



CONCOURS INTERNE POUR LE RECRUTEMENT D'ELEVES INGENIEURS DES TRAVAUX DE LA METEOROLOGIE

SESSION 2016

EPREUVE FACULTATIVE A OPTION :

TRAITEMENT DE L'INFORMATION

Durée : 3 heures

Coefficient : 2 (pour les points au dessus de 10)

La rigueur, le soin et la clarté apportés à la rédaction des réponses seront pris en compte dans la rédaction.
L'utilisation de toute documentation sur support papier ou support électronique est strictement interdite.

Cette épreuve comporte cinq parties indépendantes. Elles peuvent être abordées dans l'ordre de choix des candidats.

Barème envisagé :

- Partie 1 : structure des calculateurs (2 points)
- Partie 2 : représentation de l'information (2,5 points)
- Partie 3 : programmation, langages et systèmes d'exploitation (8 points)
- Partie 4 : réseaux de communication (4 points)
- Partie 5 : tri de données (3,5 points)

Cette épreuve comporte 10 pages (page de garde incluse).

Première partie : structure des calculateurs

Question 1 (1 point) : citez les quatre éléments principaux que comprend l'architecture dite de Von Neumann, et faites en une représentation graphique montrant les relations existantes entre elles.

Question 2 (1 point) : supposez l'usage d'un micro-contrôleur fonctionnant à une fréquence de 100 MHz. Mettez en relation les ordres de grandeur pour les temps d'opérations proposés ci-dessous (listés de A à E) avec leurs opérations respectives (listées de 1 à 5).

Opérations :

1. Période d'un cycle processeur
2. Temps d'accès à la mémoire cache
3. Temps d'accès à la mémoire vive (RAM)
4. Temps de commutation de contexte
5. Temps d'accès aux données stockées sur un disque dur

Ordre de grandeur des opérations citées ci-dessus :

- A. 10 ns
- B. 200 ns
- C. 10 ms
- D. 100 μ s
- E. 30 ns

(Exemple de réponse attendue : le temps A correspond à l'opération 1, le temps B à l'opération...)

Deuxième partie : représentation de l'information

Question 1 (0,5 point) : donnez la représentation hexadécimale de la valeur décimale 43981.

Question 2 (1 point) : donnez les représentations hexadécimale et décimale de la valeur binaire 10101010101010.

Question 3 (1 point) : que signifie le terme « endianness » pour un processeur et comment sont rangés en mémoire les mots de 32 bits d'un processeur fonctionnant en big-endian (vous supposerez que chaque adresse de mémoire contient 8 bits d'information) ?

Troisième partie :

programmation, langages et systèmes d'exploitation

Question 1 (2 points) : donnez les noms des quatre principales phases permettant l'obtention d'un programme exécutable à partir d'un code source écrit dans un langage compilé en indiquant le rôle de chacune d'elles et la nature des fichiers qu'elles manipulent.

