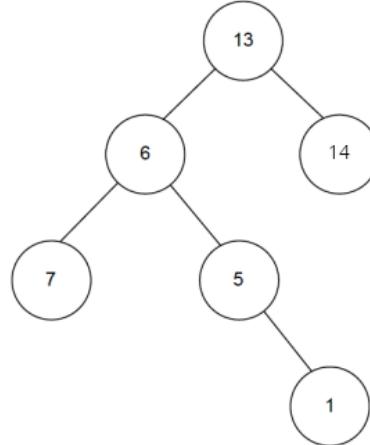


DS3**1. Arbres**

Cet exercice porte sur les arbres binaires, les arbres binaires de recherche, la programmation orientée objet et la récursivité.

1. On considère l'arbre ci-dessous :



a. Justifier que cet arbre est un arbre binaire.

b. Indiquer si l'arbre ci-dessus est un arbre binaire de recherche (ABR). Justifier votre réponse.

2. On considère la classe `Noeud`, nous permettant de définir les arbres binaires, définie de la manière suivante en Python :

```

class Noeud:
    def __init__(self, g, v, d):
        """crée un noeud d'un arbre binaire"""
        self.gauche = g
        self.valeur = v
        self.droit = d
  
```

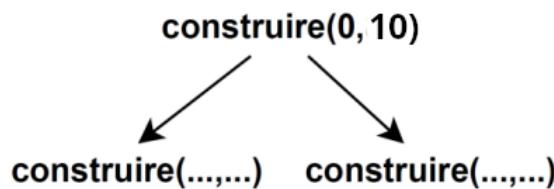
On considère également la fonction `construire` ci-dessus qui prend en paramètre deux entiers `mini` et `maxi` et qui renvoie un arbre.

```

def construire(mini, maxi):
    """mini, maxi: entiers respectant mini <= maxi"""
    assert isinstance(mini, int) and isinstance(maxi, int) and ...
    if maxi - mini == 1 or maxi - mini == 0:
        return Noeud(None, mini, None)
    elif maxi - mini == 2:
        return Noeud(None, (mini+maxi)//2, None)
    else:
        sag = construire(mini, (mini+maxi)//2)
        sad = construire((mini+maxi)//2, maxi)
        return Noeud(sag, (mini+maxi)//2, sad)
  
```

La fonction `isinstance(obj, t)` renvoie `True` si l'objet `obj` est du type `t`, sinon `False`.

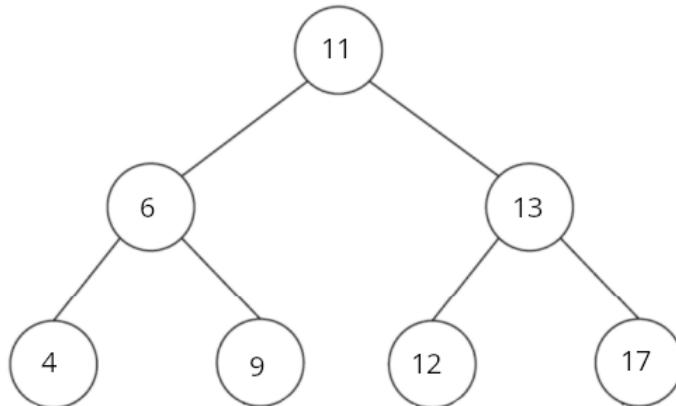
- Recopier et compléter sur votre copie la ligne 3 de l'assertion de la fonction `construire` de manière à vérifier les conditions sur les paramètres `mini` et `maxi`.
- On exécute l'instruction `construire(0,10)`. Déterminer quels sont les différents appels récursifs de la fonction `construire` exécutés. On représentera ces appels sous forme d'une arborescence, par exemple :



- Dessiner l'arbre renvoyé par l'instruction `construire(0,10)`.
- Dessiner l'arbre renvoyé par l'instruction `construire(1,6)`.
- Donner le résultat d'un parcours infixe sur l'arbre obtenu à la question 2.c. Expliquer pourquoi ce parcours permet d'affirmer que l'arbre est un arbre binaire de recherche.
- La fonction récursive maximum ci-dessous prend en paramètre un arbre binaire de recherche `abr` et renvoie la valeur maximale de ses nœuds. Recopier et compléter les lignes 5 et 7 de cette fonction.

```
def maximum(abr):
    if abr is None:
        return None
    elif abr.droit is None:
        return .....
    else :
        return .....
```

