

## 컴퓨터네트워크

### 실습 #12 문제 및 보고서

이름	황명원
학번	20185309
소속 학과/대학	콘텐츠 it 전공/정보과학대학
분반	01 (담당교수: 박찬영)

### <주의사항>

- 개별 과제입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)
- **각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.**
  - 소스코드/스크립트 등을 작성한 경우, 해당 파일의 이름도 적어주세요.
- SmartLEAD 제출 데드라인:
  - **6/20 23:55 까지 (4 주간 진행하는 과제입니다)**
  - **데드라인을 지나서 제출하면 0 점**
  - **주말/휴일/학교행사 등으로 인한 데드라인 연장 없음**
  - 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0 점 처리함
- SmartLEAD 에 아래의 파일을 제출해 주세요
  - **보고서(PDF 파일로 변환 후 제출을 권장하나, WORD 로 제출해도 됨)**
  - 보고서 파일명에 이름과 학번을 입력해 주세요.
  - **소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 모든 파일 제출(또는 본 문서에 소스코드 화면 캡처해서 붙여넣기)**
  - **이번 과제는 [데모 시나리오]에 따라서 실행하는 화면을 녹화한 동영상도 첨부해야 합니다.**

### <개요>

**이번 과제는 소켓 프로그래밍을 통한 통신 프로그램을 구현하는 내용으로 구성되어 있습니다.**

**이번 과제 총점은 200 점 입니다.**

**\*\* 보고서, 소스코드, 동영상을 첨부해서 제출하세요**

**\*\* 데모 결과와 본 보고서의 내용을 모두 채점하여 최종 점수가 부여됩니다.**

● **공통 제출물 및 공통 답변**

- 기본적으로, [Q 1], [Q 2]는 반드시 답해야 합니다.
- 본 보고서, 소스코드, 동영상 녹화 파일 모두 제출해야 합니다.
- 서버 터미널 1 개 및 8 개의 클라이언트 터미널이 모두 보이도록 영상을 녹화하세요. 녹화 영상의 해상도를 높게 설정하세요. 녹화 영상에 녹화된 모든 터미널에서 문자가 식별 가능해야 합니다.
- Make clean 을 수행하는 순간부터 녹화하면 됩니다.

● **구현 완료/미완료 여부에 따른 답변**

- 구현이 완료된 후 제출하는 경우, [Q 4]에 답할 필요 없지만 [Q 3]은 답해야 합니다.
- 구현이 완료되지 않은 상태로 제출하는 경우, [Q 3]에 답할 필요 없지만 [Q 4]는 답해야 합니다.

〈실습 과제〉

**[Q 1] 실행 동영상 제출 [배점: 40]**

데모 시나리오에 따라서 실행하는 화면을 녹화한 동영상을 첨부하세요. 동영상을 첨부했나요?

답변 (‘예’ 또는 ‘아니오’로 답하시오):

예

**[Q 2] Makefile [배점: 10]**

Makefile 의 내용을 캡처하여 아래에 첨부하세요

답변:

```
CC = gcc
CFLAGS = -Wall
```

```
all: server client1 client2 client3 client4 input1 input2 input3 input4
```

```
server: server.c
    $(CC) -o server server.c
```

```
client1: client-monitor.c
    $(CC) -DCLIENT1 -o client1 client-monitor.c
```

```
client2: client-monitor.c
    $(CC) -DCLIENT2 -o client2 client-monitor.c
```

```
client3: client-monitor.c
    $(CC) -DCLIENT3 -o client3 client-monitor.c
```

```
client4: client-monitor.c
    $(CC) -DCLIENT4 -o client4 client-monitor.c
```

```
input1: client-input.c
    $(CC) -DCLIENT1 -o input1 client-input.c
```

```
input2: client-input.c
    $(CC) -DCLIENT2 -o input2 client-input.c
```

```
input3: client-input.c
    $(CC) -DCLIENT3 -o input3 client-input.c
```

```
input4: client-input.c
    $(CC) -DCLIENT4 -o input4 client-input.c
```

```
clean:
    rm -f server client1 client2 client3 client4 input1 input2 input3 input4
```



**[Q 3] 구현 완료: 동시 동작, 비동기, 다중 채팅 프로그램 구현하기 [배점: 150]**

실습 12 강의자료 중 [실습과제 소개/설명] 문서를 참고하여 동시 동작, 비동기, 다중 채팅 프로그램을 구현하세요. 채팅 프로그램은 아래의 요건을 만족해야 합니다.

