

TP 2 de familiarisation avec Unix

Terminal – Éditeur (x)emacs – SGF

Vincent Berry - vberry@lirmm.fr

Indications

Les temps indiqués à la fin des titres de paragraphes ne sont qu'indicatifs et approximatifs : votre expérience passée vous amènera à dépasser ou ne pas utiliser complètement le temps indiqué : ce n'est pas grave. Ceux qui ont plus d'expérience que les autres, sont invités à explorer à fond chaque aspect évoqué dans le TP, au besoin en effectuant des expérimentations complémentaires.

Rappel : les questions annotées par un symbole E peuvent être laissées de côté pour être regardées à la fin de la séance si le temps le permet, sinon elles devront être examinées en dehors des séances de TP.

Table des matières

1	Gestion des répertoires	1
1.1	C'est parti pour le grand plongeon!	1
1.2	Création d'une première arborescence	1
1.3	De l'image à la réalité	2
1.4	Une histoire de (pré)noms	2
2	Apprivoisons l'éditeur de texte nano (15 mn)	3
3	Les commandes minimum sur les processus (10 mn)	3
3.1	Processus et commandes	3
3.2	Interrompt un processus lancé par le terminal	4
3.3	Gestion des processus E	4
4	Le terminal de commandes (25 mn) – vérifications	4
4.1	Les parties de l'arborescence du SGF situées en dehors de votre espace personnel (10mn)	4
4.2	Expérimentation dans votre espace de travail (15mn)	5

1 Gestion des répertoires

Nous allons réaliser un peu de gymnastique avec les commandes de base. Certaines questions vous paraîtront répétitives, mais à travers la pratique intensive vous allez maîtriser certaines commandes très importantes sous Unix.

1.1 C'est parti pour le grand plongeon!

Ouvrez un terminal permettant de taper des commandes Unix. Affichez le nom du répertoire courant avec la commande `pwd` (*print working directory*).

Changez votre répertoire courant pour remonter dans le répertoire parent avec la commande `cd` (*change directory*) suivie du nom du répertoire. Indiquez trois moyens par lesquels vous pouvez vérifier que la commande précédente a fonctionné.

Changez votre répertoire courant pour revenir à votre répertoire initial à l'aide de la commande `cd` sans argument. Comment vérifier que vous êtes bien revenu à votre point de départ? Essayez.

1.2 Création d'une première arborescence

Placez vous dans votre répertoire d'accueil (utilisez la tilde – sur Mac elle s'obtient par la combinaison `alt + N`). Soit la succession suivante de commandes :

```
cp /etc/hosts a
mkdir b c
cd b
cp ../a d
```

```

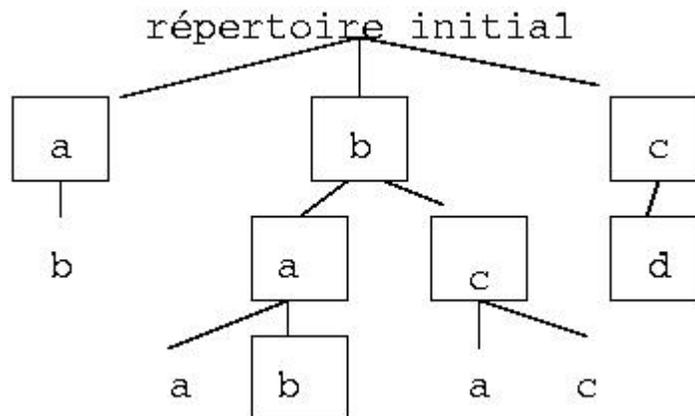
mkdir ../e f
cd
cp a b/f/g
cd b/f
cp g ../../e
cd ..
rm ../a
rmdir ../c
mv ../e/g ../e/x

```

Dessinez l'arborescence résultante. Quel est le répertoire courant à la fin de l'opération ?
 Pouvez-vous expliquer pourquoi la commande `cd ~/a/f` ne fonctionne pas ?

1.3 De l'image à la réalité

Quelle séquence de commandes vous permet de créer l'arborescence de la figure ci-dessous ?



(note : la commande `touch nom-fichier` permet de créer un fichier du nom indiqué, de taille nulle mais existant quand même).

Comment pouvez-vous expliquer que le système d'exploitation ne se trompe pas de fichier quand on lui désigne un fichier du nom `a` (il y en a deux dans l'arborescence que vous venez de créer).