3. On donne l'arbre binaire de recherche `abr_7_noeuds` suivant :



On donne également ci-dessous la fonction `mystere` qui prend en paramètres un arbre binaire de recherche `abr`, un entier `x` et une liste `liste`.

```
def mystere(abr, x, liste):
    """
        abr -- arbre binaire de recherche
        x -- int
        liste -- list
        Renvoie -- list"""
    if abr is None:
        return []
    else:
        liste.append(abr.valeur)
        if x == abr.valeur:
            return liste
        elif x < abr.valeur:
            return mystere(abr.gauche, x, liste)
        else:
            return mystere(abr.droit, x, liste)
```

a. Donner les résultats obtenus lorsque l'on exécute les instructions `mystere(abr_7_noeuds,5,[])`, puis `mystere(abr_7_noeuds,6,[])` puis `mystere(abr_7_noeuds,12,[])`.

b. Décrire quel peut être le rôle de la fonction `mystere`.

2. Gestion des processus

Cet exercice porte sur les systèmes d'exploitation, les commandes UNIX, les structures de données (de type LIFO et FIFO) et les processus.

“Linux ou GNU/Linux est une famille de systèmes d’exploitation open source de type Unix fondée sur le noyau Linux, créé en 1991 par Linus Torvalds. De nombreuses distributions Linux ont depuis vu le jour et constituent un important vecteur de popularisation du mouvement du logiciel libre.”

Source : Wikipédia, extrait de l'article consacré à GNU/Linux.

“Windows est au départ une interface graphique unifiée produite par Microsoft, qui est devenue ensuite une gamme de systèmes d’exploitation à part entière, principalement destinés aux ordinateurs compatibles PC. Windows est un système d’exploitation propriétaire.”

Source : Wikipédia, extrait de l'article consacré à Windows.

1. Expliquer succinctement les différences entre les logiciels libres et les logiciels propriétaires.
2. Expliquer le rôle d'un système d'exploitation.

On donne ci-dessous une arborescence de fichiers sur un système GNU/Linux (les noms encadrés représentent des répertoires, les noms non encadrés représentent des fichiers, / correspond à la racine du système de fichiers) :

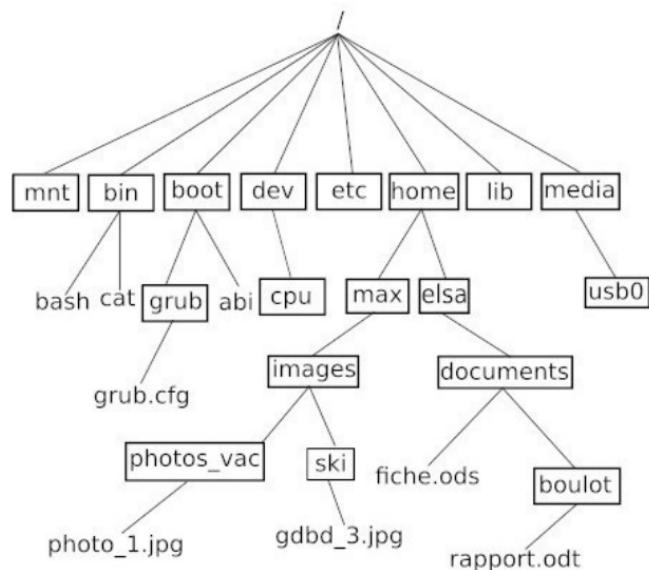


Figure 1. Arborescence de fichiers

3. Indiquer le chemin absolu du fichier grub.cfg.

On suppose que le répertoire courant est le répertoire elsa.

4. Indiquer le chemin relatif du fichier rapport.odt.

L'utilisatrice Max ouvre une console (aussi parfois appelée terminal), le répertoire courant étant toujours le répertoire max. On fournit ci-dessous un extrait du manuel de la commande UNIX cp :

```
NOM          cp - copie un fichier  
UTILISATION cp fichier source fichier destination
```

5. Déterminer le contenu du répertoire `images` et du répertoire `photos_vac` après avoir exécuté la commande suivante dans la console :

```
cp images/photos vac/photo 1.jpg images/ski
```

“Un système d’exploitation est multitâche (en anglais : multitasking) s’il permet d’exécuter, de façon apparemment simultanée, plusieurs programmes informatiques. GNU/Linux, comme tous les systèmes d’exploitation modernes, gère le multitâche.” “Dans le cas de l’utilisation d’un monoprocesseur, la simultanéité apparente est le résultat de l’alternance rapide d’exécution des processus présents en mémoire.”

