

2 Manuali di Giobe2000

# TUTORIAL ASSEMBLER

## Nuovo Ambiente Assembler

**Come si utilizza**

**con l'editor **Qedit 3.00c** di SemWare Corp.**

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) ©

## Come si utilizza

con l'editor **Qedit 3.00c** di **SemWare Corp.**

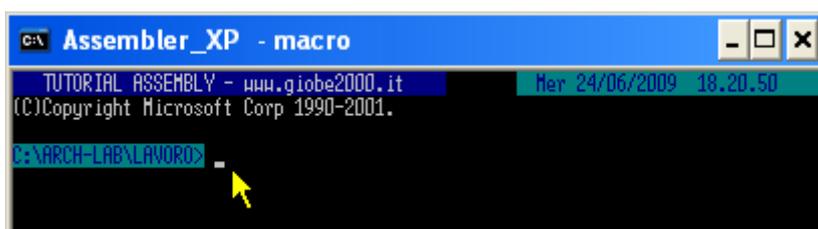
Per accedere alla creazione di un *progetto ASM* con il servizio di *editor* assicurato da **Qedit 3.00c** [Copyright 1994 **SemWare Corp.**] basta un *doppio-click* con il *tasto sinistro del mouse* su una delle 2 *iconcine*, scelte tra quelle disponibili nella cartella principale del **Nuovo Ambiente Assembler**, **C:\Arch-Lab\=%SystemDrive%\Arch-Lab\**:

The image shows a rectangular button with a light gray background. On the left is a small icon of a computer monitor with a grid pattern. To the right of the icon, the text "Assembler\_XP" is written in a monospaced font.The image shows a rectangular button with a light gray background. On the left is a small icon of a computer monitor with a grid pattern. To the right of the icon, the text "Assembler\_Vista&XP" is written in a monospaced font.

Si tratta del modo d'uso utilizzato finora dalla *vecchia versione* dell'**Ambiente**, basati sull'utilizzo del glorioso (ma a molti sgradito) *editor* **Qedit**; a differenza di quelli utilizzati dagli altri *sistemi d'accesso* (descritti nelle pagine simili a questa) esso è un *editor a caratteri*, cioè in *modo Testa*, tipico delle *schermate nere* del **DOS**, con 25 righe e 80 colonne, *coerente* con l'ambito in cui saranno ospitati i nostri progetti sempre di tipo DOS) ma *obiettivamente disagiata* durante lo sviluppo del progetto, in tutte le sue fasi (editing, compilazione e linking); tra l'altro, *non è gratuito* come tutti gli altri ma **shareware**.

A differenza da ciò che succede con gli altri metodi, questi 2 *non aprono un editor* ma si limitano ad *aprire una finestra DOS*, al prompt della quale si dovranno operare le scelte desiderate, comunque coordinate per assicurare un *ambiente integrato* per lo *sviluppo* di programmi.

L'accesso con la prima iconcina (**Assembler\_XP**) è quello originale, nato nei primi anni novanta e utilizza **Command.COM**, distribuito fino a *Windows XP* (e quindi non più utilizzabile con *Windows Vista*); la sua caratteristica è quella di coinvolgere il **driver ANSI**, in quel tempo particolarmente sfizioso e ricercato, in grado di *personalizzare il prompt del dos* (esso pure tale da infastidire l'arrogante *Windows Vista*), per cui dopo il *doppio-click* viene mostrata questa immagine **a schermo intero** (*monitor tutto nero* con intestazione, data e ora corrente e percorso della cartella corrente in evidenza).



Sebbene durante la progettazione dei programmi assembly sia conveniente lavorare in una shell Dos *a pieno schermo* puoi disporre in ogni momento della shell Dos *in una finestra*

semplicemente premendo **Alt** e **Invio**, e tornare poi di nuovo *a pieno schermo* premendo di nuovo **Alt** e **Invio**.

L'accesso con la seconda (**Assembler\_Vista&XP**) è quello creato per mantenere la portabilità del vecchio **Ambiente** anche sotto *Windows Vista*, praticamente e intrinsecamente *poco tollerante* nei confronti del DOS: per questo si è dovuto **rinunciare** al *driver ANSI* e ad altri *accessori di contorno* e alla possibilità di operare in una **shell a schermo intero** (esclusa a priori da *Vista*); in più è stato necessario sostituire il gestore di shell **Command.COM** con **CMD.EXE**, unico disponibile in questo sistema operativo (.. almeno nelle versioni a 32 bit); il **doppio-click** apre dunque questa immagine, priva della personalizzazione del prompt e **in finestra piccola**:



In ogni momento (o al termine della sessione di lavoro) è possibile chiudere definitivamente la shell DOS digitando **Exit** al prompt e confermando con **Invio**.



**Dunque:** il modo di procedere, che descriveremo tra poco, è **lo stesso** per entrambi i metodi, con l'unica differenza che, sotto *Windows Xp*, esso potrà essere attivato con entrambe le icone (sebbene sia preferibile la prima) e ospitato sia **a schermo pieno** che **in una piccola finestra** mentre sotto *Windows Vista* potrà essere attivato **solo** con la seconda icona e ospitato solo **in finestra**.

L'**Ambiente Assembler** è, in sostanza, una *raccolta di eseguibili*: in questo caso essi sono coordinati dal cosiddetto **Gestore d'Ambiente Macro.bat**, un potente *file batch*, cioè un semplice *file di testo* nel quale sono raccolti, in sequenza, diversi comandi da eseguire nell'ordine con cui sono elencati; come dice la parola stessa si tratta di un **gruppo (batch)** di cose da fare automaticamente, una dopo l'altra.

Un *file batch* deve avere estensione **BAT** (appunto da **batch**) ed è uno dei 3 riconosciuti *eseguibili* dal Sistema Operativo, insieme ai files **COM** e **EXE**; il tentativo di mettere in esecuzione un file con qualunque altra *estensione* produce una segnalazione d'errore: se digitiamo per esempio **"xxx"** il sistema operativo ci segnala **"xxx" non è riconosciuto come comando interno o esterno, un programma eseguibile o un file batch**", esattamente quello che abbiamo spiegato poco fa.



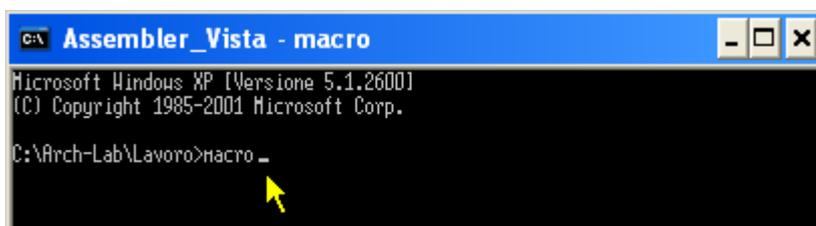
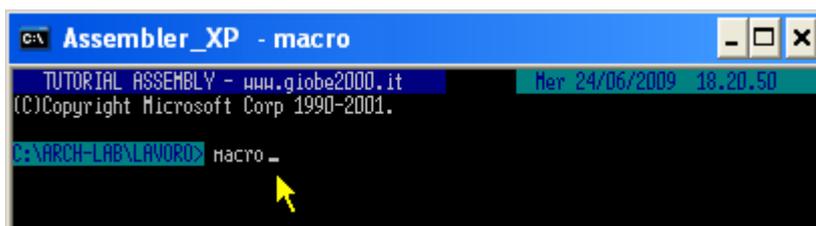
**NB:** se ti dovesse capitare di leggere un **messaggio** simile a quello appena descritto il tuo **Ambiente Assembler** non è stato installato correttamente: probabilmente hai **rinominato qualche cartella** o hai **spostato quelle ufficiali su un disco diverso da C:**; il contenuto del file **Macro.bat** contiene riferimenti precisi alle *sottocartelle* dell'ambiente **C:\Arch-Lab\=%SystemDrive%\Arch-Lab\**, per cui ogni variazione non prevista può **generare malfunzionamenti**.

