

2 Manuali di Giobe2000

TUTORIAL ASSEMBLER

Nuovo Ambiente Assembler

Come si utilizza

con l'editor **PSPad 4.5.2** di **Jan Fiala Prog-Soft s.r.o**

Copyright © luglio 2009

Studio Tecnico ing. **Giorgio Ober** 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 ©

Come si utilizza

con l'editor **PSPad 4.5.2** di **Jan Fiala Prog-Soft s.r.o**

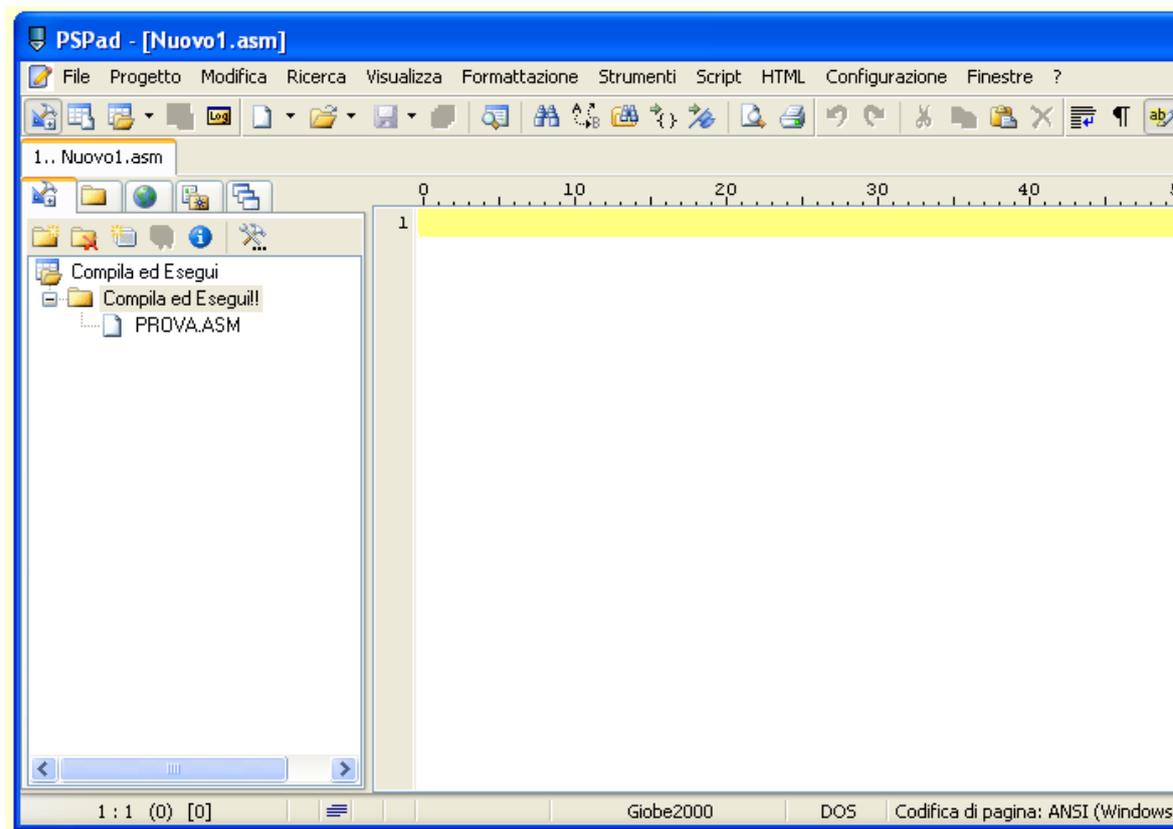
Per accedere alla creazione di un *progetto ASM* con il servizio di **editor** assicurato da **PSPad 4.5.2** [Copyright **Jan Fiala Prog-Soft s.r.o**] basta un *doppio-click* con il *tasto sinistro del mouse* sulla seguente **iconcina**, scelta tra quelle disponibili nella cartella principale del **Nuovo Ambiente Assembler**, **C:\Arch-Lab\=%SystemDrive%\Arch-Lab**:



Anche **PSPad 4.5.2** è stato pazientemente configurato per rendere automatica la fase di sviluppo di un eseguibile; la scelta di includerlo tra le possibili *modalità d'ingresso* al mio **Ambiente** è legata alla sua **straordinaria dotazione** (**unica** nel suo genere) di **funzioni intrinseche** (tra le quali un eccellente *modo colonna*) e di **numerose utility** ad esso integrate (tra le quali un eccellente *editor esadecimale*): decisamente **il migliore editor freeware** presente in rete!, anche se ho rilevato (talvolta, specialmente sotto Vista) qualche imprecisione nel presentare l'esito delle operazioni, nel riquadro ad esso riservato, sotto il sorgente; ma a questo si potrà porre rimedio, pur potendo generare qualche insicurezza, non adatte ad un utente alle prime armi.

Dotato di una *buona e chiara interfaccia grafica*, supportata da *menu in italiano*, è *straordinariamente configurabile* in molti dettagli (tra i quali capacità di interpretare le parole chiave dell'**assembly** con colori diversi) è *un peccato* non poterne disporre e *un peccato* non conoscerlo.

Non appena si clicca sull'iconcina l'**editor PSPad 4.5.2** si apre mostrando questa immagine:



Faccio notare che nella colonna di sinistra viene mostrato il *metodo di default* per *accedere ai files* da coinvolgere con questo programma: quella affidata ad un "**progetto**", una tecnica che **rende disponibili in un unico raccoglitore virtuale tutte le risorse necessarie**, anche se fisicamente localizzate nelle più svariate cartelle del nostro disco; con certi tipi di programmazione (non dimentichiamo che **PSPad** è **splendido** con molti linguaggi) la cosa è di importanza vitale, ma nel nostro caso, avendo a che fare con un semplice e unico **testo ascii** (appunto il nostro sorgente ASM) non è necessario nè opportuno seguire questa via, ... e non la seguiremo!!

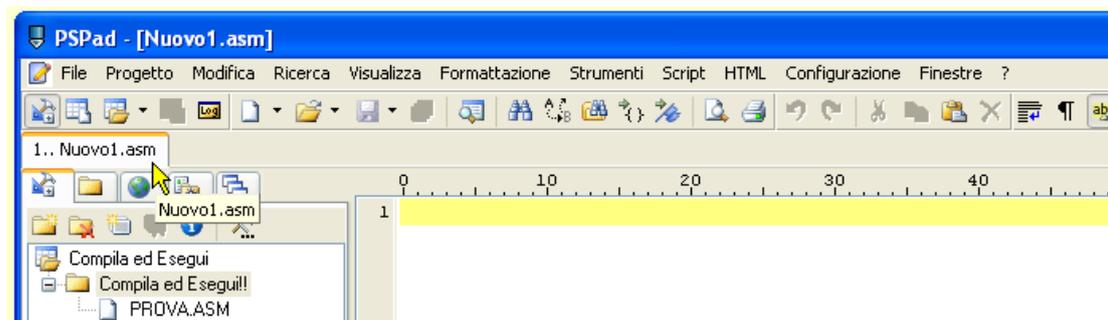
Per non lasciare *vuota* questa opzione *di default* ho comunque predisposto un *nostro* progetto, chiamato "**Compila ed esegui**", contenente un solo file (quello *di prova* classico, **PROVA.ASM**). La creazione e la gestione di questa *azione coordinata* è supportata da appositi files **.ppr**, **Pspad project files**; per conservare nell'**Ambiente** la nostra proposta (di nome "**Compila ed Esegui.ppr**") ho predisposto una apposita cartella di riferimento, **C:\Arch-Lab\Lavoro\Progetti\=%SystemDrive%\Arch-Lab\Lavoro\Progetti**.

Poichè tutto è già **predisposto e funzionante** puoi tentare immediatamente la tua prima compilazione; il *piano d'editazione* è **ancora vuoto**, ma pronto per la **creazione** di un **nuovo sorgente ASM**, nominato d'ufficio **Nuovo1.asm**, come si vede nella *linguetta* in alto a sinistra, sotto il menu.

