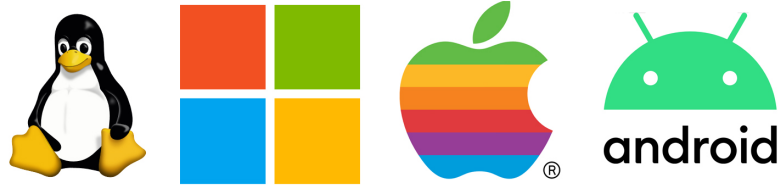
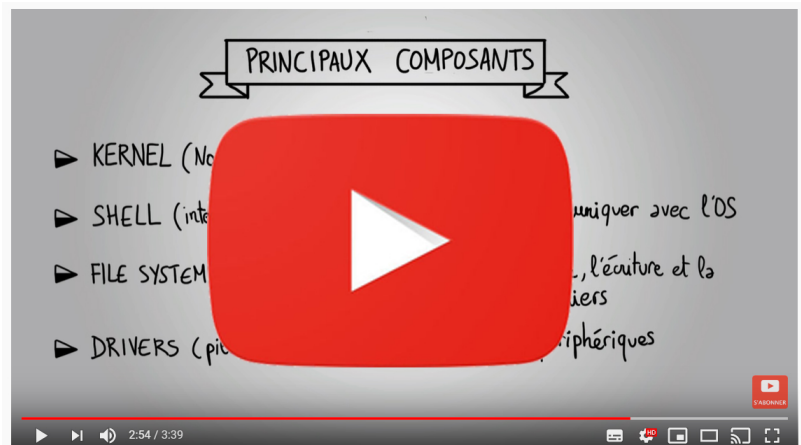


# Systèmes d'exploitation



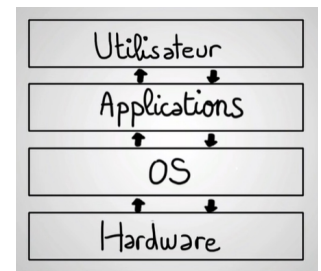
## 1. Principales fonctions

([https://www.youtube.com/watch?time\\_continue=211&v=AcZ87MTiXr4](https://www.youtube.com/watch?time_continue=211&v=AcZ87MTiXr4))Après avoir regardé la vidéo ci-contre répondre aux questions ci-dessous :



### Exercice 1 :

1. Citer des exemples de systèmes d'exploitations.
2. Expliquer le schéma ci-contre :
3. Quelles sont les quatre composantes principales d'un système d'exploitation ?

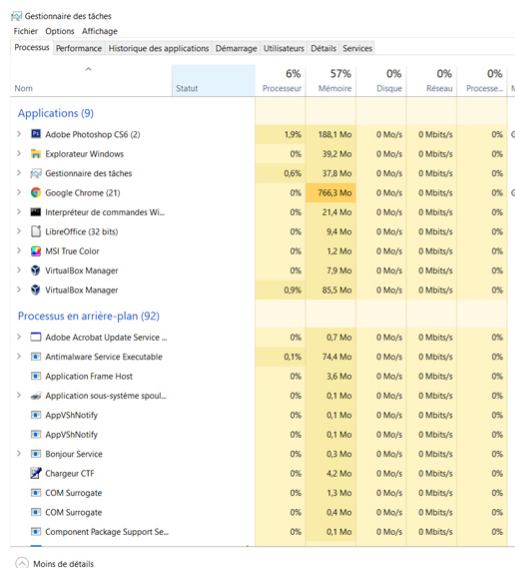


Réponses :

