

# 네트워크 프로그래밍

## 3차 과제

---

소프트웨어공학과 3학년  
2016156026 학번 이형석

# 목차

## 1

### 최종 목표

단체 채팅방 구현  
로그인/회원가입 기능 추가  
재 로그인 가능

## 2

### 설계 관련 내용

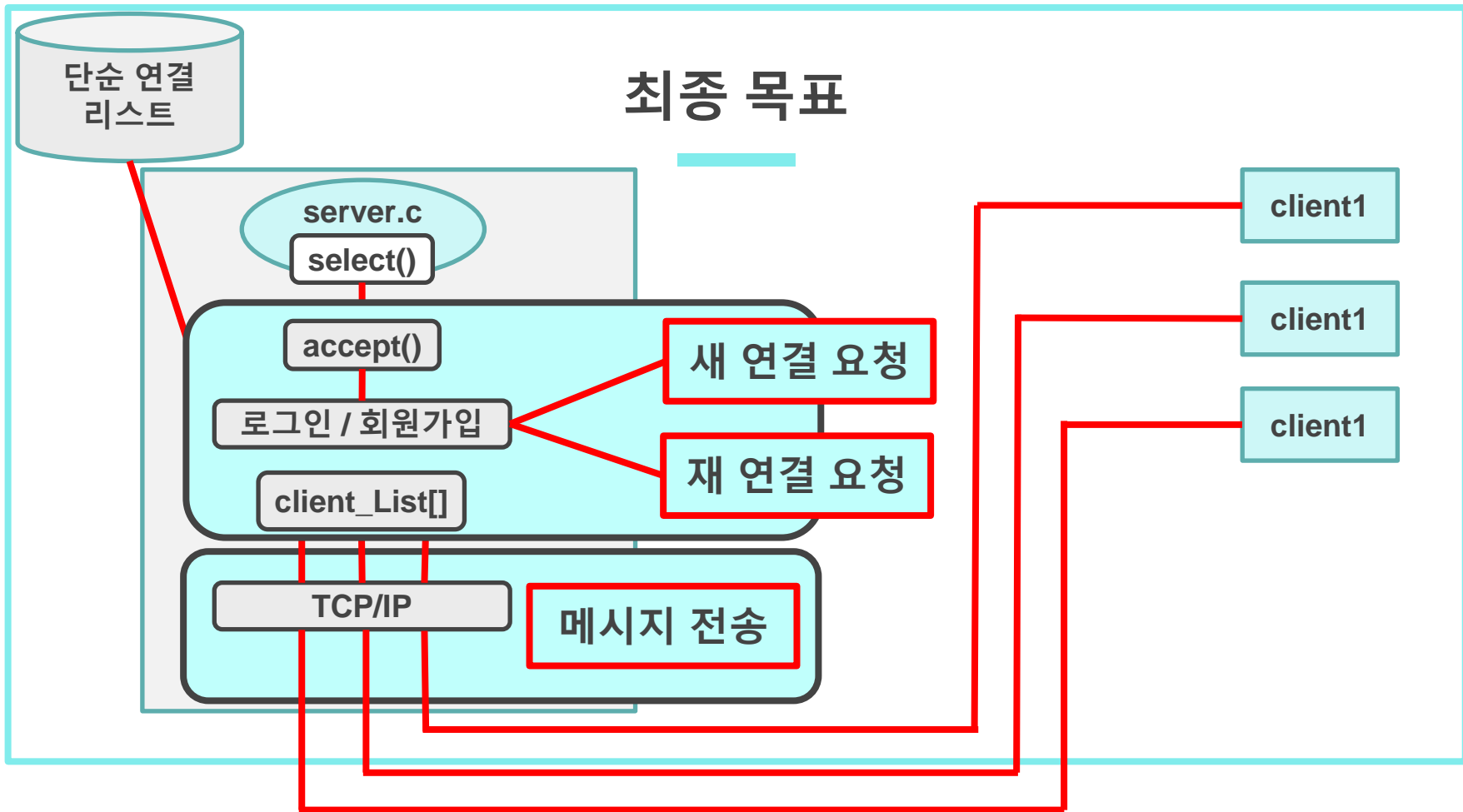
1. 설계 환경 : 우분투 16ver
2. 설계 내용 및 결과
3. 변동 내용

