

Sujet de TP - Systèmes d'exploitation

Utilisation du Korn shell

Caractère spéciaux du terminal

Lancez un xterm.

- Avec la commande **stty -a**, observez les caractères spéciaux reconnus par le terminal. Relevez les combinaisons permettant de générer les signaux INTR, QUIT et EOF.
- Lancez la commande **ls -R /** et interrompez la avec le signal INTR.
- Lancez la commande **ls -R /** et interrompez la avec le signal QUIT.
- Lancez la commande **top &**. La commande **top** est ainsi placée en arrière plan. Générez un EOF au clavier. Qu'observez-vous? Générez à nouveau EOF au clavier.
- Redémarrez un xterm. Redéfinissez la combinaison pour émettre un EOF avec la combinaison CTRL+N. Emettre EOF avec la nouvelle définition.
- Redémarrez un xterm et lancez la commande **ksh**. Tapez **set -o** pour afficher les options de démarrage du shell et positionner l'option emacs par **set -o emacs**.
- Exécutez la commande **mail -S pour \$LOGIN_VOISIN_DROIT \$VOTRE_LOGIN** et contrôlez vos mails avec la commande **mail**.
- Corrigez la ligne de commande précédente en plaçant le curseur en début de ligne avec CTRL+A et en remplaçant **-S** par **-s**. Contrôlez vos mails avec la commande **mail**.
- Corrigez la ligne de commande précédente en plaçant le curseur devant **\$VOTRE_LOGIN** et en supprimant le sujet avec CTRL+W. Remplacez le par **"pour \$LOGIN_VOISIN_DROIT"**. Validez la commande avec CTRL+J et contrôlez vos mails.
- Tapez **set -o vi** et placez vous sur la ligne de commande précédente. En cas de pb, émettez KILL. Positionnez l'option emacs à nouveau, ainsi que l'option ignoreeof. Emettez le signal EOF. Qu'observez vous? Si d'aventure votre terminal se fermait, relancez-le et lancez la commande ksh avec les options emacs (sauf si vous préférez déjà vi) et ignoreeof.

Redirections

- Lancez **ls mail123**. Si ce fichier n'existe pas une erreur est renvoyée.
- Lancez **ls mail123 2>/dev/null**.
- Lancez **ls mail mail123 2>/dev/null**.
- Lancez **ls mail mail123 1>LS_OUT 2>LS_ERR**. Listez le contenu de ces fichiers avec la commande **cat**.
- Lancez **ls /home mail mail123 1>LS_OUT 2>LS_ERR**. Listez le contenu de ces fichiers avec la commande **cat**. Quelles différences avec les contenus précédents faites vous?
- Lancez **ls mail mail123 1>>LS_OUT 2>>LS_ERR**. Listez le contenu de ces fichiers avec la commande **cat**.
- Lancez **ls mail mail123 1>LS_OUT 2>&1**. Quelle redirection a été produite?

Patrons

- Créez des familles de fichiers commençant par fam1, fam2 ou fam3 et d'extensions .tex, .log, .dvi ou .ps. Utilisez pour cela la commande **touch** et les caractères { et }.
- Listez tous les fichiers.
- Listez tous les fichiers d'extensions .ps.
- Listez tous les fichiers dont l'extension n'est pas .ps.
- Listez tous les fichiers comportant un caractère **1** ou **3** et d'extensions .dvi ou .ps.
- Créez la famille de fichiers commençant par fam et d'extensions .tex, .log, .dvi ou .ps. Listez maintenant les noms de fichiers ne contenant pas de chiffres. Vous essayerez d'utiliser un patron avec [:digit:] puis un autre avec 0-9.

Quotes

- Créez un répertoire de test nommé TestKsh dans votre home directory. Positionnez vous dans ce répertoire. Créez un fichier de nom Mon Fichier dans le répertoire courant.
- Faites afficher la chaîne (*_*) en utilisant la commande **echo** ou mieux **print**... (pourquoi mieux?..)
- Créez le fichier nommé (fich_1).
- Faites afficher la chaîne (*_*) (en utilisant la commande **print**).
- Créez les fichiers nommés FichSuperImportant et *. Supprimez le fichier nommé * (et seulement lui!).
- Créez les fichiers Zôzô, Zéz? et Zâzâ. Listez vos fichiers. Hum... Demandez à votre binome de détourner son attention afin de supprimer l'un des deux fichiers Zôzô ou Zâzâ sans lui indiquer lequel. Votre binome doit supprimer l'autre fichier en laissant Zéz? intact.
- Faites le ménage en supprimant les fichiers de noms dangereux.

Variables

- Stockez dans la variable BINOME la chaîne "prenom1 nom1 prenom2 nom2". Affichez le contenu de la variable.
- En utilisant le mécanisme d'expansion de variable, faites afficher la chaîne : "Nous sommes prenom1 nom1 et prenom2 nom2".
- Créez dans un répertoire nommé Sound les fichiers titre1.ogg titre2.ogg ... titre24.ogg. Faites afficher la chaîne : "Nous allons numéroter les fichiers en fonctions de leur date de création. Par exemple, si le fichier titre14 a été créé le quatorzième créé, il sera renommé 14-titre14.ogg". Pour l'affichage des noms de fichiers dans cette chaîne, vous utiliserez une variable FILE dans lequel sera placé à l'avance le nom de fichier titre14.ogg.

