

2 Manuali di Giobe2000

# TUTORIAL ASSEMBLER

## Nuovo Ambiente Assembler

IDE **Winasm Studio** di **Antonis Kyprianou**,  
**Winasm.net Team**

**Come si installa e configura**

Copyright © luglio 2009

Studio Tecnico ing. **Giorgio Ober** [contatto@giobe2000.it](mailto:contatto@giobe2000.it)

Questa **Monografia** può differire in parte dalla versione *on-line*  
soggetta a probabili aggiornamenti e integrazioni.

Verifica sempre le eventuali novità direttamente sul Sito

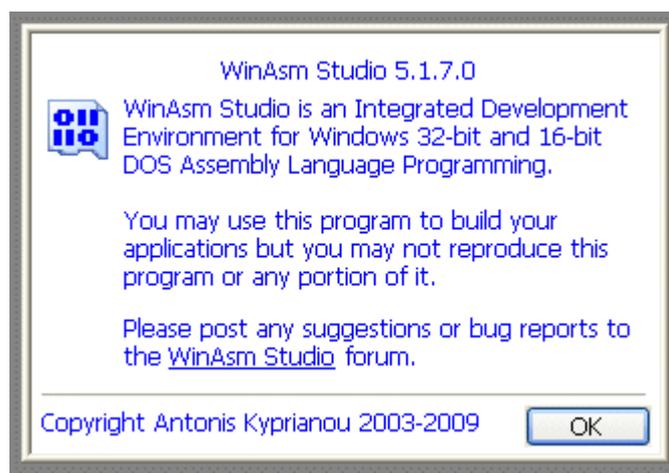
Copyright [www.Giobe2000.it](http://www.Giobe2000.it) ®

IDE **Winasm Studio** di Antonis Kyprianou,  
Winasm.net Team

## Come si installa e configura

Il **Nuovo Ambiente Assembler** è particolarmente interessante anche per il fatto di avvalersi di potenti e versatili *editor di testo* con i quali sviluppare e condurre ogni fase del processo di creazione di un *progetto ASM*, appositamente cercati su internet e scelti tra numerose alternative, dopo accanito e puntiglioso collaudo.

Queste pagine forniscono ogni dettaglio sulla *natura* e sulla *messa a punto* dell'*editor* di **WinAsm Studio 5.1.70 Full Package**, creato e offerto **freeware** da **Antonis Kyprianou, Winasm.net Team**; la sua intrinseca predisposizione per la *programmazione Assembly*, sia **DOS a 16 bit** (da noi praticata in questo sito) che **Windows a 32 bit**, pone in particolare luce l'utilizzo di questo strumento, anche per la sua *struttura* pensata per organizzare un **IDE** (*Integrated Development Environment*), sebbene *anche gli altri* editor si prestano benissimo per creare un *ambiente integrato* per lo *sviluppo* di programmi.



Di certo il contesto **WinAsm Studio** è decisamente ricco, pensato e ottimizzato (sia per dimensione e che per velocità) specificatamente per l'*Assembly*, con molte caratteristiche utili anche per la creazione di programmi basati sulle *API Win32*; supporta *intrinsecamente* il compilatore **Microsoft Macro Assembler (MASM)**, pur consentendo l'utilizzo anche di altri assembleri (come FASM) con l'aggiunta di comodi e pratici *Add-In*, spesso creati dalla stessa comunità di *sviluppatori* di programmi ASM che hanno contribuito a rendere disponibile una dotazione sempre più grande di strumenti.

**WinAsm Studio** è scaricabile gratuitamente dal sito dell'autore, <http://www.winasm.net/>, che mette a disposizione, oltre alle dovute informazioni, molte altre pagine utili (selezionabili da un suo menu interno) e dispone di un **Gruppo di discussione** i cui membri sono sempre pronti a

sostenere i *nuovi arrivati* e *dare risposta* ai possibili quesiti in materia di IDE, *fornire* consigli e trucchi di programmazione, *offrire* progetti di ogni tipo completi di codice sorgente, oppure moltissimi *Add-Ins* anche in merito alla *API di Windows* (per altro non coinvolte nell'ambito di questo sito).



**winasm.net**  
X86 ASSEMBLY COMMUNITY

Official Home of...  
**WinAsm Studio & HiEditor**

Home Forum Screenshots Downloads Link To Us Programming Tools

Welcome to winasm.net

**The x86 Assembly community and official home of WinAsm Studio and HiEditor**

winasm.net was originally founded as the home of the WinAsm Studio IDE, evolved as a site for the Assembler programmers and the home for HiEditor, the Large File Editor. Currently, **software developers** of almost any **programming language** such as C, C++, all flavors of Basic, Delphi, .NET etc can find a lot of answers for their programming needs.

WinAsm Studio is a free Integrated Development Environment **IDE** for developing **32-bit Windows** and 16-bit **DOS** programs using the **Assembler**. The Microsoft Macro Assembler (**MASM**) is supported inherently, while the FASM Add-In adds support for FASM and other assemblers.

WinAsm Studio is optimized for size and speed, yet contains many useful features such as **intellisense** which will assist you in creating Windows **API** based programs. Highly extensible via user-created add-ins. Includes a powerful **Visual Resource Editor**.

WinAsm Studio is a Multilingual User Interface (MUI) application meaning that you can set everything of its GUI to be displayed in the language of your choice. It has already been translated to many languages; You are more than welcome to make or help with the translation to your native one.

You are more than welcome to participate in our forums and ask for any help you might need. Our members are always ready to support the newcomers to start programming. The board is divided in 5 categories which are composed of several forums in which you can find answers to your questions regarding the IDE, programming **tips and tricks**, full **source code** projects, Add-Ins, discussions about the Windows API, **custom controls** and a lot more.

Enjoy,  
The winasm.net team

Si tratta di **molto di più** di una semplice alternativa al *Blocco Note* di Windows; ho raccolto le mie impressioni nei seguenti punti:

- la sua caratteristica **MUI (Multilingual User Interface)** consente l'installazione di un **menu in italiano**, cosa sempre gradita e, per taluni miei amati visitatori, addirittura indispensabile; l'interfaccia grafica è decisamente semplice e di immediata usabilità.
- non è particolarmente dotato di **comandi tipici** di un editor, ma è risaputo che **non è questa la sua forza**; dispone della possibilità di *indentare* e *togliere l'indentazione*; consente la *rimozione* e/o la *presenza dei commenti* (toglie o rimette i testi a destra del ";"); gestisce la *conversione da maiuscolo a minuscolo* e viceversa; *scambia* le lettere *maiuscole* con quelle *minuscole*; permette la *presenza dei numeri di linea*; **ma non** dispone della *selezione del testo* in *modo colonna*, spesso molto utile, e **non** dispone del *registratore di macro*
- non dispone intrinsecamente di **programmi di utilità** ma offre la possibilità ad essere integrato con l'aiuto di una infinita dotazione di **Add-In** (scaricabili ed installabili a parte) che rende questo ambiente molto versatile ed espandibile
- buona ed efficiente gestione della *catena compilazione*, attivabile direttamente con **Shift**

+ F8, oppure cliccando su un'iconcina a ciò dedicata, oppure a partire dal menu in alto, via **Compilazione** > **Compila tutto**; data la sua alta specificità **consente anche** la gestione separata delle 2 fasi di **compilazione** (attivabile direttamente con Shift + F6, oppure con un'iconcina a ciò dedicata, oppure via **Compilazione** > **Assembla**) e di **collegamento** (direttamente con Shift + F7 oppure con un'iconcina a ciò dedicata, oppure via **Compilazione** > **Link**); permette **infine** di vedere l'effetto dell'eseguibile COM (o EXE se configurato diversamente) appena compilato e linkato, direttamente con Ctrl + F8 oppure con un'iconcina a ciò dedicata, oppure via **Compilazione** > **Esegui**

consente una **eccellente configurazione per l'Assembly**, assicurata da un file di associazione, "**MASM.vas**", già ricco in origine e comunque da me **riadattato** e **integrato** per definire ben 14 **categorie da colorare**: 3 dedicate alle **istruzioni** (*Cpu* , *System*. *Fpu*, *Multimedia*), una ai **registri** (*CPU*, *CONTROL*, *DEBUG*, *TEST*, *MMX*, *COPROCESSORE*, *XMM*), 3 dedicate alle **istruzioni** (*Cpu* , *Fpu*, *Multimedia*), 2 alle **direttive MASM** (*KeyWords*, *PAROLE RISERVATE*, *PSEUDOISTRUZIONI*, *SEPARATORI*), una a **Giobe2000** (*PROCEDURE*, *MACRO*), 3 alle **Api win32** (*Functions*, *Structures*, *Constants*) e altre più specifiche e meno importanti