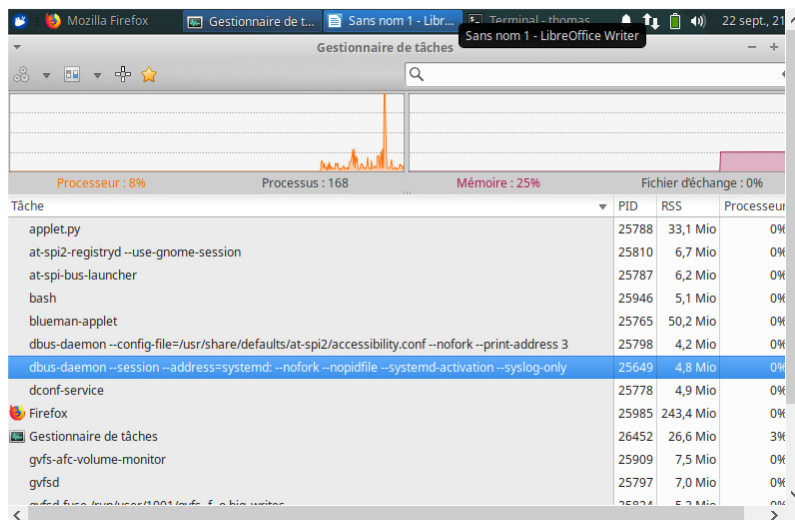
- 1.
- 2.
3.
  - 
  - 
  - 
  -

## Exercice 2 : ouvrir gestionnaire des tâches

Ouvrir le gestionnaire des tâches sur votre ordinateur ( Ctrl+Alt+Suppr sous Windows ou bien Système > Gestionnaire des tâches sous Linux).



| Nom                                   | Statut | 6%         | 57%      | 0%     | 0%        | 0%         |
|---------------------------------------|--------|------------|----------|--------|-----------|------------|
|                                       |        | Processeur | Mémoire  | Disque | Réseau    | Processeur |
| <b>Applications (9)</b>               |        |            |          |        |           |            |
| Adobe Photoshop CS6 (2)               |        | 1,9%       | 188,1 Mo | 0 Mo/s | 0 Mbits/s | 0%         |
| Explorateur Windows                   |        | 0%         | 39,2 Mo  | 0 Mo/s | 0 Mbits/s | 0%         |
| Gestionnaire des tâches               |        | 0,6%       | 37,8 Mo  | 0 Mo/s | 0 Mbits/s | 0%         |
| Google Chrome (21)                    |        | 0%         | 766,3 Mo | 0 Mo/s | 0 Mbits/s | 0%         |
| Interpréteur de commandes Wi...       |        | 0%         | 21,4 Mo  | 0 Mo/s | 0 Mbits/s | 0%         |
| LibreOffice (32 bits)                 |        | 0%         | 9,4 Mo   | 0 Mo/s | 0 Mbits/s | 0%         |
| MSI True Color                        |        | 0%         | 1,2 Mo   | 0 Mo/s | 0 Mbits/s | 0%         |
| VirtualBox Manager                    |        | 0%         | 7,9 Mo   | 0 Mo/s | 0 Mbits/s | 0%         |
| VirtualBox Manager                    |        | 0,9%       | 85,5 Mo  | 0 Mo/s | 0 Mbits/s | 0%         |
| <b>Processus en arrière-plan (92)</b> |        |            |          |        |           |            |
| Adobe Acrobat Update Service ...      |        | 0%         | 0,7 Mo   | 0 Mo/s | 0 Mbits/s | 0%         |
| Antimalware Service Executable        |        | 0,1%       | 74,4 Mo  | 0 Mo/s | 0 Mbits/s | 0%         |
| Application Frame Host                |        | 0%         | 3,6 Mo   | 0 Mo/s | 0 Mbits/s | 0%         |
| Application sous-système spool...     |        | 0%         | 0,1 Mo   | 0 Mo/s | 0 Mbits/s | 0%         |
| AppVSIHnotify                         |        | 0%         | 0,1 Mo   | 0 Mo/s | 0 Mbits/s | 0%         |
| AppVSIHnotify                         |        | 0%         | 0,1 Mo   | 0 Mo/s | 0 Mbits/s | 0%         |
| Bonjour Service                       |        | 0%         | 0,3 Mo   | 0 Mo/s | 0 Mbits/s | 0%         |
| Chargeur CTF                          |        | 0%         | 4,2 Mo   | 0 Mo/s | 0 Mbits/s | 0%         |
| COM Surrogate                         |        | 0%         | 1,3 Mo   | 0 Mo/s | 0 Mbits/s | 0%         |
| COM Surrogate                         |        | 0%         | 0,4 Mo   | 0 Mo/s | 0 Mbits/s | 0%         |
| Component Package Support Se...       |        | 0%         | 0,1 Mo   | 0 Mo/s | 0 Mbits/s | 0%         |