Substitution de commande et expression arithmétiques

- Envoyez-vous un mail de sujet "AutoMail" sans taper votre adresse mail. Utilisez pour cela la sortie de la commande **whoami**.
- Affichez la déprimante chaîne : "Il est HH:MM. Il reste XX minutes à tenir avant 21h. Pfff...", où HH :MM symbolise l'heure actuelle à la précision des minutes et XX est le nombre calculé de minutes restantes avant 21h. On utilisera la sortie de la commande **date** pour construire ces chaînes.

Pipeline de commandes

- En combinant les commandes **ls** et **wc**, affichez le nombre de fichiers présents dans votre pwd. N'hésitez pas à consulter les pages man. Vérifiez que la sortie indique bien le nombre correct de fichiers.
- Redirigez la sortie de la commande **man ksh** vers le fichier Man_Ksh. Visualisez le contenu de Man_Ksh avec **less**. La commande **a2ps** permet de convertir son entrée en postscript. Une application standard pour manipuler des fichiers postscript est **gv** (un clone de **ghostview**). Appliquez **a2ps** à Man_Ksh en redirigeant sa sortie vers l'entrée de **gv**.
- Entre les commandes **a2ps** et **gv**, intercalez à nouveau la commande **a2ps** afin d'obtenir une sortie comportant quatre écrans par page.
- Supprimez le fichier Man_Ksh et utilisez cette fois la sortie de la commande **man ksh** pour afficher le même document que précédemment.

Commandes composées (for f in word do liste-de-commandes done)

- Supprimez tous les fichiers du répertoire, et utilisez un **for** pour re-créez les fichiers titreXX.ogg, où XX=1,...,24 et ce avec une seconde entre chaque création. Pour forcer le délai, utiliser la commande **sleep**.
- Créer un répertoire Titre. Pour chaque élément du répertoire Sound, produisez un affichage indiquant "Fichier" si le fichier est un fichier régulier ou "pas Fichier" sinon.
- Renommez chaque fichier d'extention .ogg en changeant cette extension par l'extention .EXT.

- ☺ Pour chaque élément du répertoire Sound, produisez un affichage indiquant "Fichier" si le fichier est un fichier régulier ou "pas Fichier" sinon. Mais cette fois, effectuez un affichage sans passage à la ligne. Après un affichage de courte durée, effacez les caractères inscrits pour afficher les informations portant sur le prochain fichier.

Script

- Créez le fichier Script1 dont la première ligne est `#!/bin/ksh`. Vous utiliserez pour cela une redirection de la commande **print** ou **echo**. Contrôlez le contenu du fichier en exécutant **cat Script1**. Ajoutez la ligne `print "Execution du Script1"# Controle d'entree dans le script` à Script, par le biais d'une redirection. Contrôlez le contenu du fichier...
- Exécutez le script en le passant en paramètre à la commande **ksh**.
- Votre but ici est de contruire la suite du script Script1 dont l'exécution engendrera une numérotation des fichiers réguliers du répertoire courant selon leur ordre de création. Ainsi, le premier fichier créé dans le répertoire sera renommé 1-Titre1.EXT. La structure logique des opérations pourra être la suivante : pour chaque élément de la liste triée par date de création des objets du répertoire courant, si l'élément est un fichier régulier, alors le renommer avec le numéro correct. Pour vous éviter de longues minutes d'exploration des pages man, sachez que la commande **ls -tr** liste les objets du répertoire courant du fichier créé le plus ancien au fichier le plus récent.
- ★ Remarquez que la commande **ls** appliquée au répertoire produit le listing :

```
1-titre1.EXT
10-titre10.EXT
11-titre11.EXT
...
2-titre2.EXT
20-titre20.EXT
...
```

Renommez les fichiers comme ils l'étaient initialement (ou supprimez les actuels et recréez les fichiers initiaux) et remodelez Script1 de façon à ce que tous les nombres utilisés pour la numérotation comportent le même nombre de chiffres en préfixant le nombre nécessaire de zeros au nombre construit. Pour cela, le nombre de fichiers à renommer sera d'abord déterminé. Le nombre de chiffres composant ce nombre sera déterminé. Parmi les moyens variés par cela, considérez la simple sortie de la commande **print "a=1(57)/1(10);scale=0;(a/1)*1+1" |bc -l**. Elle donne bien la réponse 2 qui correspond bien au nombre de chiffre. Vous pouvez vérifier pour 100 et 12345678. Puis pour chaque fichier, la chaîne correspondant à son numéro d'ordre préfixée d'une chaîne de zeros (éventuellement vide) sera construite pour opérer sa rebaptisation.
- Créez le fichier "Artiste Truc". Le script se comporte-t-il bien avec l'arrivée de ce nouveau fichier ? Si ce n'est pas le cas, déterminez l'origine du problème.
- ★★ Construisez un autre script gérant les fichiers dont le nom comporte des espaces. Indice : considérez la sortie de la commande **FILE=\$(ls -tr |sed -e 's/\$/\\/g'); echo \${FILE}; IFS='/' ; echo \${FILE}**.
- Pour finir en douceur, définissez un alias sur la commande **ls -l** avec la chaîne "l". Nous allons écrire une commande qui simule l'exécution d'un script par ksh. Ce script aurait pour contenu :

```
#!/bin/ksh
l
alias
```

 Pour cela utilisez la commande **ksh - <<FIN_DU_SCRIPT**. La chaîne FIN_DU_SCRIPT placée après le << (*here document*) indique le mot clé qui servira à indiquer la fin du script. Validez. Le prompt secondaire est affiché, indiquant que ksh attend la suite de l'input. Saisissez le corps du script, puis, sur une nouvelle ligne, saisissez FIN_DU_SCRIPT. Qu'observez vous ?