
Linux - Ligne de commande : Complément Shell

Devoir maison (*Rev* : 674) – octobre 2013 –

Académie de Nice - Lycée général et technique Les Eucalyptus
Section de Technicien Supérieur (STS)

Informatique et Réseaux pour l'Industrie et les Services techniques (IRIS)

Enseignant - **Éric Valade**

Objectif

Savoir automatiser des tâches simples en utilisant les commandes **for** et **seq**.

Pré requis

- commandes de base de Linux : **ls**, **mkdir**, **cd**, **find**, **rm**, **mv**

Ressources

A votre disposition :

- un poste de travail sous Ubuntu ;
- la commande **man** ;
- le support de cours électronique `coursPres121_135.pdf` ;

Durée

vacances de la Toussaint

Mots-clé

boucle, séquence, variable, itération

Document à rendre

Aller sur pronote télécharger le TD `LinuxQ4.pdf` (ce document) et le fichier `votreNom.sh` ; renommer le fichier `votreNom.sh` en remplaçant `votreNom` par votre nom de famille.

Pour chaque question de ce document, substituer les points d'interrogation dans le script `votreNom.sh` ; pour la question 1, insérer le résumé entre les lignes `«comment` et `comment`.

Le fichier `votreNom.sh` est à envoyer par mél eric.valade@ac-nice.fr avant le 3 novembre 2013, 18H00.

AUCUNE EXCUSE NE SERA RETENUE !

Méthodologie de travail

La figure 1 ci-dessous modélise une tâche à réaliser. Le traitement transforme les données d'entrées (*i.e.* consignes, questions, problèmes) en données de sortie (*i.e.* mise en œuvre, réponses, solutions). Le traitement à effectuer doit rester simple (*i.e.* K.I.S.S : « Keep It Simple and Stupid ». Le traitement se base sur la documentation disponible qu'il est nécessaire de lire attentivement (R.T.F.M. = « Read The Fucking Manual » = « Lis le foutu manuel »).

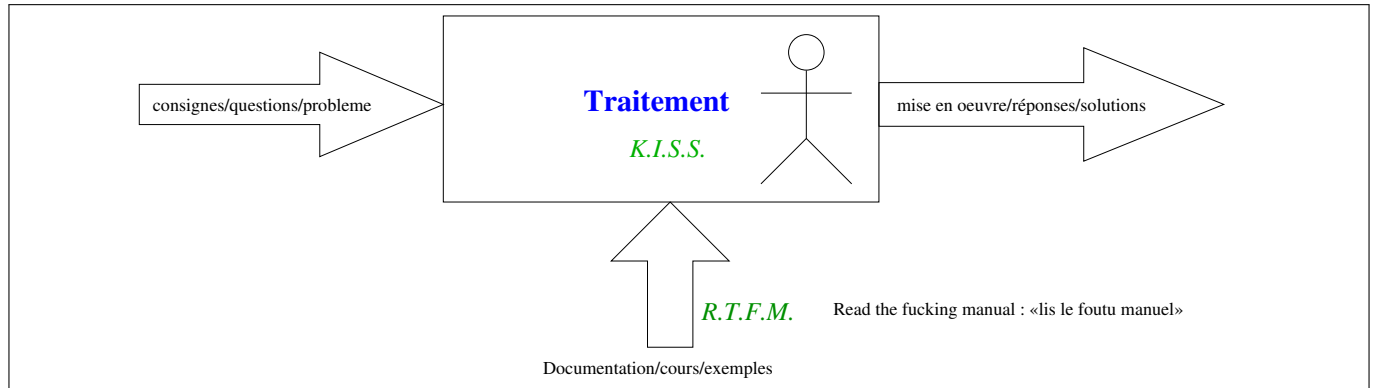


Figure 1 – Modélisation d'une tâche

La documentation pouvant être conséquente, il est nécessaire d'en faire un résumé pour faciliter le définition du traitement à effectuer. Le résumé ne contient que l'information jugée utile à la réalisation de la tâche.