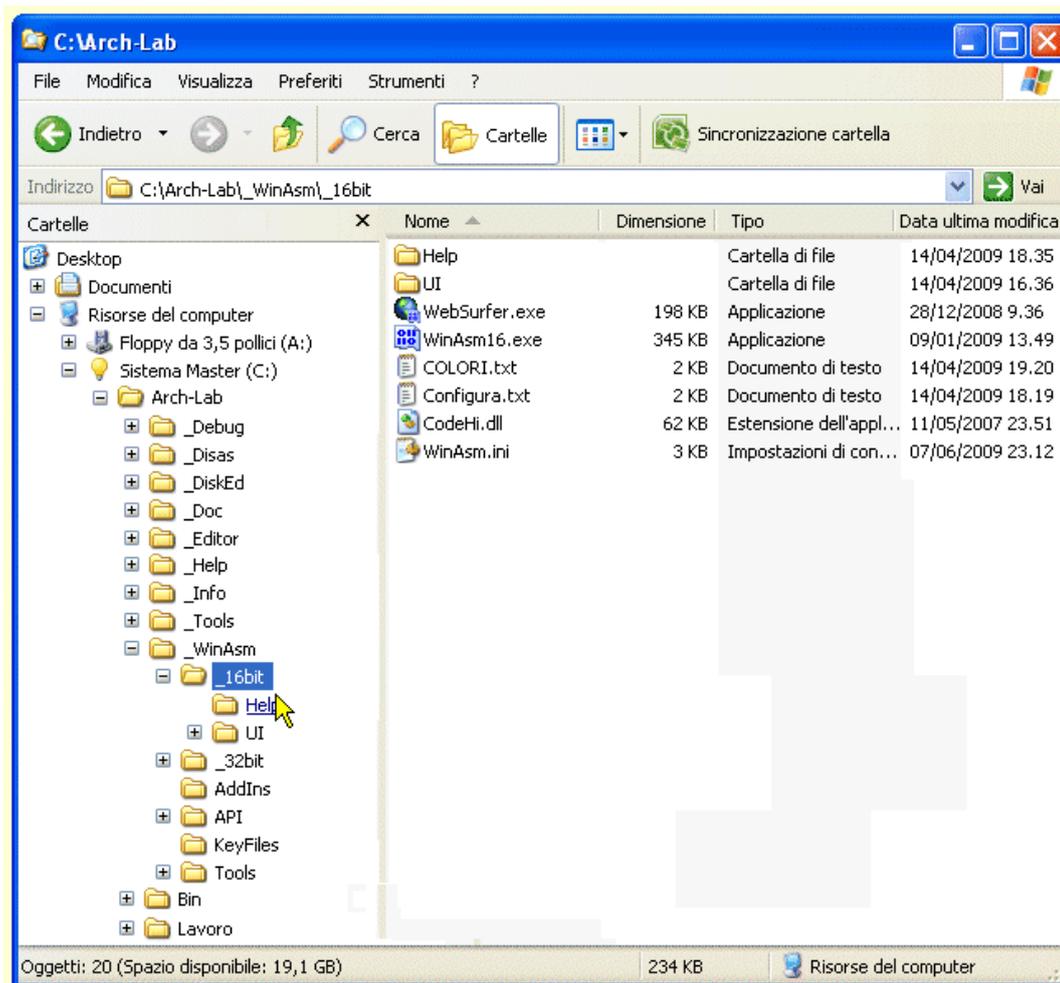
è possibile affidare il compito di **debug** a qualunque debugger, come *WinDbg* o *OllyDbg*; in quest'ambito ho però preferito attivare il classico *Debug* DOS, la cui apertura sul file eseguibile corrente non è per altro automatica

non è consentita la gestione di comandi esterni, per altro non particolarmente sentita dato che **WinAsm Studio** prevede intrinsecamente la messa a punto delle principali azioni funzionali all'IDE che rappresenta; per questo non esistono icone interne per la gestione di questo tipo e quindi non è possibile, per esempio, affidare ad una icona interna la gestione del prezioso programma *Weg\_2.15*, straordinario gestore di archivi di dati da altri editor attivato dal loro interno

mostra, nella parte in basso dell'editor, un'area dedicata alla proposta degli eventuali **messaggi d'errore** generati dalla **catena di compilazione** oppure per segnalare **eventi positivi** e **suggerimenti** sul da farsi: grande cosa!!

Le seguenti considerazioni riguardano l'**installazione** e la **messa a punto** di **WinAsm Studio**: sono **argomenti di servizio**, probabilmente non necessari per l'utilizzo appropriato di questo strumento, aggiunti solo per soddisfare qualche eventuale curiosità e per mantenere memoria delle operazioni condotte nella creazione **Assembler**.

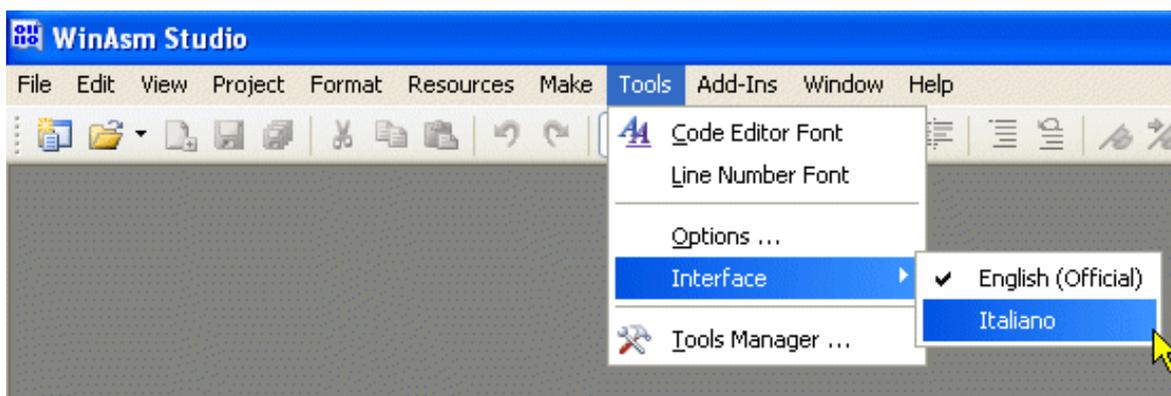
**1.** Il programma è stato **installato** nella cartella **C:\Arch-Lab\\_WinAsm\**=%SystemDrive%\Arch-Lab\\_WinAsm: ad essa faranno riferimento tutti gli automatismi dell'**Ambiente** per cui non è opportuno rinominarla o spostarla; tra l'altro ho pensato di predisporre due diversi ambienti dedicati separatamente alla **programmazione Assembly DOS a 16 bit** (da noi praticata in questo sito) e **Windows a 32 bit**, pensando al futuro...



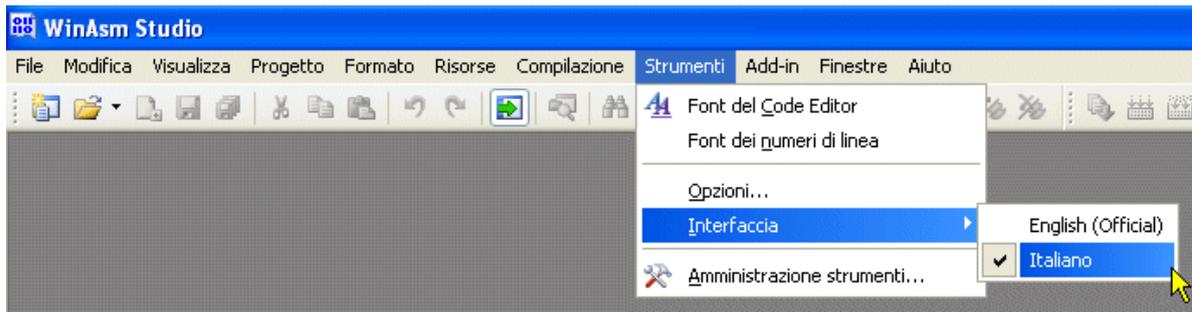
La differenza sostanziale sta nella fase di collegamento, dato che per i programmi DOS è richiesto un **linker Windows compatibile a 16-bit**, dato che quello di ultima generazione (**Link 32**) non può essere utilizzato.

