

Mutatók használata 1. Változók átadása

Kormányos Andor

Komplex Rendszerek Fizikája Tanszék

2022. szeptember 27.

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?