Intro Système

TD/TP #3

Guillaume Santini
guillaume.santini@iutv.univ-paris13.fr

IUT de Villetaneuse - Paris13
Département d' informatique

Préparation du TP

- **Téléchargez les données nécessaires au TP. Celles-ci sont dans le fichier `donnees_tdtp3.tar.gz` disponibles sur le site http://www-lipn.univ-paris13.fr/~santini/documents_pedagogiques.**
  Les documents téléchargés sont enregistrés dans le répertoire `~/Desktop` dans votre répertoire personnel.
- **Q1: Quelle séquence de commandes permet de vérifier que le fichier a été téléchargé à cet endroit?**
  Quel est le type de fichier que vous venez de télécharger?
- **Dans la suite des correction nous supposerons toujours que vous avez tapé toutes les commandes des précédents exercices dans l'ordre.**
- **Q2: Quelle commande permet de vérifier que votre répertoire courant est `~/Desktop`?**
  Quelle commande permet de revenir dans votre répertoire personnel?
- **Q3: Quelle commande permet de créer un répertoire `TP_3/` dans le répertoire `~/Intro_Systeme/` définit lors des précédents TP?**
  Vous utiliserez un chemin relatif pour ne pas avoir à changer de répertoire.
- **Q4: Quelle commande permet de déplacer le fichier téléchargé dans le répertoire `~/Intro_Systeme/TP_3`?**
  Vous utiliserez encore des chemins relatifs, et ne changerez pas de répertoire.
- **Q5: Quelle commande permet de vous placer dans le répertoire contenant désormais l'archive?**
- **Q6: La commande `tar xvzf` *`nom_fichier_archive.tar.gz`* permet d'extraire et de décompresser l'archive en une seule fois.**
  Décompressez le fichier téléchargé et identifiez le nom du répertoire qui a été créé.

Exploration des données et édition d'un fichier html

- **Q1: Placez-vous dans le répertoire `donnees/`.**
  Combien de fichiers de page web (extension `.html`) identifiez-vous?
- **Q2: Créez un répertoire `Intro_Systeme/` dans le répertoire `public_html/` se trouvant dans votre répertoire personnel.**
- **Q3: Copiez le fichier `photo_espace.html` et le répertoire `images/` (avec son contenu) dans le répertoire `~/public_html/Intro_Systeme` que vous venez de créer.**

- **Désormais la page web `photo_espace.html` est accessible depuis le réseau depuis l'adresse suivante:**

  `http://aquanux/~`*`votre_identifiant`*`/Intro_Systeme/photo_espace.html`
- **Q4: Affichez la page `photos_espace.html` dans le navigateur.**
- **Dans l'éditeur de texte de votre choix (`gedit` par exemple), modifiez le fichier `photos_espace.html` pour remplacer le titre 'Mon Titre', par le texte que vous voulez.**
  Enregistrez vos modifications et rafraichissez la page dans le navigateur. Vérifiez que la modification a bien été prise en compte.
- **Q5: Modifiez à nouveau le fichier source de la page web de façon à afficher 5 fois le titre.**
  Vous utiliserez les racourcis clavier CTRL+C, CTRL+V et SHIFT+CTRL+S pour faire cela rapidement.
  Enregistrez vos modifications dans un nouveau fichier.
  Afficher cette nouvelle page dans le navigateur en modifiant directement l'adresse.
  Vérifiez que la modification du titre a bien été prise en compte.
- **Dans le langage HTML, la balise `'img'` permet d'insérer dans une page web une image.**
  Dans cette balise, le champ `'src'` spécifie le fichier image qui doit être affiché.
- **Q6: Identifiez dans les sources de la page web le code correspondant à la première image de la deuxième ligne.**
  Le nom du fichier pointé est `./images/cena.jpg`.
- **Q7: Quelle commande permet de vérifier que le fichier correspondant à l'image existe bien dans l'arborescence à l'endroit spécifié?**
- **Q8: Modifiez les sources de la page web en remplacant le nom du fichier `'cena.jpg'` par `'cena_2.jpg'`.**
  Enregistrez vos modifications et rafraichissez la page dans le navigateur. Qu'observez-vous?
- **Q9: En admettant que votre répertoire courant est `~/public_html/Intro_Systeme/`, quelle commande permet modifier le nom du fichier associé à l'image pour que la page web affiche à nouveau l'image.**

La commande `grep`

Rappel de la syntaxe de la commande `grep`:

```
grep "motif" fichier
```

où:

- `"motif"` est une chaîne de caractères,
- `fichier` est le nom ou le chemin vers un fichier.