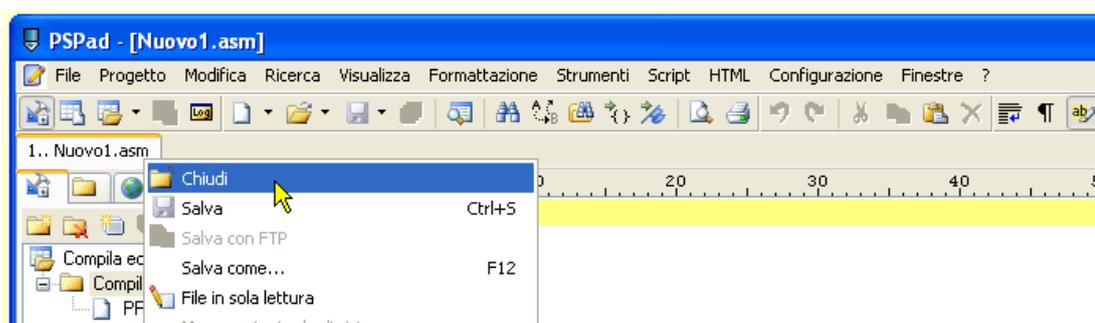
Per prendere confidenza con l'**Ambiente** è però più semplice provare a **caricare un sorgente già pronto**, riservandoci la possibilità di **scriverne uno nuovo** solo tra un po'.

Dunque la cosa da fare è **scegliere** (tra le centinaia di proposte offerte dal mio **Ambiente**) un

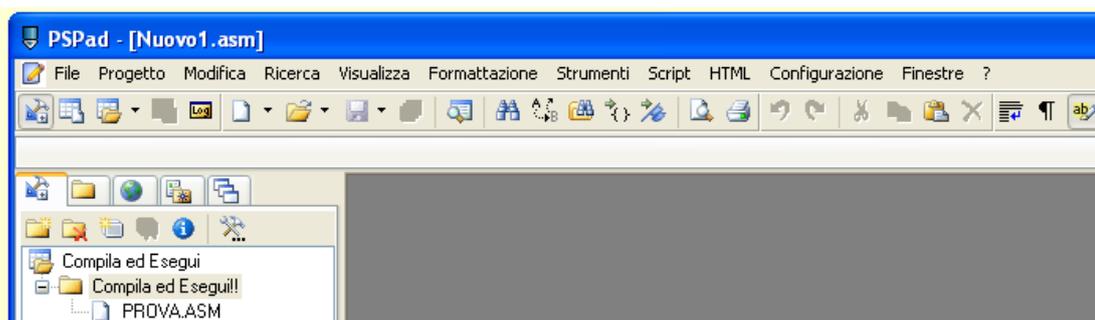
file con estensione ASM, ma prima chiudiamo il sorgente vuoto aperto d'ufficio; puntiamo la linguetta con il nome (**Nuovo1.asm**) del file da chiudere ...



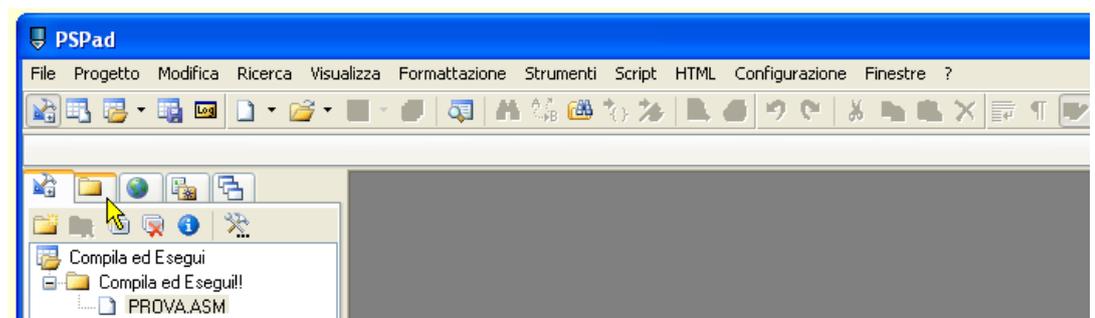
... e, con il *tasto destro del mouse* , scegliamo l'opzione **Chiudi**:



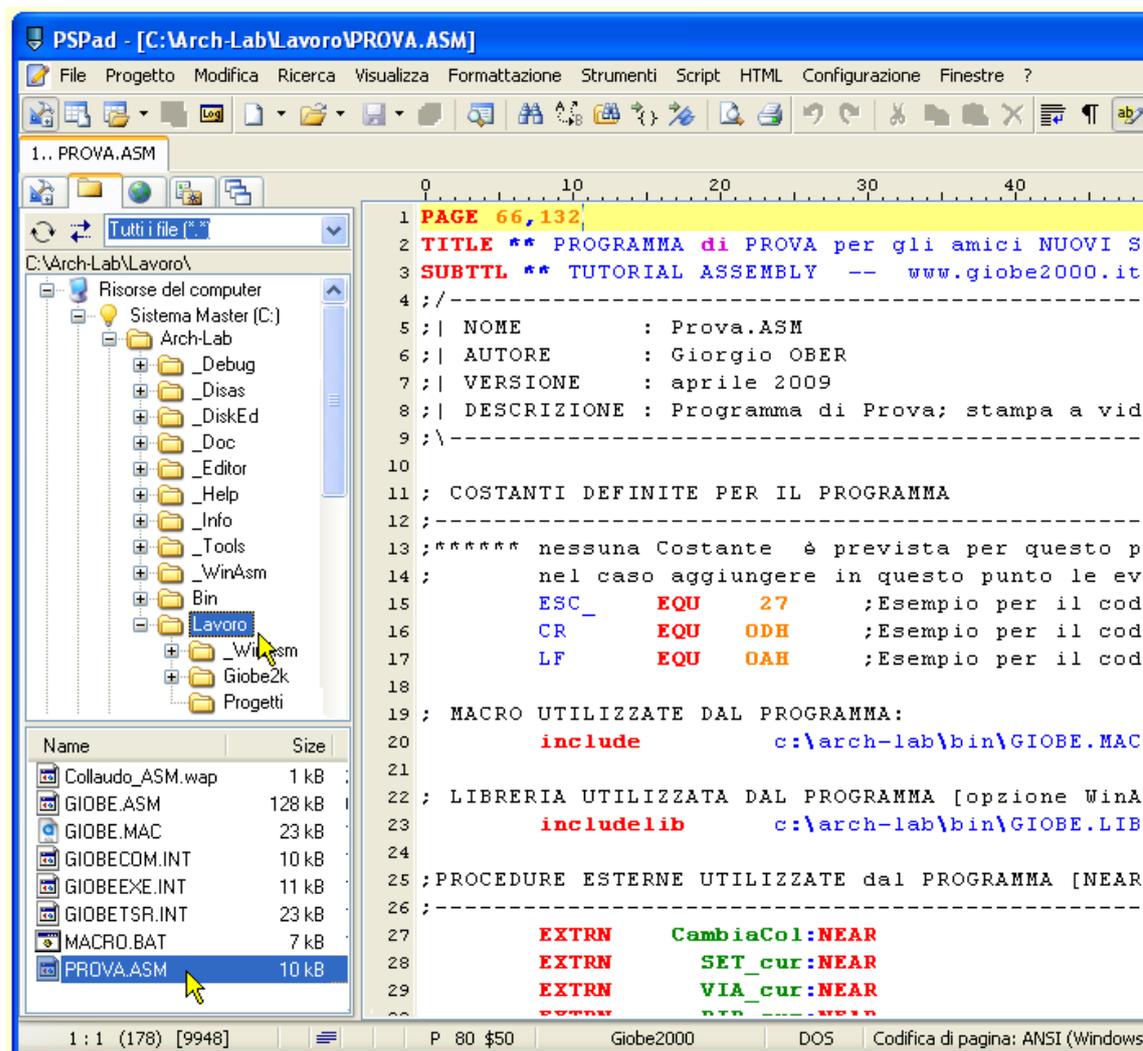
Il *piano d'editazione* tornerà vuoto e grigio:



