

<p style="text-align: center;">Politechnika Świętokrzyska Wydział Elektrotechniki, Automatyki i Informatyki</p>		
<p style="text-align: center;">Programowanie Usług Sieciowych - laboratorium</p>		
Laboratorium 4	Mateusz Hupa	Grupa: 1ID21A

Laboratorium 4

Aby zaliczyć laboratorium 4 należy:

- Zmodyfikować aplikację serwera z poprzednich zajęć tak aby działała współbieżnie.
- Zabezpieczyć programy przed błędami funkcjonowania.

Napisane programy:

- serwer:

```
#include <iostream>
#include <netinet/in.h>
#include <sys/wait.h>
#include <cstring>
#include <err.h>
#include <unistd.h>

#define MAXLINE 4096
struct sockaddr_in serv_addr, cli_addr;

void sig_chld(int signo){
    pid_t pid;
    int stat;
    while((pid = waitpid(-1,&stat, WNOHANG))>0)
        printf("Dziecko %d zatrzymane\n", pid);
}

int main() {

    char recvline [MAXLINE +1];
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0){
        perror("socket");
    }

    bzero(&serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);
    serv_addr.sin_port = htons(4000);
    if(bind(sockfd, (struct sockaddr *) &serv_addr, sizeof(serv_addr)) < 0){
        perror("bind");
    }

    if(listen(sockfd, 24) < 0){
        perror("listen");
    }

    signal(SIGCHLD, sig_chld);

    int connfd;

    for(;;){

        socklen_t clen = sizeof(cli_addr);
```

```

if(connfd = accept(sockfd, (struct sockaddr *) &cli_addr, &clilen))
{
    if(errno==EINTR)
        continue;
    else
        perror("accept");
}

int dzieckopid, n;
if((dzieckopid=fork()) == 0){
    close(sockfd);
    while (n = read(connfd, recvline, MAXLINE) > 0) {
        printf("%s\n", recvline);

        char resline[strlen(recvline)];
        for(int t=0;t<strlen(recvline);t++){
            resline[t]=recvline[strlen(recvline)-t-1];
        }
        resline[strlen(recvline)] = '\0';
        printf("\nPó odwróceniu: %s\n", resline);
        write(connfd,resline, strlen(resline));

    }
    if(n<0){
        perror("read");
    }
}

}

}

```

- klient:

```

#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#define MAXLINE 4096

struct sockaddr_in serv_addr;

int main() {

    char recvline [MAXLINE +1];
    int i = 0;
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    bzero(&serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;

    inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr);
    serv_addr.sin_port = htons(4000);

    connect(sockfd, (struct sockaddr *) &serv_addr, sizeof(serv_addr));

    //pętla do zadania A
    /*
    while(1){
        char buff[100];
        bzero(buff,sizeof(buff));
        printf("Wprowadz dane w postaci np.: a<spacja>+<spacja>b\n");
        scanf("%s", buff);
        write(sockfd,buff, strlen(buff));
    }
    */
}

```

```

    }

    /*
    //pętla do zadania B
    /*
    while(1){
        char buff[100];
        bzero(buff,sizeof(buff));

        if(i == 0){
            printf("Wprowadz liczbe a \n");
            scanf("%s", buff);
            write(sockfd,buff, strlen(buff));

            if(i == 1){
                printf("Wprowadz liczbe b \n");
                scanf("%s", buff);
                write(sockfd,buff, strlen(buff));

                if(i == 2){
                    printf("Wprowadz liczbe dzialanie: +, -, * \n");
                    scanf("%s", buff);
                    write(sockfd,buff, strlen(buff));
                    i = -1;

                    if(read(sockfd, recvline, MAXLINE) > 0) {
                        printf("%d \n", recvline[0]);
                    } else {
                        perror("read");
                    }
                }

                i++;
            }
        }
    }

    //pętla do zadania C

    while(1){
        char buff[100];
        bzero(buff,sizeof(buff));
        printf("Wprowadz dane \n");
        scanf("%s", buff);
        write(sockfd,buff, strlen(buff));

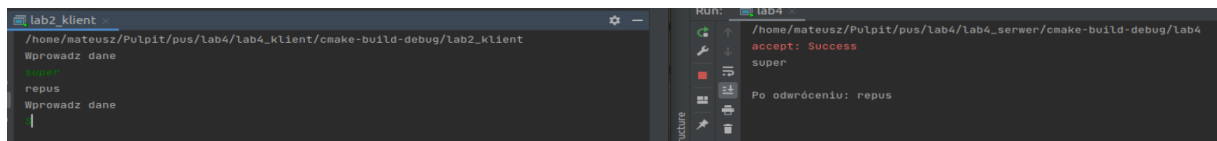
        if(read(sockfd, recvline, MAXLINE) > 0) {
            printf("%s \n", recvline);
        } else {
            perror("read");
        }

    }

}

```

Działanie aplikacji:



The screenshot displays the CMake GUI interface with two panels. The left panel, titled 'lab2_client', shows the source directory as '/home/mateusz/Pulpit/pus/lab4/lab4_klient/cmake-build-debug/lab2_client'. It lists the source files 'Wprowadz_dane.cpp' and 'repus.cpp', and shows the command 'Wprowadz_dane' being executed. The right panel, titled 'lab4', shows the build directory as '/home/mateusz/Pulpit/pus/lab4/lab4_serwer/cmake-build-debug/lab4'. It displays the output of the build process, including the message 'accept: Success' and 'super', and the command 'Po odróceniu: repus'.

```
lab2_client
/home/mateusz/Pulpit/pus/lab4/lab4_klient/cmake-build-debug/lab2_client
Wprowadz_dane
repus
Wprowadz_dane

lab4
/home/mateusz/Pulpit/pus/lab4/lab4_serwer/cmake-build-debug/lab4
accept: Success
super
Po odróceniu: repus
```