



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

---

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

Дисциплина: «Операционные системы»

Лабораторная работа №6

Тема работы:  
«Сокеты»

Студент: Левушкин И. К.

Группа: ИУ7-62Б

Преподаватель: Рязанова Н. Ю.

Москва, 2020 г.

## Задание 1.

Написать приложение по модели клиент-сервер, демонстрирующее взаимодействие параллельных процессов на отдельном компьютере с использованием сокетов в файловом пространстве имен: семейство - AF\_UNIX, тип - SOCK\_DGRAM. При демонстрации работы программного комплекса необходимо запустить несколько клиентов (не меньше 5) и продемонстрировать, что сервер обрабатывает обращения каждого запущенного клиента.

Ниже приведен программный комплекс, реализующий поставленную задачу.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <unistd.h>
5  #include <sys/types.h>
6  #include <signal.h>
7  #include <sys/socket.h>
8
9  #include "info.h"
10
11 int sockfd;
12
13 void sigint_catcher(int signum)
14 {
15     printf("Closing socket 'cause of Ctrl+C...\n");
16     close(sockfd);
17     unlink(SOCKET_NAME);
18 }
19
20
21 int main(void)
22 {
23     char msg[MSG_LEN];
24     struct sockaddr client_addr;
25
26     if ((sockfd = socket(PF_LOCAL, SOCK_DGRAM, 0)) < 0)
27     {
28         perror("Error in socket(): ");
29         return sockfd;
30     }
31
32     client_addr.sa_family = PF_LOCAL;
33     strcpy(client_addr.sa_data, SOCKET_NAME);
34
35     if (bind(sockfd, &client_addr, sizeof(client_addr)) < 0)
36     {
37         printf("Closing socket...\n");
38         close(sockfd);
39         unlink(SOCKET_NAME);
40         perror("Error in bind(): ");
41         return -1;
```

```

42     }
43
44     printf("\nServer is waiting for the message...\n");
45     signal(SIGINT, sigint_catcher);
46
47     int recievedSize;
48
49     while(1)
50     {
51         if ((recievedSize = recv(sockfd, msg, sizeof(msg), 0)) < 0)
52         {
53             close(sockfd);
54             unlink(SOCKET_NAME);
55             perror("Error in recv(): ");
56             return recievedSize;
57         }
58
59         msg[recievedSize] = 0;
60         printf("Client send this message: %s\n", msg);
61     }
62
63     printf("Closing socket...\n");
64     close(sockfd);
65     unlink(SOCKET_NAME);
66     return 0;
67 }
68

```

Рис. 1: server.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <unistd.h>
5  #include <sys/types.h>
6  #include <sys/socket.h>
7
8  #include "info.h"
9
10 int main(void)
11 {
12     int sockfd = socket(PF_LOCAL, SOCK_DGRAM, 0); //socket descriptor
13     if (sockfd < 0)
14     {
15         printf("Error in socket();\n");
16         return sockfd;
17     }
18
19     struct sockaddr server_addr;
20     server_addr.sa_family = PF_LOCAL;
21     strcpy(server_addr.sa_data, SOCKET_NAME);
22
23     char msg[MSG_LEN];
24
25     sprintf(msg, "Hello, I am a client, pid = %d\n", getpid());
26     sendto(sockfd, msg, strlen(msg), 0, &server_addr, sizeof(server_addr));
27
28     close(sockfd);
29     return 0;
30 }
31

```

Рис. 2: client.c

```

1  #ifndef INFO_H
2  #define INFO_H
3
4  #define MSG_LEN 256
5  #define SOCKET_NAME "socket.soc"
6
7  #endif // INFO_H

```

Рис. 3: info.h

## Демонстрация работы программы

Ниже представлены результаты работы программного комплекса с 5-ю запущенными клиентами. Выход из программы осуществляется сигналом SIGINT.

```

ilalevuskin@ubuntu:~/Desktop/lab6/unix$ ./server
Server is waiting for the message...
Client send this message: Hello, I am a client, pid = 2495
Client send this message: Hello, I am a client, pid = 2508
Client send this message: Hello, I am a client, pid = 2517
Client send this message: Hello, I am a client, pid = 2520
Client send this message: Hello, I am a client, pid = 2543
^CClosing socket 'cause of Ctrl+C....