Puntiamo ora la seconda cartella di selezione, cliccandoci sopra per adeguare il contenuto della colonna di sinistra con le nuove informazioni:

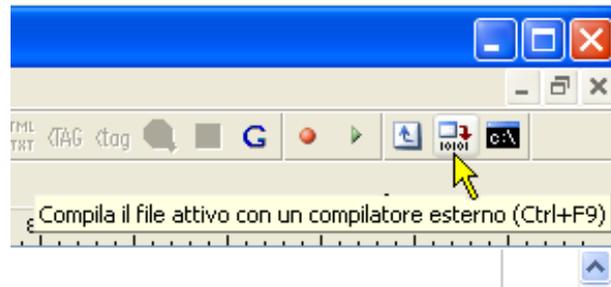


Cerchiamo con pazienza *nella metà alta* la *cartella di lavoro standard* del nostro **Ambiente**, **C:\Arch-Lab\Lavoro=%SystemDrive%\Arch-Lab\Lavoro**, e *clicchiamoci sopra* per mettere in chiaro (*nella metà bassa*) l'elenco dei files disponibili; scegliamo dall'elenco il sorgente **PROVA.ASM**, già bell'e pronto. Facendo *doppio-click* sulla sua icona con il *tasto sinistro del mouse* l'immagine cambia così:



Guardando l'immagine già puoi capire come verrà trattato il testo presente nel documento, evidenziandone **a colori** ogni *dettaglio sintattico e funzionale*; con la *barra di scorrimento laterale* (a destra) puoi scorrere tutto il sorgente e cercare di capire la sintassi necessaria per un progetto assembly: per aiutarti in questa analisi ognuno dei sorgenti offerti dal mio sito è **sempre ampiamente commentato**.

Se sei sufficientemente preparato puoi tentare di modificare qualche istruzione ma prima è conveniente imparare a **rendere eseguibile il sorgente**, così com'è; per provvedere a questa operazione (che consiste nella gestione completa della *catena compilazione*) basta *clicare* sulla **seconda iconcina** posta in alto a destra, sotto il menu dell'editor (attivabile anche con **Ctrl + F9**):



Immediatamente parte un gestore che organizza e coordina **automaticamente** ogni operazione necessaria, provvedendo a lanciare in sequenza l'**assembler**, il **linker** e l'**exe2bin**, in stretta coerenza con quanto descritto nel **capitolo2** del **Tutorial Assembly**; l'area occupata dall'editor si divide in due parti, comprimendo le informazioni visualizzate in precedenza *nella metà superiore* e segnalando, *nella metà inferiore*, l'esito finale di questa fase; **se tutto va bene** avremo un messaggio simile a quello catturato qui sotto:

```

Log Risultati ricerca Risultati ricerca nei file FTP
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997. All rights reserved.

Assembling: C:\Arch-Lab\Lavoro\PROVA.asm

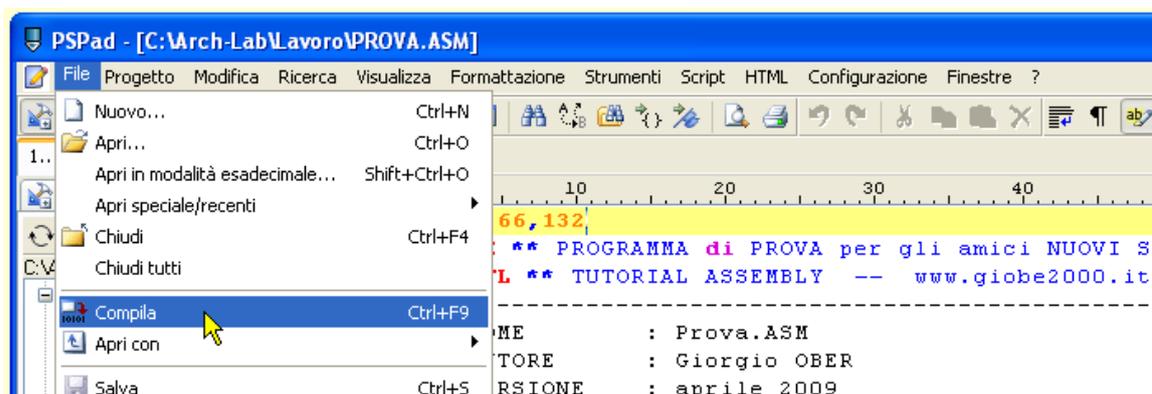
Microsoft (R) Segmented Executable Linker Version 5.60.339 Dec 5 1994
Copyright (C) Microsoft Corp 1984-1993. All rights reserved.

• LINK : warning L4021: no stack segment
EXE2BIN R1.01 Convert EXE files to BIN or COM files.
Copyright (C) 1989,1997 Caldera, Inc. All rights reserved.

Source file      : C:\Arch-Lab\Lavoro\PROVA.exe
Destination file : C:\Arch-Lab\Lavoro\PROVA.com

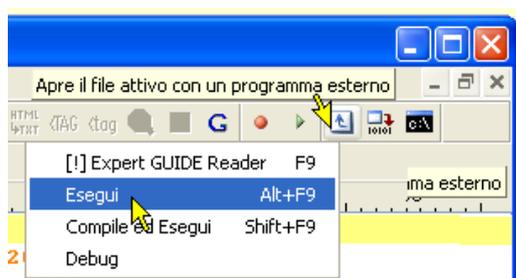
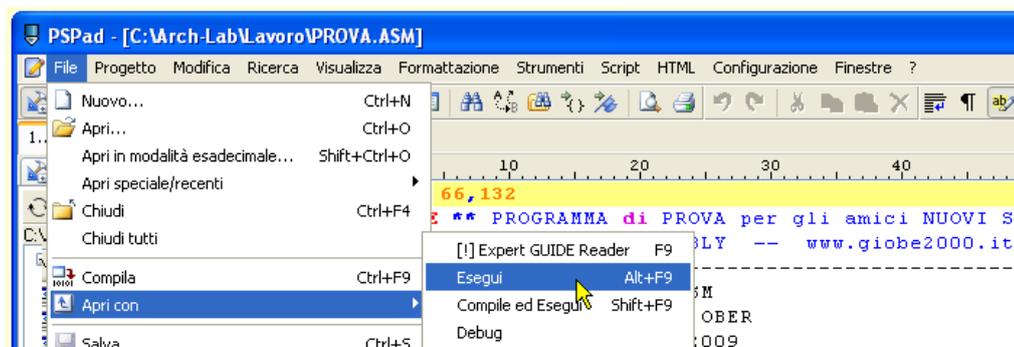
Process completed, Exit Code 0.
Execution time: 00.00.453
  
```

Gli stessi risultati possono ottenuti anche a partire dal menu in alto, via **File > Compila**:

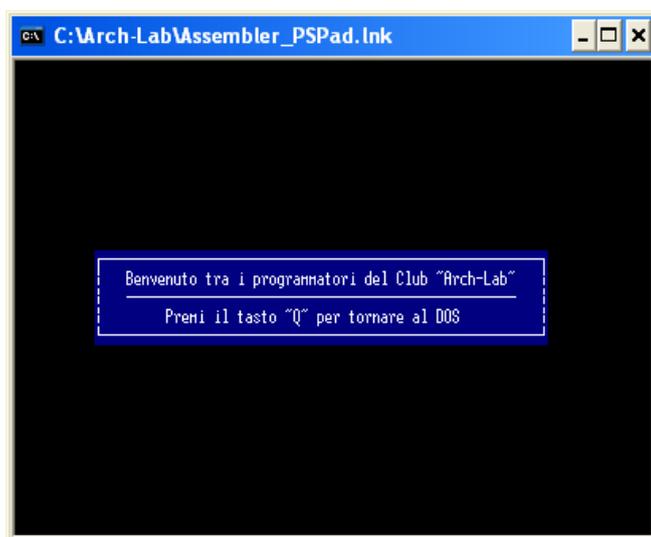


In entrambi i modi, durante le operazioni di creazione, si aprirà (oltre al riquadro che ne segnala l'esito) *anche* una **shell dos "in finestra"** (più volte e per brevi istanti), in corrispondenza dell'apertura e della chiusura degli strumenti coinvolti, ma **non è previsto** (salvo eccezioni dovute al progetto in esecuzione) **che la shell dos rimanga aperta** con il **file eseguibile EXE o COM in esecuzione**.

