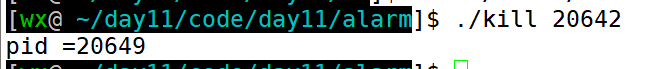
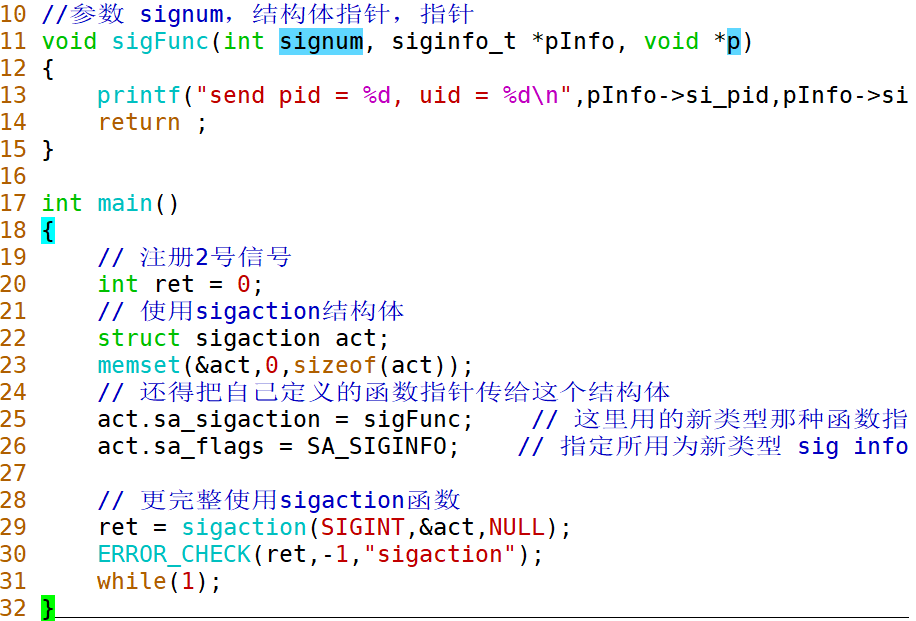
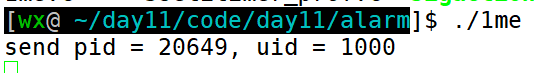
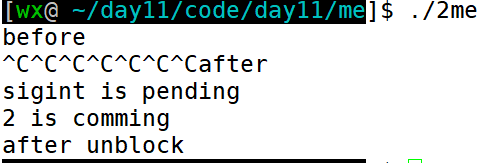
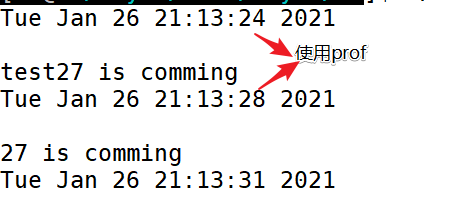
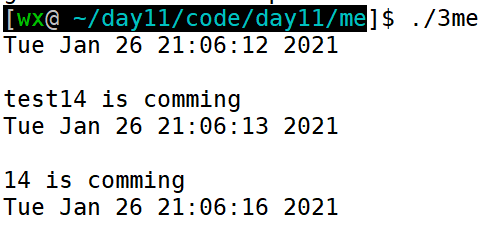
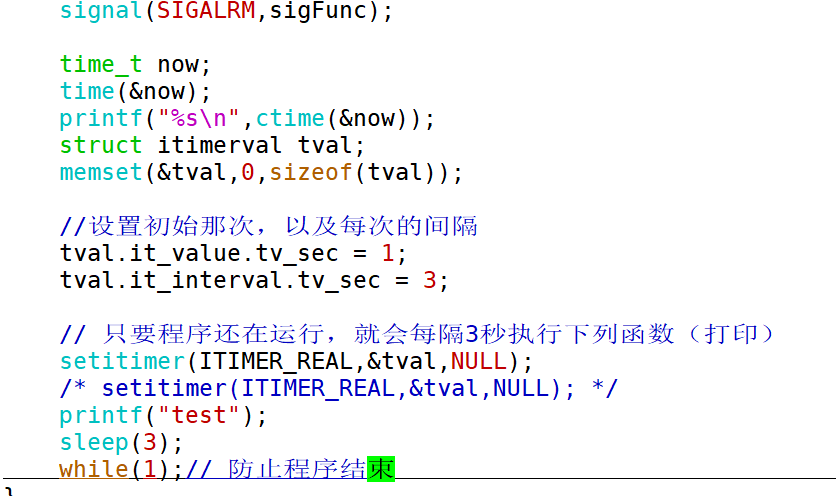
基础题

