SVM Mac 156
Rubrique Pratique
Terminal
1 page

Le format texte, par opposition à binaire, est LE format communément adopté par une énorme majorité de programmes qui évoluent dans le monde Unix.
Yannick Cadin

Afficher toutes les livraisons en les numérotant?
cat -n livraisons.txt
1 CAT 7800 1730-09
2 UA 4504.2 1731-09
3 CA 6522 1735-09
4 IBE 10610.05 1740-09
5 CA 3585 1742-10
6 CAT 5000 1743-10
Notre fichier contient le détail de 6 livraisons (numérotées par la commande cat au début de chaque ligne). Nos livraisons se composent d'un code client, d'un montant et d'un numéro de bordereau. L'usage le plus répandu de la commande cat est de con**cat**éner le contenu d'un unique fichier vers la sortie standard, en l'occurrence la fenêtre Terminal par défaut.

Présenter les 2 plus importantes livraisons ?
sort -n -r +1 livraisons.txt | head -2
Nous devons combiner ici deux commandes, sort va trier les lignes du fichier livraisons.txt selon les critères formulés : -n pour considérer les valeurs manipulées comme des nombres et non du texte, -r pour inverser l'ordre d'affichage du résultat et +1 (le nombre un) pour désigner le deuxième champ comme clé de tri (la numération des champs commence à 0 pour sort, ceci est expliqué dans sa documentation électronique : man sort). Le résultat devrait apparaître à l'écran (sortie standard) mais nous redirigeons ce flux via un tube de communication (symbolisé par la barre verticale entre deux programmes) vers la commande head (tête) qui limitera l'affichage aux deux premières lignes (option -2).

Présenter uniquement la colonne montants des 2 plus petites livraisons ?
cut -d ' ' -f 2 livraisons.txt | sort -nr | tail -2
Ici encore nous sommes contraints de combiner plusieurs commandes reliées par ces pipes (le terme anglais pour tube de communication). Le programme cut peut (entre plein d'autres choses) extraire des colonnes d'un fichier texte. Dans le cas présent cut nous découpera la deuxième colonne (option -f pour field, un champ). L'option -d précise que le séparateur de champs est l'espace et non pas la tabulation comme c'est la cas en standard. La colonne, en fait les lignes d'origine privées de leurs champs code client et bordereau de livraison sont transmises à sort qui les tries à nouveau pour finalement communiquer son résultat à tail (queue) qui n'en montrera que les deux dernières.

Compter le nombre total de livraisons enregistrées pour le client China Airlines ?
grep -w CA livraisons.txt | wc -l
Nous avons déjà vu l'usage de la commande grep dans un précédent numéro, elle sert à chercher une expression, le plus souvent une séquence de caractères au sein de fichiers. Ici on

ne veut que les lignes contenant le mot CA (option -w pour ne pas trouver aussi les lignes mentionnant Cathay, code client CAT). La modeste commande wc se contente quant à elle de compter (Word Count) les lettres, les mots et les lignes… Sauf si l'on précise ce que l'on veut dénombrer, ici uniquement des lignes (celles obtenues par grep), d'où l'option -l.

Donner le total de livraisons enregistrées pour chaque compagnie ?
cut -d ' ' -f 1 livraisons.txt | sort | uniq -c
À l'aide de cut, on ne garde que les codes représentant.le client pour chaque livraison effectuée (le premier champ car contrairement à sort, cut les numérote à partir de 1). Nous trions par ordre alphabétique la liste obtenue puis uniq en supprimera tous les doubles en comptabilisant toutefois leur nombre pour chaque élément.

Mettre en forme : les détails de deux livraisons par ligne et un alignement des colonnes ?
paste - - < livraisons.txt | column -t
Les commandes telles que paste et column, il en existe beaucoup d'autres, servent essentiellement à produire des présentations un peu plus lisibles que celles obtenues brutes à partir d'un fichier texte où les espaces tiennent lieu de séparateurs de champs. La commande paste va lire deux lignes consécutives du fichier livraisons.txt pour les agréger (ce que l'on exprime en lui donnant deux tirets comme options et en allant lire le flux de données depuis notre fichier à l'aide de la redirection de l'entrée standard symbolisée par un signe inférieur). Notre dernière commande, column, se chargera de terminer la mise en page en alignant les valeurs sur des colonnes tirées au cordeau.

Lister les bordereaux avec les noms des clients plutôt qu'avec leurs codes ?
sort livraisons.txt | join -o '2.3 1.2' clients.txt -
Presque aussi fort qu'un FileMaker ou un MySQL, du quasi relationnel ou comment mettre en relation (joindre) deux fichiers qui en commun des clefs, ici notre code client.
La commande join qui réalise ce petit exploit ne peut fonctionner correctement que si ses deux sources sont minutieusement triées. On s'assurera l'avoir fait à chaque actualisation de notre fichier clients.txt, par contre nous le faisons en instantané pour livraisons.txt car ce dernier évolue en permanence. Par défaut, join prend la première valeur de chaque fichier et la considère comme la clef permettant de réunir les lignes des deux sources. L'option -o permet de définir très précisément quelles valeurs des lignes mises en relation on veut présenter et dans quel ordre. Ici 2.3 signifie dans le fichier 2 (livraisons.txt) le troisième champ, c'est à dire le bordereau (join commence sa numération des champs à 1). L'autre expression, 1.2, de la même façon signifie dans le fichier 1 (clients.txt), piocher le deuxième champ (le nom de la compagnie) comme la constitution du fichier ci-dessous le met en évidence.
Notre fichier clients.txt ressemble à cela (remarquer que les entrées sont triées) :
CA China
CAT Cathay
IBE Iberia
UA United
Un point très important à relever et qui n'est pas un cas isolé propre à join, beaucoup de commandes Unix interprètent un tiret seul comme étant un fichier dont les données sont lues depuis l'entrée standard. Ci-dessus, ça nous permet d'aller chercher les données du deuxième fichier à traiter dans le tube de communication en provenance du résultat de la commande sort.