Source : Wikipédia, extraits de l'article consacré au Multitâche.

Dans la suite de l'exercice, on s'intéresse aux processus. On considère qu'un monoprocesseur est utilisé. On rappelle qu'un processus est un programme en cours d'exécution. Un processus est soit élu, soit bloqué, soit prêt.

6. Recopier et compléter le schéma ci-dessous avec les termes suivants :
élu, bloqué, prêt, élection, blocage, déblocage

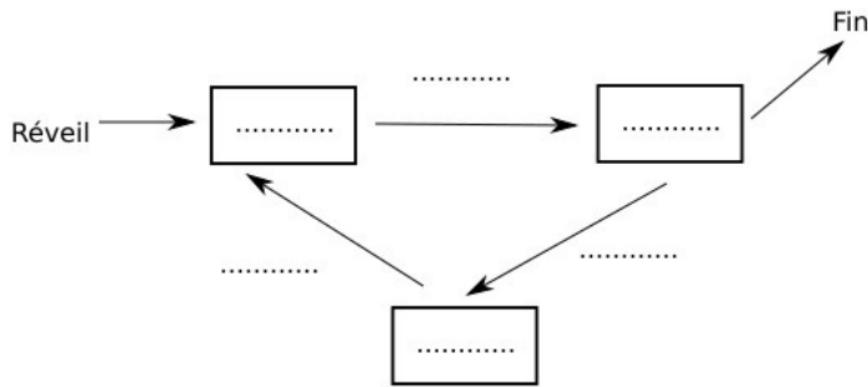


Figure 2. Schéma processus

7. Donner l'exemple d'une situation qui constraint un processus à passer de l'état élu à l'état bloqué.

“Dans les systèmes d’exploitation, l’ordonnanceur est le composant du noyau du système d’exploitation choisissant l’ordre d’exécution des processus sur le processeur d’un ordinateur.”

Source : Wikipédia, extrait de l’article consacré à l’ordonnancement.

L’ordonnanceur peut utiliser plusieurs types d’algorithmes pour gérer les processus.

L’algorithme d’ordonnancement par “ordre de soumission” est un algorithme de type FIFO (First In First Out), il utilise donc une file.

8. Nommer une structure de données linéaire de type LIFO (Last In First Out).

À chaque processus, on associe un instant d’arrivée (instant où le processus demande l’accès au processeur pour la première fois) et une durée d’exécution (durée d’accès au processeur nécessaire pour que le processus s’exécute entièrement).

Par exemple, l’exécution d’un processus P4 qui a un instant d’arrivée égal à 7 et une durée d’exécution égale à 2 peut être représentée par le schéma suivant :



Figure 3. Utilisation du processeur

Figure 3. Utilisation du processeur

L’ordonnanceur place les processus qui ont besoin d’un accès au processeur dans une file, en respectant leur ordre d’arrivée (le premier arrivé étant placé en tête de file). Dès qu’un processus a terminé son exécution, l’ordonnanceur donne l’accès au processus suivant dans la file.

Le tableau suivant présente les instants d’arrivées et les durées d’exécution de cinq processus :

| Processus | instant d’arrivée | durée d’exécution |
|-----------|-------------------|-------------------|
| P1 | 1 | 3 |
| P2 | 0 | 6 |
| P3 | 3 | 4 |
| P4 | 5 | 2 |
| P5 | 6 | 1 |

9. Recopier et compléter le schéma ci-dessous avec les processus P1 à P5 en utilisant les informations présentes dans le tableau ci-dessus et l’algorithme d’ordonnancement “par ordre de soumission”.

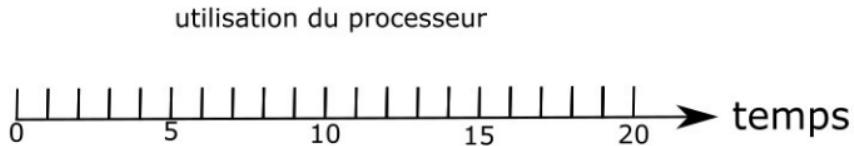


Figure 4. Utilisation du processeur

On utilise maintenant un autre algorithme d'ordonnancement : l'algorithme d'ordonnancement “par tourniquet”. Dans cet algorithme, la durée d'exécution d'un processus ne peut pas dépasser une durée Q appelée quantum et fixée à l'avance. Si ce processus a besoin de plus de temps pour terminer son exécution, il doit retourner dans la file et attendre son tour pour poursuivre son exécution. Par exemple, si un processus P_1 a une durée d'exécution de 3 et que la valeur de Q a été fixée à 2, P_1 s'exécutera pendant deux unités de temps avant de retourner à la fin de la file pour attendre son tour ; une fois à nouveau élu, il pourra terminer de s'exécuter pendant sa troisième et dernière unité de temps d'exécution.

