

Mise à niveau Unix

TP 5

Juliusz Chroboczek

22 Septembre 2010

Scripts shell

Exercice 1. Écrivez un script shell qui supprime du répertoire courant les fichiers dont le nom se termine par « ~ », ceux qui s'appellent « core » et ceux dont le nom se termine par « .o ».

Exercice 2. Écrivez un script shell qui supprime du répertoire donné en paramètre de ligne de commande les fichiers dont le nom se termine par « ~ », ceux qui s'appellent « core » et ceux dont le nom se termine par « .o ». Votre script fonctionne-t-il sur les répertoires dont le nom contient un espace ?

Tubes

Exercice 3. En une seule ligne de shell, déterminez le nombre de fichiers qui se trouvent dans le répertoire qui vous a servi à faire le TP précédent.

Exercice 4. En une seule ligne de shell, déterminez le nombre d'utilisateurs connectés sur la machine où vous vous trouvez.

Exercice 5. Affichez la liste des utilisateurs qui se trouvent sur la machine où vous êtes et rien d'autre — pas de nom de terminal, etc. (*Indication : vous pourrez utiliser la commande cut.*)

Ruptures de boucle

Exercice 6. Écrivez une version plus efficace du programme premier.c du TP précédent qui arrête l'itération lorsqu'un diviseur a été trouvé.

Exercice 7. Écrivez un programme qui lit un entier n et affiche s'il s'agit du cube d'un nombre entier.

Exercice 8.

1. La moyenne harmonique H d'une famille de nombres $(a_i)_{1 \leq i \leq n}$ est définie par

$$H = 0 \quad \text{s'il existe } k \text{ tel que } a_k = 0$$
$$H = n / \sum_{1 \leq i \leq n} \frac{1}{a_i} \quad \text{sinon}$$

Écrivez un programme qui lit un entier n , puis qui lit n nombres à virgule flottante et affiche leur moyenne harmonique. (Il faudra lire n nombres même si l'un d'entre eux est nul.)

2. La moyenne logarithmique L d'une famille de nombres $(a_i)_{1 \leq i \leq n}$ strictement positifs est définie par

$$L = e^{(\sum_{1 \leq i \leq n} \log a_i) / n}$$

Écrivez un programme qui lit un entier n , puis qui lit n nombres strictement positifs et affiche leur moyenne logarithmique. Votre programme devra afficher un message d'erreur compréhensible et s'arrêter dès lors qu'il aura lu un nombre négatif ou nul.

Exercice 9.

1. Écrivez un programme qui affiche la table de multiplication :

1	2	3	4	5	...	10
2	4	6	8	10	...	20
⋮	⋮	⋮	⋮	⋮	⋮	⋮
10	20	30	40	50	...	100

Pour aligner les valeurs, vous pourrez vous servir du descripteur « %4d » avec `printf`.

2. Écrivez un programme qui lit un entier n puis qui affiche la table de multiplication en s'arrêtant dès qu'il rencontre n .

10
1 2 3 4 5 ... 10
2 4 6 8 10

3. Écrivez un programme qui détermine si un entier est contenu dans la table de multiplication.