Politechnika Świętokrzyska Wydział Elektrotechniki, Automatyki i Informatyki		
Programowanie Usług Sieciowych - laboratorium		
Laboratorium 5	Mateusz Hupa	Grupa: 1ID21A

Celem laboratorium było napisanie programów: serwera współbieżnego oraz klienta, wykorzystujących do komunikacji funkcję select().

Serwer:

```
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>
#include <unistd.h>
#include <stdlib.h>

#define MAXLINE 4096
#define MAX 4096
#define LISTENQ 1024
#define SIZE 10

struct sockaddr_in serv_addr, cli_addr;

void err_sys(char *s){
    printf("%s\n",s);
    exit(-1);
```

```
}

int main() {

    fd_set rset, allset;
   // FD_ZERO(&rset);
    socklen_t clilen;
    char recvline [MAXLINE +1] = {0};
    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");
    }

    int n, connfd, i, d, e;
    float a, b, c;
```

```
int client[10] ={-1,-1,-1,-1,-1,-1,-1,-1,-1,-1};
int maxfd = sockfd;
FD_ZERO(&allset);
FD_SET(sockfd, &allset);

for (;;) {

    rset=allset;
    select(maxfd+1, &rset, NULL, NULL, NULL);

    if(FD_ISSET(sockfd, &rset)){
        clilen = sizeof(cli_addr);
        bzero(recvline, MAXLINE);
        if(connfd = accept(sockfd, (struct sockaddr *) NULL, NULL))
        {
            perror("accept");
        }

        for(i=0; i<SIZE; i++)
            if(client[i]<0){
                client[i]=connfd;
                break;
            }
        if(i==SIZE){
            err_sys("za duzo klientow");
        } else {
            if(connfd>maxfd)
```

```
            maxfd = connfd;
        FD_SET(connfd,&allset);
    }
}



for(i=0;i<=maxfd; i++){

    if((sockfd=client[i])<0)
        continue;

    if(FD_ISSET(sockfd, &rset)){
        if(n = read(connfd, recvline, MAXLINE) <= 0){

            close(sockfd);
            FD_CLR(sockfd,&allset);
            client[i]=-1;

        }else
        {
            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("\nPo odwróceniu: %s\n", resline);
                write(connfd,resline, strlen(resline));
            }
        }
    }
}
```

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");
        }
    }
}
```

Wyniki działania programów: