

컴퓨터네트워크

과제 #01 보고서

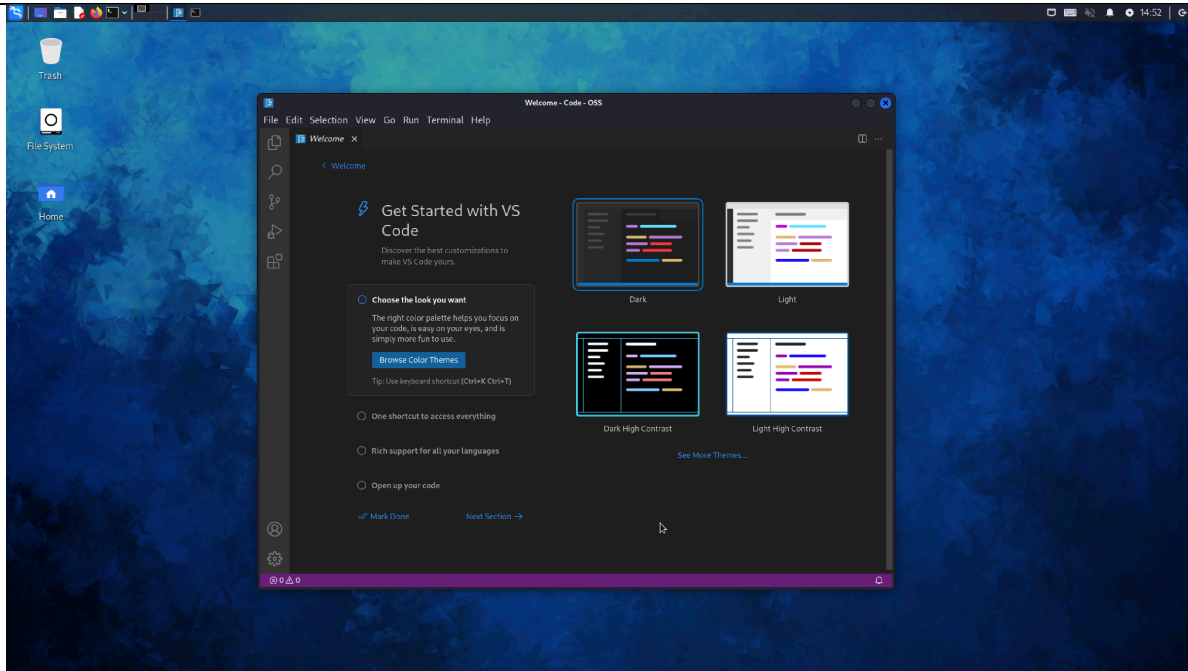
이름	김대욱
학번	202255513
소속 학과/대학	정보의생명공학대학 정보컴퓨터공학부
분반	061

<과제>

[Q 1] C 코딩 [배점: 30]

가상머신 리눅스에서 Visual Studio Code 를 사용하여 간단한 C 프로그램을 작성하는 과제입니다.

[TODO 1] 가상머신 리눅스에서 VS Code 를 설치하고, 실행 화면을 캡처하여 아래에 첨부하세요



[TODO 2] rand 함수를 사용하여 무작위로 생성된 0~99 사이의 정수(0 과 99 포함)를 5 번 printf(“%d\n”); 하는 out.c 프로그램을 작성하세요. 프로그램을 실행하면, 실행 결과가 터미널에 출력됩니다. 소스코드를 수정하지 않고 실행 결과를 data.txt 파일에 저장하게 하려면, 터미널에서 어떤 명령을 입력해야 하나요? 직접 실행하고, 터미널 화면(입력한 명령어가 보이도록 화면 캡처) 및 \$ cat data.txt 실행 결과(터미널 출력)를 아래에 첨부하세요.

```
(nozerose@nozerose) - [~/Documents/HW1/Q1_TODO2]
$ gcc out.c

(nozerose@nozerose) - [~/Documents/HW1/Q1_TODO2]
$ ./a.out > data.txt

(nozerose@nozerose) - [~/Documents/HW1/Q1_TODO2]
$ cat data.txt
78
71
65
59
66
```

