

# Assembly nyelv x86-os processzorokhoz

## Microsoft Visual Studio 2015 és MASM beállítása

### 32 bites alkalmazás készítéséhez szükséges előkészületek:

Telepítsük fel a Visual Studio2015 fejlesztőkörnyezetet. A Visual Studio 2015 tartalmazza a Microsoft Assembler (MASM) 14.0-es verzióját. Ezt mi magunk is leellenőrizhetjük az **ml.exe** fájl lokalizálásával a Visual Studio telepítési könyvtárán belül a **\vc\bin** mappában (Pl.: *c:\Program Files\Microsoft Visual Studio 12.0\vc\bin*).

Minden, SZTE Informatikai Tanszékcsoportnál tanuló hallgató jogosult Microsoft termékek igénylésére, így a Visual Studio2015 fejlesztőkörnyezet is elérhető minden hallgató számára, aki eleget tesz az említett követelménynek. Igényléssel kapcsolatos részletes információért mindenki keresse fel a gyakorlatvezetőjét. Licence igénylése persze nem kötelező. A *Visual Studio Express for Windows Desktop* ingyenes kiadásával is tökéletesen működik az assembly fordítás. Emellett a programok jó eséllyel működni fognak a Visual Studio korábbi kiadásaival is, tehát nem feltétlenül szükséges a legújabb verzió beszerzése.

### Példa programcsomag telepítése:

Az alábbi weboldalon érhető el a legújabb verziója a szükséges könyvtáraknak és példa programoknak:

<http://www.kipirvine.com/asm/examples/index.htm>

A weboldal megnyitása után válasszuk ki a legfelső, legutóbbi verziót:

[Example programs and link library source code for the Seventh Edition, using Visual Studio 2012 or 2013](#)

A kiválasztással letöltésre kerül a telepítendő programcsomag, egy **.msi** kiterjesztésű fájl formájában. A telepítést a **c:\Irvine** könyvtárba végezzük. Lehetőleg ne változtassunk ezen az elérési útvonalon. Nyomós indok esetén az elsőnek megjelölt weboldalon belül a *"Creating Project from Scratch"* címszó alatt lehet segítséget találni más útvonal beállításához.

## Visual Studio beállítása:

Az alábbi műveleteket csak egyszer kell elvégezni.

### A Debug menüben a "*Start Without Debugging*" hozzáadása:

Válasszuk ki a **Tools** menüpont alatt a **Customize** almenüpontot. Válasszuk a **Commands** fület. A **Menu bar** rádiógomb legyen kiválasztva, majd kattintsunk az **Add Command...** gombra. A **Categories** listából válasszuk ki a **Debug** elemet. A jobb oldalon megjelenő listából válasszuk a **Start Without Debugging** elemet, majd kattintsunk az **OK**, majd a **Close** gombra.

### C++ konfiguráció kiválasztása:

Válasszuk ki a **Tools** menüpont alatt az **Import and Export Settings** almenüpontot. Válasszuk az **Import selected environment settings**, majd továbblépve a **No, just import...** rádió gombot. A **Default Settings** listából válasszuk a **Visual C++** elemet (előfordulhat, hogy ilyen elem nem található, ebben az esetben a **General** elemet válasszuk, később visszatérünk erre az esetre). **Next**, majd **Finish** gomb választásával zárjuk be az ablakot.

### Tab méretének és a sorszámok megjelenítésének beállítása:

Utolsó kiegészítő beállításként, a jobb áttekinthetőség érdekében állítsuk át a Tabulátor méretét 5-re, illetve jelenítsük meg a sorok számát.