Da notare la presenza di una cartella con la **raccolta di numerose ADD-Ins**, irrilevanti nella programmazione a 16 bit ma utili e già disponibili per l'installazione nel caso di utilizzo Win32.

**2.** Al primo avvio dell'editor ho provveduto a configurarne i menu per la **lingua italiana**: dal menu in alto selezionare **Tools > Interface > Italiano** e cliccare su di essa:

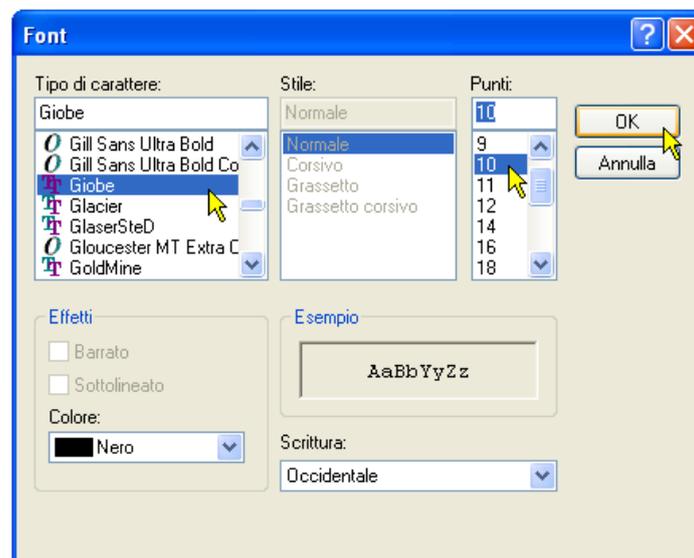
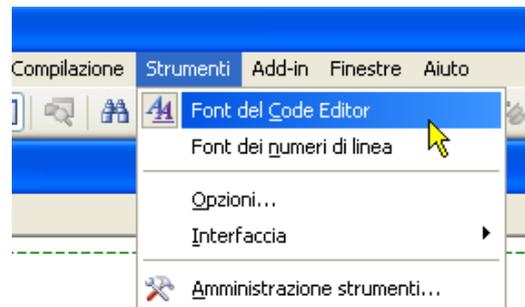


Chiudere e rientrare; la stessa sequenza di comandi mostra ora tutti i menu in **Italiano**:

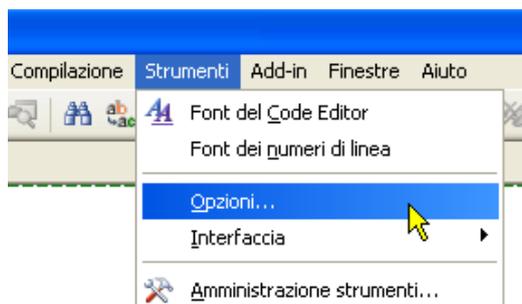


La disponibilità di poter disporre della traduzione è dovuta al contenuto dei files **Italiano.vas** e **Italiano.rc** conservati nella sottocartella `\_WinAsm\_16bit\UI\Italian\` e al contenuto della libreria **Italiano.DLL**.

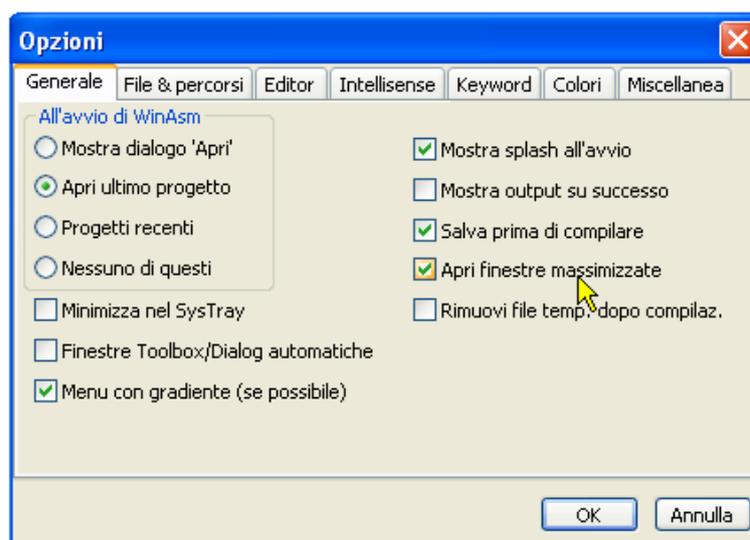
**3.** Passiamo ora a configurare il **tipo di carattere** da utilizzare **per mostrare i sorgenti ASM**; da molto tempo ho creato un font adatto a mostrare tutti i **caratteri Ascii** (in particolare quelli **estesi**, non visualizzabili con altri font *monospace* come *Courier*). Si tratta del Font **Giobe.TTF**, fornito in dotazione nella cartella `C:\Arch-Lab` e da trasferire nella cartella `C:\Windows\fonts=%SystemRoot%\fonts`; la predisposizione si esegue dal menu in alto via **Strumenti > Font del code editor**, selezionando **Giobe** tra i font offerti dall'opzione **Tipo di carattere** e fissandone a **10 i Punti** e a **normale** lo **Stile**:



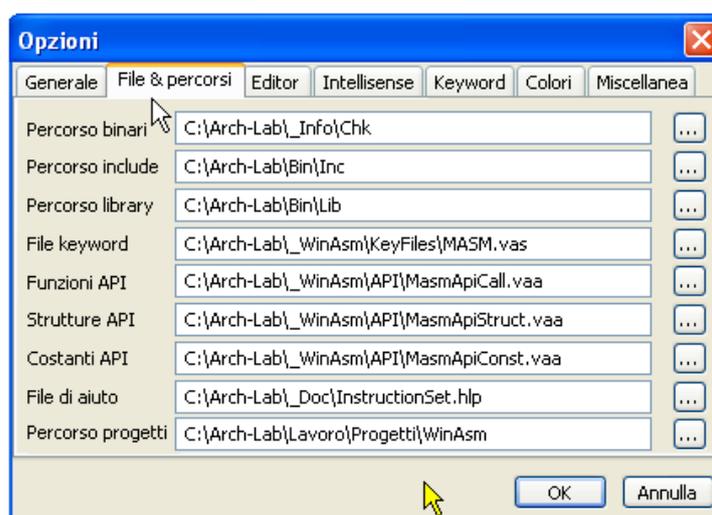
4. Passiamo ora a configurare le **opzioni principali** da imporre all'ambiente integrato; la predisposizione si esegue dal menu in alto via **Strumenti > Opzioni**:



Tra quelle della cartella **Generale** confermiamo le scelte di figura:



Più importanti quelle della cartella **File & percorsi**, fondamentali per un corretto funzionamento dell'IDE; le impostiamo così:



La **funzione più importante** da configurare è quella che deve assicurare la gestione completa della *catena compilazione*, cioè le predisposizioni da fare per garantire l'organizzazione e il coordinamento **automatico** di ogni operazione necessaria alla realizzazione di un *eseguibile* a partire dal sorgente ASM ospitato da **WinAsm Studio**; questo servizio è assicurato dalla *prima voce* dell'elenco, *Percorso binari*, che deve puntare la cartella contenente l'**assembler ML.EXE (MASM.EXE)** e il **linker LINK16.EXE (o LINK.EXE)**; se per qualche ragione il *percorso* scritto nella prima casella non è corretto ciascuna delle 2 fasi produrrà (rispettivamente) questi messaggi d'errore:

```
Output
\ML /c /I"C:\Arch-Lab\Bin\Inc" "C:\Arch-Lab\Lavoro\PROVA.ASM"
Errore nella creazione del processo.
Compilazione terminata. 1 errori occorsi.
```

```
Output
\Link16 @"C:\Arch-Lab\Lavoro\link.war"
Errore nella creazione del processo.
Compilazione terminata. 1 errori occorsi.
```

Gli strumenti **ML.EXE (assembler)** e **LINK16.EXE (linker)** sono ritenuti *intrinsecamente indispensabili* da **WinAsm Studio** e *dati per scontati* per il suo corretto funzionamento (anche se nessuno dei 2 è incluso nel suo pacchetto di distribuzione; entrambi sono indispensabili per il funzionamento del **Nuovo Ambiente Assembler**, una struttura **assolutamente senza scopo di lucro** e con **esclusive finalità didattiche**, creata per lo **studio** e la **pratica** dell'**assembly**, integrata con le lezioni del mio **Tutorial**, offerto **gratuitamente** a tutti insieme al certoso lavoro necessario per tenere aggiornato e questo sito.



