



HEB-ESI
1^{re} année

Année 2005-2006
BEJ, HAL, JCJ,
NVS, PBT

LMI-TD02 : Instructions de contrôle et de test, lecture au clavier et affichage à l'écran

1 But du TD

Manipulation de l'instruction de saut inconditionnel : `jmp`.

Manipulation des instructions de saut conditionnel : `jcc` et, éventuellement, `loop`.

Manipulation de l'instruction de test : `cmp`.

Éventuellement, manipulation de l'instruction arithmétique : `neg`.

Utilisation d'interruptions logicielles pour la lecture de données au clavier et l'affichage d'informations à l'écran.

2 Comparaisons et sauts

Ex1. Soient `a` et `valAbs` deux variables de taille un octet. Écrivez un code qui stocke dans `valAbs` la valeur absolue du contenu de `a`.

Testez votre code dans `td` avec les valeurs suivantes :

- `a` : 25 ;
- `a` : -30 ;
- `a` : 200.

Qu'observez-vous ?

Ex2. Soient `a`, `aMin` et `aMax` trois variables de taille deux octets et `estIn` une variable de taille un octet. On suppose avoir toujours `aMin < aMax`. Écrivez deux codes qui stockent dans `estIn` :

- la valeur 1 si `aMin ≤ a < aMax` ;
- la valeur 0 sinon.

Le premier code ne manipule que des entiers non signés. Le second est adapté aux entiers signés.

Testez vos codes dans `td`.

3 Boucles

Ex3. Soient `a` et `somme` deux variables de taille deux octets. On considère le contenu de `a` toujours positif. Écrivez un code qui stocke dans `somme` la somme des entiers positifs de 0 à `a` ($\text{somme} \leftarrow \sum_{i=0}^a i$).

Testez votre code dans `td` avec les valeurs suivantes :

- `a` : 25 ;
- `a` : 361 ;
- `a` : 400.

Qu'observez-vous ?

Modifiez votre code de sorte que la valeur nulle (0) est stockée dans `somme` si $\left(\sum_{i=0}^a i\right)$ n'est pas stockable sur deux octets.

4 Lecture au clavier et affichage à l'écran

De nombreux services d'interruption permettent la lecture d'un caractère au clavier. Citons, entre autre, :

- `int 16h`, service `00h` : interruption BIOS, lecture d'un caractère, sans écho ;
- `int 16h`, service `10h` : interruption BIOS, lecture d'un caractère (clavier étendu), sans écho ;
- `int 21h`, service `01h` : interruption DOS, lecture d'un caractère (clavier étendu), avec écho ;
- `int 21h`, service `07h` : interruption DOS, lecture d'un caractère (clavier étendu), sans écho ;
- `int 21h`, service `08h` : interruption DOS, lecture d'un caractère (clavier étendu), sans écho.

Pour plus de détails à leur sujet, référez-vous, par exemple, à `helppc`.

Notez qu'il existe également des fonctions d'interruptions permettant la lecture d'une chaîne de caractères (`int 21h`, service `0Ah` ou `int 21h`, service `3Fh`).

Pour ce qui concerne l'affichage de données à l'écran, la première chose à faire est de sélectionner le mode vidéo. Le service d'interruption `00h` de l'interruption BIOS `int 10h` le permet.

Au cours de ce TD, nous allons afficher uniquement du texte à l'écran. Pour sélectionner le mode vidéo texte, il faut placer la valeur `03h` dans `a1` avant l'appel de l'`int 10h`, fonction `00h`. Réalisez cet appel avant tout affichage (et toute lecture avec écho) dans vos codes solutions aux [Ex4](#) et [Ex5](#).

En toute rigueur, cet appel n'est pas nécessaire puisque le mode texte est le mode par défaut de l'invite de commande. Cependant, il a aussi pour effet de nettoyer l'écran.

Pour réaliser l'affichage de texte à l'écran, il faut savoir que celui-ci, en mode texte, est constitué de 25 lignes (numérotées de 0 à 24) de 80 colonnes (numérotées de 0 à 79) de caractères. La coordonnées [ligne,colonne] du caractère tout en haut et tout à gauche de l'écran est [0,0], celle du caractère tout en haut et tout à droite est [0,79] et celle du caractère tout en bas et tout à droite est [24,79].

Pour positionner le curseur à l'écran, on utilise le service 02h de l'interruption `int 10h`. Son usage requiert de charger préalablement la valeur 00h dans `bh`, le numéro de la *ligne* dans `dh` et celui de la *colonne* dans `dl`.

L'affichage d'un caractère, à la position du curseur, peut être réalisé par les fonctions d'interruptions :

- `int 10h`, service 09h : interruption BIOS, affichage d'un caractère, avec attributs (couleur d'avant et d'arrière plan), sans déplacement du curseur ;
- `int 21h`, service 02h : interruption DOS, affichage d'un caractère avec avancement du curseur.

Pour plus de détails à leur sujet, référez-vous, par exemple, à `helppc`.

Il existe également un service d'interruption pour afficher une chaîne de caractères : `int 21h`, fonction 09h. Pour l'utiliser, il suffit de charger dans `dx` l'adresse du premier caractère de la chaîne à afficher et *il faut que la chaîne s'achève par le caractère \$*.

Ex4. Écrivez un code qui lit une chaîne de maximum 40 caractères puis la réaffiche en majuscule, en minuscule puis en inversant sa casse. Si la chaîne fait moins de 40 caractères, l'utilisateur frappe la touche « return » pour signaler sa fin. On suppose que l'utilisateur n'encode que des lettres, des chiffres et le « return ». Ne vous occupez pas de valider les données fournies. Les chiffres ne sont pas modifiés. La lecture se fait à partir de la ligne 1, colonne 0. Les affichages des différentes chaînes transformées se font aux lignes 3, 5 et 7, toujours à partir de la colonne 4. Après l'affichage des résultats, l'utilisateur doit enfoncer une touche quelconque pour mettre un terme au programme.

Conseils : Pour lire les chaînes, utilisez l'`int 21h`, fonction 01h. Pour les affichages, l'`int 21h`, fonction 09h est de loin la plus aisée !

5 Exercice d'évaluation : lecture de deux nombres, calcul et affichage de leur produit

Ex5. Écrivez un programme qui lit deux nombres entiers, positif ou négatif, constitué de maximum trois (3) chiffres et stockables chacun dans un octet. Le second entier est encodé sur une nouvelle ligne. Pour valider un nombre de moins de trois (3) chiffres, l'utilisateur doit frapper la touche « return ». Le produit de ces nombres est ensuite calculé puis affiché, deux lignes plus bas. Pour quitter, l'utilisateur doit enfoncer une touche quelconque.

Veillez à valider les données fournies par l'utilisateur :

- si il encode autre chose qu'un chiffre, « return » ou, au début de l'encodage d'un nombre, le caractère « - » ou le caractère « + » (ce dernier étant facultatif) ;
- si il frappe la touche « return » avant d'avoir fourni un nombre valide ;
- si l'un des nombres qu'il fournit n'est pas représentable sur un octet (voir également la remarque plus bas) ;

affichez un message d'erreur et attendez que l'utilisateur frappe une touche quelconque pour quitter.

Remarque : Libre à vous de considérer les nombres encodés comme des nombres signés ou non...