Pour le début de cette section nous supposerons que votre répertoire courant est: `~/Intro_Systeme/TP_3/donnees`.

- **Q1: À l'aide des commandes `cat` ou `less` identifiez dans le fichier `Oreadme` le nombre de fois ou le mot 'src' apparait.**
- **Q2: La commande `grep "src" Oreadme` permet d'affichez les lignes du fichier `Oreadme` où le mot 'src' apparait. Vérifiez qu'il y en le bon nombre?**
- **Q3: Répétez l'opération pour affichez les lignes du fichier `photos_espace.html` où le mot 'src' apparait. Combien y en a-t-il?**
- **Q4: L'option `-v` permet d'inverser son comportement. Au lieu d'afficher les lignes qui présentent le motif, `grep` affiche alors les lignes qui ne présentent pas le motif.**
  Affichez les lignes du fichier `Oreadme` ne présentant pas le mot 'http'? Combien y en a-t-il?
- **Q5: Quelle commande permet d'afficher les lignes du fichier `photos_espace.html` ne comportant pas le motif '<'? Combien y en a-t-il?**

Les redirections de la sortie standard

Pour le début de cette section nous supposerons que votre répertoire courant est: `~/Intro_Systeme/TP_3/`.

- **Q1: `ls -la` affiche en format long les fichiers et répertoires (y compris ceux qui sont cachés) contenu dans le répertoire courant.**
  Donnez une explication à chaque ligne de l'affichage.
- **Q2: La commande suivante crée un fichier texte:**

```
ls -la > ls.out
```

  Que contient-il ? Si vous ne savez pas, utilisez la commande `less` pour visualiser son contenu.
  Pourquoi le contenu de ce fichier est-il différent de l'affichage précédent?
- **Q3: Proposez deux commandes permettant de créer un répertoire pour les futurs fichiers de redirections que vous allez créer.**
  Ce répertoire s'appellera `redirections/`. Il est créé dans le répertoire courant.
  La première commande utilisera un chemin de répertoire relatif, l'autre s'appuira sur un chemin partant du répertoire partant de votre répertoire personnel (`~/`).
- **Q4: Que font les commandes suivantes?**

```
echo "Bonjour"
echo "Bonjour" >  redirection/bonjour.out
echo "Salut"   >  redirection/bonjour.out
echo "Bonjour" >> redirection/bonjour.out
```

  Suivez le contenu du fichier `redirection/bonjour.out`.
- **Q5: Entrainez-vous avec les commandes suivantes. Profitez-en pour comprendre les affichages produits par les commandes `ps`, `file` et `cat`:**

```
ps   >   redirections/essai_ps.out
file /usr/include/stdio.h > redirections/file.out
cat  /etc/passwd          > redirections/cat.out
```

- **La commande suivante:**

```
ls donnees/images/*.png > redirections/fichiers_PNG.txt
```

  permet d'écrire dans le fichier `redirections/fichiers_PNG.txt`, le nom de tous les fichiers du répertoire `image/` présentant l'extension `*.png`.
- **Q6: Définissez les commandes pour créer deux autres fichiers similaires (`redirections/fichiers_JPG.txt` et `redirections/fichiers_GIF.txt`) pour les images en format JPG et GIF.**
- **La commande**

```
cat fichier1 fichier2 ...
```

  permet d'afficher sur la sortie standard le contenu des fichiers passés en paramètre.
- **Q7: Définissez une commande permettant de créer un fichier concaténant les trois fichiers crées à la question Q6.**

La commande `wc`

Pour le début de cette section nous supposerons que votre répertoire courant est: `~/Intro_Systeme/TP_3/donnees/`.