10. Recopier et compléter le schéma ci-dessous, en utilisant l'algorithme d'ordonnancement “par tourniquet” et les mêmes données que pour la question 9, en supposant que le quantum Q est fixé 3.

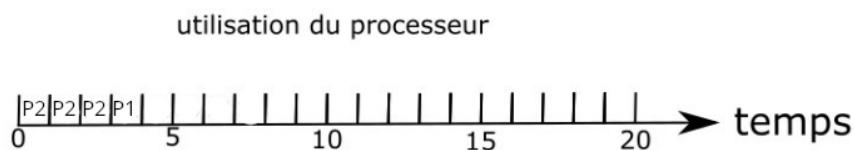


Figure 5. Utilisation du processeur

On considère deux processus P_1 et P_2 , et deux ressources R_1 et R_2 .

11. Décrire une situation qui conduit les deux processus P_1 et P_2 en situation d'interblocage.

3. Réseau

Cet exercice porte sur l'architecture matérielle, les réseaux, les routeurs et les protocoles de routage.

On considère un réseau local N1 constitué de trois ordinateurs M1, M2, M3 et dont les adresses IP sont les suivantes :

- M1 : 192.168.1.1/24 ;
- M2 : 192.168.1.2/24 ;
- M3 : 192.168.2.3/24.

On rappelle que le “/24” situé à la suite de l'adresse IP de M1 signifie que l'adresse réseau du réseau local N1 est 192.168.1.0. Depuis l'ordinateur M3, un utilisateur exécute la commande ping vers l'ordinateur M3 comme suit :

```
util@M1 ~ % ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
Hôte inaccessible
```

1. Expliquer le résultat obtenu lors de l'utilisation de la commande ping (on part du principe que la connexion physique entre les machines est fonctionnelle).

On ajoute un routeur R1 au réseau N1 :

“Un routeur moderne se présente comme un boîtier regroupant carte mère, microprocesseur, ROM, RAM ainsi que les ressources réseaux nécessaires (Wi-Fi, Ethernet...). On peut donc le voir comme un ordinateur minimal dédié, dont le système d'exploitation peut être un Linux allégé. De même, tout ordinateur disposant des interfaces adéquates (au minimum deux, souvent Ethernet) peut faire office de routeur s'il est correctement configuré (certaines distributions Linux minimales spécialisent la machine dans cette fonction).”

Source : Wikipédia, article “Routeur”

2. Définir l'acronyme RAM.
3. Expliquer le terme Linux.
4. Expliquer pourquoi il est nécessaire d'avoir “au minimum deux” interfaces réseau dans un routeur.

Le réseau N1 est maintenant relié à d'autres réseaux locaux (N2, N3, N4) par l'intermédiaire d'une série de routeurs (R1, R2, R3, R4, R5, R6) :

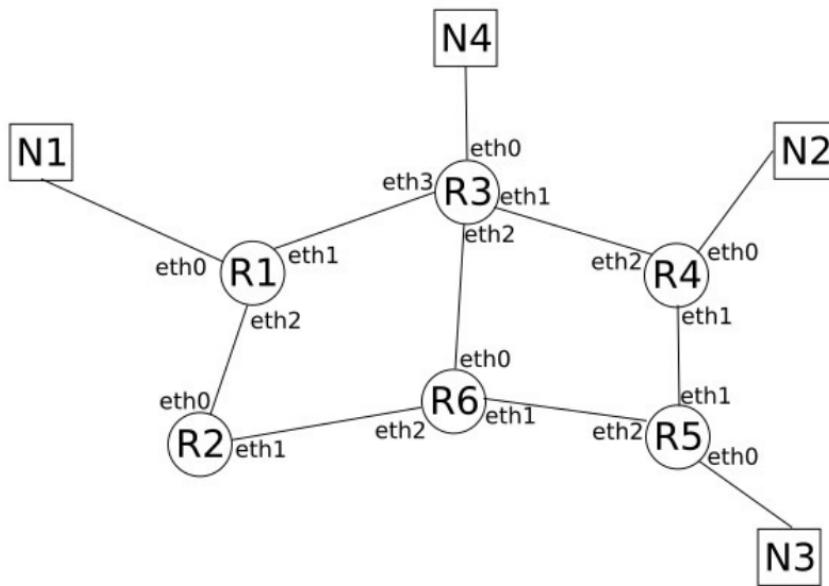


Figure 1. Schéma du réseau

5. Attribuer une adresse IP valide à l'interface eth0 du routeur R3 sachant que l'adresse réseau du réseau N4 est 172.16.2.0/24.