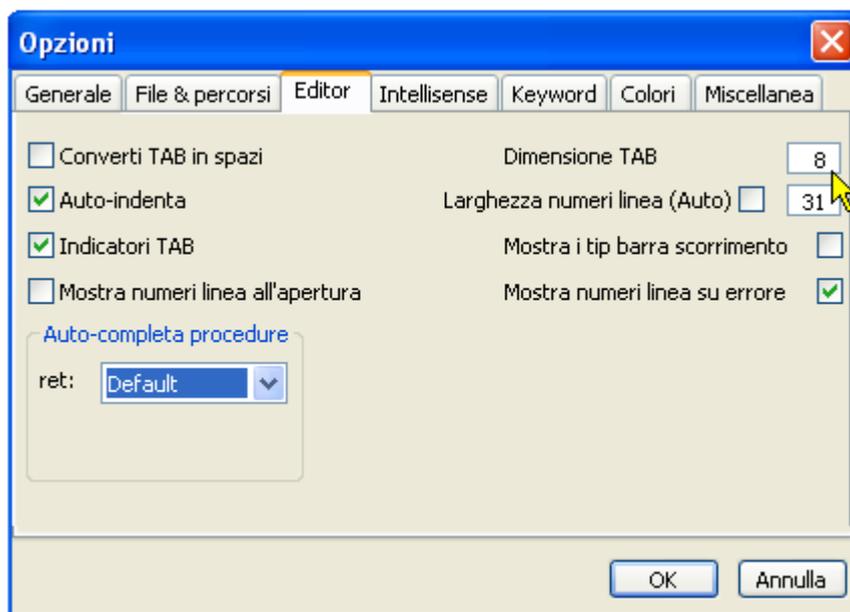
**NB:** Gli eseguibili **ML.EXE (Macro Assembler Version 6.14.8444 Copyright (C) 1981-1997)** e **LINK.EXE (Segmented Executable Linker Version 5.60.339 Copyright (C) Corp 1984-1994)** sono di **esclusiva proprietà** di **Microsoft Corp.**; il pacchetto MASM (almeno nella forma *stand-alone*) **non è più venduto, supportato e distribuito** da molto tempo: l'ultima versione resa disponibile da Microsoft è stata la **6.14** per la piattaforma **DOS/Win16**, comunque in grado di supportare i processori **Intel e AMD 80x86** con architettura a 32 bit, anche multimediale. Versioni più recenti sono oggi presenti come *accessorio* a prodotti Microsoft più moderni ed attuali, spesso **messi a disposizione gratuitamente** dalla stessa **Microsoft**.

Per questa esplicita ragione questi **2 importanti files** sono **disponibili**, insieme alle *centinaia di applicativi e utility* (di norma **freeware**) offerte di da **numerosi altri autori** (rigorosamente citati e calorosamente ringraziati in occasione di ogni loro presentazione), **con l'impegno di toglierli immediatamente** qualora qualche responsabile **Microsoft** ritenesse la cosa scorretta e/o inopportuna.

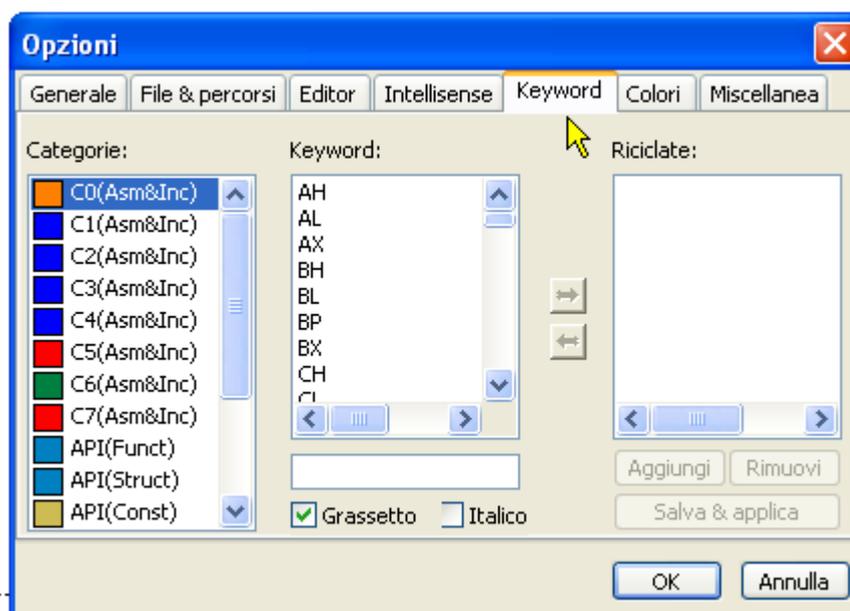
Nella fattispecie *essi* sono stati estratti da **MASM32** dalla **Hutch's Masm32 page (www.masm32.com)**, uno straordinario pacchetto **freeware** nato dallo sforzo di molte persone coordinate in *team di sviluppo* al fine di mettere a disposizione una enorme quantità di librerie, di file di inclusione e di documentazione appositamente pensate per la programmazione **Assembly** sotto Windows (**Win32**) con riferimento a tutte le sue **API**.

La cartella di **MASM32** non può mancare ed è certamente **da installare**, *in parallelo* a quella del mio **Ambiente**, nella futura prospettiva di sviluppare i progetti windows, magari proprio a partire dalla struttura offerta da **WinAsm Studio**.

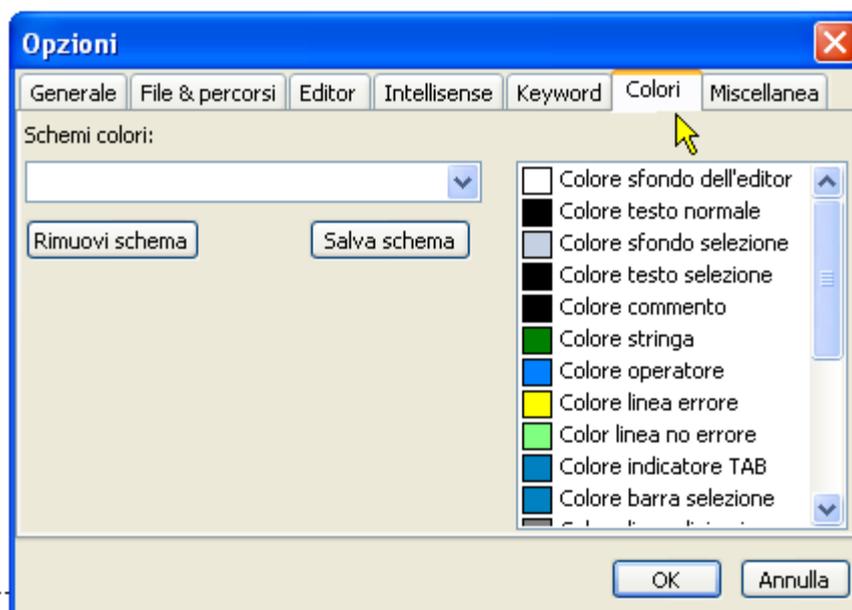
Detto questo, passiamo alle opzioni destinate all'ambiente di editazione, confermando le scelte suggerite in figura per la cartella **Editor**:



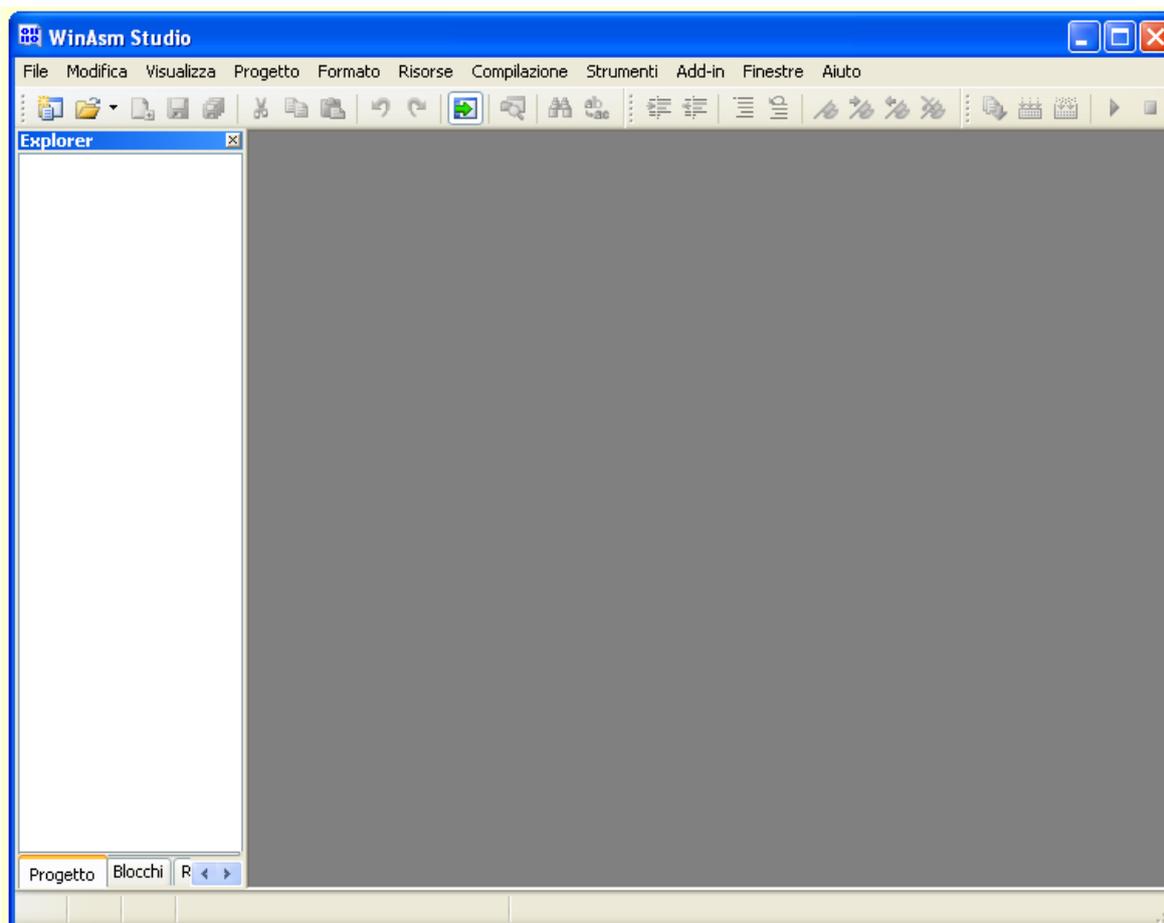
**5.** La successiva messa a punto è quella che da forte **personalizzazione al testo dei sorgenti ASM**; si avvale delle informazioni raccolte nel file **MASM.vas** appositamente creato per consentire a **WinAsm Studio** di riconoscere, evidenziandoli, i principali comandi assembly, e depositato (anche in versione originale) nella sottocartella **\_WinAsm\KeyFiles\**; dal menu in alto, via **Strumenti > Opzioni > Keyword**, è possibile intervenire su ciascuna categoria si evidenziatori, per modificare eventualmente le scelte da me imposte nel file citato:



Le opzioni della cartella **Colori** consentono di fissare i colori di gestione nel campo di editazione:



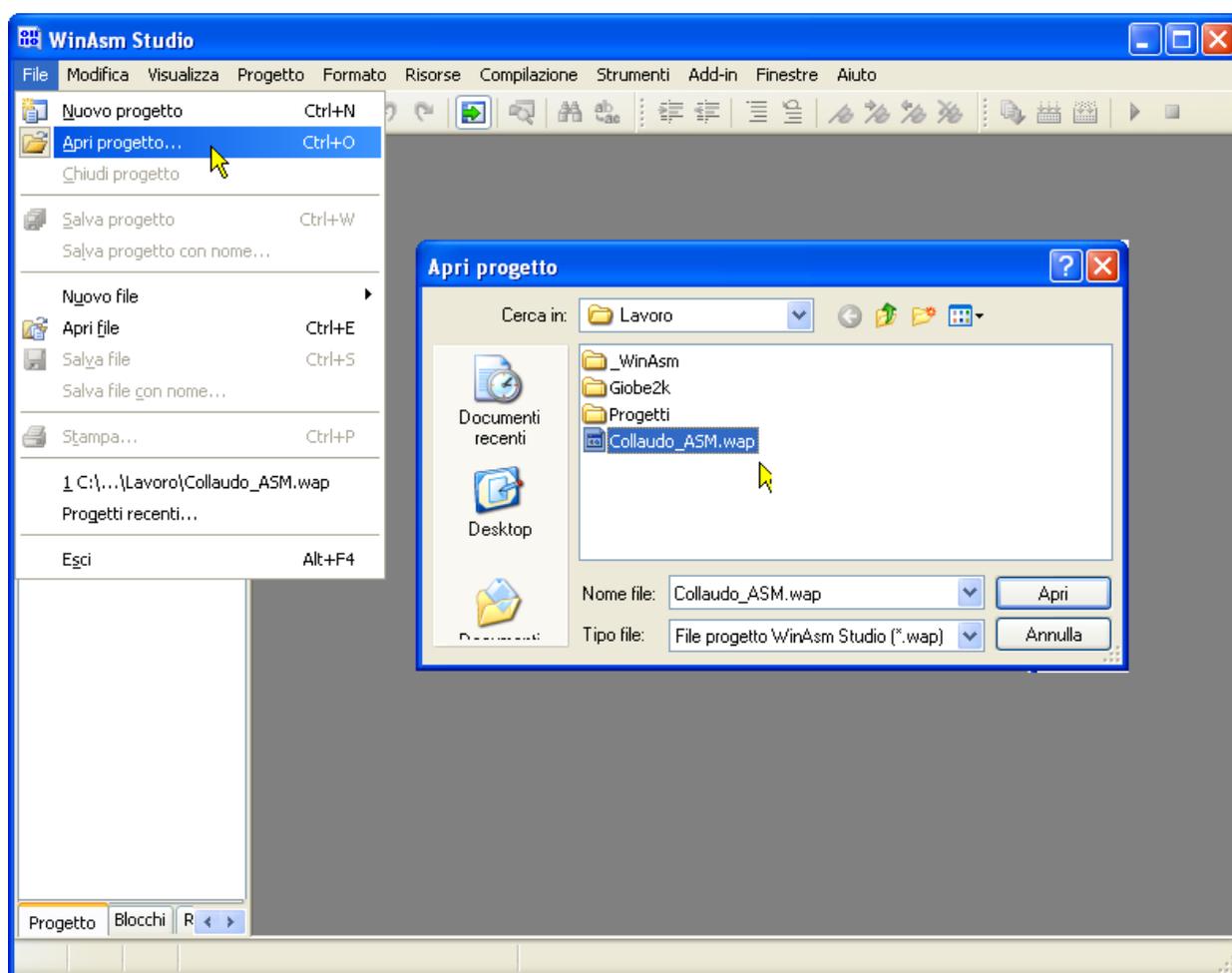
**6.** La prima volta che si avvia **WinAsm Studio** viene proposta una situazione simile a questa:



Il piano di lavoro è completamente vuoto come la *finestra di dialogo* sul lato destro, nominata **Explorer**. Per dare inizio alla realizzazione di un eseguibile a partire da un sorgente assembly è necessaria la creazione di una *struttura* detta **nuovo progetto**; per rendere immediata questa operazione ho provveduto a renderne disponibile uno già pronto, di nome **Collaudo\_ASM.wap** e predisposto nella cartella

**C:\Arch-lab\Lavoro=%SystemDrive%\Arch-Lab\Lavoro;**

basta cliccare su **Apri progetto ..** dalla voce **File** del menu in alto, selezionando poi il file appena indicato:

