MN 2.1 Initiation Unix – TP2 : Shell zsh

Olivier Dalle (`Olivier.Dalle@unice.fr`)

---

1 Echauffement

Terminez la feuille d'exercice de la première séance si ce n'est pas déjà fait ...

1.1 Manipulation sur le contenu des fichiers, sans utiliser d'expressions régulières

Quel est l'effet des commandes suivantes :

- `date | cut -c12-16`
- `date | awk '{print $2, $6}'`
- `ps -aux | sort`
- `ps -aux | cut -f2,4` (pourquoi y-a-t'il "tout" ?)
- `ps -aux | cut -f2,4 -d" "` (pourquoi manque-t'il des informations ?)
- `ps -aux | awk '{print $2 " " $2}'` (comment supprimer la première ligne ?)

1.2 Utilisation de `find`

Faites afficher le nombre de sous-répertoires de toute l'arborescence du répertoire `/usr`. Utilisez l'option de find `-type d` qui permet de préciser le type des objets recherchés (ici, d pour "directory"). N'utilisez pas `wc` pour compter, mais awk. Lancez la commande en arrière-plan (car elle est longue !) et en enlevant les messages d'erreurs (redirection vers le fichier `/dev/null`).

2 Variables

2.1 Variables d'Environnement

- Créez de nom `vara` contenant la chaîne `toto`.
- A l'aide d ela commande `echo`, affichez le contenant de `vara`
- Lancez la commande `zsh` de façon à créer un nouveau processus shell "au-dessus" du shell courant.
- Affichez à nouveau le contenu de variable `vara`. Rien ne s'affiche ? Pourquoi ?
- Revenez au shell parent (en tapant la commande `exit`), puis affichez à nouveau `vara`. Cette fois-ci vous devriez retrouver son contenu.
- Tapez la commande `export vara`, puis affichez à nouveau le contenu de cette variable depuis le shell courant d'abord, et depuis un shell fils ensuite.
- Faites maintenant l'expérience inverse : créez un shell fils, puis dans ce shell déclarez une variable `varb` contenant la chaîne `titi`.
- Quand vous revenez au shell parent, cette variable existe-t-elle toujours ?
- Quand vous créez à nouveau un shell fils, la variable `varb` existe-t-elle à nouveau ?
- Dans un shell fils créez une variable `varc` contenant la chaîne `tutu`. Puis utilisez la commande `export` pour faire passer cette variable dans l'environnement. Lorsque vous revenez au shell parent, cette variable d'environnement du fils existe-t-elle dans l'environnement du père ?

2.2 Concaténation de variables chaînes

Exécutez et interprétez le résultat des commandes suivantes :

- `a=toto ; b=titi ; echo $a$b`
- `a="toto ; b=titi" ; echo $a$b`
- `a=toto ; salut ; echo $a$b`
- `a=toto \; salut ; echo $a$b`
- `a=toto\ \;\ salut ; echo $a$b`
- `a="toto ; salut" ; echo $a$b`
- `a="toto ; $b" ; echo $a$b`

2.3 Ambiguité de nom

Exécutez et interprétez le résultat des commandes suivantes :

- `vara=toto ; varaa=titi`
- `echo $vara`
- `echo $varaa`
- `echo "$vara"a`
- `echo "${vara}a"`
- `export vara varaa`
- `echo "$vara"a`
- `echo "${vara}a"`

Sachant que vous n'avez pas vous-même déclaré la majorité des variables d'environnement de votre shell, laquelle des syntaxes suivantes vous préserve le mieux des ambiguités ?

- `$vara`
- `${vara}`
- `"$vara"`
- `"${vara}"`

2.4 Double substitution

Exécutez et interprétez le résultat des commandes suivantes :

- `a="b" ; vara="toto" ; varab="truc"`
- `echo $a ; echo $vara ; echo $varab`
- `echo $vara$a`
- `echo \$vara$a`
- `eval "echo \$vara$a"`
- `echo "eval \$vara$a"`

3 XWindow

3.1 Sources de configuration des ressources

- La variable `XAPPLERESDIR` existe-t-elle dans l'environnement de votre shell ? Si tel est le cas, supprimez sa définition à l'aide la commande `unset`
- Créez un répertoire `app-defaults` dans votre homedir.
- Dans le répertoire précédent, créez un fichier de nom `XClock` contenant la ligne suivante : `*background: green`
- Lancez la commande `xclock&`
- Créez une variable `XAPPLRESDIR` contenant le chemin absolu vers le répertoire app-defaults que vous avez créé ci-dessus. Puis exportez cette variable dans votre environnement.
- Lancez à nouveau la comande `xclock&`. (Si vous n'avez pas commis d'erreur, l'hrologe devrait s'afficher en vert.)
- Lancez la commande `xeyes&`
- Copiez le fichier `app-defaults/XClock` dans votre homedir, sous le nom `.Xdefaults` puis sous le nom `.ressources`.
- Editez le fichier `.Xdefaults` et changez `green` en `red`. Puis lancez à nouveau `xclock&` et `xeyes&`

- Editez à nouveau le fichier `.Xdefaults` et remplacez son contenanu par la déclaration suivante : `XEyes*background: yellow`
- Lancez à nouveau `xclock&` et `xeyes&`.
- Renommez le fichier `.Xdefaults` en `.Xdefaults.old` et recommencez. Que peut-on en déduire ?
- Editez le fichier `.ressources` et changez `green` en `blue`.
- Lancez à nouveau `xclock` et `xeyes`. Rien n'a changé ? c'est normal.
- Lancez la commande suivante puis à nouveau les deux commandes :
  `xrdb $HOME/.ressources`
- Renommez le fichier `.ressources` en `.ressources.old` et recommencez. Que peut en déduire cette fois-ci ?
- Vous trouverez la solution pour vous débarrasser de cette couleur embarassante dans la page de manuel de `xrdb` ...

3.2 Quelques commandes amusantes

- Lancez la commande `xev` et observez ce qui se passe quand vous déplacez la souris au-dessus. Testez aussi les frappes clavier, et clics de souris.
- Lancez le programme `xfontsel` et trouvez une fonte non proportionnelle qui vous plaise. En utilisant le bouton select et le copier/coller (Coller = bouton du milieu de la souris), testez cette fonte avec emacs.
- Lancez `xcalc` et `editres` afin de reproduire l'exemple de modification à "à chaud" présenté dans le cours.
- Trouvez un moyen de rendre la modification de couleur sur la calculette définitive et vérifiez que cela fonctionne en lançant à nouveau la culculette.