

IUT DE COLMAR

R405 AUTOMATISATION DES TÂCHES

ANNÉE 2022-23

---

## TP 3 : PING

---

MARTIN BAUMGAERTNER

29 mars 2023

---

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Exercice 1</b>	<b>2</b>
<b>3</b>	<b>Exercice 2</b>	<b>4</b>
<b>4</b>	<b>Exercice 3</b>	<b>5</b>

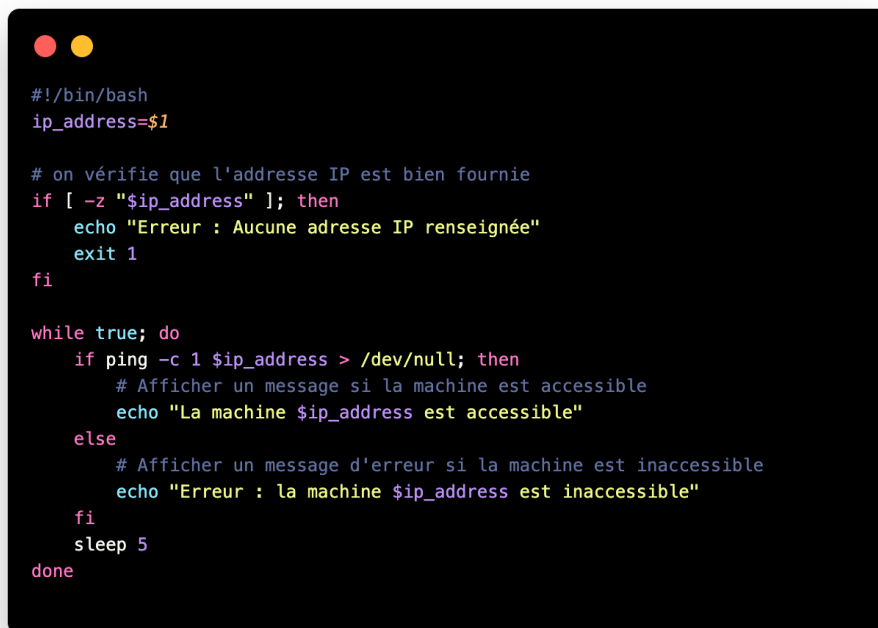
---

# 1 Introduction

Ce premier TP a pour objectif de vous faire manipuler le programme ping sous Bash. Vous utiliserez pour cela, une machine virtuelle Linux. Durant ce TP vous devrez produire un compte rendu au format PDF qui contient les commandes et scripts Bash associés aux questions accompagnés d'une explication. La première page du rapport devra contenir votre nom et prénom ainsi que le numéro du TP réalisé. Ces comptes rendus seront relevés et devront être déposés sur Moodle, les noms des comptes rendus devront être de la forme TP1-Bash-Nom- Prénom. Vous utiliserez la commande ping en respectant cet ordre pour les paramètres : ping [option] adresseIP et vous n'utilisez pas l'option -w (en raison de l'implémentation bash sous MAC).

## 2 Exercice 1

Voici donc le script que j'ai trouvé. Premièrement on demande à l'utilisateur de rentrer l'adresse IP qu'il veut tester en boucle. Puis on vient la tester comme demandé dans l'énoncé.

A terminal window with a dark background and light-colored text. The script is a Bash script that takes an IP address as input and checks if it is accessible via ping. It uses a while loop to keep checking until the user presses Ctrl+C. The script includes comments in French explaining each step.

```
#!/bin/bash
ip_address=$1

# on vérifie que l'adresse IP est bien fournie
if [ -z "$ip_address" ]; then
    echo "Erreur : Aucune adresse IP renseignée"
    exit 1
fi

while true; do
    if ping -c 1 $ip_address > /dev/null; then
        # Afficher un message si la machine est accessible
        echo "La machine $ip_address est accessible"
    else
        # Afficher un message d'erreur si la machine est inaccessible
        echo "Erreur : la machine $ip_address est inaccessible"
    fi
    sleep 5
done
```

FIGURE 1 – Premier script

---

Voici les droits sur mon script pour vous prouver qu'il n'a pas été réalisé en **root** :

```
parallels@debian-gnu-linux-11:~/Documents$ ls -l
total 4
-rw-r--r-- 1 parallels parallels 372 Mar 22_08:54 script1.sh
```

