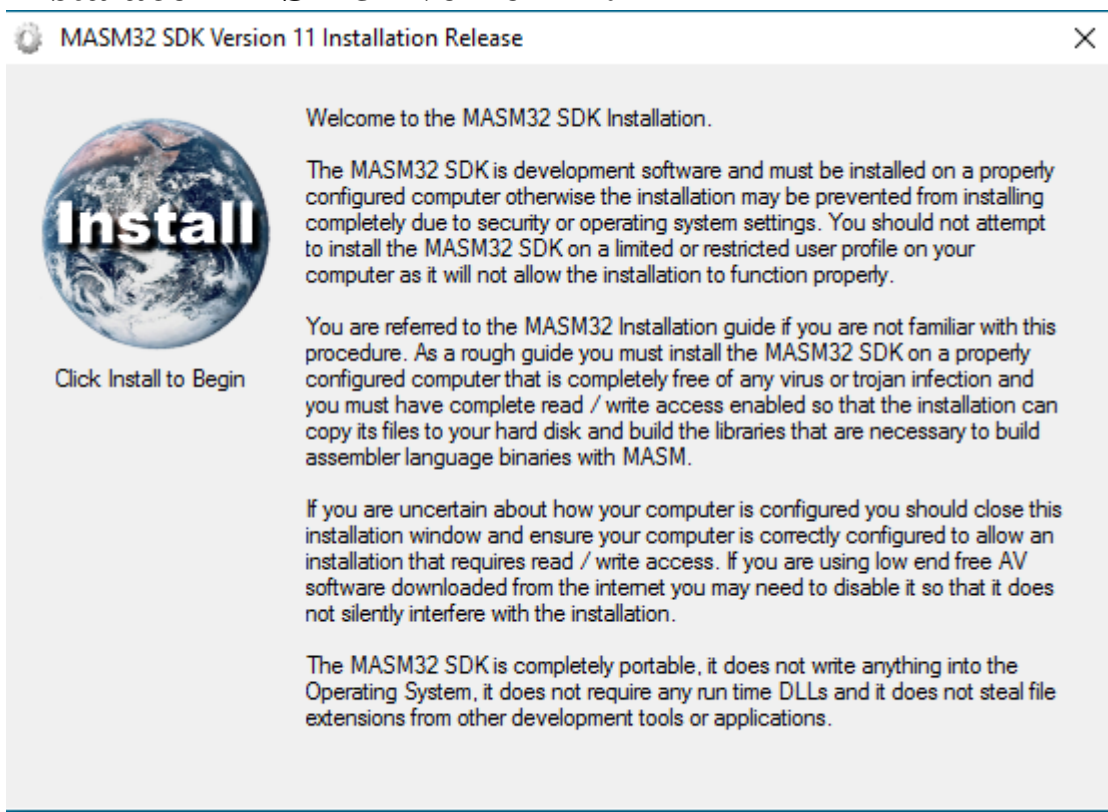


## Úvod

**C**o je to assembler, je zbytečné psát, protože na toto téma se dá najít na internetu hodně informací. V tomto seriálu si ukážeme, jak využít MASM32 na programování Windows aplikací. Jistě jste slyšeli o tom, že assembler je nepřehledný a podobné věci, ale myslím, že po pár dílech uvidíte, že to není až taková pravda. V tomto seriálu budu popisovat pouze podstatné věci, nikoliv alternativy. Ty podstatné věci budou stačit pro víc jak 90% běžného programování v MASM32. Abych nějak zakončil úvod, tak nastíním, co jsem pro vás připravil a na co se můžete do budoucna těšit.

- instalace MASM32, IDE, nastavení,
- základní práce v IDE WinAsm Studio,
- naučíme se základní direktivy MASM32,
- ukážeme si jak vytvořit aplikace typu console, dialog a windows,
- naučíme se využívat msvcrtdll pro ještě lepší přehlednost programu,
- řekneme si něco o makrech,
- .... různé, tipy a triky,