Per vedere l'effetto del tuo lavoro (cioè per mettere **solo in esecuzione** l'**eseguibile EXE o COM** appena compilato e linkato) è dunque previsto un altro comando, attivabile direttamente con **Alt + F9**, oppure a partire dal menu in alto, via **File > Apri con > Esegui**, oppure **cliccando** sulla **prima iconcina** posta in alto a destra, sotto il menu dell'editor:

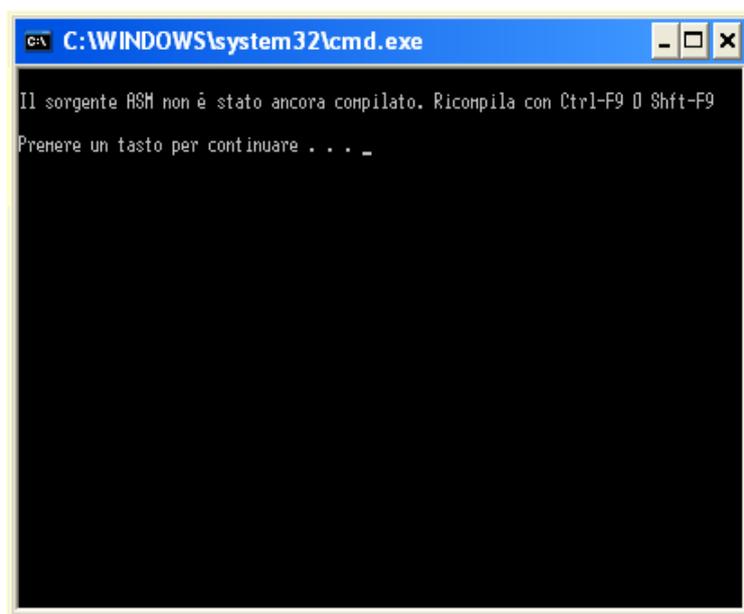


L'effetto prodotto è quello di mostrare l'**eseguibile in esecuzione** in una **shell dos "in finestra"**; per esempio con il nostro **sorgente pilota, PROVA.asm**, vedremo:



Ricordo che, per fruire di maggiore dettaglio è possibile aprire il progetto "*a pieno schermo*" semplicemente premendo **ALT + Invio** (questo con tutte le versioni di Windows **meno Vista!!**); premendo di nuovo **ALT + Invio** si ritorna alla esecuzione della *shell dos* "*in finestra*".

Naturalmente se il sorgente **non è stato ancora compilato** l'operazione non sarà possibile e, in alternativa al servizio, verrà suggerito cosa fare:



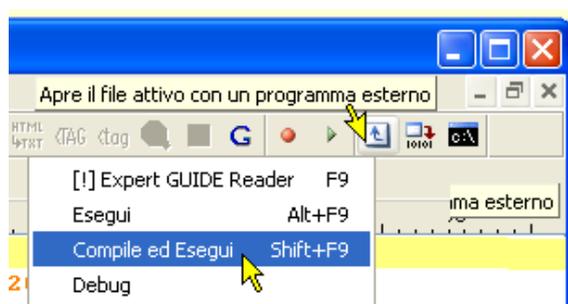
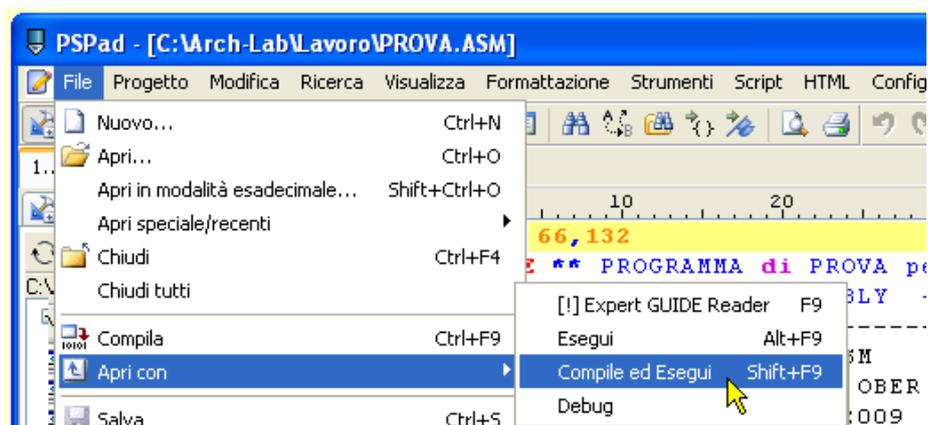
Anche questo potente editor consente di **gestire** fino a 5 **particolari programmi esterni**; approfittando di questa opportunità ho voluto assicurare servizio a 4 delle 5 opzioni disponibili.

La **prima** opzione è stata descritta poco fa, essendo destinata a garantire l'esecuzione del frutto della compilazione del sorgente attualmente *sotto i ferri*..

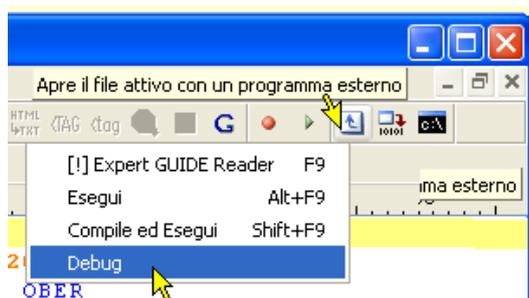
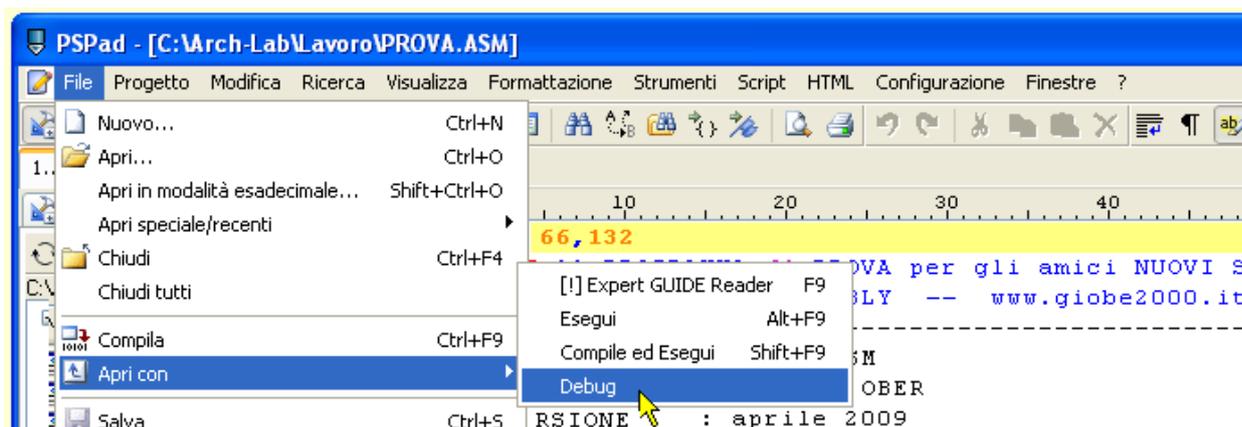


Riassumendo: dopo aver **caricato** (o **creato**) un sorgente ASM potrai prima **compilarlo** con **Ctrl F9** e poi **vederlo** con **Alt F9**

