

## EXERCICE 1 (4 points)

### Thèmes abordés : systèmes d'exploitation linux

L'entreprise capNSI gère les contrats de ses clients en créant pour chacun d'eux un sous-dossier dans le dossier Contrats sur leur ordinateur central. Le système d'exploitation de cet ordinateur est une distribution linux. Quelques commandes de bases pour ce système d'exploitation sont rappelées en annexe 1 en fin de sujet.

Dans la console représentée sur la figure ci-dessous, on peut visualiser les répertoires (ou dossiers) à la racine de l'ordinateur central avec l'instruction `ls` :

```
gestion@capNSI-ordinateur_central:~$ ls
Bureau      Documents  Modèles    Public
Téléchargements  Contrats   Images     Musique
Vidéos
```

1.

- Donner le nom de l'utilisateur et le nom de l'ordinateur correspondant à la capture d'écran précédente.
- Ecrire les instructions permettant d'afficher la liste des dossiers clients du répertoire `Contrats` en partant de la situation ci-dessous :

```
gestion@capNSI-ordinateur_central:~$
```

Après une campagne de démarchage, l'entreprise a gagné un nouveau client, Monsieur Alan Turing. Elle souhaite lui créer un sous-dossier nommé `TURING_Alan` dans le dossier `Contrats`. De plus, elle souhaite attribuer tous les droits à l'utilisateur et au groupe et seulement la permission en lecture pour tous les autres utilisateurs. La commande `chmod` permet de le faire.

2.

- Ecrire les instructions permettant de créer le sous-dossier `TURING_Alan` à partir du répertoire racine.
- Ecrire l'instruction permettant d'attribuer les bons droits au sous-dossier `TURING_Alan`.

En Python, le module `os` permet d'interagir avec le système d'exploitation. Il permet de gérer l'arborescence des fichiers, des dossiers, de fournir des informations sur le système d'exploitation. Par exemple, le code de la page suivante, exécuté dans la console, permet de créer le sous-dossier `TURING_Alan` précédent :

```
>>> import os
>>> os.mkdir("Contrats/TURING_Alan")
>>> os.chmod("Contrats/TURING_Alan", 774)
```

L'entreprise dispose d'un tableau de nouveaux clients :

```
tab_clients = [  
    ('LOVELACE', 'Ada'),  
    ('BOOLE', 'George'),  
    ('VONNEUMANN', 'John'),  
    ('SHANNON', 'Claude'),  
    ('KNUTH', 'Donald')  
]
```

Elle souhaite automatiser le formatage des tableaux des nouveaux clients. Elle souhaite également automatiser la création et l'attribution des droits des dossiers portant les noms des nouveaux clients.

3. Ecrire une fonction `formatage(tab)` qui prend en paramètre un tableau de tuplets (`Nom`, `Prenom`) des nouveaux clients et renvoie un tableau de chaînes de caractères. Par exemple, `formatage(tab_clients)` renvoie `['LOVELACE_Ada', 'BOOLE_George', 'VONNEUMANN_John', 'SHANNON_Claude', 'KNUTH_Donald']`
4. Ecrire une fonction `creation_dossiers(tab)` qui prend en paramètre un tableau de chaînes de caractères et qui crée et modifie les droits des dossiers au nom de ces chaînes de caractères avec les mêmes droits que le sous-dossier `TURING_Alan`.

## EXERCICE 2 (4 points)

Cet exercice porte sur la gestion des processus par les systèmes d'exploitation et sur les opérateurs booléens.

### Partie A

Cette partie est un questionnaire à choix multiples (QCM).

Pour chacune des questions, une seule des quatre réponses est exacte. Le candidat indiquera sur sa copie le numéro de la question et la lettre correspondant à la réponse exacte.

Aucune justification n'est demandée. Une réponse fautive ou une absence de réponse n'enlève aucun point.

1. Parmi les commandes ci-dessous, laquelle permet d'afficher les processus en cours d'exécution ?
  - a. `dir`
  - b. `ps`
  - c. `man`
  - d. `ls`
  
2. Quelle abréviation désigne l'identifiant d'un processus dans un système d'exploitation de type UNIX ?
  - a. PIX
  - b. SIG
  - c. PID
  - d. SID
  
3. Comment s'appelle la gestion du partage du processeur entre différents processus ?
  - a. L'interblocage
  - b. L'ordonnancement
  - c. La planification
  - d. La priorisation
  
4. Quelle commande permet d'interrompre un processus dans un système d'exploitation de type UNIX ?
  - a. `stop`
  - b. `interrupt`
  - c. `end`
  - d. `kill`

## Partie B

1. Un processeur choisit à chaque cycle d'exécution le processus qui doit être exécuté. Le tableau ci-dessous donne pour trois processus P1, P2, P3 :

- la durée d'exécution (en nombre de cycles),
- l'instant d'arrivée sur le processeur (exprimé en nombre de cycles à partir de 0),
- le numéro de priorité.