Un *file batch* può utilizzare *anche* delle semplici *parole di programmazione* (come **IF** o **GOTO**) che consentono di dare struttura alle operazioni da fare; il nostro **file Macro** è un eccellente esempio per imparare a creare queste preziose raccolte; esso mantiene *nascosta* l'attivazione degli *applicativi* necessari alla *catena compilazione* (in stretta coerenza con quanto

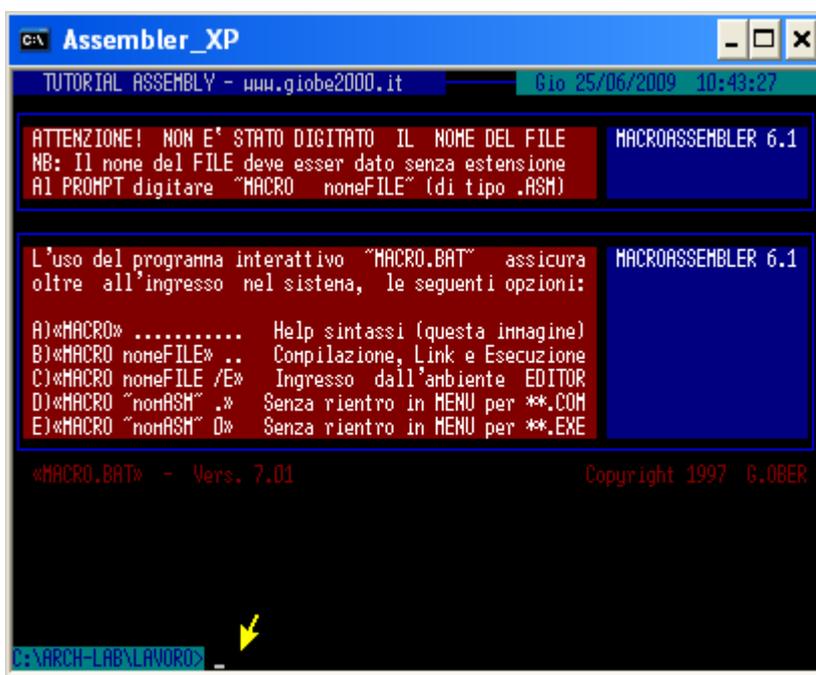
decritto nel **capitolo2** del **Tutorial Assembly**) organizzando e coordinando **automaticamente** ogni operazione necessaria allo **sviluppo del testo** del programma (**editor**), alla sua **compilazione (assembler)**, al suo **collegamento con i moduli delle librerie utilizzate (linker)** e al **riconoscimento e alla gestione degli eventuali errori**.

Pur nella sua semplice estetica esso garantisce a tutti gli effetti un efficiente **IDE (Integrated Development Environment)** cioè un **ambiente integrato** per lo **sviluppo** di programmi, appositamente configurato per il supporto alla **programmazione Assembly DOS a 16 bit** (da noi praticata in questo sito) ma perfettamente adatto ad interpretare anche il **set d'istruzioni** dei processori più moderni.

Per poterne disporre è dunque necessario e sufficiente digitare la parola **Macro** al prompt (cioè **la dove lampeggia il cursore** dopo la frase) confermandolo con **Invio**:



Non appena il comando **Macro** è confermato con **Invio** la **shell Dos** mostra il **menu di aiuto** con i suggerimenti utili al corretto utilizzo del gestore d'ambiente:



```

C:\> Assembler_Vista

ATTENZIONE! NON E' STATO DIGITATO IL NOME DEL FILE      MACROASSEMBLER 6.1
NB: Il nome del FILE deve esser dato senza estensione
Al PROMPT digitare "MACRO nomeFILE" (di tipo .ASM)

L'uso del programma interattivo "MACRO.BAT" assicura   MACROASSEMBLER 6.1
oltre all'ingresso nel sistema, le seguenti opzioni:

A)«MACRO» ..... Help sintassi (questa immagine)
B)«MACRO nomeFILE» .. Compilazione, Link e Esecuzione
C)«MACRO nomeFILE /E» Ingresso dall'ambiente EDITOR
D)«MACRO "nonASM" .*» Senza rientro in MENU per *.COM
E)«MACRO "nonASM" 0» Senza rientro in MENU per *.EXE

«MACRO.BAT» - Vers. 7.01                               Copyright 1997 G.OBER

C:\Arch-Lab\Lavoro> _

```

Si tratta di una tipica immagine di **Help**, proposta da ogni *applicativo ben progettato* in caso di insufficienti informazioni per il suo corretto funzionamento, proprio per suggerire le sue disponibilità, cioè l'elenco delle *opzioni* previste; le immagini utilizzate dal **Gestore d'Ambiente Macro.bat** (come quella di figura, presentate di volta in volta nelle varie fasi di progetto chiamate in causa) sono tutte generate dal file **MASM\_1.COM**, presente nella cartella **C:\Arch-Lab\bin=%SystemDrive%\Arch-Lab\bin**, una utility appositamente creata (in *Assembly*) per dare messaggi gradevoli e in grado di mostrare la potenza della programmazione con questo linguaggio.

In pratica con questa utility è possibile mettere a video il *giusto messaggio* al *momento giusto* semplicemente aggiungendo un **parametro numerico** subito dopo il suo nome; per esempio i testi della figura precedente sono ottenuti con i comandi **Masm\_1 20** e **Masm\_1 02**; la descrizione dell'effetto prodotto da ciascun possibile comando è leggibile [qui](#).

DOS

Prova a digitare il comando **Masm\_1 20** (scrivi la parola **Masm\_1** e il numero **20** con uno spazio in mezzo, confermando con **Invio**), oppure il comando **Masm\_1 02**, oppure uno degli altri possibili, sostituendo il numero con uno tra **01** e **21** (escluso **07**) ...

NB: con *Windows 2000* può essere necessario aggiungere un "\_", un underscore, al numero (es: digita il comando **Masm\_1 \_20**) [vedi anche in [questa pagina](#)]

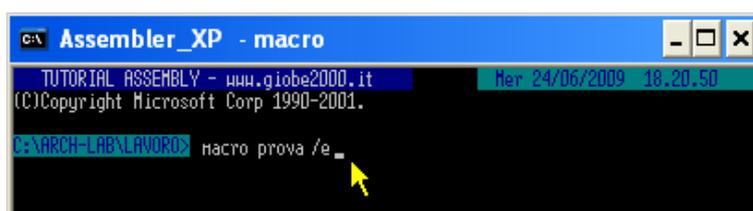
Le **opzioni** proposte dall'immagine di **HELP** suggeriscono i modi di uso del *file batch*; in particolare:

- digitando **Macro senza parametri (opzione A)**, esattamente come abbiamo fatto poco fa, viene mostrata di nuovo l'immagine di figura
- facendo seguire il **nome di un file sorgente (senza estensione e senza alcun parametro)** e confermando con **Invio (opzione B)** viene immediatamente attivata la *catena di montaggio* del file eseguibile, con rientro automatico nel **Menu Interattivo interno**. In questo caso si da per scontata la presenza (nella cartella di lavoro, **C:\Arch-Lab\lavoro=%SystemDrive%\Arch-Lab\lavoro**) di questo file *sorgente* con **estensione ASM**
- se si desidera ottenere lo stesso risultato (**compilare il programma sorgente**) *senza* rientrare nel Menu Interattivo basta *aggiungere un punto* dopo il nome del file sorgente (**opzione D**); in caso viene comunque garantita la supervisione degli eventuali errori di compilazione e/o di collegamento
- l'**opzione E** è del tutto simile alla precedente: la specifica del **parametro 0** (zero) evita di attivare il programma **Exe2Bin**, rendendo disponibile il **file EXE** invece del **COM**, anche quando la conversione sarebbe tecnicamente possibile
- l'**opzione C** è quella che si usa la prima volta, all'inizio di un nuovo progetto, studiata per aprire l'**editor Qedit** e consentire la scrittura di un nuovo *sorgente ASM* o la sua modifica (se già esiste)

**Dunque:** poichè tutto è già **predisposto e funzionante** puoi tentare immediatamente la tua prima compilazione; la prima cosa da fare è **scrivere un nuovo programma** sorgente (tipo ASM) o **leggerne uno scritto in precedenza**; per entrambe queste operazioni è necessario entrare nell'**Ambiente** con lo switch **/E**, da digitare dopo il comando **Macro** e il nome del file sorgente desiderato.

Vediamo dunque cosa bisogna fare per la **creare l'eseguibile COM o EXE** di un *sorgente ASM* già disponibile nella cartella di lavoro, per esempio quello predisposto come file di riferimento di nome **PROVA.ASM**; per prima cosa bisogna provvedere alla sua **apertura**, per la qual cosa al prompt digiteremo (con i 2 spazi al posto giusto e comunque senza estensione):

```
C:\ARCH-LAB\LAVORO> macro prova /e
```



```
Assembler_XP - macro
TUTORIAL ASSEMBLY - www.giobe2000.it
(C)Copyright Microsoft Corp 1990-2001.
C:\ARCH-LAB\LAVORO> macro prova /e
```

```
C:\ Arch-Lab - macro
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Arch-Lab\Lavoro> macro prova /e _
```

**DOS**

Prova anche tu nella shell aperta. **NB:** se il click sull'icona non funziona scrivi **file:///c:/arch-lab/Assembler\_Vista&XP.PIF** direttamente nel campo indirizzo del Browser.

