


## 1 Utiliser l'environnement de bureau Gnome 3

Ici, nous verrons qu'il n'y a pas besoin d'être informaticien pour utiliser Linux (avec Gnome 3) au quotidien. Par la suite, nous verrons que le terminal nous apporte cependant plus de puissance et de rapidité.

### Exercice 1 : Le navigateur de fichiers et le visionneur evince

1. Lancer le navigateur de fichiers de Gnome 3 (cherchez-le dans les activités). En français, il s'appelle simplement « Fichiers » (son ancien nom est *nautilus*). Vous pouvez aussi appuyer sur la touche « Windows » (que certains appelle aussi **Super** pour éviter la référence à Microsoft) puis taper le début de Fichiers puis .
2. Au départ, ce qu'on vous montre est votre *répertoire personnel*, c'est chez vous ! Vous pouvez y faire à peu près tout ce que vous voulez. Quels fichiers et répertoires sont déjà là ?
3. Créez un nouveau répertoire appelé `tp0` (indice : clic-droit).
4. Dans `firefox`, télécharger le fichier `tp0_milieu.pdf` (celui que vous êtes en train de lire). Où est-il ? Le trouvez-vous dans votre répertoire personnel ?
5. Déplacez ce fichier dans `tp0`, s'il n'y est pas déjà (par exemple, vous pouvez utiliser « couper » en faisant un clic droit dessus puis le « coller » dans ce répertoire `tp0`).
6. Double-cliquez sur `tp0_milieu.pdf` pour l'afficher.
7. Essayez de renommer `tp0_milieu.pdf` pour l'appeler simplement `tp0-2.pdf`.
8. Créer une copie de `tp0-2.pdf` puis supprimer cette copie. Voir le contenu de la « Corbeille » et la vider.

On s'arrête là pour Fichiers, si vous voulez l'utiliser d'avantage, vous pouvez regarder plus attentivement les paramètres et les raccourcis claviers à un autre moment.

--- \* ---

### Exercice 2 : gedit : l'éditeur de texte de Gnome 3

1. Ouvrir l'éditeur de texte `gedit` aussi appelé simplement « Éditeur de texte » dans Gnome 3.
2. **Recopier** le texte suivant dans la fenêtre de l'éditeur, ne faites pas de copier-coller, il faut s'entraîner à écrire des caractères spéciaux !  
Respecter les espaces et les sauts de ligne.

```
# Notes sur le TP0

## Fichiers

Avec *Fichiers*, on peut facilement :

- *créer* un nouveau répertoire
- **déplacer** des fichiers avec couper/coller
- renommer un fichier
- ~~écrire du texte dans un fichier~~ [^1]

[^1]: non, pour ça il faut un éditeur de texte
```


3. Sauvegarder dans votre répertoire `tp0` le fichier en lui donnant le nom `notes_tp0.md` (attention à l'extension `.md`). Remarquer la coloration syntaxique après la sauvegarde : l'extension `.md` a permis à `gedit` de comprendre que le format du fichier est *markdown* (un petit format bien pratique et très à la mode de mise en forme du texte).
4. Configurer `gedit` avec les options indispensables, dans l'éditeur, cliquer sur l'icône en haut à droite avec les trois traits horizontaux, puis sur « préférences » et :
  - dans « Affichage », cocher numéros de ligne et marge à droite ;
  - dans « Éditeur », cocher l'indentation automatique.
5. Fermer l'éditeur de texte.

--- \* ---

## 2 Utiliser le shell dans un terminal

On peut faire beaucoup de choses avec l'environnement graphique, mais on finit par être un peu limité. Nous allons apprendre à utiliser un interpréteur de commandes appelé *shell* dans un *terminal*. Ici, nous n'entrons pas dans les détails et vous donnons juste un aperçu. Nous approfondirons tout ça dans le cours « Initiation à l'environnement Unix ».

### Exercice 3 :

1. Ouvrir l'application « Terminal ».
2. Une commande est une suite de mots séparés par des blancs et terminée par . Le caractère `$` en début de ligne représente l'invite de commande (*prompt*), il n'est pas à recopier.
  - a) Pour voir votre nom d'utilisateur, entrer :

```
$ id -un
```

(Mnémonique : *IDentifier User Name*)
  - b) Pour écrire la date et l'heure sur l'écran, entrer :

```
$ date
```
  - c) Pour voir dans quel répertoire vous vous trouvez, entrez

```
$ pwd
```

(Mnémonique : *Print Working Directory*)  
Comparer avec :

```
$ echo ~
```

pour vérifier qu'il s'agit bien de votre répertoire personnel.
  - d) Pour voir le contenu du répertoire courant, entrer :

```
$ ls
```

(Mnémonique : *LiSt*)  
Comparer avec ce qui est affiché dans « Fichiers ».
3. On va « se déplacer » dans `tp0`.

```
$ cd tp0
```

(Mnémonique *Change Directory*, changer de répertoire)

4. Listez les fichiers de ce répertoire. Il devrait y avoir au moins **notes\_tp0.md** et le sujet du TP.
5. On va générer un fichier **pdf** avec vos notes en utilisant le logiciel (non standard et très puissant) **pandoc**  

```
$ pandoc notes_tp0.md -o notes_tp0.pdf
```

