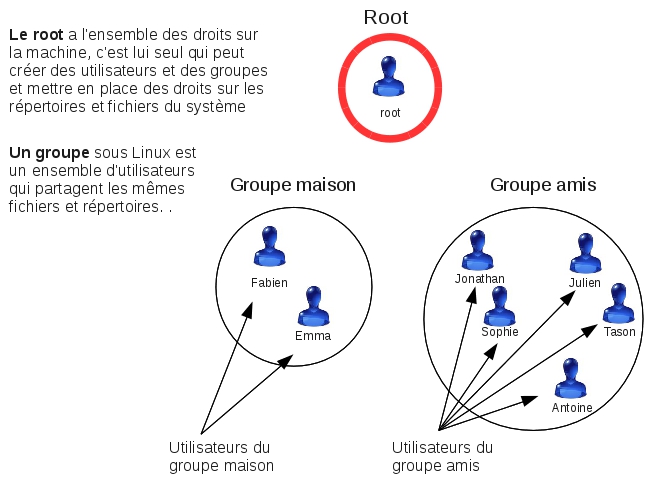
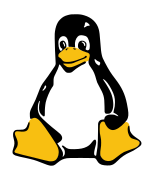
|  |  |  |
| --- | --- | --- |
| Numérique et Sciences Informatiques | | |
| 2h | **Linux** |  |
| **Objectif** : utiliser les commandes de base en ligne de commande et gérer les droits et permissions d’accès aux fichiers. | | |
| **Matériel**: PC Linux. | | |



Tux (symbole de Linux) est un manchot et non un pingouin.

**Utilisation d’une distribution Linux**

**Debian** (également connu sous le nom Debian GNU/Linux) est un système d’exploitation Linux composé exclusivement de logiciels libres, développé par le Debian Project.

La différence principale entre Windows et une distribution Linux est la façon dont sont traités les fichiers et les paramètres du système.

Par exemple si je souhaite modifier le comportement de mon système d’exploitation alors il existe un fichier qui contient les paramètres.

Ouvrez une session Linux en sélectionnant l’utilisateur **eleve** et en entrant le mot de passe **ordipass**.

Sur Linux il y a toujours par défaut un mot de passe pour se connecter à un utilisateur.

Cliquez sur **Emplacements** (barre en haut de l’écran) puis **nsi suivi d’un chiffre (ordinateur)**.

Vous êtes ici à la racine de l’ordinateur que l’on notera **"/"**, c’est ici que l’arborescence commence.

Les différents fichiers permettant le fonctionnement du système se trouvent dans les répertoires que vous voyez.

Le répertoire **media** contient les supports externes (clés USB, disques durs ...).

Le répertoire **/usr/bin** contient les applications.

Nous allons nous déplacer directement dans le répertoire **home** puis dans le répertoire **eleve**.

Le répertoire **eleve** est votre répertoire personnel, c’est ici que sont situés vos documents personnels (bureau, images, vidéos ...), on l’appelle communément le **home** (≠ répertoire home) et on le note **"~".**



**~**

**Les lignes de commandes**

Bien que Linux puisse s’utiliser comme Windows à l’aide de l’interface graphique, dans ce TP nous allons l’utiliser avec des lignes de commandes dans un terminal.

Un **terminal** est un interpréteur de commandes qui va nous permettre de contrôler l’ordinateur à l’aide de lignes de commandes plutôt qu'avec l’interface graphique.

Le terminal va nous permettre d’effectuer des actions de manière plus rapide que l'interface graphique (une fois qu’on le maîtrise).

Ouvrez un terminal : **Applications** → **Utilitaires** → **Terminal**

***Conseils pratiques :***

*Si vous appuyez sur la flèche ↑ dans le terminal, cela vous proposera les dernières commandes utilisées.*

*Si vous appuyez sur la touche tabulation après avoir tapé un début de commande, le terminal tentera de compléter votre commande automatiquement par le nom qui correspond.*

*Chaque commande s’exécutera après avoir appuyé sur la touche entrée.*

Pour commencer nous allons afficher le répertoire dans lequel nous nous situons avec la commande : **pwd** ("print working directory").

Puis nous allons afficher ce qui se trouve dans ce répertoire avec la commande : **ls** ("list").

Créer le répertoire TPlinux avec la commande : **mkdir TPlinux** ("make directory").

Faites de nouveau **ls** pour constater la création du répertoire.

Pour se déplacer dans les différents répertoires de l’ordinateur : **cd chemin** ("change directory").