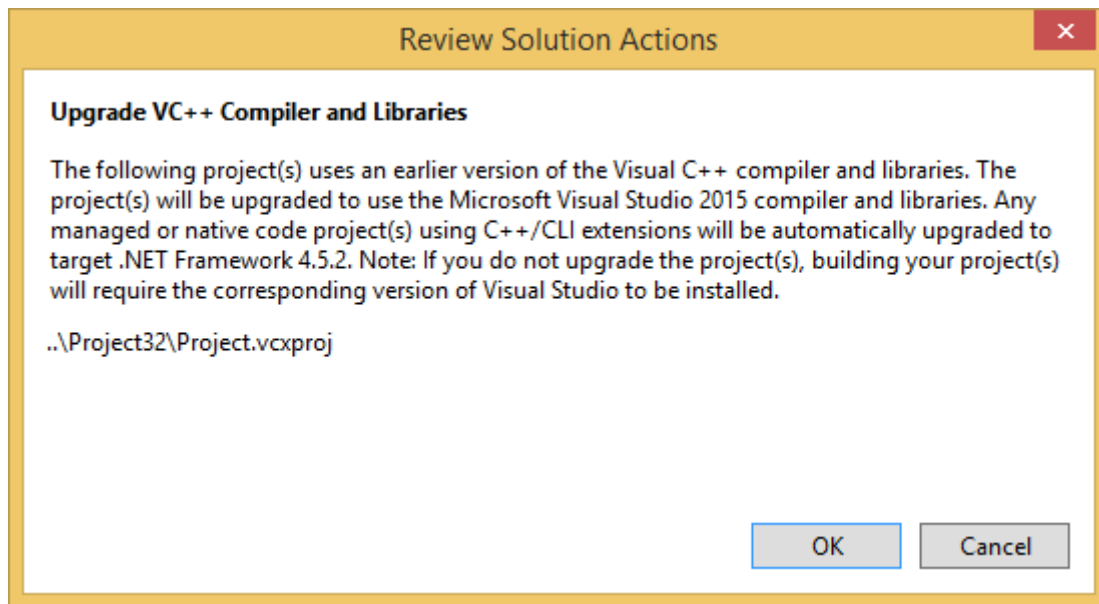
Válasszuk ki a **Tools** menüpont alatt az **Options** almenüpontot. A baloldali listában navigáljunk a **Text Editor** hierarchia ponton keresztül az **All Languages** elemhez. Látni fogunk egy checkbox-ot, mely mellett a **Line numbers** szöveg olvasható. Pipázzuk ki.

Ezután lépünk tovább az **All Languages** elemen belül a **Tabs** elemre. A **Tab Size** és **Indent Size** méreteket is állítsuk 5-re, majd az **OK** gomb lenyomásával lépünk ki.

A beállítások végeztével célszerű lehet a Visual Studio-n egy újraindítást végrehajtani.

## 32-bites Assembly program Build-elése:

Az előzőleg a `c:\Irvine` könyvtárba telepített programcsomagon belül található egy **Examples\Project32** mappa. A Visual Studio-ban válasszuk ki a **File/Open/Project** menüpontot. Navigáljunk a `c:\Irvine\Examples\Project32` mappához és válasszuk ki a **Project.sln** fájlt megnyitásra. Ekkor be fog tölteni a *Project.sln* fájlt a Visual Studio-ba. Előfordulhat, hogy az 1. ábrán látható üzenetet kapjuk.

