

Mise à niveau Unix

TP 4

Juliusz Chroboczek

22 septembre 2014

Comme d'habitude, créez un répertoire dans lequel vous travaillerez durant ce TP.

Redirections et tubes

Exercice 1.

1. Créez un fichier `liste.text` qui contient la liste des fichiers contenus dans `/usr/bin`.
2. Sans exécuter la commande `ls` ou utiliser une expansion de motifs, affichez les noms des fichiers contenus dans `/usr/bin` dont le nom se termine par `s`. (Indication : `grep` est votre ami, même s'il porte un nom bizarre.)

Exercice 2.

1. En une seule ligne de shell, déterminez le nombre de fichiers qui se trouvent dans le répertoire `/bin`. (Indication : utilisez la commande `wc`.)
2. En une seule ligne de shell, déterminez le nombre d'utilisateurs connectés sur la machine où vous vous trouvez.
3. 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`.)

Premières boucles

Exercice 3. Écrivez un programme `vertical.c` qui demande à l'utilisateur un entier n et affiche une colonne de « * » de longueur n .

Exercice 4. Écrivez un programme `horizontal.c` qui demande à l'utilisateur un entier n et affiche une ligne de « * » de longueur n .

Exercice 5. Écrivez un programme `carres.c` qui affiche les carrés des 10 premiers nombres naturels, c'est-à-dire la suite d'entiers 1, 4, 9 ... 100.

Exercice 6. Écrivez un programme `multiples.c` qui demande à l'utilisateur un entier n puis affiche les 10 premiers multiples de n . Par exemple, si l'utilisateur entre 7, votre programme devra afficher la suite 7, 14, 21 ... 70.

Boucles avec accumulation

Exercice 7. Écrivez un programme `somme-cubes.c` qui demande à l'utilisateur un entier n puis qui affiche la somme des cubes des n premiers nombres entiers. Par exemple, si l'utilisateur entre 5, votre programme devra afficher 225, car

$$\begin{aligned}\sum_{k=1}^5 k^3 &= 1^3 + 2^3 + 3^3 + 4^3 + 5^3 \\ &= 1 + 8 + 27 + 64 + 125 \\ &= 225\end{aligned}$$

Exercice 8. Écrivez un programme `somme.c` qui demande à l'utilisateur un entier n , puis qui lit n entiers et affiche leur somme et leur moyenne. (Attention — pensez au type des données.)

Boucles avec *flags*

Exercice 9. Écrivez un programme `sept.c` qui demande à l'utilisateur un entier n puis qui lit n entiers et indique à l'utilisateur si le nombre 7 se trouvait parmi ces n entiers.

Exercice 10. Écrivez un programme `premier.c` qui demande à l'utilisateur un entier n puis qui indique à l'utilisateur si cet entier est premier.

Boucles imbriquées.

Exercice 11. Écrivez un programme `carre1.c` qui demande à l'utilisateur un entier n et affiche un carré plein de « * » de côté n .

```
*****
*****
*****
*****
*****
```

Exercice 12. Écrivez un programme `carre2.c` qui demande à l'utilisateur un entier n et affiche un carré creux de « * » de côté n .

```
*****
*   *
*   *
*   *
*****
```

Exercice 13. Écrivez un programme `triangle1.c` qui demande à l'utilisateur un entier n et affiche un triangle plein de « * » de largeur n avec la pointe en haut à gauche.

```
*
**
***
****
*****
```

Exercice 14. Écrivez un programme `triangle2.c` qui demande à l'utilisateur un entier n et affiche un triangle plein de « * » de largeur n avec la pointe en bas à droite.

```
*****
****
***
**
*
```