**Exercice 01 – Cours 04  
Opérations sur les fichiers et répertoires**

**Objectifs**

Dans cet exercice, vous vous familiariserez avec les différentes commandes Linux qui permettent de manipuler les fichiers et les répertoires.   
Souvenez-vous que, sous Linux, tout est fichier.   
Un fichier peut contenir du texte ou un programme exécutable, être un répertoire, voire même un périphérique tel qu'un disque ou un port USB.

Pour manipuler les fichiers et les répertoires, nous utiliserons fréquemment les commandes **ls**, **cp**, **mv**, **mkdir** et **rm**.

**Rappel** …  
On peut toujours obtenir de l’aide sur une commande à l’aide des modules d’aide.  
Comme exemple, pour obtenir de l’aide sur la commande uname …   
**>> uname --help**

**Déroulement**

**Étape 01**La commande ls permet de lister les fichiers.   
Elle accepte un nombre arbitraire d'arguments, puis elle liste le contenu de chacun des répertoires qui lui ont été fourni ...  
**>> ls <Options> ... <Fichier> ...**

Utilisée seule, sans option ni argument, la commande ls liste les fichiers du répertoire courant …  
**>> ls**

**Étape 02**  
Pour lister tous les fichiers de votre répertoire personnel, incluant les fichiers cachés, utilisez le commutateur **--all** (forme courte **-a**).  
**>> ls --all**

On peut aisément repérer les fichiers cachés : ils débutent par le caractère point . .

**Étape 03**  
Pour obtenir davantage d'informations sur les fichiers, comme par exemple le type, le propriétaire ou la date de dernière modification, on utilise le commutateur **-l** de la commande **ls**.

Nous allons maintenant lister le contenu du répertoire **/usr/lib/openssh** au format détaillé …  
**>> ls -l /usr/lib/openssh**

**Remarque** …  
Il n’existe pas vraiment de forme longue pour le commutateur **-l**.

Le premier champ indique le type de fichier avec les permissions …  
**-rwxr-xr-x 1 root root 105608 mar 4 2019 sftp-server**

Le deuxième champ indique le nombre de liens physiques (*hard links*) vers le fichier …  
**-rwxr-xr-x 1 root root 105608 mar 4 2019 sftp-server**

Les troisième et quatrième champs représentent respectivement l'utilisateur propriétaire et le groupe propriétaire du fichier …  
**-rwxr-xr-x 1 root root 105608 mar 4 2019 sftp-server**

Le cinquième champ indique la taille en octet du fichier …  
**-rwxr-xr-x 1 root root 105608 mar 4 2019 sftp-server**

Le sixième champ représente la date de dernière modification (horodatage) du fichier …  
**-rwxr-xr-x 1 root root 105608 mar 4 2019 sftp-server**

Finalement le dernier champ indique le nom du fichier ou du répertoire …  
**-rwxr-xr-x 1 root root 105608 mar 4 2019 sftp-server**

La sortie de la commande **ls -l** est présentée en ordre alphabétique du nom de fichier.   
Il existe d'autres options qui permettent de trier la sortie de la commande **ls -l**, notamment l'option **-S** pour trier par taille et **-t** pour trier par date de modification.

**Étape 04**  
Nous allons maintenant afficher la liste des fichiers du répertoire **/etc/ssh** au format long et triée par taille …  
**>> ls -lS /etc/ssh**

**Étape 05**  
Pour inverser le tri de la commande lancée à l'étape 4, on ajoutera le commutateur **--reverse** (forme courte -r) …  
**>> ls -lS --reverse /etc/ssh**  
ou  
**>> ls -lSr /etc/ssh**

**Étape 06**  
L'option **--recursive** (forme courte **-R**) de la commande **ls** est utilisée pour lister de manière récursive le contenu d'un répertoire.   
Cette option aura pour effet de lister non seulement le contenu d'un répertoire, mais également le contenu des   
sous-répertoires et de leurs sous-répertoires …  
**>> ls --recursive /usr/lib/systemd**

**Étape 07**  
Pour lister le répertoire lui-même, sans afficher son contenu, on utilise le commutateur **--directory** (forme courte **-d**). Cette option est souvent combinée avec le commutateur **-l** pour afficher les permissions d'un répertoire spécifique …  
**>> ls --directory /usr/lib**

**Étape 08**

Certains caractères d'englobement (*wildcard*), comme par exemple l'astérisque **\***, sont utilisés avec la commande **ls**.   
Par exemple, pour lister tous les fichiers qui commencent par la lettre **l** dans le répertoire **/etc** …  
**>> ls /etc/l\***

Remarquez ici que le caractère **\*** fait en sorte que ls répertorie de manière récursive.   
Autrement dit, elle liste le contenu des répertoires et des sous-répertoires qui débutent par la lettre **l**.

