

TD2

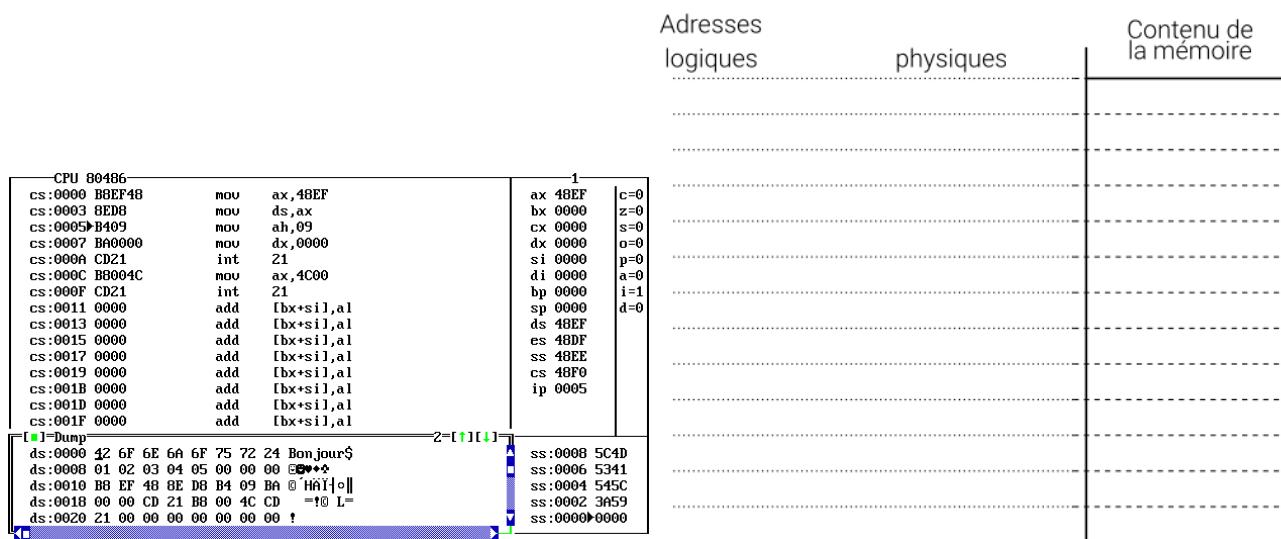
## Segmentation de la mémoire

## **Exercice 1 – Segmentation de la mémoire**

Un programme assembleur occupe plusieurs zones de la mémoire, appelées *segments*. On peut visualiser ces segments avec *Turbo Debugger*. Par exemple, dans la figure ci-dessous :

- Le **segment de code** (cadre haut-gauche) est la zone qui contient les instructions du programme
  - Le **segment de données** (cadre bas-gauche) est la zone mémoire qui contient les données du programme

Un programme peut également occuper deux segments supplémentaires (segment de pile et segment auxiliaire) que nous aborderons dans les prochains TD.



Le schéma ci-contre est une représentation de la mémoire sous la forme d'une colonne de cases mémoire, organisées dans l'ordre croissant de leurs adresses.

1. En supposant que la première case en haut de la colonne est la première case mémoire occupée par le programme de la figure ci-dessus, à quel segment appartient cette case ? quelle est son adresse logique ? son adresse physique ?
  2. Complétez le schéma en donnant toutes les adresses logiques et physiques des cases mémoire de ce même segment ainsi que leurs contenus en hexadécimal.
  3. Quel est l'adresse physique de la première case mémoire du prochain segment dans la mémoire ?
  4. Complétez le schéma en donnant les adresses logiques et physiques ainsi que le contenu de ce segment.

## **Exercice 2 – Lecture dans le segment de données**

La figure ci-dessus donne le contenu du segment de données d'un programme assembleur.

On suppose que le registre DS contient la valeur hexadécimale 48EF.

1. Quel est le contenu de la première case mémoire du segment de données en hexadécimal? en décimal? Quelle est son adresse physique?
  2. Quelle est la valeur contenue dans la case mémoire d'adresse logique DS:0036? Quelle est l'interprétation de cette valeur dans le programme? Quelle est l'adresse physique de cette case mémoire?
  3. Une partie de ce segment de données contient la suite de valeurs 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 (suite de Fibonacci). Quelles sont les adresses logiques des cases qui contiennent ces valeurs? Quel est le format et le mode de stockage de ces valeurs?
  4. Les octets contenus dans les cases d'adresses physiques 48EF1 et 48EF2 forment une seule et même valeur dans le programme. Quelle est cette valeur en hexadécimal? Donnez l'instruction assembleur qui a permis de déclarer cette valeur dans le code source.

## Exercice 3 – Déclaration des segments en assembleur

On souhaite écrire un programme assembleur qui déclare les variables suivantes dans le segment de données :

- une variable **A** qui contient la valeur décimale 42, stockée sur un octet
  - une variable **B** qui contient la valeur hexadécimale **FEDC** sur 2 octets
  - une variable **C** qui contient la valeur hexadécimale **BA987654** sur 4 octets
  - un tableau **puissances** qui contient les valeurs décimales 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024
  - un tableau **vincent** qui contient 20 fois la valeur décimale 100
  - une chaîne de caractères **citation** qui contient "Science sans conscience n'est que ruine de l'âme"

1. Écrivez la déclaration assembleur du segment de données contenant ces variables.
  2. Combien d'octets occupe ce segment ?
  3. Représentez schématiquement l'extrait de mémoire correspondant aux 32 premiers octets de ce segment. Quelles sont les valeurs contenues dans les cases mémoires d'adresse DS : 0006 et DS : 000D ?