

TP4-5 d'administration Unix

F. Butelle

M. Mayo

2015

Conseil : rédiger un compte rendu de TP.

1 Les variables Shell en bash

```
# rappel : les commentaires commencent par #
# à faire en /bin/bash,
echo $SHELL
var='une variable'
echo $var
echo 'bonjour, '$var
age=2
echo toto a ${age}2 ans
echo a$b
echo "toto a ${age}2 ans"
echo 'toto a ${age}2 ans'
echo `toto a ${age}2 ans`

Fichier=$(ls /usr/include/*.h |grep std)
echo $Fichier
```

```
F='truc/rep/toto.tar.gz'
echo ${F#*/}
echo ${F##*/}
echo ${F%.*}
echo ${F%.*}

var1=bonjour
var2=aurevoir
export var2
echo $var1
echo $var2
bash
echo $var1
echo $var2
exit
```

2 Gestion de fichier

Pour chaque exemple, créez-vous des fichiers texte de test.

1. Que fait la commande `sed -e 's/\([0-9][0-9]*\)/**\1**/' fichier` ?
2. Ecrire un script sed qui remplace, dans un fichier, les caractères =E0, =E8, =E9, =3D ,... par â, è, é, =, etc.
Il s'agit de l'encodage de type "Quoted-printable", que l'on retrouve dans les entêtes de mail par exemple. Attention, ici on suppose la norme ISO 8859-15 pour l'encodage des accents, voir `man iso_8859-15`. Attention UTF8 est différent.
3. Ecrire une commande sed qui va récupérer dans `/etc/passwd` les logins et leurs UID correspondants (et rien de plus).
4. Ecrire une commande sed qui efface toutes les lignes d'un fichier, sauf les lignes commençant par From
5. Afficher le nombre total de lignes du fichier `/etc/passwd` avec awk
6. Ecrire une commande awk qui affiche chaque ligne du fichier `/etc/passwd` après avoir effacé le deuxième champ
7. Afficher le login et la date et heure de connexion de tous les utilisateurs connectés. On pourra utiliser la commande "who".
8. Afficher les lignes de plus de 60 caractères de `/etc/passwd`. (indication : print équivaut à print \$0)
9. A partir d'un fichier de notes formaté `Prénom Nom Note1 Note2`, créer un autre fichier qui contienne `Nom Prénom Note1 Note2 Moyenne` trié suivant l'ordre croissant des moyennes.
10. Même question mais sans connaître à l'avance le nombre de notes dans le fichier (sans le tri final).
11. Ecrire une commande awk qui analyse `/etc/passwd` et qui affiche les utilisateurs groupés par GID (indication : l'espace est l'opérateur de concaténation de chaînes de caractères).
12. Comment faire des tableaux à deux dimensions avec awk ? Essayez !

13. Une partition de type home est pleine, on cherche le coupable, pour cela on voudrait une liste utilisateur-groupe avec la place disque utilisée. On utilisera comme input `ls -AslR`. Il serait bon aussi d'avoir pour cette liste un cumul intégré utilisateur-* (groupe quelconque) ainsi que *-groupe et aussi *-* pour le total global.
- Astuce : on supposera qu'il n'y a pas d'espace dans les noms de fichiers, et on utilisera un tableau à 2 dimensions.

3 Les signaux, avec Python

Indications : importer le module time (`from time import *`) pour sleep, et le module signal pour l'appel signal. Autre source d'information : `man 7 signal`. Note : un programme python doit commencer par `#!/usr/bin/python` et avoir les droits d'exécution ou encore être lancé par `python <fichier>`

1. Écrire un programme qui affiche "Hello" toutes les secondes.
2. Lancer le programme, le stopper avec CTRL-Z, le relancer, puis le tuer avec CTRL-C.
3. Modifier le programme pour qu'il ignore le CTRL-Z.
4. Modifier le programme pour qu'il affiche un message lors de la réception de CTRL-Z.
5. Modifier le programme pour qu'il affiche "signal X reçu" quel que soit le signal numéro X reçu. On peut faire une boucle en python avec la commande for, mais attention, python refuse de détourner les signaux 9 et 19.
6. Quel signal doit-on envoyer à ce dernier programme pour le tuer ?