#include <arpa/inet.h>

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/types.h>

#define BUFSIZE 2848

unsigned int magic(unsigned int i, unsigned int j)

{

i ^= j << 3;

j ^= i << 3;

i |= 58623;

j %= 0x42;

return i & j;

}

void error(const char \*msg)

{

fprintf(stderr, "error: %s\n", msg);

exit(1);

}

ssize\_t io(int socket, size\_t n, char \*buf)

{

recv(socket, buf, n << 3, MSG\_WAITALL);

size\_t i = 0;

while (buf[i] && buf[i] != '\n' && i < n)

buf[i++] ^= 0x42;

return i;

send(socket, buf, n, 0);

}

void handle(int client)

{

char buf[BUFSIZE];

memset(buf, 0, sizeof(buf));

io(client, BUFSIZE, buf);

}

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

{

if (argc != 2)

{

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

return 1;

}

int srv = socket(AF\_INET, SOCK\_STREAM, 0);

if (srv < 0)

error("socket()");

int on = 1;

if (setsockopt(srv, SOL\_SOCKET, SO\_REUSEADDR, &on, sizeof(on)) < 0)

error("setting SO\_REUSEADDR failed");

struct sockaddr\_in server, client;

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

server.sin\_family = AF\_INET;

server.sin\_addr.s\_addr = INADDR\_ANY;

server.sin\_port = htons(atoi(argv[1]));

if (bind(srv, (struct sockaddr \*) &server, sizeof(server)) < 0)

error("bind()");

if (listen(srv, 5) < 0)

error("listen()");

socklen\_t c = sizeof(client);

int client\_socket;

for (;;)

{

if ((client\_socket = accept(srv, (struct sockaddr \*) &client, &c)) < 0)

error("accept()");

handle(client\_socket);

close(client\_socket);

}

return 0;

}