
컴퓨터 네트워크

과제 3 보고서

코코아톡 프로그래밍

공헌도 : 1인 팀

이름	김효성
학번	201400948
전공	중국어언어문화전공
제출일	2018.11.18

목차

- ① 문제 내용과 해결 방안
- ② 설계 내용 및 기능
 - ✓ 서버
 - ✓ 클라이언트
- ③ 프로토콜 설계서

1. 문제 내용과 해결 방안

문제 내용(Problem)과 해결 방안(Solution)

P : 카카오톡 처럼 동작하는 서버와 클라이언트 프로그램을 만든다.

S : 서버와 클라이언트를 설계한다.

P : 메시지를 주고 받는 기능은 반드시 있어야 한다.

S : 닉네임 설정 기능

S : 전체 채팅 기능

S : 귓속말 기능

S : 서버 알림 기능

S : 클라이언트 알림 기능

S : N개의 클라이언트가 접속하면 N개의 클라이언트가 하나의 대화방에 있는 상황이다.

P : 프로토콜 설계를 해야 한다.

S : 서버와 클라이언트 사이의 프로토콜을 설계해야 한다.

S : 서버와 클라이언트는 프로토콜 대로 동작해야 한다.

P : 서버/소켓/쓰레드 포함 어떤 라이브러리/모듈이든 사용하여 더 좋은 서버를 만들어야 한다.

S : 멀티쓰레드를 이용한다.

S : 지속 연결 방식을 이용한다.

2. 설계 내용

I. 서버

➤ 총 두 가지 종류의 스레드로 구동된다.

① 메인 스레드

서버의 소켓에 접속하는 클라이언트를 서버와 연결하고, 클라이언트의 소켓 객체, IP주소 등의 정보를 저장한 뒤, 해당 클라이언트를 담당하는 스레드를 생성한다. 즉, 메인 스레드는 서버와 클라이언트의 연결을 담당한다.

② 클라이언트 스레드

클라이언트가 접속할 때마다 만들어지며 해당 클라이언트와 서버의 통신을 담당한다. 즉, 클라이언트 스레드는 코코아톡을 구동하기 위한 채팅, 알림 등 서버-클라이언트 간 연결을 제외한 모든 기능을 담당한다.

➤ 서버의 기능은 크게 3가지로 구성된다.

1 클라이언트와 통신

1 - 1 response message 송신

- ✓ 모든 클라이언트에게 전체 채팅 송신
- ✓ 귓속말 상대 클라이언트에게 귓속말 송신
- ✓ 모든 클라이언트에게 새로운 클라이언트 채팅방 입장 알림 송신
- ✓ 모든 클라이언트에게 특정 클라이언트 채팅방 퇴장 알림 송신
- ✓ 귓속말 발신 클라이언트에게 귓속말 전달 완료 알림 송신
- ✓ 귓속말 발신 클라이언트에게 귓속말 전달 에러 알림 송신

1 - 2 request message 수신

- ✓ 닉네임 설정 정보 수신
- ✓ 전체 채팅 내용 수신
- ✓ 귓속말 내용 및 수신자 정보 수신

1 - 3 프로토콜을 통한 메시지 해석 및 변환

2 서버 알림

- ✓ 서버 구동 시작 알림
- ✓ 클라이언트 접속 알림
- ✓ 클라이언트 닉네임 설정 알림
- ✓ 현재 접속자 수 알림
- ✓ 현재 접속자 리스트 알림
- ✓ 클라이언트 연결 종료 알림

3 DB 입출력

- ✓ 클라이언트와의 통신을 위한 connection 정보 입출력
- ✓ 클라이언트 닉네임 정보 입출력
- ✓ 귓속말 기능을 위한 클라이언트 포트 번호 정보 입출력

II. 클라이언트

➤ 서버와 마찬가지로 총 두 가지 종류의 쓰레드로 구동된다.

- ① 메인 쓰레드는 서버로부터 response message를 수신하여 프로토콜을 통해 해석한 다음, 해석한 내용을 클라이언트에 출력하는 역할을 담당한다. 또한 서버로 메시지를 송신하는 쓰레드를 생성한다.
- ② 채팅 쓰레드는 메시지 내용 및 상대를 입력받고 해당 메시지를 프로토콜에 맞게 request message로 변환하여 서버로 송신하는 역할을 담당한다.

➤ 클라이언트의 기능은 크게 2가지로 구성된다.