Déplacez-vous à la racine de l’ordinateur : **cd /**

Vérifiez que vous êtes bien à la racine : **pwd**

Déplacez-vous dans le **home** : **cd** ~

Pour aller dans le répertoire Linux que vous avez créé si vous êtes dans votre **home** : **cd TPlinux**

Pour aller dans le répertoire Linux que vous avez créé depuis n’importe quel endroit :

**cd** **~/TPlinux** ou **cd /home/eleve/TPlinux**

Une fois dans le répertoire Linux, testez la commande : **cd ..**

Les caractères **".."** représentent le répertoire parent.

Le caractère **"."** représente le répertoire courant.

Nous allons maintenant ajouter un répertoire et un fichier dans le répertoire **Linux**, pour cela :

* Créez un répertoire appelé **monRepertoire**
* Créez un fichier appelé **Fichier1** avec la commande : **touch monFichier**

Affichez le contenu du répertoire **TPlinux** pour constater la création du répertoire **monRepertoire** et du fichier **monFichier**.

Il est aussi possible de déplacer, copier ou supprimer un fichier avec les commandes suivantes :

* Pour copier monFichier en le nommant autreFichier : **cp monFichier1 autreFichier**
* Pour supprimer Fichier2 : **rm autreFichier**
* Créez un nouveau fichier qui s’appelle **.FichierCache** en pensant bien à mettre le **"."** au début du nom.
* Affichez à nouveau le contenu du répertoire, que remarquez-vous ?

Le fichier n'apparait pas

Le **"."** sert à indiquer que le fichier est caché (il ne s’affiche donc pas dans le gestionnaire de fichiers avec la commande **ls** seule).

La commande **ls** possède des options utilisables avec la syntaxe suivante : **ls -[options]**

Il est possible de consulter les options disponibles et le fonctionnement d’une commande en utilisant le manuel : **man ls**

Affichez la liste des fichiers : **ls -a**

Affichez les informations sur les fichiers et répertoires présents dans le répertoire : **ls -l**

Ouvrez maintenant le fichier **monFichier1** avec l'applicationgedit : **gedit monFichier**

Ecrivez dans le fichier le texte "Numérique Sciences Informatique" puis enregistrez.

En retournant sur le terminal vous verrez que celui-ci est bloqué car il attend la fermeture du programme **gedit** → fermez **gedit**.

Vérifiez qu'il est possible d’afficher le contenu d’un fichier avec la commande : **cat monFichier**

Affichez à nouveau les fichiers présents dans le répertoire **TPlinux** avec la commande : **ls -l**

Dans l’ordre on retrouve les informations suivantes :

* 1ière colonne : les **droits** (sous la forme drwxr-xr-x).
* 2ième colonne : un **numéro** : 1 pour fichier et 2 pour répertoire.
* 3ième colonne : l’**utilisateur propriétaire** (eleve pour vos fichiers).
* 4ième colonne : le **groupe propriétaire** (eleve pour vos fichiers).
* ...

Un **utilisateur** Linux appartient à un ou plusieurs **groupes**, ces groupes servent à rassembler des utilisateurs afin de leur attribuer des droits communs.

* A quoi correspond la 5ième colonne sachant que sous Linux les textes sont codés en utf-8, un caractère ascii est codé sur 1 octet alors qu'un caractère accentué est codé sur 2 octets.

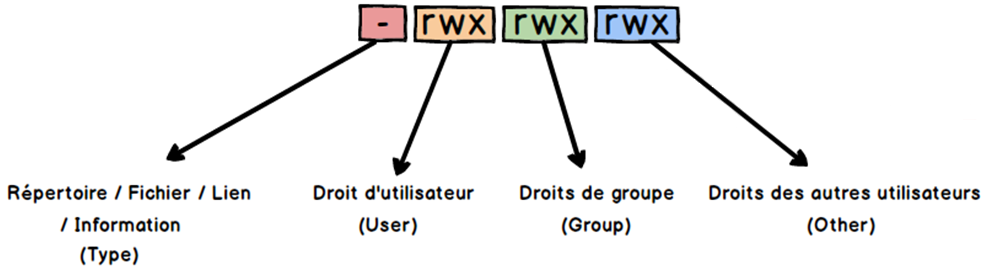
La taille du fichier en octet

**Les droits**

Sur la 1ière colonne vous devez avoir des lignes ressemblant à :

**-rw-r--r--**

**drwxr-xr-x**



Chaque ligne renseigne des droits octroyés aux fichiers et répertoires.

