

컴퓨터네트워크

실습 #11 문제 및 보고서

| | |
|-------------|------------------|
| 이름 | 황명원 |
| 학번 | 20185309 |
| 소속 학과/대학 | 정보과학대학/콘텐츠 it 전공 |
| 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:
 - **다음 다음 실습시간 전날 23:55 까지 (2 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)**

<개요>

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

**** 전체 소스코드를 압축하여 첨부파일로 제출하세요.**

<실습 과제>

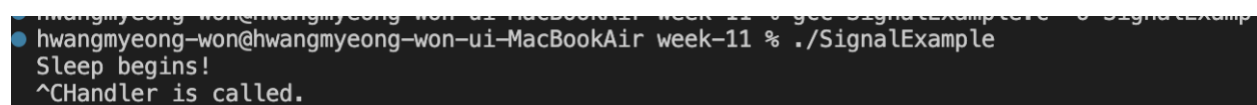
[Q 1] CTRL+C 시그널 핸들러 [배점: 30]

아래와 같이 동작하는 SignalExample.c 프로그램을 작성하세요. 아래의 코드는 fork/thread 등을 사용하지 않습니다.

```
main {  
    SIGINT 시그널 핸들러를 handler 함수로 등록  
    "Sleep begins!" 문자열 출력  
    1000 초 동안 sleep  
    "Wake up!" 문자열 출력  
}  
handler {  
    "Handler is called." 문자열 출력  
    exit(EXIT_SUCCESS) 호출  
}
```

프로그램을 실행하고, 1000 초가 만료되기 전에 CTRL+C 를 입력하세요. 터미널 출력 화면을 캡처하여 아래에 첨부하세요. 소스코드도 제출해야 합니다.

답변:



```
hwangmyeong-won@hwangmyeong-won-ui-MacBookAir week-11 % ./SignalExample  
Sleep begins!  
^CHandler is called.
```

[Q 2] 동시 동작 서버 : Echo 서버 [배점: 35]

하나의 서버에 여러 사용자가 접속하여 Echo 서버 기능을 사용할 수 있는 프로그램을 작성하세요. SOCK_STREAM 소켓 및 AF_INET 연결을 사용하세요. 클라이언트는 단일 터미널을 사용해서 메시지를 입력하고, 서버로부터 받은 메시지를 표시합니다.

하나의 서버가 다수의 클라이언트를 동시에 서비스 할 수 있도록, 동시 동작 서버로 구현하세요 (즉, fork 를 사용하여, 각 클라이언트 별로 자식 프로세스를 1 개씩 할당). Echo 서버는 클라이언트가 전송한 메시지를 그대로 클라이언트에게 돌려주는 방식으로 동작하는 방식으로 동작합니다. 클라이언트가 quit 메시지를 입력하면, 해당 클라이언트와의 접속이 종료되도록 구현하세요.

서버는 접속한 클라이언트 수를 `int client_cnt` 변수에 저장합니다. 새로운 클라이언트가 접속하면, `client_cnt` 변수를 증가하고, 다음과 같이 출력합니다: `printf("Number of service client : %d\n", client_cnt);`

*** 기존에 처리 중인 클라이언트의 서비스가 종료되는 경우에 `client_cnt` 변수를 감소시키는 부분은 구현할 필요 없습니다.*

[Task] 문서 하단의 <참고 1>에 작성된 것과 동일한 순서로 프로그램을 실행하고, 동일한 순서로, 동일한 메시지를 입력하세요. <참고 1> 과 같이 터미널 화면을 캡처하여 아래에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

서버 코드 실행 결과:

```

hwangmyeong-won@hwangmyeong-won-ui-
○ MacBookAir week-11 % ./s
Server started. Waiting for connect
ions...
New Client!
Number of service client: 1
Recv from client: hello from client
1
New Client!
Number of service client: 2
Recv from client: hello from client
2
New Client!
Number of service client: 3
Recv from client: hello from client
3
Recv from client: nice to meet you
from client 2
Recv from client: good to see you c
lient 1

