

TSSR - Compte rendu TP6: Scripts Bash Sous Debian

Jean-Paul MELISSE

23 Janvier 2025

1 Introduction

L'objectif de ce TP est de nous familiariser avec les scripts Bash sur Linux. Pour cela, on va créer une série de 9 petits scripts Bash sur une machine virtuelle Debian. Ce petit document va nous décrire les étapes à suivre.

2 Créer un dossier de travail

On choisit un emplacement de travail où on va garder tous les fichiers installés et la configuration de la machine Debian. On décide de sauvegarder dans le chemin : "C:\Users\Jean-Paul\Desktop\tp6", comme montre la Figure 1.

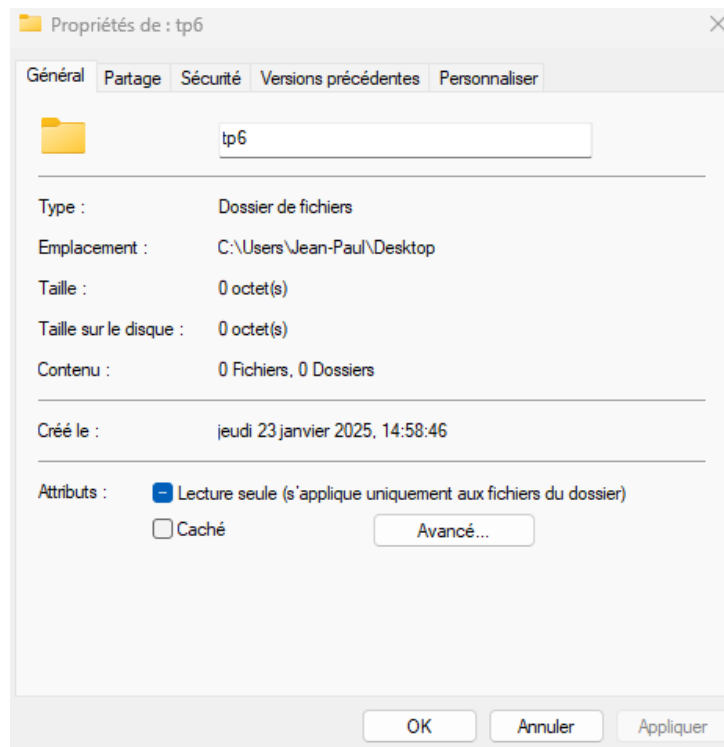


Figure 1

3 Télécharger Debian

La suite est de télécharger le système d'exploitation Debian. On y accède via le lien <https://www.debian.org/>. On clique sur "Téléchargement". On le sauvegarde aussi dans le dossier tp4 sous le format .iso (Comme montre la Figure 2).

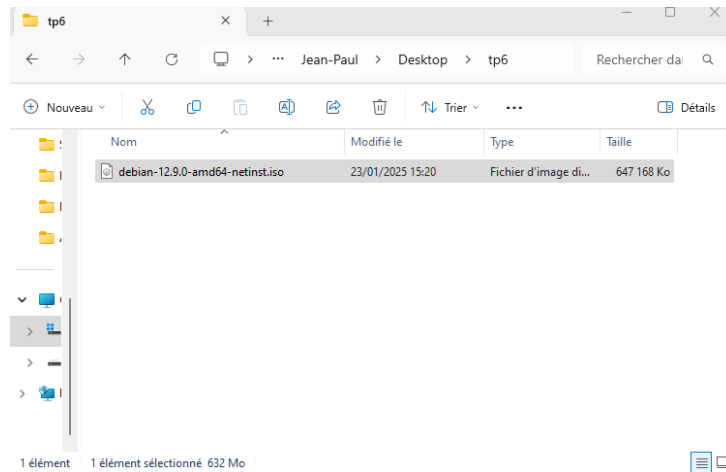


Figure 2

4 Créer la machine virtuelle tp6-debian

On ouvre VirtualBox. On clique sur "Nouveau" pour créer une nouvelle machine virtuelle. On met "tp6-debian" comme nom de la machine virtuelle. On choisit son emplacement de sauvegarde et on importe le CD contenant le système d'exploitation de Debian. En détectant le CD, le type "Linux", le subtype "Debian" et la version "Debian (64 bits)" sont automatiquement mis. On coche la case "Skip Unattended Installation" et on clique sur "Suivant". On lui donne une mémoire vive (RAM) de 2048 Mo et un processeur, et on clique sur "Suivant". On lui donne un disque dur de 20 Go et on clique sur "Suivant". On regarde bien le récapitulatif et on clique sur "Finish" (Voir la Figure 3).