[TODO 3] 다섯 번 scanf 를 실행하여 int 형 정수 5 개를 읽고 총합을 출력하는 in.c 프로그램을 작성하세요: printf("SUM: %d\n", sum); 프로그램을 실행하면, 터미널에서 5 개의 숫자를 입력해야 합니다. 소스코드를 수정하지 않고, data.txt 파일의 내용을 scanf 의 입력으로 전달하려면, 터미널에서 어떤 명령을 입력해야 하나요? 직접 실행하고, 터미널 화면(입력한 명령어가 보이도록 화면 캡처 + 총합이 출력된 화면)을 아래에 첨부하세요.

```
(nozerose@nozerose) - [~/Documents/HW1/Q1_TODO3]
$ gcc in.c

(nozerose@nozerose) - [~/Documents/HW1/Q1_TODO3]
$ ./a.out < data.txt
SUM: 339
```

[Q 2] 네트워크 도구 [배점:30]

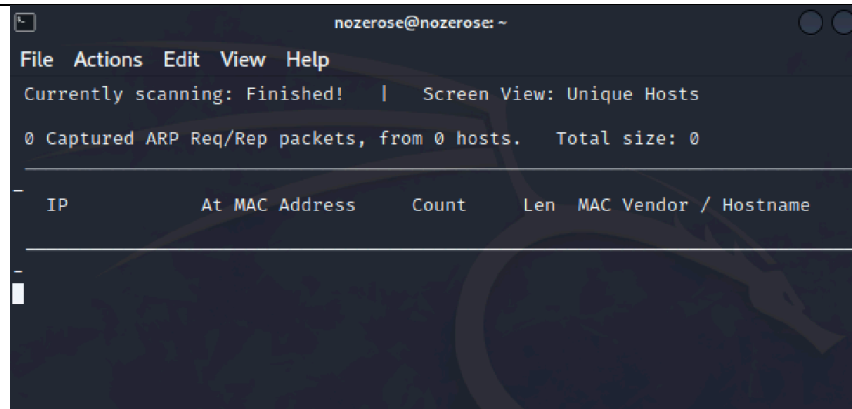
가상머신 리눅스에서 동일한 네트워크에 연결된 장비를 검색하고, 포트를 스캔하는 과제 입니다.

[TODO 1] VBox 가상머신의 네트워크 어댑터를 [어댑터에 브릿지]로 설정하세요. 가상머신에 할당된 IP 주소는 무엇인지, ifconfig 실행결과 화면을 아래에 캡처하여 첨부하세요.

```
(nozerose@nozerose)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::4c6d:fcff:fe4a:e5e0 prefixlen 64 scopeid 0x20<link>
    ether 4e:6d:fc:4a:e5:e0 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 16 bytes 3151 (3.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

[TODO 2] 가상머신 IP 가 속한 C 클래스 주소 전체를 대상으로 네트워크에 연결된 기기를 탐색하는 명령을 실행하고, 그 결과를 아래에 캡처하여 첨부하세요.



[TODO 3] 탐색된 기기 중 한 대를 골라(없으면 가상머신의 IP 를 사용), 해당 IP 를 대상으로 1-1000 번까지 포트 스캐닝을 수행하고, 그 결과를 아래에 캡처하여 첨부하세요.