- 동시 동작 : 다수의 클라이언트 및 다수의 채팅방을 동시에 서비스 할 수 있어야 함
- 비동기 채팅 : 사용자는 언제든지 메시지를 입력할 수 있어야 함
- 다중 채팅 : 하나의 채팅방에 다수의 사용자가 참여할 수 있어야 함

[문제] 데모 시나리오에 따라 프로그램을 실행하고 화면 전체를 녹화 후 첨부하세요. **데모 시나리오를 모두 수행한 후,**

- 1) 서버 프로그램의 터미널 화면을 캡처하여 아래에 첨부하세요. (가장 마지막 메시지 일부만 캡처해도 괜찮습니다.)
- 2) 클라이언트 1 번의 모니터 터미널을 캡처하여 아래에 첨부하세요. (가장 마지막 메시지 일부만 캡처해도 괜찮습니다.)
- 3) 클라이언트 2 번의 모니터 터미널을 캡처하여 아래에 첨부하세요. (가장 마지막 메시지 일부만 캡처해도 괜찮습니다.)
- 4) 클라이언트 3 번의 모니터 터미널을 캡처하여 아래에 첨부하세요. (가장 마지막 메시지 일부만 캡처해도 괜찮습니다.)
- 5) 클라이언트 4 번의 모니터 터미널을 캡처하여 아래에 첨부하세요. (가장 마지막 메시지 일부만 캡처해도 괜찮습니다.)

소스 코드를 하나의 압축파일로 압축하고, SmartLEAD 에 보고서 및 압축파일을 함께 업로드 하세요.

## 답변 1 (서버 터미널 화면 캡처):

```

[CH.0] 사용자 2 메시지 : hi
[CH.0] 사용자 1 메시지 : hi
사용자 3가 1번 채팅방에 접속합니다.
사용자 4가 1번 채팅방에 접속합니다.
[CH.1] 사용자 3 메시지 : hi
[CH.1] 사용자 4 메시지 : hi~
[CH.1] 사용자 3를 채팅방에서 제거합니다.
사용자 3가 0번 채팅방에 접속합니다.
[CH.0] 사용자 1 메시지 : hi (1)
[CH.0] 사용자 2 메시지 : hi(2)
[CH.0] 사용자 3 메시지 : hi(3)
[CH.1] 사용자 4를 채팅방에서 제거합니다.
[CH.0] 사용자 1를 채팅방에서 제거합니다.
[CH.0] 사용자 2를 채팅방에서 제거합니다.
[CH.0] 사용자 3를 채팅방에서 제거합니다.
321^C
(시그널 핸들러) 마무리 작업 시작!

```

## 답변 2 (클라이언트 1 터미널 화면 캡처):

```

(0을 입력하면 메뉴가 다시 표시됩니다.)
<ChatRoom info>
[ID: 0] Chatroom-0 (0/5)
[ID: 1] Chatroom-1 (0/5)
[ID: 2] Chatroom-2 (0/5)

<MENU>
1. Chat Room List
2. Join a Chat Room (Usage: 2 <room number>)
3. Exit
(0을 입력하면 메뉴가 다시 표시됩니다.)
<MENU>
1. Chat Room List
2. Join a Chat Room (Usage: 2 <room number>)
3. Exit
(0을 입력하면 메뉴가 다시 표시됩니다.)

