

**Міністерство освіти та науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет прикладної математики
Кафедра системного програмування і спеціалізованих
комп'ютерних систем**

Лабораторна робота №7

**з дисципліни
«Комп'ютерні мережі»
«Обмен дейтаграммами»**

**Виконали студенти четвертого курсу
групи КВ-32**

**Гудіков Владислав
Непокритий Микола
Коваль Андріян**

**м. Київ
2016**

Задание:

- 1) Изучить изложенный материал.
- 2) Набрать примеры и показать преподавателю.
- 3) Быть готовым ответить на вопросы по изложенному материалу.

Код программы:

```
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <netinet/in.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <algorithm>
#include <set>
using namespace std;

int main()
{
    int listener;
    struct sockaddr_in addr;
    char buf[1024];
    int bytes_read;

    listener = socket(AF_INET, SOCK_STREAM, 0);
    if(listener < 0)
    {
        perror("socket");
        exit(1);
    }

    fcntl(listener, F_SETFL, O_NONBLOCK);

    addr.sin_family = AF_INET;
    addr.sin_port = htons(3425);
    addr.sin_addr.s_addr = INADDR_ANY;
    if(bind(listener, (struct sockaddr *)&addr, sizeof(addr)) < 0)
    {
        perror("bind");
        exit(2);
    }

    listen(listener, 2);

    set<int> clients;
    clients.clear();

    while(1)
    {
        // Заполняем множество сокетов
        fd_set readset;
```

```

FD_ZERO(&readset);
FD_SET(listener, &readset);

for(set<int>::iterator it = clients.begin(); it != clients.end(); it++)
    FD_SET(*it, &readset);

// Задаём таймаут
timeval timeout;
timeout.tv_sec = 15;
timeout.tv_usec = 0;

// Ждём события в одном из сокетов
int mx = max(listener, *max_element(clients.begin(), clients.end()));
if(select(mx+1, &readset, NULL, NULL, &timeout) <= 0)
{
    perror("select");
    exit(3);
}

// Определяем тип события и выполняем соответствующие действия
if(FD_ISSET(listener, &readset))
{
    // Поступил новый запрос на соединение, используем ассепт
    int sock = accept(listener, NULL, NULL);
    if(sock < 0)
    {
        perror("accept");
        exit(3);
    }

    fcntl(sock, F_SETFL, O_NONBLOCK);

    clients.insert(sock);
}
for(set<int>::iterator it = clients.begin(); it != clients.end(); it++)
{
    if(FD_ISSET(*it, &readset))
    {
        // Поступили данные от клиента, читаем их
        bytes_read = recv(*it, buf, 1024, 0);

        if(bytes_read <= 0)
        {
            // Соединение разорвано, удаляем сокет из множества
            close(*it);
            clients.erase(*it);
            continue;
        }
        // Отправляем данные обратно клиенту
        send(*it, buf, bytes_read, 0);
    }
}

return 0;
}

```