1.4 Une histoire de (pré)noms

Créez maintenant deux sous répertoires, l'un ayant comme nom votre prénom, et l'autre comme nom votre nom de famille (par exemple Charles et DeGaulle – attention, ne mettez pas d'espace dans les noms de fichiers et répertoire, vous vous rappelez pourquoi ?). Affichez ensuite le contenu de votre répertoire d'accueil de trois manières différentes :

- Par ordre alphabétique
- Par date de création des fichiers
- Une autre option de votre choix (expliquer)

En partant de votre répertoire d'accueil, changez de répertoire pour vous positionner dans le répertoire (prénom), et créez trois fichiers comme indiqué ci-dessous

```

touch fichier1 > fichier2
ls -la > fichier3
cat fichier3 > fichier4
cat > fichier5 (tapez à ce moment du texte ensuite vous terminez par ctrl + d).

```

Copiez au choix deux de ces fichiers dans le répertoire (nom de famille) à partir du répertoire courant (prénom) puis effacez une des copies (se trouvant dans le répertoire nom de famille), toujours à partir du répertoire courant (prénom), c'est-à-dire que vous agissez à distance (c'est possible en indiquant le chemin des fichiers concernés dans les commandes que vous tapez).

Changez le nom d'un des fichiers se trouvant dans le répertoire courant (commande `mv`).

Déplacez un de ces fichiers vers le répertoire nom de famille.

En restant dans le répertoire courant (prénom), copiez-y un des fichiers se trouvant dans le répertoire nom de famille.

Allez maintenant dans le répertoire nom de famille et affichez son contenu.

Exécutez la commande `cd` sans arguments et expliquez ce qui se passe.

2 Apprivoisons l'éditeur de texte nano (15 mn)

Il s'agit d'un éditeur minimaliste fonctionnant à l'intérieur d'un terminal. Cet avantage permet de l'utiliser à distance, quand on est connecté sur une machine distante sans avoir besoin de voir son écran (trop fort!). Avant tout, trouvez vos notes de cours sur l'éditeur nano et serrez les très fort, on va en avoir besoin (raccourcis, ...).

Dans un terminal de commandes lancez l'éditeur nano en tapant simplement son nom. Pouvez-vous taper de nouvelles commandes dès lors? S'exécutent-elles? Pourquoi (expliquez ce qui se passe)?

Quelle que soit vos manipulations précédentes, assurez-vous que vous êtes toujours dans nano (sinon, revenez-y), tapez une ligne de texte puis sauvegarder tout ça dans un fichier nommé `essai-nano.txt`.

Quittez maintenant l'éditeur et par la commande `ls -l` assurez-vous que vous avez bien créé le bon fichier et qu'il n'est pas vide.

Pouvez-vous relancer nano en lui demandant de charger le fichier `essai-nano.txt`?

Couper - Coller - Copier

Dans le navigateur web ouvrez la page <http://www.nano-editor.org> prenez dans le presse-papier le contenu de la page Documentation. Revenez dans le terminal (sortez de nano si vous y êtes), et tapez la commande suivante :

`cat >> essai-nano.txt` puis videz le contenu du presse-papier, et tapez `CTRL + C`.

Normalement votre fichier `essai-nano.txt` a grossi de quelques lignes. Pouvez-vous indiquer deux façons de le vérifier?

Lancez à nouveau nano pour éditer ce fichier. Quels raccourcis vous permettent de localiser les occurrences du mot "new" dans le fichier? Essayez.	
--	--

Couper-Coller

Sélectionnez maintenant la 1ère phrase du 2ème paragraphe de votre texte puis déplacez le curseur avec les flèches du clavier) et coupez cette phrase. Quel est le raccourci pour faire cette dernière action? Placez ensuite la phrase que vous avez coupée au début du 3ème paragraphe.	
---	--

Ouvrez une autre fenêtre de type terminal (dans le fenêtre en cours restez dans nano). Dans ce nouveau terminal, utilisez la commande `verb!rm!` pour supprimer le fichier `essai-nano.txt`. Utilisez la commande `verb!ls!` pour vérifier que le fichier n'existe plus sur disque.

Question : revenez dans le terminal où nano s'exécute, vous devez constater que le contenu du fichier <code>essai-nano.txt</code> existe toujours, mais comment est-ce possible puisque vous venez de le supprimer? Est-ce vraiment le cas? Pouvez-vous expliquer la situation au moment où vous venez de détruire le fichier par la commande <code>verb!rm!</code> ?	
---	--

Revenez maintenant dans l'éditeur de textes et sauvegardez votre texte à nouveau. Dans l'autre terminal inspectez le contenu du répertoire et regardez ce qui c'est produit.

3 Les commandes minimum sur les processus (10 mn)

3.1 Processus et commandes

À l'exécution de tout programme, le Système d'exploitation crée un **processus**, sorte d'entité qui enregistre les informations sur l'exécution de ce processus (sous quelle identité s'exécute-t-il, à quelle instruction en est-il, etc).