1. 通过sigaction，打印发送信号的发送者的pid和uid。

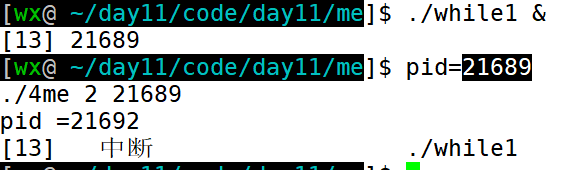
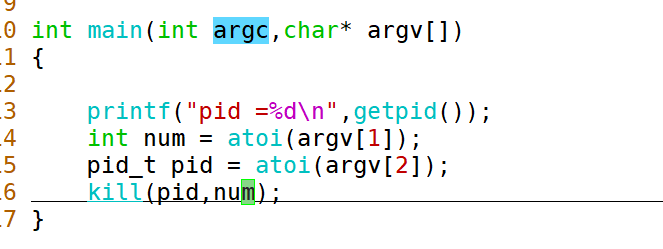




2、通过sigprocmask阻塞2号信号，睡眠5秒后，解除阻塞，2号信号得到执行；在睡眠后，解除阻塞之前，通过sigpending检测是否有信号挂起

3、使用真实计时器，实用计时器，统计程序执行时间，在实用计时器及虚拟计时器设定计时后，先睡眠，再让程序处于while（1）++

4、实现一个自己的kill命令，传参时传入信号编号和进程id。



5、什么是线程？线程和进程有哪些区别？各自的优缺点是什么？

*线程：是cpu调度和程序执行最小单元，是再共享内存中并行执行的多道执行路径，更加接近于执行体。*

*区别：线程同进程中其他线程共享进程空间（堆，代码，数据，文件描述，信号等）。只拥有自己的栈空间，大大减少了上下文切换的开销。*

*进程：*

*优点：相互之间独立，有自己的地址空间和资源，利于资源管理和保护，一个进程的崩溃不会影响其他进程。*

