

Mutatók használata 1.

Kormányos Andor

Komplex Rendszerek Fizikája Tanszék

2020. szeptember 21.

A következő példákat tekintjük:

- változók cím szerinti átadása függvénynek
- egy szám beolvasása file-ból

Miért lehet szükség egy változó memóriacímére?

- C-ben, amikor paramétert adunk át egy függvénynek, annak az **értéke** adódik át egy lokális másolatnak
- a függvény ezt a másolatot tudja átírni, az eredetit nem
- sokszor el akarjuk érni, hogy a függvények bele tudjanak írni a változókba

Ötlet:

- a változó értéke helyett adjuk át a változóra mutató memóriacímet
- ekkor a függvény be tudja írni az új értéket a változónak megfelelő memóriaterületre

Változó átadása függvénynek *cím szerint*

```
1 void increment(int a, int *b, int *c) {
2     *b = a + 5;
3     *c = a + 10;
4 }
5
6 int main() {
7     int a = 25;
8     int b, c;
9     increment(a, &b, &c);
10    printf("%d %d %d\n", a, b, c); // eredmény: 25 30 35
11    return 0;
12 }
```

- ez arra is jó, ha több “visszatérési értéke” van egy függvénynek
- a return utasítással csak egy értéket tudunk visszaadni

Változó átadása függvénynek *cím szerint*

```
1 void increment(int a, int *b, int *c) {  
2     *b = a + 5;  
3     *c = a + 10;  
4 }  
5  
6 int main() {  
7     int a = 25;  
8     int b, c;  
9     increment(a, &b, &c);  
10    printf("%d %d %d\n", a, b, c); // eredmény: 25 30 35  
11    return 0;  
12 }
```

- ez arra is jó, ha több “visszatérési értéke” van egy függvénynek
- a return utasítással csak egy értéket tudunk visszaadni

Feladat

Írjunk egy swap függvényt, amely pl egy tömb két elemét fel tudja cserélni!
Hogyan néz ki a main() függvényben ennek a swap függvénynek a hívása?

Itt most szöveges fájlokkal foglalkozunk

- ezekből is csak számokat olvasunk be
- a számok 10-es számrendben vannak tárolva

Egy fájl olvasásához vagy írásához

- a fájlt meg kell nyitni
- át kell tudni adni az olvasó/író függvénynek
- a végén a fájlt be kell zárni

A fájlra egy pointeren keresztül tudunk hivatkozni

- ennek a típusa mindig **FILE***

Példa: egyetlen szám beolvasása fájlból

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int main(int argc, char* argv[]) {
5     double a;
6     FILE* f = fopen(argv[1], "r");
7     fscanf(f, "%lf", &a);
8     fclose(f);
9     printf("%f\n", a);
10    return 0;
11 }
```

- A beolvasandó szám az `a` változóba kerül

Példa: egyetlen szám beolvasása fájlból

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int main(int argc, char* argv[]) {
5     double a;
6     FILE* f = fopen(argv[1], "r");
7     fscanf(f, "%lf", &a);
8     fclose(f);
9     printf("%f\n", a);
10    return 0;
11 }
```

- A beolvasandó szám az **a** változóba kerül
- Megnyitjuk a fájlt az **fopen** függvénnyel
 - a fájlnev az első parancssori paraméter
 - az **"r"** jelentése: olvasásra

Példa: egyetlen szám beolvasása fájlból

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int main(int argc, char* argv[]) {
5     double a;
6     FILE* f = fopen(argv[1], "r");
7     fscanf(f, "%lf", &a);
8     fclose(f);
9     printf("%f\n", a);
10    return 0;
11 }
```

- A beolvasandó szám az **a** változóba kerül
- Megnyitjuk a fájlt az **fopen** függvénnyel
 - a fájlnev az első parancssori paraméter
 - az **"r"** jelentése: olvasásra
- Fájlból olvasás **fscanf** függvénnyel

Példa: egyetlen szám beolvasása fájlból

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int main(int argc, char* argv[]) {
5     double a;
6     FILE* f = fopen(argv[1], "r");
7     fscanf(f, "%lf", &a);
8     fclose(f);
9     printf("%f\n", a);
10    return 0;
11 }
```

- A beolvasandó szám az **a** változóba kerül
- Megnyitjuk a fájlt az **fopen** függvénnyel
 - a fájlnev az első parancssori paraméter
 - az **"r"** jelentése: olvasásra
- Fájlból olvasás **fscanf** függvénnyel
- Bezárjuk a fájlt **fclose** paranccsal

Példa: egyetlen szám beolvasása fájlból

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int main(int argc, char* argv[]) {
5     double a;
6     FILE* f = fopen(argv[1], "r");
7     fscanf(f, "%lf", &a);
8     fclose(f);
9     printf("%f\n", a);
10    return 0;
11 }
```

Az `fscanf` függvény működése

- az első paraméter a fájl, amiből olvasunk
- a második egy formátumstring: double típus `%lf`
- Az utolsó paraméter egy **pointer** arra a memóriacímre, ahova a beolvasott számot tárolni szeretnénk
- az `fscanf` visszatérési értékét most nem használjuk
 - azt adja vissza, hogy hány számot sikerült beolvasni