Confermando con **Invio**, se si tratta del **primo ingresso** della giornata, *prima di qualunque altra operazione*, viene mostrato (una tantum) il seguente riquadro che ci ricorda l'eventuale possibilità di assumere una delle 3 **matrici** da me predisposte per facilitare la creazione del sorgente:

```
MACROASSEMBLER 6.1

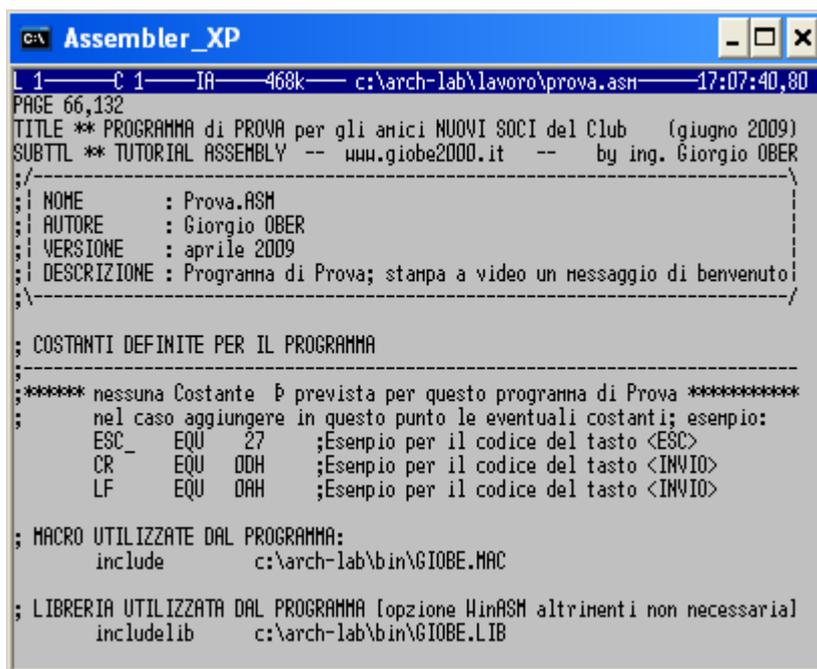
E' possibile predisporre l'assunzione automatica di
una matrice di supporto adatta alla stesura di un
nuovo programma premendo uno dei seguenti tasti nel
ambiente dell'elaboratore di testi aperto tra poco:

SHIFT-F3 per preparare sorgenti INTRAsegmentali .COM
SHIFT-F4 per preparare sorgenti EXTRAsegmentali .EXE
SHIFT-F5 per programmi da lasciare residenti (TSR)

Premere un tasto qualunque per entrare in EDITOR
```

Prendiamo atto di questo suggerimento e annotiamo mentalmente la sequenza **Shft-F3**: servirà quando saremo costretti a **scrivere un programma da zero**, non nel caso di questa presentazione iniziale, dato che per ora **tutto è già predisposto**. Alla pressione di un tasto qualunque verrà automaticamente messo in esecuzione l'**editor Qedit**, che provvederà a cercare nella cartella di lavoro, **C:\Arch-Lab\lavoro=%SystemDrive%\Arch-Lab\lavoro**, il file **prova.ASM**; l'estensione ASM è data per scontata ed assunta come default, per cui **non va scritta!!**

Naturalmente un file con questo nome è già stato predisposto in questa cartella, per cui l'**editor** trova qualcosa da aprire... e lo metterà a nostra disposizione per **leggerlo** ed eventualmente **modificarlo**: sul monitor apparirà l'immagine:



```

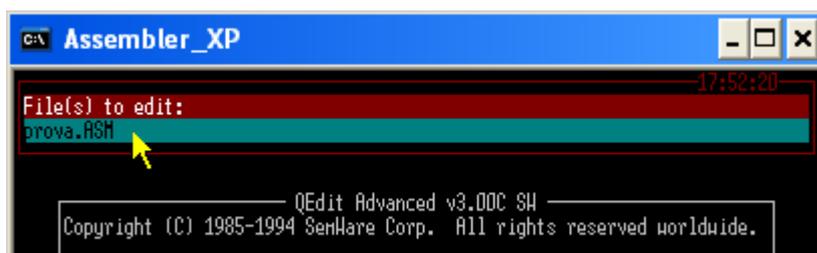
C:\ Assembler_XP
L 1 C 1 IA 468k c:\arch-lab\lavoro\prova.asm 17:07:40,80
PAGE 66,132
TITLE ** PROGRAMMA di PROVA per gli amici NUOVI SOCI del Club (giugno 2009)
SUBTTL ** TUTORIAL ASSEMBLY -- www.giobe2000.it -- by ing. Giorgio OBER
;-----
; NOME      : Prova.ASM
; AUTORE    : Giorgio OBER
; VERSIONE  : aprile 2009
; DESCRIZIONE : Programma di Prova; stampa a video un messaggio di benvenuto
;-----
; COSTANTI DEFINITE PER IL PROGRAMMA
;-----
;***** nessuna Costante   prevista per questo programma di Prova *****
; nel caso aggiungere in questo punto le eventuali costanti; esempio:
ESC_ EQU 27 ;Esempio per il codice del tasto <ESC>
CR EQU 0DH ;Esempio per il codice del tasto <INVIO>
LF EQU 0AH ;Esempio per il codice del tasto <INVIO>
;
; MACRO UTILIZZATE DAL PROGRAMMA:
include c:\arch-lab\bin\GIOBE.MAC
;
; LIBRERIA UTILIZZATA DAL PROGRAMMA [opzione MinASM altrimenti non necessaria]
includelib c:\arch-lab\bin\GIOBE.LIB

```

Il *piano d'editazione* è pronto con un *sorgente ASM* affidabile (di nome **PROVA.ASM**); guardando l'immagine già puoi capire come verrà trattato il testo presente nel documento; con la *freccia verso l'alto* o con la *freccia verso il basso* 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**.

La spartana gestione in modo testo puro non consente la possibilità di evidenziare **a colori** ogni *dettaglio sintattico e funzionale*, disponibile con tutti gli altri metodi d'accesso all'**Ambiente**.

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 è necessario, anzitutto, **chiudere** il file attualmente ospitato dall'**editor**. Per conservare la versione originale **senza eventuali modifiche** dobbiamo dare il comando **Ctrl-K Q, ^kQ** (tenendo premuto **Ctrl** digita in sequenza prima **K** e poi **Q**); se non hai toccato altri tasti se non quelli delle *freccie*...) il testo non ha subito variazioni per cui il *file sorgente* sarà immediatamente chiuso e ti verrà proposta la seguente immagine (con la quale l'editor ti suggerirà di riaprire il file...):

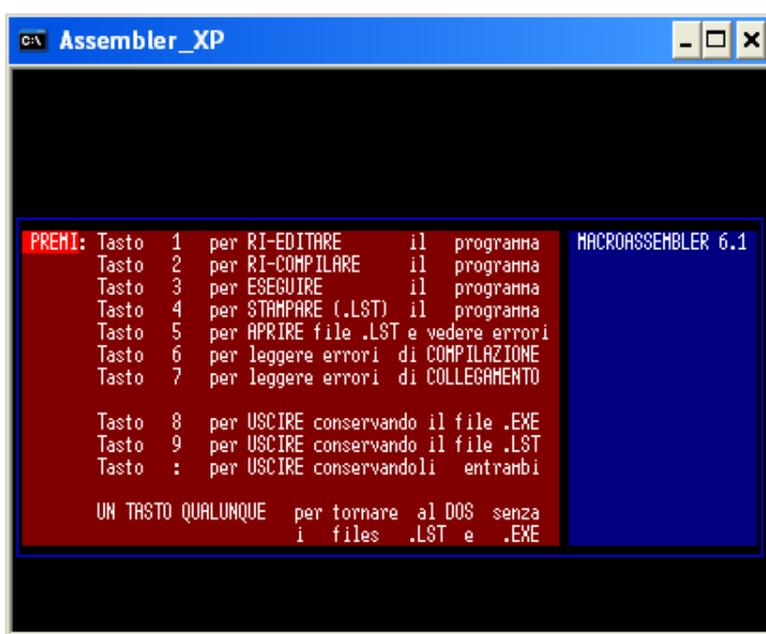


Se invece il testo avesse subito casuali modifiche, per non rischiare di perderlo nella sua stesura originale alla domanda *Lose changes?* bisogna rispondere **Yes**:



Basta premere **invio** (o cliccare su **Yes**) e il *file sorgente* sarà immediatamente chiuso senza conservare le eventuali indesiderate modifiche, con la stessa proposta di prima di **riaprire il file**; naturalmente in questa fase la cosa non ci interessa per cui ignoreremo il messaggio, premendo **ESC**.

