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

2018 – 2019
DEV1

DEV1 – Laboratoires d'environnement

TD 07 – Filtres, tubes et redirections

Objectifs

Dans ce TD, vous étudierez les mécanismes de redirections, de tubes et les filtres sous Linux. Vous apprendrez à :

- ▷ rediriger les entrées et les sorties de vos programmes ;
- ▷ combiner plusieurs commandes Linux pour réaliser des opérations complexes sur vos fichiers ;

Table des matières

1 TD7 - Filtres, tubes et redirections **2**
1.1 Entrées et sorties standards 2
1.2 Rediriger la sortie 2
1.3 Rediriger l'entrée 3
1.4 Les tubes (pipes en anglais) 4
1.5 Rediriger les erreurs 4
1.6 Les filtres Linux 5

Version du document
4 novembre 2018

1 TD7 - Filtres, tubes et redirections

1.1 Entrées et sorties standards

Les fichiers standards

Tout programme qui s'exécute dispose de trois fichiers ouverts d'office par le système pour lui : l'entrée standard, la sortie standard et la sortie d'erreur, identifiés respectivement par les numéros 0, 1 et 2.
En Java, on retrouve ces trois fichiers :

- `System.in` pour 0 (entrée standard) qu'on retrouve dans la déclaration `Scanner clavier = new Scanner(System.in);`
- `System.out` pour 1 (sortie standard) qu'on retrouve dans l'instruction `System.out.println();`
- `System.err` pour 2 (erreur standard) qu'on retrouve dans l'instruction `System.err.println();`

Cela peut vous paraitre bizarre de dire que le clavier et l'écran sont des fichiers mais c'est bien ainsi que le programme les voit. Et c'est pratique, car nous allons pouvoir *rediriger* ces entrées et ces sorties vers de vrais fichiers de façon tout-à-fait transparente pour le programme ; il ne sera pas nécessaire de le modifier.

1.2 Rediriger la sortie

Il est possible, au moment où on lance un programme, de rediriger sa sortie. Tout ce que le programme enverra sur sa sortie standard (par exemple avec un `System.out.println()` en Java) ne sera pas visible à l'écran mais sera envoyé dans, par exemple, un fichier.

Notation

Une redirection de sortie standard se note «>»ou «1>»lors du lancement du programme. Ces redirections sont réalisées par le shell avant l'exécution de la commande et sont transparentes pour cette commande.

Exemple 1 **Redirection de sortie**

```
ls -l > liste
```

ou

```
ls -l 1> liste
```

Ces deux commandes sont équivalentes ; elles n'affichent pas le résultat à l'écran, mais l'écrivent dans le fichier `liste` créé ici ou écrasé s'il préexistait. Par défaut `ls -l` affiche le résultat à l'écran. Nous avons *redirigé* cette sortie standard vers un fichier, en l'occurrence `liste`.

Faites l'essai et vérifiez le contenu du fichier créé.

Exercice 1 **Rediriger la sortie**

- ▷ Écrivez un programme qui affiche la suite des pas croissants :
  1, 2, 4, 7, 11, 16, 22, 29, ...
  Exécutez-le pour afficher les 1000 premiers nombres de cette suite.
- ▷ Sauvez le résultat dans un fichier pour pouvoir l'examiner à votre aise.
  Rappel : pour examiner le contenu d'un fichier, inutile de passer par un éditeur, la commande `more` suffit.
- ▷ Est-ce que le nombre 15007 en fait partie ? (aide : vous vous rappelez de la commande `grep` ?)

Note

Avec la simple redirection en sortie, si le fichier existe déjà, il est écrasé. Nous pouvons choisir de ne pas l'écraser, mais de le compléter (ajouter du contenu à la fin du fichier) via la double redirection (en sortie) notée `>>`.

1.3 Rediriger l'entrée

L'«entrée standard»peut être associée à un fichier au lieu du clavier. Cela permet d'utiliser les données à partir du fichier au lieu de les entrer au clavier. C'est ce qu'on appelle une redirection de l'entrée.

Notation

La redirection d'entrée se note «<».

Exemple 2 **Redirection d'entrée**

```
commande < data
```

Les lectures de la commande se feront dans le fichier `data` et pas au clavier.

Tutoriel 1 **Expérience**

- ✎ Écrivez une classe `Multiples4` qui lit une série de nombres et n'affiche que ceux qui sont des multiples de 4. À la fin, elle affiche le nombre de multiples de 4.
- ✎ Exécutez votre programme et entrez des nombres au clavier. La combinaison de touches `Ctrl-d` est l'équivalent de la marque «fin de fichier»pour le clavier, c'est ainsi que vous terminerez l'acquisition de la série de nombres au clavier.
- ✎ Il ne faut pas confondre `Ctrl-d` et `Ctrl-c` qui tue le processus. Comment mettre en évidence la différence ?
- ✎ Exécutez la classe en associant le clavier à un petit fichier texte où vous aurez écrit les nombres au préalable, séparés par des blancs (caractères d'espacement).

Exercice 2 **Rediriger l'entrée**

Pour résoudre cet exercice, vous devez combiner votre programme des pas croissants et le programme qui sélectionne les multiples de 4. On vous demande d'afficher, parmi les 1000 premiers nombres de la suite des pas croissants, tous ceux qui sont des multiples de 4. Combien y en a-t-il ?

1.4 Les tubes (pipes en anglais)

Pour résoudre l'exercice de la section précédente, vous avez dû créer un fichier temporaire qui n'a servi qu'à ça. Vous avez redirigé la sortie de la première commande dans un fichier et redirigé ce fichier comme entrée de la seconde commande.

