Leçon 2 : Scripting sous Linux

Dans la version CentOS 7, la version de *Python* installée est la version 2.x. Il faut, dès lors, commencer par installer Python 3. Pour ce faire, il faut entrer la commande suivante (en *root*) :

```
# yum -y install python3
```

Une fois ce package installé, il est possible de créer un script *Python* simplement :

```
#
1 #! /usr/bin/python3
2 print("Hello World !")
```

Script 2.1: le programme helloworld.py

La 1^{ère} ligne indique que le script doit être exécuté avec l'interpréteur *python3*. A partir de la 2^{ème} ligne, on trouve le script en lui-même.

Pour **exécuter** le script sous Linux, il faut commencer par lui donner la permission en exécution (à faire une seule fois) :

```
$ chmod +x helloworld.py
$ ./helloworld.py
```

Exercices

- 1. Créer un script *Python*, nommé mklist.py, qui va **produire** à <u>l'écran</u> le fichier CSV auquel on aura ajouté 2 colonnes supplémentaires : les logins et mots de passe.
 - a. Les logins pour les utilisateurs suivront la règle : en fonction de la section de l'étudiant, la 1ère lettre sera a, b, i ou t. Nous aurons ensuite 4 chiffres représentant une séquence. Ainsi, a0001 représente le 1ier étudiant de la section automatique, i0054, le 54ème étudiant de la section informatique...
 - b. Pour générer le mot de passe de chaque utilisateur, utilisez la commande mkpasswd¹. Le mot de passe doit faire 12 caractères et comporter des lettres (majuscules et minuscules) et chiffres (reportez-vous à la documentation pour les paramètres à passer à mkpasswd).

Testez votre script et vérifiez si l'information à l'écran est correcte. **Ensuite seulement,** lancez votre script comme suit :

```
$ mklist.py > liste-login-pass.csv
```

Vous obtiendrez ainsi **un nouveau fichier CSV** contenant en plus les logins et mots de passe.

- 2. Créer un script Python nommé mkuser.py qui va, en se basant sur le fichier CSV obtenu à l'étape précédente, créer les comptes locaux des différents étudiants. On vous demande également de fixer, par ce script, les mots de passe et les données (noms et prénoms) de ces utilisateurs.
 - On vous demande de placer les étudiants dans le groupe principal (et local) *users* et dans un groupe secondaire correspondant à leur classe (en minuscule).

¹ mkpasswd est une commande système existante. Vous pouvez d'ailleurs voir son résultat :

^{\$} mkpasswd

3. Créer un script *Python* permettant d'ajouter facilement un utilisateur dans le LDAP. Pour ce faire l'utilisateur sera ajouté dans le OU=People, il sera configuré comme utilisateur POSIX, aura comme groupe principal, le groupe users (gid=100) et comme shell bash (/bin/bash). Le nom, le prénom, l'UID, le login et le mot de passe seront donnés en argument de votre script.

Ressources

- Lire un fichier ligne par ligne en *python* : https://www.geeksforgeeks.org/read-a-file-line-by-line-in-python/
- Utiliser des expressions régulières en *Python* : https://docs.python.org/fr/3/library/re.html
- Exécuter un programme et récupérer le résultat en *Python*:
 https://stackoverflow.com/questions/4760215/running-shell-command-and-capturing-the-output ou encore https://stackoverflow.com/questions/1410976/equivalent-of-bash-backticks-in-python
- Formater des nombres (ajouter des 0 en tête) : https://stackoverflow.com/questions/134934/display-number-with-leading-zeros
- Récupérer les arguments de la ligne de commande en *Python* avec *sys.argv* : https://www.geeksforgeeks.org/how-to-use-sys-argv-in-python/