*缺点：开销大，进程间独立，所以对资源访问不方便，借助于进程间通信，开销变大。*

*线程：*

*优点：共享进程的资源信息，开销小，便于数据的共享，线程间通信方便。*

*缺点：共享进程的资源，不利于资源的管理和保护。*

6、主线程创建子线程，并等待子线程退出，子线程退出时返回退出码10，主线程等到子线程退出后，打印退出码。





--------------------------------------------

难度题

完成4窗口聊天

通信方式：

4个进程，A进程和B进程负责通信，从标准输入读到的字符串通过管道发给对方，A1和B1进程负责显示。

A进程从标准输入读到的字符串发给B进程后，放到共享内存里，从管道中读到的字符串也放到共享内存里，A1进程到共享内存中拿到字符串，打印到屏6幕上。

B进程从标准输入读到的字符串发给B进程，同时通过消息队列发给B1进程，B1进程从消息队列中+读出消息，打印到屏幕上。

退出 方式：

第一种：可以通过ctrl+C退出，任意一个进程收到CTRL+C时，给所有进程（包括自己）发送10号信号，每个进程收到10号信号后，

开始执行有序退出（这里有两个信号，收到2号信号表示通知所有进程退出（包括自己），收到10号信号开始执行每个进程自己的退出流程，也就是在信号处理函数里退出）。

当然也可以只用一个信号，通知其他进程并自己执行退出过程。

每个进程在自己的信号处理函数中执行有序退出。

有序退出要做的主要有：关闭管道，解除对共享内存的映射，删除共享内存，删除信号量，删除消息队列。

注意：A和A1进程都要访问共享内存，需要同信号量完成同步，所以退出时要也删除信号量。

提示：A1进程和B1进程需要知道每条消息是对方发来的消息还是发给对方的消息，用消息队列的一方可以通过mtype的值来标识，

用共享内存的进程可以自己定义一个类似 msgbuf的结构体，然后结构体中也用一个类似mtype的成员来标记。

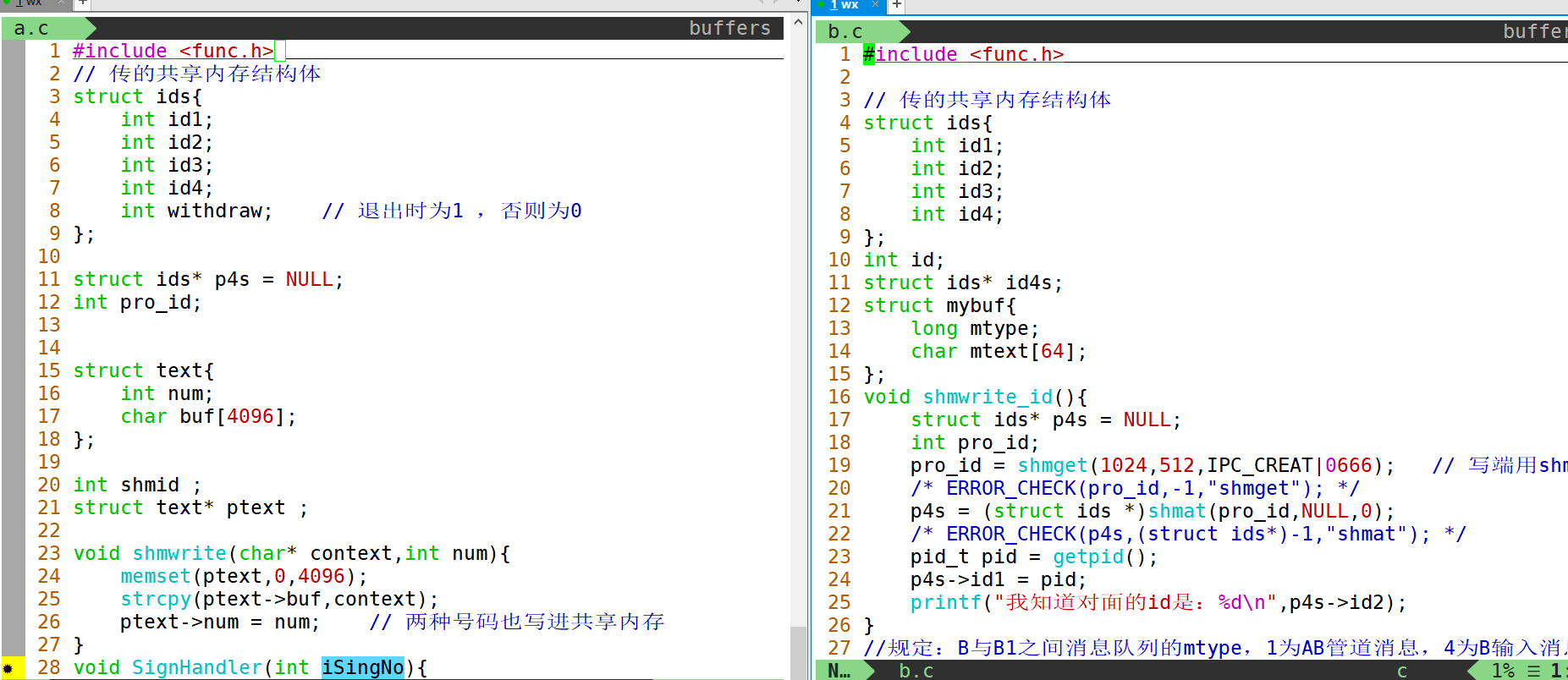
左为b1.c 右为b2.c

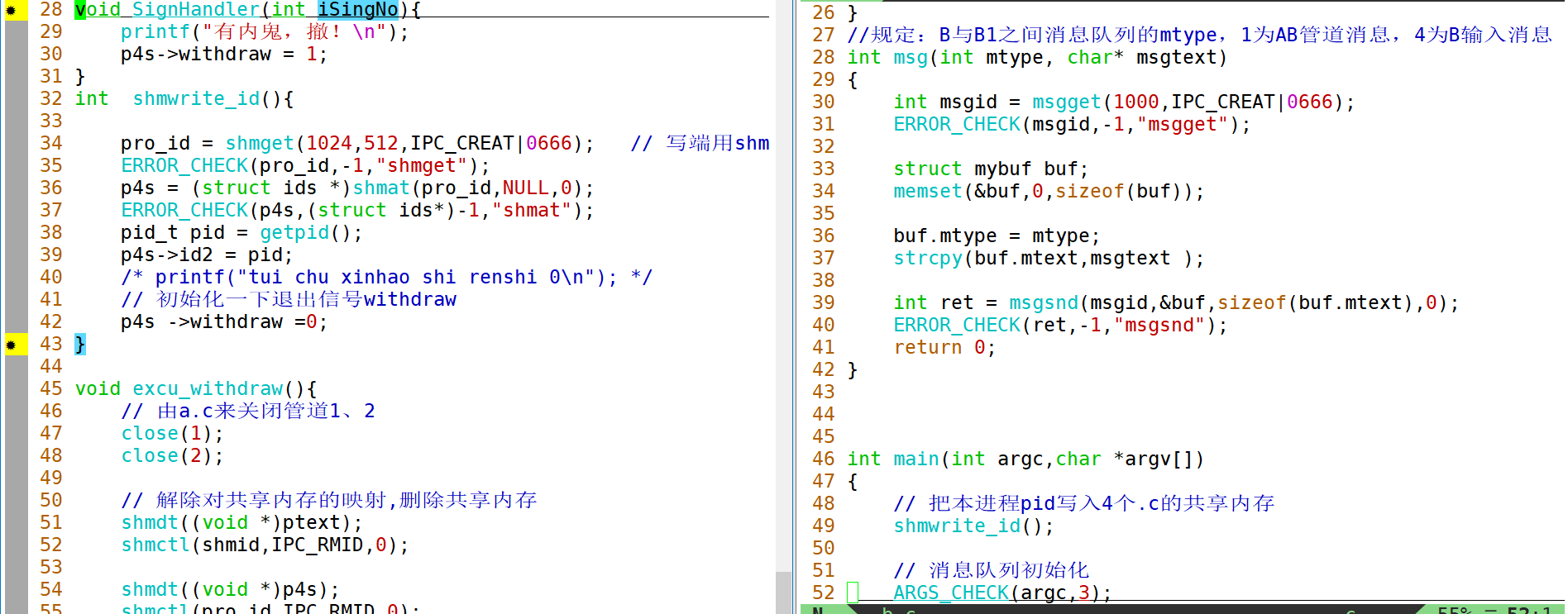
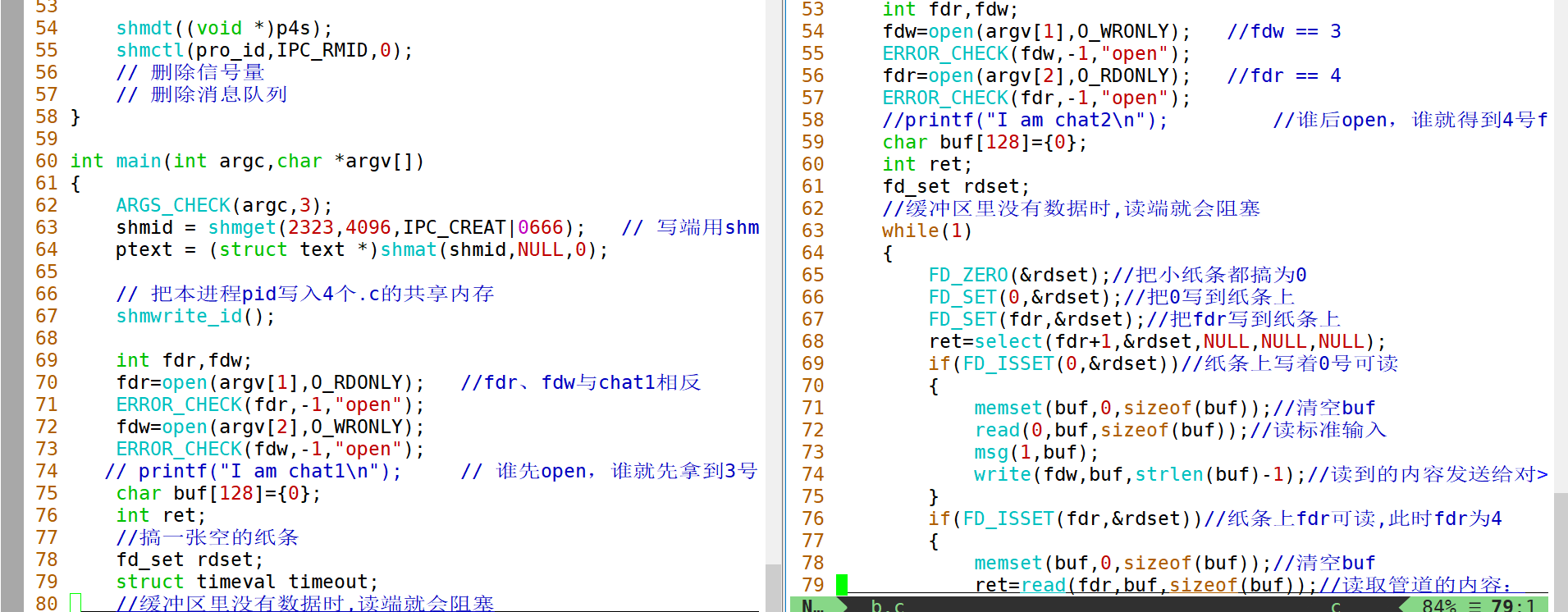