## 3

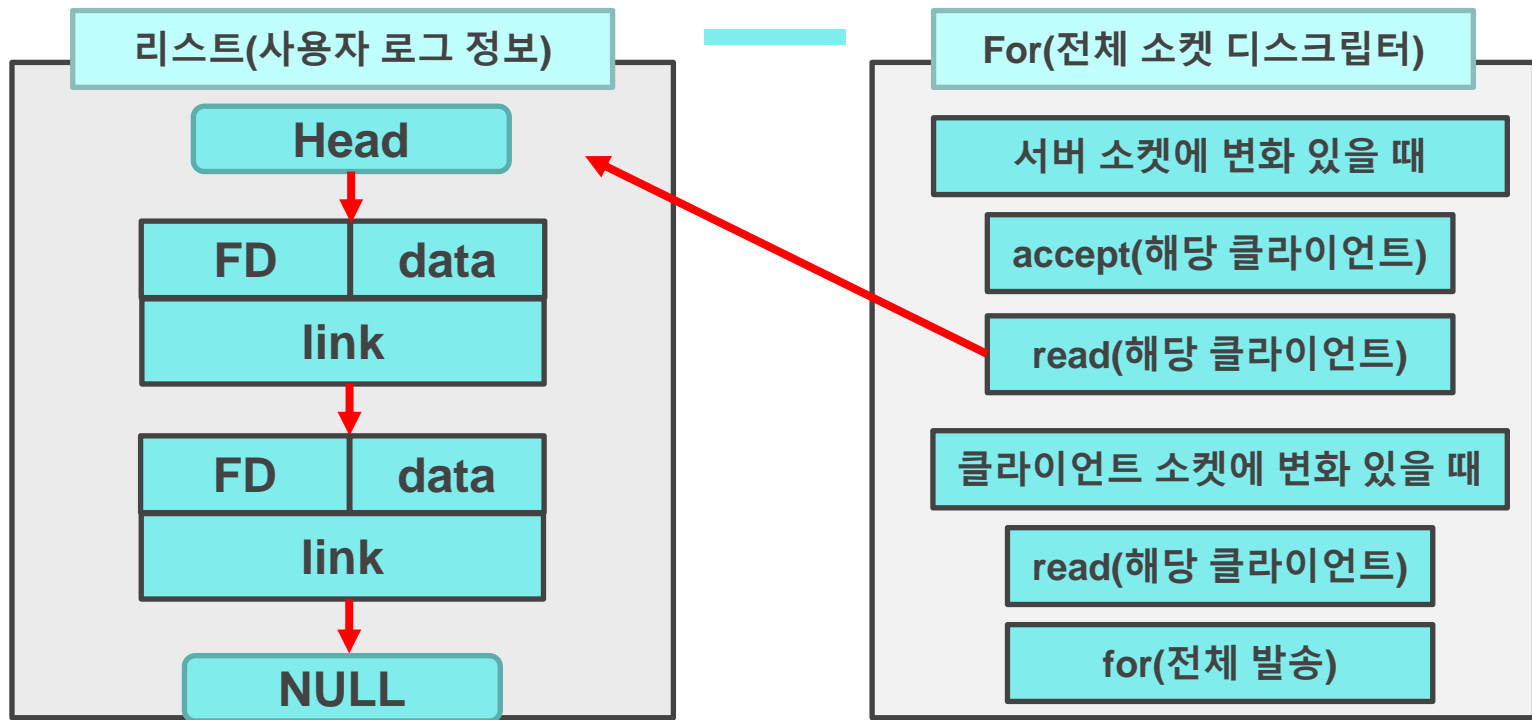
### 교훈

프로젝트 이후 교훈

# 최종 목표



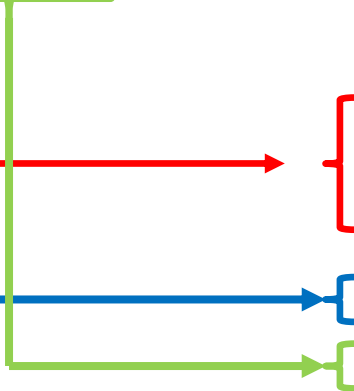
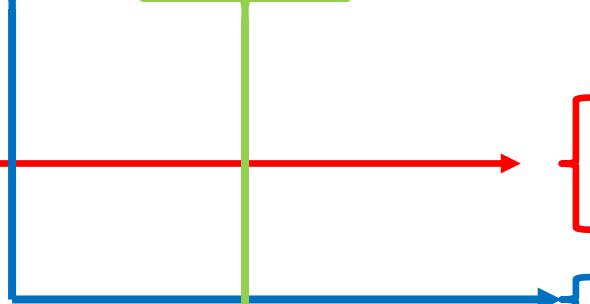
# 로그인 / 회원가입 기능(서버)



# select 함수의 사용

for

0	0	0	0	0	0	0
0	1	2	3	4	5	6



0	입력
1	출력
2	에러
3	서버 소켓
4 이상	클라이언트 소켓

# 새로운 소켓 연결 요청

for

0	0	0	1	0	0	0
0	1	2	3	4	5	6

서버 파일 디스크립터에 읽기 감지

accept()

연결 요청 수락

read()

클라이언트로부터 로그 정보 읽기

SearchListNode()