La commande `wc` permet de compter les éléments d'un fichier texte. La syntaxe d'utilisation de la commande est la suivante:

`wc` *`[options]`* `fichier`

La commande `wc` affiche par défaut le nombre de lignes, de mots, et d'octets du fichier. La commande admet plusieurs options (cf. `man wc`).

- **Q1: Vérifiez le comportement de la commande wc sur le fichier `0readme`.**
  En comparant cet affichage au contenu du fichier (affiché par la commande `less`) vous identifierez les différents champs correpondant au nombre de ligne, de mots, et d'octets, de l'affichage de la commande `wc`.
- **Q2: Quelle option de la commande wc devez-vous utilser pour que celle-ci n'affiche que le nombre de lignes?**
  Si vous ne savez pas, consultez la page de man.
- **La commande `wc` fonctionne sur des fichiers dont le nom est passé en paramètre.**
  Elle peut également travailler en lisant les données sur l'entrée standard.
  Une façon de le vérifier est de taper la commande `wc`+ENTER, sans nom de fichier. Une nouvelle ligne apparait.
  Ce que vous taperez alors sera envoyé sur l'entrée standard.
  Pour signaler la fin de la rédaction sur l'entrée standard, tapez CTRL+D.
- **Q3: Faites quelques essais et vérifez le comportement de la commande `wc`.**
- **Q4: La commande `grep` fonctionne de la même façon que `wc`. Testez l'écriture directe sur l'entrée standard en tapant la commante grep `"motif"` + ENTER, puis en saisissant du texte.**
- **Q5: `wc` et `grep` sont donc des commandes qui lisent sur l'entrée standard. Vous vérifierez l'équivalence des syntaxes suivantes:**

```
wc -l 0readme          et wc -l < 0readme
grep "src" 0readme     et grep "src" < 0readme
```

Les tubes |

Pour le début de cette section nous supposerons que votre répertoire courant est: `~/Intro_Systeme/TP_3/donnees/`.

Le tube est un outil puissant permettant de combiner plusieurs commandes en redirigeant les sorties standards sur les entrées standards. Cela permet d'introduire une très grande flexibilité et de réaliser des opérations complexes.

- **Q1: Évaluez les commandes suivantes, analysez leur résultat et justifez le comportement.**
  Pour vous aidez vous pouvez évaluer les commandes pas à pas en vous arrêtant avant chaque tube.

```
grep "img" photos_espace.html | wc -l
ls images/ | grep "gif" | wc -l
```

- **Q2: La commande `ps -ef` affiche l'ensemble des processus s'exécutant sur l'ordinateur.**
  Il y en a beaucoup... Trouver parmi ceux-ci les processus bash est fastidieux.
  Proposez une commande utilisant un tube et la commande `grep` qui permet de n'afficher que les processus bash.
- **Q3: Proposez une commande permettant de compter le nombre de fichiers dans le répertoire `images/`.**

Les variables Shell

Il est possible de définir des variables dans l'environnement de chaque Shell. En bash, la syntaxe pour définir une variable est la suivante:

`nom_var=valeur`

Pour utiliser la valeur associée à la variable, il faut faire précéder le nom de celle-ci du caractère `$`. Par exemple:

```
phrase = "Bonjour tout le monde"
echo $phrase
```

produit l'écriture suivante sur la sortie standard:

```
Bonjour tout le monde
```

- **Q1: Que fait la séquence de commandes suivante?**

```
nom="Smith"
prenom="John"
serveur="iutv.univ-paris13.fr"
echo $prenom.$nom@$serveur
```

- **Q2: Que fait la séquence de commandes suivante?**

```
tp="TP_1"
ls -la ~/Intro_Systeme/$tp
```

Changez la valeur de tp pour `"TP_3"`. Quelle affichage obtenez-vous si vous réévaluez la deuxième commande?

