

Medii de programare pentru limbajul de asamblare al procesoarelor Intel

Scopul acestor laboratoare nu este de a învăța scrierea de cod exclusiv în limbaj de asamblare, ci de a scrie porțiuni de cod în limbaj de asamblare inserate în programe C/C++.

Există mai multe medii de programare care permit combinarea de cod scris în C/C++ și respectiv în limbaj de asamblare. Două dintre acestea sunt mai ușor de folosit:

- Microsoft Visual Studio
- Dev-C++

Nu este necesară instalarea ambelor programe. Varianta recomandată este Microsoft Visual Studio. Doar în cazul în care, din diverse motive, Microsoft Visual Studio nu poate fi folosit, se va apela la Dev-C++.

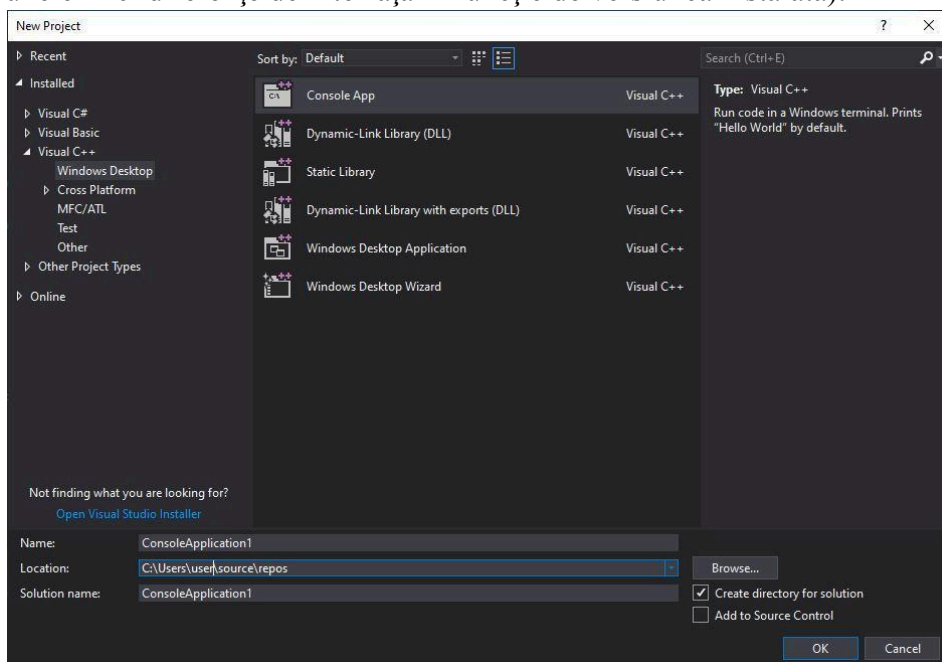
Visual Studio

Versiunea gratuită a acestui mediu de programare, numită Visual Studio Community, poate fi descărcată de la adresa <https://visualstudio.microsoft.com/vs/community/>.

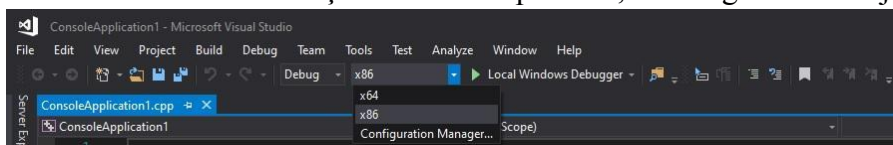
Observație: A nu se confunda mediul de programare Visual Studio cu editorul Visual Studio Code. Acesta din urmă este la rândul său un instrument util în multe situații, dar nu reprezintă un mediu integrat și ca atare nu este de interes la această disciplină.

Pentru instalare este necesară crearea unui cont gratuit pe platforma Microsoft. Instalarea poate dura un timp relativ lung, în funcție de versiunea utilizată și de configurația hardware a calculatorului pe care se face instalarea. La începutul instalării vi se vor solicita unele informații de configurare; acordați atenție acestei faze, întrucât o instalare nepotrivită va face programul inutil și va necesita reluarea întregului proces. În mod special, aveți grijă ca limbajul C++ să fie selectat pentru instalare (desigur, puteți selecta și alte limbaje, în funcție de ceea ce considerați că vă poate fi de ajutor în viitor).

După instalare, Visual Studio poate fi folosit pentru dezvoltarea de programe C++. Este de reținut că, pentru crearea unei aplicații noi, trebuie creat mai întâi un proiect (meniul `File/New/Project...`). La momentul creării trebuie selectat tipul proiectului. Pentru programele scrise la această disciplină, proiectul trebuie să fie de tipul `Visual C++/Windows Desktop/Console App`, așa cum este arătat în figura de mai jos (pot exista unele mici diferențe de interfață în funcție de versiunea instalată):



Odată creat proiectul, trebuie stabilită platforma pentru care se dorește compilarea programului. Din bara de unelte se accesează lista derulantă care conține variantele posibile, ca în figura de mai jos:



În mod implicit proiectele Visual C++ au setată varianta `x64` (compilare de programe pe 64 biți), care trebuie înlocuită, deoarece nu permite inserarea de cod în limbaj de asamblare. Trebuie selectată varianta `x86`, care corespunde compilării de programe pe 32 biți.

