ULB - Ecole Polytechnique
Première/seconde année Bachelier

Examen d'informatique – INFO – H100
8 juin 2009

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)
Quelle méthode d'adressage serait utile pour faciliter l'implémentation des instructions C++ suivantes ? Justifiez brièvement vos réponses.

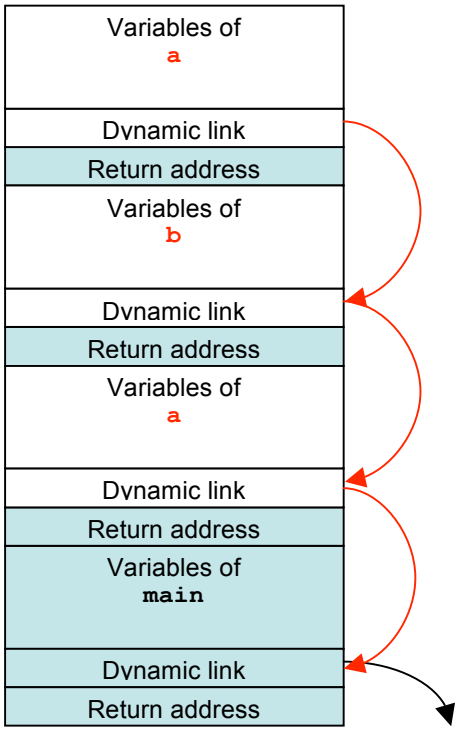
Instruction C++	Adressage	Justification
`a[i] = 5;`	**relatif**	**Il faut additionner l'adresse du début de a et la valeur de l'index i**
`void unefonction (int &x) {…}`	**indirect**	**Les paramètres passés par référence doivent être accédés indirectement**
`b = *unpointeur;`	**indirect**	**La variable `unpointeur` contient l'adresse de la valeur à assigner à `b`.**
`break;`	**direct**	**Saut vers une adresse prédéterminée**

Cette question est destinée à vérifier la compréhension de l'utilité des différents modes d'adressage.

Question 2. (2p)

Complétez le schéma ci-contre au moment où la fonction **`main`** a appelé la fonction **`a`**, qui elle-même a appelé la fonction **`b`**, qui à son tour a appelé la fonction **`a`**. Représentez les liens dynamiques par des flèches. Il ne faut pas représenter les adresses de retour ni les liens statiques.

Cette question sert à vérifier la bonne compréhension de l'allocation dynamique de mémoire lors de l'appel de fonctions.



Nom, Prénom :

Question 3. (3p)

Qu'écrit le programme suivant sur l'écran ?

```
#include <iostream.h>
#include <stdlib.h>
char a = '1'; char b = '2';  char c = '3'; char d = '4';
 void change(char b, char &c)
 { char a = '5'; b = '6'; c = '7'; d = '8';
   cout << a <<b <<c <<d <<" "; }
 void main()
 { cout << a <<b <<c <<d <<" ";
   change(b,b); cout << a <<b <<c <<d <<" ";
   change(b,c); cout << a <<b <<c <<d <<"\n";
   cin >> a; }
```

1	2	3	4		5	6	7	8		1	7	3	8		5	6	7	8		1	7	7	8								

Il existe deux variables appelées a, l'une déclarée globalement, l'autre déclarée dans la fonction **change**. *Elles sont bien entendues indépendantes l'une de l'autre.*
Il existe une variable b, déclarée globalement, et, dans **change**, *un paramètre formel b, passé par valeur.*
Il existe une variable c, déclarée globalement, et, dans **change**, *un paramètre formel c, passé par référence.*
Il existe enfin une variable d, déclarée globalement et utilisée aussi bien dans le bloc global que dans la fonction **change**.
Cette question sert à vérifier si vous comprenez les variables locales, les variables globales, les paramètres passés par valeur et ceux passés par référence. Il sert également à vérifier si vous avez la patience et la méticulosité nécessaires à la vérification de programmes.

Question 4. (2 p)

Ecrivez en assembleur pour la machine à registres décrite dans le chapitre 3.4 le code qui permet de placer dans le registre D0 la valeur absolue du nombre en complément à 2 qui s'y trouvait. Ce n'est pas une fonction complète qui est demandée, simplement un petit morceau de programme. Vous pouvez utiliser D1 et D2 si nécessaire.

```
// Calcul de la valeur absolue du contenu de D0
       LDI   D1,0   //
       CMP   D0,D1  //
       JNN   pos    // si valeur D0 est >0, rien à faire.
       SUB   D1,D0  // D1 = -D0.
       LOD   D0,D1  // D0 = D1
pos    ...          // suite du programme
```

Nom, Prénom :

Cette question permet de voir si vous avez des notions de programmation en langage élémentaire.

