PIN203

T.D. numéro 1

Codage des informations physiques

1 Introduction

Cette partie a pour but de sensibiliser l'étudiant au problème de l'espace nécessaire au codage des informations physique. En effet, bien que l'ordinateur soit un allié puissant pour le physicien, il ne permet tout de même pas de traiter tous les problèmes et surtout de les traiter n'importe comment.
La première difficulté réside souvent dans la gestion de la place mémoire occupée pour le traitement du problème. Avant de démarrer le codage d'un problème de physique, il faut se poser la question de savoir combien son codage et son traitement va prendre de place en mémoire.

2 Combien de place

Exercice 1.1

Sachant qu'un caractère se code sur un octet, le texte du dernier «Harry Potter » (l'Ordre du Phoenix, version anglaise) qui compte 766 pages de 37 lignes de 62 caractères en moyenne peut-il être stocké sur une disquette de 1,44 Mo ? Qu'elle remarque peut-on faire quant à la dénomination disquette 1,44Mo ?

Exercice 1.2

Le Canon PowerShot A80 est un appareil photo numérique permettant d'obtenir des images de résolution maximale 2272×1704 pixels. Combien de photos haute résolution codées en 32 bits prises avec cet appareil pourrait-on stocker dans une carte mémoire de 64Mo si aucun algorithme de compression n'était utilisé pour diminuer la taille mémoire occupée par l'image ?

Exercice 1.3

La gamme des fréquences audibles par l'homme est en moyenne 20 Hz-20 kHz. On montre en théorie de traitement du signal que la fréquence minimale d'échantillonnage d'un signal doit être supérieure à deux fois la fréquence haute du signal analogique si on ne veut pas perdre d'information (théorème de Shannon). C'est pour celà que l'on échantillonne le signal analogique à 44100Hz lors du codage d'un CD Audio. En simplifiant, on mesure la hauteur du signal sonore 44100 fois par seconde, afin d'avoir une restitution HiFi, cette valeur est codée sur 16 bits.
1. Le morceau «Never Get Gold » de l'album « Reality » de David Bowie dure 4 min 27 s. Quelle est sa taille en Mo sur le disque ? Attention, réfléchissez bien, il y a un piège.
2. Quelle est la durée maximale en minutes des morceaux que l'on peut stocker sur un CD de 650 Mo ? En réalité cette valeur est d'environ 74 minutes ce qui est approximativement la durée de la IX^{e} symphonie de Beethoven, c'est une des raisons qui a déterminé la taille des CDs standards. Quelle est alors la capacité du CDs suivant cette nouvelle information ? D'où peut provenir cette différence ?

Exercice 1.4

L'Université Bordeaux I regroupe environ 11500 étudiants. Un numéro unique étant attribué à chaque étudiant :

1. sur combien de bits au minimum faut-il coder ce numéro si on ne veut pas avoir de doublons dans le fichier informatique des étudiants ?
2. En réalité, ce numéro est codé dans l'ordinateur sous la forme d'un nombre entier pair d'octets. Combien d'octets occupe se numéro ?
3. La quantité d'information que souhaite conserver l'Université sur chaque étudiant (nom, prénom, date de naissance, cursus) représente environ 1 ko quel support vous semble le mieux adapté au stockage de cette information ?

Exercice 1.5

Etude de la quantité d'information microscopique nécessaire à la description d'un cm^3 de gaz parfait.
– On supposera que le gaz est pris dans les conditions standard de température et de pression.
– On suppose qu'il n'y a pas d'interaction entre molécules (gaz parfait)
– Chaque molécule est décrite par ses trois coordonnées de position et ses trois coordonnées d'impulsion.
– On supposera que l'on code ces grandeurs physiques comme des réels double précision (8 octets).
1. Pensez vous à priori que la description complète du système soit accessible au calcul informatique.
2. Calculer le nombre de molécules présentes dans un cm^3 de gaz.
3. Calculer le nombre de réels nécessaires à décrire le système à un instant t.
4. Calculer le nombre de Go nécessaires au stockage de cette information.
5. Un CD-ROM contient environ 700 Mo, combien de CD-ROMs doit on utiliser pour stocker cette information.
6. Le volume d'une boîte de CD-ROM est d'environ 100 $cm3$, quel volume occupent les CDs ainsi stockés. Un camion contient un volume de $40m^3$ combien de camions sont nécessaires au transport de l'information.
7. En supposant que l'on puisse stocker cette information dans une mémoire vive au rythme de 1 octet·ns^{-1}. Combien de temps faudrait-il pour stocker l'information de l'état microscopique d'un litre d'air à un instant t .
8. Conclure.

T.P. numéro 1

Introduction au C et à Unix

3 Découverte de l'environnement Unix

3.1 Principales commandes Unix :

Une fois connecté sur une des machines Linux de la salle de calcul, ouvrir un fenêtre contenant un terminal xterm dans laquelle on peut taper du texte. Ce texte est analysé par un interpréteur de commandes (dans notre cas bash) qui exécute alors les programmes associés aux commandes.
Ces commandes sont principalement des commandes de gestion des fichiers et de l'arborescence du disque sur lequel sont stockés ces fichiers. La ligne de commande permet aussi d'exécuter des programmes plus évolués comme par exemple mozilla (navigateur internet) ou gedit (éditeur de texte ou nedit) ou encore des programmes que l'on aura écrit nous-mêmes. La plupart des commandes Unix sont documentées et on peut obtenir de l'aide en ligne en tapant par exemple

```
man ls
```

pour la commande ls. Voici une liste des principales commandes :