In risposta si aprirà il seguente **Menu interattivo interno** (generato, come gli altri messaggi, da **Masm\_1.com** con parametro **09**, **Masm\_1 09**):



Di norma al termine della **scrittura** di un nostro *programma sorgente* desideriamo infatti vedere se funziona; a questo (ed a altro) provvede il **gestore interattivo** che propone (sotto forma di **Menu**) numerose alternative; ciascuna di esse è qui descritta sinteticamente, lasciando gli approfondimenti alle considerazioni successive a questo elenco:

- di solito, dopo la scrittura del programma sorgente, si desidera **creare l'eseguibile** per

cui bisogna senz'altro premere "2": i programmi che realizzano questo risultato (MASM, LINK PROVA, EXE2BIN, ecc) sono attivati in sequenza e automaticamente, senza alcun ulteriore intervento

- con "1" **si rientra nell'editor**; di solito di fa per *correggere* eventuali errori di compilazione o di collegamento, oppure semplicemente per *aggiungere nuove istruzioni* al programma sorgente
- con "3" **si mette in esecuzione** il programma dentro il gestore interattivo; può essere utile per *vedere l'effetto* del programma, sebbene di norma, il collaudo finale si faccia al di fuori
- con "4" è possibile **aprire il file listato**; l'opzione del menu è stata pensata per consentire la stampa di questo documento: poiché il file *prova.LST* è generato su 130 colonne, si dovrebbe provvedere a settare la stampante per un formato compatibile con il testo da stampare (per esempio in *condensato*). Con stampanti moderne conviene importare il file *prova.LST* direttamente in *Word* (o in altro wordprocessor) (vedi opzione "9") ricordando di usare eventualmente il font **GIOBE.TTF**, per rendere visibili le eventuali cornici (consulta le problematiche relative nella *scheda*)
- con "5" è possibile **aprire il file listato** per facilitare la correzione degli errori: ciascuno può essere letto sul monitor direttamente in prossimità della linea di programma che lo ha generato
- con "6" è possibile **leggere** gli eventuali **errori rilevati dall'assemblatore** in fase di compilazione: apre un file di testo appositamente creato dal gestore interattivo, **ErrCmp.txt**
- con "7" è possibile **leggere** gli eventuali **errori rilevati dal linker** in fase di collegamento: apre un file di testo appositamente creato dal gestore interattivo, **ErrLnk.txt**
- con "8" si torna al prompt del Dos **senza cancellare il file rilocabile EXE**; dato che, quando è possibile, esso è automaticamente convertito in COM da **exe2bin**, la sua presenza nella cartella è inutile e controproducente: per questo nella norma *viene cancellato* prima di rientrare nel menu interattivo (o prima di uscire)
- con "9" si torna al prompt del Dos **senza cancellare il file listato**; di solito questo file non è necessario e la sua presenza nella cartella è inutile e controproducente: per questo nella norma *non viene nemmeno creato*. Per altro in alcuni casi (verifica di errori particolarmente fastidiosi o necessità di esportare *prova.LST* su stampanti moderne) è opportuno prevedere questa disponibilità
- con "!" si torna al prompt del Dos **con entrambi i files** discussi nei 2 punti precedenti (file listato, *prova.LST*, file rilocabile, *prova.EXE*)

Se non si desidera nessuna delle precedenti opzioni (cioè se si vuole terminare la sessione di lavoro) basta *premere un tasto qualunque*; come già anticipato uno dei pregi del gestore interattivo **Macro.bat** è quello di *lasciare pulita* la cartella di lavoro: quando si torna definitivamente al Dos in cartella **ci saranno solo 2 files**, il **sorgente ASM** e l'**eseguibile COM** (o **EXE** se la conversione non ha avuto buon esito), evitando la presenza di tutti gli inutili files intermedi (come **OBJ**, **LST**, **BAK** e, naturalmente quelli di errore, **TXT**).



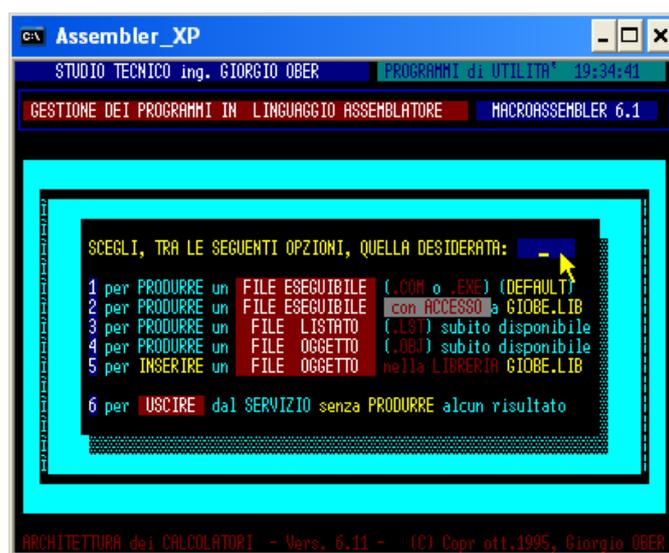
Avrai notato che ogni compito atto a **creare l'eseguibile COM o EXE** è svolto dal *Gestore interattivo Macro,bat*; se esso non fosse disponibile spetterebbe a noi organizzare tutte le operazioni necessarie: dovremmo **compilare** (ML /I. /Zm /c PROVA.asm | ERR-COMP;), **linkare** (LINK PROVA; /NOE | ERR-LINK), provare a **convertire** l'eseguibile in COM (EXE2BIN PROVA) e, infine, decidere di **rientrare nell'editor** (Q PROVA) o **chiudere la sessione** di lavoro.

Dalle descrizioni precedenti avrai capito che la buona conoscenza dell'**editor Qedit** ti aiuterà moltissimo nella scrittura dei sorgenti.

Prima di cominciare ti consiglio di divertirti un po' scrivendo con esso cose casuali: quando sarai pratico capirai la differenza tra un editor qualunque e il magico Qedit.

Ti ricordo i 2 *entry-point* disponibili sul sito: **per in imparare ad usarlo e per cambiare il suo aspetto** (se non ti piace così come l'ho pensato io...).

Disponendo, nella cartella di lavoro, del **sorgente PROVA.ASM**, appena salvato o nella sua forma originale, approfondiamo ora le operazioni necessarie per l'effettiva **creazione dell'eseguibile COM o EXE**: **in condizioni normali** l'*opzione* da scegliere è **sempre** la "2" (Tasto 2 per RI-COMPILARE il programma); come già anticipato il **gestore interattivo** organizza automaticamente la **compilazione** e il **collegamento** dei programmi a lui sottoposti; non appena è premuto il tasto "2" si presenta l'immagine seguente (generata dal programma **Masm\_in1.COM** (da me appositamente creato per offrire le possibili opzioni di compilazione) esso pure presente nella cartella **C:\Arch-Lab\bin=%SystemDrive%\Arch-Lab\bin**:



Di solito si sceglie dunque la *sottoOpzione* "2" ( **PRODURRE un FILE ESEGUIBILE con ACCESSO a GIOBE.LIB** ) che provvede a generare un **COM** (o un **EXE**, se la conversione non è possibile) **linkando ad esso** le **procedure** raccolte nella libreria **GIOBE.LIB** (**NB**: questo è il **nome di default della libreria** coinvolta). Se tutto va bene vengono proposta la schermata con i messaggi progressivi:

```

c:\ Assembler_XP
CREAZIONE di FILE ESEGUIBILE con accesso alla LIBRERIA MACROASSEMBLER 6.1
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997. All rights reserved.

Assembling: 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

EHE2BIN R1.01 Convert EXE files to BIN or COM files.
Copyright (C) 1989,1997 Caldera, Inc. All rights reserved.

Source file      : prova.exe
Destination file : prova.com

PREMI UN TASTO QUALUNQUE per ritornare al MENU MACROASSEMBLER 6.1

```

Premendo *un tasto qualunque* si rientra nel **Menu interattivo**; da notare che i 2 messaggi in testa e in coda sono sempre generati da **Masm\_1.com** rispettivamente con parametro **13**, **Masm\_1 13**, e con parametro **10**, **Masm\_1 10**).

DOS

**Riassumendo:** per **creare un eseguibile** basta attivare il **gestore interattivo** (con **macro nomefile + invio**) e scegliere "2" tra le sue *opzioni* e poi ancora "2" dalla tabella delle *sottoopzioni* che ti verrà proposta: partirà, in sequenza, la serie delle stringhe bianche presenti nell'ultima immagine e, al termine, la pressione di un tasto qualunque ti riporterà al **Menu interattivo**. Puoi provare nella finestra DOS la sequenza di comandi descritta in questa pagina. **NB**: se il click sull'icona non funziona scrivi **file:///c:/arch-lab/Assembler\_Vista&XP.PIF** direttamente nel campo indirizzo del Browser.

