/\* CS480, Spring 2016

\*

\* Simple UDP client, to demonstrate use of the sockets API

\*/

#include <stdio.h>

#include <stdlib.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <string.h>

#include <assert.h>

#include <arpa/inet.h>

#include <netdb.h>

#include <limits.h>

#include <unistd.h>

#include <linux/if\_packet.h>

#include <sys/ioctl.h>

#include <net/if.h>

#define BUF\_SIZE 1024

void handle\_error(const char\* s)

{

perror(s);

exit(1);

}

int main(int argc, char\*\* argv)

{

int sock\_fd, ret, len;

struct sockaddr\_in addr;

size\_t num\_to\_send;

size\_t num\_sent;

struct sockaddr\_ll addr\_int;

struct ifreq ifr;

char \*Interface\_name;

char buf[BUF\_SIZE];

if (argc != 2) {

fprintf(stderr, "usage: %s interface name\n", argv[0]);

exit(1);

}

if ((sock\_fd = socket(AF\_PACKET, SOCK\_DGRAM, 0)) < 0)

handle\_error("socket");

memset(&addr, 0, sizeof(addr));

addr.sin\_family = AF\_INET;

addr.sin\_addr.s\_addr = inet\_addr("192.168.1.10");

Interface\_name = argv[1];

//ifr.ifr\_flags = IFF\_TAP;

strcpy(ifr.ifr\_name, Interface\_name);

if((ret = ioctl(sock\_fd, SIOCGIFINDEX, &ifr)) == -1){

perror("SIOCGIFINDEX");

exit(1);

}

memset(&addr\_int, 0, sizeof(addr\_int));

addr\_int.sll\_family = AF\_PACKET;

addr\_int.sll\_ifindex = ifr.ifr\_ifindex;

if((ret = bind(sock\_fd, (struct sockaddr \*)&addr\_int, sizeof(addr\_int))) == -1){

perror("bind");

exit(1);

}

len = 0;

buf[len++] = 0x88;

buf[len++] = 0x65;

buf[len++] = 0x12;

buf[len++] = 0xde;

buf[len++] = 0xe3;

printf("I am about to send %s to IP address %s\n", argv[1], inet\_ntoa(addr.sin\_addr));

num\_to\_send = strlen(buf);

num\_sent = sendto(sock\_fd, buf, num\_to\_send, 0, (const struct sockaddr\*)&addr\_int, sizeof(struct sockaddr\_ll));

printf("number of bytes sent:%d %d\n", num\_sent, num\_to\_send);

if (num\_sent != num\_to\_send) {

assert(num\_sent < 0);

handle\_error("sendto");

}

printf("Bytes sent !!!\n");

exit(0);

}