**Étape 09**  
Pour empêcher la commande ls de lister le contenu des répertoires dans l'exemple précédent, on utilise le commutateur **--directory** (forme courte **-d**) …  
**>> ls --directory /etc/l\***

**Étape 10**

La commande **file** permet d'inspecter le contenu d'un fichier pour indiquer à l'utilisateur quel est son type.   
Par exemple, pour vérifier le type du fichier **/etc/hosts** ...   
**>> file /etc/hosts**

Plusieurs commandes sous Linux exigent un certain type de fichier pour s'exécuter correctement, notamment le type texte.   
La commande file peut être utilisée pour valider qu'un fichier est du bon type avant de le traiter avec une commande.

**Étape 11**

Pour vérifier le type de fichier de **/usr/bin/passwd**…  
**>> file /usr/bin/passwd**

La sortie de la commande indique que le fichier est de type **Executable Link Format** (ELF), ce qui n'est pas un fichier texte.

**Étape 12**

La commande **touch** est utilisée pour créer un nouveau fichier vide ou mettre à jour l'horodatage d'un fichier existant.  
On l'utilise comme suit …  
**>> touch <Option(s)> ... <Fichier> ...**

Si le fichier spécifié en argument n'existe pas, alors la commande crée un nouveau fichier …  
**>> touch nouveau**  
**>> ls -l nouveau**

**Étape 13**  
Si le fichier existe déjà, alors l'horodatage du fichier est mis à jour sans modifier son contenu ...À  
**>> touch nouveau**  
**>> ls -l nouveau**

**Étape 14**  
La commande **cp** sert à copier des fichiers.   
Elle est analogue au **copier/coller** en environnement GUI.  
**>> cp <Option(s)> ... <Source(s)> <Destination(s)>**

La commande **cp** prend au minimum deux arguments ...

* le premier argument est le(s) fichier(s) à copier   
  et
* le second argument est la destination de la copie.

Pour copier le fichier **/etc/hosts** dans votre répertoire personnel …  
**>> cp /etc/hosts ~**

**Rappel** …  
**~** représente le répertoire de l’utilisateur en cours.

/

**Étape 15**

Pour copier le fichier **/etc/hosts** dans votre répertoire personnel et le renommer à destination comme **hosts.original** …  
**>> cp /etc/hosts ~/hosts.original**

**Étape 16**  
Pour lister et ensuite copier le répertoire **/etc/network** vers le répertoire personnel de notre utilisateur ...  
**>> cp /etc/network/ ~**

**Étape 17**  
Sans option, la commande **cp** omet les répertoires ; elle copie seulement les fichiers.   
Remarquez que rien n'a été copié suite à la commande précédente.

Pour copier les répertoires, il est essentiel d'ajouter le commutateur **--recursive** (forme courte **-r**) à la commande **cp**.  
**>> cp --recursive /etc/network/ ~**

**Étape 18**  
La commande **mv** permet de déplacer un ou des fichier(s) d'un emplacement vers un autre dans le système de fichiers.  
**>> mv <Option(s)> ... <Source(s)> <Destination>**

Nous allons d'abord créer le répertoire **~/documents/exercices/cours04** ...mv  
**>> mkdir -p ~/documents/exercices/cours04**

Nous allons par la suite créer un nouveau fichier …  
**>> touch ~/nouveau.texte**

Pour déplacer le fichier **~/nouveau.texte** vers le répertoire **~/documents/exercices/cours04** ...  
**>> mv ~/nouveau.texte ~/documents/exercices/cours04**

**Étape 19**  
La commande **mv** permet de déplacer plusieurs fichiers d'un coup vers un nouvel emplacement.   
Par exemple, nous allons déplacer les fichiers **hosts.original**, **network/** et nouveau,texte vers le répertoire documents/exercices/cours04/.   
Le commutateur **--verbose** (forme courte **-v**) de la commande **mv** permet d'afficher à l'écran ce que la commande effectue …  
**>> mv --verbose nouveau.texte hosts.original network/ documents/exercices/cours04/**

**Remarque importante** …  
Le commutateur **--recursive** (forme courte **-r**) n'est pas nécessaire avec la commande **mv** pour déplacer des répertoires.

**Étape 20**  
La commande **mv** est également utile pour renommer un ou des fichiers ou un ou des répertoires.   
Par exemple, pour renommer le répertoire **~/documents/exercices/** pour **~/documents/exercices-pratiques/ …**  
**>> mkdir--verbose documents/exercices/**   
**>> mv --verbose documents/exercices/ documents/exercices-pratiques**

**Remarque** …  
Le caractère tilde **~** est facultatif ici puisque notre répertoire courant est notre répertoire personnel.