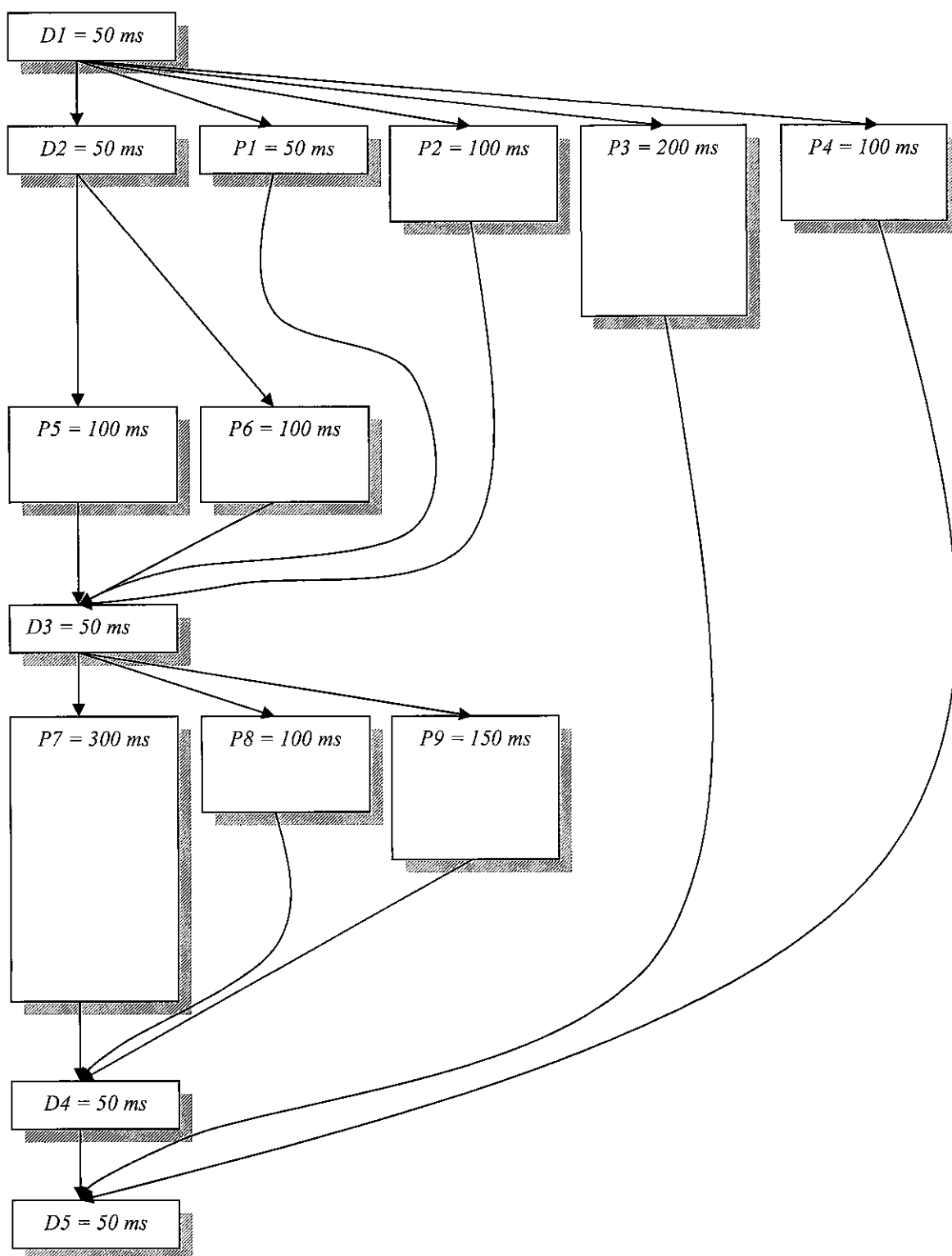
Question 2 (1 point) : vous souhaitez produire un programme exécutable pour une machine qui n'est pas celle sur laquelle vous réalisez la compilation de votre code source (processeurs différents, systèmes d'exploitation différents,...). Parmi les quatre étapes nécessaires à l'obtention de votre programme exécutable que vous avez identifiées à la question précédente, quelle(s) est (sont) celle(s) qui va (vont) poser problème et pourquoi ?

Question 3 (0,5 point) : comment se nomme une chaîne de compilation permettant d'obtenir un exécutable pour un système informatique qui n'est pas celui sur lequel elle est utilisée ?

Question 4 (0,5 point) : en programmation orientée-objet, quelle est la différence entre une classe et un objet ?

Question 5 (0,5 point) : en programmation orientée-objet, qu'est-ce qu'un constructeur ?

Pour les cinq questions qui suivent, vous considèrerez le graphe d'exécution des tâches du programme multi-processus ci-dessous. Sur ce graphe, chaque tâche représentée est associée à son temps d'exécution. Vous considèrerez en outre que l'exécution des tâches les unes par rapport aux autres est fondamentalement asynchrone. Enfin, vous ne prendrez pas en considération un quelconque temps d'ordonnancement, de création/destruction de tâches, ni le surcoût dû aux temps de communication.



Question 6 (0,5 point) : supposez une machine ne disposant que d'un seul processeur mono-cœur et mono-threadé. Sur cette machine, quel est le temps d'exécution du programme dont le graphe d'exécution a été donné précédemment ?

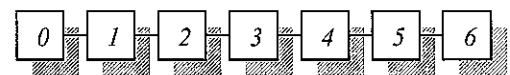
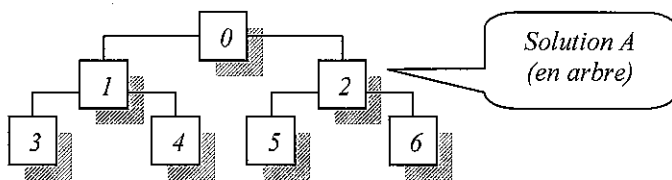
Question 7 (0,5 point) : supposez maintenant une machine disposant de seize processeurs du même type que ceux décrits à la question précédente. Sur cette machine, quel est le temps d'exécution du programme dont le graphe d'exécution a été donné précédemment ?

