

ULB - Ecole Polytechnique
Première/seconde année Bachelier

Examen d'informatique – INFO – H100
9 juin 2010

Nom, Prénom : **Tiberghien Jacques**

*Répondez de façon concise aux questions posées dans l'emplacement prévu à cet effet. Il ne s'agit pas d'écrire un cours, il suffit de montrer, en quelques mots, que vous avez compris le sujet sur lequel porte la question. Des éléments de réponse qui ne contribuent pas à cet objectif peuvent vous faire perdre des points.
Je vous souhaite un excellent examen ! Jacques Tiberghien.*

Question 1. (2 p)

Un environnement de programmation particulier pour systèmes embarqués vous permet, au choix, d'appeler des fonctions ou d'insérer leur code là où elles sont appelées (« macros »). Le code de la fonction *f* occupe 3000 octets. Pendant l'exécution du programme *f* est appelée 1000 fois, de 10 endroits différents. L'ensemble des opérations nécessaire pour appeler une fonction et en revenir dure 2 μ s. Donnez pour chaque solution, la perte de place ou de temps par rapport à l'autre solution.

Approche	Place perdue	Temps perdu
Appel de fonction	Optimal du point de vue place (une seule copie de <i>f</i>)	Chaque appel à la fonction coûte 2 μ s $2 * 1000 = 2 \text{ ms.}$
Expansion de macro	Puisque <i>f</i> est appelée de 10 endroits différents, il faudra insérer 10 copies de <i>f</i> . Si l'on utilise une fonction le code ne sera présent qu'une fois, la perte de place est donc de $3000*(10-1)= 27000$ octets	Optimal du point de vue temps (pas de temps perdu pour appeler la fonction, gérer les liens et retourner de la fonction)

Cette question sert à vérifier si on a compris la différence entre l'expansion statique ou dynamique de fonctions.

Nom, Prénom :

Question 2. (2p)

Quels bénéfices peut on attendre si, outre l'adressage direct, les instructions de saut peuvent utiliser l'adressage relatif par rapport au compteur ordinal ? Expliquez !

	Nature de l'avantage	Explication
Avantage 1	Programmes plus courts	La plupart des sauts étant à courte distance dans des programmes bien structurés, il faudra moins de bits pour donner la destination
Avantage 2	Programme peut être placé n'importe où dans la mémoire	Si l'on utilise exclusivement l'adressage relatif au compteur ordinal, le programme peut être chargé n'importe où en mémoire sans nécessiter d'adaptations des adresses. Ceci est intéressant, entre autres, pour les programmes vendus en ROM et destinés à des petits systèmes embarqués.

Cette question sert à vérifier si les différents modes d'adressage et les motivations qui les sous-tendent sont comprises.

Remarque relative aux réponses: Ce n'est pas l'emploi de l'adressage relatif qui rend les programmes bien structurés. L'adressage relatif est d'autant plus efficace que le programme est mieux structuré!

Nom, Prénom :

Question 4. (3 p)

Ecrivez en assembleur **pour la machine à pile** décrite dans le chapitre 3 de la troisième partie du cours le code qui permet de remplacer la valeur d'une variable locale qui se trouve à l'adresse x dans la pile par sa valeur absolue. Ce n'est **pas une fonction** complète qui est demandée, simplement un petit morceau de programme.

```
// Calcul de la valeur absolue du contenu de x
    LIT    0        // Vérifier le signe de x
    LOD    0,x      // valeur de x au sommet de la pile
    LE?   // si x>=0 il y a true sur la pile
    JPT   pos      // x est positif, il ne faut rien faire
    LOD    0,x      // valeur de x au sommet de la pile
    CHS   // Changer le signe
    STO    0,x      // réécrire la valeur de x dans x
pos     ...      // suite du programme
```

Cette question permet de voir si vous avez des notions de programmation en langage élémentaire et de récompenser ceux qui se sont entraînés avec les outils didactiques mis à votre disposition.

Remarque relative aux réponses:

- 1. Il est souhaitable de LIRE la question! Un tres grand nombre d'étudiants ont fourni le code d'une fonction pour la machine à registres!!!*
- 2. Beaucoup d'étudiants oublient que les opérations faites sur la pile « consomment » les valeurs au sommet de cette pile. Après avoir vérifié le signe de x , il faut remettre x sur la pile.*
- 3. Il ne sert à rien de faire un saut vers l'instruction suivante:*

```
    JPT neg
    neg CHS
```

Nom, Prénom :

Question 5. (2 p)

Un ordinateur dispose d'une horloge à 1 GHz, La mémoire centrale a un temps d'accès de 50 ns et la mémoire cache un temps d'accès de 5 ns. Donnez le nombre de temps d'attente t_w qui seront insérés dans les cycles mémoire, dans le cas où l'information désirée ne se trouve pas dans le cache, se trouve dans le cache et enfin en moyenne, en sachant que le taux de succès du cache est de 90%.

