

MN 2.1 Initiation Unix – TP1 : Shell zsh

Olivier Dalle (Olivier.Dalle@unice.fr)

1 Login et Consoles

1.1 Expérimentations avec les consoles

- Si vous êtes déjà connectés, quittez votre session, et basculez sur une console texte (`ctrl-alt-F1` à `ctrl-alt-F5` et saisissez vos login/password).
- Remarquez que vous pouvez basculer sur une console texte et revenir à tout moment en utilisant les combinaisons `alt-FX` (la touche `ctrl` n'est plus nécessaire quand vous êtes déjà en mode texte).
- Revenez à la console graphique avec `alt-F7` et connectez-vous à nouveau, dans l'environnement graphique cette fois.
- Remarquez que vous pouvez basculer vers les consoles texte à tout moment, vous retrouverez toujours les choses que vous avez laissées. **Avant de partir, n'oubliez pas de vous déconnecter de toutes ces consoles !!**

Pour les questions suivantes, vous pourrez tester indifféremment les commandes depuis l'environnement texte ou graphique.

1.2 Découverte de l'organisation disque

- Affichez l'occupation des différentes partitions montées sur votre machine avec la commande `df`. Remarquez que l'affichage est plus lisible avec l'option `-h` . . .
- Dans quel répertoire vous trouvez-vous par défaut (immédiatement après l'ouverture d'un nouveau terminal). Donnez son chemin absolu.
- Listez son contenu :
 - testez les différentes options présentées dans le cours
 - recherchez dans le manuel une nouvelle option de `ls` qui permet d'afficher le résultat dans un format "humainement compréhensible".
 - expérimentez de nouvelles options au fur et à mesure que vous les découvrez dans la page de manuel. Et n'oubliez pas que vous pouvez les combiner entre elles !!
- calculez l'espace occupé par tous les fichiers présents dans votre HD (commande `du`).

Pensez à utiliser ces commandes régulièrement pour vérifier l'occupation de votre HD et de son contenu sur le disque : votre "espace vital" est limité!

2 Le B-A-BA

2.1 Création de fichiers et répertoires

- Créez un fichier vide de nom "toto" (commande `touch`) et vérifiez le résultat avec la commande `ls`
- Détruire le fichier "toto" (et vérifier).
- Créer un répertoire "titi" et un fichier "toto" (à nouveau).
- Déplacez le fichier "toto" dans le répertoire "titi".
- Déplacez le fichier "titi/toto" dans le répertoire courant en le renommant "tata" (tout ça en une seule commande!)
- Listez le contenu de la racine
- Listez le contenu détaillé de votre HD en le désignant par son chemin absolu.

2.2 Le manuel

- Testez la commande `man` sur quelques commandes que nous avons vues.
- Remarquez que leur structure est toujours identique : NAME, SYNOPSIS, OPTIONS, ..., SEE ALSO, BUGS, AUTHOR.
- Testez la navigation dans une page de manuelle (touches 'espace', 'b', 'entrée', 'q', '?')
- Testez la fonctionnalité de recherche d'un mot dans la page (touche '/')
- Vous pouvez aussi afficher les pages manuel de quelques fonctions du langage C (essayez de trouver celles de "printf", "open", "lseek". Attention au numéro de section, certaines pages existent en double dans plusieurs sections!)
- En parcourant rapidement la page de manuel de la fonction C `signal(2)` (appel système, essayer de trouver la liste des signaux Unix.
- Affichez la page d'info de `zsh`. Testez la navigation hypertexte. Vous pourrez comparer avec la(les) page(s) de manuel, c'est quand même plus pratique ...

2.3 Fichiers Bizarres

Essayez de créer (puis détruire) les fichier suivants :

- ''''';;;;
- --help

3 Redirections

3.1 Initiation à la plomberie

- A l'aide de la commande `cat`, créez un fichier "toto" contenant deux lignes de texte saisies au clavier, **dont l'une comporte au moins quelques 'a'**.
- Toujours avec `cat`, ajoutez une nouvelle ligne de texte au fichier "toto" (sans écraser son contenu précédent).
- Encore avec `cat`, créez un fichier "tata" contenant 2 fois de suite le contenu du fichier "toto".
- Avec la commande `tr`, créez maintenant un fichier "tAtA" à partir de "tata" dans lequel les 'a' sont remplacés par des 'A'.
- En consultant la page de manuel de `tr`, trouvez un moyen simple de remplacer les majuscules par des minuscules.
- A l'aide de la commande `wc`, comptez le nombre de fichiers présents dans le répertoire `/bin`

3.2 Un gros chantier de plomberie

L'idée ici est de compliquer un peu ce que l'on vient de voir de façon à bien comprendre la philosophie des tubes, et en particulier l'idée que l'on peut les enchaîner les uns à la suite des autres, en récupérant éventuellement le résultat de traitements intermédiaires (grâce à la commande `tee`). Dans la question qui suit, vous n'aurez donc besoin que des commandes `ls`, `tr` et `tee` (sur laquelle vous pouvez commencer par vous documenter). Procédez par étapes successives, l'enchaînement final est aussi long... qu'inutile!