1 서버와 통신

1 - 1 request message 송신

- ✓ 전체 채팅 내용 송신
- ✓ 귓속말 내용 및 수신자 정보 송신
- ✓ 닉네임 설정 정보 송신

1 - 2 response message 수신

- ✓ 전체 채팅 내용 수신
- ✓ 귓속말 내용 및 발신자 정보 수신
- ✓ 귓속말 전달 에러 알림 수신
- ✓ 귓속말 전달 완료 알림 수신
- ✓ 특정 클라이언트 채팅방 입장 알림 수신
- ✓ 특정 클라이언트 채팅방 퇴장 알림 수신

1 - 3 프로토콜을 통한 메시지 해석 및 변환

2 채팅 내용 및 알림 출력

- ✓ 전체 채팅 내용 출력
- ✓ 귓속말 내용 및 발신자 정보 출력
- ✓ 귓속말 전달 에러 알림 출력
- ✓ 귓속말 전달 완료 알림 출력

- ✓ 특정 클라이언트 채팅방 입장 알림 출력
- ✓ 특정 클라이언트 채팅방 퇴장 알림 출력

III. 본 코코아톡 프로그램 성능 및 기능

- 접속자 수에 제한이 없다.

컴퓨터의 자원이 견딜 수 있을 때까지 접속할 수 있다.

- 채팅하는 순서는 상관이 없다.

A가 채팅한 후 B가 채팅할 수 있다 와 같은 제약사항이 없다.

- 클라이언트는 닉네임 설정을 해야한다.

- 클라이언트는 전체 채팅을 할 수 있다.

- 클라이언트는 다른 클라이언트에게 귓속말을 보낼 수 있다.

- 클라이언트가 귓속말을 보내면, 귓속말 성공 여부를 알 수 있다.

- 클라이언트는 다른 클라이언트의 채팅방 입장, 퇴장을 알 수 있다.

- 서버(관리자)는 접속한 클라이언트의 IP, 포트번호를 알 수 있다.

- 서버(관리자)는 접속한 클라이언트가 닉네임을 설정하면 설정한 닉네임을 알 수 있다.

- 서버(관리자)는 클라이언트의 접속 종료를 알 수 있다.

- 서버(관리자)는 현재 접속자 수를 알 수 있다.

- 서버(관리자)는 현재 접속자 리스트를 알 수 있다.

IV.프로토콜 설계서

1. 메시지 유형

본 프로그램의 메시지는 서버가 클라이언트에게 보내는 response message와 클라이언트가 서버에게 보내는 request message로 분류할 수 있다.

Request | Response

2. 메시지 구성

요구 및 응답 메시지는 각각 헤더를 가지고 있으며 하나 또는 그 이상의 필드를 가지고 있고 헤더와 필드를 구분하기 위해 checkmark('@')를 사용한다.

Header@field@field@....

3. 메시지 헤더

헤더는 서버와 클라이언트가 메시지의 용도와 해당 용도에 맞게 메시지를 해석하는 역할을 한다.

4. 메시지 필드

필드는 헤더와 필드의 순서에 따라 의미가 해석되기 때문에, 순서를 변경해서는 안된다.

5. 메시지 해석

서버와 클라이언트는 "받은 메시지.split('@)" 를 통해 checkmark로 구분된 메시지를 리스트로 반환받는다. 리스트 0번 인자를 헤더로 파악하고, 헤더에 맞는 기능을 하도록 필드를 해석한다.

I. Response Message

서버가 클라이언트에 전송하는 메시지이다. 채팅 내용을 전송하는 경우, 클라이언트에게 받은 Request Message를 해석하여 필요한 정보를 첨가, 필요하지 않은 정보는 삭제하여 프로토콜 형식으로 변환 후 클라이언트에게 전송한다. 한편, 알림을 전송하는 경우 각 상황에 맞도록 헤더와 필드를 프로토콜 형식으로 변환하여 클라이언트에게 전송한다.

1) 구성

responseheader + checkmark + field + checkmark + field....

2) 유형

i. 전체 채팅 내용 전송 msgToAll@sender@msg

헤더	의미
Responseheader	'msgToAll'
Sender	채팅을 보낸 접속자의 닉네임이다.
msg	접속자가 타이핑한 채팅 내용 그 자체이다.

클라이언트 대응

헤더가 'msgToAll'이면,

- ① 키보드 입력,알림 내용과 채팅 내용을 구분하기 위해 'wt'을 4번 출력한다.
- ② Sender가 msg라는 전체 채팅을 하였다고 출력한다.

ii. 귓속말 내용 전송 whisper@sender@msg

헤더	의미
Responseheader	'whisper'
Sender	채팅을 보낸 접속자의 닉네임이다.
msg	접속자가 타이핑한 귓속말 내용 그 자체이다.

