Introducció.

L'objectiu d'aquesta pràctica és iniciar-nos en la manipulació de les eines de desenvolupament d'aplicacions per a computadors del tipus "PC". Cal notar que, sobre aquests sistemes, hi ha un sistema operatiu (tipus MS-DOS, Windows 98). En les nostres pràctiques, gairebé sempre, prescindirem dels serveis oferts pel sistema operatiu.

La eina que tenim al laboratori és el 'paquet' de Borland, concretament el 'Borland C' versió '3.1'. Aquest paquet comprèn diferents 'utilitats' (un compilador de C/C++, Assemblador, Depurador...) dins un 'entorn integrat' (IDE).

En aquesta pràctica 'executarem' les aplicacions des de la línia de comandes del MS-DOS (o bé des de 'fitxers de procés per lots' .BAT) prescindint de les 'automatitzacions' que aporten els 'entorns visuals' i que, sovint, oculten el veritable sentit del que s'està fent a l'usuari. Conscients que 'desaprofitem' unes facilitats orientades a augmentar el 'rendiment productiu' en el desenvolupament d'aplicacions, procedirem igual que fèiem amb el microcontrolador (editar el codi font, assemblar, enllaçar i depurar o simular).

> ***Assemblador Turbo Assembler***: Aplicació que ens permet codificar (traduir) instruccions escrites en format text (mnemònics, operands i directives) en instruccions màquina. També ofereix la possibilitat de manegar símbols o etiquetes (que posteriorment seran traduïts a adreces físiques) i "macros" (agrupació de seqüències d'operacions i/o instruccions que seran "replicades" cada cop que aquesta sigui referenciada). Anomenarem mòdul a cadascuna de les "porcions" de codi descrites en llenguatge assemblador.
>
> ***Enllaçador Turbo Linker***: Programa que permet enllaçar els diferents mòduls de programa i/o de llibreria a fi de generar un codi únic executable. Cal remarcar que, a partir de diferents mòduls (objectes) generem un únic programa. Aquest programa es crea amb la estructura necessària per ser executat (carregat a memòria i passar-li el control) sobre MS-DOS, i que no és una imatge binària del que hi ha a memòria en el moment d'execució.
>
> ***Depurador Turbo Debugger***: Programa que permet la simulació de l'execució de programes escrits en assemblador. Ens permet la visualització i la interacció amb les diferents àrees de memòria, registres interns, ports, així com l'execució de programes pas a pas amb els resultats d'execució de cada instrucció.

1-L'Assemblador TASM (Turbo Assembler).

Cal escriure el codi font en llenguatge assemblador (format ASCII utilitzant un editor de texts del tipus EDIT del DOS, WordPad de Windows...). Per tal de simplificar el procediment d'assemblatge en aquesta primera pràctica, cal donar a aquest fitxer l'extensió ".ASM". Aquesta aplicació funciona sobre MS-DOS, per això, cal donar noms curts a tots els fitxers (8 caràcters pel nom i 3 per a la extensió, sense símbols "estranys"). Per executar-lo, el cridarem des de la línia del intèrpret de comandes (des de Windows "Inicio, Ejecutar, Command" o bé fer click sobre la icona del MS-DOS).

La comanda bàsica per assemblar és: ***'tasm /zi nomfitxer.asm (retorn)'***

En aquest cas, es genera el codi objecte amb extensió “.OBJ”. Encara que no cal indicar la extensió (si existeix el fitxer amb extensió .asm) nosaltres ho farem per evitar equivocacions. La opció '/zi' genera informació que serà utilitzada posteriorment en el procés de depuració. Aquesta opció ha d'estar en minúscules.

2- L’enllaçador TLINK.

Aquest programa presenta una gran quantitat d’opcions que permeten generar executables optimitzats segons diferents criteris. No oblidem que aquesta aplicació és per “agrupar” diferents mòduls objecte (que poden ser descrits en diferents llenguatges; ASM, C,...). A continuació es descriuen solament la comandes mínimes que permeten generar el codi executable a partir d’un fitxer objecte amb extensió “.OBJ”. Tot i així, les opcions poden ser consultades escrivint: '*tlink (return)'*