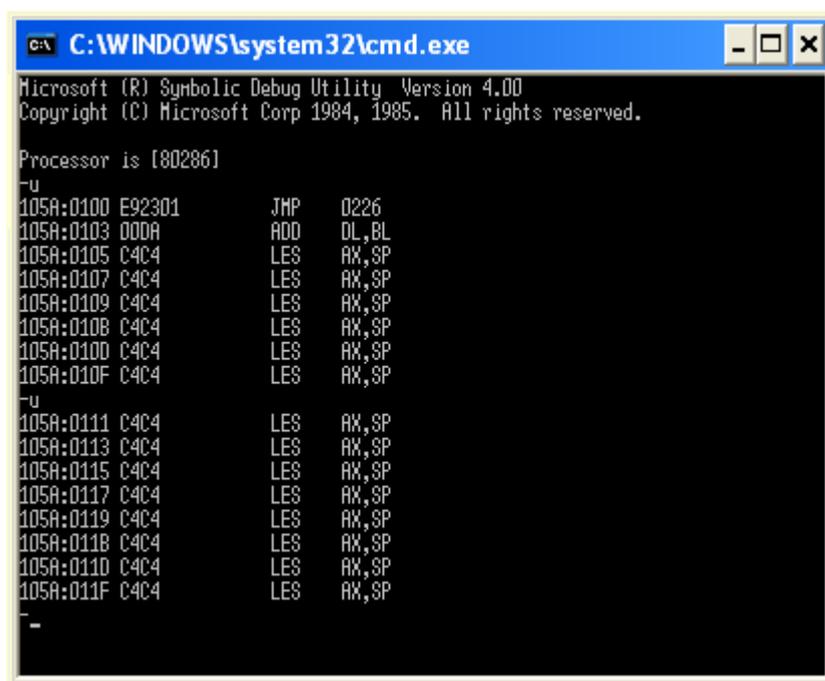
La **seconda** opzione ripropone la sequenza di comandi, appena descritta, che gestisce la *catena compilazione*, ed è studiata per mettere in atto **entrambe** le azioni *conseguenzialmente*, cioè per **compilare e vedere con un comando solo**; sfortunatamente il modo usato da **PSPad** per eseguire i comandi esterni è piuttosto rigido, tanto da rendere questa seconda proposta **meno efficiente** della precedente, perchè **non gestisce il riquadro dei messaggi**, e quindi è **poco pratica!!** Ciononostante fa il suo lavoro e, a piacere, può essere attivata direttamente, con **Shift + F9**, oppure a partire dal menu in alto, via **File > Apri con > Compila ed Esegui**, oppure *clickando* sulla **prima iconcina** posta in alto a destra, sotto il menu dell'editor:



Il **terzo** comando, tra i 4 possibili, è programmato per attivare automaticamente la **fase di debug** dell'eseguibile *cliccando* sulla **prima iconcina** posta in alto a destra, sotto il menu dell'editor, oppure a partire dal menu in alto, via **File > Apri con > Debug**:



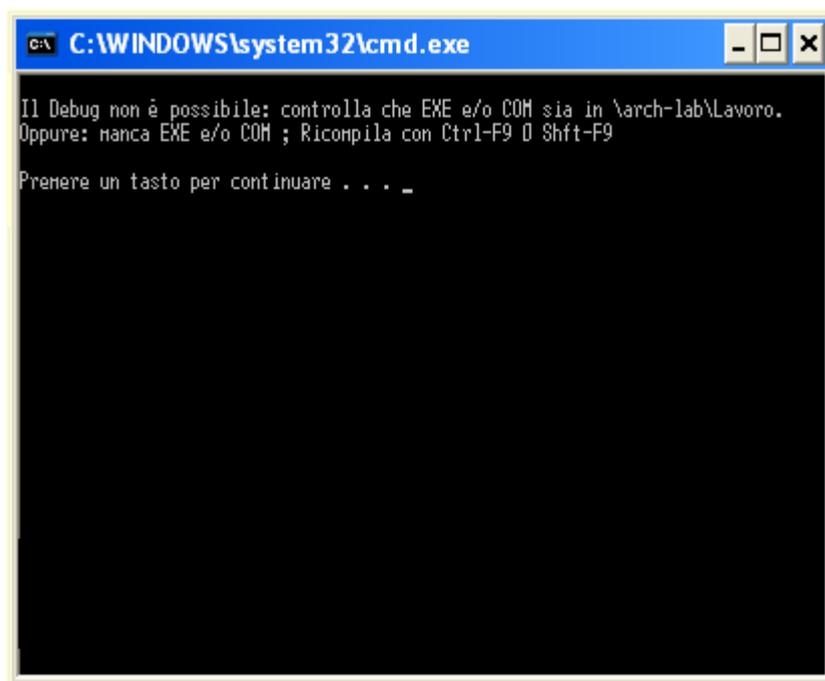
Se l'eseguibile è disponibile (cioè se la fase di compilazione è stata portata a termine senza errori) si aprirà ancora una *shell dos "in finestra"*, anch'essa a piacere gestibile *"a pieno schermo"* con le modalità descritte poc'anzi:



```
C:\WINDOWS\system32\cmd.exe
Microsoft (R) Symbolic Debug Utility Version 4.00
Copyright (C) Microsoft Corp 1984, 1985. All rights reserved.

Processor is [80286]
-u
105A:0100 E92301      JMP    0226
105A:0103 000A      ADD    DL,BL
105A:0105 C4C4      LES   AX,SP
105A:0107 C4C4      LES   AX,SP
105A:0109 C4C4      LES   AX,SP
105A:010B C4C4      LES   AX,SP
105A:010D C4C4      LES   AX,SP
105A:010F C4C4      LES   AX,SP
-u
105A:0111 C4C4      LES   AX,SP
105A:0113 C4C4      LES   AX,SP
105A:0115 C4C4      LES   AX,SP
105A:0117 C4C4      LES   AX,SP
105A:0119 C4C4      LES   AX,SP
105A:011B C4C4      LES   AX,SP
105A:011D C4C4      LES   AX,SP
105A:011F C4C4      LES   AX,SP
-
```

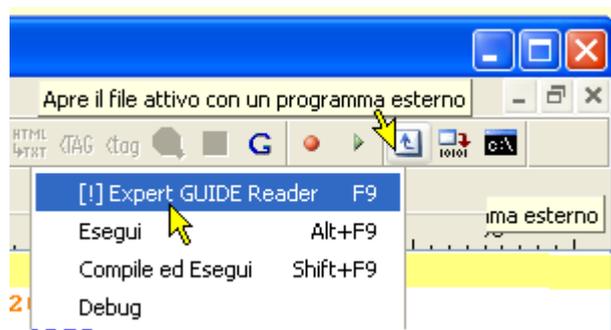
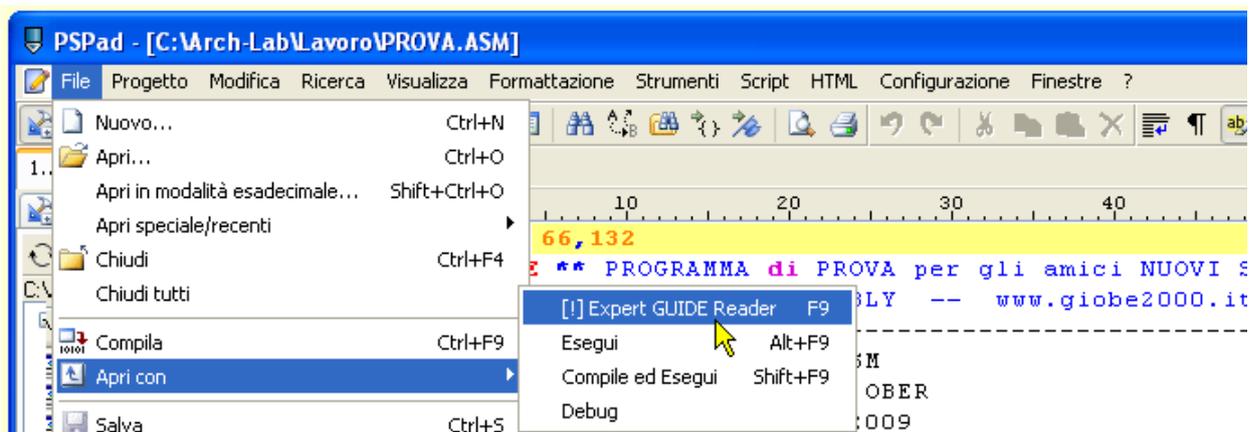
Se l'eseguibile *non è ancora stato creato* l'operazione di debug non sarà possibile e, in alternativa al servizio, verrà suggerito cosa fare:



