

JEGYZŐKÖNYV

Operációs rendszerek BSc

2021. tavasz féléves feladat

Készítette: **Rontó Eszter**

Neptun kód: **QTFL19**

Feladat leírás

Írjon egy olyan C programot, ami egy bemeneti fájlból 3 adatot olvas ki (háromszög oldalai) és eldönti, hogy szerkeszthető-e belőlük 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 a háromszög): **x y z q***

Feladat elkészítésének lépései

Felvettem a szükséges header fájlokat.

A **#define**-nal létre hozom a kulcsot, ami az üzenetsor azonosítója lesz.

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <string.h>
#include <stdlib.h>
#define MSGKEY 6543L
#define MSGSZ 4

struct msgbuf {
    long mtype;
    int mtext[MSGSZ];
}sendbuf, *msg;
```

Ezek után elkészítem az üzenetsoros kommunikációhoz szükséges struktúrát. Amely tartalmazza az **mtype** változót (ami a különböző üzenettípusokkal történő kommunikációra szolgál) és **mtext** változó tömböt (amelynek méretét a korábbiakban definiált **MSGSZ** pozitív érték határozza meg).

```
void main()
{
    FILE* fp;
    FILE* ki;

    int x,y,z;
    int q;

    int msgid;
    key_t key;
    int msgflg;
    int ret;
```

Következőleg a **main** függvényen belül deklaráltam

- az **fp** és **ki** változókat, amelyekkel a későbbiekben hivatkoztam a két fájlra
- az **x**, **y**, és **z** változókat, amik az **fp** fájlból fogják megkapni az értékeiket (a három oldal hosszát)
- a **q** változót, amely meg fogja kapni az (1 || 0) értéket
- az **msgid**-t ami az üzenetsor azonosítója lesz
- a **key**-t ami a kulcs az üzenetsorhoz lesz
- az **msgflg** változót, amely az üzenetküldéséhez szükséges
- a **ret** változót, amely a return értékhez kell

```
key = MSGKEY;
msgflg = 00666 | IPC_CREAT;
msgid = msgget(key, msgflg);
if (msgid == -1)
{
    perror("\nAz msgget rendszerhívás sikertelen.");
    exit(-1);
}
```

A **key** egyenlő lesz a korábban definiált kulccsal.

Az **msgflg** meg kapja a jogosultságot és létre hozza a sort, ha még nem létezik.

Az **msgget** függvénnyel lekértem az üzenetsor azonosítóját, ami egyenlő lesz az **msgid** változóval, aminek ha az értéke -1 akkor a lekérdezés sikertelen volt.

```
msg = &sendbuf;

fp = fopen("haromszog.txt", "r");
if(fp == NULL)
{
    perror("Nem sikerült a fájlmegnyitás.");
}

fscanf(fp, "%d %d %d", &x, &y, &z);
printf("Az oldalak hossza: \n x = %d \n y = %d \n z = %d \n", x, y, z);
fclose(fp);
```

A **sendbuf** címet az **msg** pointerhez rendeltem.

A **fopen** paranccsal megnyitottam a már létező állományomat (r miatt) olvasásra. Hibakezelés képpen írtam egy feltételt, ami kezeli, ha valamilyen oknál fogva nem tudta meg nyitni az állományt a program.

Ha azonban meg tudta nyitni az állományt, akkor tovább lép ahol is létre hoztam egy **fscanf**-et, ami beolvassa a jelenlegi 3 adatot a fájlból és ezeket át adja a programon belül definiált integer változóknak.

Ön ellenőrzés miatt **printf**-fel kiírtattam a nem rég beolvasott méreteket, majd **fclose** paranccsal bezártam az állományt.

```

if((x+y)>z && (y+z)>x && (x+z)>y)
{
    q = 1;
    printf("\nEzekből az oldal méretekből szerkeszthető háromszög.\n");
}
else
{
    q = 0;
    printf("\nEzekből az oldal méretekből NEM szerekeszthető háromszög.\n");
}

```

A fő feladatomban az volt, hogy ki számoljam, hogy szerkeszthető-e a megadott három oldal hosszából egy háromszög. Ezt egy `if`-fel ellenőriztem ahol a három oldalra vonatkozó összefüggésektől függően a `q` változó meg kapta az értékét.

```

ki = fopen("visszateres.txt", "w");

msg->mtype = 1;
msg->mtext[0] = x;
msg->mtext[1] = y;
msg->mtext[2] = z;
msg->mtext[3] = q;

ret = msgsnd(msgid, (struct msgbuf *)msg, MSGSZ, msgflg);
//printf("\nA vissza kapott értékek : ");
for(int i = 0; i < MSGSZ; i++)
{
    fprintf(ki, "%d ", msg->mtext[i]);
    // printf("%d ", msg->mtext[i]);
}
fclose(ki);
}

```

A `fopen` paranccsal megnyitottam a már létező állományomat (w miatt) írásra. Meg adtam az üzenetsor típusát majd feltöltöttem az `mtext` tömb elemeit a három oldal hosszal (`x`, `y`, `z`) és a `q` változóval (`1 || 0`).

Ezek után az `msgsnd` függvénnnyel elküldtem az üzenetet az `msgid` által megadott üzenetsor-azonosítójához társított sorba és az `fprintf` paranccsal és egy `for` ciklussal át adtam az üzenetet a kimeneti fájlba, majd lezártam.

Futtatás eredménye

Első eredmény:

```
reszter@reszter-VirtualBox: ~/Asztal/felev
Fájl Szerkesztés Nézet Keresés Terminál Súgó
reszter@reszter-VirtualBox:~/Asztal/felev$ nano 18.c
reszter@reszter-VirtualBox:~/Asztal/felev$ gcc -o 18 18.c
reszter@reszter-VirtualBox:~/Asztal/felev$ ./18
Az oldalak hossza:
x = 3
y = 8
z = 2
Ezekből az oldal méretekből NEM szerekeszthető háromszög.
reszter@reszter-VirtualBox:~/Asztal/felev$
```

```
haromszog.txt (~/Asztal/felev)
Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó
3 8 2
Egyszerű szöveg Szóközök: 4 1. sor, 6. oszlop BESZ

visszateres.txt (~/Asztal/felev)
Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó
3 8 2 0
Egyszerű szöveg Szóközök: 4 1. sor, 9. oszlop BESZ
```

Második eredmény:

```
reszter@reszter-VirtualBox: ~/Asztal/felev
Fájl Szerkesztés Nézet Keresés Terminál Súgó
reszter@reszter-VirtualBox:~/Asztal/felev$ ./18
Az oldalak hossza:
x = 3
y = 3
z = 3

Ezekből az oldal méretekből szerkeszthető háromszög.
reszter@reszter-VirtualBox:~/Asztal/felev$
```

```
haromszog.txt (~/Asztal/felev)
Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó
3 3 3

visszateres.txt (~/Asztal/felev)
Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó
3 3 3 1 |
```