

TP3 : Pratique des machines

Créer des archives, permissions

am@up8.edu

Octobre 2025

1 Création d'archives zip

→ Dans un dossier TP3, créez un fichier vide appelé `reponses.txt`.

Consultez le lien <https://pads.up8.edu/s/27-owSEqm> section Comment faire un rendu propre sur moodle ? et suivez les instructions pour faire un rendu propre pour votre fichier `reponses.txt` et déposez l'archive créée dans la zone de dépôt moodle.

→ L'archive doit avoir le nom `VOTRE_NOM_TP3.zip`, contenir un dossier `VOTRE_NOM_TP3` qui contient votre fichier `reponses.txt`.

2 Permissions

2.1 Méta-données d'un fichier

On a vu que la commande `ls -l` nous donne plus de détails qu'un simple `ls`. On peut notamment y trouver les [permissions](#) des fichiers.

```
$ ls
$ ou
$ 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

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 :

Groupe propriétaire
Utilisateur propriétaire

r	w	x	r	w	x	r	-	-
r	w	-	r	w	-	r	-	-

Autres

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

→ affichez votre UID et le GID du groupe auquel vous appartenez avec :

```
$ id
```

→ utilisez whoami pour afficher le nom de l'utilisateur actuel.

→ Créez un dossier `dir`. Créez un fichier `file`, et affichez ses permissions avec `ls -l file`.

On utilise la commande chmod pour gérer les permissions.

- Pour **ajouter une permission** en écriture (*write*) aux autres utilisateurs (**o**) sur le fichier `file`, il faut faire :

```
$ chmod o+w file
```

Refaites `ls -l file` et remarquez ce qui a changé.

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

```
$ chmod u-r,o-x dirs
```

3 Protéger son espace personnel

Tous les fichiers et dossiers n'ont pas les mêmes permissions (lecture, écriture, exécution).

1. Quelles sont les permissions sur votre dossier personnel (`~`)?
2. Quelles sont les permissions sur le dossier personnel de votre voisin ? Pouvez-vous créer un fichier **test_VOTRENOM.txt** dans le dossier personnel de votre voisin ?
3. Essayez depuis l'explorateur de fichiers de changer les paramètres de votre dossier personnel pour interdire l'écriture aux autres *users*. Regardez si cela a eu un effet.
4. Proposez une commande `bash` permettant de faire ce changement de permissions.

4 Exercice 3 : créer des scripts bash

Pour l'instant, nous n'avons exécuté d'instructions `bash` qu'en [ligne de commande](#). Nous allons voir ici comment les stocker dans des scripts (fichiers [.sh](#)) pour pouvoir les réutiliser.

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 :

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

Rq 2 : le premier 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 :

```
$ 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`

```
-----  
#!/bin/bash
```

```
VAR="Hello world"  
echo "$VAR"
```

```
-----
```

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

Rq : la commande

```
$ 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 :

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

Quel est le message d'erreur que vous recevez ?

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

```
$ 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 :

```
$ 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` :

```
$ 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.