**Étape 21**  
La commande **rm** permet de supprimer des fichiers et des répertoires.

Tentez de supprimer le répertoire **exercices-pratiques** comme suit ...  
**>> rm documents/exercices-pratiques/**   
**rm: impossible de supprimer 'documents/exercices-pratiques/': est un dossier**

On remarque que la commande rm seule refuse de supprimer un répertoire.   
Il faut donc ajouter le commutateur **--recursive** (forme courte **-r**) pour supprimer un répertoire ...  
**>> rm --recursive documents/exercices-pratiques/**

**Englobement simple**

**Étape 01**Dans votre répertoire personnel, créez un répertoire nommé **glob** puis déplacez-vous à l'intérieur de ce répertoire ...  
**>> mkdir --verbose glob**   
**>> cd glob/**

**Étape 02**Nous allons maintenant créer quelques fichiers d'exercices.   
Pour ce faire, nous allons utiliser une nouvelle astuce du shell, les accolades {} pour désigner un intervalle.   
Les accolades ne font pas partie de l'englobement.

Lancez la commande suivante pour créer plusieurs fichiers vides en une seule commande ...  
**>> touch data\_{0..10}.txt**  
**>> touch data\_{a..d}.txt**  
**>> touch archives{2000..2022}.tar.gz**  
  
Listez ensuite le contenu du répertoire glob ...  
**>> ls**

**Étape 03**Nous allons maintenant utiliser l'englobement avec le métacaractère **\*** pour lister tous les fichiers dont le nom débute par **data\_ suivi** de n'importe quoi et de suffixe **.txt** ...  
**>> ls data\_\*.txt**

**Étape 04**Remplacez le caractère **\*** de la commande précédente par **?** ...  
**>> ls data\_?.txt**

Constatez l'absence d'un fichier ici: celle de **data\_10.txt** qui n'a pas satisfait l'englobement **data\_?.txt**.   
Cela est dû au fait que le caractère **?** ne représente qu’un et un seul caractère.

**Étape 05**Pour lister tous les fichiers dont le nom débute par **data\_ suivi** d'exactement deux caractères et du suffixe **.txt** ...  
**>> ls data\_??.txt**

Seul le fichier **data\_10.txt** satisfait cette expression.

**Étape 06**Les crochets sont utilisés pour désigner un ensemble de caractères dans une expression d'englobement.

Pour lister tous les fichiers débutant par **data\_ suivi** des chiffres **2**,**3** ou **8** et du suffixe **.txt** ... **>> ls data\_[238].txt**

**Étape 07**Il est également possible de spécifier une intervalle de chiffres (et non de nombres !) et de lettres entre crochets **[ ]**.

Pour lister tous les fichiers débutant par **data\_ suivi** des chiffres entre **0** et **6** et du suffixe **.txt** ...  
**>> ls data\_[0-6].txt**

**Étape 08**La négation peut également être utilisée dans une expression d'englobement pour exclure certains résultats.

Pour lister tous les fichiers débutant par **data\_ suivi** d'une valeur autre que **0**,**1**,**2**,**3**,**4**,**5** et **6** et du suffixe **.txt** ...  
**>> ls data\_[!0-6].txt**

**Questions de révision et d'exploration**

**Question 01** (**chemin relatif et absolu**)

* Ouvrez un terminal.   
  Quel est votre répertoire courant ?   
  Quelle commande vous permet de le savoir ?  
  **/home/utilisateur comande pwd --------------**
* Déplace-vous dans votre répertoire personnel.   
  Comment pouvez-vous le faire en utilisant le chemin relatif spécial ~ ?  
  **--------------**
* Trouvez la commande pour se déplacer dans votre répertoire personnel, mais avec son chemin absolu.  
  **-~-------------**
* À partir de votre répertoire personnel, comment pouvez-vous vous déplacer dans le répertoire **/home** à l'aide de son chemin relatif ?  
  **------cd .. --------**
* À partir de votre répertoire personnel, comment pouvez-vous vous déplacer à la racine **/** à l'aide de son chemin relatif ?  
  **--------------cd ../..**

**Question 02** (**se déplacer dans l'arborescence**)  
Observez la structure de répertoire suivante.   
  
**/  
├── etc/  
│ ├── network/  
│ │ └── interfaces  
│ ├── systemd/  
│ │ ├── resolved.conf  
│ │ ├── system/  
│ │ ├── system.conf  
│ │ ├── user/  
│ │ └── user.conf  
│ └── udev/  
│ ├── rules.d/  
│ └── udev.conf  
└── home/  
 └── etudiant/  
 ├── documents/  
 ├── code/  
 └── téléchargements/**

Entrez la commande la plus courte possible pour chacune des questions suivantes.