| Tâche   | PID   | RSS       | Processeur |
|---|-------|-----------|------------|
| applet.py   | 25788 | 33,1 Mio  | 0%         |
| at-spi2-registryd --use-gnome-session   | 25810 | 6,7 Mio   | 0%         |
| at-spi-bus-launcher   | 25787 | 6,2 Mio   | 0%         |
| bash  | 25946 | 5,1 Mio   | 0%         |
| blueman-applet  | 25765 | 50,2 Mio  | 0%         |
| dbus-daemon --config-file=/usr/share/defaults/at-spi2/accessibility.conf --nofork --print-address 3 | 25798 | 4,2 Mio   | 0%         |
| dbus-daemon --session --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only    | 25649 | 4,8 Mio   | 0%         |
| dconf-service   | 25778 | 4,9 Mio   | 0%         |
| Firefox   | 25985 | 243,4 Mio | 0%         |
| Gestionnaire de tâches  | 26452 | 26,6 Mio  | 3%         |
| gvfs-afc-volume-monitor   | 25909 | 7,5 Mio   | 0%         |
| gvfsd   | 25797 | 7,0 Mio   | 0%         |
| gvfsd-fuse (mounts/1001/rofs-fs-bio-writer)   | 25924 | 5,2 Mio   | 0%         |

Déterminer le nombre de processus en train de s'exécuter.

Réponses :

## 2. Systèmes libres et propriétaires

Qu'est-ce qu'un logiciel libre ?



(<https://education.francetv.fr/matiere/technologie/seconde/video/comprendre-les-logiciels-libres-1-2>)

Regarder l'interview de Philippe Lhardy de l'association Linux Azur, puis répondre aux questions ci-dessous

### Exercice 3 :

1. Quelles sont les 4 libertés que doit proposer un logiciel libre ?
2. Donner quelques exemples de logiciels libres

Réponses :

1.

- 
- 
- 
- 

1.

- 
- 
- 

## Exercice 4 :

Voici une liste de systèmes d'exploitations. Pour chacun d'entre eux, indiquer s'il s'agit d'un système libre ou propriétaire :

- Windows :
- MacOS :
- Windows Server :
- Linux :
- iOS :
- android :

Réponses :

- Windows :
- MacOS :
- Windows Server :
- Linux :
- iOS :
- android :

## Un système d'exploitation libre : Linux

Regarder la vidéo ci-contre puis répondre aux questions ci-dessous. (<https://www.youtube.com/watch?v=Uq5ddTUtdKM>)



## Exercice 5 :

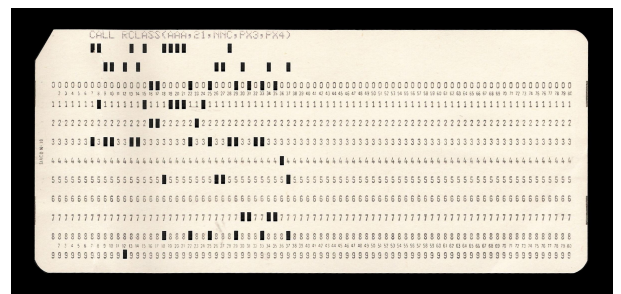
1. En quelle année est né Linux ?
2. Qui est son concepteur ?
3. Mais au fait, qu'est-ce que Linux ?

Réponses:

- 1.
- 2.
- 3.

## 3. Systèmes de fichiers

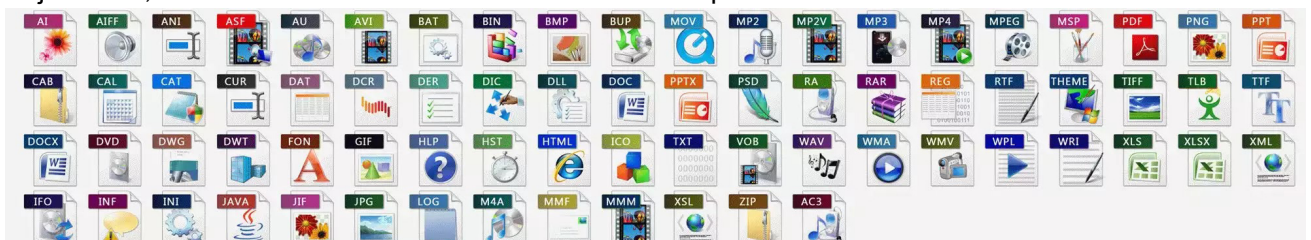
- Pour les premiers ordinateurs, les données étaient écrites sous forme de cartes perforées(des fiches).



- Il fallait bien les organiser et les stocker quelque part...Ci-contre : Archivage de cartons de cartes perforées archivés au service du NARA( National Archives and Records Administration) en 1959. Chaque carton peut contenir 2 000 cartes.



