#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <netdb.h>

#define PORT 80

#define BUF\_SIZE 4096

int main() {

int sockfd;

struct sockaddr\_in server\_addr;

struct hostent \*server;

char request[BUF\_SIZE];

char response[BUF\_SIZE];

int bytes\_received;

// The web server hostname and path

const char \*hostname = "example.com";

const char \*path = "/";

// Create socket

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

if (sockfd < 0) {

perror("Socket creation failed");

exit(EXIT\_FAILURE);

}

// Get the server's IP address by hostname

server = gethostbyname(hostname);

if (server == NULL) {

fprintf(stderr, "Error, no such host\n");

exit(EXIT\_FAILURE);

}

// Setup server address struct

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

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

server\_addr.sin\_port = htons(PORT);

memcpy(&server\_addr.sin\_addr.s\_addr, server->h\_addr, server->h\_length);

// Connect to the server

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

perror("Connection failed");

close(sockfd);

exit(EXIT\_FAILURE);

}

// Manually craft the HTTP GET request

snprintf(request, sizeof(request),

"GET %s HTTP/1.1\r\n"

"Host: %s\r\n"

"Connection: close\r\n"

"\r\n", path, hostname);

// Send the HTTP request

if (send(sockfd, request, strlen(request), 0) < 0) {

perror("Send failed");

close(sockfd);

exit(EXIT\_FAILURE);

}

// Receive the HTTP response

printf("Received response:\n");

while ((bytes\_received = recv(sockfd, response, sizeof(response) - 1, 0)) > 0) {

response[bytes\_received] = '\0'; // Null-terminate the response

printf("%s", response);

}

if (bytes\_received < 0) {

perror("Receive failed");

}

// Close the socket

close(sockfd);

return 0;

}