#include <fcntl.h>

#include <stdio.h>

#include <unistd.h>

#include <string.h>

#include <arpa/inet.h>

#include <linux/if.h>

#include <linux/if\_tun.h>

#include <sys/ioctl.h>

#define BUFF\_SIZE 2000

#define PORT\_NUMBER 55555

#define SERVER\_IP "127.0.0.1"

struct sockaddr\_in peerAddr;

int createTunDevice() {

int tunfd;

struct ifreq ifr;

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

ifr.ifr\_flags = IFF\_TUN | IFF\_NO\_PI;

tunfd = open("/dev/net/tun", O\_RDWR);

ioctl(tunfd, TUNSETIFF, &ifr);

return tunfd;

}

int connectToUDPServer(){

int sockfd;

char \*hello="Hello";

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

peerAddr.sin\_family = AF\_INET;

peerAddr.sin\_port = htons(PORT\_NUMBER);

peerAddr.sin\_addr.s\_addr = inet\_addr(SERVER\_IP);

sockfd = socket(AF\_INET, SOCK\_DGRAM, 0);

// Send a hello message to "connect" with the VPN server

sendto(sockfd, hello, strlen(hello), 0,

(struct sockaddr \*) &peerAddr, sizeof(peerAddr));

return sockfd;

}

void tunSelected(int tunfd, int sockfd){

int len;

char buff[BUFF\_SIZE];

printf("Got a packet from TUN\n");

bzero(buff, BUFF\_SIZE);

len = read(tunfd, buff, BUFF\_SIZE);

sendto(sockfd, buff, len, 0, (struct sockaddr \*) &peerAddr,

sizeof(peerAddr));

}

void socketSelected (int tunfd, int sockfd){

int len;

char buff[BUFF\_SIZE];

printf("Got a packet from the tunnel\n");

bzero(buff, BUFF\_SIZE);

len = recvfrom(sockfd, buff, BUFF\_SIZE, 0, NULL, NULL);

write(tunfd, buff, len);

}

int main (int argc, char \* argv[]) {

int tunfd, sockfd;

tunfd = createTunDevice();

sockfd = connectToUDPServer();

// Enter the main loop

while (1) {

fd\_set readFDSet;

FD\_ZERO(&readFDSet);

FD\_SET(sockfd, &readFDSet);

FD\_SET(tunfd, &readFDSet);

select(FD\_SETSIZE, &readFDSet, NULL, NULL, NULL);

if (FD\_ISSET(tunfd, &readFDSet)) tunSelected(tunfd, sockfd);

if (FD\_ISSET(sockfd, &readFDSet)) socketSelected(tunfd, sockfd);

}

}