Les tubes

Le symbole «|»permet de chainer des commandes ; la sortie de l'une sert d'entrée à la suivante. On parle de "pipe" en anglais et de "tube" en français.

C'est une situation qui se présente souvent, surtout en Linux qui propose de nombreuses commandes qui font une seule chose (plutôt bien) et qu'on combine pour obtenir un résultat plus conséquent. Des commandes comme `more`, `grep`, `wc`... peuvent prendre leurs données sur l'entrée standard.

Exemple 3 **Enchaîner les commandes**

La commande `more` permet d'afficher un message page par page. Si on ne lui donne pas de nom de fichier, il pagine les données reçues sur l'entrée standard. On peut donc remplacer

```
ls /home >temp
more temp
```

par

```
ls /home | more
```

Exercice 3 **Utilisation des tubes**

- ▷ Utilisez un tube pour refaire l'exercice précédent (afficher parmi les 1000 premiers nombres de la suite des pas croissants, tous ceux qui sont des multiples de 4).
- ▷ Supprimez du programme `Multiples4` la ligne finale qui affiche le nombre de multiples trouvés.
- ▷ Relancez votre commande de l'exercice précédent. Vous ne voyez plus, à la fin, le nombre de multiples, ce qui est normal. Quelle enchainement de commandes permet d'afficher ce nombre (et uniquement ce nombre) ? Rappelez-vous, il existe une commande Linux qui "compte".
- ▷ Affichez, parmi les 1000 premiers nombres de la suite des pas croissants, tous ceux qui contiennent un 0.

1.5 Rediriger les erreurs

On a vu qu'il est possible de rediriger la sortie d'un programme. Il est aussi possible de rediriger ses erreurs.

Attention ! C'est au programme à décider ce qui est un message d'erreur (en utilisant `err` au lieu de `out`).

Notation

Une redirection de la sortie d'erreur se note «2>».

Exemple 4

Redirection d'erreurs

Supposons que vous ayez écrit une classe `Malfaite` qui provoque beaucoup d'erreurs à la compilation. Pour les regarder à votre aise, vous pouvez les rediriger dans un fichier `erreur` via :

```
javac MalFaite.java 2>erreur
```

Vous pouvez alors examiner les erreurs en ouvrant le fichier `erreur`, par exemple via :

```
more erreur
```

Exercice 4

Redirection

Les professeurs se demandent combien d'étudiants ont chez eux un fichier `Multiples4.java`. Pouvez-vous indiquer la (suite de) commande(s) qui permet de répondre à la question ?

1.6 Les filtres Linux

De nombreuses commandes Linux sont basées sur le principe `KISS` (Keep It Simple, Stupid). Elles font peu, mais le font bien et surtout, peuvent facilement coopérer pour, au final, obtenir un résultat bluffant. Parmi toutes les commandes, les filtres sont à mettre en évidence.

Les filtres

Un `filtre` est une commande Linux qui acquiert des données sur l'entrée standard et les envoie vers la sortie standard après les avoir éventuellement transformées.

Nous allons nous concentrer sur les filtres suivants : `tr`, `cut`, `cat`, `sort`, `head`, `tail`, `split`, `uniq`, `grep`, `more`, `less`, `wc`, ... Vous en connaissez déjà ; voyez-en les pages de manuel respectives et l'aide (`--help`) pour en connaitre le détail.

Tutoriel 2

Utilisation des filtres

On voudrait connaitre le nombre d'utilisateurs qui sont connectés à linux1 pour le moment. Voyons, étape par étape, comment on peut obtenir le résultat.

- La première étape est de penser à la commande `who` qui affiche toutes les connexions actives.
- C'est gagné, il suffit de compter le nombre de lignes, direz-vous ! Allons ! Ne nous arrêtons pas là ; l'ordinateur peut compter pour nous, ce qui donne :

  ```
  who | wc -l
  ```

- Cette fois, ça y est, on a la réponse ! Ben, non ! Il y a un piège car la commande `who` ne donne pas les utilisateurs mais les connexions ce qui n'est pas tout à fait la même chose ; un utilisateur peut avoir ouvert plusieurs fenêtres "putty".

  La commande `uniq` peut venir à notre secours en supprimant les doublons mais il faut que les lignes soient parfaitement identiques et contigües.

  Parfait ! La commande `cut` peut ne garder que certaines colonnes et la commande `sort` peut trier les lignes. On obtient alors :

  ```
  who | cut -f 1 -d ' ' | sort | uniq | wc -l
  ```

  Je ne me réjouis pas trop vite ; je suis sûr que vous allez encore trouver une faille, n'est-ce pas ? Non, pas cette-fois ; on y est !

Exercice 5 Nombre de connexions d'un utilisateur

Trouvez un enchainement de commandes qui permet de donner le nombre de connexions d'un utilisateur donné.

Il existe de nombreuses façons de résoudre cet exercice. Celle à laquelle nous pensons fait intervenir : `grep`, `wc` et `who`.

Exercice 6 Nombre de PC connectés

Trouvez un enchainement de commandes qui permet de donner le nombre de PC connectés à linux1. Ce n'est pas exactement le nombre d'utilisateurs car un utilisateur pourrait être connecté sur plusieurs machines.

À nouveau, il existe de nombreuses façons de résoudre cet exercice. Celle à laquelle nous pensons fait intervenir la commande `tr -s ' '` qui supprime plusieurs occurences consécutives d'un même caractère facilitant ainsi la sélection par colonne de la commande `cut`.

Exercice 7 Droits sur les dossiers personnels

Trouvez un enchainement de commandes qui permet de donner le nombre de professeurs qui ont donné le droit à ceux qui ne font pas partie de leur groupe d'entrer dans leur dossier personnel.

À votre imagination...