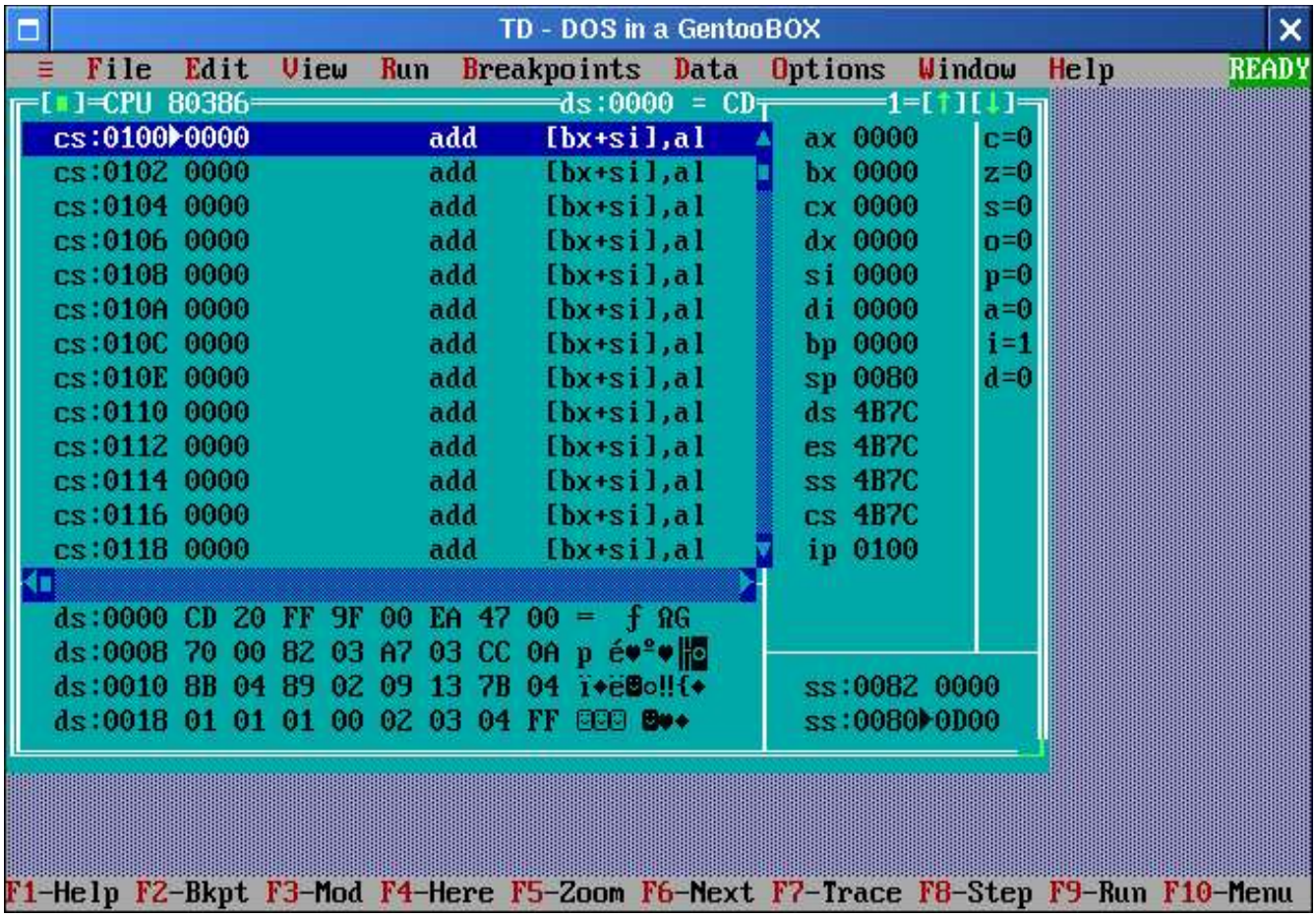
La comanda bàsica per enllaçar el fitxer és: ***'tlink /v /3 nomftxer.OBJ (retorn)'***

En aquest cas el nom del fitxer pot estar en majúscules o minúscules. Les opcions obligatòriament en minúscules. '/v' 'deixa' informació pel depurador i '/3' enllaça la biblioteca d'instruccions del 'x386'.

3-Depurador TDEBUG.

Aquest programa permet la depuració del codi executable. Per cridar aquest programa cal escriure: ***'td nomfitxer.EXE (retorn)'***

En la següent figura es mostra la pantalla principal del depurador.



