

Topic Ids will be sent in as integers and they are broken into groupId and topicId.

Subscribe:

- Subscribe checks if it is a valid topicId, and then checks if a process is not trying to subscribe to the same topicId again (with same or different groupId).
- If the syscall moves further then it puts the function handler groupId in the topicTab[][] table.

Publish:

- This syscall just enqueues the topic and data that needs to be published in pendingQueue[] and broker picks it up one by one and publishes to the subscribed processes.

Broker:

- This process is always running, and it picks up events to be published from pendingQueue[], and after processing them it dequeues them.
- It checks if groupId is zero then it calls function handler of all the processes that have subscribed to that particular topicId, otherwise it also matches the group Id and then calls the function handler.
- It sends a signal in both the above cases to the process whose function Handler is called.
- If there are no processes that have subscribed to a given topicId then it just prints "No functions Called".

Unsubscribe:

- Unsubscribe updates groupId in topicTab[][] of the corresponding topicId to -1.

Data Structures Used:

2-D Array

- 1) The data structure I have chosen to store topic Ids is a 2-d Array, whose elements are a struct containing groupId and function handler.
- 2) This table is called topicTab[TOPIC_TABLE_SIZE][ARRAY_SIZE], where TOPIC_TABLE_SIZE is 256 and ARRAY_SIZE is 8.
- 3) Each column in this array corresponds to a process Id, as NPROC is 8 and hence process Id is not separately stored.
- 4) Each row represents a topicId.
- 5) This declaration is done in process.h, and the 2-d array is initialized in initialize.c with groupId as -1.
- 6) groupId = -1 means no process has subscribed for that particular topic Id.

Circular Queue

- 1) A circular queue is maintained to store all the topicIds and data a process wants to publish to a particular topicId.
- 2) It is declared in queue.h
- 3) This queue is called pendingQueue[QUEUE_SIZE], where QUEUE_SIZE is 256.
- 4) Methods like enqueueProc(), dequeueProc() are implemented in queue.c