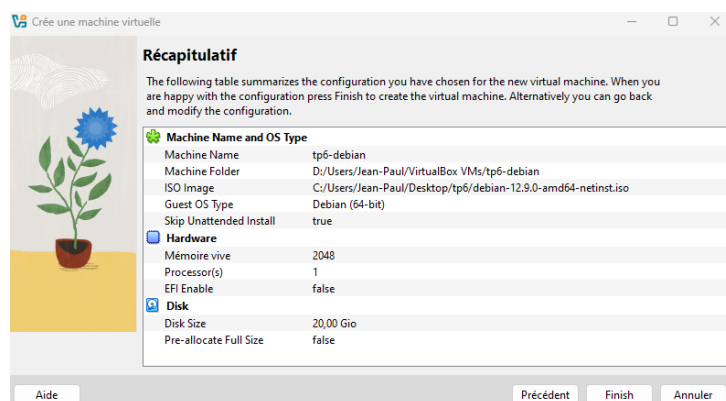


Figure 3

Maintenant que la nouvelle machine est créée, il faut la configurer. On clique sur "Configurations". On clique sur "Réseau" et on change le mode d'accès réseau à "Accès par pont", puis on clique sur "OK". Le mode "Accès par pont" va permettre à la machine virtuelle de récupérer une adresse IP directement

depuis un serveur DHCP (par exemple : notre box). Cela permet de connecter la machine virtuelle à l'Internet. Maintenant qu'on a configuré la machine virtuelle, on clique sur "Démarrer" pour l'allumer.

5 Installer Debian sur tp6-debian

Après avoir démarré la machine virtuelle, on choisit l'option "Graphical Install". On choisit la langue "English" et on clique sur "Continue". Le fait qu'on ait choisi l'anglais comme langue, Debian va nous proposer des pays anglophones par défaut pour la géolocalisation. On choisit donc "other" et on clique sur "Continue". Il nous propose donc dans un premier temps, des zones continentales. On choisit "Europe" pour la localisation car on se trouve en Europe et on clique sur "Continue". Enfin, on choisit "France" pour la localisation et on clique sur "Continue". Debian comprend qu'il y a une incohérence entre la langue choisie et le pays. Il nous propose différentes versions de la langue anglaise en fonction des pays. On choisit "United States - en_US.UTF-8" et on clique sur "Continue". Pour la configuration du clavier, on choisit "French" et on clique sur "Continue". Après avoir reçu par le service DHCP une adresse IP, on renomme le nom de l'hôte (Hostname) en "tp6-debian" et on clique sur "Continue" (Figure 4).