Le premier caractère indique le type : fichier **"-"**, répertoire **"d" ...**

Les lettres qui suivent représentent les droits :

* **r** : read (les droits de lecture).
* **w** : write (les droits d’écriture).
* **x** : executable (les droits d’exécution).

Les droits pour :

* **u** : user (l’utilisateur).
* **g** : group (les utilisateurs appartenant au même groupe).
* **o** : other (les autres utilisateurs).
* Quels sont les droits de l’utilisateur (user) pour le fichier **monFichier** ?

Lire et écrire

* Quels sont les droits pour les autres utilisateurs (others) pour le fichier **monFichier** ?

Lire

* Modifiez les droits du fichier **monFichier** pour que les users ne puissent pas écrire dans le fichier : **chmod u-w monFichier**

*u-w signifie qu'on enlève le droit w (write) à u (user).*

Quel sont les nouveaux droits de l’utilisateur (user) pour le fichier **monFichier** ?

Lire

* D’après vous quelle commande permet d'ajouter le droit w (write) à g (group) ?

chmod g+w monFichier

On a utilisé la commande **chmod** comme suit : **chmod [ugoa] [+-=] [rwx] monFichier**

Avec **a=u+g+o**

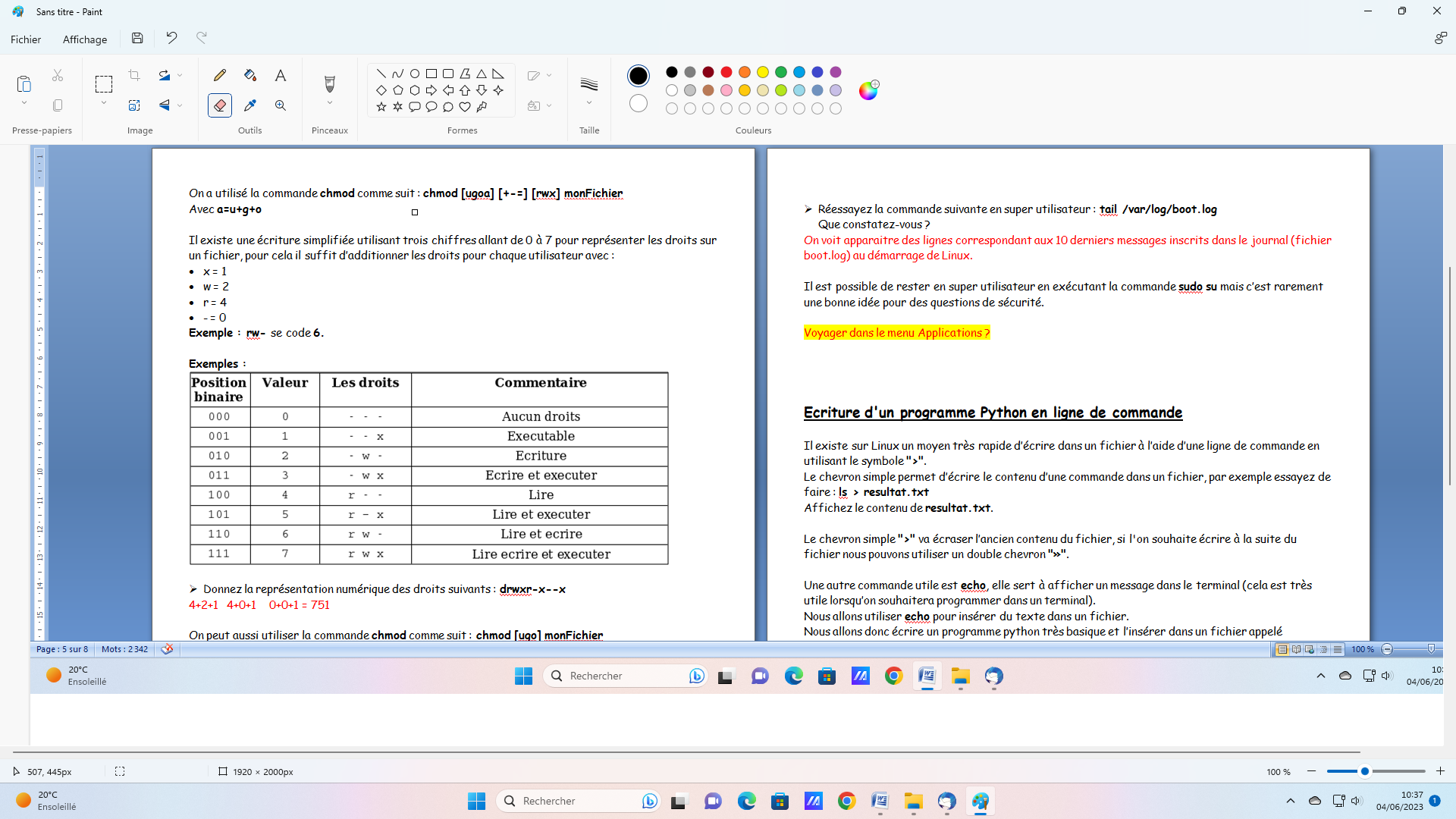
Il existe une écriture simplifiée utilisant trois chiffres allant de 0 à 7 pour représenter les droits sur un fichier, pour cela il suffit d’additionner les droits pour chaque utilisateur avec :

