

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”  
КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ УПРАВЛІННЯ

Звіт з лабораторної роботи № 7  
з дисципліни “Основи архітектури ЕОМ та операційні системи”

Виконав: ст. гр. КН-221В

Шулюпов Єгор Русланович

Перевірив: Асистент каф. ПІІТУ

Дмитро Миколайович Ковальчук

Харків

2022

## ТЕМА: ВВЕДЕННЯ У МІЖПРОЦЕСОРНУ ВЗАЄМОДІЮ. СОКЕТИ.

– Відтворити приклади, що є у методичних вказівках

### 2 ОСНОВНА ЧАСТИНА

Файл socket.c

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <string.h>
4  #include <errno.h>
5  #include <sys/types.h>
6  #include <sys/socket.h>
7  #define SOCK_NAME "socket.soc"
8  #define BUF_SIZE 256
9  int main(int argc, char** argv)
10 {
11     struct sockaddr srvr_name, rcvr_name;
12     char buf[BUF_SIZE];
13     int sock;
14     int namelen, bytes;
15     sock = socket(AF_UNIX, SOCK_DGRAM, 0);
16     if(sock < 0)
17     {
18         perror("socket failed");
19         return EXIT_FAILURE;
20     }
21     srvr_name.sa_family = AF_UNIX;
22     strcpy(srvr_name.sa_data, SOCK_NAME);
23     if(bind(sock, &srvr_name, strlen(srvr_name.sa_data)+sizeof(srvr_name.sa_family)) < 0)
24     {
25         perror("bind failed");
26         return EXIT_FAILURE;
27     }
28     bytes = recvfrom(sock, buf, sizeof(buf), 0, &rcvr_name, &namelen);
29     if(bytes < 0)
30     {
31         perror("recvfrom failed");
32         return EXIT_FAILURE;
33     }
34     buf[bytes] = 0;
35     rcvr_name.sa_data[namelen] = 0;
36     printf("Client sent: %s\n", buf);
37     close(sock);
38     unlink(SOCK_NAME);
39     return 0;
40 }
```

## Файл fsclient.c

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <string.h>
4  #include <errno.h>
5  #include <sys/types.h>
6  #include <sys/socket.h>
7  #define SOCK_NAME "socket.soc"
8  #define BUF_SIZE 256
9  int main(int argc, char** argv)
10 {
11     int sock;
12     sock = socket(AF_UNIX, SOCK_DGRAM, 0);
13     char buf[BUF_SIZE];
14     struct sockaddr srvr_name;
15     if(sock < 0)
16     {
17         perror("socket failed");
18         return EXIT_FAILURE;
19     }
20     srvr_name.sa_family = AF_UNIX;
21     strcpy(srvr_name.sa_data, SOCK_NAME);
22     strcpy(buf, "Hello, Unix sockets");
23     sendto(sock, buf, strlen(buf), 0, &srvr_name,
24            strlen(srvr_name.sa_data)+sizeof(srvr_name.sa_family));
25     return 0;
26 }
```

```
21 {
22     close(sockets[1]);
23     write(sockets[0], STR1, sizeof(STR1));
24     read(sockets[0], buf, sizeof(buf));
25     printf("%s\n", buf);
26     close(sockets[0]);
27 }
28 else
29 {
30     close(sockets[0]);
31     read(sockets[1], buf, sizeof(buf));
32     printf("%s\n", buf);
33     write(sockets[1], STR2, sizeof(STR2));
34     close(sockets[1]);
35 }
36 }
```

```
vorpolochek@vorpolochek-VirtualBox:~/lab_ECM/#7$ ./sockpair
123
net 321
```

