

Test case

Group 6.

Members: Li,Lun Zhou,Yating Ban,Guangwei Zhang,Xuliang

1.Topic_Create:

```
Please choose the function that you want to use[Exit/1-10] ? 1
Please type in topic name: Leung
```

```
Topic Received=> Leung
```

```
Leung Topic is added.....
```

```
-----
Welcome to IPC Test Programm 1
Author: Yating Zhou, Lun Li, GuangWei Ban, Xuliang Zhang
-----
```

```
Debug: debug mode turn on.
```

```
[ 1 ] TopicCreate
[ 2 ] TopicLookup
[ 3 ] TopicPublisher : Declare as a Publisher
[ 4 ] TopicSubscriber: Declare as a Subscriber
[ 5 ] Publish
[ 6 ] Retrieve
[ 7 ] Available Publisher(s)
[ 8 ] Available Subscriber(s)
[ 9 ] Flush Message Region
[10 ] Exit
```

```
Please choose the function that you want to use[Exit/1-10] ? _
```

```
# cc syscl_test.c -o syscltest
```

```
# ./syscltest
```

```
Please type in number of different interesting topics: 7
```

Group Topic Info

Topic	GroupID
0	109
1	110
2	111
3	112
4	113
5	114
6	115

```
-----
Select Topic you want: 5
```

```
Interest Topic 5 with GroupID 114 has been selected.
```

```
Please enter integer message: _
```

2.Topic_lookup

```
----- Topics Available -----
0.      Leung
1.      Yating
----- End -----

Welcome to IPC Test Programm 1
Author: Yating Zhou, Lun Li, GuangWei Ban, Xuliang Zhang

Debug: debug mode turn on.
[ 1 ] TopicCreate
[ 2 ] TopicLookup
[ 3 ] TopicPublisher : Declare as a Publisher
[ 4 ] TopicSubscriber: Declare as a Subscriber
[ 5 ] Publish
[ 6 ] Retrieve
[ 7 ] Available Publisher(s)
[ 8 ] Available Subscriber(s)
[ 9 ] Flush Message Region
[10 ] Exit

Please choose the function that you want to use[Exit/1-10] ? _
```

3.Topic_Publisher and

4.Topic_Subscriber

```
# cc syscl_test.c -o syscltest
# ./syscltest
Please type in number of different interesting topics: 7

      Group Topic Info
-----
Topic      GroupID
-----
0          116
1          117
2          118
3          119
4          120
5          121
6          122
-----

Select Topic you want: 5
Interest Topic 5 with GroupID 121 has been selected.
Please enter integer message: 99
Type in number of process you want to create: 5
Register pid = 670 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 0
Register pid = 671 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 672 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 673 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 674 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
```

0 = Publisher

1 = Subscriber

2 = None

5.Publish Message and

6.Retrieve message

(1) Single publisher to multiple subscribers

```
Topic          GroupID
-----
0              116
1              117
2              118
3              119
4              120
5              121
6              122
-----
Select Topic you want: 5
Interest Topic 5 with GroupID 121 has been selected.
Please enter integer message: 99
Type in number of process you want to create: 5
Register pid = 670 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 0
Register pid = 671 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 672 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 673 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 674 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Publisher[670]::Message to be sent is 99
Subscriber[671]::Message recived and message is 99
Subscriber[672]::Message recived and message is 99
Subscriber[673]::Message recived and message is 99
Subscriber[674]::Message recived and message is 99
#
```

(2)multiple publishers to single subscriber

```
Subscriber[674]::Message recived and message is 99
# ./syscltest
Please type in number of different interesting topics: 1

Group Topic Info
-----
Topic          GroupID
-----
0              123
-----
Select Topic you want: 0
Interest Topic 0 with GroupID 123 has been selected.
Please enter integer message: 555
Type in number of process you want to create: 5
Register pid = 676 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 677 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 678 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 679 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 680 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 0
Publisher[680]::Message to be sent is 555
Subscriber[676]::Message recived and message is 555
Subscriber[677]::Message recived and message is 555
Subscriber[678]::Message recived and message is 555
Subscriber[679]::Message recived and message is 555
#
```

