

feuille de td 3

1 Exercice *Base et conversion*

Q.1. Compléter le tableau suivant

décimal	octal	hexadécimal	binaire
123			
	123		
		D3	
			1001101

2 Exercice *Octal et droits*

Dans les systèmes Linux, les fichiers possèdent des droits d'accès. Ces droits sont les suivants :

nom	signification
r	le droit de lire le fichier
w	le droit d'écrire le fichier
x	le droit d'exécuter le fichier

On peut préciser individuellement chacun des droits pour

- le propriétaire du fichier.
- les utilisateurs étant dans le même groupe.
- tous les autres.

la commande `ls -l` permet de voir (entre autre) les droits.

```
-rw---xr-- 1 weinberl enseign 0 oct  2 21:53 tata
-----rw- 1 weinberl enseign 0 oct  2 21:53 tete
-r---w---- 1 weinberl enseign 0 oct  2 21:53 titi
----rwx--- 1 weinberl enseign 0 oct  2 21:53 toto
-rw-r--r-- 1 weinberl enseign 0 oct  2 21:53 tutu
```

Q.2. sachant que les droits propriétaires sont suivis de ceux du groupe et qu'enfin il y a les droit des autres, dire quels sont les droits de chacun sur les fichiers suivants.

la commande `chmod` permet de changer les droits. On peut l'utiliser de plusieurs manière.

Notamment en précisant les droits à l'aide de 3 chiffres en octal . par exemple le droit en octal du fichier `tata` est 614

Q.3. Quel sont les droits en octal des autres fichiers ?

Q.4. Quel sera l'effet de la commande suivante :

```
chmod 660 titi
```

3 Exercice *Capacité de stockage*

Q.5. Un fichier contenant un texte français de 25 lignes de 80 colonnes, nécessite une capacité de stockage de

- une ou deux dizaine octets
- un ou deux kio
- un ou deux Mio
- un ou deux Gio

Q.6. Un fichier contenant 5 minutes de musique au format "CD" occupe

- moins d'une dizaine d'octets
- moins d'une centaine de kio

- moins d'une centaine de Mio
- moins d'un Gio

Q.7. Un fichier source Pascal donnant la solution à l'exercice 1 de la feuille sur l'unité carte occupe sur le disque :

- une ou deux dizaine d'octets
- un ou deux kio
- un ou deux Mio
- un ou deux Gio

(et le fichier executable?)

Q.8. le codage de votre adresse électronique occupe :

- une ou deux dizaine d’octets
- un ou deux kio
- un ou deux Mio
- un ou deux Gio

Q.9. un fichier contenant le texte de *Germinal* occupe sur le disque (indication : dans ce texte il y a environ 166000 mots et que la longueur moyenne des mots est à peu près de 5 caractères, (la plupart des caractères sont codés sur un octet))

- une ou deux dizaine d’octets
- un ou deux kio
- un ou deux Mio

- un ou deux Gio

Q.10. Un fichier contient l’image d’un logo. Pour coder cette image on a utilisé un tableau rectangulaire de points. La largeur de ce tableau est 128 et la hauteur 60. Chaque point de ce tableau possède une couleur, et il faut trois octets pour représenter une couleur. Ce fichier occupe sur le disque :

- un ou deux dizaine d’octets
- une ou deux dizaine de kio
- une ou deux dizaine de Mio
- une ou deux dizaine de Gio

4 Exercice *Codage base 64*

Certains protocoles de communication imposent que le contenu des message soit du texte. Pour contourner ce problème, et afin de pouvoir transmettre des fichiers binaires en utilisant ce protocole, a été inventé le codage dit "Base64". Il s’agit d’utiliser les majuscules, les minuscules, les chiffres et deux autres symboles (+/), et d’un symbole supplémentaire utilisé uniquement à la fin = dont le rôle sera expliqué plus loin. On dispose 64 symboles distincts.

Q.11. Combien de bits sont nécessaires si on souhaite coder ces symboles en binaire ?

On donne le codage suivant :

000000	A	010000	Q	100000	g	110000	w
000001	B	010001	R	100001	h	110001	x
000010	C	010010	S	100010	i	110010	y
000011	D	010011	T	100011	j	110011	z
000100	E	010100	U	100100	k	110100	0
000101	F	010101	V	100101	l	110101	1
000110	G	010110	W	100110	m	110110	2
000111	H	010111	X	100111	n	110111	3
001000	I	011000	Y	101000	o	111000	4
001001	J	011001	Z	101001	p	111001	5
001010	K	011010	a	101010	q	111010	6
001011	L	011011	b	101011	r	111011	7
001100	M	011100	c	101100	s	111100	8
001101	N	011101	d	101101	t	111101	9
001110	O	011110	e	101110	u	111110	+
001111	P	011111	f	101111	v	111111	/

Q.12. En remarquant que $6 \times 4 = 3 \times 8$, expliquer comment on peut remplacer un groupe de 3 octets par 4 symboles du code Base64

Biensûr tout fichier n’a pas nécessairement sa longueur qui est multiple de 3 !

- S’il reste un octet à coder à la fin, on ajoute à droite 4 bit 0 et on signale à la fin cet ajout par deux signes ==
- s’il reste deux octets à coder à la fin, on ajoute à droite deux bit 0 et on signale à la fin cet ajout par un signe =

Q.13. Voici le contenu d’un fichier listé en hexadécimal. Convertir en base64 ce fichier.

55 53 49 20 65 6e 20 73 27 61 6d 75 73 61 6e 74