## Файл netserver.c

```

home > vorpolochek > lab_EOM > #7 > C netserver.c > main(int, char **)
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <errno.h>
4  #include <string.h>
5  #include <sys/types.h>
6  #include <sys/socket.h>
7  #include <netinet/in.h>
8  #define BUF_SIZE 256
9  int main(int argc, char** argv)
10 {
11     int sock, newsock, port, clen;
12     char buf[BUF_SIZE];
13     struct sockaddr_in serv_addr, cli_addr;
14     if(argc < 2)
15     {
16         fprintf(stderr, "usage: %s <port_number>\n", argv[0]);
17         return EXIT_FAILURE;
18     }
19     sock = socket(AF_INET, SOCK_STREAM, 0);
20     printf("accept() failed: %d\n", errno);
21     return EXIT_FAILURE;
22 }
23
24     memset(buf, 0, BUF_SIZE);
25     read(newsock, buf, BUF_SIZE-1);
26     buf[BUF_SIZE] = 0;
27     printf("MSG: %s\n", buf);
28     write(newsock, "OK", 2);
29     close(newsock);
30     close(sock);
31 }

```

## Файл netclient.c

```

home > vorpolochek > lab_EOM > #7 > C netclient.c > main(int, char **)
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <errno.h>
4  #include <string.h>
5  #include <sys/types.h>
6  #include <sys/socket.h>
7  #include <netinet/in.h>
8  #include <netdb.h>
9  #define BUF_SIZE 256
10 int main(int argc, char** argv)
11 {
12     int sock, port;
13     struct sockaddr_in serv_addr;
14     struct hostent *server;
15     char buf[BUF_SIZE];
16     if(argc < 3)
17     {
18         fprintf(stderr, "use: %s <hostname> <port_number>\n", argv[0]);
19         return EXIT_FAILURE;
20     }
21     port = atoi(argv[2]);
22     sock = socket(AF_INET, SOCK_STREAM, 0);
23     if(sock < 0)
24     {
25         printf("socket() failed: %d", errno);
26         return EXIT_FAILURE;
27     }
28     server = gethostbyname(argv[1]);
29     if(server == NULL)
30     {
31         fprintf(stderr, "gethostbyname() failed: %d\n", errno);
32         return EXIT_FAILURE;
33     }
34     serv_addr.sin_family = AF_INET;
35     bzero(serv_addr.sin_addr, sizeof(serv_addr.sin_addr));
36     memcpy(&serv_addr.sin_addr, server->h_addr, server->h_addr_len);
37     serv_addr.sin_port = htons(port);
38     if(connect(sock, (struct sockaddr*)&serv_addr, sizeof(serv_addr)) < 0)
39     {
40         printf("connect() failed: %d\n", errno);
41         return EXIT_FAILURE;
42     }
43     read(sock, buf, BUF_SIZE-1);
44     buf[BUF_SIZE] = 0;
45     printf("MSG: %s\n", buf);
46     write(sock, "OK", 2);
47     close(sock);
48 }

```

```

28  server = gethostbyname(argv[1]);
29  if(server == NULL)
30  {
31  printf("Host not found\n");
32  return EXIT_FAILURE;
33  }
34  memset((char*)&serv_addr, 0, sizeof(serv_addr));
35  serv_addr.sin_family = AF_INET;
36  memcpy(&serv_addr.sin_addr.s_addr, server->h_addr, server->h_length);
37  serv_addr.sin_port = htons(port);
38  if(connect(sock, &serv_addr, sizeof(serv_addr)) < 0)
39  {
40  printf("connect() failed: %d\n", errno);
41  return EXIT_FAILURE;
42  }
43  printf(">");
44  memset(buf, 0, BUF_SIZE);
45  fgets(buf, BUF_SIZE-1, stdin);
46  write(sock, buf, strlen(buf));
47  memset(buf, 0, BUF_SIZE);
48  read(sock, buf, BUF_SIZE-1);
49  printf("%s\n", buf);
50  close(sock);
51  return 0;

```

hinal

```

vorplochek@vorplochek-VirtualBox:~/lab_ECM/#7$ ./netserver 1
bind() failed: 13
vorplochek@vorplochek-VirtualBox:~/lab_ECM/#7$ ./netclient www.unix.com 8080
connect() failed: 111

```

### Висновок

Під час виконання цієї лабораторної роботи я ознайомився з сокетами та міжпроцесною взаємодією у Linux. Навчився працювати з сокетами.