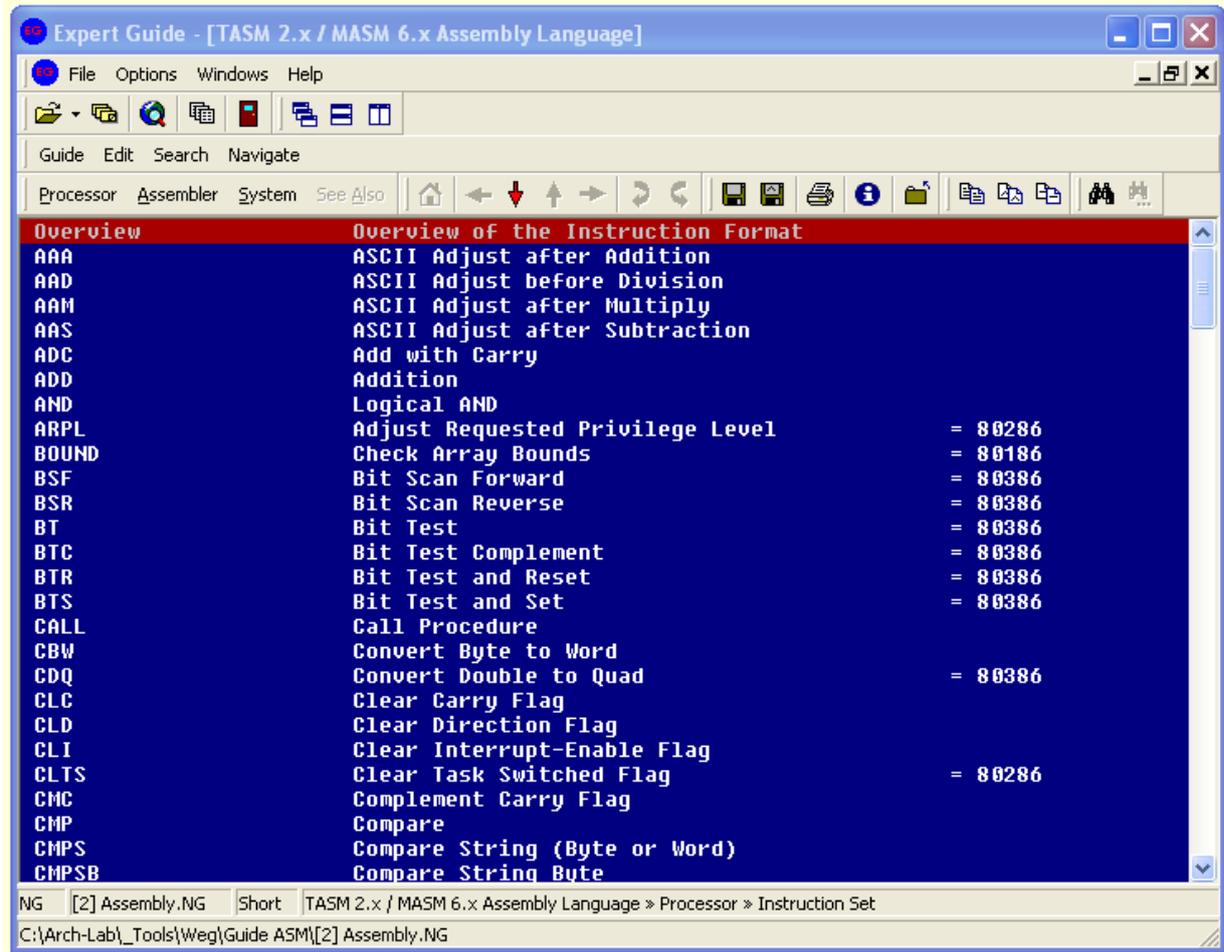
```
C:\WINDOWS\system32\cmd.exe
Il Debug non è possibile: controlla che EXE e/o COM sia in \arch-lab\lavoro.
Oppure: manca EXE e/o COM ; Ricompila con Ctrl-F9 o Shift-F9
Premere un tasto per continuare . . . _
```

Rimane da descrivere l'ultimo dei 4 servizi garantiti dall'editor **PSPad** ad un programma esterno; si tratta di una speciale e grossa opportunità che ho pensato di aggiungere, per aiutarti a **scoprire molti gioielli** di questo nostro *tesoro*.

Cliccando sulla **prima iconcina** posta in alto a destra, sotto il menu dell'editor, oppure a partire dal menu in alto, via **File > Apri con > [!] Expert GUIDE Reader**, oppure direttamente, con **F9**, si aprirà il programma **Weg_2.15** [Copyright **Dave Pearson**], uno straordinario **gestore di archivi di dati** (**Expert Guide Reader for Windows**), già configurato per vedere una **ricchissima raccolta** di informazioni tipiche dell'*assembly*.



La figura seguente ti da un assaggio di quello che potrai avere, ma non basta!! Tramite il menu di questo programma (**File > Open..**) selezionando il percorso **C:\Arch-Lab_Tools\Weg\Guide ASM**=%SystemDrive%\Arch-Lab_Tools\Weg\Guide ASM\ puoi aprire più di una ventina di importanti alternative, da me raccolte e rese disponibili nella cartella citata.



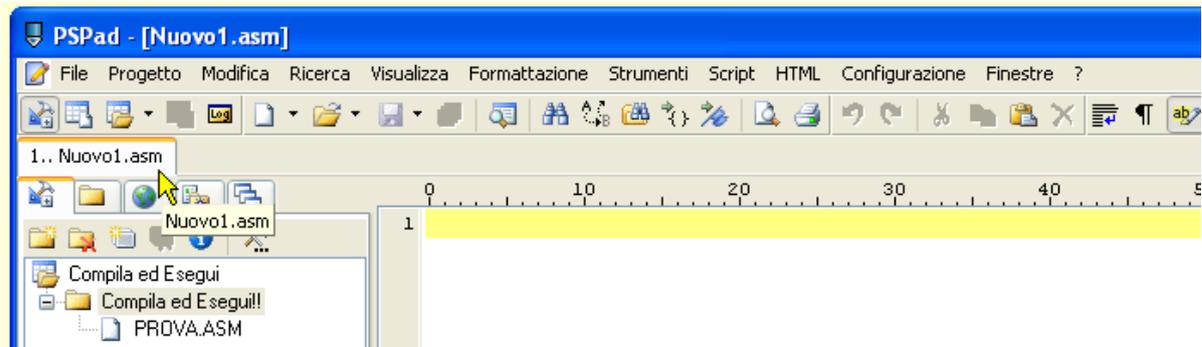
Dal menu inferiore (**Processor Assembler System**) di **Weg_2.15** puoi accedere, inoltre, a tutte le categorie d'informazione necessarie.



