

[506489] 시스템프로그래밍**실습 #09 문제 및 보고서**

이름	곽영주
학번	20175105
소속 학과/대학	빅데이터
분반	01 (담당 교수: 김태운)

<주의사항>

- 개별 과제 입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)

- 각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.

- 소스코드/스크립트 등을 작성 한 경우, 해당 파일의 이름도 적어주세요.

- SmartLEAD 제출 데드라인:

- 월요일 분반: 다음 실습 시간 전날까지(일요일 까지)

- 수요일 분반: 다음 실습 시간 전날까지 (화요일 까지)

- 데드라인을 지나서 제출하면 24 시간 단위로 20%씩 감점(5 일 경과 시 0 점)

- 주말/휴일/학교행사 등 모든 날짜 카운트 함

- 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0 점 처리함

- 예외 없음

- SmartLEAD 에 아래의 파일을 제출해 주세요

- 보고서(PDF 파일로 변환 후 제출 권장하나, WORD 형식으로 제출도 가능)

- 보고서 파일명에 이름과 학번을 입력해 주세요.

- 소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 모든 파일 제출 (zip 파일로 압축하여 하나의 파일로 제출)

<개요>

이번 과제는 소켓 프로그래밍의 기본적인 기능을 사용하는 내용입니다.

<실습 과제>

[Q 0] 요약 [배점: 10]

이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지 2~3 문장으로 요약하세요.

답변: 이번 과제를 통해 클라이언트와 서버 간 데이터를 주고받을 때, 사소한 오류를 해결하면서 소켓 프로그래밍에 대한 개념을 잡을 수 있었습니다. 또한, Port 와 IP 를 어떠한 방식으로 검색, 조회할 수 있는지도 알게 되었습니다.

[Q 1] 포트 번호로 서비스 명 조회 [배점: 15]

잘 알려진 포트 번호를 입력 받아 이에 해당하는 서비스 명을 출력하는 getServiceNameByPortNumber.c 프로그램을 작성하시오. 사용자 입력을 받는 방법은 명령 행 인자로 구현해도 되고, 프로그램 구동 중에 터미널 입력을 받는 방식을 사용해도 됨.

[Task 1] 잘 알려진 포트번호 3 개를 나열하시오. 예) HTTP(80)

[Task 2] Task 1 에서 답변한 포트번호 3 개를 조회한 후, 터미널 출력 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변 1)

1. SSH(22)
2. HTTPS(443)
3. DNS(53)

답변 2)

```
yeongju@vm-ubuntu20:~/sp2021-2/w09/q01$ ./getServiceNameByPortNumber
Enter to Port Number: 22
Port(22) Name: ssh
yeongju@vm-ubuntu20:~/sp2021-2/w09/q01$ ./getServiceNameByPortNumber
Enter to Port Number: 443
Port(443) Name: https
yeongju@vm-ubuntu20:~/sp2021-2/w09/q01$ ./getServiceNameByPortNumber
Enter to Port Number: 53
Port(53) Name: domain
```

[Q 2] 호스트 파일 조회 [배점: 15]

/etc/hosts 파일에 특정 호스트 명이 있는지 확인하고, 있으면 해당 IP 주소를 출력하고, 없으면 “Not Found”를 출력하는 localDNS.c 프로그램(입력: 호스트 명, 출력: IP 주소)을 작성하시오.

[Task 1] /etc/hosts 파일의 내용을 캡처해서 첨부 하시오.

[Task 2] 파일에 기록된 호스트 명과, 그렇지 않은 호스트 명을 각각 1 개씩을 입력으로 해서 프로그램을 구동 하시오. 입력을 처리하는 방법은 명령 행 인자 또는 프로그램 구동 중 터미널 입력 중 하나를 선택하세요. 터미널 출력 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변 [Task 1]:

```
yeongju@vm-ubuntu20:~/sp2021-2/w09/q02$ cat /etc/hosts
127.0.0.1        localhost
127.0.1.1        yeongju-VirtualBox

# The following lines are desirable for IPv6 capable hosts
::1             ip6-localhost ip6-loopback
fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
```

답변 [Task 2]:

```
yeongju@vm-ubuntu20:~/sp2021-2/w09/q02$ ./localDNS localhost
Name: localhost, IP: 127.0.0.1
yeongju@vm-ubuntu20:~/sp2021-2/w09/q02$ ./localDNS yeongju
Not Found
```

[Q 3] 서버-클라이언트 통신 [배점: 20]

