

Système d'exploitation (partie 1)

Rentrer le code suivant <https://cocalc.com/app?project-invite=PChXzGVZ5e2C9wug> ou passez par le site www.dimension-k.com et cliquer sur le lien CoCalc du chapitre 10 de la page NSI pour pouvoir vous connecter au serveur linux de la classe. Si c'est votre première connexion il vous faudra vous créer un compte.

Une fois que c'est fait relancer le lien, acceptez l'invitation, puis lancer le serveur linux (c'est le gros + entouré qui est dans la colonne d'icônes à gauche.

A la découverte de notre environnement

Exercice 1

- 1) Utiliser la commande « **uname -o** » puis la commande « **uname -r** » pour déterminer le système d'exploitation utilisé par le serveur Cocalc dans lequel vous allez travailler.
- 2) Grâce à la commande « **man uname** », consulter la page de manuel de la commande « **uname** » et expliquer à quoi correspondent les options « **-o** » et « **-r** ».

Remarque : Pour sortir du manuel, il suffit d'appuyer sur la lettre « **q** ».

- 3) Grâce à la commande « **lsb_release -a** », retrouver la distribution du système d'exploitation utilisée par le serveur.
- 4) Suite aux résultats obtenus dans les questions 1 et 3, expliquer les notions de « système d'exploitation », « noyau » et « distribution ».

Système de fichiers

Exercice 2 :

- 1) Lire dans le cours l'utilité de la commande **ls**.
- 2) Taper la commande « **ls /home/Prof** », dire ce que vous obtenez comme réponse et dire à quoi ça correspond.
- 3) Lire dans le cours l'utilité de la commande **tree**.
- 4) Observer ce qu'on obtient en tapant « **tree /home/user** »
- 5) Taper « **tree /** » (vous pouvez interrompre l'affichage avec ctrl+c) quel est le problème ?
- 6) Taper « **tree -L 1 -nF /** ». Que se passe-t-il ?
- 7) En vous aidant de la page de manuel de la commande « **tree** », expliquer les options « **-L 1** » « **-nF** » « **/** ».

Exercice 3 :

- 1) Grâce à la commande « **tree** » vue dans l'exercice 2, afficher toute l'arborescence de fichiers du répertoire « **/home/user/** « votre prénom ».
- 2) Grâce à la commande « **mkdir** », dans le répertoire portant votre prénom créer un répertoire « **tpSE1** » pour le TP. En utilisant la commande « **ls** », vous pourrez vérifier que le répertoire est bien créé.

La commande qui permet de se déplacer dans les répertoires se nomme « **cd** » (Change Directory).

- 3) En utilisant la commande « **cd** », entrer dans le répertoire « **tpSE1** », puis avec la commande « **pwd** » vérifier que le déplacement a bien eu lieu.

Rappel : le répertoire dans lequel on se trouve actuellement est appelé « répertoire courant ».

- 4) Taper la commande « **cd ..** » puis « **pwd** ». Que constate-t-on ? Que peut-on en déduire pour le paramètre « **..** ».

- 5) Retourner dans le répertoire « **tpSE1** ». Vérifiez que vous êtes bien dans le bon répertoire avec la commande « **pwd** ».
- 6) Taper la commande « **cd /home/user** », puis « **pwd** » pour vérifier que vous vous retrouvez au même endroit qu'à la question 1.
- 7) Comparez la commande « **cd ..** » avec la commande « **cd /home/user** ». On utilise deux méthodes différentes afin de donner le chemin à suivre pour se déplacer dans le système de fichiers.
 - a. Quel qualificatif utilise-t-on pour qualifier le chemin lorsque l'on utilise « **..** » ?
 - b. Quel qualificatif utilise-t-on pour qualifier le chemin lorsque l'on utilise « **/home/user** » ?

Exercice 4 :

- 1) Se mettre dans le répertoire « **tpSE1** » créé à l'exercice précédent en utilisant le chemin absolu. Vérifier le répertoire courant en tapant « **pwd** ».
- 2) Créer un répertoire « **documents** » dans « **tpSE1** », et vérifier avec la commande « **ls** » qu'il existe bien.
- 3) Entrer dans ce répertoire « **documents** ». Vérifier votre répertoire courant en tapant « **pwd** ».
- 4) Faire avec la commande « **tree** » les deux lignes suivantes : « **tree ../../.. -Fn** » et « **tree /home/user/ -Fn** »
- 5) Expliquer pourquoi, à part la première ligne, les deux résultats sont identiques.
- 6) Créer un répertoire « **boulot** » dans « **documents** », et vérifier avec la commande « **ls** » qu'il existe bien.
- 7) Entrer dans ce répertoire « **boulot** ». Vérifier votre répertoire courant en tapant « **pwd** ».