* x = 1
* w = 2
* r = 4
* - = 0

**Exemple : -rw-r--r--**

* rw- : 4+2=6
* r-- : 4+0+0=4
* r-- : 4+0+0=4

→ commande : **chmod 644 monFichier**

****

* Donnez la représentation numérique des droits suivants : **drwxr-x--x**

chmod 751 monRepertoire

* Donnez la représentation numérique des droits suivants : **rwx r-- r--**

chmod 744 monFichier

* Testez cette façon de faire pour obtenir les propriétés **rwx r-- r--** sur **monFichier**

chmod

* Essayez la commande suivante pour afficher le contenu du fichier boot.log : **tail /var/log/boot.log**,

Quel message avez-vous ?

Permission non accordé

* Faites ensuite : **ls -l /var/log/boot.log**

Les droits sur le fichier boot.log sont attachés à quel utilisateur ?

root

**root** est un utilisateur spécial sur Linux, c'est un super utilisateur.

**root** est par défaut le seul à pouvoir agir sur les fichiers de configuration du système comme boot.log.

Pour exécuter une commande en super utilisateur (**root** aura tous les droits) nous allons utiliser la commande **sudo** avant la commande que l’on souhaite exécuter.

Pour passer en super utilisateur un mot de passe vous sera demandé, le mot de passe **root** pour les ordinateurs est **ordipass** (attention quand vous rentrerez le mot de passe vous ne le verrez pas apparaitre à l'écran : confidentialité oblige).

* Réessayez la commande suivante en super utilisateur : **tail /var/log/boot.log**

Que constatez-vous ?

Il est possible de rester en super utilisateur en exécutant la commande **sudo su** mais c’est rarement une bonne idée pour des questions de sécurité.

**Ecriture d'un programme Python en ligne de commande**

Il existe sur Linux un moyen très rapide d’écrire dans un fichier à l’aide d’une ligne de commande en utilisant le symbole **">"**.

Le chevron simple permet d’écrire le contenu d’une commande dans un fichier, par exemple essayez : **ls > resultat.txt**

* Affichez le contenu de **resultat.txt** avec **gedit**.

Le chevron simple **">"** va écraser l’ancien contenu du fichier, si l'on souhaite écrire à la suite du fichier nous pouvons utiliser un double chevron **"»"**.

Une autre commande utile est **echo**, elle sert à afficher un message dans le terminal (cela est très utile lorsqu’on souhaitera programmer dans un terminal).

Nous allons utiliser **echo** pour insérer du texte dans un fichier.

Nous allons donc écrire un programme python très basique et l’insérer dans un fichier appelé **Programme.py**.

* Essayez la commande suivante : **echo "print(’hello world’)" > Programme.py**
* Exécutez le programme en utilisant l’interprète Python : **python3 Programme.py**

Pour finir nous allons voir la puissance du terminal en écrivant un court programme qui va créer 1000 fichiers numérotés de 0 à 999 contenants chacun la liste des fichiers présents dans le répertoire au moment de sa création.

Pour cela nous allons utiliser des variables et des boucles dans le terminal.

* Créez un répertoire qui s’appelle **Rep** puis écrivez la ligne suivante, en langage bash, dans le terminal : **for i in {0..999}; do ls -l Rep > Rep/Fichier$i; done**
* Vérifiez le contenu du répertoire **Rep**.

Enfin pour terminer et rendre l’ordinateur dans le même état que lorsque vous êtes arrivés on va nettoyer ce que l’on a fait.

* Utilisez la commande **rm** avec l’option **-r** (qui permet de supprimer un répertoire et tout son contenu) pour supprimer le répertoire **TPlinux**.