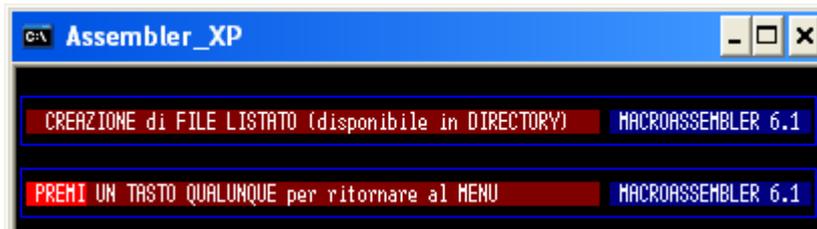
Se siamo sicuri di non dover assumere *oggetti esterni* (cioè se il nostro programma **non utilizza** le **procedure** della libreria **GIOBE.LIB**) possiamo scegliere la *sottoOpzione* "1" ( **PRODURRE un FILE ESEGUIBILE (.COM o .EXE) (DEFAULT)** ), che provvede a generare l'eseguibile **senza coinvolgere alcuna libreria**; i messaggi proposti sono esattamente quelli della figura precedente con eccezione del primo (in alto), riportato qui sotto (generato con **Masm\_1 12**):

```

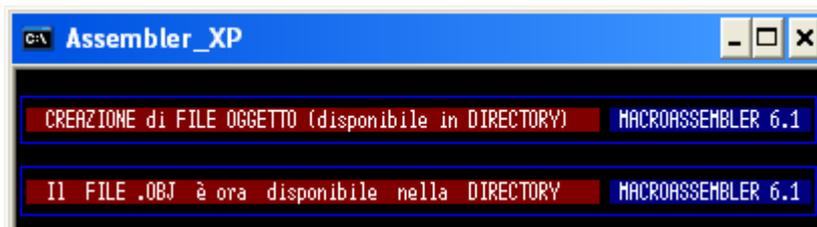
c:\ Assembler_XP
CREAZIONE di un FILE ESEGUIBILE senza USO di LIBRERIA MACROASSEMBLER 6.1

```

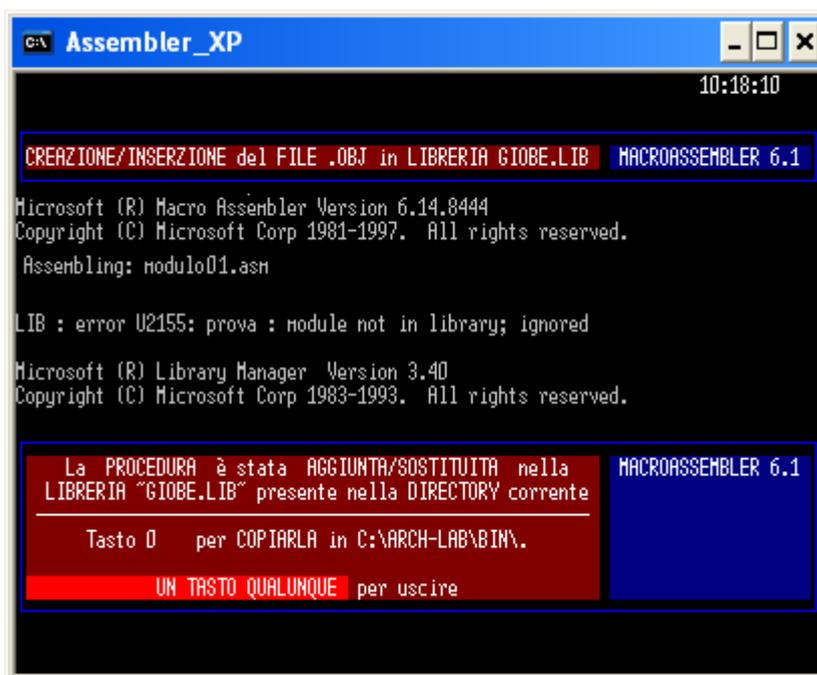
La *sottoOpzione* "3" ( **PRODURRE un FILE LISTATO (.LST) subito disponibile** ) attiva **solo l'assemblatore** MASM, creando e lasciando nella cartella di lavoro il file **LST**; anche in questo caso il primo messaggio è personalizzato (**Masm\_1 14**) e, data la particolarità del servizio, al termine viene proposto il rientro nel **menu interattivo**:



Anche la *sottoOpzione* "4" ( **PRODURRE un FILE OGGETTO (.OBJ) subito disponibile** ) attiva **solo l'assemblatore** MASM, creando e lasciando nella cartella di lavoro il file **OBJ**; il primo messaggio è personalizzato (**Masm\_1 15**) e, al termine del servizio, viene imposto il **ritorno al Dos** con il messaggio **Masm\_1 16**:



La *sottoOpzione* "5" ( **INSERIRE un FILE OGGETTO .nella LIBRERIA GIOBE.LIB** ) è un servizio molto utile: permette di aggiungere nuove procedure alla libreria di default; se il programma sorgente **rispetta le regole di un oggetto da libreria** (descritte nel **paragrafo ad esse dedicato**) il **Gestore interattivo** organizza il suo caricamento immediato nella **GIOBE.LIB** (**NB**: questo è il nome di **default** della libreria coinvolta), attivando in sequenza il **compilatore** e il **gestore di librerie**. Se tutto va bene viene proposta la schermata con in testa il messaggio **Masm\_1 18** e in coda il messaggio **Masm\_1 19**:



Rimane da vedere che bisogna fare se desideri **creare un nuovo sorgente ASM**; la via più pragmatica è quella di digitarne il nome (senza estensione e con i 2 spazi al posto giusto) nel comando scritto al prompt; supponendo di voler creare il sorgente **nuovo.ASM** al prompt digiteremo:

**C:\ARCH-LAB\LAVORO> macro nuovo /e**

Poichè l'**editor Qedit** non potrà ovviamente trovare questo file nella cartella di lavoro, provvederà comunque ad **aprire una sessione di lettura/scrittura**, presentando una **pagina vuota**, in attesa della stesura del testo del nuovo programma, fissando automaticamente il nome del nuovo lavoro in **nuovo.ASM** (cioè aggiungendo d'ufficio l'estensione **.ASM** al nome prova dichiarato all'inizio con il comando **macro nuovo /e**); l'editor evidenzia questo fatto riportando il nome del file (completo del percorso) nella barra rossa in alto, dichiarando così di accingersi ad editarlo:

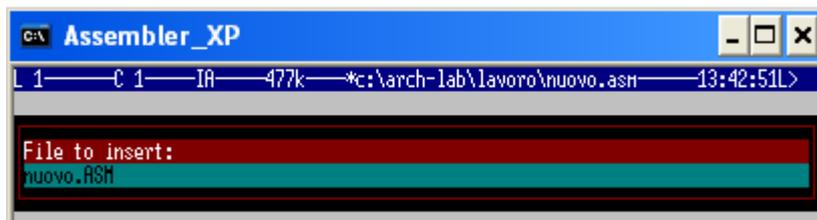
In questo caso, poichè il nostro programma sorgente non esiste ancora, non ci rimane che scriverlo, riga dopo riga, cercando di rispettare le regole e la sintassi richiesta dall'assemblatore: si tratta ovviamente di un compito piuttosto delicato, almeno la prima volta, perchè ogni piccola imperfezione produce segnalazione d'errore; e, almeno all'inizio, è complicato anche capire di che accidente d'errore si tratta...

In alternativa ti ho preparato 3 **Matrici**, pronte per essere riadattate ad ogni tua esigenza di progetto: ciascuna di esse contiene la struttura di un sorgente ASM adatta alla creazione dei 3 tipi principali di eseguibile: **COM**, **EXE** e **TSR** (per maggiori dettagli vedi il **capitolo2** del **Tutorial Assembly**); per assumer una **Matrice** basta premere una delle previste combinazioni di tasti, **Shft-F3**, per la **GiobeCOM.INT**, **Shft-F4**, per la **GiobeEXE.INT** e **Shft-F5**, per la **GiobeTSR.INT**; per non appesantire troppo l'inizio del tuo studio ti consiglio di provare con la matrice **GiobeCOM.INT**, pensata proprio per produrre la versione di eseguibile più **COMpatta** prevista dalla attuale configurazione.

