

TD 1 : Création de cinq fichiers vides avec la commande touch et la boucle for :

```
#!/bin/bash
# La commande mkdir permet de créer un dossier/répertoire
mkdir folder
# La commande cd permet de se déplacer dans un dossier/répertoire
cd folder
for i in {1..5}; do
    # la commande touch permet pour créer un fichier vide
    touch "file$i"
done
# La commande echo permet d'afficher du texte sur le terminal
echo "voici la liste des fichier"
# la commande ls pour lister les fichiers créer
ls
```

TD 2 : Recherche d'un fichier avec la commande find, la condition if et l'utilisation d'une variable entrée par l'utilisateur:

```
#!/bin/bash
echo "Entrer le fichier à rechercher"
# La commande read attend une entrée de l'utilisateur sur le terminal et
l'enregistre dans la variable search
read search
# La commande find recherche un fichier dans une hiérarchie de
répertoires
find "$search"
# $? prend le code de retour de la commande précédant la ligne
actuelle. Si la commande a réussi, le code est 0 ; sinon, il est différent de
0
if [ $? -eq 0 ]; then
    echo "Le fichier a été trouvé"
else
    echo "Le fichier n'existe pas"
```

TD 3 : Vérification de la permission d'un fichier avec les opérateurs des scripts bash:

Ce script est l'équivalent de la commande ls -l, qui affiche les droits sur un fichier donné

```
#!/bin/bash
echo "Entrer le fichier à vérifier"
read file
if [ -e "$file" ]; then
    echo "le fichier existe"
    if [ -r "$file" ]; then
        echo "Le fichier $file est lisible"
    else
        echo "Le fichier $file n'est pas lisible"
    fi
    if [ -w "$file" ]; then
        echo "Le fichier $file est modifiable"
    else
        echo "Le fichier $file n'est pas modifiable"
    fi
    if [ -x "$file" ]; then
        echo "Le fichier $file est exécutable"
    else
        echo "Le fichier $file n'est pas exécutable"
    fi
else
    echo "le fichier n'existe pas"
```

TD 4 : Planificateur de tâches avec la boucle while en utilisant la commande date:

Ce script vérifie si l'heure actuelle du système d'exploitation est égale à l'heure de référence.

```
#!/bin/bash
target="20:24"
while [ $(date +%H:%M) != $target ]; do
    # La commande date affiche l'année, le mois, le jour ainsi que
    l'heure actuelle du système d'exploitation
    echo "ce n'est pas encore l'heure de sortir"
    # la commande sleep 1 met la boucle en pause pendant 1
    seconde
    sleep 1
done
```