Implementation of setsockopt,getsockopt system calls

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <unistd.h>

int main() {

int sockfd;

int optval = 1;

int optlen;

struct sockaddr\_in server\_addr;

// Create a socket

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

if (sockfd < 0) {

perror("socket");

exit(EXIT\_FAILURE);

}

// Set the `SO\_REUSEADDR` option on the socket using setsockopt()

if (setsockopt(sockfd, SOL\_SOCKET, SO\_REUSEADDR, &optval, sizeof(optval)) < 0) {

perror("setsockopt");

close(sockfd);

exit(EXIT\_FAILURE);

}

// Retrieve the `SO\_REUSEADDR` option using getsockopt() to verify it was set

optlen = sizeof(optval);

if (getsockopt(sockfd, SOL\_SOCKET, SO\_REUSEADDR, &optval, (socklen\_t \*)&optlen) < 0) {

perror("getsockopt");

close(sockfd);

exit(EXIT\_FAILURE);

}

// Print the value of the SO\_REUSEADDR option

printf("SO\_REUSEADDR is %s\n", optval ? "ON" : "OFF");

// Setup the server address struct

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

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

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

server\_addr.sin\_port = htons(8080);

// Bind the socket to the port 8080

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

perror("bind");

close(sockfd);

exit(EXIT\_FAILURE);

}

printf("Socket successfully created and bound to port 8080.\n");

// Close the socket

close(sockfd);

return 0;

}