Essai TC 2.0

La note que voici est un aide mémoire pour retrouver comment configurer l'IDE afin de créer une petite application avec cette version du compilateur.
Elle montre aussi comment utiliser ce compilateur en mode ligne de commande.

Le Turbo C version 2 de Borland date de 1989. C'est dire s'il est ancien ! Il convenait pour l'écriture de programmes en langage C pour application tournant sous DOS.

Turbo C est un environnement de développement IDE qui permet l'apprentissage du langage C. Il peut être téléchargé gratuitement sur le site de Borland :

http://community.borland.com/article/0,1410,20841,00.html

Supposons que le TURBO C 2.01 est installé dans le répertoire C:\TC20.
Les projets sont placés dans un répertoire du disque C soit le répertoire C:\PRJ
Puisque nous jouons avec divers logiciels de développement (compilateur, linker, IDE etc.) il est nécessaire de ne pas tous les mélanger. Ainsi, pour travailler avec la version 2.01, il faut que lorsqu'on invoque les commandes TC ou TLINK se soient les fichiers TC.exe et TLINK.exe du répertoire \TC20 qui soient appelés.
On crée pour indiquer cela un fichier TC20.BAT qui adapte le PATH à ces exigences.
Voici ce que contient ce fichier de commande :

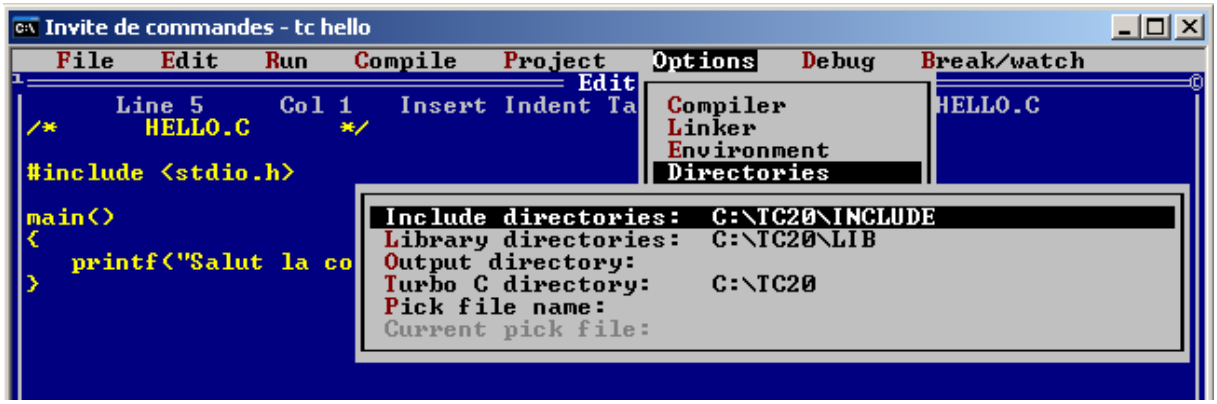
```
path %WINDIR%\System32;%WINDIR%;C:\Tc20
cd \Prj\tc20
```

NB. La variable d'environnement %WINDIR% prendra pour valeur C:\Windos ou C:\WINNT suivant la versions de Windows que vous utilisez (98, 2000 ou XP) Après exécution de TC20.bat nous sommes dans notre répertoire C:\PRJ\TC20 où nous plaçons les essais réalisés avec cette version du compilateur.

Soit à faire un programme Hello.exe qui affiche «Salut la compagnie »
Faites un fichier Hello.prj qui contient le nom du seul fichier source : Hello.c
Appeler TC HELLO ce qui ouvre HELLO.C où on écrit

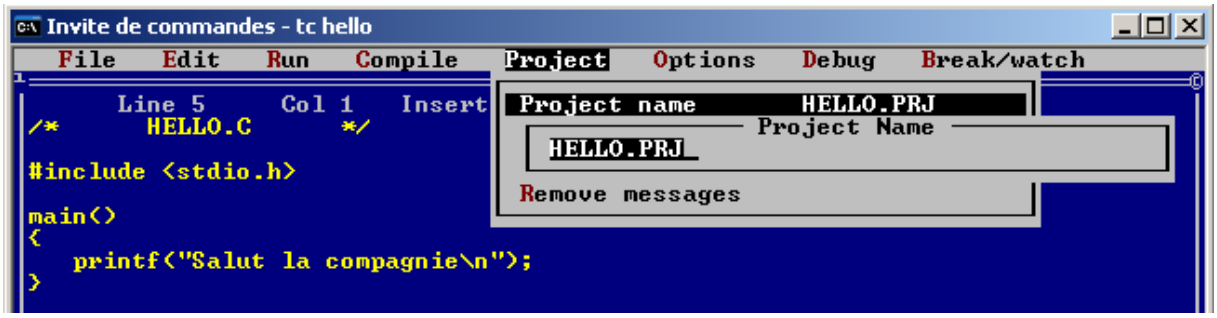
```
#include <stdio.h>
main()
{
    printf("Salut la compagnie\n");
}
```

