

Initiation à l'automatisation des tâches - LI218
Interrogation écrite de TD du 02 octobre 2013 - Groupe 4
Aucun document autorisé - 20 minutes

Olga Kupriianova, Thibaut Verron

Question 1 – 1 point

Citez deux commandes permettant d'accéder à la documentation relative à une commande UNIX.

Réponse :

`man, info, help.`

`help` ne fonctionne qu'avec les primitives `bash`. Beaucoup d'entre vous ont également cité le paramètre `--help`, mais attention, celui-ci n'est pas supporté par toutes les commandes.

Question 2 – 2 points

Quelle commande permet de visualiser le contenu d'un répertoire ? Citez en deux options et précisez à quoi elles servent.

Réponse :

`ls`. Les options les plus utiles sont par exemple `-a` pour lister également les fichiers commençant par `« . »`, `-l` qui permet un affichage large ou encore `-t`, qui trie suivant la date de dernière modification.

Question 3 – 2 points

Vous vous trouvez dans votre `« home »`, dont le chemin absolu est `/home/numero_étudiant/`. Quel est l'effet des commandes `cd .`, `cd ..` et `cd ../..` ?

Réponse :

`cd .` se déplace vers le répertoire courant (c'est à dire qu'elle ne fait rien !), tandis que les suivantes remontent respectivement d'un ou deux crans dans l'arborescence, c'est à dire dans `/home/` ou à la racine `/`.

Attention à la lecture du chemin ! Si le dossier courant est `/home/numero_etudiant`, `cd ..` ne nous ramène pas à `/home/numero_etudiant` (on y est déjà !).

Question 4 – 2 points

Comment créer un répertoire nommé `Rep` dans le répertoire courant ? Comment faites-vous pour créer un répertoire `R1` à l'intérieur de celui-ci ? Citez une commande (avec option(s) si besoin) permettant de supprimer `R1`.

Réponse :

`mkdir Rep`, puis `mkdir Rep/R1` ou `cd Rep/ ; mkdir R1`.

`rm -r` ou `rmdir` permettent de supprimer `R1`.

Question 5 – 2 points

Quel est l'effet de la commande `tr 'a-z' 'A-Z' < fichier` ?

Réponse :

Remplace tous les caractères minuscules du texte en entrée par la majuscule correspondante. Des exercices d'entraînement à cette commande assez utile sont disponibles parmi les micro-sujets, sur le site de l'UE.

Question 6 – 1 point

Vous utilisez `less` pour lire le contenu d'un fichier. Que tapez-vous pour rechercher la chaîne de caractère « toto » dans le fichier ? ?

Réponse :

`/toto RET.`

Ne pas confondre avec les commandes `C-s` (`emacs`, `info` principalement), et `C-f` (beaucoup d'applications graphiques, comme `firefox` ou `gedit`).

Question 7 – 2 points

Comment extraire les 10 premières lignes d'un fichier texte ? Les 10 dernières ? Les 5 dernières ?

Réponse :

La commande `head` pour les 10 premières lignes, `tail` pour les 10 dernières et `tail -n 5` pour les 5 dernières. Ces commandes ont été vues brièvement en TME, vous trouverez plus d'exercices parmi les micro-sujets.

Question 8 – 1 point

Quelles sont les trois informations données par la commande `wc` sur un fichier ?

Réponse :

Le nombre de lignes, de mots et de caractères (ou plus précisément, d'octets). Dans la plupart des cas, le nombre d'octets correspond au nombre de caractères du fichier, mais selon les encodages, ça peut ne pas être le cas.

Attention, vous êtes beaucoup à confondre les informations données par `wc` et par `ls -l`. `wc` (Word Count) ne sert qu'à compter certains éléments du fichier, pas à donner des détails génériques.

Question 9 – 1 point

Que fait la commande `chmod +x fichier` ? Et la commande `chmod -w fichier` ?

Réponse :

La première ajoute les droits d'exécution à `fichier` (à tout le monde). La seconde enlève les droits en écriture.

Question 10 – 2 points

Que fait la commande `find /usr/bin -type f -exec grep -l bash {} \;` ?

Réponse :

Recherche et affiche (option `-l` de `grep`) les fichiers (option `-type f` de `find`) contenant la chaîne `bash` (option `-exec grep "bash" {} \;`) dans l'arborescence de `/usr/bin`.