리스트에서 사용자 정보 찾기

if

print(사용자 로그정보)

회원가입(리스트에 삽입)

채팅방 입장

# 전체 발송, 연결 종료

for

0	0	0	0			
0	1	2	3	4	5	6

클라이언트 파일 디스크립터에 읽기 감지

1

셋 중에 하나의 FD에서 입력을 감지

read(감지된 클라이언트 소켓FD)

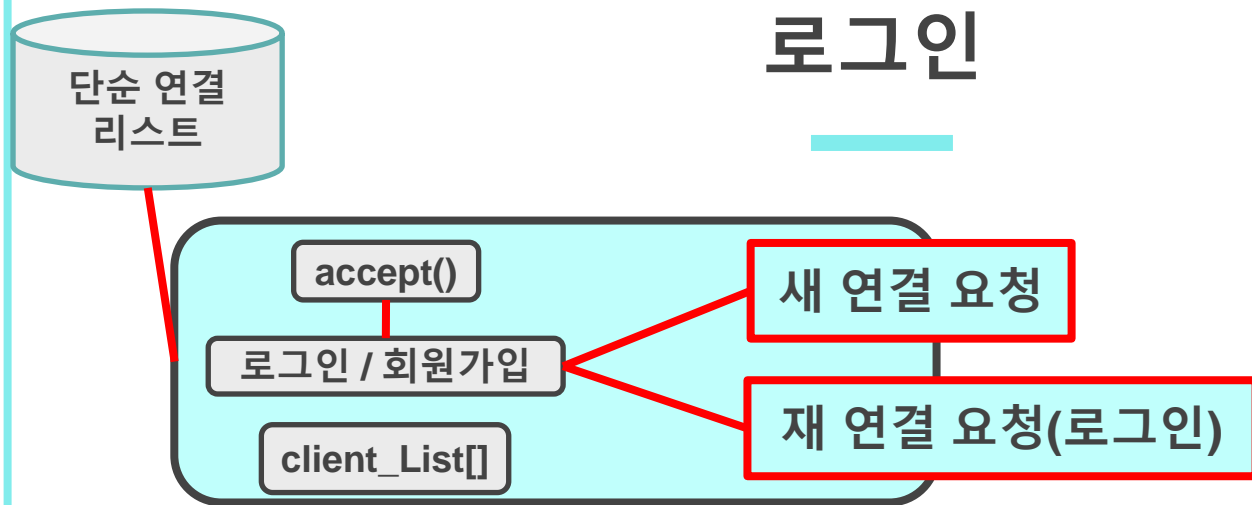
if

str\_len == 0(연결 종료)

else

for(전체 클라이언트로 발송)

# 로그인



- **새로운** 연결 요청 : 새로운 클라이언트의 연결 요청
- **재** 연결 요청 : 기존 클라이언트의 `close()` 이후에 해당 파일디스크립터로 재 연결 요청



# 재 연결 요청

client\_List[]

