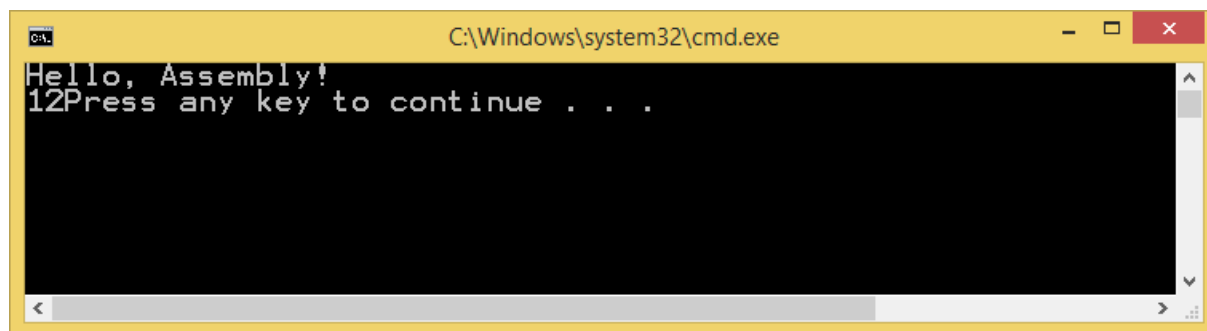


Irvine csomag használata

Az Irvine fejlesztőcsomag sok függvényt biztosít számunkra a könnyebb fejlesztés érdekében. Így például számok, sztringek kiírása képernyőre, vagy bekérése billentyűzetről, képernyőtörlés, sztring hosszának lekérése, késleltetés egy adott parancs végrehajtása előtt, stb. Ha a megvalósított függvényeket szeretnénk használni, nincs más teendőnk, mint be include-olni az *Irvine32.inc* fájlt, majd megfelelő módon meghívni az adott függvényt. A megvalósított függvényekkel részletesebben később foglalkozunk. Vegyük a következő programot (*HelloAsm.asm*) példaként (ne felejtjük el, hogy a sorszámozás nem része a programnak):

```
1: TITLE Hello, Assembly! (32 bites)
2: INCLUDE Irvine32.inc
3:
4: .data
5:   HelloTxt    BYTE    "Hello, Assembly!", 0ah, 0
6:   A           DWORD   5
7:
8: .code
9: ; Belépési pont
10: main PROC
11:   MOV     EDX, OFFSET HelloTxt
12:   CALL  WriteString
13:
14:   MOV     EAX, 7
15:   ADD     EAX, A
16:   CALL  WriteDec
17:
18:   INVOKE ExitProcess, 0
19: main ENDP
20:
21: END main
```

Az adatszégmensben egy sztring (*HelloTxt*, 5.sor) és egy duplaszó (*A*, 6.sor) számára történt területfoglalás. A *HelloTxt*-t a 12., míg az *A*-t a 16. sorban írjuk ki a képernyőre. Ennek megfelelően a képernyőn az alábbi kell, hogy megjelenjen:



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window contains the following text:

```
Hello, Assembly!
12Press any key to continue . . .
```

Feladatok:

1. Nyissuk meg a *HelloAsm.asm* fájlt a projektünkben. Fordítsuk, majd futtassuk le.
2. Lépkedjünk végig a programon debug módban, nyomkövetéssel.
 - a) Figyeljük meg, hogy melyik regiszter értéke hogyan változik a futás közben!
 - b) Figyeljük a terminált is, hogyan kerülnek egymás után végrehajtásra a kiíratások az adott függvényhívások hatására!
3. Helyezzünk el egy töréspontot(*breakpoint*) a *main* függvény valamely sorában, és csak attól a ponttól kezdve lépkedjünk végig nyomkövetéssel a programon. Ha már nem szeretnénk nyomkövetést alkalmazni:
 - a) Állítsuk le a program futását kézzel (Shift + F5), vagy
 - b) lépünk ki a debug módból és folytassuk úgy a program futtatását.
4. Helyezzünk valamilyen hibát szándékosan a kódba (pl. rontsuk el valahol egy parancselírással). Fordítsuk a programot.
 - a) Nézzük meg milyen hibaüzenetet kapunk.
 - b) Majd nézzük meg a lista fájlban is, hogy milyen visszajelzést kaptunk a hibáról.
5. Írjuk át az *A* változónak adott értéket hexadecimálisról binárisra vagy decimálisra (az átadott érték után elhelyezett **b** vagy **d** betű segítségével). Majd figyeljük meg, most miként változik az EAX regiszter értéke. Nagyobb számértékek használata esetén nem egyforma értéket látunk az EAX regiszterben megjelenítve, illetve a képernyőre kiírva, vagy mégis?
6. Módosítsuk az adatszegmensben létrehozott *A* változó előjelét. Nézzük meg milyen hatásai vannak ennek a műveletnek ha **DWORD** vagy ha **SDWORD** adattípusként hoztuk létre.
7. Az *Irvine32.inc* fájlba belenézve megláthatjuk, milyen függvényeket tudunk még használni, többek között kiíratásra is. Válasszunk más számrendszerben való kiíratást és nézzük meg, miként változik a kiírt eredmény.
8. A sztring szövegét változtatva figyeljük meg a Disassembly listában a memória címeket. Hogyan változnak a memória címek a sztring hosszát módosítva? (Elsősorban a *HelloTxt* és az *A* változó címeinek különbségét figyelembe véve.)