

# Assembly Programozás

## Nagy ZH (valahanyadik). csoport

---

### Általános információk

- A ZH-ra **45 perc + 5 perc** van.
- **FIGYELMESEN OLVASD EL A FELADATOT!**
- A feladat megoldása során választható, hogy az eljárás **ARM vagy Intel x86** assembly-ben kerül implementálásra.
- Az eljárást célszerű egy **megoldas.S** nevű fájlba elkészíteni.
- A **minta.zip**-ben található a C keret alkalmazás amivel lehetőség van tesztelni az eljárást.
- A biro-ra csak a assembly fájlt kell feltölteni.
- Az assembly-ben írt eljárás neve a feladatleírásban található. Ha nem karakterre pontosan egyezik a megoldás az elvárt névvel akkor 0 pont.
- Figyelni kell arra, hogy a tömbök milyen hosszúak! Ne legyen tömb túlírás!
- Figyelni kell az adatok méretére és előjelességére!
- A gyakorlati és előadás anyagok a [https://biro.inf.u-szeged.hu/kozos/assembly/assembly\\_anyag.zip](https://biro.inf.u-szeged.hu/kozos/assembly/assembly_anyag.zip) linkel elérhetőek

### Fordítási és futtatási segédlet

Intel x86 Linux:

```
1 $ gcc -m32 -static -g feladat.c megoldas.S -o program
2 $ ./program
```

---

ARM:

```
1 $ arm-linux-gnueabi-gcc -marm -mcpu=cortex-a7 ↵
    -static -g feladat.c megoldas.S -o program
2 $ qemu-arm-static ./program
```

---

### Gyakori exit kódok:

- **-8 SIGFPE**: floating point exception, aritmetikai számítás hiba.
- **-11 SIGSEG**: segmentation fault, tipikusan rossz címzés vagy eljáráshívási konvenciók be nem tartása (helló cdecl!).

**Feladat leírás: A következő oldalon!**

1. (15 pont) Adaptáljuk az alábbi C-s eljárást Intel vagy ARM 32 bites assemblyre:

---

```
1  int feladatX(int input[], int op, int opArg, int ↵
    output[]) {
2      int counter = 0;
3      for (int idx = 0; input[idx] != 0; idx++) {
4          int value = input[idx];
5
6          if (op == 0xbadc0ffe) {
7              if (value % opArg != 0) {
8                  output[counter] = value;
9                  counter++;
10             }
11         } else if (op == 0x11223344) {
12             if (value / 3 < opArg) {
13                 output[counter] = value;
14                 counter++;
15             }
16         } else {
17             output[counter] = -2048;
18             counter++;
19         }
20     }
21 }
22
23 return counter;
24 }
```

---

Az eljárás visszatérési értéke az az, hogy hány elemet másolt a kimeneti tömbbe.

**Az elkészítendő assembly eljárás C-s fejléce:**

---

```
1  int feladatX(int input[], int op, int opArg, int ↵
    output[]);
```

---