Reste à adapter les options de l'IDE pour que lors de la compilation celui-ci puisse retrouver les répertoires à utiliser :

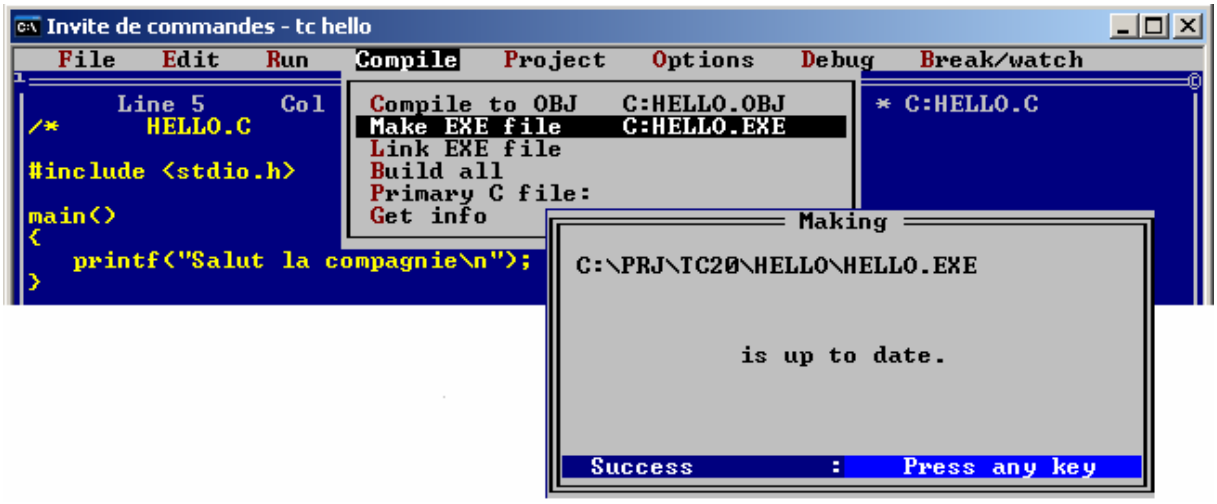


Il est encore nécessaire de sauver cette configuration pour qu'elle soit automatique la prochaine fois. => C:\TC20\TCCONFIG.TC

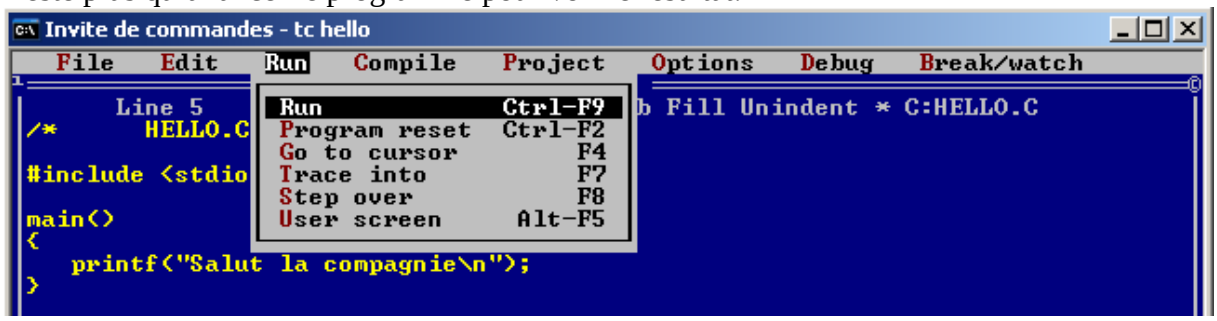
Avant de lancer la compilation il faut que soit défini un projet, ici HELLO.PRJ. Ce dernier fichier contient la liste des fichiers sources qui constituent l'application, un seul ici : HELLO.C



Le compilateur crée un fichier HELLO.OBJ dont le linker se sert pour produire l'exécutable HELLO.EXE



Reste plus qu'à lancer le programme pour voir le résultat.