<del>4</del>	5	6	.	.	.	.
--------------	---	---	---	---	---	---

0

1

2

3

4

5

6

클라이언트 FD를 가진 배열

현재 : num\_chat = 3

4번 FD 연결 종료



4번 파일디스크립터 클라이언트 재요청

client\_List[]

0	5	6	4	.	.	.
---	---	---	---	---	---	---

0

1

2

3

4

5

6

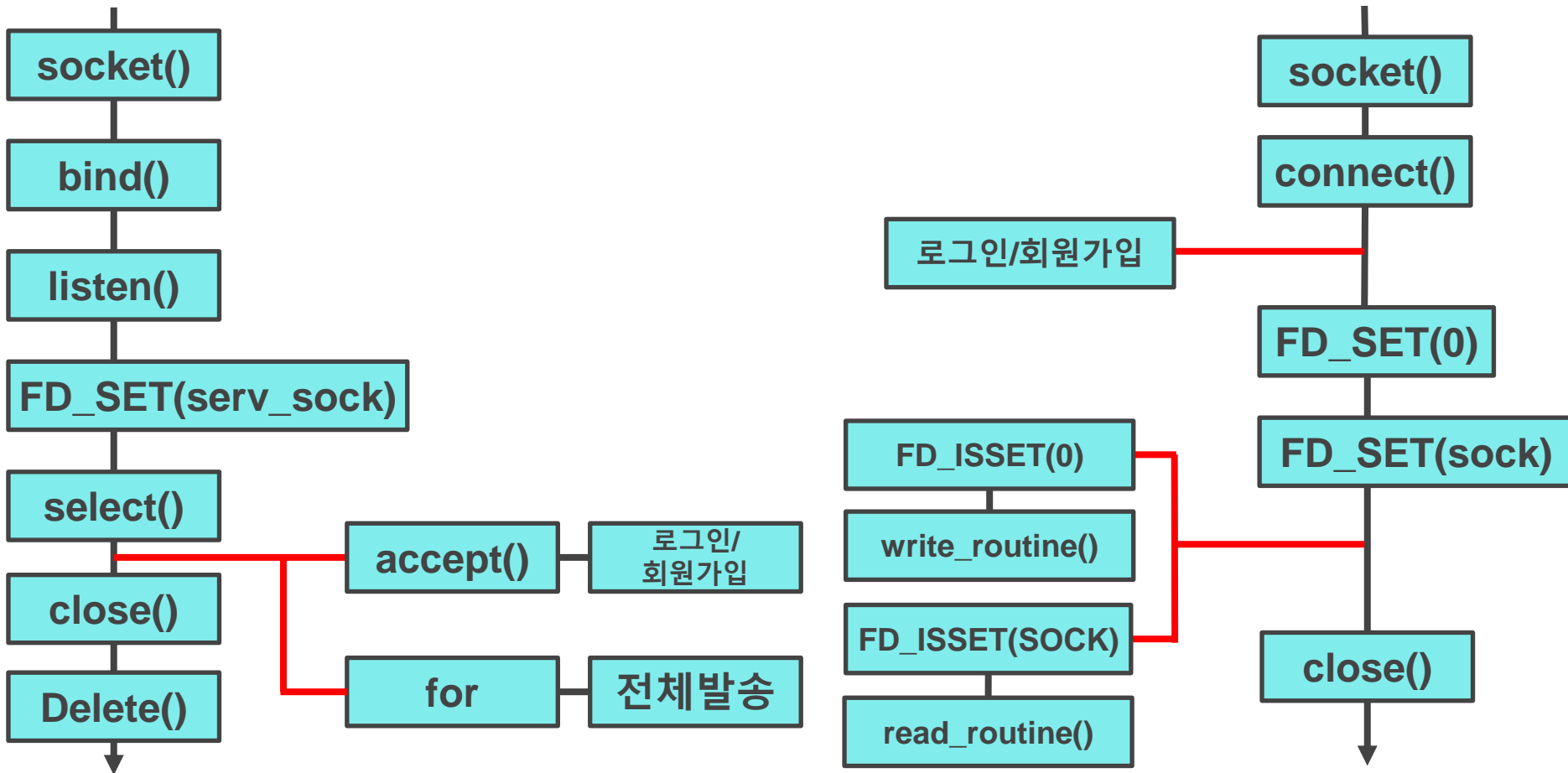


정렬

server.c

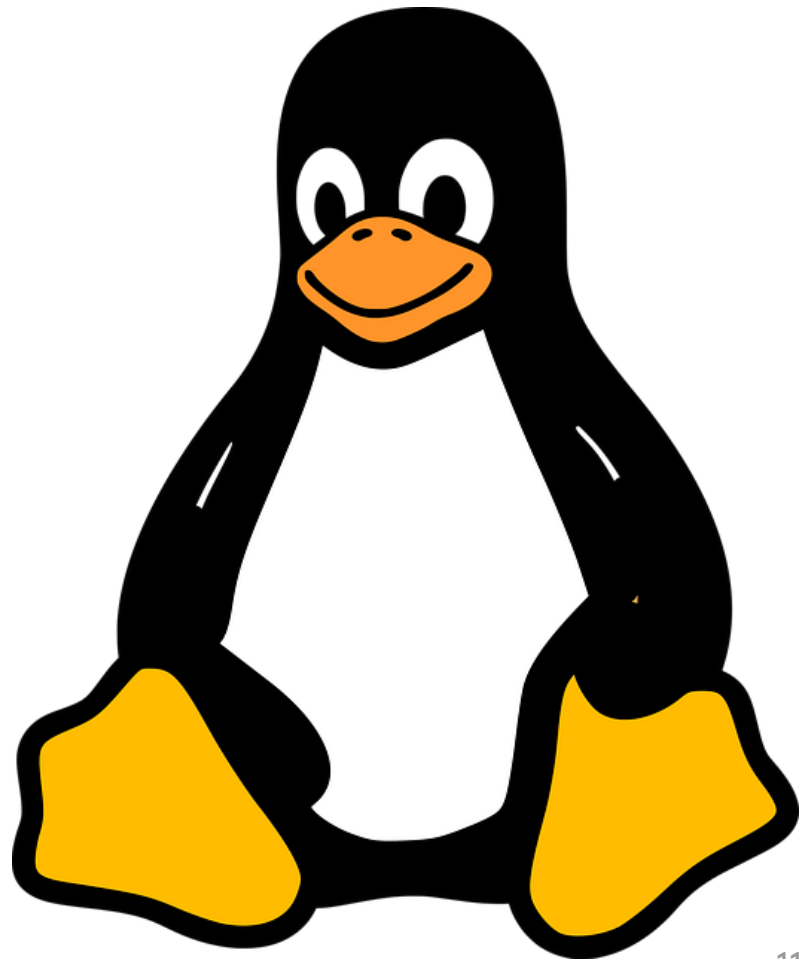
## 전체 흐름

client.c



# 1. 설계 환경

VirtualBox Ubuntu 16.04.6  
유틸리티 : leafpad



# 설계내용

# 클라이언트 1명 접속

```
lee@lee-VirtualBox: ~/NT_Project/presentation
```

```
lee@lee-VirtualBox:~/NT_Project/presentation$ ./serv 9190
```

```
kim | 123
```

```
connected client : 4
```

```
현재 인원 수 : 1
```

```
[[접속된 로그 리스트]]
```

```
kim | 123
```

```
-> FD : 4
```

server

```
lee@lee-VirtualBox: ~/NT_Project/presentation
```

```
lee@lee-VirtualBox:~/NT_Project/presentation$ ./clnt 127.0.0.1 9190 kim
```

```
해당 터미널로 재접속 가능====>'q' 또는 'Q'입력 시 연결 종료
```

```
로그인(1), 회원가입(2)
```

```
2
```

```
id, pass 입력 :
```

```
ex) UserID | Password
```

```
kim | 123
```

```
5. 회원가입 성공, 채팅방 입장
```

client

# 클라이언트 2명 접속

lee@lee-VirtualBox: ~/NT\_Project/presentation

lee@lee-VirtualBox:~/NT\_Project/presentation\$ ./serv 9190

kim | 123

connected client : 4

현재 인원 수 : 1

[[접속된 로그 리스트]]

kim | 123

-> FD : 4

mike | 456

connected client : 5

현재 인원 수 : 2

[[접속된 로그 리스트]]

mike | 456

-> FD : 5

kim | 123

-> FD : 4

]

lee@lee-VirtualBox: ~/NT\_Project/presentation

lee@lee-VirtualBox:~/NT\_Project/presentation\$ ./clnt 127.0.0.1 9190 mike

해당 터미널로 재접속 가능====>'q' 또는 'Q'입력 시 연결 종료

로그인(1), 회원가입(2)

