

# Mutatók

Kormányos Andor

Komplex Rendszerek Fizikája Tanszék

2022. szeptember 27.

A processzor (elvileg) egységes memóriát lát

- a bájtok külön-külön címezhetők  $\Rightarrow$  memóriacímek
- 0-tól kezdődően a memória méretéig

A program szempontjából a memória több dologra is való:

- ebben található a programkód, amit a processzor végrehajt
- itt található a **verem (stack)**, ami pl a függvények paramétereinek átadására való
- a fennmaradó, szabadon **foglalható** memória a **halomterület (heap)**

Eddig egyszerű változókat használtunk

- egy változóban egyetlen számot tároltunk *valahol* a memóriában
- valójában ezek a változók a **veremben** tárolódnak
- a verem mérete általában véges<sup>1</sup>, spórolni kell vele

Kérdések, amelyekre megoldást keresünk:

- egyszerre sok számnak is kellhet memória (pl egy mátrixnak)
- egy nagy mátrix már nem biztos, hogy belefér a stack-be
- ha a vermen kívüli memóriára van szükség, azt **le kell foglalni**

---

<sup>1</sup>64 bites Linuxon alapértelmezésben 8MB

## Mutatók (Pointerek) a C-ben

- olyan változó, ami egy memóriacímet tartalmaz
- de van egy típusa is, mégpedig az, hogy az adott címen *milyen típusú* adat található
- alapesetben a stack-en foglal helyet

## Mutatók (pointer) a C nyelvben

Egy egyszerű példa:

```
1  int main() {
2      double a = 5.0;
3      double *dp;
4      // a * operatorral deklaraljuk a double tipusu dp pointert
5
6      dp= &a; // az & operatorral beallitjuk, hogy dp mely
7             // memoriacimre mutasson
8
9      printf("%p\n", dp); // eredmény: 0060FF00
10     printf("%f\n", *dp); // eredmény: 5.0
11
12     double b = *dp; //adott memoriacimre mutato pointer
13                   //erteke kiolvashato
14     printf("%f\n", b);
15
16     return 0;
17 }
```

A pointer által mutatott memóriacím felülírható:

```
1 double a = 5;
2 double *dp = &a;
3
4 *dp = 6;
5 // az a változó értéke most 6 lesz
```

A pointer által mutatott memóriacím felülírható:

```
1 double a = 5;
2 double *dp = &a;
3
4 *dp = 6;
5 // az a változó értéke most 6 lesz
```

**Figyelem!** Ez nem működik:

```
1 double *dp;
2
3 *dp = 5;
```

- mert a *dp* a deklaráció után nem mutat egy konkrét memóriacímre
- *dp*-t rá kell állítani egy memóriacímre az `&` utasítás segítségével