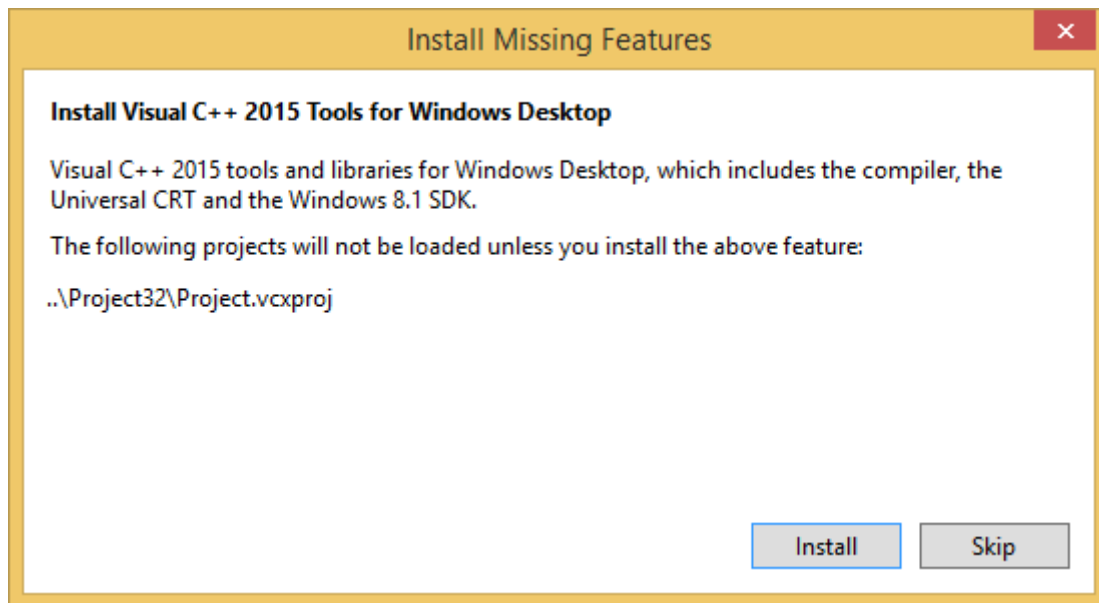


1. ábra Betöltött projekt frissítése

Ez annyit jelent, hogy a betölteni kívánt projekt egy előbbi verzióval készült és a Visual Studio felajánlja, hogy frissíti a projektet, hogy a VS 2015-ös verziója is felismerje. Fogadjuk el.

Ezután a **Solution Explorer**-ben meg is jelenik a betöltött program. Amennyiben a *Solution Explorer* nem látható, a *View/Solution Explorer* menüponton belül előhozható.

Abban az esetben, ha a C++ konfigurációkor a **General** elemet tudtuk csak kiválasztani, akkor a betöltött projekt nem használható. Ekkor a projekt mellett az *'unavailable'* felirat jelenik meg. A probléma kiküszöbölése egyszerű, kattintsunk jobb egérgombbal a betöltött projektre a *Solution Explorer*-ben, majd válasszuk ki az **Install Missing Feature(s)** elemet és telepítsük fel a *Visual C++ Tools*-t (2. ábra). Ezután töltsük be újra a projektet és folytassuk a következő lépésekkel.



2. ábra Visual C++ 2015 Tools telepítése

Következő lépésként töltsünk be egy kész Assembly program fájlt. A *Solution Explorer*-ben jobb kattintás a **Project**-re. Válasszuk ki az **Add**, majd az **Existing Item** menüpontot. Navigáljunk az *Examples\ch03* mappához és válasszuk ki az **AddTwo.asm** fájlt. Az **Add** gomb lenyomásával adhatjuk hozzá a projekthez a fájlt, majd csukjuk be az ablakot. Ekkor meg fog jelenni az *AddTwo.asm* fájl is a *Solution Explorer* ablakon belül.

Az alábbi programot fogjuk látni a szerkesztőben (*editor*):

```
; AddTwo.asm - addstwo 32-bit integers.  
; Chapter 3 example  
  
.386  
.model flat, stdcall  
.stack 4096  
ExitProcess proto, dwExitCode:dword  
  
.code  
main proc  
    mov     eax, 5  
    add     eax, 6  
  
    invoke ExitProcess, 0  
main endp  
end main
```

Az elkövetkezőkben használhatjuk ezt a programot kiinduló pontnak.

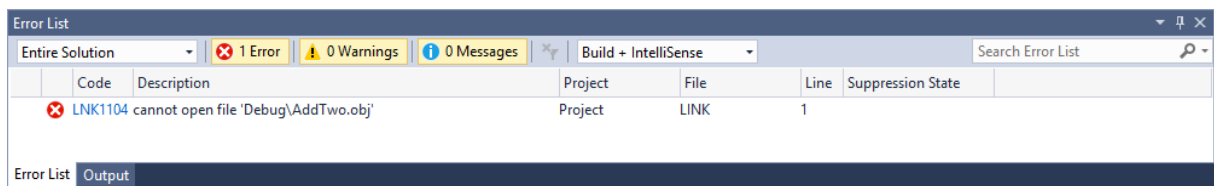
## Program build-elése:

Amennyiben a *short\_exercises.asm* nevű fájl megtalálható a *Solution Explorer*-en belül, azt mindenképpen töröljük ki onnan! Jobb kattintás a fájlra, majd *Remove*.