```

**답변 3** (클라이언트 2 터미널 화면 캡처):

```
(0을 입력하면 메뉴가 다시 표시됩니다.)  
<ChatRoom info>  
[ID: 0] Chatroom-0 (0/5)  
[ID: 1] Chatroom-1 (0/5)  
[ID: 2] Chatroom-2 (0/5)  
  
<MENU>  
1. Chat Room List  
2. Join a Chat Room (Usage: 2 <room number>)  
3. Exit  
(0을 입력하면 메뉴가 다시 표시됩니다.)  
<MENU>  
1. Chat Room List  
2. Join a Chat Room (Usage: 2 <room number>)  
3. Exit  
(0을 입력하면 메뉴가 다시 표시됩니다.)
```

**답변 4** (클라이언트 3 터미널 화면 캡처):

```
(0을 입력하면 메뉴가 다시 표시됩니다.)  
<ChatRoom info>  
[ID: 0] Chatroom-0 (2/5)  
[ID: 1] Chatroom-1 (1/5)  
[ID: 2] Chatroom-2 (0/5)  
  
<MENU>  
1. Chat Room List  
2. Join a Chat Room (Usage: 2 <room number>)  
3. Exit  
(0을 입력하면 메뉴가 다시 표시됩니다.)  
<MENU>  
1. Chat Room List  
2. Join a Chat Room (Usage: 2 <room number>)  
3. Exit  
(0을 입력하면 메뉴가 다시 표시됩니다.)
```



답변 5 (클라이언트 4 터미널 화면 캡처):

```
3. Exit
(0을 입력하면 메뉴가 다시 표시됩니다.)
<MENU>
1. Chat Room List
2. Join a Chat Room (Usage: 2 <room number>)
3. Exit
(0을 입력하면 메뉴가 다시 표시됩니다.)
<ChatRoom info>
[ID: 0] Chatroom-0 (3/5)
[ID: 1] Chatroom-1 (0/5)
[ID: 2] Chatroom-2 (0/5)

<MENU>
1. Chat Room List
2. Join a Chat Room (Usage: 2 <room number>)
3. Exit
(0을 입력하면 메뉴가 다시 표시됩니다.)
```

**[Q 4] 구현 미완료: 구현 상황 보고 [배점: 50 점]**

데모 시나리오 중에서 정상적으로 실행되는 부분 및 그렇지 않은 부분을 아래의 테이블에 체크하세요.  
소스 코드를 하나의 압축파일로 압축하고, SmartLEAD 에 보고서, 압축파일, **녹화 동영상**을 함께 업로드 하세요. (데모 시나리오를 따라서, 정상 동작 하는 부분까지 녹화한 영상을 첨부하세요)

**답변)** 데모 시나리오에서 정상적으로 실행되는 부분 및 그렇지 않은 부분 체크하기:

데모 시나리오	구현 완료 및 정상 실행됨	구현 미완료 또는 실행 시 오류 발생
1 소스코드 빌드 (Makefile)	○	
2.1 서버 시작	○	
2.2 클라이언트 시작	○	
2.3 클라이언트가 서버에 접속하고, 접속 직후 대기실로 이동	○	
3.1 클라이언트 1 번이 채팅방 목록 조회	○	
3.2 클라이언트 1 번이 0 번 채팅방 참가	○	
4.1 클라이언트 2 번이 채팅방 목록 조회	○	
4.2 클라이언트 2 번이 0 번 채팅방 참가	○	
5 클라이언트 1 번과 2 번이 서로 채팅	○	
6.1 클라이언트 3 번이 채팅방 목록 조회	○	
6.2 클라이언트 3 번이 1 번 채팅방 참가	○	
7.1 클라이언트 4 번이 채팅방 목록 조회	○	
7.2 클라이언트 4 번이 1 번 채팅방 참가	○	
8 클라이언트 3 번과 4 번이 서로 채팅	○	
9.1 클라이언트 3 번이 채팅방 1 번 탈퇴하고 대기실로 이동	○	
9.2 클라이언트 3 번이 대기실에서 채팅방 목록 조회	○	
9.3 클라이언트 3 번이 0 번 채팅방에 참가	○	
10 클라이언트 1, 2, 3 번이 서로 채팅	○	
11.1 클라이언트 4 번이 채팅방 1 번 탈퇴하고 대기실로 이동	○	
11.2 클라이언트 4 번이 대기실에서 채팅방 목록 조회	○	

12.1 클라이언트 1,2,3 번은 채팅방에서 탈퇴하고 대기실로 이동	O	
12.2 모든 클라이언트는 채팅 프로그램을 종료	O	
13 서버 터미널에서 [Ctrl+C]를 입력하면 시그널 핸들러가 호출되고, 지정된 종료 절차를 수행하고, 서버 프로그램이 종료됨	O	

끝! 수고하셨습니다 ☺