Pr: Ismaili Alaoui El Mehdi A.U: 2021/2022

Architecture des ordinateurs
Prise en main d'emu8086

Dans un premier temps, pour simplifier l'apprentissage de l'assembleur, nous allons travailler sur Emu8086 (émulateur logiciel qui simule un microprocesseur).
Emu8086 combine un éditeur de texte, un compilateur (FASM), un débogueur, et un émulateur de logiciel dans un programme simple fonctionnant sur la plate-forme de Windows.
Le fichier squelette.asm (Figure 1) est un fichier que nous vous fournissons pour vous aider à écrire vos programmes assembleur. Enregistrez le fichier dans un répertoire, puis lancez l'Emu8086, ouvrez le fichier avec Emu8086.
Emu8086 se présente dans un premier temps comme un éditeur de texte classique, avec le support d'une colorisation syntaxique du code assembleur (mode éditeur).

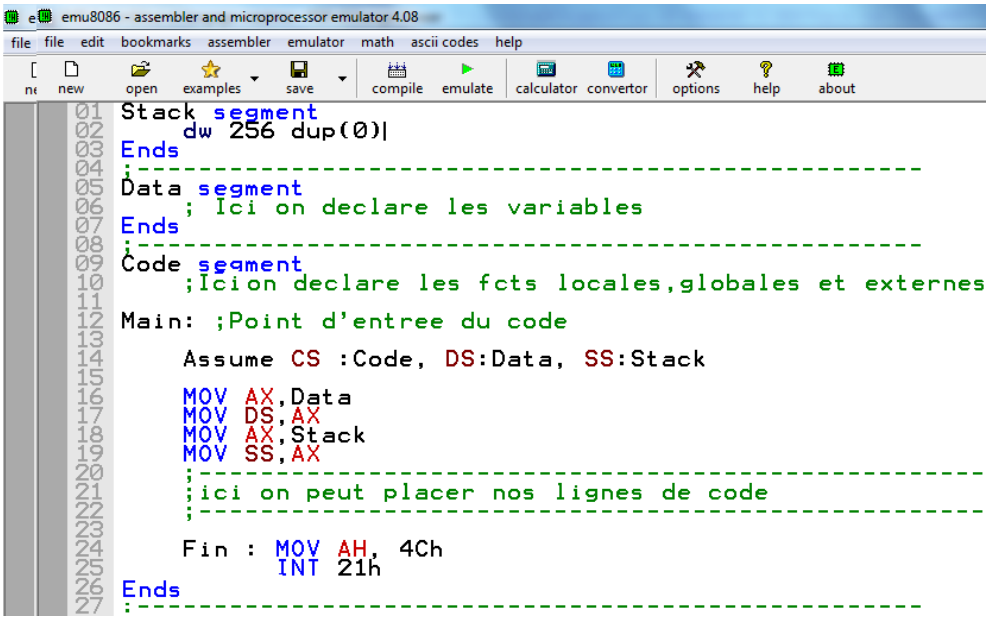

FIGURE 1 – squelette.asm

Le fichier squelette.asm contient la déclaration d'un segment de pile de 512 octets, la déclaration d'un segment de données vide ainsi que la déclaration d'un segment de code. Dans ce dernier, on retrouve la directive ASSUME permettant d'initialiser les trois registres de segments CS, DS et SS. On trouve aussi une zone de déclaration de fonctions publiques , une zone

de déclaration de fonctions externes, une zone de déclaration de fonctions locales, puis du code assembleur.
Ce code débute par l'étiquette debut, puis on retrouve l'initialisation des segments de données et de pile. Un espace réservé à l'écriture de vos programmes suit ces éléments. Enfin, l'étiquette fin indique (informellement) la fin du programme. Les deux lignes suivantes correspondent à l'appel standards de l'interruption DOS dont le rôle est de terminer d'un programme de manière normal (libération de la mémoire, ...). Ensuite on trouve la fin du segment de code puis celle du programme.
Pensez lorsque vous écrirez un programme à bien partir de ce fichier squelette en le sauvegardant sous un autre nom avant de le modifier.
Créez un fichier TP1.asm identique au fichier squelette.asm. Compilez et exécutez le programme (bouton RUN). Validez la fin de l'exécution avec OK.
Deux nouvelles fenêtres viennent de s'ouvrir, il s'agit du mode exécution de l'émulateur (Figure 2).