Exercice 5 :

- 1) Si ça n'est pas le cas, se placer dans votre répertoire « **documents** ».
- 2) Après avoir tapé : « **ls** », puis « **ls cp /home/user/Prof** », puis « **cp /home/user/Prof/cesarSNT.py .** » et enfin « **ls** ». expliquer ce qui a été fait avec la seconde commande.
- 3) Taper une des deux lignes suivantes :
 - **cp /home/user/Prof/cesarSNT.py /home/user/FakeStudent**
 - **cd /home/user/FakeStudent cp /home/user/Prof/cesarSNT.py .**
- 4) Expliquer pourquoi elles ont le même effet.
- 5) Nettoyez derrière vous ! Vous avez le droit de modifier votre répertoire perso mais pas celui nommé « **FakeStudent** ». Il était vide avant de commencer cet exercice, donc vous devez utiliser la commande « **rm** » (voir photocopié de cours) avec les bons paramètres pour faire place nette.
- 6) Avant de vous déconnecter taper la commande suivante :
« **history > /home/user/votrePrenom/tpSE1/SeanceHistorique.txt** ».

Cette dernière commande sera expliquée dans la seconde partie de ce chapitre.

Système d'exploitation (partie 2)

Exercice 0

1. Connectez-vous au serveur de la classe sur le site CoCalc.
2. Taper la commande « history -c », afin d'effacer votre historique du TP précédent.
3. créer un répertoire « tpSE2 » dans votre répertoire personnel. Vérifier que ça a fonctionné puis se placer dans le répertoire tpSE2.

Exercice 1 : afficher les droits des fichiers (ls -l)

1. Avec la commande « cp », copier dans votre répertoire tpSE2 le fichier dont le chemin absolu est le suivant : /home/user/Prof/solution2.py
2. Taper la commande « ls -l » puis expliquer chaque élément de la ligne qui apparaît.

Exercice 2 : redirection des flux de sortie (>)

1. a) dans votre répertoire tpSE2 taper la commande « echo Bonjour » et valider.
b) A l'aide du cours, expliquer à quoi sert cette commande
2. a) Taper la commande echo Bonjour > fiche.txt
b) Taper la commande « ls -l ». Pourquoi la valeur devant le mois est « 8 » ?
3. Taper la commande « cat fiche.txt » puis expliquez le résultat obtenu.

```
cat fiche.txt
echo Bonsoir > fiche.txt
cat fiche.txt
echo Good Night >> fiche.txt
cat fiche.txt
```

Exercice 3 : redirection des flux de sortie (> et >>)

Tapez les commandes de la copie d'écran ci-contre :
En quoi les caractères de redirection « > » et « >> » diffèrent ils ?

Exercice 4 : redirection des flux de sortie (> et >>)

Dans cet exercice, vous allez utiliser la commande « find » qui permet de « trouver » (find) des éléments (Répertoires et/ou fichiers) dans l'arborescence d'un système de fichiers.

1. Avec la commande « cd », revenir dans votre répertoire personnel.
 2. Taper la commande suivante : « find . -iname '*.txt' »
Explication : le caractère « * » remplace n'importe quelle chaîne de caractère y compris la chaîne de caractère vide. Ainsi, les fichiers « bonjour.txt », « b.txt » et « .txt » seront retenus par la commande « find ».
 3. Que signifie les deux paramètres « . » et « -iname » ?
- Remarques : Vous pouvez trouver une aide intéressante en tapant : « man find »
Pour chercher le mot clé « iname » dans le manuel, vous pouvez taper « /iname ».
Pour continuer la recherche dans le document tapez sur la lettre « n » (n=new).
Pour sortir du manuel, il suffit d'appuyer sur la lettre « q ».

Exercice 5 : redirection des flux de sortie (>)

1. a) placer vous dans /home/user/vosre_prenom puis taper la commande : « find . -iname '*.txt' > result.txt »
b) Pourquoi la commande « find » ne génère-t-elle aucune sortie sur l'écran ?
2. a) Taper la commande « cat result.txt »
b) Comment le fichier « result.txt » a été créé et pourquoi il contient ces données ?

Exercice 6 : permissions

1. Taper la commande « find /etc/ -iname '*.txt' ».
2. Expliquez, au niveau des paramètres de la commande « find », la différence avec ceux de l'exercice 4 ?
3. Pourquoi avez-vous toutes ces lignes « Permission non accordée » ?

