Lab 2
Introduction to the Socket API

Prof. Kredo

Due: Start of lab Friday, February 6

Name:	
Name:	

Introduction

In this lab you will accomplish several goals:

- Discover Beej's Guide to Networking (BGN)
- Explore some of the Linux networking API
- Write a program to process networking addresses

In pairs for this lab using the equipment at your desk. Distribute the work evenly to make sure both group members know the material, as you will be required to know the material for evaluation.

1 Prelab [10 Points]

Perform the prelab reading and answer the questions before coming to lab.

1.1 Reading

Read §3[1] and §5.1[2] of BGN[3].

Questions

Answer these questions before lab.

1. Write a small segment of code that checks whether a `sockaddr*` named `sptr` points to an IPv4 address and, if so, casts it to a `sockaddr_in*` named `ipv4_ptr`.

[1] http://beej.us/guide/bgnet/output/html/singlepage/bgnet.html#ipstructsdata
[2] http://beej.us/guide/bgnet/output/html/singlepage/bgnet.html#getaddrinfo
[3] Beej's Guide to Network Programming, http://beej.us/guide/bgnet/output/html/singlepage/bgnet.html

2 Socket API

Let's begin our investigation of the socket API by working with programs that display addresses. Connect your host to the EXT network and configure your network interface as you did in Lab 1.

2.1 BGN `getaddrinfo` Example [30 Points]

Find the `showip.c` example of §5.1 of BGN. Download, compile, and run the example program. You should have already examined this code in the prelab. Pick a valid hostname (www.example.com) and use `showip` to find the addresses associated with the hostname.

1. What was your command and what was the output? Don't use www.example.com.

Modify the `showip` program to: (1) return addresses for all socket types and (2) display the socket type in the program output. You only need to consider sockets of type `SOCK_STREAM`, `SOCK_DGRAM`, and `SOCK_RAW`.

The output below is an example, but yours may have a slightly different format.

```
student@host ~ $ showip www.ietf.org
IP addresses for www.ietf.org:
IPv4(STREAM): 64.170.98.30
IPv4(DGRAM): 64.170.98.30
IPv4(RAW): 64.170.98.30
IPv6(STREAM): 2001:1890:126c::1:1e
IPv6(DGRAM): 2001:1890:126c::1:1e
IPv6(RAW): 2001:1890:126c::1:1e
```

When complete, have the lab TA sign for credit.

2.2 `getifaddrs` for Local Addresses [60 Points]

Your last assignment is to write a program to display local interface addresses. Specifically, you need to display the interface name, the address family, and the actual address itself. The primary system call for this is `getifaddrs(3)`. It has similar processing (linked list of address structures) to `getaddrinfo(3)` used in `showip`. You only need to consider the `AF_PACKET` (Link Layer), `AF_INET` (IPv4), and `AF_INET6` (IPv6) address families. The `ip(7)`, `ipv6(7)`, `packet(7)`, `inet_ntop(3)`, and `ether_ntoa(3)` man pages will be useful when writing this program. You must use system calls whenever possible (i.e., don't process addresses manually).

Submit your program and associated Makefile through Learn.

The output below is an example, but yours may have a slightly different format.

```
student@host ~ $ local_addrs
lo     AF_PACKET  0:0:0:0:0:0
eth0   AF_PACKET  0:21:CC:4A:F5:F6
lo     AF_INET    127.0.0.1
eth0   AF_INET    192.168.18.15
lo     AF_INET6   ::1
eth0   AF_INET6   fe80::8ea9:82ff:fe62:636c
```

For the more adventurous, display the netmask and broadcast address as well. When provided, the netmask and broadcast address can be handled just like a regular address.

Submit your completed lab handout and files for §2.2 before the next lab.

3 Lab 2 Addresses

	Host Addresses	
Desk	**PC 1**	PC 2
A	10.11.50.101	10.11.50.201
B	10.11.50.102	10.11.50.202
C	10.11.50.103	10.11.50.203
D	10.11.50.104	10.11.50.204
E	10.11.50.105	10.11.50.205
F	10.11.50.106	10.11.50.206
G	10.11.50.107	10.11.50.207
H	10.11.50.108	10.11.50.208
I	10.11.50.109	10.11.50.209
J	10.11.50.110	10.11.50.210
K	10.11.50.111	10.11.50.211
L	10.11.50.112	10.11.50.212
M	10.11.50.113	10.11.50.213
N	10.11.50.114	10.11.50.214
O	10.11.50.115	10.11.50.215