- Aujourd'hui, un fichier est une suite de bits formatée à laquelle on a donné un nom.:

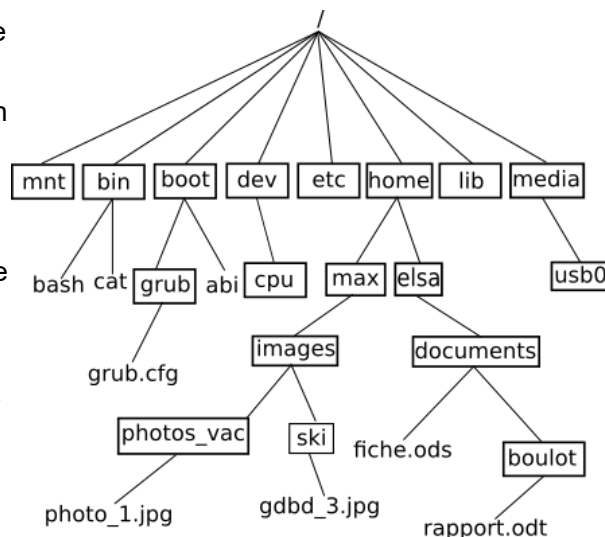


## Deux exemples d'organisations : L'arbre(linux) et la forêt(windows)

### Sous Linux, comme un arbre :

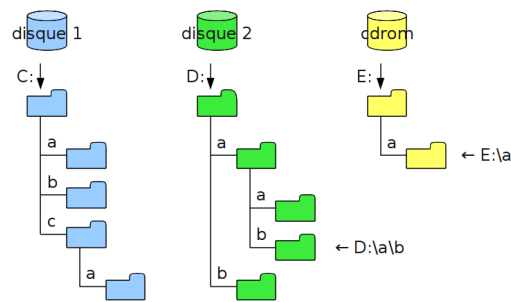
Dans les systèmes de type Linux, toute l'information stockée dans vos supports de stockage (disques durs, clé USB, cartes SD, CD-ROM, etc.) est nécessairement accessible en suivant un chemin partant d'un emplacement logique appelée la racine (notée par une simple barre oblique '/' ).

La racine symbolise une partition définie comme étant la base du stockage de vos fichiers. Puis, cette base se sépare (comme des branches d'un arbre) logiquement en répertoires (dossiers), eux-mêmes séparés en sous-répertoires et sous-sous-répertoires, etc. dans lesquels sont enregistrés les fichiers (symboliquement, les feuilles de l'arbre).



### Sous Windows, comme une forêt :

Dans le système Windows, chaque périphérique possède sa propre racine, identifiée par une lettre (disque dur, clé usb, lecteur dvd,...). La répartition suit ensuite le principe de l'arbre avec des répertoires, des sous répertoires et des fichiers. Ce système est donc un ensemble d'arbres.



## Chemins absolus et relatifs.

Pour indiquer la position d'un fichier (ou d'un répertoire) dans l'arborescence, il existe 2 méthodes : indiquer un chemin absolu ou indiquer un chemin relatif.

- Le chemin absolu se décrit depuis la racine. Par exemple, dans l'arborescence Linux, le chemin absolu du fichier `fiche.ods` sera : `/home/elsa/documents/fiche.ods` ce chemin démarre bien de la racine `/` (les symboles de séparation sont aussi des `/` ).
- Il est possible d'indiquer le chemin non pas depuis la racine, mais depuis un répertoire quelconque, nous parlerons alors de chemin relatif. Le chemin relatif permettant d'accéder au fichier `photo_1.jpg` depuis le répertoire `max` est : `images/photo_vac/photo_1.jpg` . On remarque l'absence du `/` au début du chemin (c'est cela qui nous permettra de distinguer un chemin relatif et un chemin absolu).

Vocabulaire : On dit que le répertoire `photo_vac` est un répertoire enfant du répertoire `image` .

- Imaginons maintenant que nous désirions indiquer le chemin relatif pour accéder au fichier `gdbd_3.jpg` depuis le répertoire `photos_vac` . Comment faire ? Il faut "remonter" d'un "niveau" dans l'arborescence pour se retrouver dans le répertoire "images" et ainsi pouvoir repartir vers la bonne "branche". Pour ce faire il faut utiliser 2 points : `..` `../ski/gdbd_3.jpg`

Vocabulaire : On dit que le répertoire `images` est le répertoire parent du répertoire `ski` .