Quand vous lancez une commande dans un terminal, vous provoquez la création d'un processus. Vérifiez en ouvrant une nouvelle fenêtre Terminal, ou un nouvel onglet dans la fenêtre Terminal actuelle. Demandez la liste des processus (commande `ps`) avec l'option permettant de voir les filiations entre processus). Indiquez les bons paramètres pour voir les processus que vous avez lancés non seulement dans le shell en cours (par défaut) mais aussi ceux lancés dans d'autres shells (option `-u`).

Essayez la commande `ps tree <votrelogin>` qui vous permet de voir explicitement la filiation entre vos processus.

3.2 Interrompre un processus lancé par le terminal

Certaines commandes tapées dans un Terminal ne se terminent pas tout de suite, ceci peut être embêtant. Dans un terminal, lancez la commande `xemacs`. Revenez dans le Terminal et essayez maintenant la commande `ls` sur le répertoire en cours.

Normalement, tapez cette dernière commande ne provoque rien dans votre terminal. La raison en est que la commande précédente (`xemacs`) n'a pas encore terminé.

En tapant `CTRL - D` dans le terminal, vous récupérez la possibilité de taper de nouvelles commandes : le système a forcé le processus qui était en cours d'exécution à se terminer (brutalement). Du coup, si vous aviez commencé à taper des modifications dans un document important, vous perdez ces modifications qui ne seront pas répercutées dans le fichier sur disque.

`CTRL - Z` permet d'interrompre un processus de façon temporaire. Faites un essai en relançant d'abord `emacs`.

Pour qu'il continue de s'exécuter en arrière plan, vous devez taper `bg` (pour *background*, c-a-d *arrière-plan*). Dans ce cas, vous gardez la main dans le terminal (enfin, c'est une façon imagée de dire que vous pouvez encore y taper des commandes).

Vous devriez aussi aimer la possibilité que le terminal vous laisse de lancer d'emblée un processus en arrière-plan : pour cela il suffit d'ajouter le caractère `&` à la fin de la commande saisie dans le terminal. Faites un essai de lancement de `xemacs` de cette façon et vérifiez que vous conservez la possibilité d'exécuter des commandes dans le terminal.

3.3 Gestion des processus `e`

La commande `top` vous donne un aperçu des processus en cours d'exécution (les plus gourmands en ressources sont listés en premier). Accessoirement, la commande vous fournit aussi l'état de la mémoire centrale.

La commande `ps` vous permet de connaître les processus lancés depuis votre terminal et encore en cours d'exécution. Les options `-x` et `-a` vous permettent de voir plus de processus. L'option `--forest` est aussi très intéressante. Faites des essais et reportez-vous au `man` pour bien maîtriser ces options.

Enfin, vous pouvez expérimenter la commande `kill` au nom très explicite. Bien-sûr vous ne pouvez tuer que des processus tournant sous votre identité.

4 Le terminal de commandes (25 mn) – vérifications

Nous revenons faire des essais sur la manipulation de fichiers et répertoires. Vu votre degré de maîtrise, cette section doit être une promenade de santé (si ce n'est pas le cas, relisez bien les exercices de la section 1 et vos réponses, cela pourrait vous aider).

Utilisez les commandes appropriées pour répondre aux questions suivantes. Les commandes que vous n'avez pas encore vues en cours sont données entre parenthèses. N'oubliez pas que pour avoir la documentation d'une commande vous pouvez utiliser la commande `man nomCommande` dans un Terminal.

Nous allons expérimenter maintenant les commandes qui permettent de manipuler les répertoires et les fichiers. Ces commandes vous seront utiles non seulement en tant qu'utilisateur de système Unix/Linux, mais aussi en tant qu'informaticien, quand vous créerez des scripts ou des programmes qui feront pour vous (ou vos clients) un certain nombre de travaux automatiquement. Dans cette partie vous n'avez pas le droit d'utiliser l'explorateur graphique de fichiers (`Nautilus` sous Linux ou le `Finder` sous OSX). Fermez donc toutes les fenêtres de cette application qui sont actuellement ouvertes dans votre session ¹.

4.1 Les parties de l'arborescence du SGF situées en dehors de votre espace personnel (10mn)

Lancez un terminal de commande et utilisez les commandes de base pour répondre aux questions suivantes. Notez les réponses dans la colonne de droite

Initialement dans quel répertoire êtes-vous ? Comment le savoir plus précisément que ce qu'indique l'invite de commande) ?	
Déplacez-vous dans le répertoire <code>/tmp</code>	
Comment vérifier que vous êtes bien arrivé dans le bon répertoire ?	
Fermez maintenant le terminal et ouvrez-le à nouveau. Dans quel répertoire vous situez-vous ? qu'en concluez-vous ?	