Nom, Prénom :

Question 5. (2 p)

Dans une station de travail personnelle, à l'adresse 20000 se trouve une instruction qui additionne deux termes qui se trouvent en mémoire à des adresses spécifiées et place la somme à une adresse spécifiée. Immédiatement après l'exécution de cette instruction, on constate que c'est l'instruction qui se trouve à l'adresse 1000 (dans la partie de la mémoire où réside le système d'exploitation) qui est exécutée. Aucune interface pour entrées-sorties n'a causé une interruption et on ne trouve nulle part le résultat de la somme.

Que s'est-il passé ? La seule explication crédible est que ce soit le TLB qui ait provoqué une interruption parce que l'adresse d'un des opérandes ou du résultat ne figurait pas dans le TLB et qu'il fallait donc faire appel au module de gestion de la mémoire du système d'exploitation pour effectuer la traduction d'adresse (et éventuellement le chargement) de la page manquante. Ensuite il faudra refaire l'instruction qui se trouve à l'adresse 20000.

Cette question permet de voir si vous comprenez suffisamment la gestion de la mémoire d'un ordinateur contemporain que pour pouvoir diagnostiquer des comportements inattendus.

Question 6. (2 p)

Minimiser les déplacements de la tête du disque est un bon argument pour placer les clusters consécutifs d'un fichier sur la même piste physique ou sur des pistes contiguës. Dans un système moderne comme Windows XP utilisant NTFS il y a deux arguments supplémentaires. Quels sont ils ?

Premier argument : Les caches en mémoire ram pour disque lisent des pistes entières. Si les clusters successifs d'un même fichier se trouvent sur la même piste, la cache sera donc très performante.

Second argument : Dans la MFT de NTFS des clusters physiquement consécutifs n'occupent qu'une seule place dans le tableau qui établit la correspondance entre LCN et PCN. Des clusters physiquement consécutifs réduisent donc la taille de la MFT.

Cette question permet de voir si vous faites l'effort d'établir des relations entre notions abordées dans des chapitres différents.

Question 7. (2p)

Un contrôleur d'accès direct à la mémoire transfère un bloc de 10^6 octets en 1 s. La mémoire centrale a un temps d'accès de 100 ns. Si l'on ne tient pas compte de l'effet de caches éventuels, de quel pourcentage le temps d'exécution des programmes sera allongé pendant ce transfert ?

La mémoire peut transférer 10^7 octets par seconde, dont 10^6 à l'usage du contrôleur DMA, les 9.10^6 autres restant disponibles pour l'unité centrale.
Celle-ci sera donc ralentie de 10%

Cette question sert à vérifier si vous comprenez le fonctionnement d'un contrôleur DMA.

Nom, Prénom :

Question 8. (2 p)
Un groupe d'étudiants doit coopérer à la rédaction d'un rapport, mais ils habitent à grande distance les uns des autres. Ils décident donc de placer le rapport sur un serveur de fichiers et d'y travailler à tour de rôle. Ainsi, quand quelqu'un veut travailler, il ou elle fait une copie locale du fichier commun, travaille sur cette copie locale et ensuite remplace la version commune par la version locale retravaillée.

Que pensez vous de cette manière de faire ? Si deux étudiants travaillent simultanément le travail du premier qui sauve son travail sera effacé par le second !
Quelles améliorations suggéreriez vous ? Utiliser un algorithme similaire à celui de Deckers permettant de faire savoir à tous les membres du groupe qui va copier le fichier et assurer à chaque instant l'unicité du « travailleur »

Cette question sert à vérifier si vous comprenez les problèmes posés par les processus concurrents.

Question 9. (3 p)
Si vous connaissez le nombre de cycles d'horloge nécessaires pour exécuter les différentes instructions de la machine à pile décrite dans le chapitre 3.3 vous devriez pouvoir transformer le simulateur décrit dans ce chapitre pour déterminer le temps nécessaire à l'exécution d'un programme donné.

Comment vous y prendriez vous ? Définir une variable `temps` initialisée au début de la simulation et incrémentée lors de l'exécution de chaque instruction par le nombre de cycles d'horloge correspondant à l'instruction simulée.
Comment pourriez vous venir compte d'une mémoire cache éventuelle ? Il faudrait pour cela simuler, outre le processeur, la mémoire cache. Cela risque d'être fastidieux, mais possible car toutes les informations nécessaires sont disponibles, à condition d'admettre que le programme est seul à être exécuté.
Les temps trouvés à l'aide de votre simulateur risquent d'être optimistes. Pourquoi ? Parce qu'ils ne tiennent pas compte d'éventuels transferts DMA.

Cette question est destinée à voir si vous avez compris les principes de la simulation d'un processeur séquentiel.