```
(nozerose@nozerose)-[~]
$ nmap -p 1-1000 127.0.0.1
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-27 16:26 KST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000038s latency).
All 1000 scanned ports on localhost (127.0.0.1) are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds
```

[Q 3] VBox 가상 네트워크 [배점: 40]

포트 포워딩을 통해 호스트 PC 에서 가상머신으로 연결하는 실습입니다.

[TODO 1] 가상머신의 네트워크 어댑터를 [NAT]로 설정하세요. 가상머신에서 `ipconfig` 명령으로 IP 주소를 확인하고, 호스트 머신에서 `ifconfig` 또는 `ipconfig` 명령으로 IP 주소를 확인하세요

```
(nozerose@nozerose)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.64.3 netmask 255.255.255.0 broadcast 192.168.64.255
    inet6 fd4b:fef7:a622:d4be:7ef9:c913:38f5:d232 prefixlen 64 scopeid
0x0<global>
    inet6 fd4b:fef7:a622:d4be:4c6d:fcff:fe4a:e5e0 prefixlen 64 scopeid
0x0<global>
    inet6 fe80::4c6d:fcff:fe4a:e5e0 prefixlen 64 scopeid 0x20<link>
    ether 4e:6d:fc:4a:e5:e0 txqueuelen 1000 (Ethernet)
    RX packets 129565 bytes 168222824 (160.4 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 36588 bytes 2573439 (2.4 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

가상머신의 IP 주소는? 192.168.64.3

호스트 머신의 IP 주소는? 127.0.0.1

[TODO 2] 가상머신에서 ssh 서비스를 실행하세요. `$ systemctl` 명령으로 ssh 서비스의 status 를 조회하고, 그 결과를 캡처하여 아래에 첨부하세요.

```
(nozerose@nozerose)-[~]
$ systemctl start ssh.service

(nozerose@nozerose)-[~]
$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; disabled; preset: disa>
   Active: active (running) since Wed 2023-09-27 16:28:27 KST; 14s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 4354 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCC>
   Main PID: 4357 (sshd)
     Tasks: 1 (limit: 4545)
    Memory: 3.0M
       CPU: 13ms
   CGroup: /system.slice/ssh.service
           └─4357 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Sep 27 16:28:27 nozerose systemd[1]: Starting ssh.service  OpenBSD Secure S
Sep 27 16:28:27 nozerose sshd[4357]: Server listening on 0.0.0.0 port 22.
Sep 27 16:28:27 nozerose sshd[4357]: Server listening on :: port 22.
Sep 27 16:28:27 nozerose systemd[1]: Started ssh.service - OpenBSD Secure Sh
lines 1-17/17 (END)
```

[TODO 3] 호스트 머신에서 가상머신으로 ssh 접속을 시도하고, 그 결과를 캡처하세요. 지금은 접속이 되지 않고, timeout 오류가 발생해야 합니다.

```
[(base) nozerose@Chuu ~ % ssh nozerose@192.168.64.3  
ssh: connect to host 192.168.64.3 port 22: Operation timed out
```

[TODO 4] VBox 에서 포트 포워딩 규칙을 설정하세요. 포트 포워딩 규칙 입력을 위한 윈도우에 규칙을 입력한 후, 화면을 캡처하여 아래에 첨부하세요.

```
(nozerose@nozerose)-[~]  
$ sudo ufw allow 22/tcp  
  
Rules updated  
Rules updated (v6)
```

[TODO 5] 호스트 머신에서 가상머신으로 ssh 접속을 시도하고, 그 결과를 캡처하세요. 이제, 가상 머신으로 ssh 접속이 되어야 합니다. 캡처 화면에 1) ssh 접속을 위해 입력한 명령, 2) ssh 접속이 완료된 상황(예: 터미널 프롬프트가 변경된 것 등)이 모두 포함되어 있어야 합니다.

```
[(base) nozerose@Chuu ~ % ssh nozerose@192.168.64.3  
[nozerose@192.168.64.3's password:  
Linux nozerose 6.3.0-kali1-arm64 #1 SMP Debian 6.3.7-1kali1 (2023-06-29) aarch64  
  
The programs included with the Kali GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Wed Sep 27 16:35:54 2023 from 192.168.64.1  
(nozerose@nozerose)-[~]  
$ █
```