Vous n'aurez pas le temps de voir ce qui se passe puisque après exécution on se retrouve automatiquement en mode édition du programme. Pressez les touches Alt-F5 pour voir l'écran utilisateur.

Observez après avoir refermé cette application que le répertoire courant contient maintenant non seulement le fichier Hello.c mais aussi Hello.obj et surtout Hello.exe qui est une application à part entière.

Compilation par ligne de commande

Syntaxe de la commande TCC

```
Turbo C  Version 2.01  Copyright (c) 1987, 1988 Borland International
Syntax is: TCC [ options ] file[s]       * = default; -x- = turn switch x off
  -1      80186/286 Instructions          -A      Disable non-ANSI extensions
  -B      Compile via assembly            -C      Allow nested comments
  -Dxxx   Define macro                    -Exxx   Alternate assembler name
  -G      Generate for speed              -Ixxx   Include files directory
  -K      Default char is unsigned        -Lxxx   Libraries directory
  -M      Generate link map               -N      Check stack overflow
  -O      Optimize jumps                  -S      Produce assembly output
  -Uxxx   Undefine macro                  -Z      Optimize register usage
  -a      Generate word alignment         -c      Compile only
  -d      Merge duplicate strings         -exxx   Executable file name
  -f    * Floating point emulator         -f87    8087 floating point
  -gN     Stop after N warnings           -iN     Maximum identifier length N
  -jN     Stop after N errors             -k      Standard stack frame
  -lx     Pass option x to linker         -mc     Compact Model
  -mh     Huge Model                      -ml     Large Model
  -mm     Medium Model                    -ms   * Small Model
  -mt     Tiny Model                      -nxxx   Output file directory
  -oxxx   Object file name                -p      Pascal calls
  -r    * Register variables              -u    * Underscores on externs
  -v      Source level debugging          -w      Enable all warnings
  -wxxx   Enable warning xxx              -w-xxx  Disable warning xxx
  -y      Produce line number info        -zxxx   Set segment names
```

Fichier de configuration pour TCC

Pour ne pas devoir taper toutes les options, il est pratique d'avoir dans le répertoire courant un fichier **`TurboC.CFG`** qui contient par exemple :

```
-IC:\TC\INCLUDE
-LC:\TC\LIB
```

Testons la compilation en ligne de commande en reprenant le même exemple que ci-dessus.

Faire un exécutable

Le seul fait de taper la commande TCC HELLO suffit pour produire un exécutable :

```
C:\Prj\tc20\Hello>tcc hello
Turbo C  Version 2.01  Copyright (c) 1987, 1988 Borland International
hello.c:
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International
        Available memory 390326

C:\Prj\tc20\Hello>dir
…
06/04/2005  22:38               104 HELLO.C
06/04/2005  23:45             5.676 HELLO.EXE
06/04/2005  23:45               326 HELLO.OBJ
06/04/2005  23:04                50 turboc.cfg
               4 fichier(s)            6.156 octets
```

Compiler uniquement

L'option –c permet de lancer uniquement la compilation sans faire l'édition de liens.

```
C:\Prj\tc20\Hello>tcc –c hello.c
```

Il faudra ensuite appeler l'éditeur de lien TLink pour obtenir l'exécutable.

Convertir un fichier .C en fichier assembleur

Compiler un fichier .C pour produire un code Assembleur :

```
C:\Prj\tc20\Hello>tcc –S hello.c
```

Cela donne un fichier HELLO.ASM lesté de quelques informations de debug.

TLink

Voici les aides affichées lorsqu'on invoque la commande TLINK : (Versions 2 et 7)

```
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International
Syntax:  TLINK objfiles, exefile, mapfile, libfiles
@xxxx indicates use response file xxxx
Options: /m = map file with publics
         /x = no map file at all
         /i = initialize all segments
         /l = include source line numbers
         /s = detailed map of segments
         /n = no default libraries
         /d = warn if duplicate symbols in libraries
         /c = lower case significant in symbols
         /3 = enable 32-bit processing
         /v = include full symbolic debug information
         /e = ignore Extended Dictionary
         /t = create COM file
```

```
Turbo Link  Version 7.00 Copyright (c) 1987, 1994 Borland International
Syntax: TLINK objfiles, exefile, mapfile, libfiles, deffile, resfiles
@xxxx indicates use response file xxxx
/x   No map                          /c  Case sensitive symbols
/m   Map including public names      /C  Case sensitive exports & imports
/M   Map with mangled public names   /ye Expanded memory swapping
/s   Map plus detailed segment map   /yx Extended memory swapping
/l   Map plus source line #s         /d  Warn if duplicate symbols in libraries
/i   Initialize all segments         /f  Inhibit optimizing far calls to near
/L   Specify library search paths    /Gx Goodies
/n   Ignore default libraries              n=discard Nonresident name table
/v   Full symbolic debug information       r=transfer Resident names to
/Tti Specify target & image type             nonresident names table
     t can be d = DOS (default)      /A=dd  Set segment alignment
              w = Windows            /R[mpekv]  Specify option to RLINK
              x = DPMI               /t  Create COM file (same as /Tdc)
     i can be e=EXE or d=DLL         /k  Suppress "No stack" warning msg
/3  Enable 32-bit processing         /Ox Optimizations
/o  Overlay switch                         c=chained fixups
/P[=dd]  Pack code segments                i=iterated data
                                           a=minimum segment alignment
                                           r=minimum resource alignment
```

