

DEV1 – ENVL – Laboratoires d'environnement**TD 5 – Linux (partie III)****Table des matières**

1	Grep - Recherche dans un fichier	2
2	Les filtres	2
2.1	Présentation	2
2.2	Nourrir les filtres	3
2.3	Trier les lignes	3
2.4	Enlever les doublons	3
2.5	Les premières/dernières lignes	4
2.6	Transformer des caractères	4
2.7	Ne garder que certaines colonnes	4
2.8	Compter	4
2.9	Résumé	5
3	Conclusion	5

Dans votre répertoire `~/dev1`, créez un répertoire `td6`. Il contiendra tous les fichiers que vous allez créer aujourd'hui.

1 Grep - Recherche dans un fichier

On a déjà vu la commande `find` qui permet de rechercher un fichier en fonction de critères. La commande `grep` permet de faire une recherche **à l'intérieur** d'un fichier.

Grep

```
grep motif fichiers...
```

Dans son utilisation la plus simple, permet d'extraire de fichiers toutes les lignes qui contiennent un certain texte (appelé motif).

Penser aux *jokers*

Comme à chaque fois qu'une commande peut recevoir plusieurs noms de fichiers, on peut les indiquer explicitement et/ou utiliser les *jokers* pour en désigner plusieurs d'un coup.

Exemple 1

Où ai-je déjà utiliser `Scanner` ?

La commande `grep Scanner *.java` montre toutes les lignes contenant le mot `Scanner` dans tous les programmes Java du dossier courant.

Consultez la page de manuel pour voir comment paramétrer la commande.

2 Les filtres

2.1 Présentation

La commande `grep` peut rechercher dans un fichier donné en paramètre mais aussi rechercher dans du texte donné en entrée.

Exemple 2

`grep` comme un filtre

```
cat Hello.java | grep Scanner
```

Avec cette écriture, le contenu du fichier `Hello.java` est envoyé à la commande `grep` qui n'affichera que les lignes contenant le mot `Scanner`.

On peut voir la commande comme un *filtre*.

Filtres - Définition

Les filtres sont des commandes qui lisent sur l'entrée standard et affichent sur la sortie standard une version réduite/modifiée de ce qui a été lu.

Il y a beaucoup d'autres commandes qui agissent comme des filtres. C'est à la base même de Linux : fournir des commandes qui font peu mais qui le font bien et les combiner (avec des tubes) pour obtenir un résultat conséquent.

Exemple 3

Combiner les filtres

```
cat Hello.java | grep Scanner | grep System
```

Produit sur la sortie standard toutes les lignes du fichier `Hello.java` qui contiennent à la fois les mots `Scanner` et `System`.

2.2 Nourrir les filtres

Les filtres traitent les données reçues. Quelles peuvent être ces données ? N'importe quoi. Nous avons vu que ça peut être le contenu d'un fichier mais ça peut aussi être le résultat d'une commande quelconque. Afin d'enrichir nos exemples, voyons 2 commandes qui produisent des données.

Commandes pour nourrir les filtres

- ▷ `du nomDossier` : donne l'espace disque occupé par chaque sous-dossier du dossier indiqué.
- ▷ `who` : donne la liste des connexions à la machine.

2.3 Trier les lignes

La commande `sort` trie les lignes reçues.

Experience 1

Trier les lignes

Expérimentons la commande `sort`.

- ✍ Placez-vous dans votre dossier `dev1`.
- ✍ `du .` affiche les dossiers et leur taille
- ✍ `du . | sort -n` affiche la même liste mais triée (numériquement sur la taille).

Consultez la page de manuel pour toutes les options.

Exercice 1

Trier le résultat de du

Reprenez l'exemple donné ci-dessus et voyez comment trier la liste des sous-dossiers et leur taille :

1. En ordre inverse de la taille (le plus gros d'abord).
2. Sur le nom du dossier.

2.4 Enlever les doublons

La commande `uniq` ne laisse passer qu'une seule fois les lignes identiques **qui se suivent**.

Experience 2

Enlever les doublons

TODO

Exercice 2

Les utilisations de Scanner

Affichez toutes les lignes **différentes** contenant le mot `Scanner` dans tous vos fichiers Java du dossier `td4`.

Aide : Si vous devez enchaîner plusieurs filtres, vous pouvez tester, étape par étape.

2.5 Les premières/dernières lignes

La commande `head` (respectivement `tail` ne laisse passer que les premières (respectivement dernières) lignes reçues.

Experience 3

Les premières/dernières lignes

TODO

Exercice 3

Les gros dossiers

Affichez le nom et la taille des 3 dossiers qui prennent le plus d'espace disque parmi tous vos sous-dossiers de `dev1`.

2.6 Transformer des caractères

La commande `tr` permet de faire des transformation du texte : remplacez certains caractères par d'autres ou simplifier plusieurs occurrences consécutives d'un caractère

Experience 4

Transformer des caractères

TODO

Elle est particulièrement utile pour la commande qui suit.

2.7 Ne garder que certaines colonnes

La commande `cut` sert à ne garder que certaines colonnes parmi les lignes reçues.

Experience 5

Les premières/dernières lignes

TODO

Exercice 4

Les gros dossiers

Affichez le nom et la taille des 3 dossiers qui prennent le plus d'espace disque parmi tous vos sous-dossiers de `dev1`.

2.8 Compter

La commande `wc` est un peu particulière. Elle compte le nombre de lignes/mots/caractères reçus.

TODO

2.9 Résumé

Filtres

Voici les quelques filtres que nous avons appris à utiliser.

- ▷ **grep** : ne laisse passer que certaines lignes.
- ▷ **sort** : trie les lignes
- ▷ **uniq** : enlève les lignes en double
- ▷ **head** : ne laisse passer que les premières lignes
- ▷ **tail** : ne laisse passer que les dernières lignes
- ▷ **cut** : ne laisse passer que certaines colonnes
- ▷ **tr** : modifie certains (groupes de) caractères
- ▷ **wc** : compte les lignes, les mots et les caractères

3 Conclusion

Notions importantes de ce TD

Voici les notions importantes que vous devez avoir assimilées à la fin de ce TD.

- ▷ Comprendre le concept de filtre.
- ▷ Savoir utiliser et enchaîner les bons filtres pour répondre aux questions qu'on se pose.

Pour aller plus loin...

Ceux qui ont du temps peuvent aborder les exercices suivants. (expressions régulières dans le grep)