```

```

ilalevuskin@ubuntu:~/Desktop/lab6/unix$ ./client
ilalevuskin@ubuntu:~/Desktop/lab6/unix$ ./client
ilalevuskin@ubuntu:~/Desktop/lab6/unix$ ./client
ilalevuskin@ubuntu:~/Desktop/lab6/unix$ ./client
ilalevuskin@ubuntu:~/Desktop/lab6/unix$ ./client

```

Рис. 4: Демонстрация работы программного комплекса

## Задание 2.

Написать приложение по модели клиент-сервер, осуществляющее взаимодействие параллельных процессов, которые выполняются на разных компьютерах. Для взаимодействия с клиентами сервер должен использовать мультиплексирование. Сервер должен обслуживать запросы параллельно запущенных клиентов. При демонстрации работы программного комплекса необходимо запустить несколько клиентов (не меньше 5) и продемонстрировать, что сервер обрабатывает обращения каждого запущенного клиента.

Ниже приведен программный комплекс, реализующий поставленную задачу.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <errno.h>
5  #include <unistd.h>
6  #include <sys/types.h>
7  #include <sys/socket.h>
8  #include <sys/select.h>
9  #include <arpa/inet.h>
10 #include <netdb.h>
11
12 #include "info.h"
13
14 #define MAX_CLIENTS 10
15 int clients[MAX_CLIENTS] = { 0 };
16
17
18 void manageConnection(unsigned int fd)
19 {
20     struct sockaddr_in client_addr;
21     int addrSize = sizeof(client_addr);
22
23     int incom = accept(fd, (struct sockaddr*) &client_addr, (socklen_t*) &addrSize);
24     if (incom < 0)
25     {
26         perror("Error in accept(): ");
27         exit(-1);
28     }
29
30     printf("\nNew connection: \nfd = %d \nip = %s:%d\n", incom,
31           inet_ntoa(client_addr.sin_addr), ntohs(client_addr.sin_port));
32
33     for (int i = 0; i < MAX_CLIENTS; i++)
34     {
35         if (clients[i] == 0)
36         {
37             clients[i] = incom;
38             printf("Managed as client #%d\n", i);
39             break;
40         }
41     }
42 }
43 }
```

Рис. 5: server.c

```

44
45 void manageClient(unsigned int fd, unsigned int client_id)
46 {
47     char msg[MSG_LEN];
48     memset(msg, 0, MSG_LEN);
49
50     struct sockaddr_in client_addr;
51     int addrSize = sizeof(client_addr);
52
53     int recvSize = recv(fd, msg, MSG_LEN, 0);
54     if (recvSize == 0)
55     {
56         getpeername(fd, (struct sockaddr*) &client_addr, (socklen_t*) &addrSize);
57         printf("User %d disconnected %s:%d \n", client_id, inet_ntoa(client_addr.sin_addr),
58             ntohs(client_addr.sin_port));
59         close(fd);
60         clients[client_id] = 0;
61     }
62     else
63     {
64         msg[recvSize] = '\0';
65         printf("Message from %d client: %s\n", client_id, msg);
66     }
67 }
68
69
70 int main(void)
71 {
72     int listener = socket(PF_INET, SOCK_STREAM, 0);
73     if (listener < 0)
74     {
75         perror("Error in sock(): ");
76         return listener;
77     }
78
79     struct sockaddr_in client_addr;
80     client_addr.sin_family = PF_INET;
81     client_addr.sin_port = htons(SOCK_PORT);
82     client_addr.sin_addr.s_addr = INADDR_ANY;
83
84     if (bind(listener, (struct sockaddr*) &client_addr, sizeof(client_addr)) < 0)
85     {
86         perror("Error in bind():");

```

Рис. 6: server.c

```

87     return -1;
88 }
89 printf("Server is listening on the %d port...\n", SOCK_PORT);
90
91 if (listen(listener, 3) < 0)
92 {
93     perror("Error in listen(): ");
94     return -1;
95 }
96 printf("Waiting for the connections...\n");
97
98 fd_set readfds;
99 int max_fd;
100 int active_clients_count;
101
102 while(1)
103 {
104     FD_ZERO(&readfds);
105     FD_SET(listener, &readfds);
106     max_fd = listener;
107
108     for (int i = 0; i < MAX_CLIENTS; i++)
109     {
110         int fd = clients[i];
111
112         if (fd > 0)
113         {
114             FD_SET(fd, &readfds);
115         }
116
117         max_fd = (fd > max_fd) ? (fd) : (max_fd);
118     }
119
120     active_clients_count = select(max_fd + 1, &readfds, NULL, NULL, NULL);
121
122     if (active_clients_count < 0 && (errno != EINTR))
123     {
124         perror("Error in select():");
125         return active_clients_count;
126     }
127
128     if (FD_ISSET(listener, &readfds))
129     {

```

Рис. 7: server.c

