

Pratique des machines

TP3 : Scripts bash

am@up8.edu

Septembre 2022

Dans ce TP :

- écriture de scripts bash
- permissions
- Sources : TPs de Valentin Richard

Avant de commencer : créer un répertoire TP3 dans le répertoire dédié à ce cours sur votre session.

1 Exercice 1 : script bash

1. Démarrage :

- (a) Écrire un script `mon_premier_script.sh` de 2 lignes qui crée un fichier de nom `fichier.txt` et liste le contenu du répertoire courant :

```
1 $ echo "touch fichier.txt"
2 > ls" > mon_premier_script.sh
```

Rq 1 : le signe \$ indique l'invite de commande du terminal.

Rq 2 : le signe ">" ligne 2 a été ajouté par le terminal pour signaler que la chaîne de caractères n'est pas terminée. Vous ne devez pas écrire ce caractère vous même.

- (b) Vérifier le contenu du fichier à l'aide de la commande `cat`.
(c) Lancer le script à l'aide de l'interpréteur :

```
1 $ bash mon_premier_script.sh
```

Les commandes présentes dans le script se sont-elles exécutées ?

- (d) **Recopiez** le contenu ci-dessous dans un autre script `lecture.sh`

```
1 #!/bin/bash
2
3 VAR="Hello world"
4 echo "$VAR"
```

Le shebang `#!` indique le chemin de l'interpréteur, ici `/bin/bash`.

Rq : la commande

```
1 $ which bash
```

vous indique le chemin absolu vers l'interpréteur bash utilisé par défaut par votre terminal.

(e) Exécutez le script grâce à la commande suivante :

```
1 $ ./lecture.sh
```

Quel est le message d'erreur que vous recevez ?

(f) Vous devez autoriser l'exécution du script grâce à la commande :

```
1 $ chmod u+x lecture.sh
```

Exécutez le script à nouveau (étape (e)).

2. Modification du script Le but est maintenant de créer un script qui demande le prénom de l'utilisateur et l'accueille.

D'abord, changer la valeur initiale donnée à VAR pour qu'elle soit votre prénom.

On peut utiliser `echo` pour afficher en même temps du texte écrit en dur et la valeur de la variable :

```
1 echo "Bonjour $VAR"
```

Exécutez pour vérifier que ça marche. Pour demander la valeur d'une variable à l'utilisateur (mode interactif), on utilise la commande `read` :

```
1 read VAR
```

Placer cette ligne à la place de l'assignation de VAR. Ajouter une ligne avant celle-ci pour afficher le message "Quel est votre nom?".

Exécuter le script.

3. On peut encore l'améliorer si on souhaite afficher "Quel est votre nom?" sur la même ligne où on doit écrire le nom.

En utilisant l'aide sur les commandes `echo` et `read`, trouvez deux manières différentes de faire ça et notez-les dans un fichier.

vous devez utiliser l'option `-n` de `echo` ou bien l'option `-p` de `read`

2 Exercice 2 : Permissions

2.1 Méta-données d'un fichier

On a vu qu'il fallait ajouter la permission d'exécution au script pour pouvoir l'exécuter. On peut avoir des informations sur les permissions des éléments du dossier courant avec la commande

```
1 $ ll
2 ou
3 $ ls -l
```

1. 2. 3. 4. 5. 6. 7.

```
drwxrwxr-x 4 vrichard vrichard 4096 sept. 14 10:23 Lds
-rwxrwxr-x 1 vrichard vrichard 43 oct. 4 14:07 lecture.sh
```

1. Type de fichier : d = dossier (répertoire), - = fichier ordinaire
2. Permissions
3. Nombre de liens physiques (par défaut 1)
4. Utilisateur propriétaire et groupe propriétaire
5. Taille du fichier (en octets)
6. Horodatage (mois, jour et heure) de la dernière modification
7. Nom de l'élément

1. En utilisant l'aide pour `ls`, affichez la liste des dossiers (seulement) avec leur taille et l'unité de taille. Notez la commande dans un fichier.

2.2 Changer une permission

Il y a trois types de permissions :

Lettre	Droit si fichier	Droit si dossier	Code octal
-	pas de permission	pas de permission	0
r	lecture	lister les éléments et copier	4
w	écriture	créer/supprimer éléments	2
x	exécuter	entrer dans le dossier	1

Il y a trois types d'entités auxquelles on peut donner des permissions :



Les utilisateurs (lettre u) sont identifiés par leur UID (identifiant unique) et les groupes (lettre g) par leur GID.