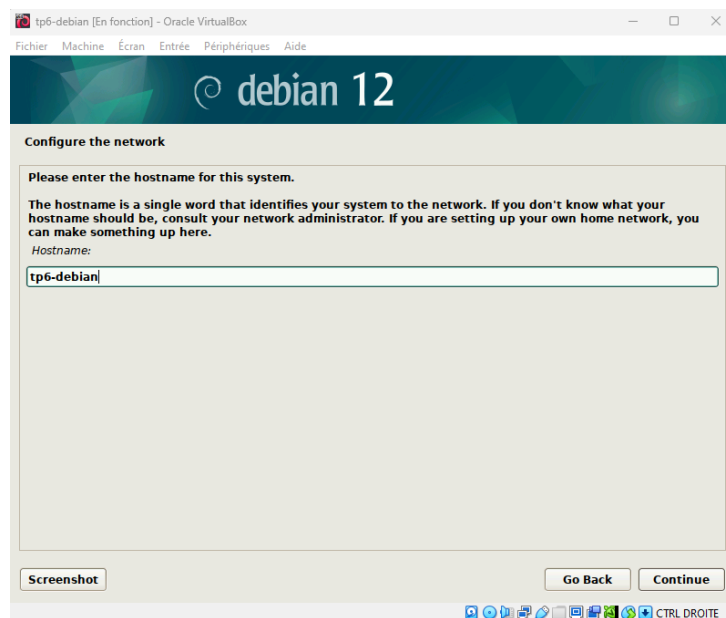


Figure 4

On lui donne un nom de domaine (Domain Name), qui sera "afpa" ici et on clique sur "Continue". On donne un mot de passe à l'administrateur root (dans ce cas ici, le mot de passe est "root" et on le met deux fois) et on clique sur "Continue". On crée un nouvel utilisateur qu'on nomme "user" et on clique sur "Continue" deux fois. Comme pour l'administrateur root, on donne un mot de passe à l'utilisateur user (dans ce cas ici, le mot de passe est "user" et on le met deux fois) et on clique sur "Continue".

On choisit l'option "Guide - use entire disk" pour la partition du disque et on clique sur "Continue" deux fois. Pour la partition, on choisit "All files in one partition" et on clique sur "Continue" deux fois. On coche la case "Yes" pour appliquer les changements sur la partition du disque et on clique sur "Continue". Comme on a mis un seul disque dur pour cette machine virtuelle, on n'a pas besoin de scanner d'autres

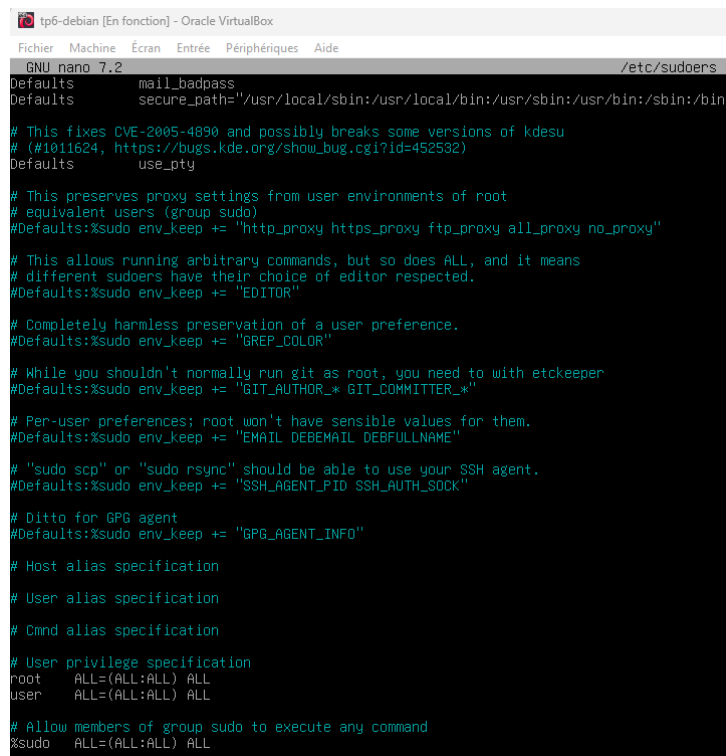
disques. On coche la case "No" et on clique sur "Continue".

La prochaine étape concerne le gestionnaire de paquets "Advanced Packaging Tool" (APT). Ce gestionnaire contient tous les programmes et applications utilisés pour les machines Debian et Ubuntu. Il gère aussi les mises à jour de ces applications. On choisit le pays "France" et on clique sur "Continue". On choisit le miroir pour le gestionnaire de paquets : "deb.debian.org" et on clique sur "Continue" deux fois. Ce qui signifie que quand on va utiliser le gestionnaire APT, il va se connecter à ce miroir pour vérifier ou installer les paquets. On coche la case "No" car on ne veut pas participer aux études statistiques et on clique sur "Continue".