下方

左为a.c 右为b.c

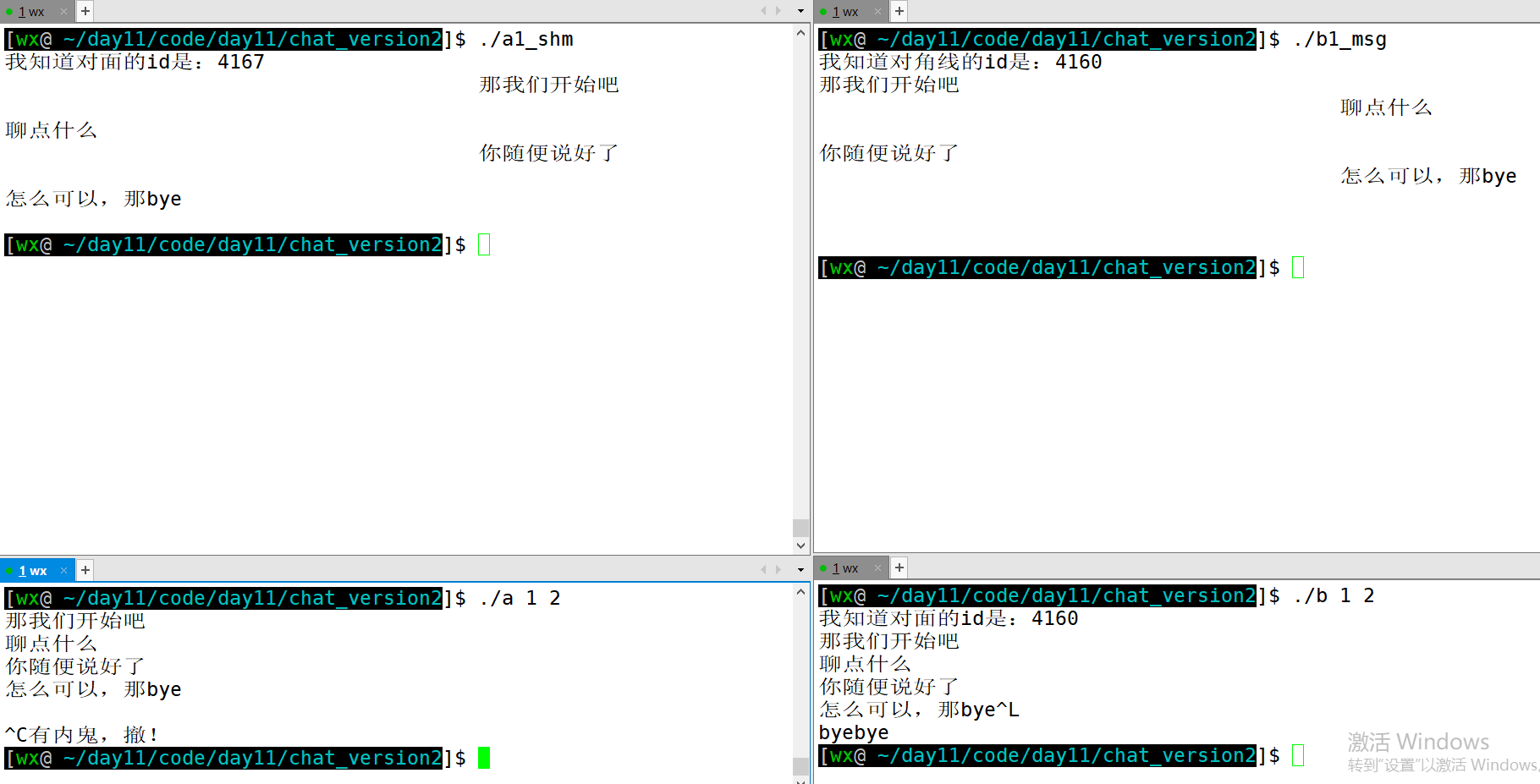






下方为四窗口聊天效果



退出：

共享内存存储结构体pid1、2、3、4，同时存储一个记录退出的信号withdraw

关闭管道1、2

按ctrl+c后，四个窗口一起关闭