Question 8 (0,5 point) : est-il possible d'exécuter simultanément toutes les tâches notées P_x ($1 \leq x \leq 9$) sur la machine décrite à la question précédente, et pourquoi ?

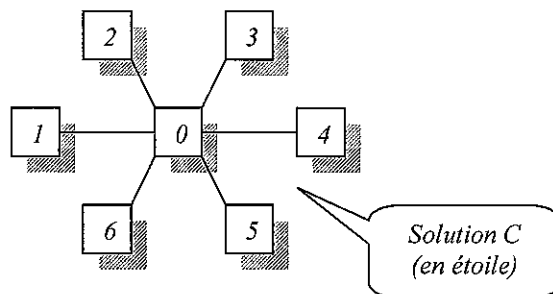
Question 9 (0,5 point) : sur la machine utilisée à la question précédente, quel est le nombre de processeurs nécessaires pour exécuter le programme sans que deux processus s'exécutent simultanément sur le même processeur tout en minimisant : 1. le temps d'inoccupation de chaque processeur et 2. le temps d'exécution global du programme ?

Question 10 (0,5 point) : considérez maintenant une machine équipée du nombre de processeurs requis pour satisfaire les critères de la question précédente. Dans ces conditions, quel est le temps d'exécution du programme sur cette machine ?

Question 11 (1 point) : considérez un ensemble de nœuds de calculs interconnectés par un réseau informatique (cluster). Vous souhaitez distribuer des données possédées par le nœud de calcul de rang 0 vers tous les autres nœuds du cluster. A cette fin, vous avez écrit un programme qui utilise des communications de données de type bloquantes et synchrones. Quelle topologie de réseau parmi les trois présentées ci-dessous (libellées de A à C) est la plus efficace pour minimiser le temps nécessaire à la diffusion de l'information détenue par le nœud 0 vers tous les autres nœuds du réseau dans cette condition (justifiez votre choix) ?



*Solution B
(en ligne)*



Quatrième partie : réseaux de communication

Question 1 (0,5 point) : donnez, sans explication, la liste des couches du modèle OSI (numéros et noms).

Question 2 (1 point) : considérez un réseau local (LAN) de type Ethernet. Donnez l'autre appellation d'une adresse Ethernet, le nombre d'octets qui l'encode, son rôle et sa particularité.

Question 3 (0,5 point) : quel est le rôle d'un concentrateur (hub) dans un réseau local de type Ethernet ?

Question 4 (0,5 point) : quelle est la fonction du protocole ARP (Address Resolution Protocol) ?

Question 5 (0,5 point) : donnez l'adresse de diffusion (broadcast) pour un réseau à adressage IP allant de 192.168.64.0 à 192.168.64.20 .

Question 6 (1 point) : quelle est l'architecture d'un pare-feu (firewall) ? Donnez-en les éléments matériels et logiciels, ainsi que le rôle de cette structure.

Cinquième partie :

tri de données

Introduction

Le tri est une opération fondamentale en informatique. Il consiste à réarranger des données selon un ordre particulier (ex. : nombres dans un ordre croissant, chaînes de caractères dans l'ordre alphabétique,...).

Disposer d'algorithmes efficaces est important car ils sont bien souvent des opérations préalables à d'autres telles que recherches ou fusions. Or, trier des ensembles préalablement à ces activités de recherches ou de fusions permet de rendre ces dernières plus efficaces.

Il existe de nombreux algorithmes de tri. Le choix d'un en particulier doit porter sur les propriétés des données à classer et sur les opérations mises en œuvre par le tri choisi pour obtenir le résultat (*i.e.*, considérer la complexité algorithmique de chaque tri, c'est-à-dire s'intéresser au temps d'exécution de son algorithme en fonction du nombre de données à trier). L'espace mémoire nécessaire à la mise en œuvre d'un tri est un élément à prendre en compte également.

Dans le cas de tris sur architectures à mémoire distribuée, la capacité d'un algorithme de tri à manipuler des données regroupées spatialement par rapport à la topologie du réseau de nœuds de calcul utilisé et d'éviter autant que faire se peut les échanges de données entre ces nœuds sont des paramètres eux aussi essentiels au choix d'un algorithme efficace pour ce type d'architecture.

