

Duration : 1h30 — All documents authorized

■ ■ ■ Unix — (2 points)

- 1– a. For a 64bits computer under Linux, the size of page is 4096 bytes : how much pages do we get ? (1pt)
- 2pts b. Does it matter if the starting address of a program in the central memory is shifted from one byte before its execution by the processor ? (1pt)
- Explain what happens.
You could think about the 6502 processor.

■ ■ ■ Python Programming — (14 points)

2–
14pts

Depuis Wikipedia

The «*parity bit*» is a bit added to a binary sequence, as the simplest form of **error detecting code**. It is applied to the smallest unit of communication : 8 bits, as one byte.

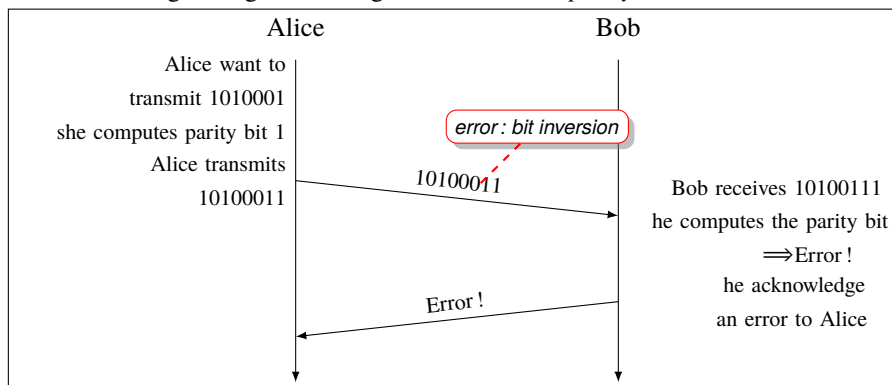
There are two variants :

- ▷ the **even parity bit** : the occurrences of bits whose value is 1 are counted. If that count is odd, the parity bit value is set to 1, making the total count of occurrences of 1s in the whole set (including the parity bit) an even number. If not the parity bit is set to 0.
- ▷ the **odd parity** : the occurrences of bits whose value is 1 are counted. If that count is even, the parity bit value is set to 1, making the total count of occurrences of 1s in the whole set (including the parity bit) an odd number. If not the parity bit is set to 0.

Example :

7 bits of data	(count of 1-bits)	8 bits including parity	
		even	odd
0000000	0	00000000	00000001
1010001	3	10100011	10100010
1101001	4	11010010	11010011
1111111	7	11111111	11111110

Error detection occurring during an exchange with the «*even parity*» :



An error is the inversion of one bit.

Questions :

To answer the questions, you will consider the «even parity» .

- a. Does the method of «parity bit» allow to detect more than one error ? Why ? (1pt)
- b. There are two methods to compute the value of the parity bit : (4pts)

- ◇ method A : counting the number of bits with the value 1 and determining the value of the parity bit ;
- ◇ method B : using the xor operation :
With the binary sequence 1001100, we compute $1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 1$ where \oplus is the «xor» operation, that gives the value of the parity bit of 1 and the final sequence to transmit 10011001.

In Python, the «xor» operation is expressed by ^ :

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

Write two Python functions, one for each method :

- ◇ taking in input the 7bits in a string ;
- ◇ returning the parity bit "1" or "0".

Example :

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

- c. Write a Python function that **checks** the even parity bit : (2pts)
- ◇ taking in input the binary representation of 8 bits as a binary sequence including even parity bit ;
 - ◇ returning True ou False as the parity bit is correct or not.

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

- d. We want to manage a string of bytes without loosing any information : each character will be dealed as a byte : it will yield 8 bits. (4pts)

Method :

- ◇ we will take 7bits and compute the even parity bit with the help of one of the previous function.
- ◇ we will add the 7bits and the parity bit to the final result ;
- ◇ the bit that remains will be concatenated to the 8bits from the next character ;
- ◇ we will restart until the last input character has been dealed with.

In the case where not enough bits are available to get 7bits, we will use as much of 0 as needed.

Example :

- ▷ with the string b'toto' in input ;
- ▷ we get ['01110100', '01101111', '01110100', '01101111'] for each character ;
- ▷ that will output : '01110100', '00110110', '11011101', '10001101' and '11110000'

Write the python function performing this work and creating the output string of bits.

- e. Write a Python program that reads the content of a file and write a file in output with the correct even parity bit for each group of 7bits. (3pts)

The output file must contain only bytes, no binary sequences.

■■■■ Programmation réseaux — (4 points)

3– We want to implement a **TCP server** that receives a binary sequence of 8bits as a string and responds if the value of the even parity bit is correct or not. (4pts)

- a. What network information must we share with client who will use this server ? (1pt)
- b. Write a Python program implementing the TCP server that, for each received connection, checks the validity of the even parity bit and responds by the string Ok or NOk according to the correctness of this bit. (3pts)

You will use the function described in 1–c as if it already exists.