El td ofereix moltes possibilitats. Només explicarem les que ens interessen a la nostra assignatura (i en concret per aquesta pràctica).

Primer de tot, ens interessa veure la informació a nivell de contingut de registres i de memòria. Per fer això anirem al menú principal (F10) escollirem la opció 'View' (V) i la subopció 'CPU' (C). És a dir, premeu consecutivament les tecles F10-V-C.

Ara podeu veure una nova pantalla amb informació per la depuració. El millor és veure aquesta pantalla el més gran possible, per tant feu un Zoom (premeu la tecla F5). La pantalla que veureu està dividida en 5 zones, que contenen informació sobre el segment de codi, els registres, els flags, la pila i el segment de dades. Podeu passar de una subpantalla a l altre prement la tecla TAB. Al començament, la pantalla de codi mostra el programa que voleu depurar. Si canvieu de pantalla (per mitjà de TAB) anireu (en aquest ordre) a la pantalla de registres, a la de flags, a la de pila i a la de dades, tornant després a la de codi (proveu de fer-ho).

A la pantalla de codi, us trobareu que les primeres instruccions no són les que heu programat vosaltres. Són les instruccions introduïdes per la directiva '**.startup'**. Aquestes instruccions són sempre les mateixes, independentment del programa, i inicialitzen tot l entorn de treball (fixeu-vos que inicialitzen els registres 'ds', 'ss' i 'sp').

La primera instrucció del vostre programa està després d aquestes instruccions, per tant, el primer que hauríeu de fer és executar-les i posar-vos a sobre de la primera instrucció pròpia del vostre programa (que està indicada per l etiqueta INICI, o bé la que hi heu escrit); podeu veure com la informació sobre etiquetes es manté: abans de la vostra primera instrucció hi ha un indicador '#nomfitxer#etiqueta' que marca el nom del fitxer i l etiqueta associats a la instrucció. Per executar una instrucció anirem al menú principal (F10), opció Run (R), subopció Trace intro (T). Això executa una (i només una) instrucció.

Hi ha accions que es fan força sovint; aquestes tenen el que s anomena un curtcircuit ('shortcut') al teclat: una tecla o una combinació de tecles que fan el mateix sense necessitat de passar pel menú cada vegada. Per exemple, al costat de Trace intro trobareu el seu curtcircuit (F7). Això vol dir que és equivalent fer F10-R-T que fer F7.

Per treballar, us recomanem que feu sempre el següent: primer executeu amb F7 les 7 instruccions del '.startup' (observeu com, conforme les executeu, a la pantalla de codi el cursor avança i com els registres i els flags que es modifiquen durant l execució es veuen ressaltats a la pantalla respectiva). Un cop heu executat aquestes instruccions, esteu a la primera instrucció pròpia del vostre programa (identificada per #nomfitxer#etiqueta).

Per veure les dades del vostres programa teniu dues possibilitats: inspeccionar una única dada o controlar el segment de dades.

Per a inspeccionar una dada podeu anar al menú principal (des de qualsevol subpantalla) per mitjà de la tecla F10, escollir la opció Data (D), subopció Inspect (I). Una finestra us preguntarà quina variable voleu veure. Per exemple, pregunteu per la variable 'factor', us sortirà una finestra que us indicarà la seva adreça (segment:desplaçament), la longitud (byte, word, dword) i el seu valor decimal i hexadecimal (per sortir d aquesta finestra, premeu ESC)

L altre possibilitat és controlar el segment de dades. Si us aneu a la subpantalla de dades (per mitjà de TAB) podeu fer ALT-F10, subopció Goto (G) i us preguntarà quina variable voleu veure. Si poseu 'factor' veureu com la pantalla de dades es situa a l adreça DS:0000 (que és on comença la dada 'factor'). Amb la opció d inspecció descrita al paràgraf anterior podeu observar com 'valor' és a DS:0000, 'factor' a DS:0002 i 'producte' a DS:0008, per tant

podeu controlar totes les variables a la mateixa finestra. El problema és que la finestra de dades us mostra la pantalla byte a byte i, en aquest cas, les variables són de mida 'word' i 'dword'.

Si voleu veure les dades del DS amb una mida diferent de byte, feu a la mateixa pantalla de dades ALT-F10 Display as (D) i escolliu mida entre Byte,Word o Long (Nota: en aquest punt, pel TD un 'double word' és un 'long word'); per tant en aquest cas s ha d escollir la opció L.

