

<b>Politechnika Świętokrzyska</b> Wydział Elektrotechniki, Automatyki i Informatyki		
Programowanie Usług Sieciowych - laboratorium		
Laboratorium 6	Mateusz Hupa	Grupa: 1ID21A

**Celem laboratorium było zrealizowanie klienta i serwera przesyłających film.**

**Serwer:**

```
#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <arpa/inet.h>

#define SIZE 1000

int main(){

    char *ip = "127.0.0.1";

    int port = 8080;

    int server_sockfd;

    struct sockaddr_in server_addr, client_addr; int e;

    server_sockfd = socket(AF_INET, SOCK_DGRAM, 0); if (server_sockfd < 0){

        perror("socket error");

        exit(1);

    }

    server_addr.sin_family = AF_INET;

    server_addr.sin_port = port;
```

```

server_addr.sin_addr.s_addr = inet_addr(ip);

e = bind(server_sockfd, (struct sockaddr*)&server_addr, sizeof(server_addr)); if (e < 0){
    perror("bind error");
    exit(1);
}

FILE *fp;

char *filename = "/home/mateusz/Pulpit/pus/lab6/lab6_serwer2/serwer.mp4";

int n;

char buffer[SIZE];

socklen_t addr_size;

char size[256] = "";

int size2;

addr_size = sizeof(client_addr);

recvfrom(server_sockfd, size, sizeof(size), 0, (struct sockaddr*)&client_addr, &addr_size);

size2 = atoi(size);

fp = fopen(filename, "wb");

for (int i = 0; i <= SIZE; i++)
{
    buffer[i] = '\0';
}

for (n = 0; n < size2; n++)
{
    recvfrom(server_sockfd, buffer, sizeof(buffer), 0, (struct sockaddr*)&client_addr,
&addr_size);

    printf("%s \n", buffer);

    for(int i = 0; i< SIZE; i++){

```

```

        char buffer4[1];

        buffer4[0]=buffer[i];

        fwrite(buffer4, 1, 1, fp);
    }

    for (int i = 0; i <= SIZE; i++)
    {
        buffer[i] = '\0';
    }
}

fclose(fp);

printf("koniec\n");

close(server_sockfd);

return 0;
}

```

## **Klient:**

```

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <math.h>

#define SIZE 1000

int main(){

    char *ip = "127.0.0.1";

```

```

int port = 8080;

int server_sockfd;

struct sockaddr_in server_addr;

FILE *fp;

char *filename = "/home/mateusz/Pulpit/pus/lab6/lab6_klient2/klient.mp4";

server_sockfd = socket(AF_INET, SOCK_DGRAM, 0); if (server_sockfd < 0){
    perror("socket error");
    exit(1);
}

server_addr.sin_family = AF_INET;

server_addr.sin_port = port;

server_addr.sin_addr.s_addr = inet_addr(ip);

fp = fopen(filename, "rb");

if (fp == NULL){
    perror("fopen");
    exit(1);
}


int n;

char buffer[SIZE];

int i = 0;

fseek(fp, 0L, SEEK_END);

int sz = ftell(fp);

double    size3 = ceil(sz/SIZE);

int size4 = (int)size3;

fseek(fp, 0, SEEK_SET);

char size[256] = "";

```

```

    snprintf(size, sizeof size, "%d", size4);

    sendto(server_sockfd, size, sizeof( size ), 0, (struct sockaddr*)&server_addr,
sizeof(server_addr ));

    for (int i = 0; i <= SIZE; i++)

    {

        buffer[i] = '\0';

    }

    while(fread(buffer, sizeof( char ), SIZE, fp) != NULL){

        n = sendto(server_sockfd, buffer, sizeof( buffer ), 0, (struct sockaddr*)&server_addr,
sizeof(server_addr));

        if (n == -1){

            perror("error sendto");

            exit(1);

        }

        usleep(1000);

        for (int i = 0; i <= SIZE; i++)

        {

            buffer[i] = '\0';

        }

    }

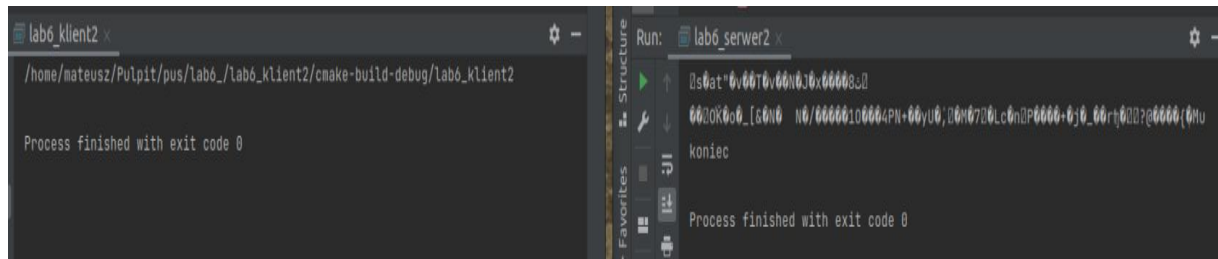

    close(server_sockfd);

    return 0;

}

```

## Wyniki działania programów:



The image shows two terminal windows side-by-side. The left window, titled 'lab6\_klient2', displays the path '/home/mateusz/Pulpit/pus/lab6/lab6\_klient2/cmake-build-debug/lab6\_klient2' and the message 'Process finished with exit code 0'. The right window, titled 'lab6\_serwer2', shows a large block of hexadecimal data, the word 'koniec' (end), and 'Process finished with exit code 0'. A vertical toolbar with icons for Structure, Favorites, and other IDE functions is visible between the two windows.

```
lab6_klient2 x
/home/mateusz/Pulpit/pus/lab6/lab6_klient2/cmake-build-debug/lab6_klient2

Process finished with exit code 0

Run: lab6_serwer2 x
D50at*0007000030x000080
0000K000_00N0 N0/00000100004PN+00yU0;00N0720Lc0n0P0000+0j0_00rt000?00000(0Mu
koniec

Process finished with exit code 0
```