Question 11 – 2 points

Citez deux façons (différentes !) de tuer un processus.

Réponse :

- on récupère le numéro *pid* du processus avec `ps` et on emploie `kill pid`.
- avec la commande `top` on récupère le numéro *pid* du processus et on lance la commande interne `k` à qui on fournit le *pid* et le numéro du signal à envoyer.
- `killall` commande au risque de tuer d'autres processus associés à la même commande.
- avec la commande `top` on lance la commande interne `k` à qui on fournit le numéro du processus à tuer.
- On se place dans le terminal ayant lancé le processus, et on récupère son identifiant avec `jobs`, on utilise ensuite `kill`.

On peut également noter la solution `Ctrl-C` (qui ne marche que si le processus est en avant-plan dans le terminal courant). Des approches plus extrêmes comme fermer le terminal, voire éteindre la machine (tant qu'on y est...) fonctionnent aussi, mais seront évidemment moins utilisables au jour le jour.

Question 12 – 2 points

Un fichier `liste_panthéon` regroupe les personnes inhumées au Panthéon, à raison d'un individu par ligne. Les lignes sont sous la forme `Nom/Prénom/Date Naissance/Date Mort/qualité`, voici les six premières :

```
Baudin/Jean-Baptiste/1811/1851/Homme politique
Béguinot/François Barthélémy/1747/1808/Militaire
Berthelot/Marcelin/1827/1907/Scientifique
Berthelot/Sophie/1837/1907/Scientifique
Bevière/Jean-Baptiste-Pierre/1723/1807/Homme politique
Bioman/Chōdenshi/1984/2000/Robot
```

Quel est l'effet de la commande `sort -t/ -k3 -n liste_panthéon` ?

Réponse :

Elle trie (`sort` signifie «trier» en anglais) le fichier `liste_panthéon` par ordre croissant (`-n`, option par défaut) sur les dates de naissance des individus : troisième (option `-k3`) champ séparé par des `/` (option `-d/`).

Question 13 – BONUS 1+2 points

Écrire la syntaxe d'une macro emacs qui itérée 10 fois, insère la chaîne "0123456789" dans le buffer. Puis écrire la syntaxe d'une autre macro qui, itérée 9 fois à partir de la chaîne précédente, produit le texte suivant :

```
0123456789
1234567890
2345678901
3456789012
```

4567890123
5678901234
6789012345
7890123456
8901234567
9012345678

On pourra utiliser notamment les touches F3, F4, M (meta, c'est à dire la touche marquée « ALT »), C (touche « Control »), et RET (retour chariot, c'est à dire « Entrée »).

Si vous ne connaissez pas les combinaisons de touches nécessaires, il est possible de gagner des points en donnant une description précise de l'action à effectuer, par exemple « positionnement du curseur en fin de ligne ». Vous pouvez également préciser ce qu'est sensée produire une combinaison dont vous n'êtes pas sûrs.

Réponse :

Pour la première macro, il suffit d'insérer un compteur :

```
( <f3>)  
<f3>; kmacro-start-macro-or-insert-counter  
( <f4>)
```

Pour la seconde macro, il y avait de nombreuses manières de procéder, par exemple en coupant le premier caractère de la ligne et en le collant en fin de ligne :

```
( <f3>)  
C-a;; beginning-of-line  
C-k;; kill-line  
C-y;; yank  
RET;; newline  
C-y;; yank  
C-a;; beginning-of-line  
C-SPC;; set-mark-command  
C-f;; forward-char  
C-w;; kill-region  
C-e;; end-of-line  
C-y;; yank  
( <f4>)
```

ou, de manière peut-être un peu plus originale, en échangeant des caractères :

```
( <f3>)  
C-a;; beginning-of-line  
C-k;; kill-line  
C-y;; yank  
RET;; newline  
C-y;; yank  
C-a;; beginning-of-line  
C-f;; forward-char  
C-u 9 C-t;; transpose-chars  
( <f4>)
```

Une macro est comme un programme informatique, il est très important de savoir où est-ce que l'on commence et où est-ce que l'on finit. En particulier, la première commande d'une macro doit en général servir uniquement à replacer le curseur là où l'on veut que l'exécution commence.