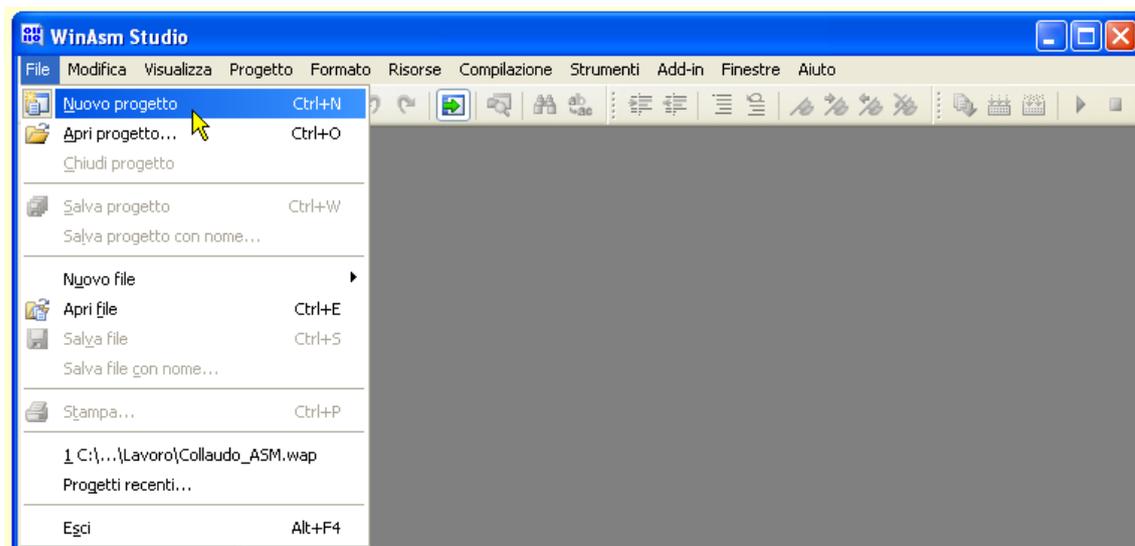


Confermando con **Apri** il piano di lavoro cambia in questo modo, offrendo il file sorgente pilota **Prova.ASM**, pronto per ogni tipo do collaudo:

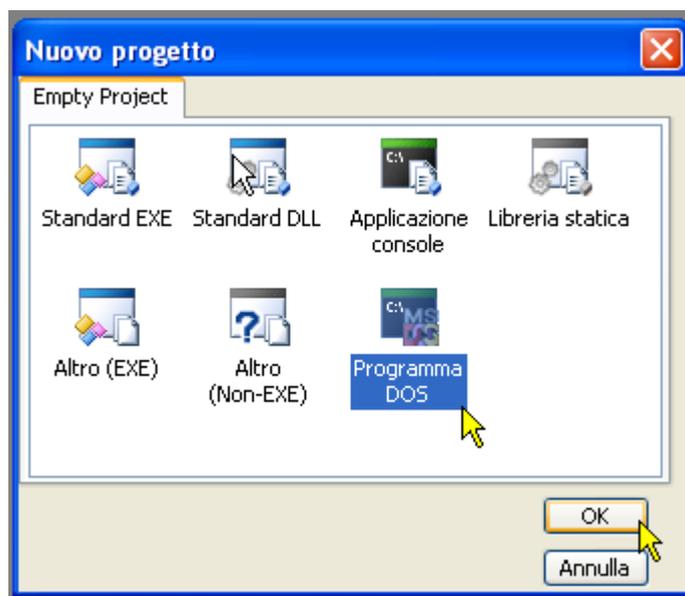
```

0001 PAGE 66,132
0002 TITLE ** PROGRAMMA di PROVA per gli amici NUOVI SOCI del Club
0003 SUBTTL ** TUTORIAL ASSEMBLY -- www.giobe2000.it -- by ing.
0004 ;/-----
0005 ;| NOME      : Prova.ASM
0006 ;| AUTORE   : Giorgio OBER
0007 ;| VERSIONE : aprile 2009
0008 ;| DESCRIZIONE : Programma di Prova; stampa a video un messaggio d
0009 ;\-----
0010
0011 ; COSTANTI DEFINITE PER IL PROGRAMMA
0012 ;/-----
0013 ;***** nessuna Costante è prevista per questo programma di Prova
0014 ; nel caso aggiungere in questo punto le eventuali costanti;
0015 ESC_ EQU 27 ;Esempio per il codice del tasto <ES
0016 CR EQU ODH ;Esempio per il codice del tasto <IN
0017 LF EQU OAH ;Esempio per il codice del tasto <IN
0018
0019 ; MACRO UTILIZZATE DAL PROGRAMMA:
0020 include c:\arch-lab\bin\GIOBE.MAC
0021
0022 ; LIBRERIA UTILIZZATA DAL PROGRAMMA [opzione WinASM altrimenti non
0023 include lib c:\arch-lab\bin\GIOBE.LIB
0024
0025 ;PROCEDURE ESTERNE UTILIZZATE dal PROGRAMMA [NEAR, dalla LIBRERIA
0026 ;/-----
0027 EXTRN CambiaCol:NEAR
0028 EXTRN SET cur:NEAR
  
```

Ma vediamo ora come fare per **creare un nuovo progetto** simile a quello *offerto dalla casa*, **Collaudo\_ASM.wap**; la procedura è semplice ed intuitiva: dal menu in alto si prosegue via **File> Nuovo progetto ...**



Sarà offerta una nutrita scelta di eccellenti possibilità; poichè il nostro scopo è quello di progettare un eseguibile con riferimento alla **struttura del DOS**, sceglieremo **Programma DOS** e confermeremo con **OK**:

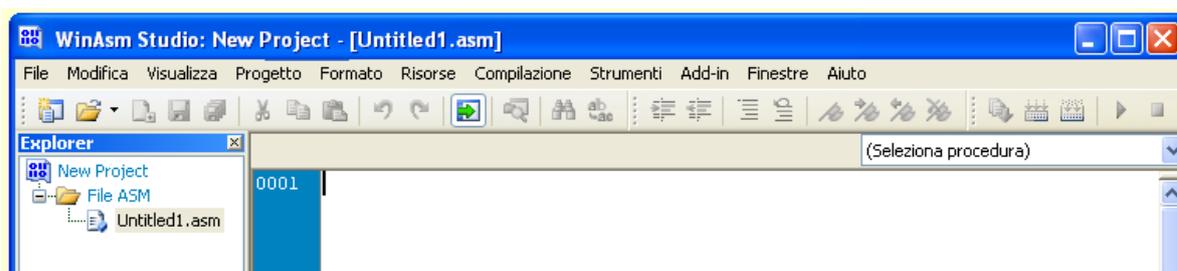


Da notare che, nella zona di download del sito di **WinAsm Studio**, è facile trovare numerosi progetti già pronti per l'uso; per amore di completezza ne ho raccolto ben 67 (a tua immediata disposizione) nella cartella a

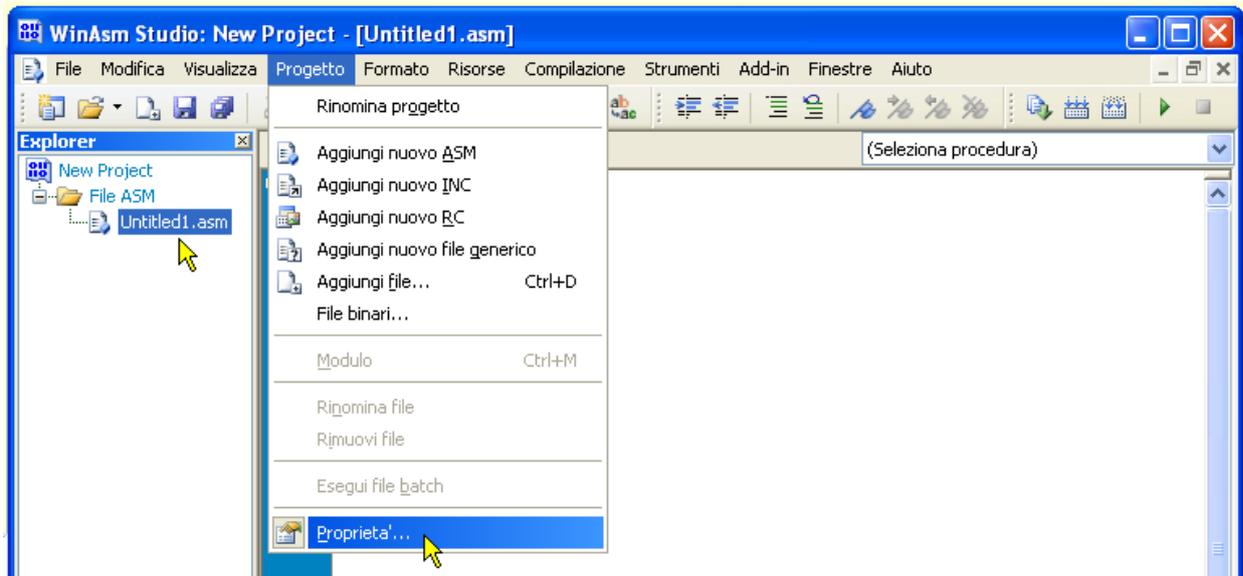
**C:\Arch-lab\Lavoro\\_Winasm\Asm32bit=%SystemDrive%\Arch-Lab\Lavoro\\_winasm\Asm32bit**

in particolare può esserti utile aprire le **matrici DOS\_Com.wap** e **DOS\_Exe.wap** dedicate rispettivamente alla creazione di **eseguibili Dos** di tipo **COM** e di tipo **EXE**.

