

JEGYZŐKÖNYV

Operációs rendszerek BSc
2021 tavasz féléves feladat

Készítette: Simon Dániel

Neptunkód:XUE9MH

A feladat leírása:

Írjon egy olyan C programot, ami

egy bemeneti fájlból 3 adatot olvas ki (háromszög oldalai) és eldönti, hogy szerkeszthető e belőlök háromszög.

A feladat megoldása során használjon message queue(üzenetsoros mechanizmust), valamint a kimenet kerüljön egy másik fájlba. Ha szerkeszthető belőlük háromszög adjon vissza 1-et, különben pedig 0-t.

A ki/bemeneti fájl struktúrája kötött!

Példa a bemeneti és kimeneti fájl struktúrájára:

Bemeneti fájl:

x y z

Kimeneti fájl(A q jelzi a visszatérési értéket, tehát hogy szerkeszthető e háromszög):

x y z q

A feladat elkészítésének lépései:

Értelmeztem a feladatot majd kerestem hasonló programokat. Ezek segítségével nekikezdttem programozni.

Először include-oltam a szükséges header-eket

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/msg.h>
#include <errno.h>
#include <fcntl.h>
```

Ezután define-oltam konstansokat

```
#define INPUT_FILE "input_file"
#define OUTPUT_FILE "output_file"

#define KEY 26061998 //egyedi kulcs az uzenetsor létrehozásához
#define FLAG 0626 //hozzaferesi jogosultságokat reprezentalo flag
```

Majd definiáltam két struktúrát

```
// a haromszog adatainak egyszeru tarolasara hasznalt struktura
struct triangle
{
    int a;
    int b;
    int c;
};

//a uzenetsoron kuldendo uzenet struktura
struct message {
    long mtype;
    int szam;
};
```

Ezek után megírtam a szükséges függvényeket

```
//beolvas egy sor karaktert a megadott fajlbol a megadott meretu megadott bufferbe
int readLine(int file, char* buff, int size)
{
    char ch=' ';
    int i = 0;
    while(ch != '\n' && ch!='\0' && i < size - 1)
    {
        if( read(file,&ch,1) <= 0)
        {
            return -1;
        }
        else if( ch != '\n')
            buff[i++] = ch;
    }
    buff[i] = '\0';
    return i;
}

//leellenorzi hogy megszerkesztheto-e a haromszog
int megszerkesztheto (struct triangle hszg)
{
    return (int)!( hszg.a + hszg.b <= hszg.c || hszg.a + hszg.c <= hszg.b || hszg.b + hszg.c <= hszg.a );
}
```

Majd megírtam a main() függvényt

```

int main()
{
    int input, output, mqid;
    struct triangle hszg;
    struct message msg;
    struct msqid_ds statbuf; // uzenetsor adatait reprezentalo struktura

    //kinyitom az input file-t es leellenorzom hogy sikeresen megnyitottam-e, ha nem akkor kilep a program
    input = open(INPUT_FILE, O_RDONLY);
    if( input < 0 )
        perror("open input"), exit( EXIT_FAILURE );

    //beolvasok egy sort az input-bol
    char buff[10];
    readLine( input, buff, sizeof(buff));

    //letrehozom az uzenetsort es leellenorzom hogy sikeresen létrejott-e, ha nem akkor kilep a program
    mqid = msgget(KEY, FLAG | IPC_CREAT);
    if( mqid < 0 )
        perror("msgget"), exit(EXIT_FAILURE);

    //beallitom a kuldendo uzenetet
    msg.mtype = 2;
    //kilvasom az uzenetet a bufferbol a haromszog strukturaba
    sscanf(buff, "%d %d %d", &hszg.a, &hszg.b, &hszg.c);
    //megnezem hogy megszerkesztheto-e es az eredmenyt beiro az uzenet strukturaba
    msg.szam = megszerkesztheto(hszg);
    //elkuldom az uzenetsorra az uzenetet
    msgsnd(mqid, &msg, sizeof(struct message), 0);
    msgctl(mqid, IPC_STAT, &statbuf);

    //kiolvasom az uzenetet
    msgrcv(mqid, &msg, sizeof(struct message), 0, 0);
    //bezaro az inputot
    close(input);
    //megnyitom az outputot es ellenorzom, hogy sikeresen megnyitottam-e, ha nem akkor kilep a program
    output = open( OUTPUT_FILE, O_CREAT | O_WRONLY | O_TRUNC, 0644);
    if( output < 0 )
        perror("open output"), exit(EXIT_FAILURE);

    //letrehozok egy uj valtozot amit feltoltok a kimenet adataival (haromszog oldalai es szerkeszthetoseg binarisan)
    char out[15];
    sprintf(out, "%d %d %d %d", hszg.a, hszg.b, hszg.c, msg.szam);
    //kiro az outputra a kimeneti valtozo tartalmat
    write(output, out, strlen(out));

    //bezaro a kimenetet, uzenetsort es kilepek a programbol
    close(output);
    msgctl(mqid, IPC_RMID, NULL);
    exit(EXIT_SUCCESS);
}

```

A program kommentezve van, amik leírják a program működését, így ezt itt nem részletezem.

A futtatás eredménye:

input_file tartalma:

```
GNU nano 2.7.4 File: input_file  
2 3 4
```

program futtatása:

```
simon28@jerry:~/oprend/second_try/beadando$ gcc ellenorzo.c -o hszg  
simon28@jerry:~/oprend/second_try/beadando$ ./hszg  
simon28@jerry:~/oprend/second_try/beadando$
```

output_file tartalma:

```
GNU nano 2.7.4 File: output_file  
2 3 4 1
```