Remarques : Le principe est le même sous Windows . Cependant, on utilise `/` comme séparateur sous Linux et `\` sous Windows.

## Exercice 6 :

1. En vous basant sur l'arborescence Linux ci-dessus, déterminez le chemin absolu permettant d'accéder au fichiers :
  - `cat`
  - `rapport.odt`
2. Toujours en vous basant sur cette même arborescence, déterminez le chemin relatif permettant d'accéder au fichier :
  - `rapport.odt` depuis le répertoire `elsa`
  - `fiche.ods` depuis le répertoire `boulot`

Réponses :

1. chemins absolus :

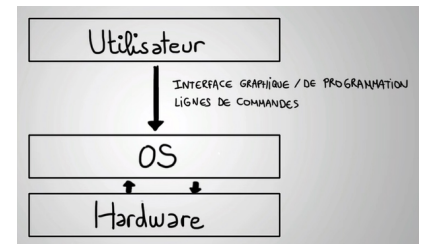
- 
- 

1. chemins relatifs

- 
- 

## 4. Lignes de commandes

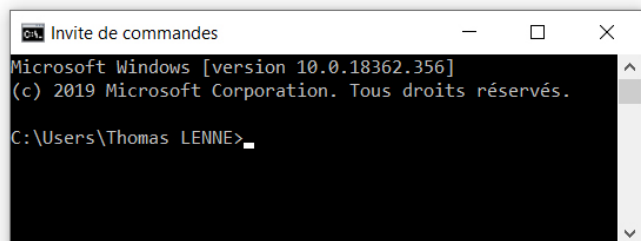
Il est possible à l'utilisateur d'envoyer des instructions directement au système d'exploitation via une interface, une console dans laquelle on écrit les commandes



## Ouvrir une console

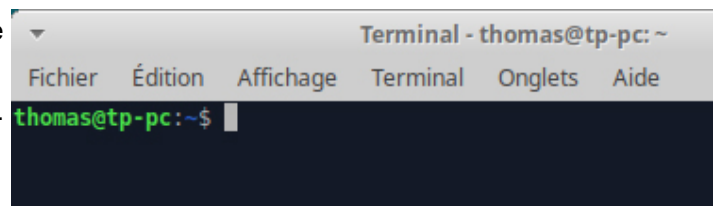
### Sous Windows

Exécuter `cmd.exe` à partir de la barre de recherche ou dans la barre d'adresse de l'explorateur pour l'ouvrir à un endroit précis. Par défaut l'invite de commandes s'ouvre dans le répertoire de l'utilisateur.



### Sous Linux

Exécuter le terminal à partir du menu principal (icône du système d'exploitation) ou par menu contextuel (clic droit) pour l'ouvrir à un endroit précis. Par défaut l'invite de commandes s'ouvre dans le répertoire de l'utilisateur. Le caractère `~` est un raccourci pour indiquer ici `/home/thomas`.





## Quelques commandes Linux et Windows

Certaines commandes sont similaires, certaines possèdent des options. Attention, sous Linux le séparateur dans l'arborescence est / , alors que pour Windows, c'est \

| Action                                      | Windows            | Linux              | Exemple de syntaxe linux  |
|---|--------------------|--------------------|---|
| Copier un fichier                           | copy               | cp                 | cp <i>thisfile.txt</i> /home/ <i>thisdirectory</i>  |
| Déplacer un fichier                         | move               | mv                 | mv <i>thisfile.txt</i> /home/ <i>thisdirectory</i>  |
| Lister le contenu d'un répertoire           | dir                | ls                 | ls  |
| Effacer l'écran de la console               | cls                | clear              | clear   |
| Fermer la console                           | exit               | exit               | exit  |
| Afficher la date                            | date               | date               | date  |
| Supprimer un fichier                        | del                | rm                 | rm <i>thisfile.txt</i>  |
| Supprimer un répertoire                     | rd                 | rd                 | rm <i>thisdirectory</i> : supprime un répertoire vide<br>rm /s <i>thisdirectory</i> : supprime un répertoire et son contenu |
| Afficher un message à l'écran de la console | echo               | echo               | echo <i>this message</i>  |
| Enregistrer un message dans un fichier      | echo               | echo               | echo « <i>this text</i> » > <i>thisfile.txt</i><br>( >> pour ajouter au fichier)  |
| Afficher de l'aide pour une commande        | command /?         | man ou info        | man <i>command</i>  |
| Créer un répertoire                         | md                 | mkdir              | mkdir <i>directory</i>  |
| Renommer un fichier                         | ren                | mv                 | mv <i>thisfile.txt</i> <i>thatfile.txt</i>  |
| Afficher le répertoire courant              | chdir              | pwd                | pwd   |
| Changer de répertoire avec un chemin absolu | cd <i>pathname</i> | cd <i>pathname</i> | cd / <i>directory/directory</i>   |
| Se placer dans le répertoire parent         | cd..               | cd ..              | cd ..   |
| Afficher l'heure                            | time               | date               | date  |