— La fenêtre "original source code" contient le code tel que vous l'avez écrit. La ligne surlignée en jaune est la prochaine instruction qui va être exécutée. Comme on le verra plus loin dans les TPs.
— La fenêtre "emulator": contient 4 zones que nous allons détailler ci-dessous:

1. Les boutons vont nous permettre d'exécuter notre programme soit pas à pas, i.e. instruction par instruction avec la possibilité de revenir en arrière, soit automatiquement (en pouvant régler un délai d'attente entre chaque instruction).
   Réinitialisez votre programme (reload) puis exécutez-le pas à pas (single step et back step) jusqu'à la fin. Faites une nouvelle réinitialisation, réglez le délai sur 400 ms et faites une exécution normale (run).

2. La deuxième zone contient une représentation des registres, et permet de visualiser leur valeur hexadécimale tout au long de l'exécution de votre programme. On notera que les registres AX, BX, CX et DX sont représenter couper en deux. Cela permet de visualiser simplement leur décomposition en registres 8 bits. Il est possible de modifier la valeur des registres de cette manière (simple clic) ou d'ouvrir une vue détaillée du registre (double clic).

3. La troisième zone représente la mémoire octet par octet. Sur une ligne, on retrouve l'adresse mémoire sur 20 bits, la valeur de l'octet en hexadécimale, en décimale et sa représentation en ASCII.

4. La quatrième zone contient le code de votre programme une fois les traductions d'adresses terminées. Il n'y a plus aucune étiquette ni nom de variable. Il s'agit du code réellement exécuté.

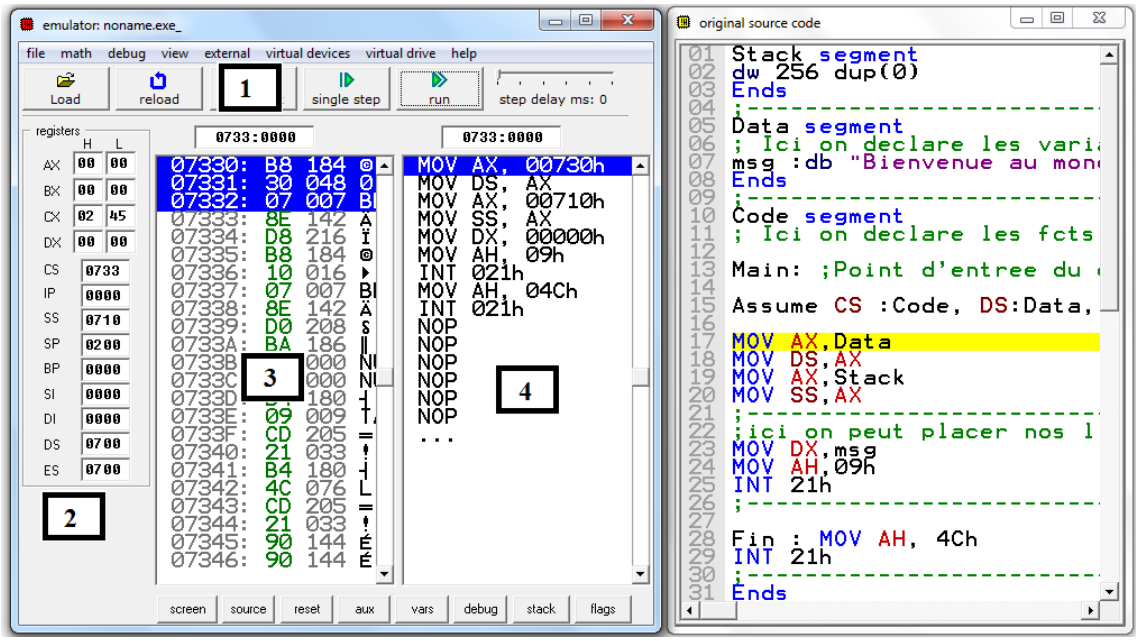

FIGURE 2 – Mode exécution de l'émulateur.