La commande TLink est appelée automatiquement par TCC, sauf si cette dernière est invoquée avec l'option –c. Les différents modules du programme sont souvent compilés séparément et ce n'est que quand tous les modules objets sont corrects qu'on lance l'édition de lien. La commande TLink risque d'être fort longue à taper car elle doit contenir la liste des fichiers objets, ainsi que celle des librairies. On a donc prévu pour ce genre de commande des fichiers de réponse dont le rôle est d'enregistre une fois pour toute la liste des argument à passer à la commande TLINK. Ainsi pour notre exemple HELLO.C on prévoit un fichier de réponse qui contient ceci :

```
Hello.obj C:\TC20\LIB\c0s.obj, Hello.exe, Hello.map, C:\CPublic\TC20\LIB\CS.lib
```

NB. Le fichier de réponse contient quatre zones : la liste des objets, le nom de l'exécutable, le nom du fichier .map et la listes des librairies à inclure. Des espaces sont utilisés comme séparateurs dans les liste, les séparateurs de zones sont des virgules ou des retors à la ligne. Ainsi le même fichier de commande aurait pu s'écire sur quatre lignes, les retours à la lignes font office de séparateurs au lieu des virgules.

```
Hello.obj C:\TC20\LIB\c0s.obj
Hello.exe
Hello.map
C:\TC20\LIB\CS.lib
```

La commande TLink est alors passée comme suit :

```
TLink /i /v /l @Hello.rsp
```

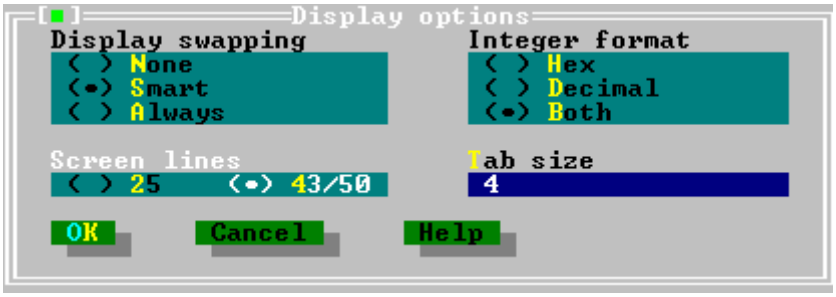
Le TLink Version 2 fonctionne de manière suffisante pour construire un programme avec l'IDE TC.
Nous lui préférons une version moins ancienne : la version 7 de TLink produit un code qui convient aussi au debugger TD. Ce debugger nous sera utile pour tester le programme au niveau de l'assembleur.

Turbo debugger

Il faut, pour pouvoir tester ce debugger, avoir une version pas trop ancienne de l'éditeur de liens, nous utilisons la version 7 de TLink.exe . Le debugger proprement dit est le fichier TD.exe mais il doit être accompagné d'un autre RTM.exe .

TD est une application très puissante. Des couleurs habilement choisies affichent des fenêtres, des menus et des boîtes de dialogue tout aussi clairement que si nous étions en mode graphique sous Windows.

Au premier lancement du programme celui-ci s'ouvre dans une fenêtre relativement petite. La première chose à faire est de demander un affichage plus large : **Options > Display options… > Screen lines : 43**



Demandez ensuite une sauvegarde de ces options : **Option > Save options… > OK**

Le Turbo Debugger ne fonctionne de manière vraiment efficace que s'il dispose de la table des symboles. Sans cela il peut juste afficher un désassemblage de l'exécutable et il est alors fort difficile de deviner ce que représentent ces instructions qui se réfèrent uniquement à des adresses plutôt qu'aux noms symboliques des fonctions et des variables.