Network Protocol Assignment : Simple Middleware

2016026026 컴퓨터전공 최솔비

# 구현한 기능

* topic 자료구조 (common.py)
* message broker (message\_broker.py)
  + - pub list, sub list 를 이용한 match 관리
    - match 되는 Topic owner의 주소를 전달
    - keep alive 관리
* publisher (sub\_pub.py)
  + - topic 등록
    - matched subscriber 에게 data (file) 전송
    - 설정에 따라 periodical 하게 data 반복 전송
    - keep alive 응답
* subscriber (sub\_pub.py)
  + - topic 등록
    - matched publisher 로부터 data 수신
    - keep alive 응답
* 모든 publisher 와 subscriber 는 하나의 토픽만 송, 수신 할 수 있으며, 하나의 publisher 는 여러 명의 subscriber를 관리하며 동시에 data 를 전송할 수 (정확히는 multi thread) 있습니다.
* publisher 간 중복되는 topic 이 없다고 가정하였습니다. 각 subscriber는 하나의 publisher와 매치되어 데이터를 전송 받게 됩니다.
* data 를 파일이라 가정하였습니다. 각각의 publisher 는 프로그램 시작 시 제공하는 file path에 해당하는 파일을 매치된 subscriber 에게 전송하게 됩니다. 파일을 전송 받은 subscriber 는 프로그램시작 시 전달받은 file path에 파일을 저장합니다.

# 사용법

1. 하나의 message broker 프로세스와 여러개의 pub, sub 프로세스를 실행한다.
2. python3 message\_broker.py
3. python3 sub\_pub.py (파일 경로 및 이름)

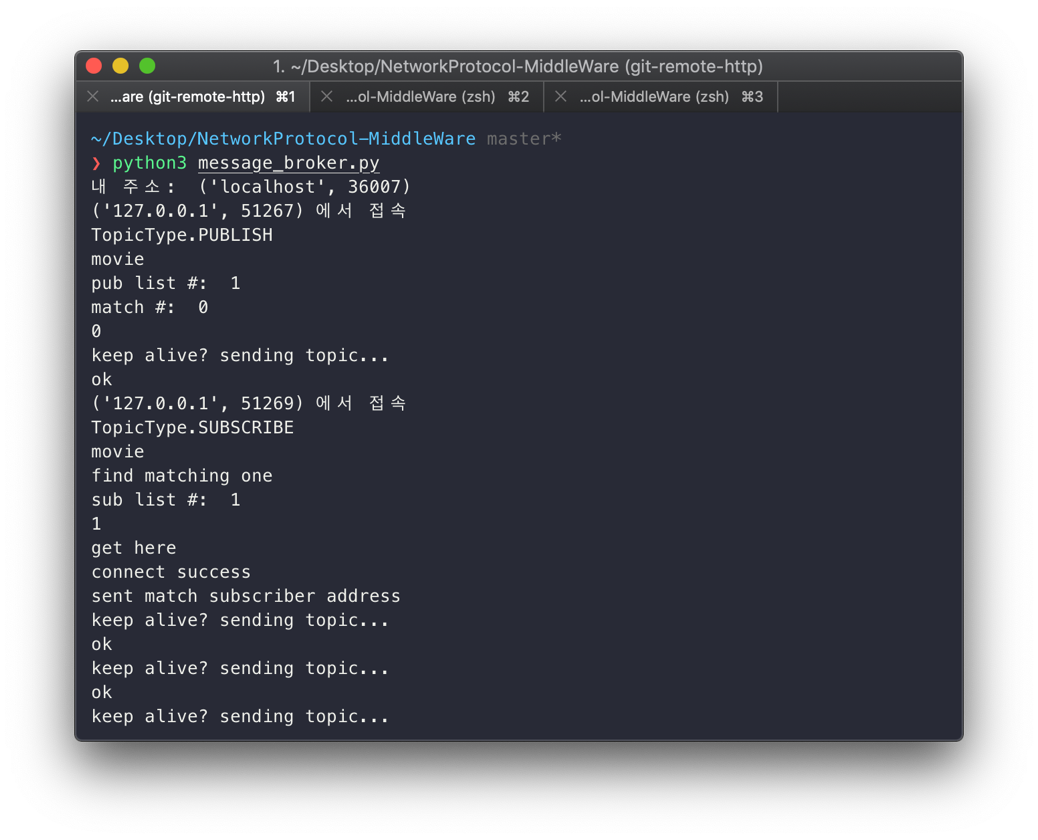
파일이름에는 publisher의 경우 publish 하고자 하는 파일, subscriber의 경우 구독한 파일을 저장할 파일 이름을 작성한다.

