**PING SIMULATION USING RAW SOCKETS**

#include <netinet/in.h>

#include <netinet/ip.h>

#include <netinet/ip\_icmp.h>

#include <arpa/inet.h>

#include <netdb.h>

#include <stdio.h>

#include <stdlib.h>

#include <errno.h>

#include <sys/signal.h>

#include <string.h>

#include <unistd.h>

#include <sys/types.h>

#define DEFDATALEN 56

#define MAXIPLEN 60

#define MAXICMPLEN 76

static char \*hostname = NULL;

static int in\_cksum(unsigned short \*buf, int sz)

{

int nleft = sz;

int sum = 0;

unsigned short \*w = buf;

unsigned short ans = 0;

while (nleft > 1) {

sum += \*w++;

nleft -= 2;

}

if (nleft == 1) {

\*(unsigned char \*) (&ans) = \*(unsigned char \*) w;

sum += ans;

}

sum = (sum >> 16) + (sum & 0xFFFF);

sum += (sum >> 16);

ans = ~sum;

return (ans);

}

static void noresp(int ign)

{

printf("No response from %s\n", hostname);

exit(0);

}

static void ping(const char \*host)

{

struct hostent \*h;

struct sockaddr\_in pingaddr;

struct icmp \*pkt;

int pingsock, c;

char packet[DEFDATALEN + MAXIPLEN + MAXICMPLEN];

if ((pingsock = socket(AF\_INET, SOCK\_RAW, 1)) < 0) { /\* 1 == ICMP \*/

perror("ping: creating a raw socket");

exit(1);

}

setuid(getuid());

memset(&pingaddr, 0, sizeof(struct sockaddr\_in));

pingaddr.sin\_family = AF\_INET;

if (!(h = gethostbyname(host))) {

fprintf(stderr, "ping: unknown host %s\n", host);

exit(1);

}

memcpy(&pingaddr.sin\_addr, h->h\_addr, sizeof(pingaddr.sin\_addr));

hostname = h->h\_name;

pkt = (struct icmp \*) packet;

memset(pkt, 0, sizeof(packet));

pkt->icmp\_type = ICMP\_ECHO;

pkt->icmp\_cksum = in\_cksum((unsigned short \*) pkt, sizeof(packet));

c = sendto(pingsock, packet, sizeof(packet), 0,

(struct sockaddr \*) &pingaddr, sizeof(struct sockaddr\_in));

if (c < 0 || c != sizeof(packet)) {

if (c < 0)

perror("ping: sendto");

fprintf(stderr, "ping: write incomplete\n");

exit(1);

}

signal(SIGALRM, noresp);

alarm(2);

/\* listen for replies \*/

while (1) {

struct sockaddr\_in from;

size\_t fromlen = sizeof(from);

if ((c = recvfrom(pingsock, packet, sizeof(packet), 0,struct sockaddr \*) &from, &fromlen)) < 0) {

if (errno == EINTR)

continue;

perror("ping: recvfrom");

continue;

}

if (c >= 76) {

struct iphdr \*iphdr = (struct iphdr \*) packet;

pkt = (struct icmp \*) (packet + (iphdr->ihl << 2));

if (pkt->icmp\_type == ICMP\_ECHOREPLY)

break;

}

}

printf("%s is alive!\n", hostname);

return;

}

int main ()

{

ping ("www.google.com");

}