A **Build** menüponton belül válasszuk a **Build Project** almenüpontot. Az alábbi üzenethez hasonló szövegnek kell megjelennie az **Output** ablakban (valószínűleg a képernyő alján):

```
1>----- Building started: Project: Project, Configuration: Debug Win32 -----
1>Assembling ..\ch03\AddTwo.asm...
1>Project.vcxproj -> C:\Irvine\Examples\Project32\Debug\Project.exe
===== RebuildAll: 1 succeeded, 0 failed, 0 skipped =====
```

Abban az esetben, ha be van kapcsolva a fordítási lista készítése (7. oldalon tárgyaljuk), akkor előfordulhat, hogy az alábbihoz hasonló hibát kapunk:

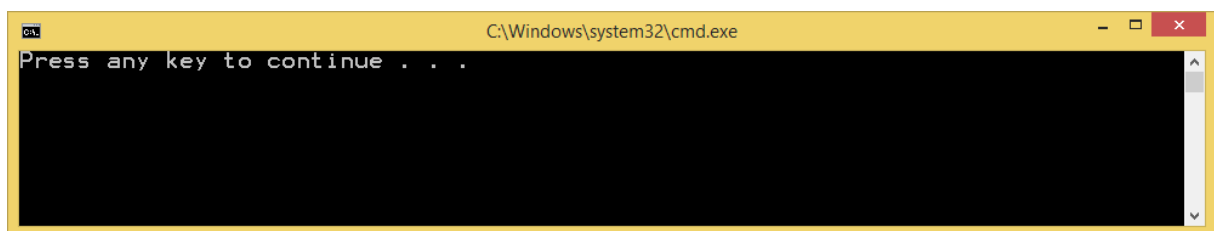


### 3. ábra Object fájl elérésére vonatkozó hibakód

Ekkor két megoldási lehetőség van. Az egyik lehetőség az, hogy a fordítandó forráskódot másoljuk be az *Irvine/Examples/Project32* mappába, majd innen adjuk hozzá a projekthez, eltávolítva az előzőt. Másik lehetőség, hogy kapcsoljuk ki a listafájl generálását.

## Program futtatása:

A **Debug** menüponton belül válasszuk a **Start Without Debugging** almenüpontot. Az alábbi konzol ablak fog megjelenni:

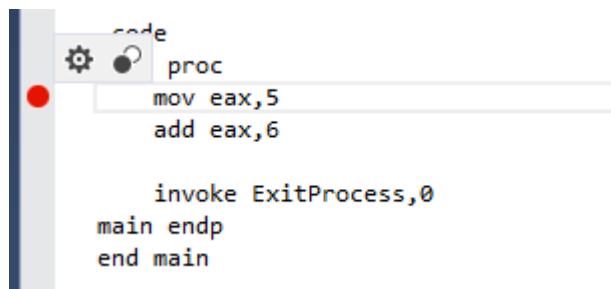


### 4. ábra A program futása közben megjelenő konzol ablak

A "Press any key to continue" a Visual Studio által automatikusan generált üzenet. Gratulálunk, sikeresen lefuttattad az első Assembly programodat. Nyomd meg bármely billentyűt az ablak becsukásához.

### Program futtatása Debug módban (nyomkövetés):

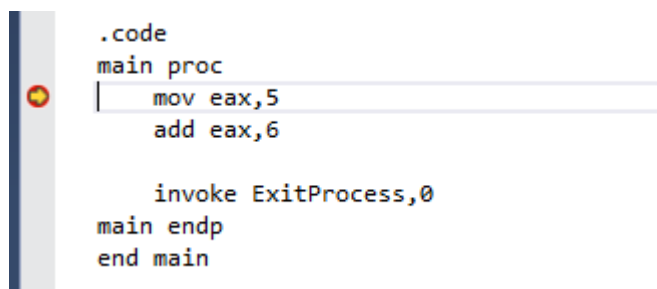
Ha megbizonyosodtunk róla, hogy van `.asm` forráskód megnyitva a szerkesztőben (editor), akkor első lépésben egy **breakpoint**-ot (töréspontot) kell elhelyezni a programon belül. Ezt úgy tehetjük meg, hogy az általunk kiválasztott sorban, a szerkesztő bal oldalán lévő keskeny függőleges sávra kattintunk bal egérgombbal. Pl.:



```
code
proc
  mov eax,5
  add eax,6

  invoke ExitProcess,0
main endp
end main
```