Cette approche peut être synthétiser par la procédure suivante et représentée par le figure 2 ci-dessous :

1. lire le travail demandé dans son intégralité. Classifier, selon vous, les questions de la plus simple à la plus difficile.
2. lire la documentation (ici le cours pages 121 à 135),
 - (a) prêter attention aux exemples,
 - (b) mettre en évidence (stylo de couleur, stabilo) le contenu qui pourrait vous aider à répondre aux questions. Associer le(s) numéro(s) de questions avec le contenu.
3. extraire de la documentation le contenu précédemment sélectionné ; autrement dit, rédiger un résumé
4. essayer de répondre aux questions en ne se servant que du résumé.

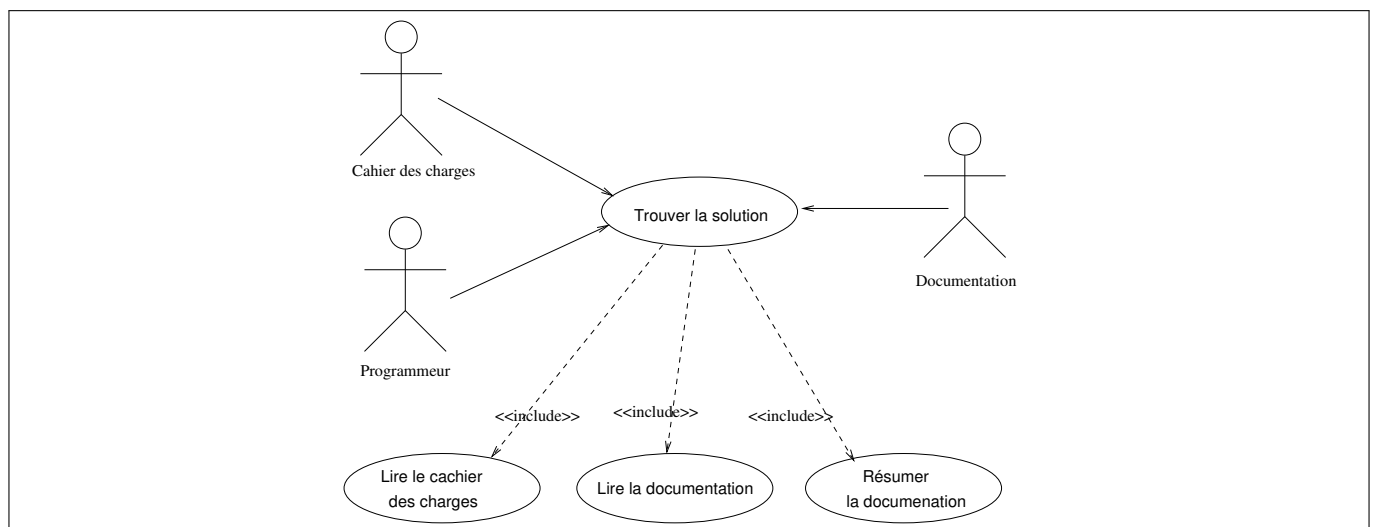


Figure 2 – Procédure conseillée

Travail demandé

Dans les commandes utilisées ci-après, remplacer les `[?]` par les chaînes de caractères adéquates. Commandes à utiliser : **for**, **seq**, **basename**, **cut**, **ls**, **touch**.