2

id, pass 입력 :

ex) UserID | Password

mike | 456

5. 회원가입 성공, 채팅방 입장

█

또 다른 클라이언트의 로그정보

# 대화

```
lee@lee-VirtualBox: ~/NT_Project/presentation
lee@lee-VirtualBox:~/NT_Project/presentation$ ./clnt 127.0.0.1 9190 kim
해당 터미널로 재접속 가능====>'q' 또는 'Q'입력 시 연결 종료
로그인(1), 회원가입(2)
2
id, pass 입력 :
ex) UserID | Password
kim | 123
5. 회원가입 성공, 채팅방 입장
[mike] hi
hi
[kim] hi
```

```
lee@lee-VirtualBox: ~/NT_Project/presentation
lee@lee-VirtualBox:~/NT_Project/presentation$ ./clnt 127.0.0.1 9190 mike
해당 터미널로 재접속 가능====>'q' 또는 'Q'입력 시 연결 종료
로그인(1), 회원가입(2)
2
id, pass 입력 :
ex) UserID | Password
mike | 456
5. 회원가입 성공, 채팅방 입장
hi
[mike] hi
[kim] hi

```

# 재연결

```
lee@lee-VirtualBox: ~/NT_Project
lee@lee-VirtualBox:~/NT_Project$ ./server 9190
-----현재 참가자 수 : 1명-----
connected client : 4
Register User Log
[[print node list]]
kim 13 | 1234
-> FD : 4
-----현재 참가자 수 : 2명-----
connected client : 5
Register User Log
[[print node list]]
mike123 | 12345
-> FD : 5
kim 13 | 1234
-> FD : 4
-----현재 참가자 수 : 1명-----
closed client : 4
-----현재 참가자 수 : 2명-----
connected client : 4
Login User : kim 13 | 1234
[[print node list]]
mike123 | 12345
-> FD : 5
kim 13 | 1234
-> FD : 4
-----현재 참가자 수 : 1명-----
closed client : 5
-----현재 참가자 수 : 0명-----
closed client : 4
-----현재 참가자 수 : 1명-----
connected client : 4
Login User : kim 13 | 1234
[[print node list]]
mike123 | 12345
-> FD : 5
```

클라이언트 1명 **접속**

클라이언트 1명 추가 **접속**

클라이언트 1명 **연결 종료**

연결 종료했던 클라이언트  
**재접속**

```
kim 13 | 1234
-> FD : 4
-----현재 참가자 수 : 2명-----
connected client : 5
Login User : mike123 | 12345
[[print node list]]
mike123 | 12345
-> FD : 5
kim 13 | 1234
-> FD : 4
```



# 변동 내용

1. 파일함수 -> 자료구조 사용

# 변동 내용

**파일 함수**를 사용하여 사용자 아이디, 비밀번호를 저장

\* 입력 형식 : UserID | Password '0'



**자료구조**를 이용하여 사용자 아이디, 비밀번호를 저장

\* 입력 형식 : UserID | Password '0'

1. 파일 함수를 사용하는 것 보다 리스트를 사용하여 증가하는 회원가입자 수에 대응할 수 있는 서버를 구현
2. 코드가 간결해지고 로그 정보 일치성 검사가 더 빠르다.

# 교훈

1. 서버에서 정보를 요청해보며 서버의 의미를 이해하게 되었다.
2. 사용 목적에 따라 효율적이고 오류에 쉽게 대응 할 수 있는 기술적 방법을 사용해야 한다.
3. 서버에 요청할 클라이언트 수를 예측하여 유동적으로 대응 가능해야 한다.
4. 자신이 사용하고자 하는 프로세싱 기법에 대해 자세히 알아야 한다.