## Exercice 7 : Réaliser cet exercice sous Windows et sous Linux

En vous aidant de la liste ci-dessus, décrire les commandes à exécuter pour :

1. Afficher le contenu de votre dossier utilisateur.
2. Créer un répertoire "A".
3. Afficher le contenu du dossier utilisateur pour vérifier.
4. Se placer dans le répertoire "A".
5. Créer deux sous répertoire "1" et "2".
6. Afficher le contenu du sous répertoire "A" pour vérifier.
7. Se placer dans le répertoire "1".
8. Créer un fichier "toto.txt" contenant le mot "texte1".
9. Ajouter le mot "texte2" à ce fichier.
10. Se placer dans le répertoire "2".
11. Créer un fichier "titi.txt" contenant le mot "texte2".
12. Remplacer le mot "texte2" par "texte3" dans ce fichier.
13. Copier le fichier titi.txt dans le répertoire "1".
14. Supprimer le répertoire "2" et son contenu



Réponse Windows :

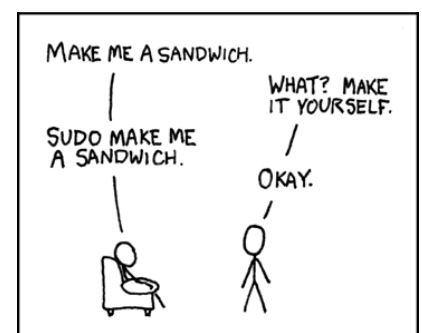
- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.
- 11.
- 12.
- 13.
14. `

Réponse Linux :

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.
- 11.
- 12.
- 13.
- 14.

## 5. Permissions

(<https://xkcd.com/149/>) Dans les systèmes multi-utilisateurs, chaque utilisateur possède certains droits lui permettant d'effectuer certaines opérations et pas d'autres. Le système d'exploitation permet de gérer ces droits. Un utilisateur un peu particulier possède tous les droits : ce "super utilisateur" est appelé "administrateur" (commande `runas` sous Windows) ou "root" (commande `sudo` sous Linux). C'est cet "administrateur" qui pourra attribuer ou retirer certains droits aux autres utilisateurs. Au lieu de gérer les utilisateurs un par un, il est possible de créer des groupes d'utilisateurs. L'administrateur attribue des droits à un groupe au lieu d'attribuer des droits particuliers à chaque utilisateur.