On peut afficher son UID et le GID du groupe auquel on appartient avec

```
1 $ id
```

`whoami` renvoie le nom de l'utilisateur actuel.

Créez un dossier `dir`. Créez un fichier `file`, et affichez ses permissions avec `$ ll file`.

On utilise `chmod` pour gérer les permissions. Pour ajouter la permission `w` au groupe sur le fichier `file`, il faut faire

```
1 $ chmod g+w file
```

Refaites 11 file et remarquez ce qui a changé.

Pour l'enlever, on met un - au lieu du +. Pour changer la permission de l'utilisateur au lieu du groupe, on utilise u au lieu de g, la lettre o pour les autres (other), et a pour tout le monde (all). On peut déclarer plusieurs changements en les séparant par des virgules, par exemple :

```
1 $ chmod u-r,o-x dir
```

Une autre façon de faire, est d'utiliser un masque avec une combinaison des codes octaux des permissions à donner (voir tableau en page précédente). Un code octal est composé de 3 chiffres : le premier code les droits de l'utilisateur, le deuxième du groupe et le troisième des autres.

Par exemple 421 signifie : lecture seule (4) pour l'utilisateur, écriture seule (2) pour le groupe et exécution seule (1) pour les autres.

On peut combiner les codes en les additionnant, par exemple 3 = 1 + 2 signifie écriture (2) + exécution (1).

Ainsi, les droits maximums donnent 7 = 1 + 2 + 4.

Par exemple, 753 signifie : tous les droits pour l'utilisateur, lecture et exécution (5 = 1 + 4) pour le groupe, et exécution + écriture pour les autres. En lettres, ça donne **rxr-xrw-**.

2.3 Changer de propriétaire

Il est aussi possible de changer l'utilisateur propriétaire d'un fichier avec :

```
1 $ chown utilisateur fichier
```

ou le groupe propriétaire :

```
1 $ chgrp groupe fichier
```

3 [ANNULÉ] Exercice 3 (script à rendre)

Créez un script qui :

1. crée un dossier env dans le dossier de téléchargement de votre home (donc Download ou Téléchargements, ou créez-le avant s'il n'existe pas)
2. se déplace dans ce dossier
3. copie le fichier /etc/environment dans le dossier courant
4. imprime le contenu de tous les fichiers du dossier courant dans le terminal
5. affiche le message "Mon ordinateur a tant de processeurs : ", sans retour à la ligne
6. Effectue l'enchaînement de commandes suivant (utiliser l'opérateur |) :
 - (a) liste le contenu du dossier cpu situé dans /dev
 - (b) filtre les noms de fichier ne contenant que des chiffres avec `grep -E "[0-9]+"`
 - (c) compte le nombre de lignes

4 Exercice 4 (script à rendre)

1. Créez un script `replicateur.sh` qui fait (une ligne par instruction) :
 - (a) crée un fichier `replicat.sh`
 - (b) ajoute le droit d'exécution à ce fichier pour l'utilisateur
 - (c) imprime le contenu de `replicateur.sh` dans `replicat.sh`
 - (d) affiche "Réplication!" dans la terminal
 - (e) attend 1 seconde (voir l'aide de la commande `sleep`)
 - (f) exécute `replicat.sh`
 - (g) imprime "Fin" dans le terminal
2. Donnez les droits nécessaires à `replicateur.sh` et lancez-le. Arrêtez au bout de quelques secondes avec `Ctrl + c`.
3. Pour éviter les fautes de frappe, il est conseillé de stocker dans une variable les valeurs couramment utilisées. Ici on fait plusieurs fois référence à `replicat.sh`. Comme dans le script `lecture.sh`, déclarez une variable `FILE` et assignez-lui la chaîne de caractères `'replicat.sh'`.
4. Substituez toutes les occurrences de `replicat.sh` dans votre code par `"$FILE"`.
5. Pour limiter le temps d'exécution du script à 3 secondes, lancez-le précédé de `timeout 3s`.

Code de `replicateur.sh`

```
1 #!/bin/bash
2 FILE="replicat.sh"
3 touch "$FILE"
4 chmod u+x "$FILE"
5 cat replicateur.sh > "$FILE"
6 echo "Réplication !"
7 sleep 1
8 ./"$FILE"
9 echo "Fin"
```