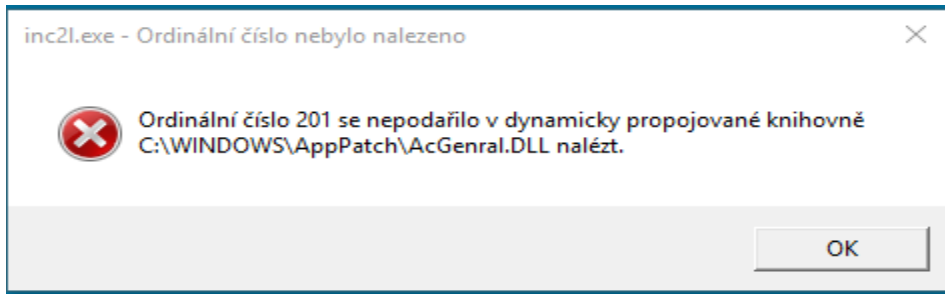
## Instalace MASM32 verze 11r.



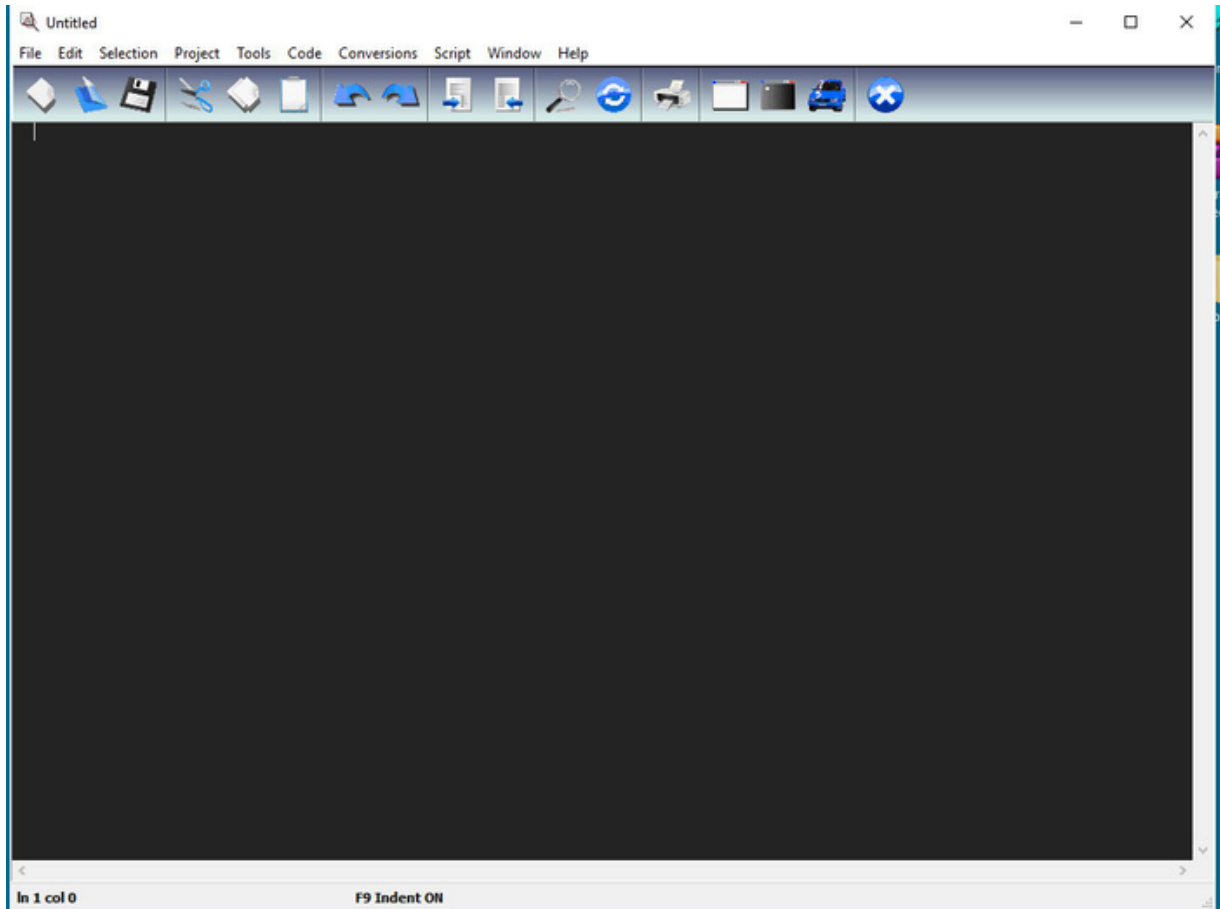
MASM32 pochází z dílny Microsoft a není již součástí vývojového nástroje Visual C++ dnešního Visual C++ 2015 .NET. V těchto vývojových nástrojích je jako samostatný kompilátor. Kompilaci vkládaného assembleru zajišťuje samotný překladač C++. Překladače assembleru se chytila partička lidí a pod licenci GPL vytvořily samostatný balík pro

