

TP2 Système d'exploitation

Programmation Shell

NOTE: il faut fournir toutes les commandes ainsi que leurs résultats dans le compte rendu, ainsi que le contenu des scripts.

Exercice 1

Un script shell c'est un fichier texte regroupant un ensemble de commandes shell. Le script va être exécuté par le système en utilisant un interpréteur : sh, bash,

1. créez un répertoire en le nommant *scripts_tp4*
2. mettez vous dans le répertoire crée
3. créez un fichier en le nommant *script_exo1.sh*
4. éditez le fichier *script_exo1.sh* à l'aide d'un éditeur de texte en incluant la commande qui permet d'afficher le message **Bonjour le monde !**
5. exécutez le script comme suit (sans \$ bien sûr !):

```
$ sh script_exo1.sh
```

6. ajoutez la commande qui permet d'afficher le message **Bonjour le monde !** avec l'option -n puis re exécutez le script.
7. Qu'elle est l'effet observé de l'option -n ?
8. ajoutez au script *script_exo1.sh* la commande **date** puis re exécutez le script.
9. éditez le fichier *script_exo1.sh* en ajoutant la ligne suivante au tout début du fichier:
#!/bin/sh

10. enregistrez les modifications et afficher le contenu du répertoire en spécifiant l'option qui permet d'afficher les détails (y compris les droits sur les fichiers et le répertoires).

11. la commande **chmod** permet de ajouter/enlever/modifier les droits d'un fichier. Par exemple la commande **chmod a-w fichier1.txt** permet d'enlever (d'où le - entre **a** et **w**) les droit d'écriture (d'où la lettre **w** pour *write*) à tous les utilisateurs (d'où la lettre **a** pour *all*). Exécutez la commande qui permet d'ajouter (+ au lieu de -) les droits d'exécution (**x** pour *execute*) à l'utilisateur courant (**u** pour *user*) pour le fichier *script_exo1.sh*.

12. affichez le contenu du répertoire et notez les nouveaux droits du fichier

```
script_exo1.sh.
```

13. Exécutez le script *script_exo1.sh* comme suit (sans commande sh et sans espace entre ./ et *script_exo1.sh*) :

```
./script_exo1.sh
```

14. quel est le rôle de la ligne **#!/bin/sh** ajoutée au début du script ?

Exercice 2

Une variable est définie dans le shell au moment de son affectation et n'a pas de restrictions de type. Par exemple, pour définir une variable **x** qui prend la valeur 10, il suffit de mettre **x=10** sans espace entre les caractères. La variable est appelée en mettant le signe \$ devant. Ainsi, pour afficher la valeur de la variable **x** précédemment définie, il faut taper la commande :**echo \$x**

Toujours dans le répertoire *scripts_tp4* :

1. créez un script shell nommé *script_exo2-1.sh*, éditez le en spécifiant l'interpréteur de commande sh, puis rajoutez les droits d'exécution à tous les utilisateurs pour ce script.
2. définissez une variable appelée **MSG1** en lui affectant la chaîne de caractère **Bonjour**
3. définissez une variable appelée **MSG2** en lui affectant la chaîne de caractère **les 2A !**
4. ajouter la commande qui permet d'afficher en une seule commande le message **Bonjour les 2A !** en utilisant les variables **MSG1** et **MSG2**

Une commande peut être évaluée en utilisant des parenthèses(), ainsi nous pouvons récupérer son résultat et l'affecter à une variable par exemple.

5. créez un script shell nommé *script_exo2-2.sh*, éditez le en spécifiant l'interpréteur de commande sh, puis rajouter les droits d'exécution à tous les utilisateurs pour ce script.
6. définissez une variable appelée **RES1** comme suit : **RES1=ls**
7. définissez une variable appelée **RES2** comme suit : **RES2=\$(ls)**
8. ajouter les deux commandes qui permettent d'afficher les valeurs des variables RES1 et RES2.
9. Exécutez le script sans commande sh. Que remarquez-vous ?

Un script shell peut prendre des paramètres au moment de son exécution comme suit :
./script.sh paramètre1 paramètre2 ...

Les valeurs des paramètres sont récupérés dans les variables \$1 pour paramètre1, \$2 pour paramètre2, etc.

10. créez un script shell nommé *script_exo2-3.sh*, éditez le en spécifiant l'interpréteur de commande sh, puis rajouter les droits d'exécution à tous les utilisateurs pour ce script.
11. définissez une variable appelée **MADATE** en lui affectant le résultat de la commande **date**

12. ajouter la commande qui permet d'afficher le message : Bonjour NOM PRENOM. La date d'aujourd'hui est : DATE. NOM et PRENOM sont donnés en paramètre au script et DATE est la date d'aujourd'hui stockée dans **MADATE**.