Haute École de Bruxelles-Brabant
École Supérieure d'Informatique
Bachelor en Informatique

2018 – 2019
DEV1

DEV1 – Laboratoire d'environnement de développement

TD 4 – Commandes grep, wc, find et variables d'environnement

Objectifs

Ce TD a pour objectif de vous familiariser avec :

- ▷ les commandes Unix `grep`, `wc` et `find` ;
- ▷ les variables d'environnement.

Table des matières

Version du document
28 septembre 2018

1 TD4 - Les commandes Unix grep, wc, find et les variables d'environnement

Consignes

- ▷ Prenez bien note des réponses aux exercices ainsi que de la façon dont vous avez trouvé ces réponses.

1.1 Recherche dans des fichiers (grep)

Vous avez déjà écrit pas mal de programmes et il devient parfois difficile de vous y retrouver. Vous allez être amenés à vous poser des questions du genre :

- ▷ Dans quel fichier ai-je écrit un programme qui calcule une surface ?
- ▷ Dans quel fichier ai-je déjà utilisé un double ?

La commande `grep` peut venir à votre secours. Dans son utilisation la plus simple, elle permet d'extraire de fichiers toutes les lignes qui contiennent un certain texte (appelé *pattern*).

Syntaxe : `grep pattern fichier...`

Tutoriel 1 Pour trouver dans quel fichier vous avez utilisé une variable nommée "surface", vous pouvez écrire :

```
grep surface *.java
```

Rappel Lorsque vous appuyez sur la touche `TAB`, le shell tente de compléter le début de commande que vous avez déjà tapé. Si plusieurs possibilités existent, elles sont affichées si vous appuyez 2x sur `TAB`.

Exercice 1 **Recherche d'un pattern dans un fichier**

1. Comment trouver les programmes Java du TD4 où vous avez déjà utilisé un "double" ?
2. Comment trouver, parmi **tous** les programmes Java que vous avez déjà écrits, ceux qui utilisent des entiers ?

1.2 Compter (wc)

Vous vous demandez peut-être combien de lignes fait un programme que vous avez écrit ou encore combien de lignes vous avez écrites aujourd'hui. La commande `wc` peut vous répondre ; elle indique le nombre de lignes, de mots et de caractères contenus dans les fichiers donnés.

Syntaxe : `wc fichier...`

Tutoriel 2 La commande suivant affiche le nombre de lignes, mots et caractères contenus dans le fichier `Ex2.java`.

```
wc Ex2.java
```

Exercice 2 **Compter le nombre de lignes**

1. Comment afficher le nombres de lignes de tous vos fichiers Java de ce TD.
2. Examinez les options de la commande et trouvez comment n'afficher **que** le nombre de lignes et pas le nombre de mots et de caractères.
3. Une convention d'écriture Java indique de ne pas dépasser la colonne 80 dans les programmes. Trouvez l'option qui permet de vérifier que tous vos programmes actuels vérifient cette convention.

1.3 Recherche de fichier (find)

`find` est une commande linux très puissante qui vous fera gagner beaucoup de temps. Elle permet de rechercher dans une arborescence de fichiers ceux qui correspondent à un critère donné (taille, droits, nom, dates...). Elle permet également d'appliquer une commande à chacun des fichiers ainsi trouvés.

Attention à ne pas la confondre avec la commande `grep` qui va examiner le contenu des fichiers.

Tutoriel 3 **Rechercher un fichier**

```
find ~ -name Ex.java
```

Recherche, chez vous (`~`), un fichier nommé `Ex.java`

Exercice 3

Trouvez avec la commande `find` tous les fichiers Java que vous avez déjà écrits.

Nous avons écrit pour vous une classe `Color` mais nous ne savons plus très bien où nous l'avons stockée. Nous nous rappelons juste l'avoir mise quelque part dans `/eCours`. Pouvez-vous la retrouver pour nous ?

1.4 Passer à la couleur

La classe `Color` que vous avez trouvée à l'exercice précédent offre quelques méthodes pour colorier des textes :

- ▷ `String toRed(String text)` qui colore le texte donné en rouge. C'est-à-dire que si on l'affiche à l'écran, il apparaitra en rouge.
- ▷ Idem pour `toGreen`, `toBlue`, `toYellow`, `toPurple` et `toCyan`.
- ▷ Exemple d'utilisation dans la classe `Color` :

```
System.out.println( toRed("Hello ") + toGreen("World !"));
```

Exercice 4

1. Copiez notre fichier `Color` chez vous.
2. Compilez-le.
3. Pour apprendre à l'utiliser, créez une méthode principale qui lit deux nombres au clavier et affiche la valeur maximale.
   Afin de rendre le programme plus agréable à utiliser, chaque lecture sera précédée d'un message en couleur indiquant ce qui est demandé (ex : "Entrez un nombre entier"). La réponse sera, elle aussi, affichée en couleur.

1.5 Les variables d'environnement

Une *variable d'environnement* est une variable associée à votre shell contenant un texte qui est accessible par toutes les applications que vous lancez. Généralement, elle permet de configurer certaines applications, d'en modifier le comportement.

Le prompt Le *prompt* (ou *invite* en français) est le texte qui apparait à gauche de ce que vous tapez dans votre shell. Il est déterminé par la variable d'environnement `PS1`.

FAQ Comment voir le contenu d'une variable d'environnement ?

Grâce à la commande : `echo $NOM_VARIABLE`

Puis-je changer son contenu ?

Oui, via la commande : `NOM_VARIABLE=nouvelle_valeur`

Exercice 5 Affichez la valeur de votre prompt. Vous remarquerez qu'il contient des codes qui seront remplacés par certaines valeurs. Par exemple, `\w` indique le dossier courant.

Exercice 6 Modifiez la valeur de votre variable `PS1`. Par exemple, modifiez l'invite en "Bonjour ! ".

Tutoriel 4 Expérience

✎ Déconnectez-vous de linux1, puis reconnectez-vous.

Surprise, votre prompt a repris sa valeur d'origine ! Pour rendre une modification permanente, il faut ajouter la commande à votre fichier `.bashrc`. C'est un fichier caché qui est exécuté par votre shell lors de votre connexion.

Exercice 7 Les possibilités de configuration du prompt sont nombreuses : afficher l'heure, le nom de la machine, votre login, utiliser des couleurs... Examinez la documentation pour configurer le prompt comme il vous plait. (`man bash`, section `PROMPT`)