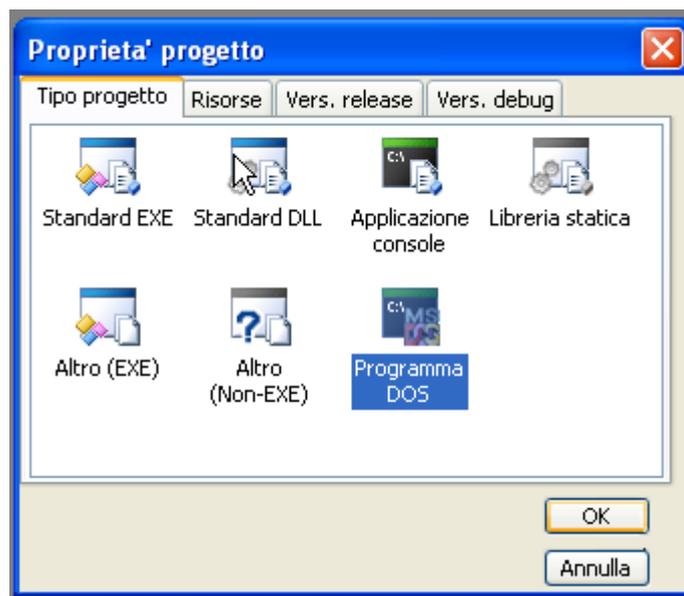
Confermando dunque con **OK** verrà creata la struttura adatta ad ospitare il Nuovo progetto, dichiarato **New Project** e predisposto per ospitare la cartella **File ASM** contenente per default il file **Untitled1.asm**, regolarmente aperto ma ovviamente ancora vuoto (come si vede dal cursore lampeggiante sul primo carattere della riga **0001**, nel piano di editazione):



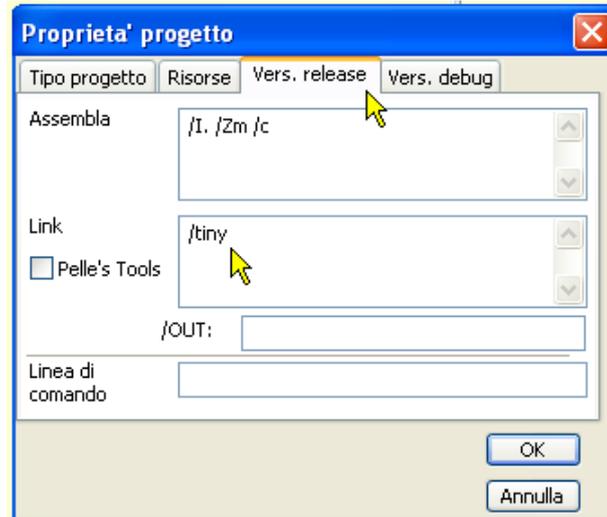
Per prima cosa è ora opportuno stabilire le caratteristiche desiderate per il progetto, fissandone le **proprietà**: cliccando con il tasto destro del mouse sulla parola **New Project** o su **File ASM**, oppure dal menu in alto, via **Progetto** > **Proprietà** ...



... si apre una finestra simile a quella tramite la quale abbiamo deciso il **tipo di progetto**, di fatto proprio aperta su **questa** scheda:



Spostandosi sulla scheda **Vers. release** nel campo **Assembla** inseriremo gli switch **/I. /Zm /c** (attenzione alle minuscole e alle maiuscole) necessari per istruire correttamente l'**assemblatore MASM**, mentre nel sottostante campo **Link**, per istruire il **linker Link16** abbiamo 2 possibilità (in entrambi i casi la fase di **assembler** viene comunque **eseguita correttamente**). Se specifichiamo lo switch **/tiny** verrà generato un **eseguibile COM** (e **non EXE**):



Si tratta della predisposizione di default, la più naturale per la programmazione **Assembly di base** riferita alle *primitive* e alle *strutture* del **DOS** (un sistema operativo a **16 bit**) e alla possibilità di fruire delle *primitive* del **BIOS**, coerente con i presupposti del mio **Ambiente Assembler**; lo switch **/tiny** deve essere presente se nel sorgente si usa **.MODEL tiny** (appunto per generare **COM**); la fase di **linker termina senza errori**, evidenziando la presenza dell'opzione predisposta (**Object Modules [.obj]: /tiny+**) e segnalando il **warning L4045** (non ritenuto errore) che conferma di aver creato un COM e non un EXE:

```

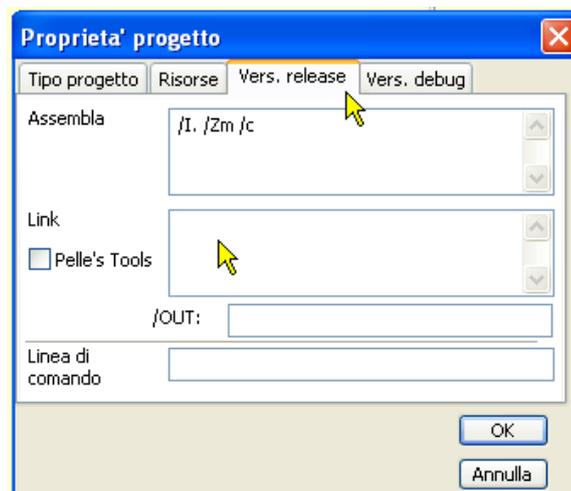
Output
C:\Arch-Lab\_Info\Chk\Link16 @"C:\Arch-Lab\Lavoro\link.war"
Microsoft (R) Segmented Executable Linker  Version 5.60.339 Dec  5 1994
Copyright (C) Microsoft Corp 1984-1993.  All rights reserved.

Object Modules [.obj]: /tiny+
Object Modules [.obj]: "C:\Arch-Lab\Lavoro\PROVA.obj";
LINK : warning L4045: name of output file is 'c:PROVA.com'

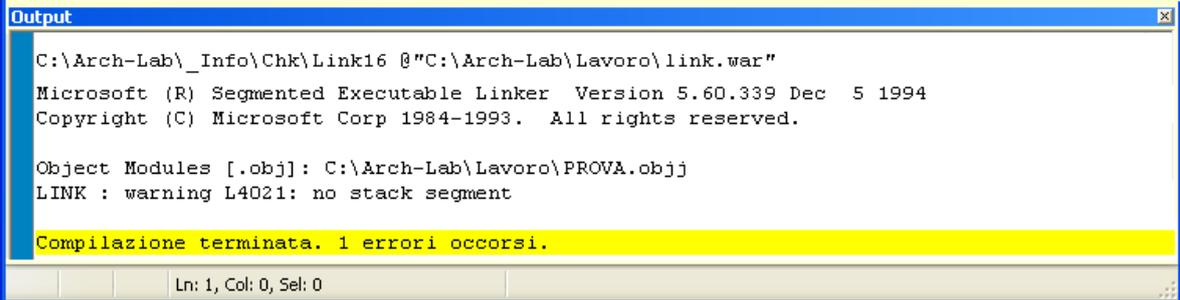
Compilazione terminata. 0 errori occorsi.
Ln: 1, Col: 0, Sel: 0

```

Se **non** specifichiamo lo switch **/tiny** verrà generato un **eseguibile EXE** (e non **COM**):



Lo switch **/tiny non** deve essere presente se nel sorgente si usa **.MODEL small** (appunto per generare **EXE**); la fase di **linker termina segnalando 1 errore**, il **warning L4021** dovuto alla **assenza del segmento di stack**, volutamente tralasciato nella struttura COM del nostro sorgente sottoposto **forzatamente** ad essere compilato come EXE:

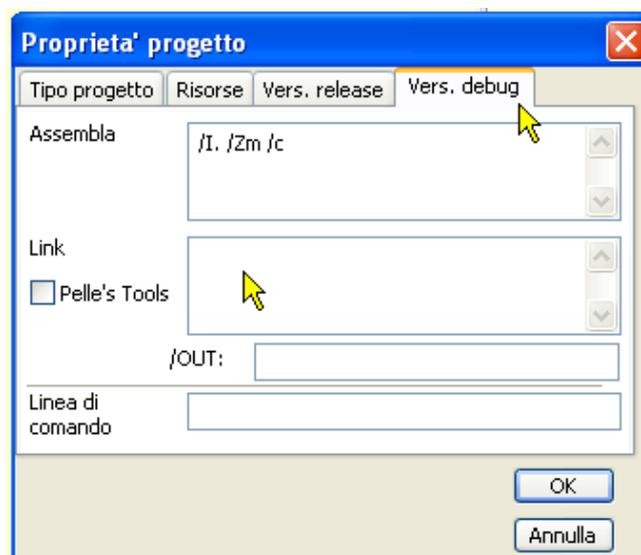
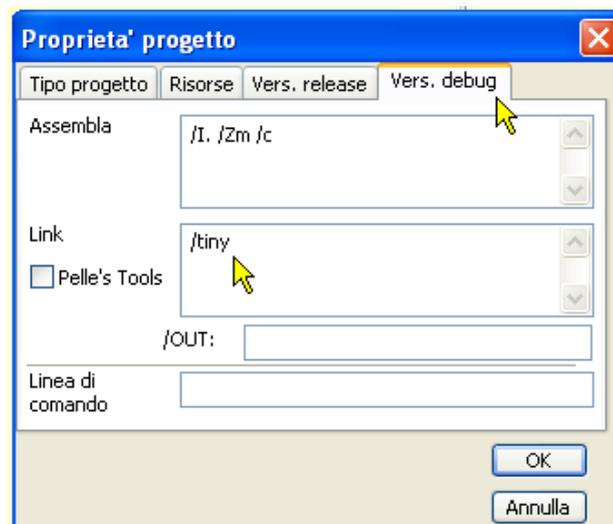


```
Output
C:\Arch-Lab\_Info\Chk\Link16 @ "C:\Arch-Lab\Lavoro\link.war"
Microsoft (R) Segmented Executable Linker  Version 5.60.339 Dec  5 1994
Copyright (C) Microsoft Corp 1984-1993.  All rights reserved.

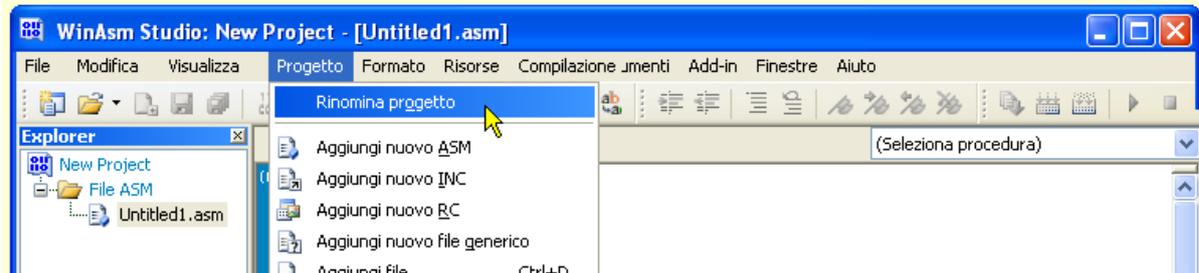
Object Modules [.obj]: C:\Arch-Lab\Lavoro\PROVA.objj
LINK : warning L4021: no stack segment

Compilazione terminata. 1 errori occorsi.
```

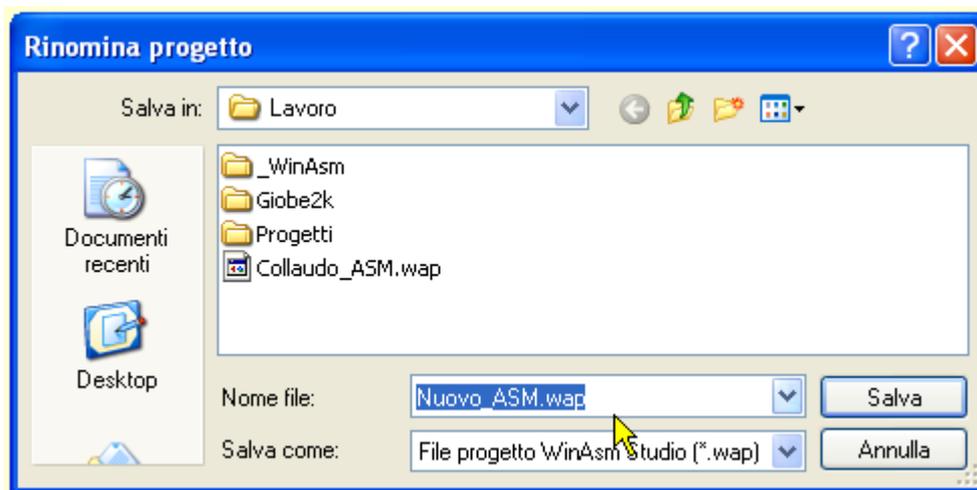
Spostandosi ora sulla scheda **Vers. debug** è possibile (con la stessa logica appena descritta) imporre gli switch nei due campi **Assembla** e **Link**, utilizzati nella eventuale fase di debug:



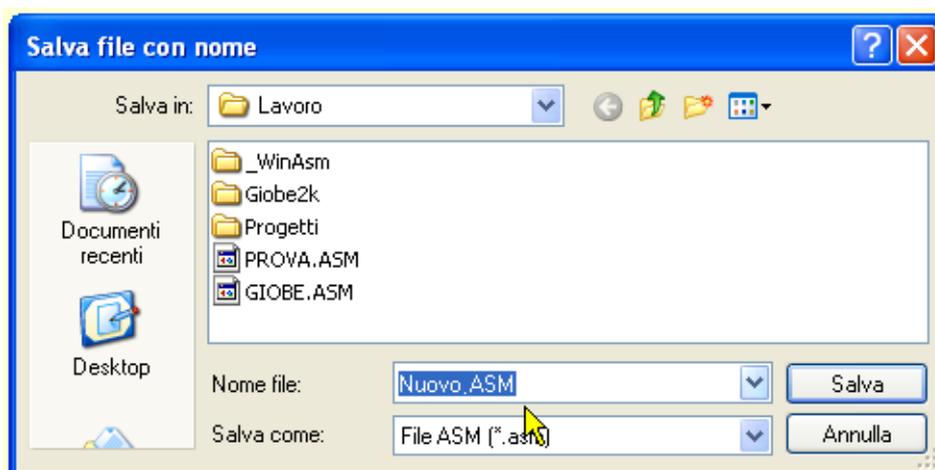
Non rimane che provvedere a dare un nome al progetto e a salvarlo sul disco: i nomi attuali del progetto e del file ASM da esso coordinato sono entrambi ancora generici e provvisori, come si vede rispettivamente **New Project** e **Untitled1.asm**; a partire dal menu in alto, via **Progetto > Rinomina progetto ...**



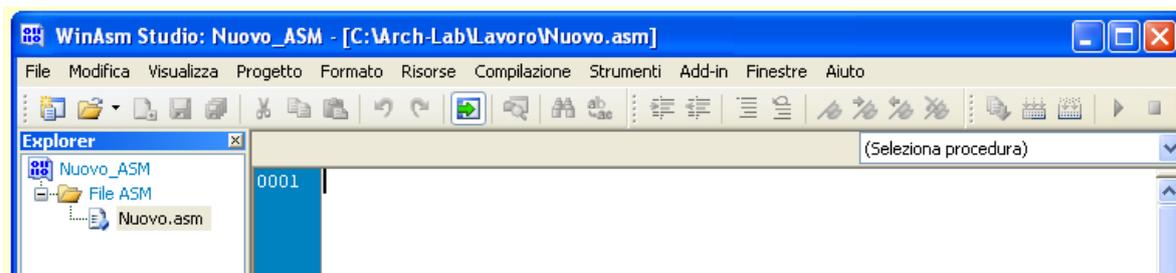
.. si aprono in sequenza le due aree di scelta, *una* per stabilire il nome del nuovo progetto (osserva la presenza di quello che ti ho preparato, **Collaudo\_ASM.wap**, per facilitarti il lavoro), chiamato a sostituire quello provvisorio **New Project** ...



... e l'*altra* per stabilire il nome del file ASM ad esso associato (osserva la presenza di quello che ti ho preparato, **Prova.ASM**, per facilitarti il lavoro), chiamato a sostituire quello provvisorio **Untitled1.asm**:



Al termine la nuova situazione sarà ben rimarcata nella *finestra di dialogo Explorer* sul lato destro, nominata:



È possibile aggiungere nuovi file, cancellare o di importare file esistenti nel progetto tramite il menu Progetto, o facendo clic destro in qualsiasi punto della finestra di Esplora risorse.