1. Donner un résumé de la documentation. Faire apparaître les commandes pouvant être utiles.
2. Créer le répertoire `linux/java`, et sous ce répertoire, créer automatiquement plusieurs fichiers (vides) dont les noms sont de la formes `tpX.exoY.java` avec `X` prenant les valeurs de `1` à `10` et `Y` prenant les valeurs de `1` à `5` (soit 50 fichiers au total).

```
mkdir [?]
cd [?] java
for x in $([?] 1 10);do [?] y in $(seq [?] 5);do touch tp [?] .exo [?] .java ; done [?]
```

Résultat (partiel) attendu (commande `ls`)

```
tp10.exo1.java  tp2.exo1.java  tp4.exo1.java  tp6.exo1.java  tp8.exo1.java
tp10.exo2.java  tp2.exo2.java  tp4.exo2.java  tp6.exo2.java  tp8.exo2.java
tp10.exo3.java  tp2.exo3.java  tp4.exo3.java  tp6.exo3.java  tp8.exo3.java
tp10.exo4.java  tp2.exo4.java  tp4.exo4.java  tp6.exo4.java  tp8.exo4.java
tp10.exo5.java  tp2.exo5.java  tp4.exo5.java  tp6.exo5.java  tp8.exo5.java
.....
```

3. Fabriquer pour chaque fichier de nom `tp9.exo*.java` ou `tp10.exo*.java` une copie de nom `tp9.exo*.java.bak` ou `tp10.exo*.java.bak`.

```
for file in [?] tp9.exo [?] .java tp10.exo [?] .java) [?] do cp [?] [?] ; done
```

Résultat (partiel) attendu (commande `ls`)

```
tp10.exo1.java  tp2.exo1.java  tp5.exo1.java  tp8.exo1.java
tp10.exo1.java.bak  tp2.exo2.java  tp5.exo2.java  tp8.exo2.java
tp10.exo2.java  tp2.exo3.java  tp5.exo3.java  tp8.exo3.java
tp10.exo2.java.bak  tp2.exo4.java  tp5.exo4.java  tp8.exo4.java
tp10.exo3.java  tp2.exo5.java  tp5.exo5.java  tp8.exo5.java
tp10.exo3.java.bak  tp3.exo1.java  tp6.exo1.java  tp9.exo1.java
tp10.exo4.java  tp3.exo2.java  tp6.exo2.java  tp9.exo1.java.bak
tp10.exo4.java.bak  tp3.exo3.java  tp6.exo3.java  tp9.exo2.java
.....
```

4. Renommer chaque fichier de nom `tp10.exo*.java.bak` en `tp10.exo*.txt`.

```
for file in $(ls tp10.exo*.java.bak);do mv $file $( [?] $file '.java.bak').txt ; done
```

ou

```
for file in $(ls tp10.exo*.java.bak);do mv $file $(echo $file|[?] -d'.' -f1,2).txt ; done
```

Résultat (partiel) attendu (commande `ls`)

```
tp10.exo1.java  tp1.exo3.java  tp3.exo5.java  tp6.exo2.java  tp8.exo4.java
tp10.exo1.txt  tp1.exo4.java  tp4.exo1.java  tp6.exo3.java  tp8.exo5.java
tp10.exo2.java  tp1.exo5.java  tp4.exo2.java  tp6.exo4.java  tp9.exo1.java
tp10.exo2.txt  tp2.exo1.java  tp4.exo3.java  tp6.exo5.java  tp9.exo1.java.bak
tp10.exo3.java  tp2.exo2.java  tp4.exo4.java  tp7.exo1.java  tp9.exo2.java
tp10.exo3.txt  tp2.exo3.java  tp4.exo5.java  tp7.exo2.java  tp9.exo2.java.bak
tp10.exo4.java  tp2.exo4.java  tp5.exo1.java  tp7.exo3.java  tp9.exo3.java
tp10.exo4.txt  tp2.exo5.java  tp5.exo2.java  tp7.exo4.java  tp9.exo3.java.bak
.....
```