programování v assembleru, který obsahuje potřebné knihovny a nástroje. Instalaci MASM32 můžete stáhnout na domovské stránce [hutch's](#). Opravenou instalaci můžete stáhnout na mé webové stránce [zde](#). Po stažení instalačního souboru (zhruba 5Mb) spusťte instalaci. Instalace je zcela automatická a požaduje po vás pouze zvolit disk, na který se má MASM32 instalovat. Instalovat budeme do kořenového adresáře zvoleného disku.

Patrně od windows 7 začínají ke konci instalace vyskakovat chybové hlášky, jejich typ závisí na použité verzi windows.



Násilně ukončíme instalaci, pokud spustíme MASM32 zdá se být vše OK, pak bychom zjistili, že neumí vše přeložit. Oprava je jednoduchá spustíme soubor „makelibs.bat“, instalace se ukončí korektně. Pokud spustíme Qeditor pak uvidíme.



Pokud jste instalaci zdárně zvládli, tak je ještě zapotřebí udělat jeden krok, a to vytvoření složky ve složce MASM32 nebo jinde s názvem **Programy**. Tuto složku může jmenovat libovolně a bude sloužit jako adresář vašich programů.

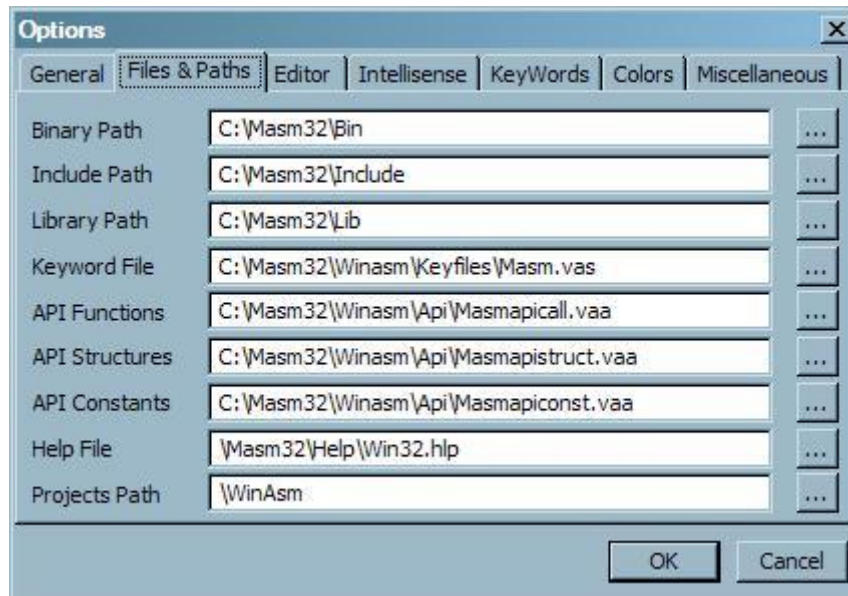
Tady taková jedna perlička. Instalace spočívá pouze ve vytvoření adresáře MASM32 kam se pak uloží potřebné soubory. Vytvoří se knihovny .LIB. Nic se nezapisuje do registru a ani nikam jinam. Takže pokud chcete MASM32 odinstalovat (ale o tom pochybuji), tak stačí smazat adresář MASM32.

## Doporučené IDE a konfigurace

Za tu dobu co jsem pracoval v MASM32 jsem vyzkoušel mnoho editoru a IDE pro psaní kódu. Ze začátku jsem používal QEditor, který je součástí instalace. Ale postupem času se ukázalo, že pro větší projekty je tento editor nevhodný. Proto jsem se začal poohlížet po něčem jiném, až jsem skončil u IDE WinAsm Studio. O tomto IDE mohu říct jen to, že to je absolutní špička v IDE pro assembler a hlavně je free. Zajímavé je na tomto IDE i to, že je celé naprogramované v MASM32. Což je důkaz, že assembler si zaslouží své čestné místo mezi programovacími jazyky. Stránky tohoto projektu najdete na <http://www.winasm.net> kde po registraci, která je zdarma, máte možnost stažení instalace. Registrace vám přinese i tu výhodu zapojit se do diskuzního fóra na stránkách nebo možnost stahovat různé doplňky k tomuto IDE.