Avec une horloge de 1 GHz un temps d'horloge est de 10^{-9} s ou 1 ns.(consultez le transparent au début du cours qui donne les noms de tous les ordres de grandeur entre 10^{-15} et 10^{+15})

	Nombre de temps d'attente
Donnée présente dans le cache	$5-3 = 2$ Un cycle mémoire sans temps d'attente prend trois temps d'horloge, ou 3 ns. Ici il faut 5ns, il faut donc insérer 2 temps d'attente.
Donnée pas présente dans le cache	$50-3 = 47$
Moyenne, taux de succès = 90%	$(10*50+90*5)/100-3 = 7$ En fait, 6.5 mais il est difficile d'insérer un demi temps d'attente! Comme il s'agit de moyennes, 6.5 est aussi acceptable.

Cette question permet de voir si vous comprenez suffisamment l'accès à la mémoire hiérarchisée ainsi que le rôle des temps d'attente.

Remarque relative aux réponses:

1. Je demandais un NOMBRE de temps d'attente. Une majorité d'étudiants ont donné une réponse en ns (ou en s)!
2. Beaucoup d'étudiants semblent croire qu'il faille attendre plus longtemps quand la donnée est présente dans le cache ???
3. Réfléchir un instant aux ordres de grandeur mentionnés ne peut pas faire de tort: 10^{11} s (environ 3000 ans) pour accéder la mémoire d'un ordi me semble long! (l'exemple est extrême, mais des dizaines de secondes n'étaient pas exceptionnelles)

Nom, Prénom :

Question 6. (2 p)

Un ordinateur a une mémoire centrale organisée en mots adressables de 8 bits (un octet) et utilise des adresses de 32 bits. Quelle est, sans utiliser d'astuces, la taille maximale de la mémoire centrale ? Serait-il possible, sans changer le format des adresses dans les instructions, de dépasser cette limite et si oui, comment ?

Taille maximum de la mémoire = 2^{32} octets , soit environ 4 GBytes

Moyen de dépasser cette limite : En utilisant la pagination, avec des numéros de pages physiques plus longs que les numéros de pages logiques. A chaque moment, seulement 2^{32} octets sont accessibles, mais il suffit de changer le contenu de la table des pages pour rendre accessibles d'autres pages.

Cette question permet de voir si vous avez compris la pagination et son utilité dans le prolongement de la durée de vie utile de l'architectures d'une famille d'ordinateurs.

Remarque relative aux réponses:

Je considérais la première partie de la question comme un cadeau d'un point, malheureusement une majorité n'a pas voulu de ce cadeau. Au moins une dizaine d'étudiants pensent que dans un ordinateur avec des adresses de 32 bits, la taille de la mémoire est limitée à 32 octets!!! Une vingtaine sont plus généreux et consentent à avoir des mémoires de $8 \times 32 (=256)$ octets! Il est vrai que certains de mes exemples d'attent un peu, mais même les premiers ordinateurs, en 1946, avaient plus de mémoire que cela.

Nom, Prénom :

Question 7. (2p)

Un texte de 10^5 caractères est imprimé en 100 secondes. Le transfert du texte de la mémoire vers l'imprimante peut se faire avec ou sans DMA. L'ordinateur exécute 10^7 instructions par seconde, le transfert d'un caractère par DMA prend 10^{-7} secondes. Le DMAC peut transférer des blocs de 10^4 caractères. Le gestionnaire d'interruptions exécute environ 1000 instructions par interruption. Calculez de quel pourcentage l'exécution d'autres programmes sera ralentie pendant l'impression si le transfert se fait avec ou sans DMA.