Majd a Visual Studio *debugger*-ét használjuk fel, hogy végig lépkedjünk a programon, egyszerre csak egy műveletet végrehajtva. Ahhoz, hogy elkezdjük a lépegetést Debug módban, nyomjuk le az F10-es billentyűt. Ekkor egy sárga nyíl fog megjelenni a kiválasztott sor mellett a piros pont helyén.



```
.code
main proc
  mov eax,5
  add eax,6

  invoke ExitProcess,0
main endp
end main
```

A nyíl jelzi, hogy melyik művelet fog következőnek végrehajtódni. Ismételt F10 gombnyomással hajthatjuk végre az adott műveletet (*Step Over*). Addig folytathatjuk az F10 billentyű nyomogatását, amíg el nem jutunk az **invoke** művelethez. Ekkor egy ismételt lenyomással a program kilép.

További billentyűk segíthetnek a debug folyamán. Töréspont elhelyezésére az F9 billentyű szolgál. Navigáljuk a kurzort abba a sorba ahol töréspontot szeretnénk elhelyezni, majd nyomjuk meg az F9 billentyűt, ekkor automatikusan elhelyezésre kerül egy töréspont az adott sorban. A program debug módban való elindítására az F5 billentyű használható (Ekkor az első töréspontnál vagy hibánál áll meg a program futása). A nyomkövetést a Shift+F5 billentyűkombináció lenyomásával állíthatjuk le. Ha nem debug módban szeretnénk elindítani a programot, akkor használhatjuk a Ctrl+F5 billentyűkombinációt.

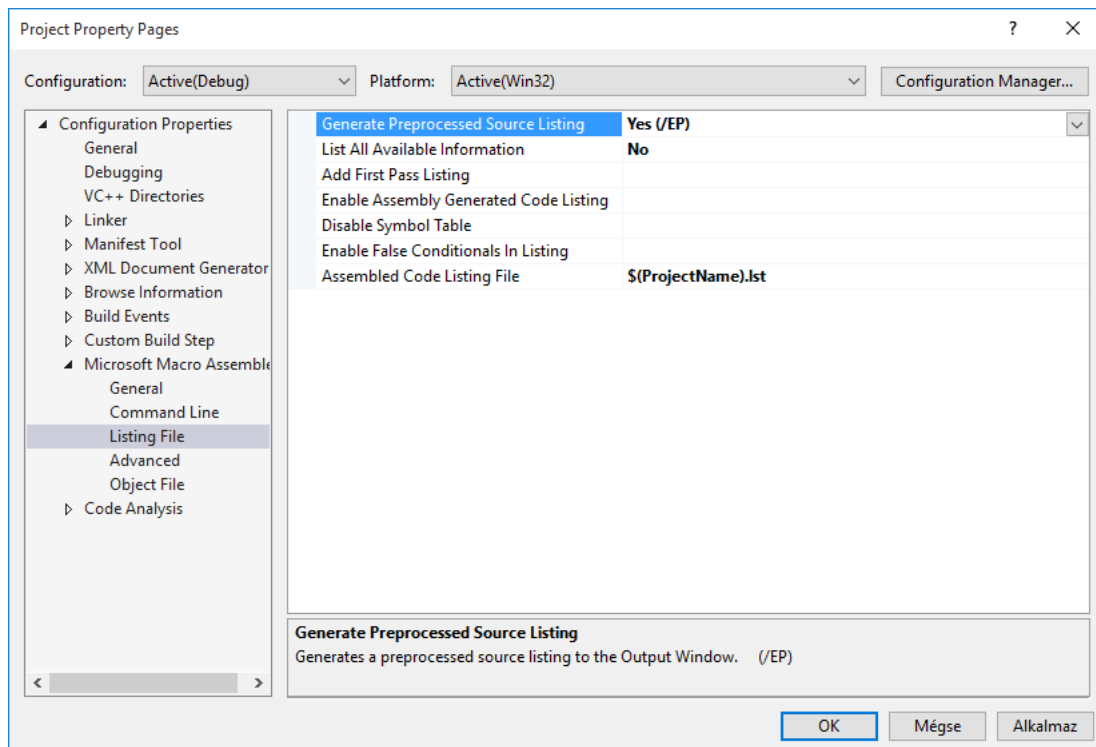
### **Regiszterek:**

Ha szeretnénk megjeleníteni a CPU regisztereket, akkor a programunk Debug módban való elindítása után válasszuk a **Debug** menüpontra belül a **Windows/Registers** almenüpontra. Ekkor meg is fog jelenni egy *Registers* nevű ablak. Az ablakon belül jobb egérgombot lenyomva kiválaszthatjuk a **Flags** elemet, így a CPU státusz *flag*-jei is megjelennek.