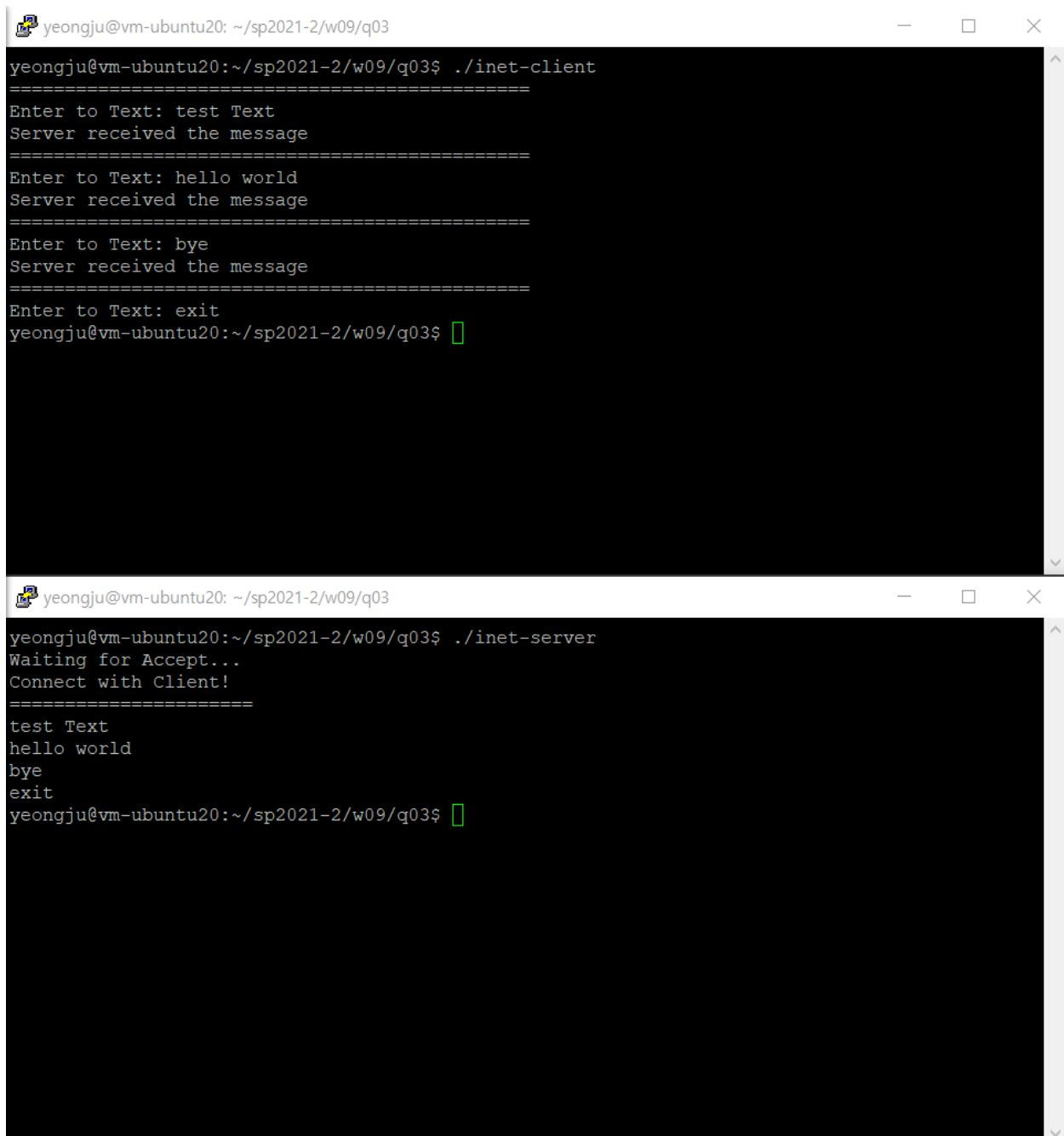
아래와 같이 서버-클라이언트 프로그램을 작성하세요: (AF_INET 소켓 사용)

(파일명: inet-client.c, inet-server.c)

- 클라이언트 프로그램은 텍스트 메시지를 사용자로부터 입력 받습니다. 입력이 완료되면 (= 사용자가 엔터 키 입력) 입력 받은 메시지를 서버로 전송합니다.
- 서버는 메시지를 수신하면 수신한 메시지를 자신의 터미널에 출력하고, “Server received the message” 라는 메시지를 클라이언트로 전송합니다.
- 클라이언트 프로그램은 서버로부터 전달 받은 메시지를 터미널에 출력합니다. 클라이언트 프로그램은 사용자가 'exit' 를 입력하기 전에는 종료하지 않습니다.

