

Compte-rendu TP1 - Utilisation du Shell

L3 - Systèmes et Réseaux - September 25, 2021

Exercice 1: Le shell

Ouvrir un interpréteur de lignes de commandes (shell).

1. À quoi correspond le préambule qui précède chacune de vos commandes.

Réponse :

```
ep298479@zola:/home4/ep298479/
```

Le préambule indique le nom de l'utilisateur, le serveur et le répertoire courant

2. Tester les commandes `hostname` et `logname`.

Réponse :

```
$ hostname : zola $ logname  
: Aucun identifiant
```

Ces commandes renvoient respectivement le nom du serveur et le nom d'utilisateur sur lequel l'utilisateur s'est connecté. On suppose que cette dernière ne renvoie rien car nous travaillons sous terminal léger.

3. Tester la commande `whoami`.

Réponse :

```
$ whoami : ep298479
```

`whoami` retourne le nom de l'utilisateur courant

4. Regarder qui est connecté sur la station par la commande `who`.
5. Utiliser la commande `echo $TERM` pour déterminer le type de terminal utilisé.

Réponse :

```
$ echo $TERM : xterm
```

Il s'agit de `xterm`

6. Utiliser la commande `uname -a` pour déterminer votre système d'exploitation.

Réponse :

```
$ uname -a : Linux zola  
4.9.0-8-amd64 1 SMP Debian  
4.9.144-3.1...
```

Il s'agit donc de **SMP Debian 4.9.144-3.1**

7. Tester la fonction `clear`. À quoi sert-elle?

Réponse : Cette instruction permet d'effacer la console

Exercice 2: Manuel et aide

Vous allez voir comment obtenir des informations sur une commande.

1. Tester les différentes commandes d'aide : `man`, `whatis`, `apropos` et `info`.

Réponse :

man [commande] : Affiche dans le terminal la page du manuel (= man) correspondant à la commande recherchée.

whatis [commande]: Affiche dans le terminal une description succincte de la commande recherchée.

apropos [mot-clé]: Permet de faire une recherche dans le manuel en fonction de mots-clés

info [commande]: Même intérêt que man, affiche le résultat au format Info (qui permet de visualiser de la documentation)

2. Que donne `>man man`.

Réponse : Ouvre la page du manuel au sujet de la commande **man**

3. À quoi sert la fonction `ls`? Tester différentes options possibles.

Réponse : `ls` : Affiche à la suite tous les fichiers du répertoire courant

OPTIONS

-l : Affiche les noms des fichiers avec plusieurs informations utiles (notamment droit en lecture, écriture et d'exécution de l'utilisateur, du groupe, et de l'entière des utilisateurs, taille, auteur...)

-i : Affiche le numéro d'index de chaque fichier (id des i-noeuds)

-t : Trie en fonction de la date de modification

Exercice 3: Shell : raccourcis clavier

Donner la fonctionnalité des raccourcis du shell proposés ci-dessous (avec un exemple) :

Réponse :

Ctrl+a : Déplace le curseur au début de la ligne (debut)

Ctrl+e : Déplace le curseur à la fin de la ligne (fin)

Ctrl+d : Supprime le caractère situé après le curseur (suppr.)

Ctrl+b : Recule le curseur d'un caractère (back)

Ctrl+f : Avance le curseur d'un caractère (forward)

Ctrl+l : Saute assez de ligne pour donner l'illusion que la console a été effacée comme **clear** le ferait

Ctrl+u : Le texte situé avant le curseur

Ctrl+w : Efface le texte situé entre le curseur et le précédent espace (donc le dernier mot)

Ctrl+r : Permet d'effectuer une recherche dans l'historique des commandes

Exercice 4: Organisation du travail

Vous allez maintenant créer dans votre HOME une arborescence qui vous permettra d'organiser votre travail. Tous doit bien évidemment être réalisé à partir du shell avec des commandes de base.

1. Créer un dossier `SystemesEtReseaux`. Entrer dans ce dossier.

Réponse :

```
>mkdir SystemesEtReseaux
>cd SystemesEtReseaux
```

2. Créer un dossier `TP`. Entrer dans ce dossier.

Réponse :

```
>mkdir TP && cd TP
```

3. Créer des dossiers `TP1` jusqu'à `TP3`.

4. Télécharger les relatifs aux 3 `TPs`.

5. Déplacer ces fichiers dans le dossier correspondant (`TP1`, `TP2` ou `TP3`).

Réponse :

```
>mv /home1/ep298479/
Téléchargements/TP1.pdf
TP1/TP1.pdf
```

On fait de même pour les autres fichiers du dossier Téléchargement...

