



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»

О Т Ч Е Т

по лабораторной работе № 0 8 (1)

Дисциплина: *Операционные системы*

Студент

ИУ7И-66Б

(Группа)

Нгуен Ф. С.

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

Рязанова Н. Ю.

(Подпись, дата)

(И.О. Фамилия)

Москва, 2021

Код программы

Server.c

```
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <signal.h>
#include <sys/types.h>
#include <sys/socket.h>

#define SOCKET_NAME "./socket"
#define BUF_SIZE 256
#define OK 0

static int sockfd;

void cleanup_socket(void)
{
    close(sockfd);
    unlink(SOCKET_NAME);
}

void sigint_handler(int signum)
{
    cleanup_socket();
    exit(OK);
}

int main(void)
{
    if ((sockfd = socket(AF_UNIX, SOCK_DGRAM, 0)) < 0)
    {
        perror("Failed to create socket");
        return EXIT_FAILURE;
    }

    struct sockaddr srvr_name;
    srvr_name.sa_family = AF_UNIX;
    strcpy(srvr_name.sa_data, SOCKET_NAME);
    if (bind(sockfd, &srvr_name, strlen(srvr_name.sa_data) +
sizeof(srvr_name.sa_family)) < 0)
    {
        perror("Failed to bind socket");
        return EXIT_FAILURE;
    }

    signal(SIGINT, sigint_handler);
    fprintf(stdout, "Server is listening.\nTo stop server press Ctrl + C.\n");

    char buf[BUF_SIZE];
    for (;;)
    {
        int bytes = recv(sockfd, buf, sizeof(buf), 0);
        if (bytes <= 0)
        {
            perror("Failed to recv");
            cleanup_socket();
            return EXIT_FAILURE;
        }

        buf[bytes] = '\0';
```

```

        fprintf(stdout, "Server read: [%s]\n", buf);
    }

    fprintf(stdout, "Server stopped listening\n");
    cleanup_socket();
    fprintf(stdout, "Socket closed\n");

    return OK;
}

```

Client.c

```

#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>

#define SOCKET_NAME "./socket"
#define BUF_SIZE 256
#define OK 0

int main(void)
{
    int sockfd = socket(AF_UNIX, SOCK_DGRAM, 0);
    if (sockfd < 0)
    {
        perror("Failed to create socket");
        return EXIT_FAILURE;
    }

    struct sockaddr srvr_name;
    srvr_name.sa_family = AF_UNIX;
    strcpy(srvr_name.sa_data, SOCKET_NAME);

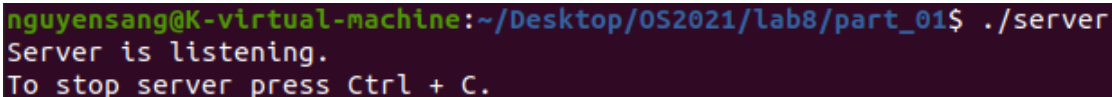
    char buf[BUF_SIZE];
    snprintf(buf, BUF_SIZE, "This Message From %d", getpid());
    if (sendto(sockfd, buf, strlen(buf), 0, &srvr_name, strlen(srvr_name.sa_data)
+ sizeof(srvr_name.sa_family)) < 0)
    {
        perror("Failed to send message");
        close(sockfd);
        return EXIT_FAILURE;
    }

    printf("Client sent: [%s]\n", buf);
    return OK;
}

```

Результат

\$./server



```

nguyensang@K-virtual-machine:~/Desktop/OS2021/lab8/part_01$ ./server
Server is listening.
To stop server press Ctrl + C.

```

\$./Client

```
nguyensang@K-virtual-machine:~/Desktop/OS2021/lab8/part_01$ ./client
Client sent: [This Message From 9820]
nguyensang@K-virtual-machine:~/Desktop/OS2021/lab8/part_01$ ./client
Client sent: [This Message From 9822]
nguyensang@K-virtual-machine:~/Desktop/OS2021/lab8/part_01$ █
```

```
nguyensang@K-virtual-machine:~/Desktop/OS2021/lab8/part_01$ ./server
Server is listening.
To stop server press Ctrl + C.
Server read: [This Message From 9820]
Server read: [This Message From 9822]
^Cnguyensang@K-virtual-machine:~/Desktop/OS2021/lab8/part_01$
```