* Votre répertoire courant est la racine (**/**).   
  Entrez la commande pour se déplacer vers le répertoire code dans votre répertoire personnel.  
  **-------cd /home/admin-------**
* Votre emplacement actuel est la racine (**/**).   
  Entrez la commande pour accéder au répertoire nommé **network**.  
  **---------cd /etc/network-----**
* Votre emplacement actuel est documents.   
  Accédez au répertoire nommé **home**.  
  **------cd /home--------**
* Votre emplacement actuel est **systemd**.   
  Accédez au répertoire nommé **network**.  
  **-----cd network/---------**

**Question 03** (**commande mv**)  
La commande mv permet de déplacer et de renommer un fichier.

* Déplacez-vous dans votre répertoire personnel (~) et créez un répertoire nommé s3.   
  Dans le répertoire s3, créez ensuite un fichier nommé q1.txt.  
  **------mkdir s3----cd s3/--touch q1.txt--**
* Utilisez la commande **mv** pour renommer le répertoire **s3** pour **semaine-3**.  
  **-------mv s3/ semaine-3-------**
* À partir de votre répertoire personnel, renommez le fichier **q1.txt** pour **question1.txt**.   
  (Ne changez pas votre répertoire courant pour **semaine-3**.   
  Ne lancez pas **cd semaine-3** avant de renommer le fichier. Faites-le à partir de votre répertoire courant.)  
  **-------** mv semaine-3/q1.txt question1.txt
* Déplacez le répertoire **semaine-3** dans **documents/**.  
  **------mv semaine-3/ /home/admin/documents--------**

**Question 4** (**commande cp**)  
La commande cp permet de copier un fichier ou un répertoire.

* Déplacez-vous dans votre répertoire personnel (**~**) et créez un répertoire nommé sauvegarde.  
  Utilisez maintenant la commande **cp** pour faire une copie du répertoire documents dans le répertoire sauvegarde.  
  **---mkdir sauvegarde ----cp /documents sauvegarde/-------**
* Lancez une seconde fois la commande **cp** pour faire une copie du répertoire documents dans le répertoire sauvegarde.   
  Que s'est-il passé ?   
  (indice: ajoutez l'option **--verbose** à la commande **cp**).  
  **------ cp –v /documents sauvegarde/ -----vous avez oublié que document est un repertoire---**
* Lancez une troisième fois la commande **cp** pour faire une copie du répertoire documents dans le répertoire sauvegarde.   
  Cette fois, ajoutez le commutateur **--interactive** (forme courte **-i**) à la commande.   
  Que se passe-t-il ?  
  **-------- cp –v -r /documents sauvegarde/ ------**

'documents/exercices/cours04/nouveau.txt' -> 'sauvegarde/documents/exercices/cours04/nouveau.txt'

'documents/exercices/cours04/fichier1' -> 'sauvegarde/documents/exercices/cours04/fichier1'

'documents/exercices/cours04/fichier3' -> 'sauvegarde/documents/exercices/cours04/fichier3'

'documents/exercices/cours04/fichier2' -> 'sauvegarde/documents/exercices/cours04/fichier2'

**Question 05** (**commande rm**)  
Utilisez maintenant la commande **rm** pour supprimer le répertoire sauvegarde que vous avez créé à la question précédente.  
Quelle commande avez-vous saisie ?  
**----rm –r sauvegarde/----------**

**Question 06** (**englobement**)  
Pour chacune des expressions suivantes, fournissez au moins trois exemples de noms de fichiers correspondant aux expressions.

Par exemple ...

|  |  |
| --- | --- |
| Expression | Exemples |
| \*e | **Marie, louise, mère** |

|  |  |
| --- | --- |
| Expression | Exemples |
| h\* | Henri hermann hermes |
| j??? | Jean-juls-junior, jjjk,jjjo |
| br[a-z][a-z]d |  |
| h\*[!a-g] |  |

**Question 07** (**englobement**)

Trouvez une expression d’englobement pour les chaînes suivantes.   
Utilisez les crochets [] et le point d'interrogation ? dans au moins une réponse.

|  |  |
| --- | --- |
| Exemples | Expression |
| Fichier\_A, GroupeA, Test-A | \*A |

|  |  |
| --- | --- |
| Exemples | Expression |
| Wind, windy, cindy |  |
| fichier1, fichier2, fichier3 |  |
| Gap, GRp, GCC |  |
| test1,table2,tree8 |  |

Avec la collaboration de **Miguel Grandmont Champagne**.