(3) multiples publishers to multiples subscribers

```
Group Topic Info
-----
Topic      GroupID
-----
0          3
1          4
-----
Select Topic you want: 1
Interest Topic 1 with GroupID 4 has been selected.
Please enter integer message: 45678
Type in number of process you want to create: 5
Register pid = 171 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 172 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 0
Register pid = 173 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 174 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 175 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 0
Publisher[172]:Message to be sent is 45678
Publisher[175]:Message to be sent is 45678
Subscriber[171]:Message recived and message is 45678
Subscriber[173]:Message recived and message is 45678
Subscriber[174]:Message recived and message is 45678
Subscriber[171]:Message recived and message is 45678
Subscriber[173]:Message recived and message is 45678
Subscriber[174]:Message recived and message is 45678
#
```

```
Please enter integer message: 1111
Type in number of process you want to create: 7
Register pid = 163 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 164 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 0
Register pid = 165 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 0
Register pid = 166 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 167 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 168 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
Register pid = 169 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 0
Publisher[164]:Message to be sent is 1111
Publisher[165]:Message to be sent is 1111
Publisher[169]:Message to be sent is 1111
Subscriber[163]:Message recived and message is 1111
Subscriber[166]:Message recived and message is 1111
Subscriber[167]:Message recived and message is 1111
Subscriber[168]:Message recived and message is 1111
Subscriber[163]:Message recived and message is 1111
Subscriber[166]:Message recived and message is 1111
Subscriber[167]:Message recived and message is 1111
Subscriber[168]:Message recived and message is 1111
Subscriber[163]:Message recived and message is 1111
Subscriber[166]:Message recived and message is 1111
Subscriber[167]:Message recived and message is 1111
Subscriber[168]:Message recived and message is 1111
#
```

6. You will maintain a buffer that can contain 5 messages for each topic. No publisher can send to the topic when the buffer is full. On the other hand, Retrieve is nonblocking. A message is removed from the buffer after it has been retrieved by all the subscribers.

```
do_topic_publisher_msg system call is involed
```

```
Entered Critical Region...!!!
```

```
Buffer is full. Can't publish msg.
```

```
Released Critical Region...!!!
```

```
-----  
Welcome to IPC Test Programm 1
```

```
Author: Yating Zhou, Lun Li, GuangWei Ban, Xuliang Zhang  
-----
```

```
Debug: debug mode turn on.
```

```
[ 1 ] TopicCreate  
[ 2 ] TopicLookup  
[ 3 ] TopicPublisher : Declare as a Publisher  
[ 4 ] TopicSubscriber: Declare as a Subscriber  
[ 5 ] Publish  
[ 6 ] Retrieve  
[ 7 ] Available Publisher(s)  
[ 8 ] Available Subscriber(s)  
[ 9 ] Flush Message Region  
[10 ] Exit
```

```
Please choose the function that you want to use[Exit/1-10] ?
```

7. A message is published once and retrieved by all subscribers who had subscribed to the topic before the message was published, at least once and at most once.

```
Please enter integer message: 1111
```

```
Type in number of process you want to create: 7
```

```
Register pid = 163 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
```

```
Register pid = 164 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 0
```

```
Register pid = 165 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 0
```

```
Register pid = 166 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
```

```
Register pid = 167 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
```

```
Register pid = 168 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 1
```

```
Register pid = 169 as a 0=Publisher/1=Subscriber/2=None[0/1/2]: 0
```

```
Publisher[164]::Message to be sent is 1111
```

```
Publisher[165]::Message to be sent is 1111
```

```
Publisher[169]::Message to be sent is 1111
```

```
Subscriber[163]::Message recived and message is 1111
```

```
Subscriber[166]::Message recived and message is 1111
```

```
Subscriber[167]::Message recived and message is 1111
```

```
Subscriber[168]::Message recived and message is 1111
```

```
Subscriber[163]::Message recived and message is 1111
```

```
Subscriber[166]::Message recived and message is 1111
```

```
Subscriber[167]::Message recived and message is 1111
```

```
Subscriber[168]::Message recived and message is 1111
```

```
Subscriber[163]::Message recived and message is 1111
```

```
Subscriber[166]::Message recived and message is 1111
```

```
Subscriber[167]::Message recived and message is 1111
```

```
Subscriber[168]::Message recived and message is 1111
```

```
#
```

8. We ask you to specifically address these questions: Can deadlock occur with your IPCs? If not, why? If yes, do you detect deadlock and how? If so, how you can recover from it?. If you do not detect deadlock, you should also say why. If deadlock can occur, can you advise the programmer to avoid it?

```
core:341      tests      testhelper.h
# gcc testdeadlock_1.c -o td1
# ./td1
You may need CTRL+C to exit block
Deadlock occur: 183001(endpoint)->183002(endpoint)
TEST_EQUAL [parent receive success(blocked), and child deadlock] OK.
_
```

```
0
# gcc testdeadlock_2.c -o td2
# ./td2
You may need CTRL+C to exit block
child id 1080
child id 1081
child id 1082
child id 1083
Deadlock occur: 183012(endpoint)->183009(endpoint)
TEST_EQUAL [deadlock occur and only occur once.] OK.
deadlock from 1083 to 1080
# _
```