6. Créer un fichier texte `monNom` contenant votre nom.

Réponse :

```
>echo "Evan Petit" > monNom
```

7. Créer un lien symbolique pointant vers le fichier `monNom`.

Réponse : `>ln -s monNom monNom2`

8. Afficher la taille de `monNom` et du lien symbolique puis comparer les deux valeurs.

Réponse : `>stat -c %g monNom && stat -c %g monNom2`
11
6

Le lien symbolique est bien plus petit en taille

9. Utiliser la commande `cat` pour afficher le texte pointé par le lien symbolique.

Réponse : `>cat monNom2`

10. Rajouter au fichier `monNom` le nom du cours. Puis ré-exécuter la commande précédente. Que se passe-t-il?

Réponse : `echo "Systèmes et réseaux - TP1" > monNom && cat monNom2`

Le lien symbolique redirige correctement vers le fichier actualisé.

11. Utilisez la commande `ls -al` sur les fichiers suivants.

- le fichier `commandes.txt`
- le lien symbolique
- le dossier `SystemesEtReseaux`
- le contenu du dossier `/dev`

À quoi correspond le premier caractère de chaque ligne ?

Réponse : Il s'agit du type (d pour directory, - pour file, l pour lien symbolique...)

Exercice 5: Affichage

La commande `echo` permet d'afficher la valeur des variables sur le shell.

1. Tester la commande `>echo "Bonjour à tous!"`
2. Initialiser une variable locale `var` à 5 (`>var = 5`)
3. Tester les commandes `>echo var` et `>echo $var`

Réponse : `> var=5`
`> echo var && >echo $var`
var
5

4. Sur un autre shell tester de nouveau la commande `>echo $var`. Que se passe-t'il?

Réponse : Le shell ne renvoie rien. On en déduit que les variables locales sont propres à un shell

5. Tester plusieurs variables d'environnement (globales). Déterminer votre numéro d'utilisateur (uid) sur le système ainsi que le nom et numéro du groupe auquel vous appartenez (gid).

Réponse : ...

`>...`

6. Initialiser plusieurs variables par la commande `read` puis afficher les.

Réponse : ...

Exercice 6: Droits

1. Créer un répertoire `rep` contenant un fichier `fich` contenant la date du jour.

2. À partir du fichier père de `rep` :

- (a) Appliquer la commande `chmod 600 rep`. Puis tester la commande `cd rep`.

Réponse : Résultats : `>Permission non accordée`
Commentaires : La permission d'exécution n'est donnée à personne. Ainsi il est impossible de se rendre dans le dossier `rep`

- (b) Appliquer la commande `chmod 500 rep`. Puis tester les commandes `cd rep`, `ls rep` et `echo 'coucou'>rep/fich2`.

Réponse :
Résultats : Toutes les commandes fonctionnent hormis la dernière dont voici l'erreur `>Permission non accordée`
Commentaires : 500 donne uniquement au propriétaire les permissions de lire et d'exécuter. Il est donc normal que l'on puisse pas créer (écrire) un fichier

- (c) Appliquer la commande `chmod 300 rep`. Puis tester les commandes `cd rep`, `ls rep` et `echo 'coucou'>rep/fich2`.

Réponse :
Résultats : Toutes les commandes fonctionnent hormis `ls rep` avec toujours la même erreur `>Permission non accordée`
Commentaires : 300 donne uniquement au propriétaire les permissions d'écrire et d'exécuter. Il est donc normal que l'on puisse pas afficher (lire) les fichiers

3. À partir du fichier `rep` :

- (a) Appliquer la commande `chmod 400 fich`. Puis tester les commandes `cat fich` et `echo 'coucou'>fich`.

Réponse : Résultats :
Commentaires :

- (b) Appliquer la commande `chmod 200 fich`. Puis tester les commandes `cat fich` et `echo 'coucou'>fich`.

Réponse : Résultats :
Commentaires :