```

클라이언트 코드 실행 결과:

| | |
|---|--|
| <pre> hwangmyeong-won@hwangmyeong-won-ui- ○ MacBookAir week-11 % ./c Enter messages:hello from client 1 Recv from server: hello from client 1 Enter messages:good to see you clie nt 1 Recv from server: good to see you c lient 1 Enter messages: </pre> | <pre> hwangmyeong-won@hwangmyeong-won-ui- ○ MacBookAir week-11 % ./c Enter messages:hello from client 2 Recv from server: hello from client 2 Enter messages:nice to meet you fro m client 2 Recv from server: nice to meet you from client 2 Enter messages: </pre> |
|---|--|

```

hwangmyeong-won@hwangmyeong-won-u
○ i-MacBookAir week-11 % ./c
Enter messages:hello from client
3
Recv from server: hello from clie
nt 3
Enter messages:

```

[Q 3] select 를 사용하여 다중 채팅 프로그램 작성하기 [배점: 35]

강의자료 [p. 24, p. 25] 에 표시된 select 함수를 사용하는 예시 프로그램이 완벽히 동작할 수 있도록 코드를 작성하세요.

프로그램을 실행하고, [p. 26] 과 같이 3 개의 클라이언트가 동시에 접속하는 상황을 재현하세요.

[p. 26]과 같이 각 클라이언트에서 1 회씩 메시지를 서버로 전송하세요.

[p. 26]과 같이 터미널 화면을 한번에 캡처하여 아래에 첨부하세요. 소스코드도 제출해야 합니다.

** 참고: 메시지 출력 형식(sprintf 에서 출력하는 내용) 및 클라이언트가 서버로 전송하는 메시지를 [p. 26]과 달라도 됩니다.

답변:

서버 코드 실행 결과:

```
hwangmyeong-won@hwangmyeong-won-ui-MacBoo
kAir week-11 % ./s3
Client 1 connected
Client 2 connected
Client 3 connected
Waiting at select...
select returned: 1
MSG from client 1: hi~ from cli-1

Waiting at select...
select returned: 1
MSG from client 2: hello from cli-2

Waiting at select...
select returned: 1
MSG from client 3: wow from cli-3

Waiting at select...
□
```

클라이언트 코드 실행결과:

| | | |
|---|---|---|
| <pre>hwangmyeong-won@hwangmyeong-won-ui-MacBoo ○ kAir week-11 % ./c3 Enter: hi~ from cli-1 Enter: █</pre> | <pre>hwangmyeong-won@hwangmyeong-won-ui-MacBoo ○ kAir week-11 % ./c3 Enter: hello from cli-2 Enter: █</pre> | <pre>hwangmyeong-won@hwangmyeong-won-ui-MacBo ○ okAir week-11 % ./c3 Enter: wow from cli-3 Enter: █</pre> |
|---|---|---|

<참고 1> [Q 1]번 문제 결과화면 예시

** 참고: 서버 프로그램(s)실행 시, 인자로 정수값을 주는 부분(예: ./s 3) 및 이에 대한 처리는 하지 않아도 됩니다.

```
daniel@danpc1: ~/comnet/sock_prog/4-multiple-echo
daniel@danpc1:~/comnet/sock_prog/4-multiple-echo$ ./s
** USAGE : ./a.out <max num of clients>
daniel@danpc1:~/comnet/sock_prog/4-multiple-echo$ ./s 3
New Client!
Number of service client : 1
Recv from Client: hello from client 1
New Client!
Number of service client : 2
Recv from Client: hello from client 2
New Client!
Number of service client : 3
Recv from Client: hello from client 3
Recv from Client: nice to meet you from client 2
Recv from Client: good to see you from client 1
█
```

```
daniel@danpc1: ~/comnet/sock_prog/4-multiple-echo
daniel@danpc1:~/comnet/sock_prog/4-multiple-echo$ ./c
Enter message: hello from client 1
Recv from server:
hello from client 1
Enter message: good to see you from client 1
Recv from server:
good to see you from client 1
Enter message: █
daniel@danpc1:~/comnet/sock_prog/4-multiple-echo$ ./c
Enter message: hello from client 2
Recv from server:
hello from client 2
Enter message: nice to meet you from client 2
Recv from server:
nice to meet you from client 2
Enter message: █
daniel@danpc1:~/comnet/sock_prog/4-multiple-echo$ ./c
Enter message: hello from client 3
Recv from server:
hello from client 3
Enter message: █
```

끝! 수고하셨습니다 ☺