실행 순서는 sub, pub 가운데 어느 것 먼저 작동 시켜도 무관하다.

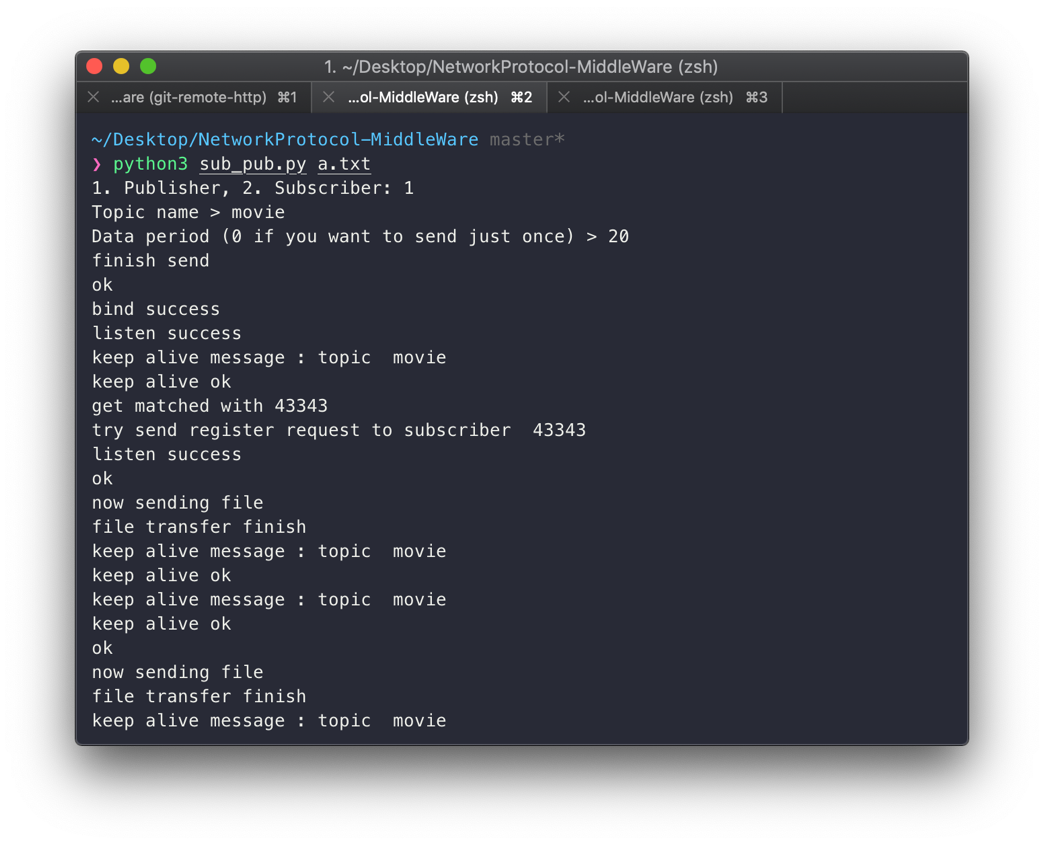
1. publisher, subscriber 프로세스 모두 하나의 파일, 하나의 토픽만 요청, 전달할 수 있다.
2. sub\_pub.py를 실행시키면, 먼저 publisher, subscriber 가운데 하나를 선택한 후 topic name 을 작성 한다.
3. publisher 의 경우 period 또한 작성한다. 작성한 period(sec) 마다 subscriber 에게 data 를 재전송하게 된다.

