

# Scripts shell : Défi

# Table des matières

<b>I - Exercice : Défi</b>	<b>3</b>
<b>II - Exercice : Auto-évaluation</b>	<b>4</b>
<b>Solutions des exercices</b>	<b>5</b>

## Exercice : Défi

---



Il va falloir réaliser un script Bash complet qui réalise la tâche suivante : il va écrire le message "L'utilisateur NOM\_UTILISATEUR a comme UID UID\_UTILISATEUR", pour chaque utilisateur existant sur la machine. NOM\_UTILISATEUR et UID\_UTILISATEUR seront bien entendu remplacé par le nom de l'utilisateur et son UID.

Le message en question sera écrit dans l'un des 3 fichiers différents :

- `users.txt` si l'UID est compris entre 1000 et 1999 (inclus)
- `system_users.txt` sur l'UI est strictement inférieur à 1000
- `other_users.txt` pour les UID supérieurs ou égaux à 2000

### Question

[solution n°1 p. 5]

Écrivez le script permettant de réaliser cela.

#### Indice :

On peut trouver la liste des utilisateurs et leur UID dans le fichier `/etc/passwd`.

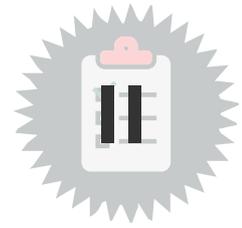
#### Indice :

La commande `cut` est très utile pour extraire des informations depuis du texte ayant un format bien défini.

# Exercice : Auto-évaluation

---

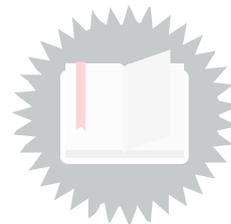
[solution n°2 p. 5]



Auto-évaluez votre capacité à répondre correctement aux questions de ce défi (sur une échelle de 1 à 4) :

1. Je ne sais répondre à aucune question (ou presque)
2. Je sais répondre à moins de la moitié des questions
3. Je sais répondre à plus de la moitié des questions
4. Je sais répondre à toutes les questions (ou presque)

# Solutions des exercices



## [exercice p. 3] Solution n°1

On crée et rend exécutable un fichier de `fi.sh` qui va contenir notre script.

```
1 #!/bin/bash
2
3 for entry in $(cat /etc/passwd | cut -d":" -f1,3)
4 do
5     user=$(echo "$entry" | cut -d":" -f1)
6     user_id=$(echo "$entry" | cut -d":" -f2)
7     if [ $user_id -ge 1000 ] && [ $user_id -lt 2000 ]
8     then
9         echo "L'utilisateur $user a comme UID $user_id" >> users.txt
10    elif [ $user_id -lt 1000 ]
11    then
12        echo "L'utilisateur $user a comme UID $user_id" >> system_users.txt
13    else
14        echo "L'utilisateur $user a comme UID $user_id" >> other_users.txt
15    fi
16 done
```

Il y a différentes manières de faire, mais celle-ci est plutôt une bonne proposition. On commence par faire une boucle `for` sur toutes les entrées du fichier `/etc/passwd`, dont l'on extrait uniquement le nom d'utilisateur et l'UID grâce à `cut`.

Pour chaque entrée on récupère le nom d'utilisateur et son UID dans deux variables séparées, toujours grâce à `cut`.

On applique ensuite une condition sur l'UID :

- si l'UID est compris entre 1000 (inclus) et 2000 (exclus) on écrit le message à l'aide d'une redirection dans le fichier `users.txt`
- si l'UID est plus petit que 1000 on écrit le message à l'aide d'une redirection dans le fichier `system_users.txt`
- sinon on écrit dans `other_users.txt`

On prends garde à écrire à l'aide de redirections `>>`, et non pas `>`, pour ne pas écraser le contenu des fichiers à chaque écriture.

## Solution n°2

[exercice p. 4]

Auto-évaluez votre capacité à répondre correctement aux questions de ce défi (sur une échelle de 1 à 4) :

1. Je ne sais répondre à aucune question (ou presque)
2. Je sais répondre à moins de la moitié des questions
3. Je sais répondre à plus de la moitié des questions
4. Je sais répondre à toutes les questions (ou presque)