- **Q3: Soit la séquence de commandes suivantes:**

```
source=/usr/include/stdio.h
cible=/tmp
```

A partir de là, donnez une commande permettant de copier le contenu du fichier `/usr/include/stdio.h` dans le répertoire `/tmp`. en utilisant les variables `source` et `cible`. Vous noterez l'emploi de chemins absolus pour les noms de fichiers.

Introduction aux scripts bash

Le caractère "`;`" est un séparateur de commandes. Il permet d'écrire sur une seule ligne, plusieurs commandes qui seront évaluées séquentiellement. À titre d'exemple vous pouvez taper sur une seule ligne la suite de commandes suivante:

```
echo "Debut"; sleep 2; echo "Apres 2 sec."; sleep 5; echo "Apres 5sec"
```

Le script affiche "Debut", puis marque une pause de 2 secondes, affiche "Apres 2 sec.", puis marque une pause de 5 secondes, puis affiche "Apres 5 sec." (la commande `sleep` produit l'exécution d'une pause).

L'utilisation du caractère "`;`" est donc une première façon d'adresser plusieurs commandes au système.

Une deuxième façon est de rédiger un script. Un script est un fichier exécutable contenant plusieurs commandes évaluées séquentiellement. Chaque ligne correspond à ce que vous pourriez écrire sur la ligne de commande. Il n'y a donc en théorie par besoin de "`;`".

- **Q1: Après avoir créé un repertoire nommé `~/Intro_Systeme/TP_3/scripts/` et l'avoir choisi comme répertoire courant, définissez et exécutez un script nommé `exo_1_script.sh` qui réalise la suite de commandes donnée en exemple.**
  Vous n'oublierez pas de rendre le fichier exécutable en changeant ses droits (`chmod u+x exo_1_script.sh`)

  Rappel: pour lancer un script il suffit de donner son nom sur la ligne de commande
  Rappel 2: pour contrôler les droits sur un fichier il faut utiliser la commande `ls -al` *`nom_fichier`*.
- **Un commentaire est une partie rédigée du script qui ne sera pas considérée comme une instruction lors de l'exécution du script. Les commentaires permettent d'ajouter du texte pour expliquer ce que fait le code, ou pour en neutraliser une partie.**
  Pour commenter une portion du script on utilise le caractère `#`. L'ensemble du texte situé sur la même ligne et après le carcactère `#` sera considéré comme un commentaire et ne sera pas évalué.
- **Q2: Que se passe-t-il si vous commentez les lignes commencant par la commande `sleep`?**
- **Le script `exo_2_script.sh` suivant affiche "Bonjour", définit le répertoire `~/Intro_Systeme/TP_3/scripts/` comme répertoire courant, puis crée dans celui-ci un répertoire `Test` et 3 autres sous répertoires.**

exo_2_script.sh

```
echo "Bonjour"
cd ~/Intro_Systeme/TP_3/scripts/
mkdir Test
mkdir Test/TP_1/
mkdir Test/TP_2/
mkdir Test/TP_3/
```


- **Q3: Copiez ce script dans un fichier nommé `exo_3_script.sh`, et modifiez-le pour qu'il affiche le contenu du répertoire `Test`, puis supprime les répertoires créés.**
- **Q4: Copiez ce script dans un fichier nommé `exo_4_script.sh`, et modifiez-le pour que le nom du répertoire `Test/` soit une variable dans le script.**
- **Q5: Copiez ce script dans un fichier nommé `exo_5_script.sh`, et modifiez-le pour que le nom du répertoire `Test/` soit passé comme un paramètre du script.**
- **Q6: À l'image de ce que vous avez fait dans la question Q1 de la partie Variables Shell, rédigez un script recevant 2 paramètres (nom et prenom) permettant l'affichage d'une adresse mail formatée.**

La commande `find`

La commande `find` est une commande très puissante. Elle permet de parcourir une partie ou l'ensemble de l'arborescence du système de fichier à la recherche des fichiers présentants certaines caractéristiques.