# 시나리오

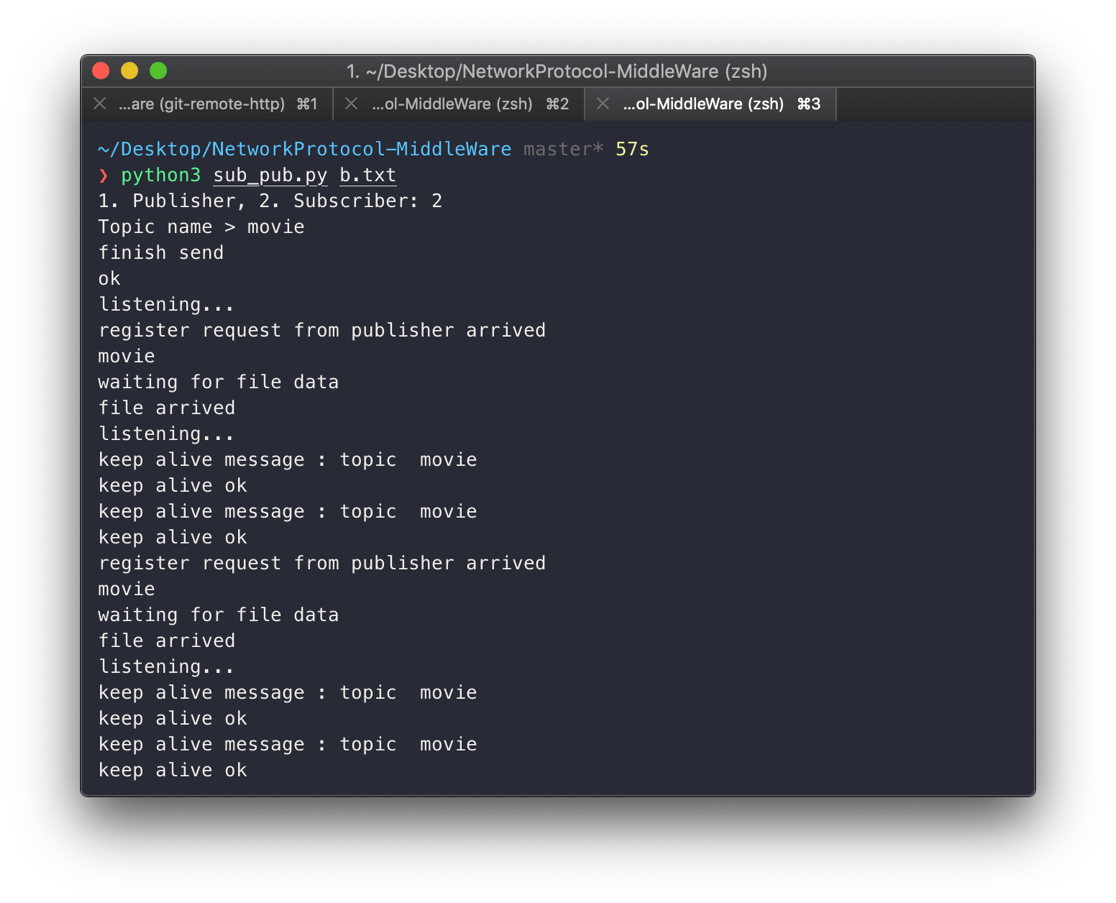
* 여러 명의 subscriber가 1개의 토픽을 기다리고 있다가 해당 토픽을 가진 publisher 가 나타난 경우 => 모두에게 파일 전송 잘 되는 것 확인
* 한 명의 publisher 가 있고 해당 topic 을 여러 명의 subscriber 가 순차적으로 원하는 경우 => 계속해서 파일 제공 하는 것 확인
* keep alive: 주기적으로 message broker 에서 pub, sub 에게 keep alive 메세지 보낸 후 응답 없거나 fail할 경우 pub list, sub list 에서 제거. => pub, sub 프로세스 종료 시 잘 제거되는 것 확인
* periodical publish: 입력한 시간(sec)마다 주기적으로 subscriber에게 connect하여 파일전송.
* subscriber가 한 publish 컨텐츠를 구독하다가 pub process 종료 후, 같은 토픽을 가진 다른 publisher 나타났을 때 matching 됨.



Message broker 프로세스 실행 화면입니다. Publisher 를 먼저 등록한 후 해당 토픽과 매치되는 subscriber가 등록되자 매치하고 있습니다. 또한 계속해서 등록 된 모든 pub, sub에게 keep alive message를 보내고 있습니다.



Publisher 프로세스 실행 로그입니다. Movie 토픽을 20초마다 제공하도록 설정하였습니다. Topic 등록한 다음부터 keep alive message 를 받고 있으며, 브로커로부터 43343 subscriber와 매치됨을 전달받은 후 해당 subscriber에게 register request 보내는 모습을 확인할 수 있습니다. Ok 를 받은 후 , a.txt 파일을 보내게 됩니다. 20초마다 커넥션을 재개해서 파일을 지속적으로 전송하고 있습니다.



Subscriber 프로세스 실행 로그입니다. Movie topic 을 등록한 후 publisher로부터 register request 를 전달 받습니다. 토픽이 movie인 것을 확인한 후 ok 메세지를 보내면 publisher 로부터 파일 데이터를 받게 됩니다. 전송 받은 파일은 b.txt로 저장됩니다. Subscriber 또한 topic 등록 후 broker로부터 keep alive message 를 받습니다. Topic 이 다르거나 프로세스가 종료되어 응답이 timeout 되면 broker 측에서 해당 토픽(및 주소 등이 들어있는 자료구조)을 삭제하게 됩니다.