[Task] 서버/클라이언트를 구동하고, 클라이언트에서 3 개 이상의 메시지를 서버로 전송하세요. 서버와 클라이언트 터미널 출력 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변 [Task]:



```
yeongju@vm-ubuntu20: ~/sp2021-2/w09/q03
yeongju@vm-ubuntu20:~/sp2021-2/w09/q03$ ./inet-client
=====
Enter to Text: test Text
Server received the message
=====
Enter to Text: hello world
Server received the message
=====
Enter to Text: bye
Server received the message
=====
Enter to Text: exit
yeongju@vm-ubuntu20:~/sp2021-2/w09/q03$ █

yeongju@vm-ubuntu20: ~/sp2021-2/w09/q03
yeongju@vm-ubuntu20:~/sp2021-2/w09/q03$ ./inet-server
Waiting for Accept...
Connect with Client!
=====
test Text
hello world
bye
exit
yeongju@vm-ubuntu20:~/sp2021-2/w09/q03$ █
```

[Q 4] 서버-클라이언트 파일 조회 [배점: 20]

아래와 같이 동작하는 서버-클라이언트 프로그램을 작성하세요: AF_UNIX 소켓을 사용하고,

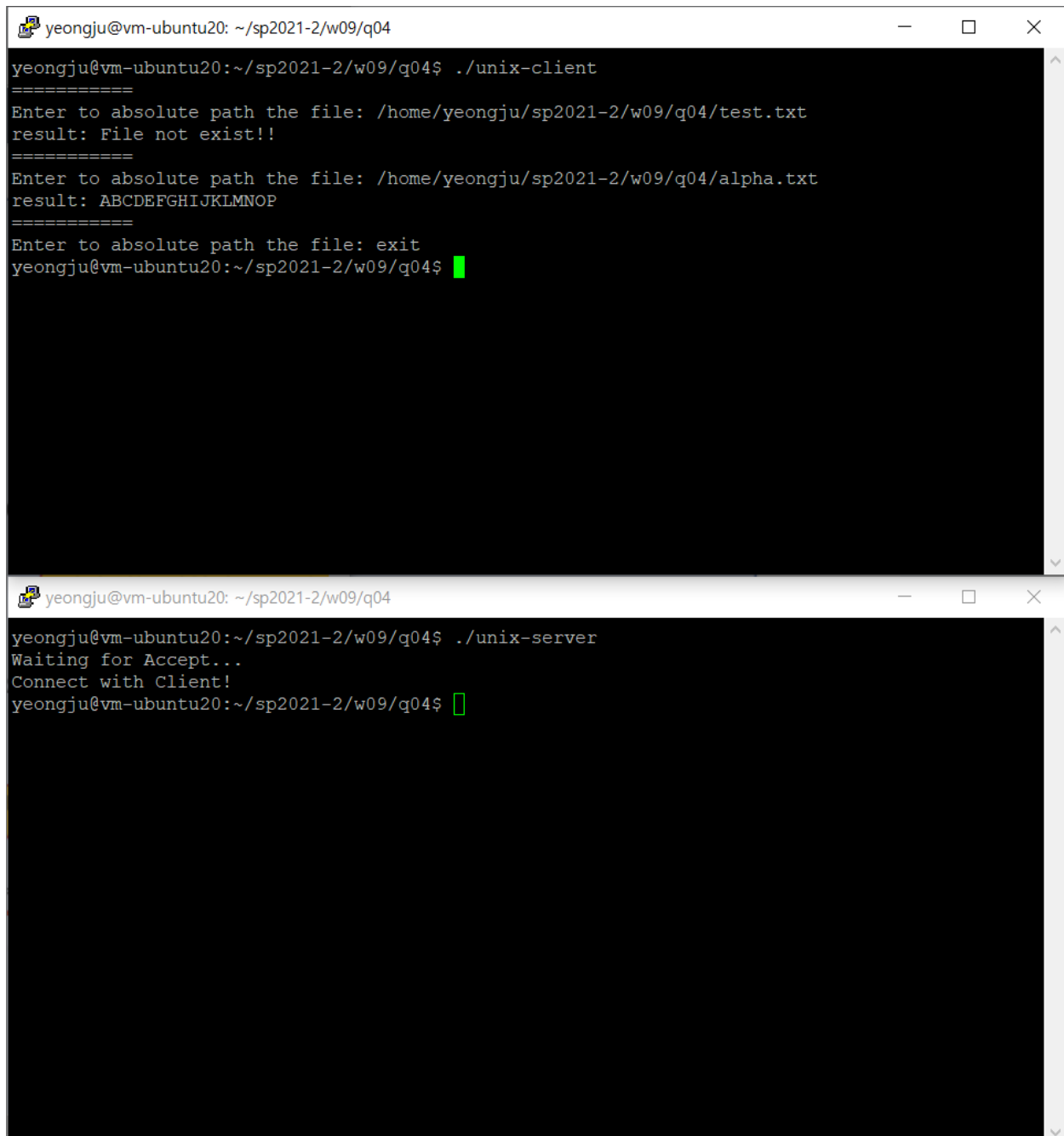
서버와 클라이언트는 같은 컴퓨터/머신에서 동작합니다.