```

130         manageConnection(listener);
131     }
132
133     for (int i = 0; i < MAX_CLIENTS; i++)
134     {
135         int fd = clients[i];
136         if ((fd > 0) && FD_ISSET(fd, &readfds))
137         {
138             manageClient(fd, i);
139         }
140     }
141 }
142
143 return 0;
144 }
145

```

Рис. 8: server.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <time.h>
5  #include <unistd.h>
6  #include <signal.h>
7  #include <sys/types.h>
8  #include <sys/socket.h>
9  #include <arpa/inet.h>
10 #include <netdb.h>
11
12 #include "info.h"
13
14 int main(void)
15 {
16     srand(time(NULL));
17
18     int sock = socket(PF_INET, SOCK_STREAM, 0);
19     if (sock < 0)
20     {
21         perror("Error in sock(): ");
22         return sock;
23     }
24
25     struct hostent* host = gethostbyname(SOCK_ADDR);
26     if (!host)
27     {
28         perror("Error in gethostbyname(): ");
29         return -1;
30     }
31
32     struct sockaddr_in server_addr;
33     server_addr.sin_family = PF_INET;
34     server_addr.sin_port = htons(SOCK_PORT);
35     server_addr.sin_addr = *((struct in_addr*) host->h_addr_list[0]);
36
37     if (connect(sock, (struct sockaddr*) &server_addr, sizeof(server_addr)) < 0)
38     {
39         perror("Error in connect():");
40         return -1;
41     }
42
43     char msg[MSG_LEN];
44     for (int i = 0; i < 10; i++)
45     {
46         memset(msg, 0, MSG_LEN);
47         sprintf(msg, "%d message was sended, pid = %d\n", i, getpid());
48         printf("%s", msg);
49
50         if (send(sock, msg, strlen(msg), 0) < 0)
51         {
52             perror("Error in send(): ");
53             return -1;
54         }
55
56         printf("Sended %d message\n", i);
57
58         int wait_time = 1 + rand() % 3;
59         sleep(wait_time);
60     }
61
62     printf("Client app is over!\n");
63     return 0;
64 }
65

```

Рис. 9: client.c



## Демонстрация работы программы

Ниже представлены результаты работы программного комплекса с 5-ю параллельно запущенными клиентами.

```
ilalevuskin@ubuntu:~/Desktop/lab6/net$ ./server
Server is listening on the 31337 port...
Waiting for the connections...

New connection:
fd = 4
ip = 127.0.0.1:50958
Managed as client #0
Message from 0 client: 0 message was sended, pid = 3704

New connection:
fd = 5
ip = 127.0.0.1:50960
Managed as client #1
Message from 1 client: 0 message was sended, pid = 3705

Message from 0 client: 1 message was sended, pid = 3704

New connection:
fd = 6
ip = 127.0.0.1:50962
Managed as client #2
Message from 2 client: 0 message was sended, pid = 3710

Message from 1 client: 1 message was sended, pid = 3705

New connection:
fd = 7
ip = 127.0.0.1:50964
Managed as client #3
Message from 3 client: 0 message was sended, pid = 3711

Message from 0 client: 2 message was sended, pid = 3704

Message from 1 client: 2 message was sended, pid = 3705

New connection:
fd = 8
ip = 127.0.0.1:50966
Managed as client #4
Message from 4 client: 0 message was sended, pid = 3715

Message from 0 client: 3 message was sended, pid = 3704
Message from 2 client: 1 message was sended, pid = 3710
Message from 1 client: 3 message was sended, pid = 3705
Message from 3 client: 1 message was sended, pid = 3711
Message from 0 client: 4 message was sended, pid = 3704
Message from 1 client: 4 message was sended, pid = 3705
Message from 3 client: 2 message was sended, pid = 3711
Message from 4 client: 1 message was sended, pid = 3715
Message from 0 client: 5 message was sended, pid = 3704
Message from 2 client: 2 message was sended, pid = 3710
Message from 1 client: 5 message was sended, pid = 3705
Message from 3 client: 3 message was sended, pid = 3711
Message from 4 client: 2 message was sended, pid = 3715
Message from 2 client: 3 message was sended, pid = 3710
Message from 0 client: 6 message was sended, pid = 3704
Message from 1 client: 6 message was sended, pid = 3705
Message from 4 client: 3 message was sended, pid = 3715
Message from 3 client: 4 message was sended, pid = 3711
Message from 4 client: 4 message was sended, pid = 3715
Message from 0 client: 7 message was sended, pid = 3704
Message from 2 client: 4 message was sended, pid = 3710
Message from 1 client: 7 message was sended, pid = 3705
```