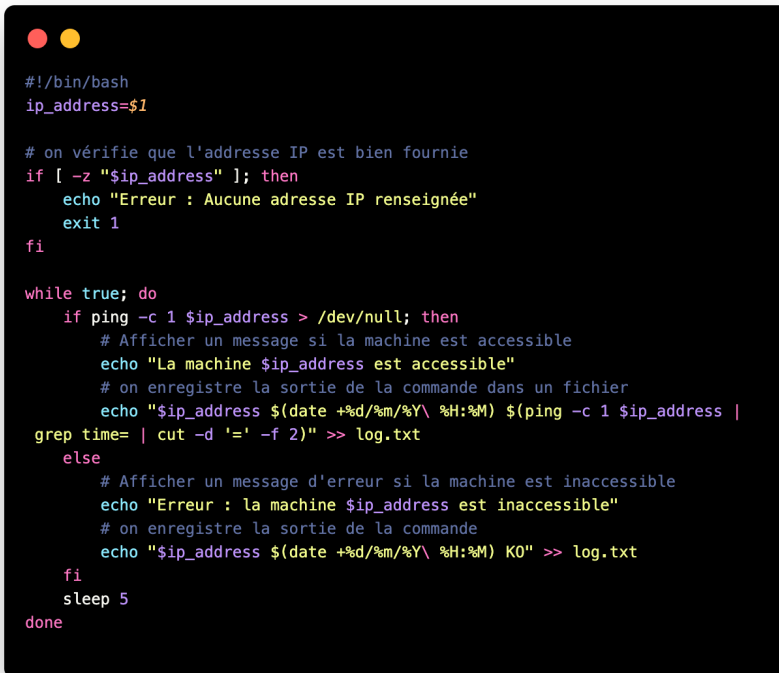
FIGURE 2 – Preuve droits script

Comme le démontre la capture d'écran, le script à été réalisé en tant que utilisateur **parallels**. Cet utilisateur correspond à l'utilisateur par défaut des machines virtuelles virtualisés par **parallels desktop**, logiciel de virtualisation sur MacOS.

---

## 3 Exercice 2

Voici donc le script que j'ai créé. J'ai juste repris le code précédent et ajouté l'enregistrement des résultats dans un fichier texte.



```
#!/bin/bash
ip_address=$1

# on vérifie que l'adresse IP est bien fournie
if [ -z "$ip_address" ]; then
    echo "Erreur : Aucune adresse IP renseignée"
    exit 1
fi

while true; do
    if ping -c 1 $ip_address > /dev/null; then
        # Afficher un message si la machine est accessible
        echo "La machine $ip_address est accessible"
        # on enregistre la sortie de la commande dans un fichier
        echo "$ip_address $(date +%d/%m/%Y\ %H:%M) $(ping -c 1 $ip_address |
grep time= | cut -d '=' -f 2)" >> log.txt
    else
        # Afficher un message d'erreur si la machine est inaccessible
        echo "Erreur : la machine $ip_address est inaccessible"
        # on enregistre la sortie de la commande
        echo "$ip_address $(date +%d/%m/%Y\ %H:%M) KO" >> log.txt
    fi
    sleep 5
done
```

FIGURE 3 – Deuxième script

Comme avant, voici les droits sur le fichier :

```
parallels@debian-gnu-linux-11:~/Documents$ ls -l
total 12
-rw-r--r-- 1 parallels parallels 180 Mar 22 09:49 log.txt
-rw-r--r-- 1 parallels parallels 514 Mar 22 09:37 script1.sh
-rw-r--r-- 1 parallels parallels 812 Mar 22 09:47 script2.sh
```

FIGURE 4 – Preuve droits script

---

Et voici ce que l'on obtient dans le fichier de log, on a bien le format qu'il était demandé :

```
parallels@debian-gnu-linux-11:~/Documents$ cat log.txt
8.8.8.8 22/03/2023 09:48 1 ttl
8.8.8.8 22/03/2023 09:48 1 ttl
8.8.8.8 22/03/2023 09:48 K0
8.8.8.8 22/03/2023 09:49 K0
8.8.8.8 22/03/2023 09:49 1 ttl
8.8.8.8 22/03/2023 09:49 1 ttl
```