Le numéro de priorité est d'autant plus petit que la priorité est grande. On suppose qu'à chaque instant, c'est le processus qui a le plus petit numéro de priorité qui est exécuté, ce qui peut provoquer la suspension d'un autre processus, lequel reprendra lorsqu'il sera le plus prioritaire.

Processus	Durée d'exécution	Instant d'arrivée	Numéro de priorité
P1 3 3 1			
P2 3 2 2			
P3 4 0 3			

Reproduire le tableau ci-dessous sur la copie et indiquer dans chacune des cases le processus exécuté à chaque cycle.

P3										
0	1	2	3	4	5	6	7	8	9	10

2. On suppose maintenant que les trois processus précédents s'exécutent et utilisent une ou plusieurs ressources parmi R1, R2 et R3.

Parmi les scénarios suivants, lequel provoque un interblocage ? Justifier.

Scénario 1	Scénario 2	Scénario 3
P1 acquiert R1	P1 acquiert R1	P1 acquiert R1
P2 acquiert R2	P2 acquiert R3	P2 acquiert R2
P3 attend R1	P3 acquiert R2	P3 attend R2
P2 libère R2	P1 attend R2	P1 attend R2
P2 attend R1	P2 libère R3	P2 libère R2
P1 libère R1	P3 attend R1	P3 acquiert R2

## Partie C

Dans cette partie, pour une meilleure lisibilité, des espaces sont placées dans les écritures binaires des nombres. Il ne faut pas les prendre en compte dans les calculs.

Pour chiffrer un message, une méthode, dite du masque jetable, consiste à le combiner avec une chaîne de caractères de longueur comparable.

Une implémentation possible utilise l'opérateur XOR (ou exclusif) dont voici la table de vérité :

a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0

Dans la suite, les nombres écrits en binaire seront précédés du préfixe 0b.

1. Pour chiffrer un message, on convertit chacun de ses caractères en binaire (à l'aide du format Unicode), et on réalise l'opération XOR bit à bit avec la clé.

Après conversion en binaire, et avant que l'opération XOR bit à bit avec la clé n'ait été effectuée, Alice obtient le message suivant :

$m = 0b\ 0110\ 0011\ 0100\ 0110$

- a. Le message  $m$  correspond à deux caractères codés chacun sur 8 bits : déterminer quels sont ces caractères. On fournit pour cela la table ci-dessous qui associe à l'écriture hexadécimale d'un octet le caractère correspondant (figure 2). Exemple de lecture : le caractère correspondant à l'octet codé 4A en hexadécimal est la lettre J.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	space	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Figure 2

- b.** Pour chiffrer le message d'Alice, on réalise l'opération XOR bit à bit avec la clé suivante :
- $$k = 0b\ 1110\ 1110\ 1111\ 0000$$