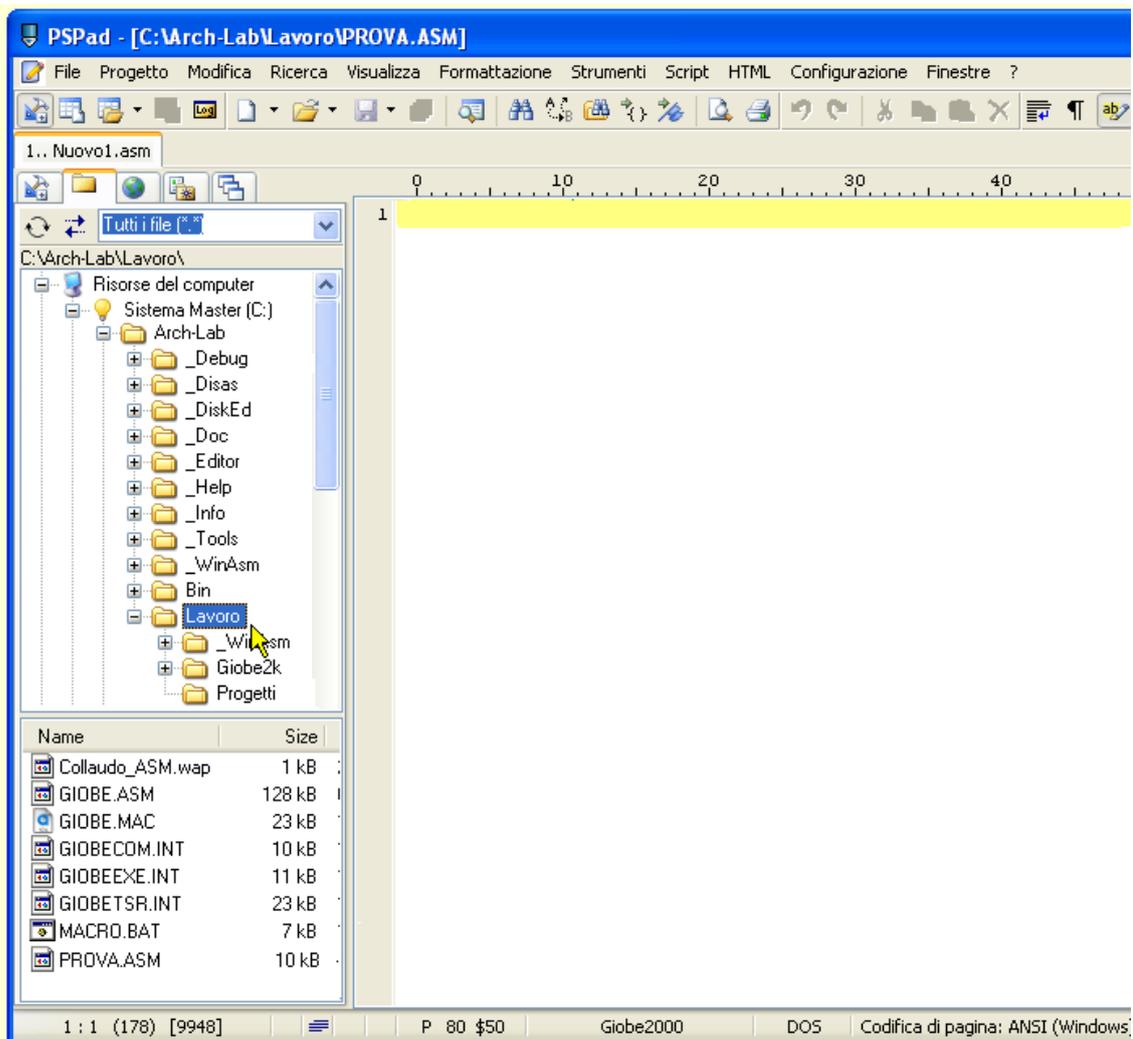
Con questo editor **non è necessario** che il **sorgente ASM** da compilare sia nella cartella di lavoro, **C:\Arch-Lab\Lavoro**, cioè che possa essere selezionato dall'elenco visibile nella colonna di sinistra; se dovesse essere assunto da qualunque altra cartella (nell'ambiente ce ne sono molte !!) per trasferirlo nel **piano di editazione** è sufficiente cliccare sopra la sua icona.

Per finire ti devo dare le indicazioni per **scrivere un nuovo sorgente Assembly**; se chiudi e riapri **PSPad**, come abbiamo già visto, il nostro **editor** si riapre mostrando il **piano d'editazione ancora vuoto** ma pronto per la **creazione** di un nuovo sorgente ASM, nominato

d'ufficio **Nuovo1.asm**, come si vede nella *linguetta* in alto a sinistra, sotto il menu:



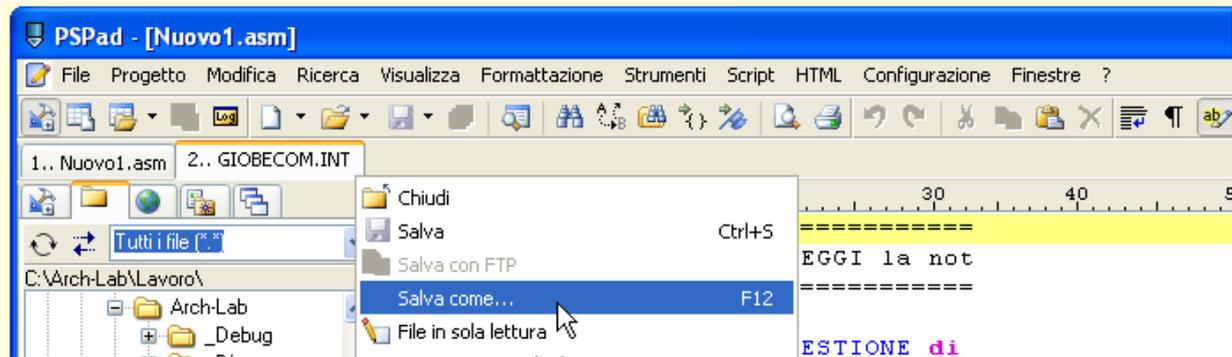
Prima di proseguire è *cosa buona e giusta* abituarsi ad aprire la *cartella di lavoro standard* del nostro **Ambiente**, **C:\Arch-Lab\Lavoro=%SystemDrive%\Arch-Lab\Lavoro**, nel modo descritto all'inizio, per poter disporre delle informazioni correnti contenute nella colonna di sinistra del nostro piano di lavoro:



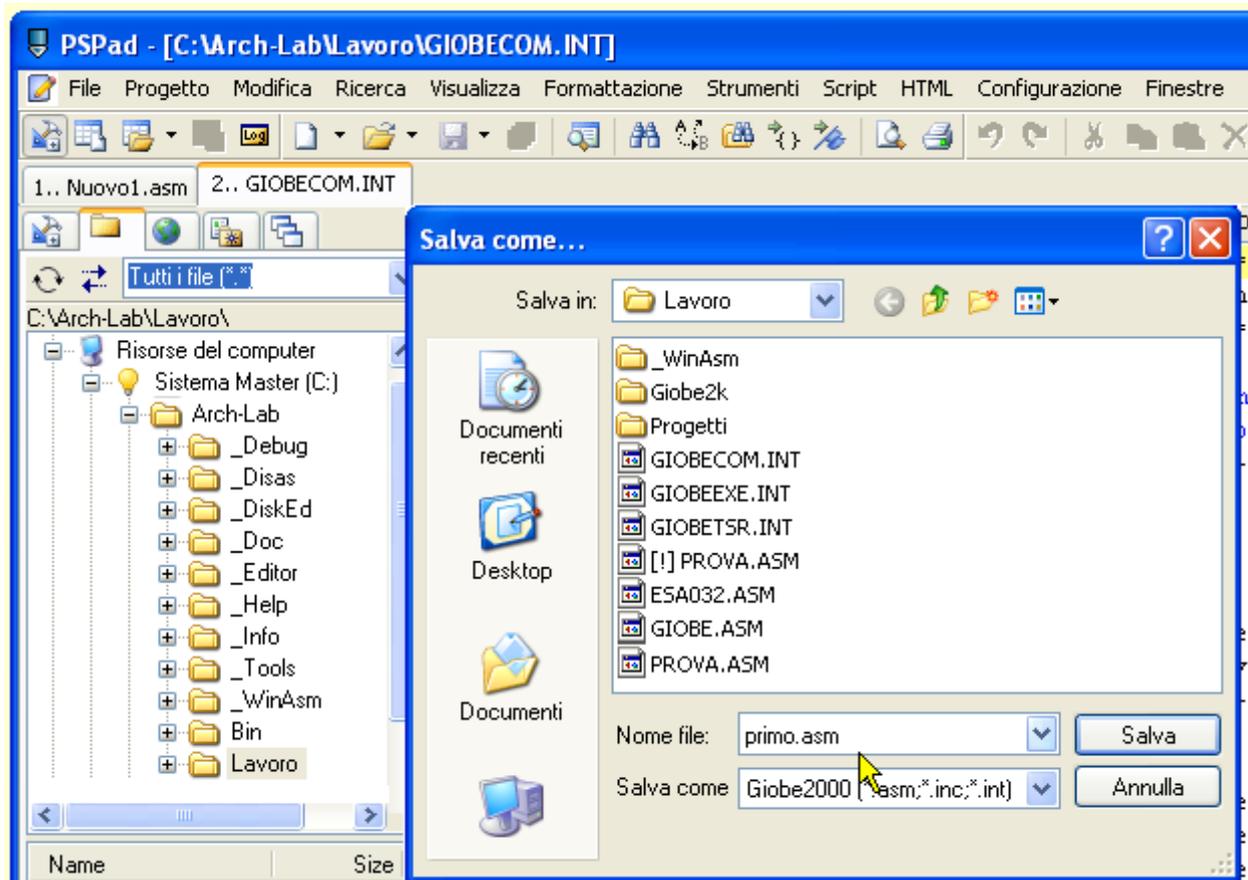
Da questo momento **puoi cominciare a scrivere il tuo nuovo sorgente**, se ne hai la competenza;