클라이언트 대응

헤더가 'whisper'이면,

- ① 키보드 입력,알림 내용과 귓속말 내용을 구분하기 위해 'wt'를 4번 출력한다.
- ② sender로부터 msg라는 귓속말이 도착했다고 출력한다.

iii. 채팅방 입장 알림 noticeJoin@justJoined

헤더	의미
Responseheader	'noticeJoin'
justJoined	채팅방에 입장한 접속자의 닉네임이다.

클라이언트 대응

헤더가 'noticeJoin'이면,

- ① 키보드 입력, 채팅 내용과 알림 내용을 구분하기 위해 'wt'를 8번 출력한다.
- ② justJoined 닉네임의 접속자가 채팅방에 입장했다고 출력한다.

iv. 채팅방 퇴장 알림 noticeExit@justExit

헤더	의미
Responseheader	'noticeExit'
justExit	채팅방을 퇴장한 클라이언트의 닉네임이다.

클라이언트 대응

헤더가 'noticeExit'이라면,

- ① 키보드 입력, 채팅 내용과 알림 내용을 구분하기 위해 'wt'를 8번 출력한다.
- ② justExit 닉네임의 접속자가 채팅방에서 퇴장했다고 출력한다.

v. 귓속말 상대 없음 알림 noticeWhisperError@

헤더	의미
Responseheader	'noticeWhisperError'

클라이언트 대응

헤더가 'noticeWhisperError'라면,

- ① 키보드 입력, 채팅 내용과 알림 내용을 구분하기 위해 'wt'를 8번 출력한다.
- ② 입력한 귓속말 상대가 채팅방에 없다고 출력한다.

vi. 귓속말 전달 완료 알림 noticeWhisperSuccess@

헤더	의미
Responseheader	'noticeWhisperSuccess'

클라이언트 대응

헤더가 'noticeWhisperSuccess'라면,

- ① 키보드 입력, 채팅 내용과 알림 내용을 구분하기 위해 'wt'를 8번 출력한다.
- ② 입력한 귓속말이 전달되었다고 출력한다.

II. Request Message

클라이언트가 서버에게 전송하는 메시지이다. 키보드로부터 입력받은 내용을 프로토콜 형식으로 변환하여 서버에게 전송한다.

1) 구성

requestheader + checkmark + field + checkmark + field....

2) 유형

i. 닉네임 설정 정보 전송 nickset@nick

헤더	의미
Requestheader	'nickset'
nick	채팅방을 퇴장한 클라이언트의 닉네임이다.

서버 대응

헤더가 'nickset'이라면,

- ① DB의 클라이언트 닉네임 정보 디렉터리에서, 클라이언트 포트번호를 키로, 메시지의 nick을 밸류로 저장한다.
- ② DB의 소켓번호 디렉터리에서, 메시지의 nick을 키로, 클라이언트 포트번호를 밸류로 저장한다.
- ③ 서버에 클라이언트 IP , 포트번호를 가지고 있는 접속자가 nick이라는 닉네임을 설정했다고 출력한다.
- ④ 모든 클라이언트에게 'noticeJoin' response message를 전송한다.

(response message 중 채팅방 입장 알림 전송 참고)

ii. 전체 채팅 내용 전송 msgToAll@msg

헤더	의미
Requestheader	'msgToAll'
msg	클라이언트가 입력한 전체 채팅 내용이다.

서버 대응

헤더가 'msgToAll'라면,

- ① 모든 클라이언트에게 'msgToAll' response message를 전송한다.

(response message 중 전체 채팅 내용 전송 참고)

iii. 귓속말 내용 전송 whisper@target@msg

헤더	의미
Responseheader	'whisper'
target	클라이언트가 타이핑한 귓속말 전달 상대이다.
msg	클라이언트가 타이핑한 귓속말 내용 그 자체이다.

서버 대응

헤더가 'whisper'이라면,

- ① 서버에서 서버와 클라이언트의 통신을 맡는 쓰레드가 클라이언트마다 돌아가고 있기 때문에, 귓속말을 보낸 클라이언트의 닉네임이 서버에 저장되어 있다.
- ② 이를 이용하여, target에게 'whisper' response message를 전송한다.

(response message 중 귓속말 내용 전송 참고)