

Durée : 1h30 — Documents autorisés

■ ■ ■ Unix — (2 points)

- 1– a. Pour un processeur 64 bits sous Linux, la taille d'une page est de 4096 octets : de combien de pages dispose-t-on ? (1pt)
2pts
- b. Est-ce que le décalage d'un octet de l'adresse de départ d'un programme en mémoire centrale est grave lors de son exécution par le processeur ? Expliquez ce qui arrive. (1pt)
Vous pourrez vous inspirer du processeur 6502.

■ ■ ■ Programmation Python — (14 points)

2–
14pts

Depuis Wikipedia

Le « bit de parité » est un bit ajouté à une séquence binaire, comme **code de détection d'erreur**. Il est utilisé sur la plus petite unité de communication : 8 bits, soit un octet.

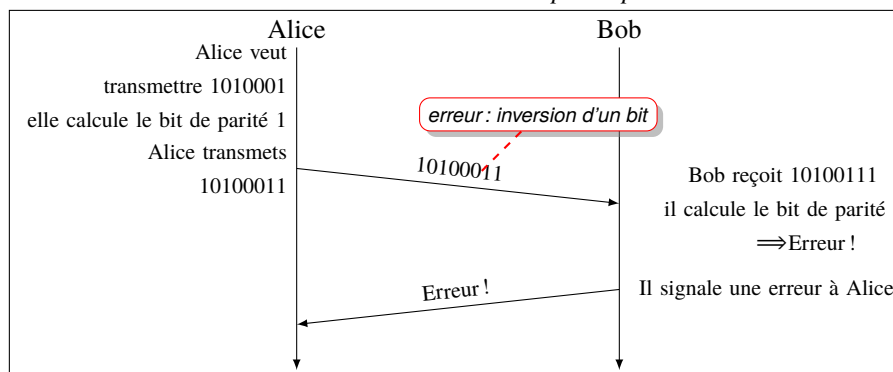
Il y a deux variantes :

- ▷ la **parité paire** : si le nombre de bits à 1 dans les données est impaire on ajoute un bit à 1 comme bit de parité, pour que le nombre total de bit à 1 soit **paire**. Sinon, on ajoute un bit à zéro.
- ▷ la **parité impaire** : si le nombre de bits à 1 dans les données est paire on ajoute un bit à 1 comme bit de parité, pour que le nombre total de bit à 1 soit **impaire**. Sinon, on ajoute un bit à zéro.

Exemple :

7 bits données	nombre de bits à 1	8 bits incluant la parité	
		paire	impaire
0000000	0	00000000	00000001
1010001	3	10100011	10100010
1101001	4	11010010	11010011
1111111	7	11111111	11111110

Détection d'erreur lors d'une communication avec bit de « parité paire » :



Une erreur correspond à l'inversion de la valeur d'un bit.

Questions :

Pour répondre aux différentes questions, on utilisera la méthode de la parité paire.

- a. Est-ce que la méthode du « bit de parité » permet de détecter plus d'une erreur ? Pourquoi ? (1pt)
- b. Il existe deux méthodes pour déterminer la valeur du bit de parité : (4pts)
- ◇ méthode A : compter le nombre de bits à 1 et décider de la valeur du bit de parité ;
 - ◇ méthode B : utiliser l'opération `xor` :
- Soit la séquence de bits 1001100, on calcule $1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 1$ où \oplus est l'opération « xor », ce qui donne le bit de parité à 1 et la séquence finale à transmettre 10011001.

Sous Python, l'opération « xor » est notée `^` :

```
xterm
>>> 1^1
0
```

Écrire deux fonctions Python, une pour chaque méthode :

- ◇ prenant en entrée les 7bits sous forme de chaîne de caractères ;
- ◇ retournant le bit de parité "1" ou "0"

Exemple :

```
xterm
>>> calcul_parite_paire("1101110")
"1"
```

- c. Écrire une fonction Python qui **vérifie** le bit de parité paire : (2pts)
- ◇ prenant en entrée la représentation binaire de 8bits sous forme de chaîne de caractères d'une séquence incluant le bit de parité paire ;
 - ◇ retournant True ou False suivant que le bit de parité est correcte ou non.

Exemple :

```
xterm
>>> verifie_parite_paire("11011101")
True
```

- d. On veut traiter une chaîne d'octets quelconque sans perte d'information, chaque caractère va être interprété sous forme d'octet : il va fournir 8 bits. (4pts)

Méthode :

- ◇ on va prendre 7 bits et calculer le bit de parité paire à l'aide d'une des deux fonctions précédentes ;
- ◇ on va ajouter les 7 bits et le bit de parité dans la séquence résultat ;
- ◇ le bit restant va être concaténé aux 8 bits du caractère suivant ;
- ◇ on recommence jusqu'à traiter tous les caractères d'entrée.

Dans le cas où on aura pas assez de 7 bits, on complètera par des 0.

Exemple :

- ▷ soit la chaîne `b'toto'` en entrée ;
- ▷ on obtient `['01110100', '01101111', '01110100', '01101111']` pour chaque caractère ;
- ▷ ce qui donne en sortie : `'01110100', '00110110', '11011101', '10001101'` et `'11110000'`

Écrire la fonction Python réalisant ce travail et qui construit une chaîne de bits en sortie. Write Do

- e. Écrire un programme Python qui lit le contenu d'un fichier et construit un fichier en sortie avec le bit correct de parité paire pour chaque groupe de 7bits. (3pts)

Le fichier de sortie ne doit contenir que des « caractère octet », pas des séquences de bit.

■■■ Programmation réseaux — (4 points)

3– On veut concevoir un **serveur TCP** qui reçoit une séquence binaire de 8bits sous forme de chaîne de caractère et renvoie si la valeur du bit de parité paire est correcte ou non. **4pts**

- a. Quelles informations réseaux faut-il « partager » avec le client qui l'utilisera ? (1pt)
- b. Écrire un programme Python TCP serveur qui **pour chaque connexion reçue**, réalise la vérification du bit de parité et qui retourne la chaîne "Ok" ou "NOK" suivant si le bit de parité est correct ou non. (3pts)
- Vous disposerez de la fonction décrite en 1–c.