Les algorithmes de tri parallèle sont des algorithmes représentatifs des benchmarks utilisés pour évaluer les machines hautes-performances actuelles, en particulier multi-cœurs.

Caractéristiques d'un tri

Considérons une liste L d'éléments à trier. Cette liste est composée de n éléments, notés L_1, L_2, \dots, L_n .

Trier L est l'opération permettant un réarrangement de son contenu selon un critère particulier et fixé pour l'opération (par exemple, selon l'ordre croissant de valeurs numériques).

Il y a donc $n!$ manières (permutations) dont les éléments L_1, L_2, \dots, L_n peuvent apparaître dans L . Par conséquent, un algorithme de tri doit être capable de traiter efficacement toutes ces possibilités.

La complexité d'un algorithme de tri mesure le temps d'exécution du tri en fonction du nombre d'éléments à trier (n). Elle mesure en fait essentiellement le nombre de comparaisons nécessaires à l'obtention d'une liste triée, mais les opérations réalisées aux différentes étapes de l'algorithme sont de trois natures :

- des comparaisons entre éléments de L ou avec une valeur pivot de la clé de tri,
- des permutations entre éléments ou avec une valeur pivot,
- des affectations, qui remplacent soit un élément par un autre, un élément par une valeur pivot, ou la valeur pivot par un élément.

La limite inférieure de la complexité pour le tri séquentiel de n éléments est $O(n \log n)$.

Par conséquent, la limite inférieure que nous pourrions attendre d'un tri parallèle utilisant n nœuds de calcul est $O(\log n)$.

On notera enfin que les éléments à trier sont en général rangés dans une structure de données de type tableau et non liste. En effet, le tableau offre la possibilité d'un accès aléatoire aux données qu'il contient alors que la liste impose un accès séquentiel.

Tri à bulle

Soit une liste L d'éléments à trier, liste composée de n éléments, notés L_1, L_2, \dots, L_n .

Soit $M[1], M[2], \dots, M[n]$ les éléments d'un tableau de dimension n représenté en mémoire et initialisé par les éléments de L de la manière suivante : $M[1] = L_1, M[2] = L_2, \dots, M[n] = L_n$.

Le tri à bulle des données contenues dans M fonctionne selon les étapes suivantes :

1. Comparer $M[1]$ et $M[2]$. Les classer dans l'ordre désiré de manière à ce que $M[1] < M[2]$. Comparer ensuite de la même manière $M[2]$ et $M[3]$ afin de les reclasser de manière à ce que $M[2] < M[3]$. Puis, comparer $M[3]$ et $M[4]$ afin de les reclasser de manière à ce que $M[3] < M[4]$. Continuer les comparaisons et les reclassements jusqu'à comparer $M[n-1]$ et $M[n]$ et les classer afin que $M[n-1] < M[n]$. Durant cette étape, la valeur de L la plus « grande » (au regard du critère retenu) rejoint la position n dans le tableau : le plus grand élément remonte ainsi comme une bulle.
2. Cette étape est identique à la précédente, sauf que les comparaisons et les reclassements ne concernent que les $n-1$ premières positions du tableau M (il y a donc une comparaison de moins à réaliser, celle comparant $M[n-1]$ et $M[n]$ puisque la donnée dans $M[n]$ est déjà bien classée. A la fin de cette étape la seconde plus grande valeur de L se retrouve à la position $n-1$ du tableau M .
3. Répéter les opérations de comparaison/reclassement pour les $n-2$ premières positions du tableau M .
4. *etc.*
 $n-1$. Comparer le contenu de $M[1]$ et $M[2]$. Les reclasser si nécessaire.

Question 1 (0,5 point) : transcrivez sous forme de pseudo-code l'algorithme du tri à bulle.

Question 2 (0,5 point) : quelle est la complexité du tri à bulle ?

Question 3 (1 point) : quelle(s) optimisation(s) de l'algorithme pourriez-vous suggérer pour améliorer son efficacité tout en conservant son principe de fonctionnement ?

Question 4 (0,5 point) : quel obstacle majeur voyez-vous à la parallélisation de ce tri ?

Question 5 (1 point) : proposez une solution pour paralléliser malgré tout ce tri sur une architecture parallèle à mémoire partagée (donnez-en seulement le principe).