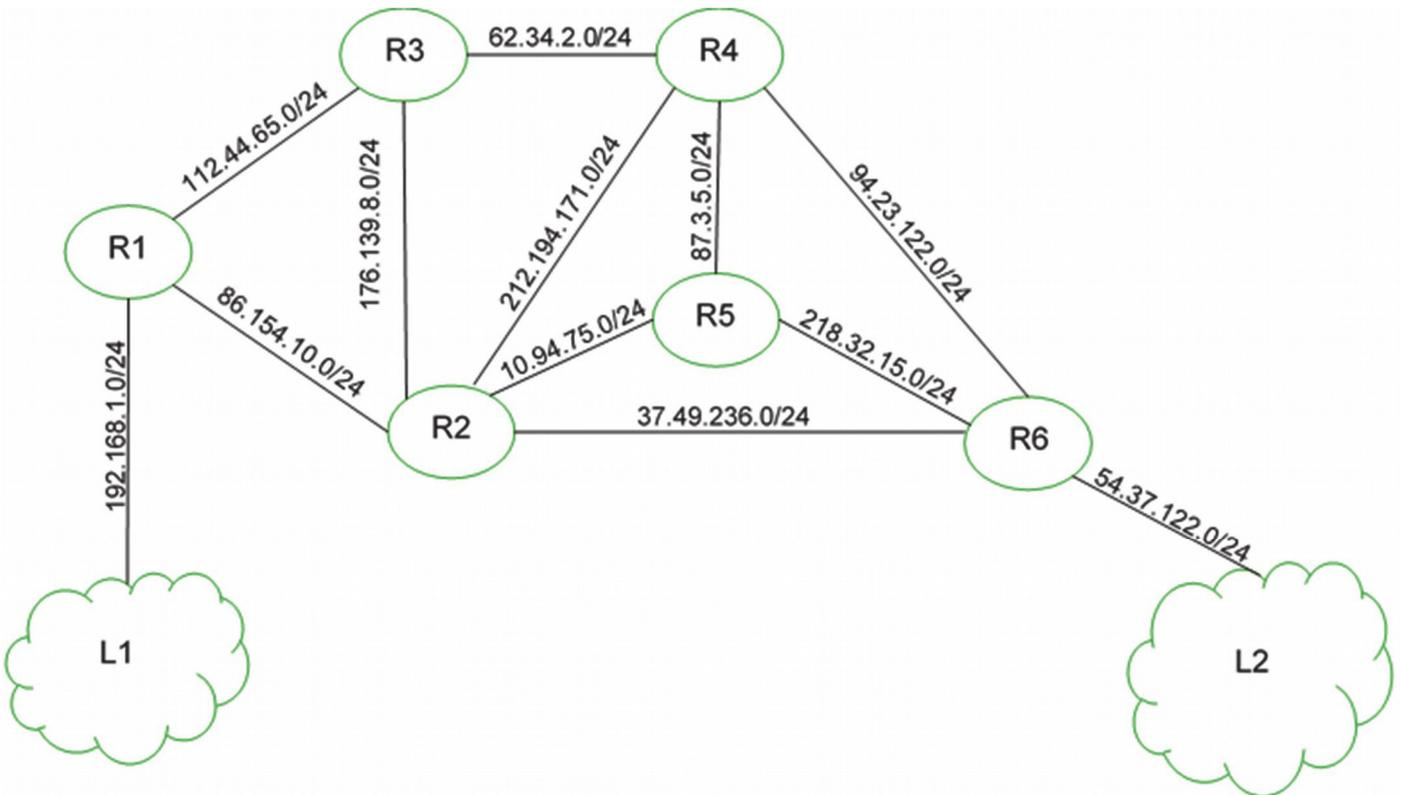
Donner l'écriture binaire du message obtenu.

- 2.**
- a.** Dresser la table de vérité de l'expression booléenne suivante :
- $$(a \text{ XOR } b) \text{ XOR } b$$
- b.** Bob connaît la chaîne de caractères utilisée par Alice pour chiffrer le message. Quelle opération doit-il réaliser pour déchiffrer son message ?

### EXERCICE 3 (4 points)

Cet exercice porte sur les réseaux et les protocoles de routage.

On représente ci-dessous un réseau dans lequel R1, R2, R3, R4, R5 et R6 sont des routeurs. Le réseau local L1 est relié au routeur R1 et le réseau local L2 au routeur R6.



#### Rappels et notations

Dans cet exercice, les adresses IP sont composées de 4 octets, soit 32 bits. Elles sont notées X1.X2.X3.X4, où X1, X2, X3 et X4 sont les valeurs des 4 octets, convertis en notation décimale.

La notation X1.X2.X3.X4/n signifie que les n premiers bits de poids forts de l'adresse IP représentent la partie « réseau », les bits suivants représentent la partie « hôte ».

Toutes les adresses des hôtes connectés à un réseau local ont la même partie réseau et peuvent donc communiquer directement. L'adresse IP dont tous les bits de la partie « hôte » sont à 0 est appelée « adresse du réseau ».

On donne également des extraits de la table de routage des routeurs R1 à R5 dans le tableau suivant :

Routeur	Réseau destinataire	Passerelle	Interface
R1	54.37.122.0/24	86.154.10.1	86.154.10.56
R2	54.37.122.0/24	37.49.236.22	37.49.236.23
R3	54.37.122.0/24	62.34.2.8	62.34.2.9
R4	54.37.122.0/24	94.23.122.10	94.23.122.11
R5	54.37.122.0/24	218.32.15.1	218.32.15.2

1. Un paquet part du réseau local L1 à destination du réseau local L2.
  - a. En utilisant l'extrait de la table de routage de R1, vers quel routeur R1 envoie-t-il ce paquet : R2 ou R3 ? Justifier.
  - b. A l'aide des extraits de tables de routage ci-dessus, nommer les routeurs traversés par ce paquet, lorsqu'il va du réseau L1 au réseau L2.
2. La liaison entre R1 et R2 est rompue.
  - a. Sachant que ce réseau utilise le protocole RIP (distance en nombre de sauts), donner l'un des deux chemins possibles que pourra suivre un paquet allant de L1 vers L2.
  - b. Dans les extraits de tables de routage ci-dessus, pour le chemin de la question 2.a, quelle(s) ligne(s) sera (seront) modifiée(s) ?
3. On a rétabli la liaison entre R1 et R2.  
Par ailleurs, pour tenir compte du débit des liaisons, on décide d'utiliser le protocole OSPF (distance liée au coût minimal des liaisons) pour effectuer le routage. Le coût des liaisons entre les routeurs est donné par le tableau suivant :