della matrice, perdendo la possibilità di riutilizzarla, in futuro.

Ecco come fare: *clickando* con *tasto destro del mouse* sulla linguetta **2.. GIOBECOM.INT** (oppure dal menu **File** in alto) scegli "**Salva con nome**" ..

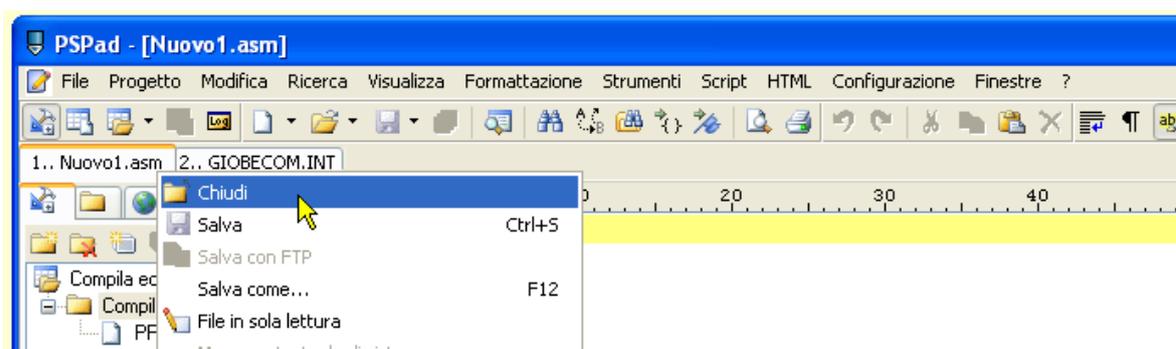


.. e nel riquadro che si aprirà, aggiungi il *nome desiderato* per il tuo sorgente, per esempio **Primo.ASM** ..



Confermando con "**Salva**" esso verrà aggiunto tra i files della cartella di lavoro, **C:\Arch-Lab\Lavoro=%SystemDrive%\Arch-Lab\Lavoro**.

Solo per completezza, se hai scelto questa via rimane ancora aperto **per nulla** il primo file sorgente, inserito d'ufficio all'apertura di **PSPad**, ed è opportuno chiuderlo: **cliccando** con **tasto destro del mouse** sulla linguetta **1.. Nuovo1.asm** (oppure dal menu **File** in alto) scegli "**Chiudi**", e il gioco è fatto!!



Bene!! Non ti resta che **smantare** e **imparare** la nobile arte .. Ma prima di lasciarti ti voglio proporre una situazione nella quale tuo malgrado incorrerai spesso, almeno le prime volte: **che succede se sbagli qualcosa?**

Per fortuna l'**Assemblatore** e il **Linker** sono in grado di accorgersi se qualcosa va storto (come al solito, per ogni dettaglio ti rimando al **capitolo2** del mio **Tutorial Assembly**); il grande vantaggio sta nel fatto che questo magico **editor PSPad** *provvede in proprio* a darti tutte le **informazioni** sull'eventuale **errore** commesso.

Supponiamo di aver digitato male il nome di una **procedura**; per esempio alla **riga 74** del

sorgente **PROVA.ASM** (ricaricalo e scorri il sorgente fino a localizzarle *questo numero*, sulla sinistra del *piano di editazione*) il sorgente prevede l'istruzione `CALL BIOScls`

```

66 ;*****
67 ; -----
68 Prova  proc  NEAR          ;   AREA DEDICATA AL MAIN PROGRAM
69 ; -----
70 ; -----
71 ; -----
72 ; Impostazione della pagina interattiva
73 ; -----
74 CALL    BIOScls           ;Provvede comunque alla cancellazione del video,
75 ;posizionando il cursore alla riga 0, colonna 0
76 CALL    VIA_cur          ;Rende il cursore invisibile
77
78     H_msgC TESTO1,10,10,1FH;
79     H_msgC TESTO2,11,10,1FH; Stampa il messaggio di benvenuto dentro una
80     H_msgC TESTO3,12,10,1FH; cornice
81     H_msgC TESTO4,13,10,1FH;

```

Supponiamo di aver scritto, invece, `CALL BIOSclr`, per disattenzione ...

```

74 CALL    BIOSclr          ;Provvede comunque alla cancellazione del video,
75 ;posizionando il cursore alla riga 0, colonna 0
76 CALL    VIA_cur          ;Rende il cursore invisibile
77
78     H_msgC TESTO1,10,10,1FH;

```

Non appena si provvede alla fase necessaria per **rendere eseguibile il sorgente** (cioè *clickando* sulla **seconda iconcina** posta in alto a destra, sotto il menu dell'editor, oppure premendo **Ctrl + F9**) il gestore che organizza e coordina **automaticamente** la sequenza **assembler + linker + exe2bin**, **si accorge subito** dell'imprecisione, segnalandola nell'area posta nella parte in basso dell'editor:

```

Log Risultati ricerca Risultati ricerca nei file FTP
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997. All rights reserved.

Assembling: C:\Arch-Lab\Lavoro\PROVA.asm
• C:\Arch-Lab\Lavoro\PROVA.asm(74) : error A2006: undefined symbol : BIOSclr

Process completed, Exit Code 0.
Execution time: 00.00.094
8:26 (8) [293] Giobe2000 DOS Codifica di pagina: ANSI (Windows)

```

In questa segnalazione c'è tutto quello che è necessario sapere: il **pallino rosso** (che in caso di esito positivo sarà **verde**) indica **su quale riga** del testo sorgente ("**PROVA.asm(74)**") si è

manifestato l'errore, chiaramente alla riga 74 del sorgente PROVA.asm, e **il tipo d'errore** ("undefined symbol: BIOSclr", chiaramente un simbolo non riconosciuto, essendo BIOSclr diverso da BIOScls).

Ti faccio notare che la mia personale predisposizione delle tabelle per la colorazione della sintassi dei principali comandi assembly (raccolte nel file **Giobe2000.ini** e conservato nella sottocartella **C:\Arch-Lab_Editor\PSPad\Syntax**) prevede una **colorazione verde** per tutte le strutture della **preziosa libreria Giobe.LIB**: già il fatto che l'etichetta sotto esame abbia **perso il colore** può insinuarti il dubbio che essa sia stata scritta in modo sbagliato!!

Certo, ora dovrai **imparare ad interpretare** questi 3 tipi di informazione cercando di capire che tipo d'errore hai commesso; naturalmente ho già previsto sul mio **Tutorial Assembly** una serie di indicazioni utili, in questo senso, che ti consiglio di leggere **qui**.