
Epreuve de moyenne durée

(Durée 1h30)

Exercice1 (5 pt)

Pour chaque question, mentionnez le numéro de la question et la ou les réponse(s) correcte(s)

Les réponses correctes sont soulignées (1pt par question).

1. La mémoire d'un processeur est divisée en bancs:
 - a. afin de désactiver des lignes du bus de données,
 - b. d'augmenter la capacité de la mémoire,
 - c. de réduire la taille de l'adresse physique.

2. 20 Mo est :
 - a. est un multiple de 16.
 - b. peut correspondre à une taille d'un segment mémoire (8086).
 - c. est égale à 2000000H.

3. Une adresse physique est :
 - a. connue à la compilation.
 - b. se calcule à partir d'une adresse logique, celle-ci est toujours connue à la compilation.
 - c. est sur 20 bits.
 - d. est stockée dans un registre.

4. La taille d'une adresse logique (8086) dépend.
 - a. du nombre de bancs de la mémoire centrale.
 - b. de la taille des registres.
 - c. de la taille maximale d'un segment.
 - d. de la taille du bus d'adresses.

5. La file des instructions contient toujours à un instant T (pendant l'exécution d'un programme) :
 - a. une seule instruction dont la taille est de 6 octets.
 - b. 2 instructions consécutives dont la somme des tailles est de 6 octets.
 - c. une ou plusieurs instructions.
 - d. un ou plusieurs octets de code.



Exercice 2 (8086)

On considère une matrice MATT d'octets, de N lignes et N colonnes, stockée ligne par ligne.

1. Déclarer la matrice.

MATT DB N dup (Ndup (?)) OU MATT DB N*N dup(?) (2pt)

2. Ecrire un programme assembleur qui permet de remplacer chaque élément de la ligne i par la valeur i. Cette mise à jour concerne toutes les lignes.

Il existe différentes solutions, en voici une ci-dessous (5 pt):

```
XOR SI,SI
XOR DI, DI
MOV AL, 0
LIGNESUIVANTE : MOV CX, N
REFAIRE: MOV MATT[SI], AL
INC SI
LOOP REFAIRE
INC AL
INC DI
CMP DI, N
JNZ LIGNESUIVANTE
.....
```

Exercice 3 (8086)

On considère un programme assembleur tel que le data segment soit décrit comme suit :

La première variable qui est un tableau tab1 de Y mots (2 octets), est suivie dans cet ordre, de 100 variables de type octet et d'une dernière variable tab2 qui est un tableau de X octets.

1. Donner une relation entre X et Y.

Il faut considérer la propriété suivante : La taille d'un segment est inférieure ou égale à 64ko. (2 pt)

2. On veut transférer les valeurs de TAB2 (en maintenant l'ordre des valeurs de TAB2) le plus proche possible de la fin de tab1. On commencera le rangement à partir de la première adresse multiple de 16 que l'on rencontre après la fin de tab1. Ecrire un programme assembleur qui réalise ce transfert. Il est évident que durant le transfert, les 100 variables peuvent être écrasées et il n'est pas demandé de les sauvegarder.

(4pt)

```
MOV SI , SIZE TAB1
MOV CX, length TAB2
MOV DI, SI
AND SI, 000FH
JZ TRANSFERT
MOV AX, 16
SUB AX, SI
ADD DI, AX
MOV SI, 0
TRANSFERT: MOV AL, byte ptr TAB2[SI]
MOV [DI], AL
INC DI
INC SI
LOOP TRANSFERT
```

3. Calculer le nombre d'accès généré par votre programme. Pouvez réduire ce nombre? Discutez. (2pts)

NOMBRE D'accès=2X ;

on peut réduire ce nombre si transfert TAB2 mot par mot. Des explications sont exigées dans cette partie. De plus, vos réponses doivent être cohérentes.