(파일명: 클라이언트 프로그램 `unix-client.c`, 서버 프로그램 `unix-server.c`)

클라이언트는 파일명(+경로)를 서버로 전달하고, 서버는 해당 파일을 열고 파일 내용을 전송하는 프로그램을 작성하시오. 클라이언트 프로그램은 절대 경로로 표현된 파일의 경로를 서버로 전송합니다. 서버는 해당 파일이 존재하지 않으면 오류 메시지(“File not exist!!”)를 클라이언트로 전송하고, 만약 파일이 존재하면 파일의 처음부터 16 바이트 만 전송합니다. 파일의 길이가 16 바이트보다 짧으면 파일의 길이만큼만 전송합니다. 클라이언트는 서버로부터 받은 메시지를 화면에 출력합니다.

[Task] 서버/클라이언트를 구동하고, 클라이언트에서 1 개의 존재하는 파일명+경로와, 1 개의 존재하지않는 파일명+경로를 서버로 전달하세요. 서버와 클라이언트 터미널 출력 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다. 존재하는 파일은 `alpha.txt` 라는 이름의 텍스트 파일을 생성하여 사용하고, 파일 내에 A 부터 Z 까지 알파벳을 입력하고 저장하세요.

답변 [Task]:



```
yeongju@vm-ubuntu20: ~/sp2021-2/w09/q04
yeongju@vm-ubuntu20:~/sp2021-2/w09/q04$ ./unix-client
=====
Enter to absolute path the file: /home/yeongju/sp2021-2/w09/q04/test.txt
result: File not exist!!
=====
Enter to absolute path the file: /home/yeongju/sp2021-2/w09/q04/alpha.txt
result: ABCDEFGHIJKLMNOP
=====
Enter to absolute path the file: exit
yeongju@vm-ubuntu20:~/sp2021-2/w09/q04$ █

yeongju@vm-ubuntu20: ~/sp2021-2/w09/q04
yeongju@vm-ubuntu20:~/sp2021-2/w09/q04$ ./unix-server
Waiting for Accept...
Connect with Client!
yeongju@vm-ubuntu20:~/sp2021-2/w09/q04$ █
```

[Q 5] ncurses 예제 1 [배점: 10]

첨부된 window-browsing.c 파일을 컴파일 하고 구동하세요. 컴파일 명령은:


```
$gcc -o window-browsing.exe window-browsing.c -lncurses -lpanel
```

탭 키를 눌러서 서로 다른 패널간 이동을 해 보세요. 터미널 출력 화면을 캡처하고 본 문서에 첨부하세요.

**** 소스코드:** <https://github.com/overegoz/sysprog/tree/master/w09-lab>

답변:



[Q 6] ncurses 예제 2 [배점: 10]

첨부된 options.c 파일을 컴파일 하고 구동하세요. 컴파일 명령은:

```
$gcc -o options.exe options.c -lncurses -lmenu
```

방향키를 눌러서 메뉴간 이동하고, 엔터키를 눌러서 선택된 메뉴를 확인하세요. 터미널 출력 화면을 캡처하고 본 문서에 첨부하세요.

**** 소스코드:** <https://github.com/overegoz/sysprog/tree/master/w09-lab>

답변:

```
yeongju@vm-ubuntu20: ~/sp2021-2/w09/q06
Choice 1 Choice 1
Choice 2 Choice 2
Choice 3 Choice 3
Choice 4 Choice 4
Choice 5 Choice 5
Choice 6 Choice 6
Choice 7 Choice 7
Exit      Exit

Item selected is : Choice 2

Press <ENTER> to see the option selected
Up and Down arrow keys to naviage (q to Exit)
```

끝! 수고하셨습니다 ☺