En une seule ligne de commande, sauvez le résultat de la commande `ls -al` appliquée au répertoire `/usr` :

- en minuscules dans le fichier "lsal"
- en majuscules dans le fichier "LSAL"
- en remplaçant les espaces par des tirets avant de l'afficher sur la sortie standard

4 Permissions

4.1 Modifier les permissions d'un fichier existant

- Créez un fichier vide de nom "fic" et un répertoire de nom "dir".
- Quelles sont leurs permissions? Pourquoi ont-ils été créés avec ces permissions?
- Ajoutez au fichier "fic" la permission d'exécution pour le groupe. Vérifiez le résultat.

- Retirez au fichier "fic" toutes les permissions pour le groupe (mais pas les autres). Et vérifiez.
- Créez un fichier "fic" **dans le répertoire "dir"**.
- Retirez au répertoire la permission d'exécution pour le propriétaire. Arrivez-vous à vous déplacer dans ce répertoire avec la commande `cd`? Redonnez-lui la permission et ré-essayez.
- Ajoutez au fichier "dir/fic2" la permission d'écriture pour tous. Et retirez la permission d'écriture sur le répertoire "dir" pour son propriétaire. Que n'avez plus le droit de faire? Essayez.

4.2 Bas les masques !

- Quel est le résultat de la commande `umask`? Que signifie-t-il?
- Faites en sorte que par défaut, les fichiers dont vous êtes propriétaires soient créés sans la permission d'écriture pour le propriétaire.
- Créez un fichier vide et vérifiez que cela fonctionne.
- Depuis un autre terminal, créez un nouveau fichier vide. Qu'observez-vous? Quelle conclusion en tirer sur l'utilisation de la commande `umask`?

5 La commande `find`

Rappelez-vous qu'à tout moment, vous pouvez interrompre l'exécution d'une commande (et pas seulement `find`) à l'aide de la combinaison `ctrl-c`.

- Listez tous les fichiers du répertoire `/usr` et de ses sous-répertoires.
- Listez tous les fichiers du répertoire précédent qui font plus de 300 kilo-octets
- Listez tous les fichiers du répertoire précédent dont le nom commence par 'e'
- Listez tous les fichiers du répertoire précédent dont le nom commence par 'e' et dont la taille est supérieure à 300 kilo-octets
- Listez tous les fichiers du répertoire précédent dont le nom commence par 'e' et dont la taille est supérieure à 100 octets
- Listez tous les fichiers du répertoire précédent dont le nom commence par 'e', dont la taille est supérieure à 100 octets et qui sont de type répertoire

6 Processus et signaux

6.1 Observation de la faune

- Lister les processus avec la commande `ps`
- Lister *tous* les processus qui vous appartiennent
- Lister absolument tous les processus en cours d'exécution
- Lister tous les processus avec le détail des commandes et arguments utilisés pour créer ces processus
- Lister tous les processus de façon à visualiser leur liens de parenté sous forme arborescente
- Que montre la commande `top`?

6.2 Avant et arrière plan

Pour cet exercice placez-vous dans l'environnement graphique.

- Lancez la commande `xeyes`
- Arrivez-vous toujours à lancer des commandes depuis le terminal que vous utilisiez pour lancer `xeyes`?
- Tapez la combinaison `ctrl-z` dans le terminal? Quelles sont les conséquences :
 - Sur le terminal?
 - Sur les yeux?
- Lancez la commande `bg` dans le terminal, et étudiez ses conséquences. Testez la combinaison `ctrl-c`.
- Lancez maintenant la commande `fg`. Que provoque-t-elle? Que se passe-t-il si vous utilisez la combinaison `ctrl-c`?
- Lancez la commande `xeyes&`. Quel est le rôle du caractère `&`?

6.3 La commande kill

- Lancez la commande `xeyes&` (si ce n'est déjà fait).
- Retrouvez son numéro de processus.
- Utilisez la commande `kill` pour tuer ce processus
- Recommencez, en utilisant cette fois le numéro de job pour tuer le processus au lieu de son numéro de processus.
- Si ce n'est pas encore fait, connectez-vous sur une console texte puis revenez à l'environnement graphique.
- Testez la commande `kill -9 -1`. Que provoque-t-elle? A-t-elle eu un effet sur la console texte?

Dans tous les cas, rappelez-vous qu'il vaut mieux essayer de quitter proprement une session et n'utiliser la "mort violente" qu'en tout dernier recours ...

7 Premier contact avec les variables

- En utilisant la commande `echo`, affichez le contenu de la variable `HOME`?
- Et que contient la variable `PATH`?
- Listez toutes vos variables d'environnement.
- Créez une nouvelle de nom "toto" contenant la chaîne "bonjour".