Les deux options les plus couramment utilisées sont `-iname` et `-exec`. La syntaxe de la commande est la suivante:

```
find chemin -iname "motif"
find chemin -iname "motif" -exec commandes \;
```

- Le *`chemin`* peut-être n'importe quel chemin valide de l'arborescence (y compris ~/ le répertoire personnel et `./` le répertoire courant).
  Il définit le point de départ de la recherche. L'ensemble des répertoires et sous-répertoires partant du chemin seront explorés.
- Le *`motif`* peut être n'importe quelle expression régulière (pouvant éventuellement contenir des méta-caractères tels que `*`).
- les *`commandes`* peuvent être n'importe quelle commande valide.
  L'utilisation dans les commandes de l'expression `{}`, sera remplacée lors de l'exécution par la liste des fichiers renvoyés par la commande `find`.
  La fin de la commande est indiquée par les caractères `\;`

- **Q1: En partant de du répertoire donnees extraie de l'archive au début du TP pour lancer votre recherche, trouvez tous les fichiers dont le nom présente l'extension `html`.**
- **Q2: Que fait la commande suivante?**

```
find ~/Intro_Systeme/TP_3/donnees -iname *.html -exec wc -l {} \;
```

- **Q3(Optionnelle): Après avoir créé un répertoire `Copie_PIA/` dans le répertoire `~/Intro_Systeme/TP_3/`, testez le résultat de la commande suivante:**

```
find ~ -iname *.html -exec echo cp {} Copie_PIA \;
```

Que se passera-t-il si vous suprimez `echo` de la commande?

La commande `sed`

La commande `sed` (pour stream editor) est une fonction très puissante dont nous n'aborderons qu'une seule utilisation. Elle permet entre autres choses d'opérer des substitutions de caractères ou de motifs de caractères. Sa syntaxe d'utilisation est alors la suivante:

```
sed -i 's/motif/nouveau/g' fichier
sed    's/motif/nouveau/g' < command
```

- L'option `-i` spécifie que la lecture du texte à modifier se fait à partir d'un fichier dont on donne le chemin.
  Sans cette option, `sed` lit les données sur l'entrée standard.

La structure entre guillemets détermine la nature de l'opération réalisée par la commande

- *s* indique que l'on souhaite faire une substitution,
- *motif* est une suite de lettres à rechercher dans les données,
- *nouveau* est la suite de lettres par lesquels doit être remplacé le *motif*
- *g* (pour ground), spécifie que la substitution doit être répétée tant que le motif est présent dans les données.

- **Q1: On suppose que votre répertoire courant est `~/Intro_Systeme/TP_3/donnees/`.**
  Donnez une commande permettant de substituer dans le fichier `Oreadme`, les occurences du mot 'src' par le mot 'Source'.
- **Q2: On suppose que votre répertoire courant est `~/Intro_Systeme/TP_3/donnees/`.**
  Donnez la commande permettant de substituer dans le fichier `photos_espace.html`, le nom de fichier `cena.jpg` en `cena_nebuleuse.jpg`.
  Enregistrer le résulat dans un nouveau fichier nommé `photos_espace.2.html`. Pour cela on fera appel à une redirection de la sortie standard de la commande `sed` pour écrire le nouveau fichier.
- **Q3: Ouvrez ce dernier fichier (`photos_espace.2.html`) dans un navigateur internet.**
  Que remarquez-vous?
  Quelle solution faisant appel à la commande `cp` (copy) proposeriez-vous pour rétablir un affichage correct de la page `photos_espace.2.html`.

Pour aller plus loin

- **Q1: Tester la commande `sed` sans l'option `-i`.**
- **Q2: Lacommande `grep` admet plusieurs fichiers en paramètre.**
  Tester cette option en recherchant par exemple les titres des fichiers `donnees/photos_espace.html` et `donnees/divers/index.html`.
  Dans ces fichiers, le titre de la page est encadré par la balise `<H1>`.
- **Q3: La commande `head` permet d'afficher par défaut les 10 premières lignes d'un fichier.**
  Afficher les 10 premières lignes des deux fichiers précédants.
- **Q4: Comptez les lignes du fichier `donnees/photos_espace.html` contenant le motif img et le motif png.**