Exercice 7 : redirection des flux de sortie (>)

1. Modifiez la ligne de commande « find » de l'exercice précédente comme ceci : « find /etc/ -iname '*.txt' > /home/user/vosre_prenom/tpSE2/findwhat.txt ».
2. Pourquoi les fichiers trouvés par « find » comme « limits.txt » n'apparaissent pas à l'écran alors que c'était le cas dans l'exercice précédent ?

3. Où se trouvent les informations sur les fichiers trouvés par la commande « find » ?
4. Prouvez que votre réponse à la question précédente est bonne en proposant une ligne de commande.

Exercice 8 : redirection des flux de sortie (>)

1. Après vous être placé dans votre répertoire « tpSE2 » taper la commande suivante : « find /etc/ -iname '*.txt' > findwhat.txt 2> erreurs.txt » Remarque : attention à bien respecter les espaces !
2. Pourquoi plus aucun résultat n'apparaît sur l'écran ? Où sont ces résultats ?

```
ls -l findwhat.txt
chmod u-w findwhat.txt
ls -l findwhat.txt
echo fin >> findwhat.txt
```

Exercice 9 : permission et droit des fichiers

1. Avec la commande cd, se mettre dans votre répertoire tpSE2
2. Taper les commandes ci-contre.
3. Expliquez pourquoi la commande « echo » échoue.
4. Que fait la commande « chmod » ?

Exercice 10 : permission et droit des fichiers

1. Taper les lignes suivantes :
2. Expliquez pourquoi la ligne « echo fin >> findwhat.txt » fonctionne alors qu'elle échouait dans l'exercice précédent.
3. La commande « cat findwhat.txt » est utilisée deux fois. Expliquez pourquoi elle échoue la première fois mais qu'à la deuxième elle fonctionne ?

```
chmod u+w,u-r findwhat.txt
ls -l findwhat.txt
echo fin >> findwhat.txt
cat findwhat.txt
chmod u+r findwhat.txt
ls -l findwhat.txt
cat findwhat.txt
```

Exercice 11 : permission et droit des fichiers

Un script est un fichier texte qui contient des commandes que l'interpréteur de commandes est capable d'exécuter. Sous Linux, l'extension d'un fichier qui est un script pour l'interpréteur de commandes est « .sh ». Rappel la famille des interpréteurs de commandes est appelée « Shell ».

1. Taper les commandes suivantes :

```
echo 'echo bonjour les NSI !' > monprog.sh
echo 'ls /home/user/Prof' >> monprog.sh
echo 'cat /home/user/vosre_prenom/tpSE2/findwhat.txt' >> monprog.sh
cat monprog.sh
```

Elles ont permis de créer un programme exécutable qui s'appelle monprog.sh

2. Taper ensuite la commande « ./monprog.sh ». Quel message apparaît ? Pourquoi ?
4. Proposer une solution pour éviter l'erreur. La mettre en place puis lancer de nouveau le fichier exécutable.

Exercice 12 : vu le fonctionnement de CoCalc on ne peut le faire avec le serveur

Mme Szpieg a créé une activité pour ses élèves, elle leur demande d'explorer un répertoire. On obtient :

```
lfclp@SshSzpiegServer:~/tpSE2 $ ls -l /home/lfclppartage/tpSE2/
total 16
-rw-r--r-- 1 marion prof 9 mars 29 10:37 fiche.txt
-rwxr-xr-x 1 marion prof 52 mars 29 10:51 marion2prog.sh
-rwxr-xr-- 1 marion lfclp 56 mars 29 10:53 marion3prog.sh
-rwxr-xr-- 1 marion prof 24 mars 29 10:41 marionprog.sh
lfclp@SshSzpiegServer:~/tpSE2 $ id
uid=1027(lfclp) gid=1012(lfclp) groupes=1012(lfclp)
```

1. A quoi sert la commande « id » ?
2. Les élèves doivent tenter d'exécuter les trois scripts. A votre avis pourquoi le noyau empêche-t-il le script « marionprog.sh » de fonctionner mais autorise les deux autres à fonctionner ?

Exercice 13 : fin de TP

1. A l'aide de la commande « mv », déplacer le fichiers result.txt, qui est dans le répertoire /home/user/vosre_prenom, dans le répertoire /home/user/vosre_prenom/tpSE2/
2. Avant de vous déconnecter taper la commande suivante : « history > /home/user/vosre_prenom/tpSE2/SE2hist.txt » Pouvez-vous maintenant expliquer ce qu'elle fait ?