On décoche toutes les cases des logiciels et on clique sur "Continue". On coche la case "Yes" pour installer le boot GRUB et on clique sur "Continue". On choisit l'option "/dev/sda" pour installer GRUB dans cet emplacement et on clique sur "Continue". On clique sur "Continue" pour finaliser l'installation de Debian et redémarrer la machine. Note : Après le redémarrage de Debian, on peut faire un clone de notre machine virtuelle.

6 Configurer sudo sur tp6-debian

Le suite maintenant est de configurer la commande sudo afin qu'on puisse l'utiliser avec l'utilisateur user. Après avoir redémarré la machine virtuelle "tp6-debian", on se connecte avec les identifiants de l'utilisateur root. On tape la commande "apt install man sudo" pour installer le paquet manuel et la commande sudo. Après avoir installé sudo, on va modifier le fichier sudoers pour ajouter l'utilisateur user dans les droits de permissions. On fait une copie du fichier sudoers en tapant la commande "cp /etc/sudoers /etc/sudoers.ori" pour éviter d'écraser le fichier original. Pour modifier ce fichier, on tape la commande : "nano /etc/sudoers". Dans ce fichier, on ajoute après la ligne du root la commande : "user ALL=(ALL:ALL) ALL". On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder avec le même nom. On peut visualiser cette étape sur la Figure 5.



```
tp6-debian [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
GNU nano 7.2 /etc/sudoers *
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# This fixes CVE-2005-4890 and possibly breaks some versions of kdesu
# (#1011624, https://bugs.kde.org/show_bug.cgi?id=452532)
Defaults        use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
user    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

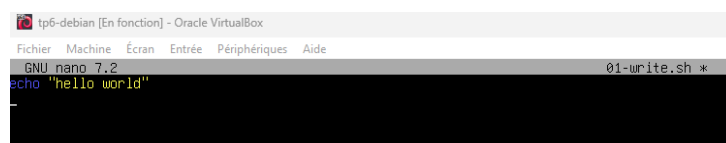
Figure 5

On a fini de configurer le fichier sudo. On peut se déconnecter du compte en utilisant la commande "logout".

7 Les sorties en Bash

Par la suite du TP, on se concentre sur les scripts Bash. On se connecte avec les identifiants de l'utilisateur user. On crée un nouveau dossier "tp6-jean-paul-melisse" qui va contenir tous les scripts Bash. Puis on se rend dans ce dossier en tapant la commande "cd tp6-jean-paul-melisse/"

Les entrées-sorties désignent les échanges de données entre le processeur (CPU) et les périphériques. Les entrées sont les données envoyées par les périphériques d'entrée (clavier, scanner, . . .) et lues par le CPU. Les sorties sont les données reçues par les périphériques de sortie (écran, imprimante, . . .) et écrites par le CPU. Dans cette section, nous nous intéressons aux sorties et à écrire à l'écran. Pour écrire à l'écran, nous utilisons la builtin echo. Pour en savoir plus sur l'utilisation de cette builtin, tapez "man bash". On va créer un script 01-write.sh qui écrit hello world à l'écran. Pour cela, on tape la commande "sudo nano 01-write.sh". Dans ce fichier, on écrit la ligne : 'echo "hello world"', comme montré sur la Figure 6.

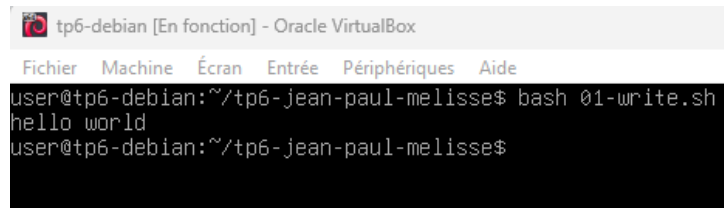


```
tp6-debian [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
GNU nano 7.2 01-write.sh *
echo "hello world"
```

Figure 6

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder.

On peut vérifier l'exécution du script en tapant la commande : "bash 01-write.sh" (Voir la Figure 7).