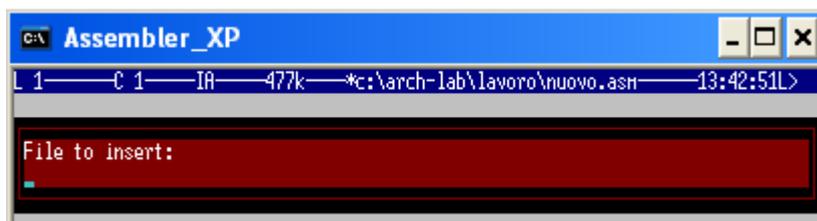
Premendo **Shft-F3** la matrice **GiobeCOM.INT** viene automaticamente inserita nel **piano di editazione**, che appare così (con sfondo celeste) in virtù del fatto che il suo testo risulta completamente marcato:



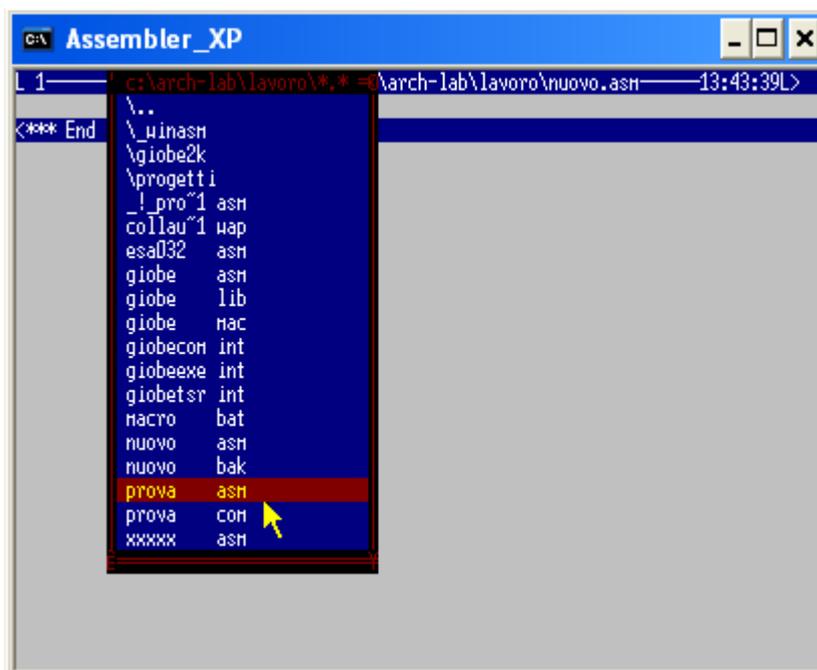
Naturalmente è saggio prendere in considerazione anche la possibilità di **far riferimento ad un lavoro precedente**, già personalizzato da commenti e conclusioni personali: basta caricarlo dentro il **foglio vuoto** con il **comando di lettura/assunzione file** dell'editor, cioè il comando **Ctrl-K R, ^kR** (tenendo premuto **Ctrl** digita in sequenza prima **K** e poi **R**), spiegato ampiamente in precedenza:



Non appena apparirà il riquadro (**File to insert:**) il nome presente (quello del file aperto) va sostituito con quello del file da assumere, purchè esso sia disponibile nell'Ambiente: confermando con **invio** il suo codice sorgente (..un testo) sarà caricato nella **pagina vuota**. In alternativa, toccando la **barra spaziatrice** il campo del nome si vuota ..



.. e, premendo invio, si apre la cartella corrente dalla quale selezionare il file desiderato, per esempio prova.asm:



Dopo le dovute modifiche e dopo le necessarie aggiunte, il nuovo programma sorgente è pronto per la compilazione, ma prima è necessario **uscire dall'editor** con il suo **salvataggio**

**nella memoria di massa**, nella cartella corrente (di lavoro); per uscire basta attivare il **comando di salvataggio**, **Ctrl-K X**, **^kX** (tenendo premuto **Ctrl** digita in sequenza prima **K** e poi **X**), anch'esso descritto **in precedenza**; non appena il file è stato trasferito sull'HD l'editor suggerirà di riaprirlo, presentando sul monitor il messaggio:

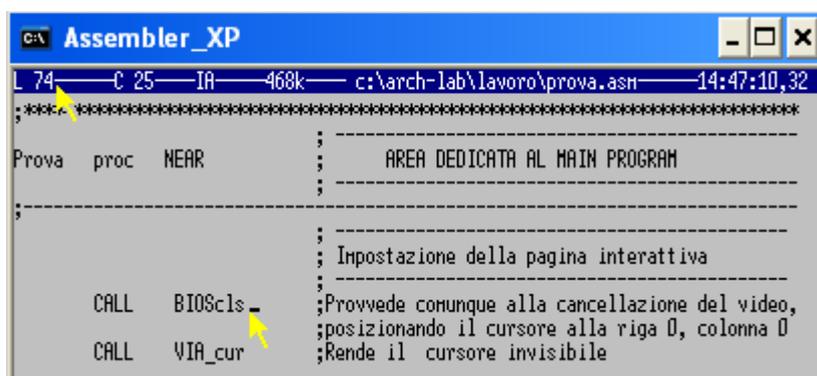


Poichè la cosa non ci interessa ignoreremo il messaggio, premendo **ESC**; in risposta verrà riaperto il **Menu Interattivo**.

Bene!! Non ti resta che **smanettare** 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 il **Gestore MACRO.BAT** *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 barra blu in alto a sinistra, nel **piano di editazione**) il sorgente prevede l'istruzione `CALL BIOScls`



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

```

C:\ Assembler_XP
L 74 C 25 IA 468k c:\arch-lab\lavoro\prova.asm 14:47:10,32
;*****
Prova proc NEAR ; AREA DEDICATA AL MAIN PROGRAM
;
;-----
; Impostazione della pagina interattiva
;-----
CALL BIOSclr ;Provvede comunque alla cancellazione del video,
CALL VIA_cur ;posizionando il cursore alla riga 0, colonna 0
;Rende il cursore invisibile

```

Osserva la *riga di controllo blu in alto*: in essa è indicata in ogni momento anche la riga e la colonna corrente del cursore, in questo caso la riga 74: annota mentalmente questa informazione, utilizzata dal compilatore per segnalare il punto in cui si è verificato l'errore.

Non appena si provvede alla fase creazione dell'eseguibile (*chiudendo* il sorgente e quindi scegliendo l'*opzione "2"* dal *menu interattivo* e poi la *sottoOpzione "2"* dal riquadro seguente) il gestore che lancia *automaticamente l'assembler* **si accorge subito** dell'imprecisione, segnalandola immediatamente con il messaggio **Masm\_1 03** (che può essere richiamato *anche* dal menu, in ogni momento, scegliendo l'opzione **"6"** (**Tasto 6 per leggere gli errori di COMPILAZIONE**)):

```

C:\ Assembler_XP
CREAZIONE di FILE ESEGUIBILE con accesso alla LIBRERIA MACROASSEMBLER 6.1
ERRORE NELLA CREAZIONE DEL FILE SORGENTE: CORREGGERE !! MACROASSEMBLER 6.1
Prebare: tasto 0 **** per ritornare subito al MENU ****
tasto 1 per elencare semplicemente gli errori
tasto 2 per aprire il file "ERRCMP.TXT"
Alla pressione di un tasto qualunque viene
creato ed aperto il file "nomefile.LST"
[ con il tasto CTRL-F1 è poi possibile isola-]
[ re le sole linee in cui contenuto un ERRORE]

```

Questo tipo d'errore è tipico della *fase di Compilazione (assembler)*; se in essa il **compilatore** rileva errori provvede ad elencarli a video uno dopo l'altro e, se essi fossero molto numerosi, non c'è verso di leggerli tutti perchè scorrono via velocemente: sul monitor restano visibili solo gli ultimi.

