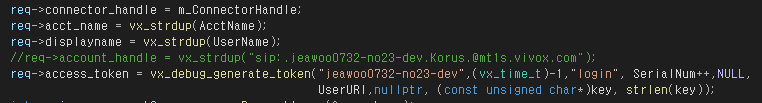
**2018180020 박재우(클라이언트) 25주차 기록**

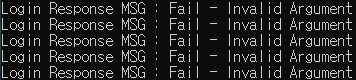
**Vivox**

저번주에 에코채널에서 음성채팅이 되는지 확인했으므로 이번엔 다른 클라이언트와의 음성채팅이 되는지 확인한다.

채팅을 위해서는 각 유저의 URI를 만들어야 한다. URI는 겹치지 않는 유저의 이름을 기반으로 만들어야 나중에 겹치지 않을 것이다. 저번주에 한 작업에서 유저의 이름이 들어가는 부분을 전부 변수로 바꿔주도록 한다.



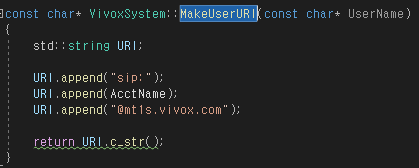
AcctName과 UserName 그리고 UserURI로 대체했다. UserName은 게임 시작하면서 닉네임을 입력할 때 받아올 예정이다. 이는 서버와 연결하는 과정에서 일어나므로 나중에 서버 담당인 롭비랑 같이 작업하도록 한다. AcctName은 유저의 이름 양옆에 . 을 붙인다. UserURI는 sip:.유저.@mt1s.vivox.com 형식이라 유저 부분에 받아온 UserName만 넣으면 완성된다. 이 변수들은 VivoxSystem 클래스 생성자에서 만들어준다.



수정을 했더니 Invalid Argument 오류가 생겼다.



UserName, AcctName, URI를 차례로 출력해봤는데 URI가 문제가 있다.



URI는 이 함수에서 만들어지는데 함수가 끝나면서 URI.c\_str()을 반환해주는데 이는 포인터가 반환되기 때문에 만들어진 지역변수 URI가 지워지면서 없어지는 문제이다. 반환을 하지 말고 직접 URI 변수에 넣어주는 방법으로 한다



Char\*로 선언되어 있는 UserURI를 URI의 길이만큼 동적할당 해준 뒤에 strcpy함수를 이용해 문자열을 저장한다.

AcctName 역시 잘못된 방법으로 생성자에서 만들어주고 있었다. 이러면 생성자가 끝나면 AcctName도 사라질 것이기 때문에 URI와 같은 방법으로 수정해준다.



문제없이 잘 생성이 되었다. 이대로 에코채널에 접속해도 잘 작동되었다.



이제 에코채널이 아니라 다른 사람과 채팅을 할 수 있는 채널에 입장하도록 한다.

채널 URI를 vx\_get\_echo\_channel\_uri 가 아니라 vx\_get\_general\_channel\_uri 함수를 이용해 생성해준다. 에코채널과 제너럴 채널의 차이는 사실 uri앞에 붙는 구절이 confctl-e- 인지 confctl-g- 인지의 차이이다.

실행을 해보니 AddSession까지 성공했다. 아마 음성채팅 채널에 입장을 한 것 같은데 확인을 할 수 없다. 그래서 일단 이름이 다른 빌드를 두개 뽑아 놓고 동시에 실행시켰다. 하나만 틀었을 때는 아무것도 들리지 않지만 다른 클라이언트를 틀면 내가 낸 소리가 다시 들린다. 일반 음성 채팅 채널에 접속한 것을 확인할 수 있다. Vivox관련한 작업은 이제 서버에서 여러 필요한 데이터를 받아와 채널과 유저명을 잘 설정해주는 작업만 남았다.