FIGURE 5 – Fichier logs

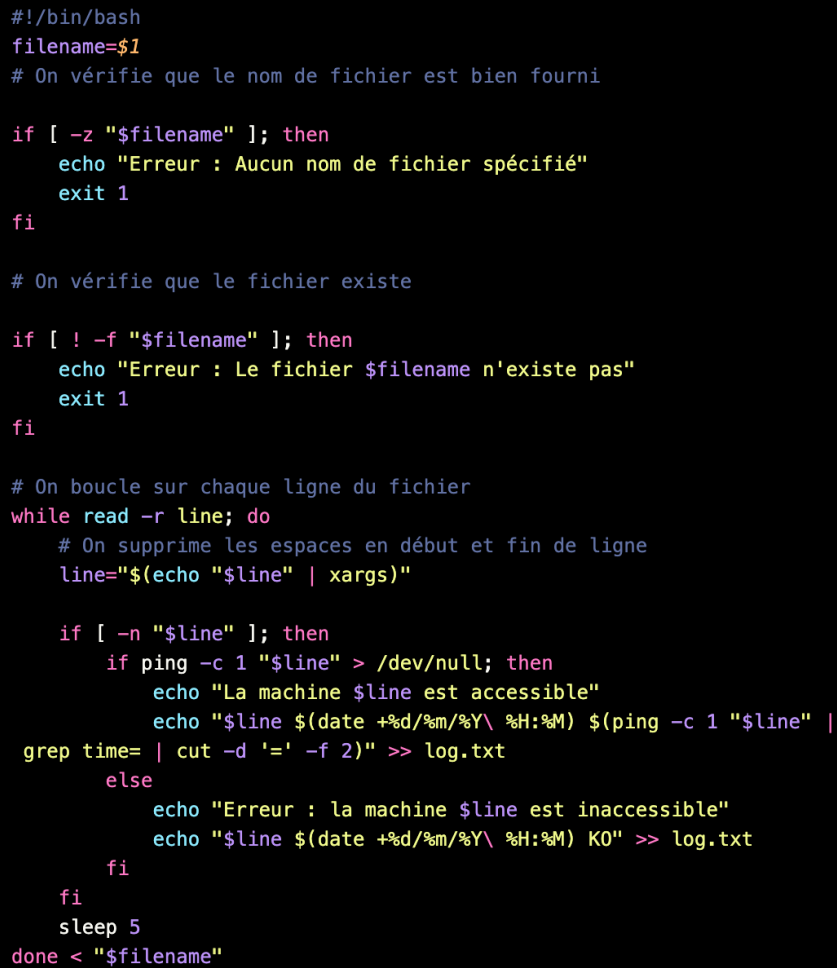
## 4 Exercice 3

Vous pouvez donc voir à la page suivante le troisième script que j'ai réalisé. Nous pouvons tout d'abord voir que j'ai créé un fichier avec les adresses IP à tester et que nous avons toujours dans le fichier de log les résultats des commandes :

```
parallels@debian-gnu-linux-11:~/Documents$ sh script3.sh addresses.txt
La machine 8.8.8.8 est accessible
La machine 1.1.1.1 est accessible
parallels@debian-gnu-linux-11:~/Documents$ cat log.txt
8.8.8.8 22/03/2023 10:34 1 ttl
1.1.1.1 22/03/2023 10:44 1 ttl
parallels@debian-gnu-linux-11:~/Documents$ cat addresses.txt
8.8.8.8
1.1.1.1
```

FIGURE 6 – Preuve 3

---



```
#!/bin/bash
filename=$1
# On vérifie que le nom de fichier est bien fourni

if [ -z "$filename" ]; then
    echo "Erreur : Aucun nom de fichier spécifié"
    exit 1
fi

# On vérifie que le fichier existe

if [ ! -f "$filename" ]; then
    echo "Erreur : Le fichier $filename n'existe pas"
    exit 1
fi

# On boucle sur chaque ligne du fichier
while read -r line; do
    # On supprime les espaces en début et fin de ligne
    line="$(echo "$line" | xargs)"

    if [ -n "$line" ]; then
        if ping -c 1 "$line" > /dev/null; then
            echo "La machine $line est accessible"
            echo "$line $(date +%d/%m/%Y\ %H:%M) $(ping -c 1 "$line" |
grep time= | cut -d '=' -f 2)" >> log.txt
        else
            echo "Erreur : la machine $line est inaccessible"
            echo "$line $(date +%d/%m/%Y\ %H:%M) KO" >> log.txt
        fi
    fi
    sleep 5
done < "$filename"
```

FIGURE 7 – Troisième script