

실습#1

interface.c는 현재 연결된 네트워크 인터페이스의 정보를 확인하기 위한 코드인 것 같다. 작성된 내용에 따르면 동작은 다음과 같다.

ifconf 구조체와 내부의 ifreq 구조체의 공간을 확보해준 뒤, ioctl()을 SIOCGIFCONF와 함께 사용하여 소켓과 연결된 네트워크 인터페이스들을 ifconf 구조체의 리스트 형태로 가져온다. 그리고 for문을 이용해 리스트를 선형적으로 탐색하며 내부의 ifreq 구조체에 담겨 있는 내용을 적절한 포맷으로 출력해준다.

이 때, ifr_addr 필드의 주소가 이 INADDR_LOOPBACK인지 확인하여 루프백, 즉 자기 자신을 가리키는 경우를 제외한다. 정보 출력 시 ioctl의 두번째 인자로 SIOCGIFBRDADDR는 장치의 하드웨어 주소를 다루고, SIOCGIFBRDADDR는 broadcast 주소를, SIOCGIFNETMASK는 네트워크 마스크를, SIOCGIFMTU는 최대 전송 단위(MTU)를 다룰 때 사용한다.

ioctl()은 리눅스가 기본적으로 제공하는, 네트워크 인터페이스를 다루기 위한 표준화된 형태이다. ioctl()과 함께 사용되는, 가져온 정보를 저장하고 확인하기 위한 표준이 바로 Ifconf 구조체이다. 멤버로 ifreq 구조체를 가지고 있으며 이것이 네트워크 인터페이스에 대한 주요 정보를 담는 구조체이다. 코드에서는 ifr라는 이름의 포인터를 변수로 두어 ifconf 내부의 ifreq를 편리하게 참조하도록 하고 있다.

실행 결과는 다음과 같다.

```
leejjju@thxuberrypi:~/IoTLab/Lab3 $ ./interface
[lo]
IP      127.0.0.1
Loop Back
BRD     0.0.0.0
MASK    255.0.0.0
MTU     65536

[wlan0]
IP      192.168.0.24
MAC     e4:5f:1:fd:79:23
BRD     192.168.0.255
MASK    255.255.255.0
MTU     1500
```

실습#2

capture.c는 인자로 받은 네트워크 인터페이스로 들어오는 패킷들을 잡아서 적절한 형식으로 정보를 출력해주는 코드이다. 패킷 넘버와 source, destination의 MAC 주소, 프로토콜 넘버를 패킷 번호와 함께 출력해준다.

이 프로그램에서는 일반적인 네트워크 통신과는 다르게, socket() 함수의 두번째 인자로 SOCK_RAW를 주어 소켓을 raw socket으로 만든다. Raw socket은 헤더를 직접 제어할 수 있는, 때문에 어떤 종류의 헤더도 송수신이 가능한 소켓이다. 다른 소켓들처럼 헤더가 예쁘게 가공되어 오는 것이 아닌, 말 그대로 날 것으로 와서 이름이 raw인 것 같다. 이 raw socket을 사용하면 네트워크 계층으로 전송되는 모든 패킷을 전부 모니터링 및 감지가 가능하다고 한다. 이 경우에는 모든 이더넷 프로토콜을 수신하기 위해 ETH_P_ALL을 네트워크 바이트 순서로 변환하여 세번째 인자로 넣어준다.

Raw socket에 stockpot(), SO_SOCKET 플래그를 사용해 소켓 옵션을 설정해 줄 때는, SO_BINDTODEVICE를 인자로 주어 소켓을 특정 네트워크 인터페이스, 즉 프로그램이 argument로 받은 인터페이스에 바인딩되어 그곳을 통해 들어오는 패킷들만 수신할 수 있도록 해준다.

그리고 recv() 함수를 통해 raw socket에서 들어오는 패킷을 buffer로 받고, print_ethernet_header() 라는 함수로 보내서 리눅스가 정의한 ethhdr 구조체로 형변환 해주고, 구조체의 멤버에 담긴 정보들을 적절한 형식으로 출력해 준다. 이 과정은 패킷이 들어올 때 마다 반복된다.

실행 결과는 다음과 같다.

```
leejjju@thxuberrypi:~/IoTLab/Lab3 $ sudo ./capture wlan0
=====
Packet No: 0
-----
Ethernet Header
|-Source MAC Address      : 60-F6-77-66-26-96
|-Destination MAC Address : E4-5F-01-FD-79-23
|-Protocol                : 8
=====

=====
Packet No: 1
-----
Ethernet Header
|-Source MAC Address      : E4-5F-01-FD-79-23
|-Destination MAC Address : 60-F6-77-66-26-96
|-Protocol                : 8
=====

=====
Packet No: 2
-----
Ethernet Header
|-Source MAC Address      : E4-5F-01-FD-79-23
|-Destination MAC Address : 60-F6-77-66-26-96
|-Protocol                : 8
=====
```