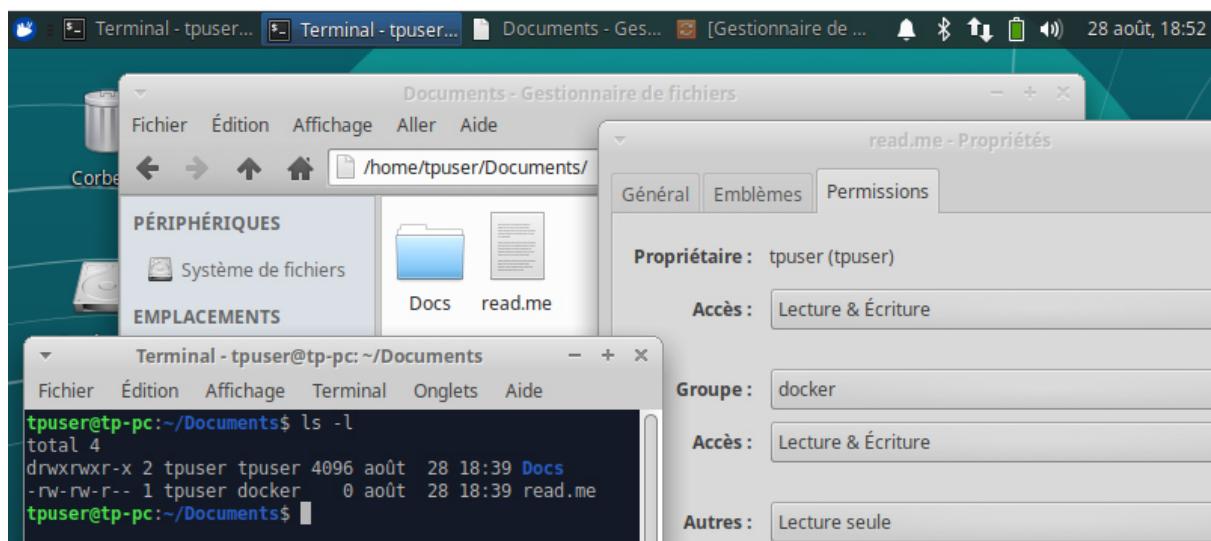


Nous nous intéresserons ici uniquement aux droits liés aux fichiers, mais vous devez savoir qu'il existe d'autres droits liés aux autres éléments du système d'exploitation (imprimante, installation de logiciels...).

## Les droits des utilisateurs sous Linux

- On définit trois type d'utilisateurs :
  - Le propriétaire du fichier (user)
  - Le groupe du propriétaire du fichier (group)
  - Les autres utilisateurs, ou encore le reste du monde (others)
- Et trois types de droits
  - r (read): droit de lecture
  - w (write): droit d'écriture, c'est à dire de modification
  - x (execute): droit d'exécution si c'est un programme, d'accès aux sous-dossiers si c'est un répertoire

Observons la capture d'écran ci-dessous



### Exercice 8 :

1. Indiquer le chemin absolu dans lequel est ouverte la console.
2. Que contient ce répertoire ?
3. Qui est le propriétaire du fichier `read.me` ?
4. Comment s'appelle le groupe du propriétaire du fichier `read.me` ?
5. Quels type de droits ce groupe a-t-il sur ce fichier ?

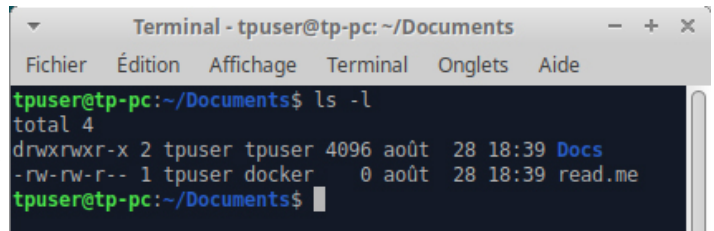
Réponses :

- 1.
- 2.
- 3.
- 4.
- 5.

## Dans la console

Observons un peu plus en détail la capture d'écran. Les droits des fichiers d'un répertoire peuvent être affichés par la commande `ls -l`. Ils apparaissent alors comme une liste de 10 symboles :

- Le premier symbole est `-` s'il s'agit d'un fichier, `d` s'il s'agit d'un dossier.
- Suivent ensuite 3 groupes de 3 symboles chacun, indiquant si le fichier (ou répertoire) est autorisé en lecture, écriture ou exécution. Les 3 groupes correspondent, dans cet ordre, aux droits du propriétaire, du groupe puis du reste des utilisateurs.



```
Terminal - tpuser@tp-pc: ~/Documents
Fichier  Édition  Affichage  Terminal  Onglets  Aide
tpuser@tp-pc:~/Documents$ ls -l
total 4
drwxrwxr-x 2 tpuser tpuser 4096 août 28 18:39 Docs
-rw-rw-r-- 1 tpuser docker  0 août 28 18:39 read.me
tpuser@tp-pc:~/Documents$
```

## Exercice 9 :

1. Analyser les symboles `drwxrwx-x` sur la troisième ligne dans la console.
2. Analyser ceux de la ligne suivante.

Réponses:

1.
  - `d` :
  - `rw` :
  - `rw` :
  - `r-x` :
1.
  - `-` :
  - `rw-` :
  - `rw-` :
  - `r--` :

## Gérer les droits d'un utilisateur sous Linux

Le propriétaire d'un fichier peut modifier les permissions d'un fichier ou d'un répertoire à l'aide de la commande `chmod`. Pour utiliser cette commande, on l'associe à des symboles pour choisir :