5. Créer automatiquement sous le répertoire `linux/java`, 10 répertoires `tp1`, `tp2`, ..., `tp10`

```
mkdir tp{ [?]
```

ou

```
for i in [?];do mkdir tp$i;done
```

ou

```
for i in $( [?] );do mkdir tp$i;done
```

ou

```
eval mkdir tp{$(seq -s [?])}
```

Résultat (partiel) attendu (commande **ls -F**)

```
tp1/          tp1.exo3.java  tp3.exo5.java  tp6.exo1.java  tp8.exo3.java
tp10/         tp1.exo4.java  tp4/          tp6.exo2.java  tp8.exo4.java
tp10.exo1.java tp1.exo5.java  tp4.exo1.java  tp6.exo3.java  tp8.exo5.java
tp10.exo1.txt tp2/          tp4.exo2.java  tp6.exo4.java  tp9/
.....
```

6. Déplacer tout fichier de nom `tpX*` dans le répertoire `tpX`.

```
for [?] in $( [?] );do mv tp [?] tp [?] ;done
```

Résultat (partiel) attendu (commande **tree**)

```
.
|-- tp1
|   |-- tp1.exo1.java
|   |-- tp1.exo2.java
|   |-- tp1.exo3.java
|   |-- tp1.exo4.java
|   |-- tp1.exo5.java
|-- tp10
|   |-- tp10.exo1.java
|   |-- tp10.exo1.txt
|   |-- ....
|-- tp2
|   |-- tp2.exo1.java
|   |-- tp2.exo2.java
|   |-- tp2.exo3.java
|   |-- tp2.exo4.java
|   |-- tp2.exo5.java
|-- ...
|-- tp9
|   |-- tp9.exo1.java
|   |-- tp9.exo1.java.bak
|   |-- tp9.exo2.java
|   |-- tp9.exo2.java.bak
|   |-- ...
```

7. Créer le répertoire `linux/passwd`, et sous ce répertoire, créer automatiquement autant de sous répertoires qu'il y a de lignes dans le fichier `/etc/passwd`, chaque sous répertoire ayant pour le nom le premier champ de chaque ligne (par exemple `root` pour la première ligne)

```
mkdir [?] cd [?]
mkdir $( [?] /etc/passwd)
```

Exemple

```
~/linux/>$head /etc/passwd # commande
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
```

Résultat attendu (commande `ls -F passwd`)

```
bin/  daemon/  games/  lp/  man/  mail/  news/
root/  sync/  sys/
```

8. Créer automatiquement dans chaque répertoire de nom `rep` du répertoire `linux/passwd` un fichier de nom `info.txt` vide (où `rep` est l'un des répertoires créés précédemment).

```
[?] = `ls`; for rep in [?] reps ;do touch [?] /info.txt; [?]
```

ou sans utiliser la variable `reps` :

```
for rep in [?] ;do touch [?] /info.txt;done
```

ou sans boucle :

```
find . -type [?] -name [?] -exec touch [?] /info.txt \;
```

Quel est l'inconvénient de cette dernière solution ?

Résultat (partiel) attendu (commande `tree`)

```
bin/
|__ info.txt
daemon/
|__ info.txt
games/
|__ info.txt
....
```

9. Ajouter à chaque fichier `info.txt` du répertoire `linux/passwd/` le nom en clair de l'utilisateur.

```
for rep in `ls`;do [?] $rep [?] $rep/info.txt [?] done
```

Résultat (partiel) attendu (commande `tree`)

```
~/linux/passwd$ cat bin/info.txt
bin
~/linux/passwd$ cat daemon/info.txt
daemon
```

10. Renommer tous les fichiers `linux/passwd/rep/info.txt` en `linux/passwd/info.rep` et effacer tous les sous répertoire `linux/passwd/rep` (où `rep` est l'un des répertoires créé précédemment)

```
[?] rep [?] `ls`; [?] mv $rep/info.txt [?] rmdir [?] ;done
```

ou

```
? rep ? `ls`; ? mv $rep/info.txt $( ? info.txt.txt). ? ; rmdir ? ;done
```

Résultat (partiel) attendu (commande **ls -F**)

```
info.bin  info.daemon  info.games  info.lp      info.man  
info.mail info.news     info.root   info.sync    info.sys
```