Per questa ragione *Gestore interattivo* provvede automaticamente al recupero di questa preziosa informazione creando un file di testo, **ErrCmp.txt**, che li raccoglie tutti e mostrando a video in che modo poterne fruire:

- tornare subito al menu (**tasto 0**) per rientrare nell'editor (e correggere gli errori)
- riprodurre di nuovo a video la sequenza degli errori (**tasto 1**, possibilità opportuna solo se gli errori sono pochi); in questo caso la schermata viene inizializzata dal messaggio **Masm\_1 11** ed è del tipo:

```

C:\ Assembler_XP
15:15:24
MESSAGGI IN FASE DI COMPILAZIONE (MASM) MACROASSEMBLER 6.1
Assembling: prova.asm
prova.asm(74) : error A2006: undefined symbol : BIOSclr
PREMI UN TASTO QUALUNQUE per ritornare al MENU MACROASSEMBLER 6.1

```

- aprire nell'editor ( **tasto 2** ) il file d'errore appena creato, **ErrCmp.txt**, come si vede in figura:

```

C:\ Assembler_XP
L 1 C 1 IA 479k c:\arch-lab\lavoro\errcmp.txt 15:15:50.32
Assembling: prova.asm
prova.asm(74) : error A2006: undefined symbol : BIOSclr
<*** End of File ***>

```

In queste segnalazioni c'è tutto quello che è necessario sapere: il messaggio restituito da MASM 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** ("error A2006: undefined symbol: BIOSclr", chiaramente un simbolo non riconosciuto, essendo BIOSclr diverso da BIOScls). Da notare che la versione precedente di MASM (la **5.10**) mostra messaggi più dettagliati, anche se quelli *di sostanza* non mancano:

```

C:\ Assembler_XP
MESSAGGI IN FASE DI COMPILAZIONE (MASM) MACROASSEMBLER 6.1
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.
prova.ASM(74) : error A2006: undefined symbol : BIOSclr
47476 + 383273 Bytes symbol space free
0 Warning Errors
1 Severe Errors
PREMI UN TASTO QUALUNQUE per ritornare al MENU MACROASSEMBLER 6.1

```

```

C:\ Assembler_XP
L 1 C 1 IA 479k c:\arch-lab\lavoro\errcmp.txt 15:15:50.32
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.
prova.asm(74) : error A2006: undefined symbol : BIOSclr
47496 + 386245 Bytes symbol space free
0 Warning Errors
1 Severe Errors
<*** End of File ***>

```



Se la **compilazione (assembler)** ha prodotto **errori** il modo migliore per correggerli è **assumere** il file **ErrCmp.TXT** direttamente dentro il file sorgente (con la tecnica **^kR**, tenendo premuto **Ctrl** digita in sequenza prima **K** e poi **R**); è così possibile consultare con comodo l'elenco degli errori, saltando alla linea indicata e cercando di eliminare la causa d'errore.

Certo, ora dovrai **imparare ad interpretare** questi 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](#).

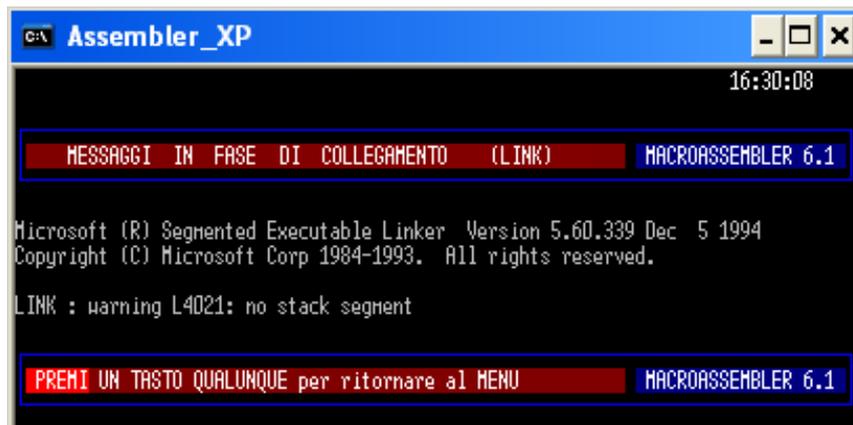
E' importante sapere che gli errori che si possono manifestare nella **creazione dell'eseguibile** possono essere dovuti anche alla **fase Collegamento (linking)** dell'oggetti (OBJ del sorgente e procedure esterne) coinvolti nella creazione stessa; anche in questo caso il gestore, lanciando **automaticamente** il **linker**, **si accorge subito** dell'imprecisione, segnalandola immediatamente con il messaggio **Masm\_1 04** (che può essere richiamato anche dal menu, in ogni momento, scegliendo l'opzione **"7"** (**Tasto 7 per leggere gli errori di COLLEGAMENTO**)):

```
C:\ Assembler_XP
ERRORE NEL COLLEGAMENTO DEL FILE: RIPETERE LE OPERAZIONI
Premere: tasto 0 *** per ritornare subito al MENU ***
tasto 1 per elencare semplicemente gli errori

Alla pressione di un tasto qualunque viene
aperto il file "ERRLNK.TXT" contenente tutti i
riferimenti non risolti, normalmente dovuti
mancata dichiarazione "PUBLIC" delle variabili
16:29:47 MACROASSEMBLER 6.1
```

Questo tipo d'errore viene rilevato dal **linker** che provvede ad elencarli a video uno dopo l'altro; anche per essi il **Gestore interattivo** crea un file di testo, **ErrLnk.txt**, che li raccoglie tutti e mostrando a video in che modo poterne fruire:

- tornare subito al menu (**tasto 0**) per rientrare nell'editor (e correggere gli errori)
- riprodurre di nuovo a video la sequenza degli errori (**tasto 1**, possibilità opportuna solo se gli errori sono pochi); in questo caso la schermata viene inizializzata dal messaggio **Masm\_1 21** ed è del tipo:



```

C:\> Assembler_XP
16:30:08
MESSAGGI IN FASE DI COLLEGAMENTO (LINK) MACROASSEMBLER 6.1

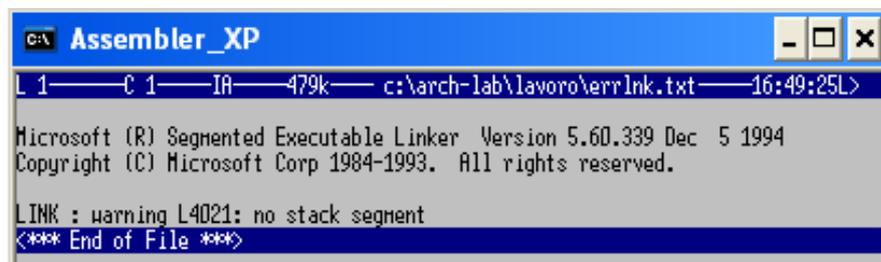
Microsoft (R) Segmented Executable Linker Version 5.60.339 Dec 5 1994
Copyright (C) Microsoft Corp 1984-1993. All rights reserved.

LINK : warning L4021: no stack segment

PREMI UN TASTO QUALUNQUE per ritornare al MENU MACROASSEMBLER 6.1

```

- aprire nell'editor ( **tasto 2** ) il file d'errore appena creato, **ErrLnk.txt**, come si vede in figura:



```

C:\> Assembler_XP
L 1-----C 1-----IA-----479k----- c:\arch-lab\lavoro\errlnk.txt-----16:49:25L>
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
<*** End of File ***>

```



Se il **collegamento (linker)** ha prodotto **errori** il modo migliore per correggerli è **assumere** il file **ErrLnk.TXT** direttamente dentro il file sorgente (con la tecnica **^kR**, tenendo premuto **Ctrl** digita in sequenza prima **K** e poi **R**); è così possibile consultare con comodo l'elenco degli errori, saltando alla linea indicata e cercando di eliminare la causa d'errore.

Questo metodo d'accesso all'**Ambiente Assembler** non prevede *esplicitamente* i **servizi accessori** integrati nei comandi degli **editor grafici** utilizzati dagli altri; ma ovviamente questo non significa che essi non possano essere disponibili.

La **fase di debug** è attivabile in ogni momento semplicemente digitando il comando **debug** al prompt e confermando con **invio**:

**C:\ARCH-LAB\LAVORO> debug**

```

C:\ Assembler_XP
TUTORIAL ASSEMBLY - www.giobe2000.it          Ven 26/06/2009 17.08.18
(C)Copyright Microsoft Corp 1990-2001.
C:\ARCH-LAB\LAVORO> debug
Microsoft (R) Symbolic Debug Utility Version 4.00
Copyright (C) Microsoft Corp 1984, 1985. All rights reserved.
Processor is [80286]
_

```

Ogni dettaglio per un suo corretto utilizzo lo puoi trovare **qui**; per esempio per analizzare l'eseguibile **prova.com** sarà necessario passargli il nome del file da aprire come parametro digitando come primi 2 comandi (al prompt di debug, cioè **subito dopo** la **lineetta lampeggiante**): **- n prova.com + <invio>** (per specificare il **nome del file da aprire** con debug, nel nostro caso **prova.com**) e poi **- l <invio>** (per dare a debug il comando di caricarlo, **load**, in memoria):

```

C:\ Assembler_XP
TUTORIAL ASSEMBLY - www.giobe2000.it          Ven 26/06/2009 17.08.18
(C)Copyright Microsoft Corp 1990-2001.
C:\ARCH-LAB\LAVORO> debug
Microsoft (R) Symbolic Debug Utility Version 4.00
Copyright (C) Microsoft Corp 1984, 1985. All rights reserved.

Processor is [80286]
- n prova.com
- l
_

