

SYSTÈMES

Contrôle de TP - groupe H1

date : 21 novembre 2012, 10h15

durée : 1 heure 55 (login déduits)

documents personnels manuscrits autorisés et réalisations de TP.

1) Arborescence à créer

Créez un répertoire appelé `tpnote` pour le TP noté, vous travaillerez dedans et n'y mettez que les fichiers demandés au cours du TP.

Créez un sous-répertoire `tpnote/partie1` et un sous-répertoire `tpnote/partie2`.

2) Partie 1 : filtres sur un fichier de données

Recopiez le fichier `/usr/local/anonymous/SYS/TPNOTE/ngc2000.txt` dans votre dossier `tpnote/partie1`.

Ce fichier contient des informations sur des objets célestes tels que les galaxies, nébuleuses et amas d'étoiles : un par ligne, les champs sont séparés par un `|`. Il s'agit d'un véritable fichier provenant d'une base de données astronomique (Vizier). En fait, il y a deux catalogues mélangés : le catalogue NGC et le catalogue IC. Dedans, les objets sont simplement numérotés : par exemple NGC7000 ou IC410. Les lignes commencent par le nom de l'objet : c'est juste un numéro quand c'est un objet du catalogue NGC tandis que le nom commence par un `I` quand c'est un objet du catalogue IC.

Ensuite, on trouve le type de l'objet, ex : `Gx` signifie galaxie, `Gb` galaxie barrée, `Nb` nébuleuse diffuse, `Pl` nébuleuse planétaire,... Parfois, il n'y a pas de type. On ne vous demande aucune connaissance sur ce sujet.

Ensuite, on trouve deux champs qui indiquent les coordonnées de l'objet : `RA` et `DE`.

Ensuite, un code indiquant la source (qui a fait la découverte), peu importe la signification. Ensuite, on trouve le code de la constellation (`And` = Andromède, mais peu importe).

Ensuite, une colonne vide qui ne sert à rien (c'est un format de fichier assez général).

La colonne suivante, `size`, donne la dimension apparente en minutes d'arc (mais peu importe). La colonne `mag` indique la classe de brillance (`> 6` = invisible à l'œil nu) appelée magnitude. Et pour finir, une description abrégée de ce qu'on voit au télescope.

Attention, les premières et les dernières lignes du fichier sont les titres, vous avez le droit de les enlever manuellement pour la suite (créer un second fichier ne contenant pas les lignes d'entête et de fin).

On demande de taper les commandes dans un fichier appelé **reponses.txt**, en écrivant clairement : « a) commande à faire » « b) commande à faire » et ainsi de suite, une réponse par ligne, une ligne vide entre chaque réponse.

- a) Compter le nombre de types d'objets (différents) qui sont mentionnés dans ce fichier.
- b) Compter le nombre d'objets dont la magnitude est supérieure ou égale à 17.
- c) Afficher (en un seul exemplaire chacun) les noms des constellations qui ne finissent par aucune voyelle (`y` est une voyelle). Exemple : `Oph`.
- d) Afficher les objets du catalogue IC (leur nom commence par `I`) classés par ordre de numéro (`I 1` puis `I 2`, etc).
- e) Compter le nombre d'objets qui sont au même endroit dans le fichier : ils ont exactement les mêmes valeurs pour les coordonnées `AD` et `DE` (champs 3 et 4).
- f) Afficher l'objet le plus brillant (valeur `mag`, champ 9, la plus petite) de la constellation `Vul` (`Vulpecula` = petit renard).
- g) Afficher les lignes concernant les objets de type `Pl` dont la valeur `AD` est comprise entre 12 inclus et 16 exclus et la valeur `DE` positive ou nulle.

3) Partie 2 : script bash

Placez-vous maintenant dans le répertoire `tpnote/partie2`. Le but du TP est de programmer plusieurs scripts permettant de gérer un budget. On voudrait gérer une liste de dépenses (sortie d'argent) et de recettes (entrées

d'argent). Toutes les opérations sont inscrites dans un fichier : type d'opération, montant et libellé. Il y a deux scripts : dépense et recette. Chacun reçoit un premier paramètre qui est le montant (en valeur absolue) de la dépense ou de la recette, ensuite, il y a un libellé (soit un seul paramètre, soit toute la suite, si vous savez faire).

Par exemple, on fait (valeurs fictives, en espérant que ça ne vous touche pas personnellement) :

```
prompt% dépense 15 paintball
```

```
Ok, il vous reste 142 euros.
```

```
prompt% recette 23 vente jeu occasion
```

```
Ok, il vous reste 165 euros.
```

```
prompt% dépense 7 tickets RU
```

```
Ok, il vous reste 158 euros.
```

```
prompt%
```

Ces deux scripts inscrivent l'opération à la fin d'un fichier appelé budget, par exemple – vous choisirez le format vous-même. Ensuite, elles recomptent toutes les sommes afin de déterminer le total restant. Pour cela, il faut procéder comme dans le TP8 et le fichier meteo, et ici, additionner ou soustraire les montants du fichier budget.

Ceux qui auront fini pourront s'attaquer à d'autres commandes ou des variantes, par exemple :

```
prompt% annuler
```

```
J'ai annulé la dépense 7 tickets RU
```

```
Ok, il vous reste 165 euros.
```

```
prompt% dépense max
```

```
La plus grande dépense a été : 15 paintball
```

```
prompt% recette max
```

```
La plus grande recette a été : 23 vente jeu occasion
```

```
prompt%
```

Votre script devra vérifier la présence des paramètres, et leur nature et signaler toute erreur. Il veillera également à ce que des fichiers ne soient pas écrasés par la commande.

La note sera meilleure si les scripts sont indentés, commentés, robustes : résistent à des erreurs telles que des mauvais paramètres, des fichiers manquants...

Faites ce que vous pouvez. La notation commence dès la première ligne écrite. Ne vous bloquez pas sur une question, passez à la suivante.

4) Rendre votre travail

Quand vous avez fini, mettez-vous dans votre répertoire tpnote qui contient partie1 et partie2 et lancer la commande `/usr/local/anonymous/SYS/jaifini h1` (en mettant votre propre groupe en paramètre).

Il s'agit d'un script qui recopie tout ce que vous avez dans le répertoire courant quelque part pour permettre la correction.