Informatique 5 – Langage assembleur
SMI – Semestre 3
Série1

Débuggeur

I. Débuggage :

Le débuggage est la recherche des erreurs.
Il existe de nombreux logiciels qui offrent un outil de débuggage dans leur environnement intégré, comme Il existe aussi des programmes spéciaux pour le débuggage.
Un débuggeur permet, entre autres, l'affichage des informations sur les variables, les registres etc. et l'exécution d'un programme pas à pas ou son exécution jusqu'à un point d'arrêt.

II. Le programme DEBUG de MSDOS :

Permet de mettre au point des fichiers exécutables

Syntaxe :

Debug [*unite*:][*chemin*]*fichier*[*param*]

Paramètres :

[*unite*:][*chemin*]*fichier*

Précisent l'emplacement et le nom du fichier exécutable à tester.

param

Précise toute information à inclure sur la ligne de commande exigée par le programme à tester.
Si vous ne spécifiez le fichier à mettre au point, vous entrez en mode "commande" de DEBUG

Commandes de Debug :

?	Affiche une liste des commandes de Debug
a	Assemble les mnémoniques
c	Compare deux zones de la mémoire
d	Affiche le contenu d'une zone de la mémoire
e	Entre des données en mémoire à partir d'une adresse donnée
f	Remplit une zone de mémoire avec des valeurs données
g	Exécute le programme chargé en mémoire
h	Réalise des opérations arithmétiques sur des valeurs hexa.
i	Affiche un octet d'un port donné
l	Charge en mémoire un fichier ou des secteurs de disque
m	Copie le contenu d'un bloc de mémoire
n	Spécifie un nom de fichier pour les commandes **l** ou **w** et spécifie les paramètres pour le fichier mis au point
o	Envoie la valeur d'un octet vers un port de sortie.
p	Exécute une boucle, une instruction de chaîne répétée, une interruption logicielle ou une sous-routine.
q	Quitte Debug

Commandes de Debug (suite) :

r	Affiche ou modifie le contenu d'un ou plusieurs registres
s	Recherche, dans une zone de mémoire, une séquence d'octets spécifiée
t	Exécute une instruction et affiche le contenu de tous les registres, indicateurs et l'instruction décodée suivante
u	Désassemble des octets et affiche les instructions source correspondantes
w	Ecrit le fichier en cours de mise au point sur disque
xa	Alloue de la mémoire paginée
xd	Désactive l'allocation d'un descripteur de mémoire paginée
xm	Affecte des pages logiques aux pages physiques
xs	Affiche des informations sur l'état de la mémoire paginée

III. Exploration de la mémoire :

1. Afficher le contenu de la mémoire à l'adresse par défaut (sans spécification de la plage mémoire) en tant que :
   - Octets en hexadécimal
   - Octets en ASCII
   - Instructions
2. Relevez l'adresse du premier octet décodé (segment et offset). Nous appellerons le segment trouvé "segment de travail".
3. Relevez les 32 premiers octets décodés sous les trois formes cités en 1.
4. Remplissez de 00 le premier Ko du segment de travail.
5. Placez les caractères "SMI" au début du segment de travail. Vérifier que ces caractères sont bien rangés en mémoire et relevez en Hexa les 16 premiers octets du segment de travail.
6. Entrez le code

   ```
   B4 4C CD 21
   ```

   à l'adresse 100h.
   A quelles instructions correspond ce code ?
   Exécutez ce code. Que se passe-t-il ? Pourquoi ?
7. Pouvez-vous relire les caractères "SMI" introduits précédemment ? Pourquoi ?

IV. Les registres :

Exécuter la commande `r` sous DEBUG

1. Quelles sont les informations affichées par cette commande ?
2. Décrivez les registres et les indicateurs que vous connaissez.
3. Avec la commande `a`, saisissez les instructions suivantes à partir de l'adresse 100h

```
MOV AX,FFFF
MOV AH,5
MOV BH,6
MOV AL,BH
```

Exécutez les commandes pas à pas et relever à chaque fois l'état des registres concernés.
Expliquez l'effet de chaque commande.

4. Avec la commande `r` affectez à IP la valeur 100h et exécuter la commande `t`. Que se passe-t-il ? Pourquoi ?
5. Saisissez les instructions suivantes

```
MOV AX,FFFF
MOV BX,FFFF
ADD AX,BX
MOV AX,0001
DEC AX
INC AX
SUB AX,0002
MOV AH,70
MOV BH,50
ADD AH,BH
```

Exécutez les pas à pas et relevez les valeurs des indicateurs et expliquez les changements.

V. Création d'un programme avec DEBUG

1. Saisissez les instructions suivantes à partir de l'offset 100h

```
MOV DX,10B
MOV AH,9
INT 21
MOV AH,4C
INT 21
DB "Mon premier programme avec DEBUG",0D,0A,'$'
```

2. Enregistrer le programme crée sous le nom `ARCHI01.COM`. (Pour enregistre ce programme, mettez dans CX le nombre d'octets qu'il occupe, utilisez `n` pour nommer le fichier puis la commande `w` pour l'écrire sur le disque)
3. Quitter DEBUG et exécutez `ARCHI01.COM`.
4. Que fait ce programme ?
5. Quel est le rôle de la première instruction ?
6. L'interruption 21h exécute la fonctions DOS dont le numéro est placé dans AH. Expliquez alors l'effet des autres instructions.

VI. Débuggage et désassemblage

1. Copier le programme `ARCHI02.EXE` dans votre répertoire de travail, et exécutez-le. Que fait ce programme ?

Entrez en mode DEBUG en exécutant la commande

```
DEBUG  ARCHI02.EXE
```

2. Exécutez le programme avec la commande g. Que remarquez-vous ?
3. Sachant que IP est positionné à la première instruction de ce programme, quelle est l'adresse de cette instruction?
   La dernière instruction de ce programme est celle qui permet de revenir sou DOS. Relever l'adresse de celle-ci.
   Relever ensuite les instructions composant ce programme, et analyser les.
4. Où se trouve la chaîne affichée par le programme ? Retrouvez l'adresse en mémoire de cette chaîne.
   A quoi correspond le segment de l'adresse trouvée ?
5. Changer le message affiché par le programme en

```
Ce programme est modifié directement en mémoire
```

Exécutez le programme pour vérifier votre changement.