```

Subito dopo potrai operare come al solito, per esempio per disassemblare (**unassembly**):

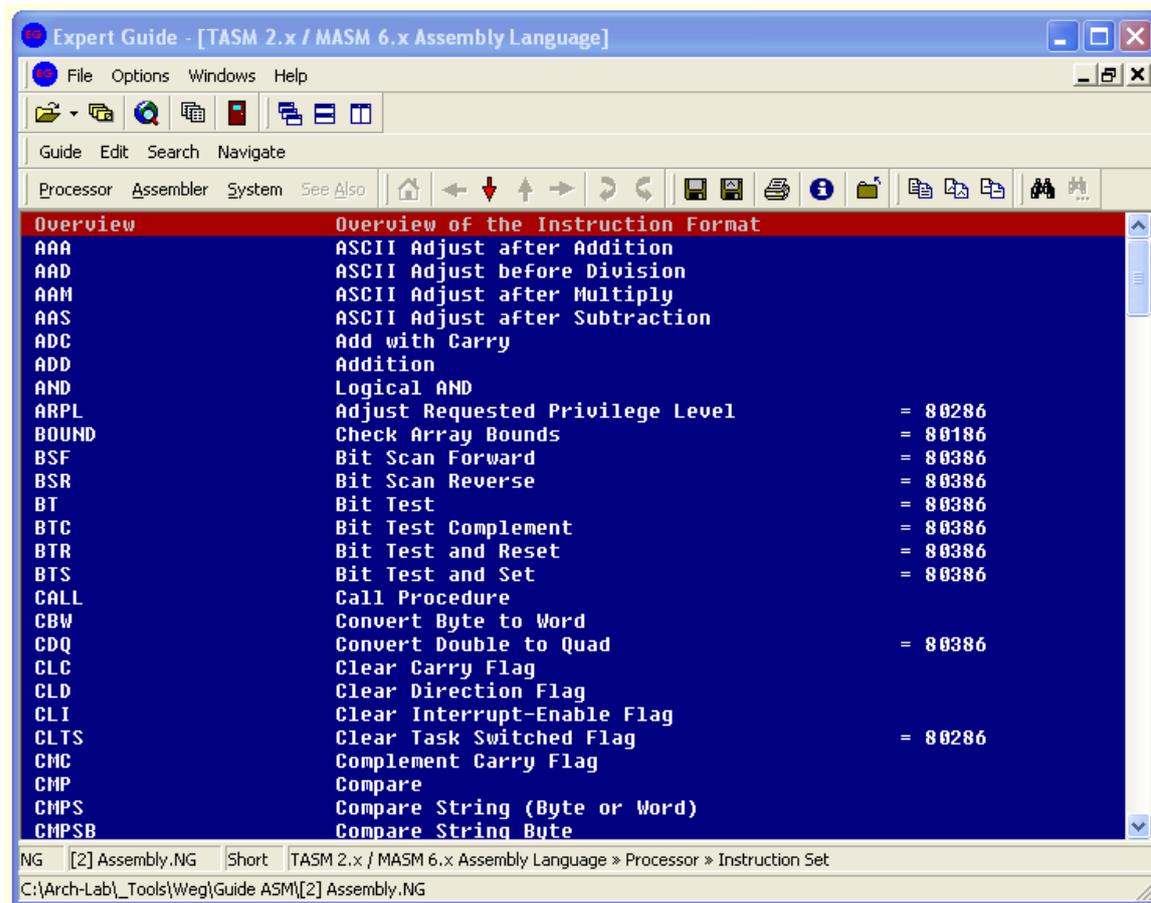
```

C:\ Assembler_XP
_ u
1F7E:0100 E92301      JMP     0226
1F7E:0103 000A      ADD    DL,BL
1F7E:0105 C4C4      LES   AX,SP
1F7E:0107 C4C4      LES   AX,SP
1F7E:0109 C4C4      LES   AX,SP
1F7E:010B C4C4      LES   AX,SP
1F7E:010D C4C4      LES   AX,SP
1F7E:010F C4C4      LES   AX,SP
_ u
1F7E:0111 C4C4      LES   AX,SP
1F7E:0113 C4C4      LES   AX,SP
1F7E:0115 C4C4      LES   AX,SP
1F7E:0117 C4C4      LES   AX,SP
1F7E:0119 C4C4      LES   AX,SP
1F7E:011B C4C4      LES   AX,SP
1F7E:011D C4C4      LES   AX,SP
1F7E:011F C4C4      LES   AX,SP
_

```

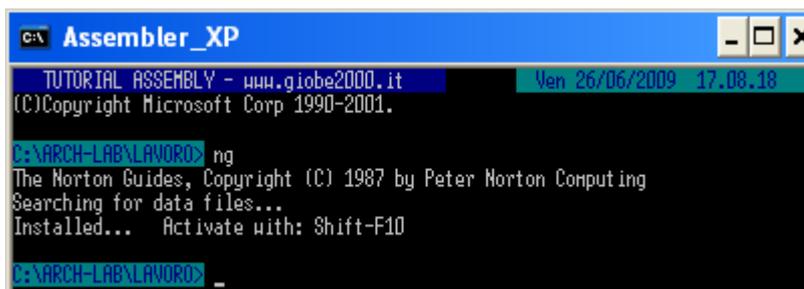
Poichè l'ambiente garantito da questo metodo d'accesso è intrinsecamente **"a carattere"** anche il prezioso programma **Weg\_2.15** [Copyright **Dave Pearson**, lo straordinario **gestore di**

**archivi di dati** (*Expert Guide Reader for Windows*) pensato per aiutarti a **scoprire molti gioielli** di questo nostro *tesoro*], non è controllabile direttamente, ma è sempre possibile attivare l'esecuzione per altra via *clickando* sul collegamento **WEG\_2.15** predisposto nella cartella **C:\Arch-Lab\\_Tools\=%SystemDrive%\Arch-Lab\\_Tools\**:



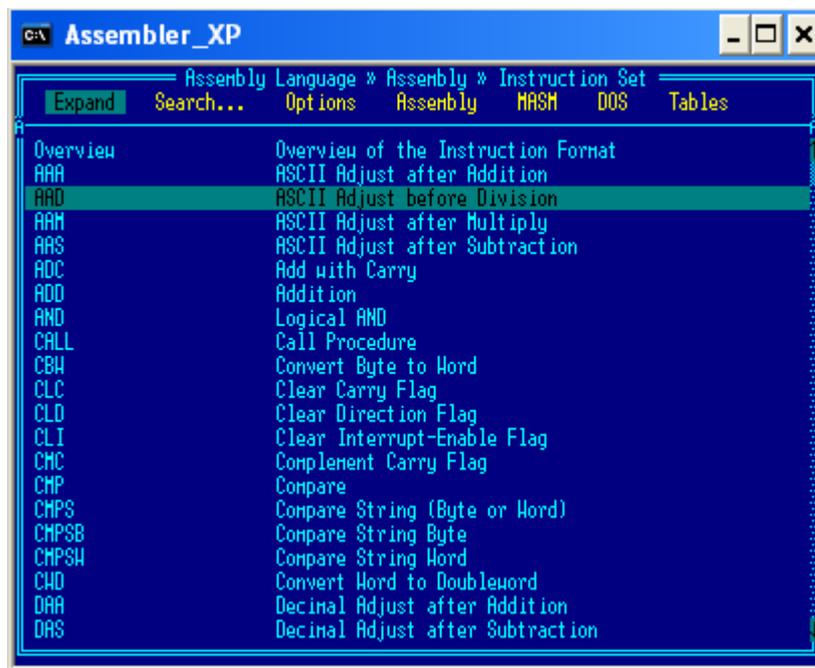
E' tuttavia previsto e configurato il gestore d'archivi TSR **The Norton Guide** [Copyright 1987 **Peter Norton Computing**, con un adeguato archivio di informazioni relative alla **programmazione Assembly DOS a 16 bit** (da noi praticata in questo sito in ogni dettaglio); per renderlo disponibile è necessario digitare il comando **ng** al prompt, confermandolo con **invio**:

**C:\ARCH-LAB\LAVORO> ng**



Come si vede viene subito suggerito il modo per utilizzarlo: in ogni momento esso offrirà i dati

dell'archivio corrente premendo i tasti **Shift + F10**, **anche mentre si sta editando il sorgente**, essendo un gestore che funziona in modo latente. La figura mostra (per esempio pratico) la cartella con le indicazioni relative alle istruzioni dell'80x86:



Ecco: per utilizzare al meglio il **metodo storico** questo è proprio tutto!