1. À qui s'applique le changement
  - `u` (user, utilisateur) le "propriétaire"
  - `g` (group, groupe) le "groupe propriétaire"
  - `o` (others, autres) le "reste du monde"
  - `a` (all, tous) représente l'ensemble des trois catégories
1. La modification que l'on veut faire
  - `+` : ajouter
  - `-` : supprimer
  - `=` : affectation
1. Le droit que l'on veut modifier
  - `r` : read (lecture)
  - `w` : write (écriture)
  - `x` : execute (exécution)

## Exercice 10

1. Que fait la commande `chmod a+w toto.txt` ?
2. Que fait la commande `chmod o-wx toto.txt` ?
3. Quelle commande permet d'oter la possibilité au groupe propriétaire de lire et écrire dans le fichier `toto.txt` ?

Réponses:

- 1.
- 2.
- 3.

## Exercice 11

Dans cette console de commandes, expliquer les lignes numéro :

- 1
- 3
- 4
- 6
- 7
- 11
- 12
- 16
- 17

Remarque : la commande `touch` permet de créer un fichier vide.

```
thomas@tp-pc:~$ ls
A  Bureau  Documents  Images  Modèles  Musique  Public  Téléchargements  Vidéos
thomas@tp-pc:~$ cd A
thomas@tp-pc:~/A$ ls
1
thomas@tp-pc:~/A$ touch tata.txt
thomas@tp-pc:~/A$ ls -l
total 4
drwxrwxr-x 2 thomas thomas 4096 sept. 25 21:33 1
-rw-rw-r-- 1 thomas thomas   0 sept. 29 21:13 tata.txt
thomas@tp-pc:~/A$ chmod o+x tata.txt
thomas@tp-pc:~/A$ ls -l
total 4
drwxrwxr-x 2 thomas thomas 4096 sept. 25 21:33 1
-rw-rw-r-x 1 thomas thomas   0 sept. 29 21:13 tata.txt
thomas@tp-pc:~/A$ chmod g-w tata.txt
thomas@tp-pc:~/A$ ls -l
total 4
drwxrwxr-x 2 thomas thomas 4096 sept. 25 21:33 1
-rw-r--r-x 1 thomas thomas   0 sept. 29 21:13 tata.txt
thomas@tp-pc:~/A$
```

Réponse: l'utilisateur:

- 1 :
- 3 :
- 4 :
- 6 :
- 7 :
- 11:
- 12:
- 16:
- 17:

## Exercice 12: Indiquer les commandes Linux pour

1. Créer un répertoire B dans votre répertoire /home
2. Se placer dans le répertoire B
3. Créer le fichier tutu.txt
4. Afficher les droits sur ce fichier
5. Modifier les permissions associées à ce fichier afin que les "autres utilisateurs" aient la permission "écriture".
6. Vérifier que la modification est effective

Réponses :

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

## 6. Exercices

### Exercice 13: Fichier avec python

La plupart des langages de programmation permet de manipuler les fichiers. Sous Python, l'accès aux fichiers est assuré par l'intermédiaire d'un objet-interface particulier, que l'on appelle objet-fichier. On crée cet objet à l'aide de la fonction intégrée `open()`. Celle-ci renvoie un objet doté de méthodes spécifiques, qui vont permettre de lire et écrire dans le fichier.

Exemple (à commenter) :

In [4]:

```
# Exécuter Le code ci-dessous
nomfichier='preums.txt'
fichier=open(nomfichier,'a')
fichier.write('texte1')
fichier.close()
```

1. Retrouver le fichier créé par python et l'ouvrir.
2. Dans le code ci-dessus , ajouter une instruction qui ajoute la chaîne `texte2` au fichier
3. Ecrire ci-dessous un programme qui crée le fichier `deuz.txt` , qui contiendra la chaîne `texte3`

Réponses :

- 1.
- 2.

In [1]:

```
#3.
```

## Exercice 14 : Fichiers avec python

Ecrire la fonction `table(n)` qui prend en paramètre un entier naturel `n` et qui crée 9 fichiers contenant chacun un résultat de la table de multiplication par `n` . Par exemple, `table(3)` créera 9 fichiers : le premier s'appellera `1x3.txt` et contiendra `1x3=3` , le dernier s'appellera `9x3.txt` et contiendra `9x3=27` .

In [2]:

```
#Réponse
def table(n):
    pass

table(3)
```

**FIN**