Explication : on demande à **pandoc** de lire le fichier **notes\_tp0.md** (donc un fichier au format *markdown*, vu l'extension) pour créer (-o comme *output*, sortie) le fichier **notes\_tp0.pdf** (donc un fichier au format **pdf**, vu l'extension).  
Lister les fichiers pour vérifier qu'un nouveau fichier a été créé.
6. On ouvre ce fichier avec **evince** :  

```
$ evince notes_tp0.pdf
```

  
Vous devriez voir vos notes mises en forme au format **pdf**.
7. Pouvez-vous entrer de nouvelles commandes dans le terminal ? En fait le terminal attend par défaut la fin de la commande. Fermez la fenêtre **evince**. Maintenant pouvez-vous entrer des commandes dans le terminal ?
8. C'est bien embêtant de devoir fermer à chaque fois la fenêtre... Lancer cette fois (noter le **&** à la fin)  

```
$ evince notes_tp0.pdf &
```

  
On dit que la commande a été lancée *en arrière-plan*. Pouvez-vous lancer des commandes maintenant ?
9. On va ajouter un lien vers l'aide de « Fichiers » dans nos notes. Pour éditer le fichier, on lance directement dans le terminal :  

```
$ gedit notes_tp0.md &
```

  
(On le lance en arrière-plan, pourquoi ?)
10. Ajouter à la fin du fichier une ligne vide puis :

```
Plus d'information  
[ici] (https://help.gnome.org/users/gnome-help/stable/files.html.fr).
```

Sauvegarder. Utiliser **Alt** + **←** pour revenir sur le terminal.

11. Relancer la commande **pandoc** précédente : utiliser **↑** jusqu'à la faire apparaître puis **↵**. Vérifier (**Alt** + **←**) que le document **pdf** a changé dans **evince**. Cliquer sur « ici » et voir le résultat dans **firefox**.
12. En fait, on peut même générer du HTML avec **pandoc**. Entrer les commandes  

```
$ pandoc notes_tp0.md -o notes_tp0.html  
$ firefox notes_tp0.html &
```

  
Vérifier que le nouveau fichier existe et le voir dans **firefox**, cliquer sur le lien (« ici ») pour vérifier qu'il fonctionne.
13. Pour finir, prendre des notes sur cet exercice dans votre fichier **notes\_tp0.md**, mettre à jour les fichiers **pdf** et **html** correspondants et voir les résultats dans **evince** et **firefox**. Utiliser au maximum les raccourcis claviers.

--- \* ---

### Exercice 4 : Plus d'opérations sur les fichiers

1. Vous devriez être dans votre répertoire `tp0`. Pour revenir au répertoire parent (qui devrait être votre répertoire personnel), vous pouvez utiliser la commande  

```
$ cd ..
```

En fait, `..` représente toujours le répertoire parent. Vérifier que vous êtes bien dans votre répertoire personnel.
2. Entrer la commande  

```
$ cal 2023
```

pour avoir un calendrier de l'année en cours.
3. On a envie de sauvegarder ce calendrier (peut-être pour l'imprimer ou autre...). On va rediriger la sortie de la commande précédente vers un fichier.  

```
$ cal 2023 >calendrier_2023.txt
```

Vérifier que le fichier `calendrier_2023.txt` existe bien, voir son contenu avec `gedit`.
4. On va faire une copie de ce calendrier :  

```
$ cp calendrier_2023.txt cal_copie.txt
```

(Mnémonique : *CoPy*)  
Vérifier que le nouveau fichier est créé. L'ouvrir avec `gedit`. Dans `cal_copie.txt`, remplacer la première ligne par `Calendrier 2023` et sauvegarder. Vérifier que `calendrier_2023.txt` n'a pas été modifié (donc qu'on a bien travaillé sur une copie).
5. On va créer un répertoire pour y mettre nos calendriers.  

```
$ mkdir cals
```

(Mnémonique : *MaKe DiRectory*, fabriquer un répertoire)  
Vérifier qu'un nouveau répertoire est créé dans le terminal et aussi avec Fichiers.
6. On va y déplacer nos calendriers.  

```
$ mv calendrier_2023.txt cal_copie.txt cals
```

(Mnémonique : *MoVe*)  
Vérifier en ligne de commande avec  

```
$ ls
```

```
$ ls cals
```

La deuxième commande liste le contenu du répertoire `cals` au lieu du répertoire courant. Voir aussi avec Fichiers.
7. En fait notre copie n'est pas très utile... On va la supprimer. Se déplacer dans le répertoire `cals` puis entrer  

```
$ rm cal_copie.txt
```

(Mnémonique : *ReMove*)  
Voir le résultat. **Attention ! Contrairement à ce qui se passait dans Fichiers, les fichiers sont définitivement supprimés ! La commande `rm` ne met pas les fichiers dans une corbeille !**
8. Revenir dans votre répertoire personnel. En fait, notre répertoire `cals` n'est pas très utile non plus... Voir que la commande suivante échoue.  

```
$ rm cals
```

Lire le message d'erreur. Taper

```
$ rm --help
```

et chercher l'option qui permet de supprimer les répertoires et tout ce qu'ils contiennent, puis l'utiliser pour supprimer le répertoire `cals`.

--- \* ---

### **3 Reprendre le sujet imprimé (« Initiation à Linux : première partie ») à l'exercice 3**

Pour ceux qui n'ont pas lu le titre de la section : reprendre le sujet imprimé (« Initiation à Linux : première partie ») à l'exercice 3.