#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <sys/types.h>

#include <sys/socket.h>

void error(const char \*msg) {

perror(msg);

exit(1);

}

int main() {

int sockfd, newsockfd, portno;

socklen\_t clilen;

struct sockaddr\_in serv\_addr, cli\_addr;

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

if (sockfd < 0)

error("ERROR opening socket");

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

portno = 12345;

serv\_addr.sin\_family = AF\_INET;

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

serv\_addr.sin\_port = htons(portno);

if (bind(sockfd, (struct sockaddr \*) &serv\_addr, sizeof(serv\_addr)) < 0)

error("ERROR on binding");

listen(sockfd, 5);

clilen = sizeof(cli\_addr);

newsockfd = accept(sockfd, (struct sockaddr \*) &cli\_addr, &clilen);

if (newsockfd < 0)

error("ERROR on accept");

struct sockaddr\_in peer\_addr;

socklen\_t peer\_addr\_len = sizeof(peer\_addr);

if (getpeername(newsockfd, (struct sockaddr \*) &peer\_addr, &peer\_addr\_len) < 0)

error("ERROR on getpeername");

char peer\_ip[INET\_ADDRSTRLEN];

inet\_ntop(AF\_INET, &peer\_addr.sin\_addr, peer\_ip, INET\_ADDRSTRLEN);

int peer\_port = ntohs(peer\_addr.sin\_port);

printf("Peer IP address: %s\n", peer\_ip);

printf("Peer port : %d\n", peer\_port);

close(newsockfd);

close(sockfd);

return 0;

}

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <sys/types.h>

#include <sys/socket.h>

void error(const char \*msg) {

perror(msg);

exit(1);

}

int main() {

int sockfd, portno;

struct sockaddr\_in serv\_addr;

struct hostent \*server;

portno = 12345;

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

if (sockfd < 0)

error("ERROR opening socket");

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

serv\_addr.sin\_family = AF\_INET;

serv\_addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

serv\_addr.sin\_port = htons(portno);

if (connect(sockfd, (struct sockaddr \*) &serv\_addr, sizeof(serv\_addr)) < 0)

error("ERROR connecting");

printf("Connected to the server\n");

close(sockfd);

return 0;

}