Liaison	R1-R2	R1-R3	R2-R3	R2-R4	R2-R5	R2-R6	R3-R4	R4-R5	R4-R6	R5-R6
Coût	100	100	?	1	10	10	10	1	10	1

- a. Le coût  $\Delta$  d'une liaison est donné ici par la formule

$$= \Delta \frac{10^8}{BP}$$

- où BP est la bande passante de la connexion en bps (bit par seconde).  
Sachant que la bande passante de la liaison R2-R3 est de 10 Mbps, calculer le coût correspondant.
- b. Déterminer le chemin parcouru par un paquet partant du réseau L1 et arrivant au réseau L2, en utilisant le protocole OSPF.
- c. Indiquer pour quel(s) routeur(s) l'extrait de la table de routage sera modifié pour un paquet à destination de L2, avec la métrique OSPF.

### EXERCICE 4 (3 points)

Cet exercice porte sur la gestion des processus et la programmation orientée objet.

On rappelle qu'un processus est l'instance d'un programme en cours d'exécution. Il est identifié par un numéro unique appelé PID. L'ordonnanceur est la composante du système d'exploitation qui gère l'allocation du processeur entre les différents processus. Nous allons nous intéresser à l'algorithme d'ordonnancement du tourniquet dont le fonctionnement est résumé ci-dessous :

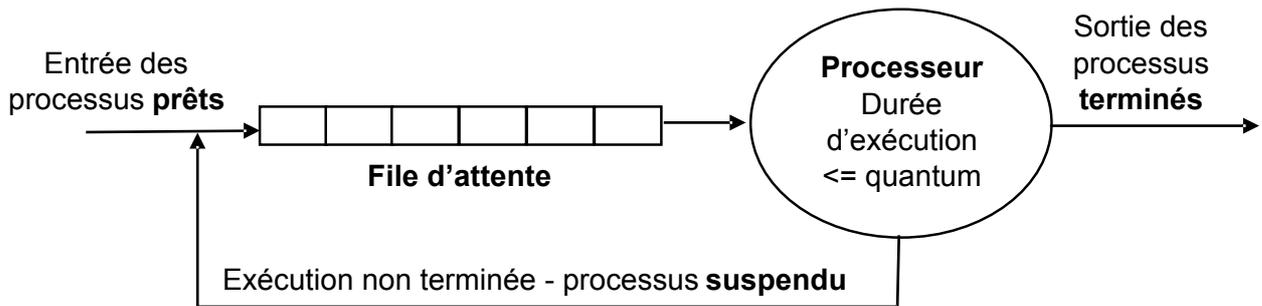


Schéma d'ordonnancement du tourniquet

- Les processus prêts à être exécutés sont placés dans une file d'attente selon leur ordre d'arrivée ;
- L'ordonnanceur alloue le processeur à chaque processus de la file d'attente un même nombre de cycles CPU, appelé **quantum** ;
- Si le processus n'est pas terminé au bout de ce temps, son exécution est suspendue et il est mis à la fin de la file d'attente ;
- Si le processus est terminé, il sort définitivement de la file d'attente.

1. On considère trois processus soumis à l'ordonnanceur **au même instant** pour lesquels on donne les informations ci-dessous :

PID	Durée (en cycles CPU)	Ordre d'arrivée
11	4	1
20	2	2
32	3	3

- Si le quantum du tourniquet est d'un cycle CPU, recopier et compléter la suite des PID des processus dans l'ordre de leur exécution :  
11, 20, 32, 11, .....
- Donner la composition de la suite des PID lorsque le quantum du tourniquet est de deux cycles CPU.



c. La fonction `tourniquet` ci-dessous implémente l'algorithme décrit dans l'exercice.

Elle prend en paramètres une liste d'objets `Processus` donnés par ordre d'arrivée et un nombre entier positif correspondant au quantum. La fonction renvoie la liste des PID dans l'ordre de leur exécution par le processeur. Recopier et compléter sur la copie le code manquant.

```
1  def tourniquet(liste_attente, quantum):
2      ordre_execution = []
3      while liste_attente != []:
4          # On extrait le premier processus
5          processus = liste_attente.pop(0)
6          processus.change_etat("En cours d'exécution")
7          compteur_tourniquet = 0
8          while ..... and .....:
9              ordre_execution.append(.....)
10             processus.execute_un_cycle()
11             compteur_tourniquet = compteur_tourniquet + 1
12             if ..... :
13                 processus.change_etat("Suspendu")
14                 liste_attente.append(processus)
15             else:
16                 processus.change_etat(.....)
17     return ordre_execution
```