Dans un premier temps, on utilise le protocole de routage RIP (Routing Information Protocol). On rappelle que dans ce protocole, la métrique de la table de routage correspond au nombre de routeurs à traverser pour atteindre la destination. La table de routage du routeur R5 est donnée dans le tableau suivant :

Table de routage du routeur R5

| destination | interface de sortie | métrique |
|-------------|---------------------|----------|
| N3 | eth0 | 0 |
| N2 | eth1 | 1 |
| N2 | eth2 | 3 |
| N4 | eth1 | 2 |
| N4 | eth2 | 2 |
| N1 | eth1 | 3 |
| N1 | eth2 | 3 |

6. Déterminer le chemin parcouru par un paquet de données pour aller d'une machine appartenant au réseau N3 à une machine appartenant au réseau N2.

Le routeur R4 tombe en panne. Après quelques minutes, la table de routage de R5 est modifiée afin de tenir compte de cette panne.

7. Dresser la table de routage du routeur R5 suite à la panne du routeur R4.

Le routeur R4 est de nouveau fonctionnel. Dans la suite de cet exercice, on utilise le protocole de routage OSPF (Open Shortest Path First). On rappelle que dans ce protocole, la métrique de la table de routage correspond à la somme des coûts :

$$\text{coût} = \frac{10^9}{d} \quad (\text{où } d \text{ est la bande passante d'une liaison en bit/s})$$

Le réseau est constitué de 3 types de liaison de communication :

- Fibre avec un débit de 1 Gbit/s ;
- Fast-Ethernet avec un débit de 100 Mbit/s ;
- Ethernet avec un débit de 10 Mbit/s.

8. Calculer le coût de chacune de ces liaisons de communication.

La table de routage du routeur R1 est donnée dans le tableau suivant :

Table de routage du routeur R1

| destination | interface de sortie | métrique |
|-------------|---------------------|----------|
| N1 | eth0 | 0 |
| N2 | eth1 | 101 |
| N2 | eth2 | 22 |
| N3 | eth1 | 111 |
| N3 | eth2 | 12 |
| N4 | eth1 | 100 |
| N4 | eth2 | 21 |

D'autre part, le type des différentes liaisons inter-routeurs sont les suivantes :

- R1 - R2 : Fibre ;
- R1 - R3 : Ethernet ;
- R2 - R6 : INCONNU ;
- R3 - R6 : Fast-Ethernet ;
- R3 - R4 : Fibre ;
- R4 - R5 : Fast-Ethernet ;
- R5 - R6 : Fibre.

9. Déduire de la table de routage de R1 et du schéma du réseau le type de la liaison inter-routeur R2 - R6.

Des travaux d'amélioration ont été réalisés sur ce réseau : la liaison inter-routeur R1-R3 est désormais de type Fibre.

10. Modifier la table de routage de R1 en tenant compte de cette amélioration.

On ajoute un réseau local N5 et un routeur R7 au réseau étudié ci-dessus. Le routeur R7 possède trois interfaces réseaux eth0, eth1 et eth2. eth0 est directement relié au réseau local N5. eth1 et eth2 sont reliés à d'autres routeurs (ces liaisons inter-routeur sont de type Fibre).

Les deux tableaux suivants présentent un extrait des tables de routage des routeurs R1 et R3 :

Extrait table de routage du routeur R1

| destination | interface de sortie | métrique |
|-------------|---------------------|----------|
| ... | ... | ... |
| N5 | eth1 | 12 |
| N5 | eth2 | 11 |

Extrait table de routage du routeur R3

| destination | interface de sortie | métrique |
|-------------|---------------------|----------|
| ... | ... | ... |
| N5 | eth1 | 13 |
| N5 | eth2 | 11 |
| N5 | eth3 | 12 |

11. Recopier et compléter le schéma du réseau (Figure. 1) en ajoutant le routeur R7 et le réseau local N5.