

JEGYZŐKÖNYV

Operációs rendszerek BSc
2022. tavasz féléves feladat

Készítette:

Ónodi Bence BSC

Programtervező
informatikus

RYSNLC

Miskolc, 2022

A feladat leírása:

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 nevesített csővezetékot, 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!

Bemeneti fájl: x y z

Kimeneti fájl (q – visszatérési érték): x y z q

A feladat elkészítésének lépései:

Egy olyan programot kellett írnom, ami egy fájlból olvas be 3 adatot, a háromszög oldalait, és ezek alapján eldönti, hogy szerkeszthető-e belőlük háromszög. A változók deklarációja után létrehozom a nevesített csővezetékot, majd megnyitom a külső fájlt, amiből az adatokat kapja. Az adatok beolvasása után ellenőrzöm a háromszög egyenlőtlenségét, elseif szerkezettel, a visszatérési érték pedig 1 lesz, ha szerkeszthető, 0 ha nem. Ha megvan az eredmény, akkor azt hozzáillesztem a külső fájlhoz. Ez volt az első tesztvezeték. Megnyitom a második tesztvezetékot amibe a kiszámolt eredményt beleírom, majd lezárom.

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <fcntl.h>
4  #include <sys/stat.h>
5  #include <sys/types.h>
6  #include <unistd.h>
7  #include <stdlib.h>
8
9  int main()
10 {
11     int fd, fd2, x, y, z, flag;
12     char a=' ', b, c, str[7], q, space=' ';
13
14     char * teszt = "teszt.txt";
15     char * eredmeny = "eredmeny.txt";
16
17     mkfifo(teszt, 0666);
18     mkfifo(eredmeny, 0666);
19
20     fd = open(teszt, O_RDONLY);
21
22     read(fd, str, sizeof(str));
23
24     a=str[0];
25     x=atoi(&a);
26
27     b=str[2];
28     y=atoi(&b);
29
30
31     c=str[4];
32     z=atoi(&c);
33
34     if(x > y && x > z)
35     {
36         flag = ((y+z) > x);
37     }
38
39     else if(y > z)
40     {
41         flag = ((x+z) > y);
42     }
43
44     else
```

```
40 {
41     flag = ((x+z) > y);
42 }
43
44 else
45 {
46     flag = ((x+y) > z);
47 }
48
49 if(flag)
50 {
51     q='1';
52 }
53 else
54 {
55     q='0';
56 }
57
58 strncat(str, &space, 1);
59 strncat(str, &q, 1);
60
61 close (fd);
62
63 fd2 = open(eredmeny, O_CREAT | O_WRONLY);
64
65 write(fd2, str, strlen(str)+1);
66
67 close (fd2);
68
69 return 0;
70 }
```

A feladat leírása:

19. Adott egy számítógépes rendszer, melyben a

- szabad memória területek: 23KB, 64KB, 10KB, 80KB, 12Kb, 50KB és 40KB, melynek
- foglalási igénye: 65kB, 21kB, 48KB, 13kB, 62kB.

Határozza meg *változó méretű partíció* esetén a következő algoritmusok felhasználásával: *next fit*, *worst fit* a foglalási igényeknek megfelelő helyfoglalást – táblázatos formában!

Magyarázza a kapott eredményeket és hogyan lehet az eredményeket javítani!

A feladat elkészítésének lépései:

Next-fit: attól a memóriától kezdi vizsgálni hova mehet Px, amelyikbe már Py-t berakta.

Next-fit		23	64	10	80	12	50	40
P1	65	23	64	10	65, 15	12	50	40
P2	21	23	64	10	15	12	21, 29	40
P3	48	23	64	10	15	12	29	40
P4	13	23	64	10	15	12	29	13, 27
P5	62	23	64	10	15	12	29	27

Ennél az esetnél P3 és P5 nem tehető sehova.

Worst-fit: mindig a lehető legnagyobb memóriába teszi.
Dinamikus esetben ez a legjobb, mivel ugyanott még marad hely más dolognak is.

Worst-fit		23	64	10	80	12	50	40
P1	65	23	64	10	65, 15	12	50	40
P2	21	23	21, 43	10	15	12	50	40
P3	48	23	43	10	15	12	48, 2	40
P4	13	23	13, 30	10	15	12	2	40
P5	62	23	30	10	15	12	2	40

Ennél az esetnél P5 nem tehető sehova, várólistát kap.