소켓통신 미니프로젝트

이찬호

구현 내용

-서버
Accept 시스템 콜을 논 블로킹을 설정해 여러 클라이언트가 동시에 접속을 할 수 있게 만들고
클라이언트 소켓을 비동기로 처리해 SIGIO 시스템 콜을 통해 IO가 일어나면 바로 이벤트를 발생시켜
다른 연결된 클라이언트들에게 입력받은 메시지를 브로드캐스팅 하게 만들어주었다

-클라이언트
서버 소켓과 연결 한 후
회원가입.로그인 루프를 돈 후 로그인 성공시 루프를 탈출하고 Fork 시스템 콜을 통해 서버로부터 입력을 받는 기능과 서버에게 메시지를 입력하는 기능을
부모 자식 프로세스에서 각각 수행하게 만들었다

정보 저장

멀티 프로세스로
각각의 클라이언트의 입력을 받아서 처리하고

회원가입은 member 테이블에 회원 정보를 저장

로그인은 membe에 id, pw를 이용한 select 쿼리를 보내 일치하면 nickname을 가져온다

사용자가 입력한 메시지는 messages 테이블에
텍스트, 입력시간, 입력한 사용자 nickname을 포함해 저장되어 차후에 해당 사용자가 입력한 텍스트 목록 조회 등의 기능에 사용하기 위해 만들어두었음

작동 화면

```
pi@raspberrypi:~/socketServer $ aarch64-linux-gnu-gcc -o tcpServer tcpServer.c -I/usr/include/mariadb -L/usr/lib/aarch64-linux-gnu -lmariadb
pi@raspberrypi:~/socketServer $ ./tcpServer
```

서버는 실행 후 데몬으로 백그라운드에서 동작

```
ubuntu@DESKTOP-2APDQL3:~/network/tcp$ gcc -o client tcpClient.c
ubuntu@DESKTOP-2APDQL3:~/network/tcp$ ./client
Connected to server
1 : 회원가입 2: 로그인
1
Enter ID: veda
Enter PW: 1111
nickname: veda
register response: register_success
register success.
1 : 회원가입 2: 로그인
2
Enter ID: asdf
Enter PW: zxcv
login response: login_fail
Login failed. Try again.
1 : 회원가입 2: 로그인
2
Enter ID: testid
Enter PW: testpw
login response: test
nickname : test
Login success!
hi
sender: veda msg: hello
sender: mariadb msg: hola
hello world!
sender: mariadb msg: a lot space key clicekd test
c a n  r e a d s p a c e
```

```
ubuntu@DESKTOP-2APDQL3:~/network/tcp$ ./client
Connected to server
1 : 회원가입 2: 로그인
2
Enter ID: veda
Enter PW: 1111
login response: veda
nickname : veda
Login success!
sender: test msg: hi
hello
sender: mariadb msg: hola
sender: test msg: hello world!
sender: mariadb msg: a lot space key clicekd test
sender: test msg: c a n  r e a d s p a c e
```

```
ubuntu@DESKTOP-2APDQL3:~/network/tcp$ ./client
Connected to server
1 : 회원가입 2: 로그인
2
Enter ID: tt
Enter PW: pp
login response: mariadb
nickname : mariadb
Login success!
sender: test msg: hi
sender: veda msg: hello
hola
sender: test msg: hello world!
a lot space key clicekd test
sender: test msg: c a n  r e a d s p a c e
```

@ 원격 데이터베이스로 회원가입/로그인, 메시지 내역 저장 기능 구현

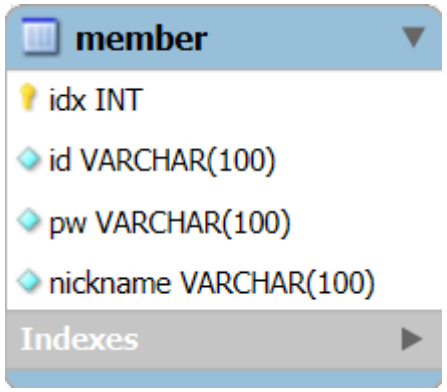
라즈베리파이에 mysql 설치를 실패해서 mariadb 설치
sudo apt-get install libmariadb-dev
라즈베리파이에서 서버를 컴파일할때 옵션 붙여주기

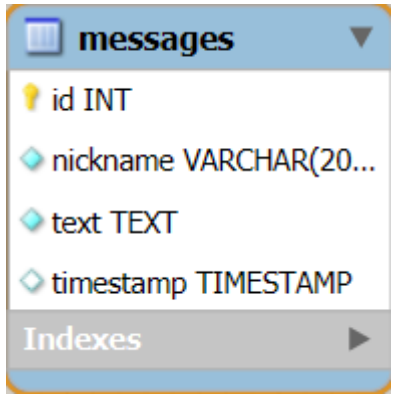
aarch64-linux-gnu-gcc -o tcpServer tcpServer.c -I/usr/include/mariadb
-L/usr/lib/aarch64-linux-gnu –lmariadb

리눅스에 있는 tcpClient는 gcc –o client tcpClient.c 형태로 컴파일 해서 사용
* 둘다 makefile로 사용 가능

회원 id, pw, 닉네임 저장
Member 테이블

idx	id	pw	nickname
1	testid	testpw	test
2	test2id	test2pw	hello
3	testid3	testpw3	test3
4	t4	p4	n4
6	t7	p7	hoshi03
7	tt	pp	mariadb
NULL	NULL	NULL	NULL





텍스트 내역 저장
Messages 테이블

1	test	hi	2024-09-13 05:38:56
2	n4	hello	2024-09-13 05:38:59
3	hello	t2t2	2024-09-13 05:41:34
4	hello	asd	2024-09-13 05:41:39
5	hello	asd	2024-09-13 05:41:40
6	hello	asd	2024-09-13 05:41:41
7	hello	asd	2024-09-13 05:41:41
8	hello	as	2024-09-13 05:41:41
9	hello	as	2024-09-13 05:41:42
10	n4	umm	2024-09-13 05:41:45
11	hello	ummm	2024-09-13 05:41:51
12	test	hi	2024-09-13 06:02:21
13	n4	hololo	2024-09-13 06:02:29
14	n4	hi	2024-09-13 06:04:58
15	test	hello	2024-09-13 06:05:01
16	hello	hey	2024-09-13 06:05:23

회원가입/로그인 기능
사용자 입력한 메시지 저장 기능을 위해
Azure에 Mysql 원격 호스팅 해서 사용
(현재 원격DB의 id/pw가 다 노출되어서 향후에 보안 강화 예정)
라즈베리파이에서 mysql 사용 불가능하기에
MariaDB로 mysql 조정

회고

구현하기 전에는 이렇게 힘든 작업이 될 지 몰랐다...
처음에 블로킹/논블로킹 개념과 동기/비동기 개념도 없는 터라 fork로 흐름만 바꿔주면 자동으로 병렬? 처럼 처리될 줄 알았는데 어림도 없다는 걸 알았고

현재 sigio를 이용해서 여러 클라이언트가 각각 입력하는 기능은 구현했으나 동시성 문제나 메모리 낭비 문제가 심각해 기능 구현에만 급급했었다.

다음에는 조금 더 효율적인 코드를 작성할 수 있도록 복습, 노력을 해야겠다고 다짐했다.

데이터베이스의 경우에도 db연동 자체는 성공했으나 저장해둔 메시지를 조회하는 기능을 넣지 못하였고 join등의 이벤트 없이 떨어진 테이블 2개만 만들었기에 많이 아쉬웠다.

다음 프로젝트에서는 더 좋은 모습을 보여줄 수 있게 노력하자..