Les dues últimes instruccions (mov ah, 4c i int 21) són afegides per la directiva '**.exit'**. Si les executeu, el programa acaba.

Ara ja teniu una idea de les possibilitats del td, una eina típica de 'depuració'. Permet la execució instrucció a instrucció del vostre codi oferint-vos la possibilitat d'examinar els canvis que es produeixen en la màquina (registres, memòria, flags...) durant aquesta execució. Deixem per a vosaltres la 'exploració' de les moltes opcions que té aquest programa (consulteu el 'HELP').

Enunciat de la pràctica.

A partir de les eines descrites anteriorment es crearan i executaran (simulació) diferents programes escrits en llenguatge assemblador. Per a tots ells caldrà assemblar-los amb el "TASM", enllaçar-los amb el "TLINK" i simular (verificar) la seva correcta execució amb el "TD".

És convenient familiaritzar-se amb les diferents opcions de les utilitats, veure (i entendre) el procés que va des de la descripció d'un algorisme fins a la seva execució en un processador així com la representació (en memòria) i evolució dels elements que hi intervenen (codi, dades, variables, registres...).

Aconsellem (i així ho fem) seguir un nivell "ascendent" en quant a complexitat i mida dels programes a avaluar i per això podeu començar amb programes com:

```
.model large    ; definició del model que fem servir
.386

.stack  100h    ; definició del segment de pila

.data           ; inici del segment de dades
valor           DW      0325h
factor          DW      0040h
resultat        DD      00h

.code           ;inici del segment de codi
.startup        ; inclusió del codi necessari (segments...)
inici:  MOV     AX,valor        ; el nostre codi efectua una multiplicació de dues variables
        MUL     factor
        MOV     producte,AX     ; deixant el resultat també en una variable
        MOV     producte+2,DX

.exit           ; inclusió del codi de retorn al sistema operatiu
END             ; final del fitxer
```

Per seguir amb bucles del tipus:

```
.data           ; inici del segment de dades
vector          DB 'hola que tal',00h     ; una cadena de text acabada amb 'NULL'
valor           DW ?            ; valor sense inicialitzar
```

```
.code           ;inici del segment de codi
.startup        ; inclusió del codi necessari (segments...)
inici:  MOV     SI,00h          ; Inicialitzo index a 0
        MOV     AX,00h          ; i resultat
bucle:  CMP     vector[SI], 00h ; és el caràcter de final de cadena?
        JE      fi_bucle
        ADD     AX,vector[SI]   ;acumulo el valor
        INC     SI
        JMP     bucle
fi_bucle:
        MOV     valor,AX
.exit           ; inclusió del codi de retorn al sistema operatiu
```

Nota: en aquest últim codi no hi ha totes les directives necessàries. Ambdós codis tenen errors sintàctics i semàntics (no podem permetre que una pràctica sigui únicament 'copiar i verificar').

Recomanacions:

Treballeu en el directori “temp” del disc dur. Al començar la sessió feu:

“CD C:\TEMP(retorn)”, “DEL *.*(retorn)” (no patiu) i
“COPY A:*.* C:\TEMP*.*(retorn)” (suposem que porteu les pràctiques en un disket).

Abans de marxar només caldrà fer “XCOPY C:\TEMP*.* A:*.*”. Amb això evitarem problemes de “desconfiguracions” **(que ningú alteri el sistema sota cap concepte ni escrigui res en cap lloc que no sigui 'C:\TEMP').**

Construïu-vos els vostres fitxers de “procés per lots” (i guardeu-los al directori temp). Podeu fer un fitxer de text (amb l'edit o el Wordpad) anomenat “ASMP.BAT” (diferent del que tenieu pels micros).

Construïu tants fitxers com calgui per tal d’agilitzar les tasques d’assemblatge i enllaçat .

Les eines són a **'C:\BORLANDC\'** però hi ha les rutes adequades a la variable 'path'. D'aquesta manera no cal que 'les crides' facin referència a rutes absolutes, només cal que les efectueu des del directori de treball.

Porteu les pràctiques “preparades” (llegides, enteses i potser “picades”) i proveïu-vos de la documentació necessària (és difícil recordar tot el repertori d’instruccions).