

Substitution et traitement de flots sous Unix

TP de LIN 3
2003-2004

Objectif

Etude des commandes `awk`, `cut`, `egrep`, `sed`, `sort` et `tr`.

Remarque : dans le manuel en ligne de la commande `sort`, il faut lire "Un argument `+pos` ou `-pos` peut également avoir un SUFFIXE" et non "un préfixe" !

La désignation des noms de fichiers sous Unix supporte le mécanisme de substitution (pattern-matching en anglais), qui repose essentiellement sur l'utilisation de 2 caractères spéciaux :

- `?` qui remplace 0 ou 1 occurrence de n'importe quel caractère
- `*` qui remplace n'importe quelle chaîne de caractères (y compris la chaîne vide)

Ainsi le modèle (pattern) `"*.c"` désigne tout nom de fichier se terminant par la sous-chaîne `".c"` et `"to*er?.?.?t"` peut désigner `"toutlefichier.txt"` comme `"touslesfichiers.dat"` ou encore `"touterien.t"`.

Questions

Recherche avec la commande `find`

1. Recopier les fichiers `main.c`, `moy.c` et `moy.h` qui se trouvent sous `~belloir/public/lin3/tp2` dans votre répertoire de travail et rechercher à partir de votre racine tous les fichiers contenant du code en langage C faisant référence à la fonction "moyenne". Le code C est généralement contenu dans des fichiers dont les extensions sont `.h` et `.c`
2. Rechercher à partir de votre racine tous les fichiers en langage C qui incluent un fichier entête. Les fichiers entête ont généralement l'extension `.h`

Recherche avec les commandes `cut` et `sort`

Les tubes (pipe) Unix permettent de connecter le flot de sortie d'une commande au flot d'entrée d'une autre. Ainsi on peut enchaîner plusieurs commandes sur un flot initial afin de transformer ce flot par étapes successives. Exemple : `head -3 notes.txt | tail -2` permet d'extraire les 2e et 3e lignes du fichier `notes.txt` A partir du fichier `notes.txt` qui se trouvent sous `~belloir/public/lin3/tp2`, écrivez les traitements suivants **en utilisant les commandes `cut` et `sort`** :

1. Afficher la liste des étudiants de la licence informatique triée par ordre alphabétique des nom et prénom.
2. Afficher les résultats de LIN2 par ordre croissant des notes. En cas d'égalité, les étudiants sont rangés par ordre alphabétique des nom et prénom. Remarque : l'option `-n` (comparaison numérique) n'a pas de sens pour les champs Nom et Prénom mais cela est sans conséquence puisque, si elle ne parvient pas à convertir une chaîne sous format numérique, la commande `sort` utilise le tri classique.
3. Même question mais classement par ordre décroissant des notes en premier.
Remarque : il est possible de placer plusieurs commandes Unix sur une même ligne, ce qui a pour avantage d'éviter que le flot de sortie ne soit coupé par le prompt et les commandes suivantes. Pour cela il faut séparer les différentes commandes par ; Exemple : `head -1 notes.txt;cat notes.txt` permet de dupliquer la 1ère ligne du fichier `notes.txt`
4. Même question avec LIN3. Les étudiants qui ne se sont pas présentés à l'examen sont listés avec la mention "abs" et classés à la fin.

La commandes *sed*

1. Filtrer le fichier notes.txt afin de remplacer toutes les occurrences de la chaîne "abs" par la note 0. La commande sed admet les expressions régulières afin de désigner les modèles de chaînes. Pour comprendre ce qu'est une expression régulière et comment l'utiliser consulter le manuel de la commande egrep.
2. Corriger la question précédente afin de ne remplacer "abs" que lorsqu'elle apparaît dans une colonne de notes.
3. Même question que la question 5 mais avec LIN2 et LIN6, en classant les étudiants par ordre croissant sur LIN2 et, en cas d'égalité, par ordre croissant sur LIN6 et, en cas d'égalité, par ordre alphabétique des nom et prénom. La mention "abs" revient à considérer la note comme étant 0 pour le classement.
4. Filtrer le fichier notes.txt afin de remplacer toutes les séquences de blancs séparant les différents champs par une tabulation.
5. Filtrer le fichier notes.txt afin qu'il ne subsiste qu'un seul blanc entre les 2 premières colonnes puis une tabulation entre les colonnes suivantes.
6. Afficher le fichier notes.txt en affichant le prénom puis le nom séparés par un blanc, puis le reste de la ligne.
7. Idem que la question 10 mais dans le cas où "abs" remplace une note, faire précéder "abs" de 3 blancs et changer "abs" par "absent".

La commandes *awk*

1. jusqu'à 20) Reprendre les questions 7 à 13 en utilisant la commande awk.