1. A l'extrême limite, s'il vous arrive d'être complètement perdu dans la suite, vous pourrez rouvrir une de ces fenêtres, mais uniquement pour connaître la forme de l'arborescence, pas pour agir dessus. J'insiste sur le fait que même ceci peut être réalisé par des commandes.

Comment savoir si vous avez les droits pour écrire dans le répertoire <code>/tmp</code> ? Par exemple, quel argument faut-il ajouter à la commande <code>ls -l</code> pour obtenir cette information? Essayez de deux façons possibles : en désignant l'endroit intéressant de façon relative, puis en le désignant de façon absolue	
Dans la liste des répertoires et fichiers présents à cet endroit, comment pouvez-vous distinguer les répertoires des fichiers?	
Revenez dans votre répertoire d'accueil en utilisant le raccourci <code>~</code> (oui, mais pas tout seul : indiquer d'abord le nom de la commande permettant de changer de dossier).	
Essayez de trouver le répertoire d'accueil de votre enseignant.	
Pouvez-vous vous déplacer dans le répertoire d'accueil de votre enseignant? Pourquoi?	
Pouvez-vous créer un fichier dans son répertoire d'accueil (par exemple par la commande <code>touch coucou.txt</code> une fois dans son répertoire)? Pourquoi? Comment prouver ce que vous dites?	

4.2 Expérimentation dans votre espace de travail (15mn)

Revenez dans l'application Terminal, où nous allons faire quelques essais de commandes.

Note : dans le Terminal, il est possible de naviguer dans les commandes que vous avez tapées précédemment : on y accède par l'intermédiaire des touches flèches du clavier (haut, bas, gauche, droite). Ceci permet souvent de gagner du temps dans la saisie des commandes futures.

Replacez-vous dans votre répertoire d'accueil	
En utilisant le manuel ci-besoin (commande <code>man</code>), trouvez l'option de la commande <code>ls</code> qui vous permet de connaître tous les fichiers qui se trouvent dans ce répertoire, y compris les fichiers cachés (préfixés par un "point" dans les systèmes Unix)?	
Pouvez-vous deviner les usages de certains fichiers cachés que vous observez?	
Créez un répertoire nommé <code>UNIX</code> (commande <code>mkdir</code>) dans votre répertoire d'accueil	
Déplacez-vous dans ce répertoire (<code>cd</code>)	
Créez un sous-répertoire <code>TP2</code> dans <code>Unix</code> et déplacez-vous y	
Créez un fichier <code>titi</code> en utilisant <code>xemacs</code> . Vous pouvez par exemple y saisir : <i>Unix, j'ai même pas peur de toi, et d'abord je connais déjà plein de commandes</i>	
Quittez l'éditeur, revenez dans le Terminal et demandez à visualiser le contenu de ce fichier (commandes <code>cat</code> et <code>more</code>)	
Faites une copie de <code>titi</code> sous le nom <code>toto</code> (commande <code>cp</code>)	
Zut, il devait en fait s'appeler <code>tutu</code> . Renommez la copie du fichier en conséquence (commande <code>mv</code>)	

Double-zut, en fait, on voulait pas mettre le fichier dans le répertoire TP2, mais plutôt à la racine de votre arborescence personnelle. Déplacez le fichier (commande <code>mv</code>)	
En utilisant le mode absolu, changez les droits de <code>titi</code> de sorte que son propriétaire et le groupe aient les droits d'écriture et d'exécution, et les autres seulement le droit d'exécution (<code>chmod</code>)	
Quelle option et quel argument de la commande <code>ls</code> vous permettent de vérifier l'attribution des droits sur ce fichier ?	
Ajoutez le droit de lecture pour les autres en utilisant le mode symbolique (c-a-d en utilisant un/des symbole/s <code>r,w,x</code>)	
Essayez de visualiser <code>titi</code> (commande <code>cat</code>). Que faut-il faire pour y parvenir ?	
Remontez à votre répertoire d'accueil	
Tapez <code>cd U</code> puis appuyez sur la touche <code>TAB</code> afin d'expérimenter la complétion automatique. Validez	
Depuis le répertoire <code>Unix</code> créez un répertoire <code>Ursule</code> dans votre répertoire d'accueil (c-a-d que <code>Unix</code> et <code>Ursule</code> doivent être répertoires frères l'un de l'autre) en utilisant un <i>chemin relatif</i>	
Revenez dans votre répertoire d'accueil (commande <code>cd</code>) et tapez à nouveau <code>ls U</code> suivi de la touche magique <code>TAB</code> . Que se passe-t-il ? Pourquoi ?	
Essayez à nouveau en tapant maintenant deux fois sur la touche magique	
Avant de passer à la suite, remettez tous les fichiers texte manipulés ci-dessus dans le répertoire <code>TP2</code>	