Bohužel v dnešní době se sice můžete zaregistrovat, ale žádnou potvrzující odpověď nedostanete, a nepodaří se Vám z tohoto webu nic stáhnout. Stáhněte si soubor WinAsm511.zip [zde](#) a masm32v11r.zip [zde](#). Rozbalte tento zip soubor do adresáře MASM32 tak, aby jste měli strukturu tohoto adresáře jako je na níže uvedeném obrázku.

Pokud máte podobnou strukturu adresáře instalace MASM32, jako je na výše uvedeném obrázku, tak se můžeme pustit do konfigurace. Nejprve vytvoříme zástupce pro **Winasm.exe** a doufám, že důvod nemusím vysvětlovat. Potom spustíme **WinAsm Studio**, kde se před vámi objeví dialog **Recent Projects**. Tento dialog přeskočte kliknutím na **Exit**. V menu pak zvolte **Tools Options ...**. V okně **Options** vyberte záložku **Files & Paths**, kde nastavíte adresáře a potřebné soubory.



*(Pokud máte MASM32 instalovaný na disku C:, tak nastavení by mělo vypadat takto)*

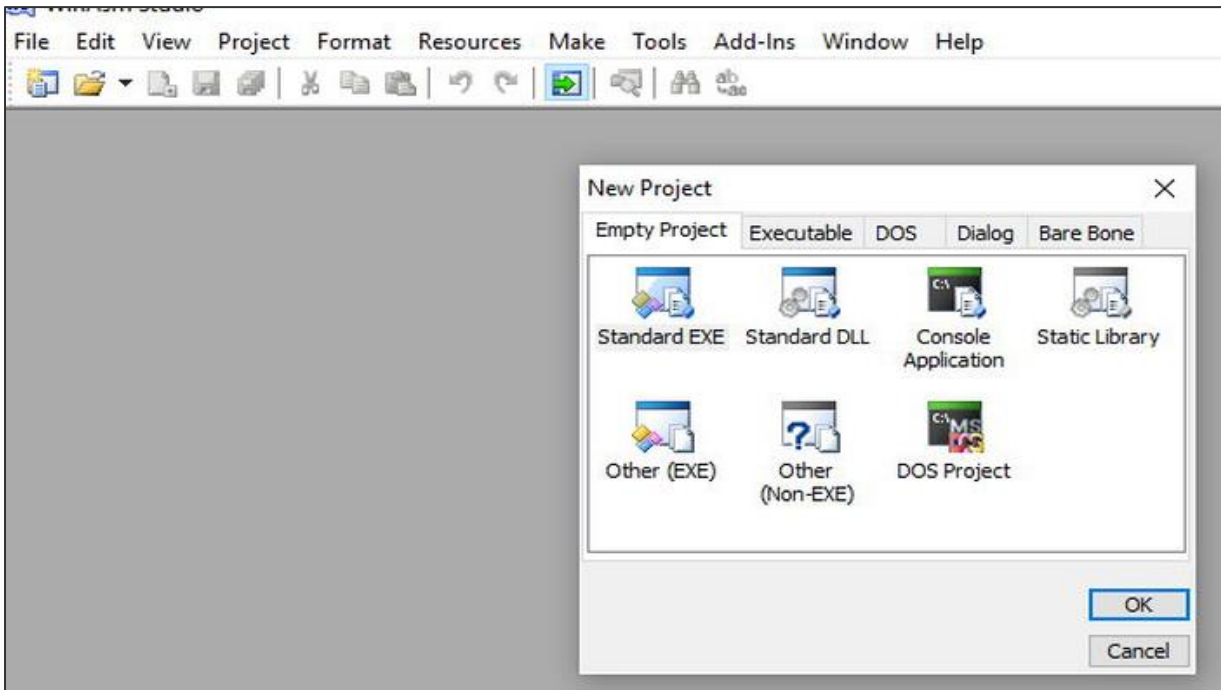
## Něco pro nedočkavé

Určitě se vás pár najde, kteří začnou "laborovat" a právě pro ně tu mám nějaké tipy. Podívejte se do následujících adresářů:

- **Example** - ukázkové programy,
- **Help** - soubory nápovědy, kde naleznete užitečné informace. Bohužel v angličtině.
- **Html** - něco málo o projektu MASM32,
- **Icztutes** - pár tutoriálů,

Důležité je znát alespoň částečně angličtinu, protože většina dokumentací je právě v tomto jazyku. Trošku předběhnu, když řeknu, že v budoucnu budeme používat soubor nápovědy kde jsou popsány API Windows. Tento soubor má zhruba 20Mb a popisuje několik set API funkcí a přeložit to vše do češtiny a udržovat to aktuální je nad lidské síly. Možné to je ale, člověk by pak nic jiného nedělal. Nehledě na to, že tento soubor je pouze mizivá část z dostupné dokumentace.

**Ukázka Winasm.**



Jako poslední výborný program je „[byme\\_win32dasm893cz](#)“, který je též zdarma a můžete si ho stáhnout zde. Po rozbalení ho nakopírujte do složky MASM32.

Jako nevýhoda je čeština, je tvořena ve starých programech. Nezkoušel jsem převody, někdo si může s tímto pohrát.

Ve své staré dokumentaci jsem našel 3 podobné manuály, bohužel odkazy na weby již nejsou funkční. Všechny manuály jsou psané v česky.

- První manuál je od pana Lukáše Valenty „[UCLVRTF.zip](#)“
- Druhý manuál neznámý autor „[Makroassembler](#)“
- Třetí manuál od Jaromír Veber „[Microsoft Assembler](#)“

```

Disassembler Edit Project Debug Search Goto Execute Text Functions HexData Refs Help
[Icons]
Disassembly of File: C:\masm32\WinAsm\Samples\DLL\2\Skeleton.dll
Code Offset = 00000400, Code Size = 00000200
Data Offset = 00000800, Data Size = 00000200

Number of Objects = 0004 (dec), Imagebase = 10000000h

Object01: .text      RVA: 00001000 Offset: 00000400 Size: 00000200 Flags: 60000020
Object02: .rdata     RVA: 00002000 Offset: 00000600 Size: 00000200 Flags: 40000040
Object03: .data      RVA: 00003000 Offset: 00000800 Size: 00000200 Flags: C0000040
Object04: .reloc     RVA: 00004000 Offset: 00000A00 Size: 00000200 Flags: 42000040

+++++ MENU INFORMATION +++++
      There Are No Menu Resources in This Application
+++++ DIALOG INFORMATION +++++
      There Are No Dialog Resources in This Application
+++++ IMPORTED FUNCTIONS +++++
Number of Imported Modules = 1 (decimal)

      Import Module 001: user32.dll
+++++ IMPORT MODULE DETAILS +++++

      Import Module 001: user32.dll

      Addr:00002038 hint(019D) Name: MessageBoxA
+++++ EXPORTED FUNCTIONS +++++
Number of Exported Functions = 0001 (decimal)

      Addr:10001070 Ord: 1 (0001h) Name: TestHello

+++++ ASSEMBLY CODE LISTING +++++
//***** Start of Code in Object .text *****
Program Entry Point = 10001000 (C:\masm32\WinAsm\Samples\DLL\2\Skeleton.dll File Offset:00001600)

//***** Program Entry Point *****
:10001000 55          push ebp
:10001001 8BEC        mov ebp,, esp
:10001003 837D0C01   cmp dword ptr [ebp+0C], 00000001
:10001007 7515       jne 1000101E
:10001009 6A00       push 00000000

* Possible StringData Ref from Data Obj ->"DLL Skeleton"
:1000100B 6800300010 push 10003000

* Possible StringData Ref from Data Obj ->"The DLL is loaded"
:10001010 683A300010 push 1000303A
:10001015 6A00       push 00000000

* Reference To: user32.MessageBoxA, Ord:019Dh
:10001017 E869000000 call 10001084
:1000101C EB49       jmp 10001067

* Referenced by a (U)nconditional or (C)onditional Jump at Address:

```