Рис. 10: Результат работы сервера

```
Message from 3 client: 5 message was sended, pid = 3711
Message from 2 client: 5 message was sended, pid = 3710
Message from 3 client: 6 message was sended, pid = 3711
Message from 3 client: 7 message was sended, pid = 3711
Message from 4 client: 5 message was sended, pid = 3715
Message from 0 client: 8 message was sended, pid = 3704
Message from 1 client: 8 message was sended, pid = 3705
Message from 0 client: 9 message was sended, pid = 3704
Message from 2 client: 6 message was sended, pid = 3710
Message from 1 client: 9 message was sended, pid = 3705
Message from 3 client: 8 message was sended, pid = 3711
Message from 4 client: 6 message was sended, pid = 3715
Message from 3 client: 9 message was sended, pid = 3711

User 3 disconnected 127.0.0.1:50964
User 0 disconnected 127.0.0.1:50958
Message from 2 client: 7 message was sended, pid = 3710

User 1 disconnected 127.0.0.1:50960
Message from 4 client: 7 message was sended, pid = 3715

Message from 2 client: 8 message was sended, pid = 3710
Message from 2 client: 9 message was sended, pid = 3710
Message from 4 client: 8 message was sended, pid = 3715
Message from 4 client: 9 message was sended, pid = 3715

User 2 disconnected 127.0.0.1:50962
User 4 disconnected 127.0.0.1:50966
^C
```

Рис. 11: Результат работы сервера

```

ilalevuskina@ubuntu:~/Desktop/lab6/net$ ./client
0 message was sended, pid = 3704
Sended 0 message
1 message was sended, pid = 3704
Sended 1 message
2 message was sended, pid = 3704
Sended 2 message
3 message was sended, pid = 3704
Sended 3 message
4 message was sended, pid = 3704
Sended 4 message
5 message was sended, pid = 3704
Sended 5 message
6 message was sended, pid = 3704
Sended 6 message
7 message was sended, pid = 3704
Sended 7 message
8 message was sended, pid = 3704
Sended 8 message
9 message was sended, pid = 3704
Sended 9 message
Client app is over!
ilalevuskina@ubuntu:~/Desktop/lab6/net$ █
ilalevuskina@ubuntu:~/Desktop/lab6/net$ ./client
0 message was sended, pid = 3705
Sended 0 message
1 message was sended, pid = 3705
Sended 1 message
2 message was sended, pid = 3705
Sended 2 message
3 message was sended, pid = 3705
Sended 3 message
4 message was sended, pid = 3705
Sended 4 message
5 message was sended, pid = 3705
Sended 5 message
6 message was sended, pid = 3705
Sended 6 message
7 message was sended, pid = 3705
Sended 7 message
8 message was sended, pid = 3705
Sended 8 message
9 message was sended, pid = 3705
Sended 9 message
Client app is over!
ilalevuskina@ubuntu:~/Desktop/lab6/net$ █
ilalevuskina@ubuntu:~/Desktop/lab6/net$ ./client
0 message was sended, pid = 3715
Sended 0 message
1 message was sended, pid = 3715
Sended 1 message
2 message was sended, pid = 3715
Sended 2 message
3 message was sended, pid = 3715
Sended 3 message
4 message was sended, pid = 3715
Sended 4 message
5 message was sended, pid = 3715
Sended 5 message
6 message was sended, pid = 3715
Sended 6 message
7 message was sended, pid = 3715
Sended 7 message
8 message was sended, pid = 3715
Sended 8 message
9 message was sended, pid = 3715
Sended 9 message
Client app is over!
ilalevuskina@ubuntu:~/Desktop/lab6/net$ █
ilalevuskina@ubuntu:~/Desktop/lab6/net$ ./client
0 message was sended, pid = 3710
Sended 0 message
1 message was sended, pid = 3710
Sended 1 message
2 message was sended, pid = 3710
Sended 2 message
3 message was sended, pid = 3710
Sended 3 message
4 message was sended, pid = 3710
Sended 4 message
5 message was sended, pid = 3710
Sended 5 message
6 message was sended, pid = 3710
Sended 6 message
7 message was sended, pid = 3710
Sended 7 message
8 message was sended, pid = 3710
Sended 8 message
9 message was sended, pid = 3710
Sended 9 message
Client app is over!
ilalevuskina@ubuntu:~/Desktop/lab6/net$ █

```

Рис. 12: Демонстрация запуска четырех клиентов

```
ilalevuskin@ubuntu:~/Desktop/lab6/net$ ./client
0 message was sended, pid = 3711
Sended 0 message
1 message was sended, pid = 3711
Sended 1 message
2 message was sended, pid = 3711
Sended 2 message
3 message was sended, pid = 3711
Sended 3 message
4 message was sended, pid = 3711
Sended 4 message
5 message was sended, pid = 3711
Sended 5 message
6 message was sended, pid = 3711
Sended 6 message
7 message was sended, pid = 3711
Sended 7 message
8 message was sended, pid = 3711
Sended 8 message
9 message was sended, pid = 3711
Sended 9 message
Client app is over!
```

Рис. 13: Демонстрация запуска 5-ого клиента