1. manipulation des fichiers :
– touch <file> : crée un fichier vide nommé file ;
– ls -al : affiche la liste de fichiers présents dans le répertoire courant ;
– cp <file1> <file2> : crée la copie file2 du fichier file1 ;
– mv <file1> <file2> : renomme le fichier file en file2 ;
– rm <file> : supprime le fichier file (attention la perte du fichier est définitive ! ! !)
– more <file> : fait défiler à l'écran le contenu du fichier file ;

2. manipulation des répertoires :
– pwd : renvoie le nom du répertoire courant ;
– mkdir <dir> : crée le répertoire dir ;
– rmdir <dir> ou rm -R <dir> : détruit le répertoire dir ;
– cd <dir> : va dans le répertoire dir (cd.. pour revenir au niveau précédent)
– cd (tout seul) ou cd ~ : vous ramène dans votre répertoire principal

Exercice 1.6
Dans votre répertoire principal, créez un répertoire nommé TP1. Depuis Mozilla (menu File puis Save as ...) sauvegardez le présent document sous le nom TP1.pdf dans votre répertoire principal.
Copiez, à l'aide des commandes citées plus haut, le fichier TP1.pdf dans le répertoire TP1. Allez dans TP1 et vérifiez que le fichier est bien là. Effacez ensuite la copie restée dans votre répertoire principal et vérifiez qu'il n'y est plus.
Préparer une arborescence de répertoire pour l'ensemble des TP : TP2,etc

3.2 Liens utiles
– l'interpréteur de commande ou shell bash
– un index des commandes Unix
– l'éditeur de texte gedit.

4 Découverte du language C
Le but de ce TD-TP est de se familiariser avec les premiers éléments de la syntaxe C a travers l'étude d'un petit programme. Un précieux soutien peut être trouvé pour les problèmes concernant le langage C. Les programmes que l'on va développer dans ces TP seront écrits dans un fichier texte grâce à l'éditeur de texte gedit (ou nedit ou emacs). Une fois le programme écrit, on le « compile » en utilisant un programme déjà existant (gcc dans notre cas) qui transforme le texte du programme en instructions directement exécutables par le processeur de la machine.

Exercice 1.7 : quelques règles de base
1. Sauvegarder le programme lineaire.c dans le répertoire TP1 déjà créé
2. Éditez-le avec gedit en lançant la commande gedit linaire.c depuis la ligne de commande d'une fenêtre terminal.
3. Que fait ce programme?
4. Faire un listing (ls) du répertoire TP1 : il ne doit y avoir que TP1.pdf et lineaire.c
5. Compilez -le avec la commande gcc lineaire.c
6. Refaire un ls ; que constatez-vous ?

7. Executer le programme avec la commande : ./a.out
En fait cette méthode de compilation écrasera les fichiers exécutables précédemment compilés. Il est donc plus sur de donner un nom explicite à l'exécutable.
8. Effacer le fichier a.out (commande rm) et faire un ls : que se passe-t-il ?
9.Compilez maintenant lineaire.c avec la commande gcc lineaire.c –o lineaire
Puis refaire un ls.
10. Exécutez ce programme en tapant la commande: ./lineaire
Et constater que vous avez la même chose qu'en 7).
11. Pour s'assurer qu'aucune possibilité d'erreur de syntaxe n'existe dans le code celui-ci doit être compilé avec l'option –Wall (all warnings turned on) et donc vous devez maintenant compiler avec la commande : gcc lineaire.c –o lineaire –Wall

Désormais pour tous les TP vous compilerez obligatoirement avec la forme de commande pécedente : gcc nomprog.c –o nomprog –Wall.
Désormais un programme ne sera jugé acceptable que s'il est compilé sans avertissement.

12. Détaillez les rôles des trois lignes finales de lineaire.c commençant par printf en commentant dans le programme même
13. Ce programme fait-il bien ce qu'il prétend faire ? Détaillez sur votre cahier de manip, l'algorithme proposé. Le programme peut-il traiter sans problème tous les jeux de valeurs a1, b1, c1, a2, b2, c2 ?

Exercice 1.8

En vous inspirant du code source lineaire.c, écrire un programme nommé lineaire2.c qui :
1. Affiche à l'écran la preuve que la solution trouvée répond bien au problème. (par exemple en affichant a1*x+b1*y = c1 et a2*x+b2*y = c2 avec les valeurs des différentes variables)

2. Compléter le programme pour qu'il traite proprement les cas non traités par lineaire.c.
On utilisera pour cela une instruction conditionnelle. On nommera lineaire3.c la nouvelle version de programme
3. Autoriser l'entrée au clavier des valeurs des 6 paramètres a1, b1, c1, a2, b2, c2. On utilisera la fonction scanf, dont on aura vu la description du fonctionnement en faisant man scanf.
On fera particulièrement attention au type des arguments de cette fonction.
On fera attention à se conformer au style standard du codage.
On vérifiera le bon fonctionnement du programme sur deux exemples bien choisis.

Remarque : Lors de l'exécution, on peut rediriger la sortie écran dans un fichier en utilisant l'opérateur de redirection > comme par l'exemple :

lineaire2 > resultat

qui redirige la sortie écran vers le fichier *résultat* et permet d'archiver le résultat dans un répertoire.

Remarques :

Il est rappelé que résultats et programmes sont à imprimer et coller soigneusement dans le cahier de manip, qui fera l'objet d'une évaluation dans le cadre de la note de contrôle continu. Chaque programme devra avoir été commenté dans son fichier source.
A la fin de séance une copie des programmes écrits par le groupe sera donnée à l'enseignant. Utiliser pour cela la commande : a2ps –Pimp0ati1 fichier.c pour l'imprimante de l'atelier1 (..ati2 pour atelier2, etc) ou imprimer directement à partir de l'éditeur de texte.