output_pipe[1], &c, 1);
        if(rc == -1) /*Lỗi ghi*/
        {
            perror("user_handler: pipe write error");
            close(input_pipe[0]);
            close(output_pipe[1]);
            exit(1);
        }

        /*Đọc lại dữ liệu từ tiến trình con trả về*/
        rc=read(input_pipe[0], &c, 1);
        if(rc <= 0)
        {
            perror("user_handler: read error");
            close(input_pipe[0]);
            close(output_pipe[1]);
            exit(1);
        }

        /*In dữ liệu do tiến trình con trả về ra màn hình*/
        putchar(c);
    }

    /*Đóng đầu ống đọc và ghi*/
    close(input_pipe[0]);
    close(output_pipe[1]);
    exit(0);
}

/*Cài đặt hàm xử lý dữ liệu cho tiến trình con*/
void translator( int input_pipe[], int output_pipe[])
{
    char c; /*Ký tự đọc được từ đường ống*/
    int rc; /*Lưu mã lỗi trả về*/

    /*Đóng các đầu ống không cần thiết*/
    close(input_pipe[1]);
    close(output_pipe[0]);

    /*Đọc dữ liệu từ đường ống do tiến trình cha chuyển đến*/
    while(read(input_pipe[0], &c, 1)>0)
    {
        /*Chuyển kí tự thường thành hoa*/
        if( isascii(c) && islower(c))
            c = toupper(c);
    }
}

```

```

/*Ghi trả dữ liệu về tiến trình cha thông qua đường ống thứ hai*/
rc = write(output_pipe[1], &c, 1);
if(rc == -1) /*Lỗi ghi*/
{
    perror("Translator: write");
    close(input_pipe[0]);
    close(output_pipe[1]);
    exit(1);
}
}

/*Đóng đường ống*/
close(input_pipe[0]);
close(output_pipe[1]);
exit(0);
}

/*Chương trình chính*/
int main()
{
    /*Khai báo mảng lưu số mô tả của các đường ống xuất nhập*/
    int user_to_translator[2];
    int translator_to_user[2];

    int pid; /*pid của tiến trình con*/
    int rc; /*Lưu mã lỗi trả về*/

    /*Tạo đường ống thứ nhất dùng gửi dữ liệu từ tiến trình cha đến tiến trình con*/
    rc=pipe(user_to_translator);
    if(rc == -1)
    {
        perror("main: pipe user_to_translator error");
        exit(1);
    }

    /*Tạo đường ống thứ hai dùng gửi dữ liệu từ tiến trình con trở lại tiến trình cha*/
    rc=pipe(translator_to_user);
    if(rc == -1)
    {
        perror("main: pipe translator_to_user error");
        exit(1);
    }

    /*Tạo tiến trình con*/
    pid = fork();

    switch(pid)
    {
        case -1: /*Không thể tạo tiến trình con*/
            perror("main: fork error");
            exit(1);
        case 0: /*Mã lệnh xử lý bên trong tiến trình con*/
            translator(user_to_translator,translator_to_user);
        default: /*Mã lệnh xử lý bên trong tiến trình cha*/
            user_handler(translator_to_user,user_to_translator);
    }
    return 0;
}

```