**#include** <stdlib.h>

**#include** <stdio.h>

**#include** <string.h>

**#include** <unistd.h>

**#include** <sys/msg.h>

**#include** <sys/wait.h>

**#define** MAX\_TEXT\_LEN 512

//不能在WSL下运行

struct user\_msg

{

**long** **int** msg\_type;

**char** text[MAX\_TEXT\_LEN];

};

**int** **main**(int argc, char \*\*argv)

{

**int** msgid = -1;

    //*建立消息队列*

    msgid = **msgget**((key\_t)1234, 0666 | IPC\_CREAT);

**if** (msgid **==** -1)

    {

**printf**("msgget error");

**exit**(EXIT\_FAILURE);

    }

    pid\_t reciver = **fork**();

**if** (reciver **<** 0)

    {

**printf**("fork error");

**return** -1;

    }

**else** **if** (reciver **==** 0)

    {

        struct user\_msg data;

        //*server 进程*

**while** (1)

        {

**if** (**msgrcv**(msgid, (**void** \*)&data, BUFSIZ, 0, 0) **==** -1)

            {

**printf**("msgrcv error\n"); //*some error here*

**goto** del;

            }

**printf**("You wrote: %s\n", data**.**text);

            //*遇到end结束*

**if** (**strncmp**(data**.**text, "end", 3) **==** 0)

            {

**break**;

            }

        }

    del:

        //*删除消息队列*

**if** (**msgctl**(msgid, IPC\_RMID, 0) **==** -1)

        {

**printf**("msgctl(IPC\_RMID) failed\n");

        }

**exit**(EXIT\_SUCCESS);

    }

**else**

    {

        pid\_t sender = **fork**();

**if** (sender **<** 0)

        {

**printf**("fork error");

**return** -1;

        }

**else** **if** (sender **==** 0)

        {

            struct user\_msg data;

**char** buffer[BUFSIZ];

            //*向消息队里中写消息，直到写入end*

**while** (1)

            {

**printf**("Enter some text: \n");

**fgets**(buffer, BUFSIZ, stdin);

                data**.**msg\_type = 1; //*注意2*

**strcpy**(data**.**text, buffer);

                //*向队列里发送数据I*

**if** (**msgsnd**(msgid, (**void** \*)&data, MAX\_TEXT\_LEN, 0) **==** -1)

                {

**printf**("msgget error");

**exit**(EXIT\_FAILURE);

                }

                //*输入end结束输入*

**if** (**strncmp**(buffer, "end", 3) **==** 0)

                {

**break**;

                }

**sleep**(1);

            }

**exit**(EXIT\_SUCCESS);

        }

**else**

        {

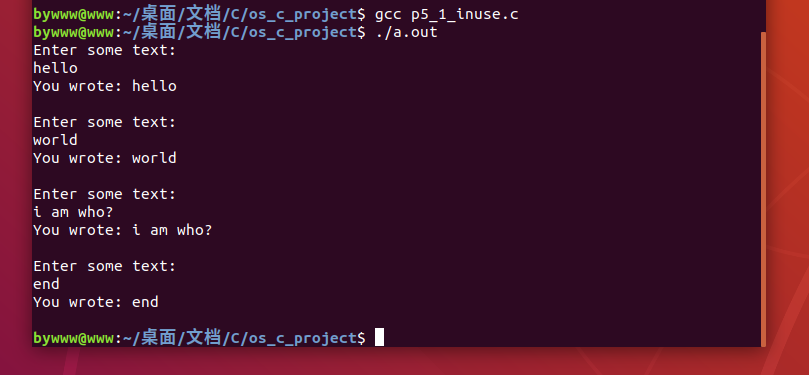
**while** (**wait**(**NULL**) **!=** -1)

            {

            }

        }

    }



**#include** <sys/types.h>

**#include** <sys/ipc.h>

**#include** <sys/shm.h>

**#include** <stdio.h>

**#include** <stdlib.h>

**#include** <unistd.h>

**#include** <sys/wait.h>

**#include** <string.h>

**#define** TEXT 2048

//这个可以在WSL下运行

struct shared\_use\_st

{

**int** read;

**char** text[TEXT];

};

typedef struct shared\_use\_st shared\_use\_st;

**int** **main**()

{

    pid\_t reciver, sender;

**int** shmid; //*共享内存的id*

    //*创建共享内存*

    shmid = **shmget**((key\_t)1234, sizeof(shared\_use\_st), 0666 | IPC\_CREAT);

**if** (shmid **==** -1)

    {

**printf**("shmget error\n");

**return** -1;

    }

    reciver = **fork**();

**if** (reciver **<** 0)

    {

**printf**("fork() error\n");

**return** -1;

    }

**else** **if** (reciver **==** 0) //*子*

    {

        //*进程链接到 共享内存*

**void** \*shm\_client = **NULL**;

        shm\_client = **shmat**(shmid, 0, 0);

**if** (shm\_client **==** (**void** \*)-1)

        {

**printf**("shmat error\n");

**return** -1;

        }

        //*指向 share mem的指针=*

        shared\_use\_st \*shared = (shared\_use\_st \*)shm\_client;

        shared**->**read = 0;

**while** (1)

        {

**if** (shared**->**read **==** 1)

            {

**printf**("you wrote:%s\n", shared**->**text);

**int** a = **atoi**(shared**->**text);

**printf**("平方根:%d\n", a \* a);

                shared**->**read = 0; //*可写*

**sleep**(1);

**if** (**strncmp**(shared**->**text, "end", 3) **==** 0)

                {

**break**;

                }

            }

**else**

            {

**sleep**(1);

            }

        }

        //*进程与共享内存分离*

**if** (**shmdt**(shm\_client) **==** -1)

        {

**printf**("shmdt error\n");

**return** -1;

        }

    }

**else**

    {

        sender = **fork**();

**if** (sender **<** 0)

        {

**printf**("fork() error\n");

**return** -1;

        }

**else** **if** (sender **==** 0) //*子*

        {

**void** \*shm\_server = **shmat**(shmid, (**void** \*)0, 0);

**if** (shm\_server **==** (**void** \*)-1)

            {

**printf**("shmat error\n");

**return** -1;

            }

            shared\_use\_st \*shared = (shared\_use\_st \*)shm\_server;

**while** (1)

            {

**while** (shared**->**read **==** 1)

                {

**sleep**(1);

**printf**("Please Waiting...\n");

                }

**printf**("Enter some text: ");

**char** buffer[BUFSIZ + 1];

**fgets**(buffer, BUFSIZ, stdin);

**strncpy**(shared**->**text, buffer, TEXT);

                shared**->**read = 1;

**if** (**strncmp**(buffer, "end", 3) **==** 0)

                {

**break**;

                }

            }

**if** (**shmdt**(shm\_server) **==** -1)

            {

**printf**("shmdt error\n");

**return** -1;

            }

        }

**else** //*父*

        {

**while** (**wait**(**NULL**) **!=** -1)

            {

            }

            //*删除共享内存*

**if** (**shmctl**(shmid, IPC\_RMID, 0) **==** -1)

            {

**printf**("shmctl IPC\_RMID error\n");

**return** -1;

            }

        }

    }

**return** 0;

}