Calculs communs :

Nombre d'instructions que l'ordinateur peut exécuter en 100 s = $10^7 * 100 = 10^9$

**Sans DMA : Nombre d'instructions à exécuter pour l'impression = $1000 * 10^5 = 10^8$
(pour chaque caractère il faut exécuter les 1000 instructions du gestionnaire d'interruptions)**

Ralentissement des autres programmes = $10^8 / 10^9 = 10\%$

Avec DMA : Nombre de blocs à transmettre = $10^5 / 10^4 = 10$

Nombre d'instructions à exécuter pour l'impression = $1000 * 10 = 10^4$

Ralentissement des autres programmes dû aux interruptions = $10^4 / 10^9 = 0,001 \%$

Ralentissement dû au vol de cycles mémoire = $10^5 / 10^9 = 0,01\%$

Ralentissement total = $0,001 + 0,01 = 0,011 \%$

Cette question sert à vérifier si vous comprenez les différentes techniques pour les opérations d'entrée-sortie et leurs conséquences sur la performance.

Remarque relative aux réponses:

- Je demandais un pourcentage de ralentissement. Une majorité d'étudiants ont donné des réponse en secondes ou en caractères!
- Beaucoup d'étudiants semblent croire que le DMA ralentit les transferts de données alors que les interruptions ne causent pas de perte de temps.

Nom, Prénom :

Question 8. (2 p)

Récemment sont apparus sur le marché des disques dont les secteurs ont 4096 octets au lieu de 512. Quels avantages peut-on en espérer et quelles difficultés conceptuelles y aurait-il pour les employer avec le système de gestion de fichiers NTFS (il ne faut pas écrire qu'il faut changer le pilote, cela est évident, mais y a-t-il des changements profonds à apporter à NTFS ?)

Que pensez vous de cette manière de faire ?

- vitesse de transfert nettement plus élevée
- capacité 8 fois plus grande tout en gardant le même nombre de bits pour identifier les secteurs

Conséquences conceptuelles pour NTFS ?

Il ne devrait pas y en avoir puisque NTFS utilise de toute façon le concept de « cluster » qui est un nombre entier de secteurs.

Cette question sert à vérifier si vous comprenez les implications des choix matériels dans le dimensionnement des disques.

Remarque relative aux réponses:

- 1. L'augmentation de la taille des secteurs n'implique pas l'augmentation de la taille du disque. Ce n'est que si le nombre bits disponibles pour numérotter les secteurs est limitatif qu'une augmentation de la taille des secteurs permet une augmentation de la taille maximale du disque.*
- 2. Augmenter la taille des secteurs n'entraîne pas nécessairement une augmentation de la taille des clusters. Si avec un disque avec secteurs de 512 octets on utilisait des clusters de 32 secteurs, il est probable qu'on utilisera des clusters de 4 secteurs avec un disque dont les secteurs ont 4096 octets!*
- 3. Il n'y a rien à changer à la MFT de NTFS. Ce système est conçu pour accommoder des clusters dont la taille peut varier de 512 octets à des dizaines de kilooctets.*
- 4. NTFS n'utilise ni FAT16 ni FAT32!*

Nom, Prénom :

Question 9. (2 p)

En sachant que dans beaucoup de réseaux de chemins de fer les signaux sont verts par défaut et ne deviennent rouges que si un train est engagé dans la section protégée par le signal, ne faudrait-il pas prendre des précautions supplémentaires pour protéger une section où deux lignes se rejoignent ? Expliquez et proposez une solution.

En effet, avec ce système simpliste, il se pourrait que deux trains se présentent simultanément dans les deux branches du Y, voient tous les deux un signal vert, s'engagent tous les deux et fassent passer les signaux au rouge ce qui empêcherait un troisième train d'entrer, mais n'éviterait nullement les deux premiers d'entrer en collision.

Pour protéger une telle section de voies il faudrait utiliser un système plus complexe, inspiré par l'algorithme de Deckers.

Cette question sert à vérifier si vous avez compris les bases essentielles des processus concurrents.

Remarque relative aux réponses:

- 1. Si certains d'entre vous iront travailler au chemins de fer sans avoir amélioré leur compréhension de sections critiques, je préfère ne plus prendre le train! Baser la sécurité des croisements de voies sur l'horaire des trains me semble être une forme de roulette russe à grande échelle et avec ceux qui sont moins téméraires, je crains que pas mal de trains doivent passer la nuit devant un signal qui protège une voie déserte.*
- 2. Placer 4 signaux au lieu de 2 n'élimine pas le problème fondamental de la simultanéité. Le « test and set » est théoriquement excellent pour des processus d'ordinateur, mais je me demande toujours comment on pourrait l'implémenter pour des trains!*

Je vous souhaite d'excellentes vacances, et à ceux qui doivent repasser des examens, je conseille d'envisager de réfléchir pendant les examens. Je vous assure, cela a un effet bénéfique sur les points.