În mediul de programare Visual Studio, pentru a insera cod scris în limbaj de asamblare într-un program C/C++ se folosește cuvântul cheie `_asm`. Cu ajutorul acestuia pot fi scrise instrucțiuni individuale sau blocuri de instrucțiuni, delimitate în același mod ca orice bloc de cod C/C++. Instrucțiunile în limbaj de asamblare pot fi separate prin ";", ca în limbajul C, sau prin trecerea la linie nouă (sau ambele). Pentru claritate, se recomandă scrierea câte unei singure instrucțiuni pe un rând.

De exemplu, dacă se dorește scrierea unei singure instrucțiuni în limbaj de asamblare, se procedează ca mai jos:

```
_asm mov eax, 5;
```

Pentru scrierea mai multor instrucțiuni în limbaj de asamblare, exemplul următor ilustrează diversele moduri de scriere acceptate sintactic în ceea ce privește delimitarea instrucțiunilor:

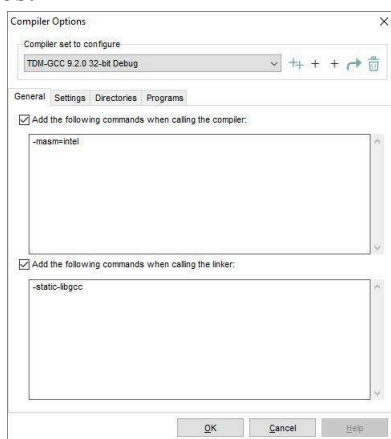
```
_asm {  
mov eax, 5  
add eax, ebx; sub eax, 3  
mov ebx, 4  
}
```

Dev-C++

Mediul Dev-C++ poate fi descărcat și instalat gratuit. De-a lungul existenței sale a fost dezvoltat și întreținut de mai multe companii, astfel încât sunt disponibile mai multe versiuni. Deși practic toate sunt utilizabile la această disciplină, se recomandă a instala versiunea cea mai recentă, care poate fi găsită la adresa <https://github.com/Embarcadero/Dev-Cpp/releases>.

În Dev-C++ nu este obligatorie crearea unui proiect, fiind posibilă compilarea unui fișier sursă individual. Acest mod de lucru este mai simplu în cazul scrierii programelor de complexitate redusă, deci se recomandă a fi folosit.

Pentru a permite inserarea de instrucțiuni în limbaj de asamblare, trebuie modificate setările generale ale compilatorului. În meniul **Tools** se alege opțiunea **Compiler Options...**, ceea ce are ca efect deschiderea fereastră de dialog de mai jos:



În tabul **General**, care este selectat implicit, se bifează caseta *Add the following commands when calling the compiler*, iar în câmpul de editare corespunzător se introduce textul `"-masm=intel"`, ca mai sus, după care fereastra este închisă prin apăsarea butonului **OK**. Atenție: nici una dintre celelalte setări nu trebuie modificată.

În cazul compilatorului GCC, utilizat de mediul Dev-C++, codul scris în limbaj de asamblare este inserat cu ajutorul cuvântului-cheie `asm`. Spre deosebire de corespondentul său din Visual C/C++, acesta nu definește un bloc de instrucțiuni, în schimb primește un șir de caractere care conține una sau mai multe instrucțiuni în limbaj de asamblare. Următoarele exemple reprezintă forme valide de a scrie cod în limbaj de asamblare:

```
asm("mov eax, 10;");  
asm("mov ebx, 7; add eax, ebx");
```

La fel ca în Visual Studio, caracterul ";" poate fi scris la finalul fiecărei instrucțiuni în limbaj de asamblare, însă nu este obligatoriu decât pentru a delimita instrucțiuni scrise în același șir de caractere (deci într-o singură instrucțiune `asm`).

În cazul secvențelor mai lungi de cod, ambele variante de mai sus (utilizarea `asm` pentru fiecare instrucțiune în limbaj de asamblare, respectiv scrierea mai multor asemenea instrucțiuni pe o singură linie) sunt incomode. Din fericire, este posibilă scrierea unei întregi secvențe într-o singură instrucțiune `asm`, dar pe linii diferite, ca în exemplul de mai jos:

```
asm("mov eax,25;"  
"mov edx,0;"  
"mov esi,10;"  
"div esi;");
```

În acest caz este obligatorie utilizarea caracterului ";" în interiorul șirurilor de caractere, pentru a delimita instrucțiunile în limbaj de asamblare. În schimb, între șirurile de caractere care descriu instrucțiunile nu trebuie scris nimic (doar finalul de linie), altfel s-ar obține o eroare la compilare.