### **Fordítási lista készítése:**

Ha beállítjuk, a fordító készít egy lista fájlt, amiben elérhető pl. az összes definíció, makró, lefordított tárgykód, kigyűjtött címkék... Hasznos lehet továbbá hiba keresések.

Válasszuk a **Debug** menüpontra belül a **Project Properties** almenüpontra. Az 5. ábrán látható módon navigáljunk a **Listing File**-hoz, majd adjuk hozzá a képen megjelölt elemet.



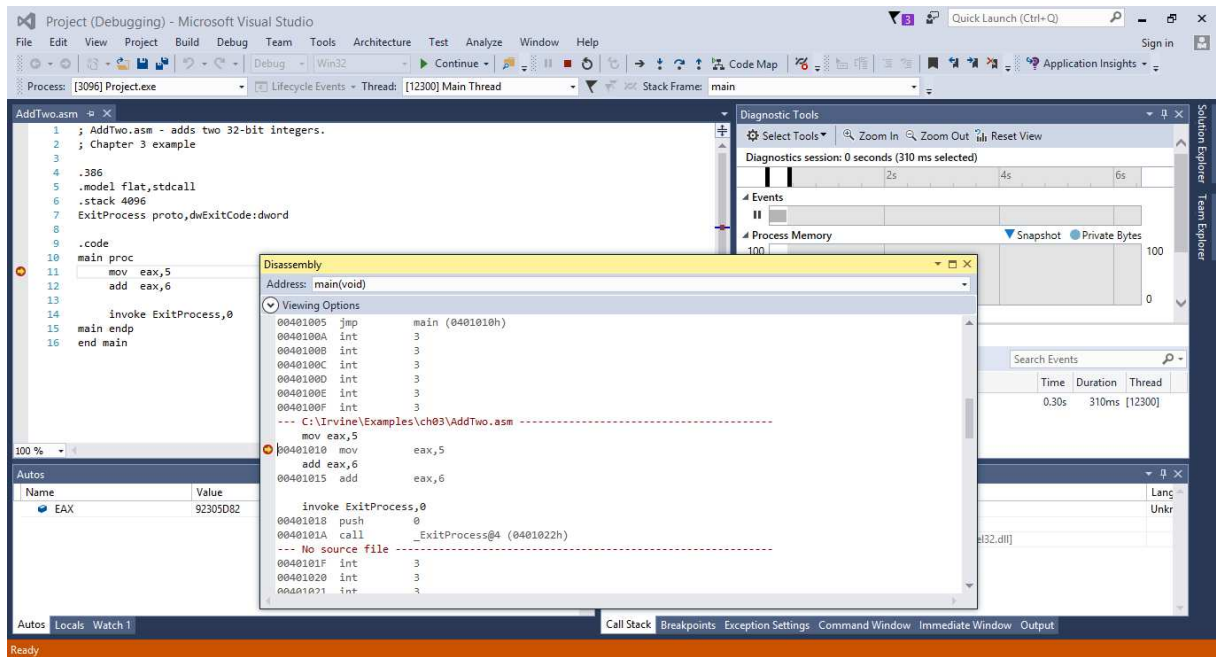
5. ábra Fordítási lista beállítása

### Disassembly listázás:

Ez a funkció csak nyomkövetés közben érhető el. Segítségével részletesebb információt kapunk a végrehajtódó parancsokról.

Tehát miután elindítottuk a program futását debug módban, válasszuk a **Debug** menüpontra belül a **Windows/Disassembly** almenüpontra. A kiválasztás után egy új ablak fog megnyílni **Disassembly** néven, melyen belül ugyanúgy egy sárga nyíl fogja mutatni, hogy mely művelet(ek) fog(nak) végrehajtódni, mint a szerkesztőben, ahogy ez a 6. ábrán is látható.





6. ábra Disassembly listázás debug módban