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 az assembly (.S) 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` linkel elérhetőek

Fordítási és futtatási segédlet

Intel x86 Linux:

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

ARM:

```
$ arm-linux-gnueabihf-gcc -marm -mcpu=cortex-a7 ↩
   -static -g feladat.c megoldas.S -o program
$ 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) Készítsünk egy **feladatCopyCount** nevű eljárást aminek négy paramétere van. Ezek a paraméterek az alábbiak:

    1. **input**: egy bemeneti 32 bites előjeles egész értékeket tartalmazó tömb.
    2. **length**: a bemeneti tömbben található elemek mennyisége (azaz a tömb "hossza").
    3. **limit**: egy 32 bites előjeles egész határérték.
    4. **output**: a kimeneti tömb, mely 32 bites előjeles egész értékeket tárol. Előre lefoglalt, nem kell allokálni neki helyet.

    Az eljárás visszatérési értéke egy 32 bites előjeles egész érték ami megadja, hogy hány elemet másolt az eljárás a kimeneti tömbbe.
    Az eljárás végig iterál a bemeneti tömb (**input**) elemein és megvizsgálja, hogy az akutális egész érték négyzete nagyobb-e mint a kapott határérték (**limit** paraméter). Amennyiben ez teljesül, akkor a kimeneti tömbbe (**output**) bemásoljuk az aktuális elem kétszeresét (nem a négyzetét!). A kimeneti tömbbe a feltételnek megfelelő értékek egymás után kell lenniük kihagyás nélkül.

    Az elkészítendő assembly eljárás C-s fejléce:

```
1 int feladatCopyCount(int input[], int length, int ↩
     limit, int output[]) {
```

    A megalósítandó eljárás beszeudó kódja

```
1  int feladatCopyCount(int input[], int length, int ↩
      limit, int output[]) {
2      count = 0;
3      for (idx = 0; idx < length; idx++) {
4          current = input[idx];
5          if ((current * current) > limit) {
6              output[count] = current * 2;
7              count++;
8          }
9      }
10
11     return count;
12 }
```