```
tp6-debian [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 01-write.sh
hello world
user@tp6-debian:~/tp6-jean-paul-melisse$
```

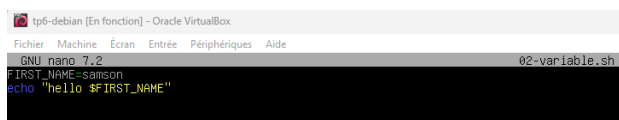
Figure 7

8 Les variables en Bash

Une variable désigne un nom affecté à une valeur stockée en mémoire. Pour déclarer une variable, il suffit de lui affecter une valeur, exemple : "NAME=value". Pour faire référence à une variable déclarée, il suffit d'indiquer son nom précédé par un \$ (dollar), exemple : \$NAME. On va créer un script 02-variable.sh qui déclare une variable FIRST_NAME en lui affectant la valeur samson et qui l'écrit à l'écran. Pour cela, on tape la commande "sudo nano 02-variable.sh". Dans ce fichier, on écrit les lignes suivantes :

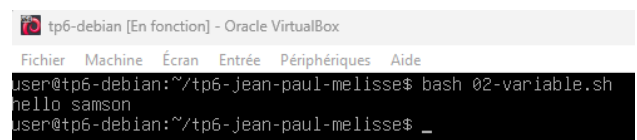
- 1 FIRST_NAME=samson
- 2 echo hello \$FIRST_NAME

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir l'étape précédente sur la Figure 8. On vérifie l'exécution du script en tapant la commande : "bash 02-variable.sh" (voir la Figure 9).



```
tp6-debian [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide
GNU nano 7.2                                02-variable.sh
FIRST_NAME=samson
echo "hello $FIRST_NAME"
```

Figure 8



```
tp6-debian [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 02-variable.sh
hello samson
user@tp6-debian:~/tp6-jean-paul-melisse$ _
```

Figure 9

9 Les entrées en Bash

Dans cette section, nous nous intéressons aux entrées et à lire au clavier. Pour lire au clavier, nous utilisons la builtin "read". On va créer un script "03-read.sh" qui va lire au clavier une chaîne de caractères représentant un prénom et qui l'écrit à l'écran. On tape la commande "sudo nano 03-read.sh". Dans ce fichier, on écrit les lignes suivantes :

- 1 read -p "first name? " FIRST_NAME
- 2 echo "Your name is : \$FIRST_NAME"

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir l'étape précédente sur la Figure 10. On vérifie l'exécution du script en tapant la commande : "bash 03-read.sh" (voir la Figure 11).

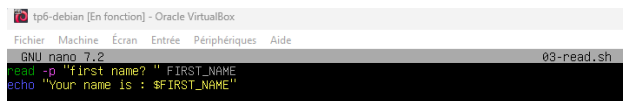


Figure 10

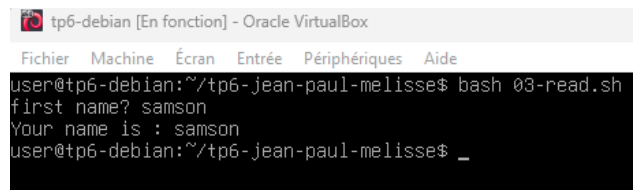


Figure 11

10 Les conditions en Bash

Bien que les instructions d'un script soient habituellement exécutées de façon séquentielle, il est possible d'en exécuter à l'intérieur d'une structure conditionnelle.

- Syntaxe :

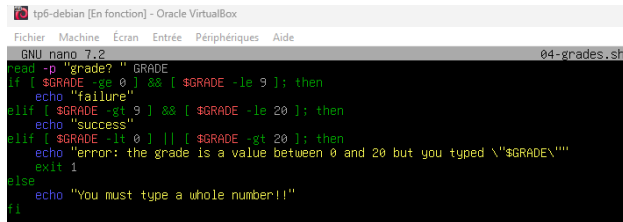
```
1 if CONDITION-1; then
2     STATEMENT-1
3 elif CONDITION-2; then
4     STATEMENT-2
5 else
6     STATEMENT-3
7 fi
```

Pour écrire une condition, nous utilisons la builtin "test" et plus précisément sa forme avec des [] (crochets). On va créer un script 04-grades.sh qui lit au clavier un entier représentant une note sur 20, écrit que c'est un échec si la note est entre 0 et 9 et écrit que c'est un succès si la note est entre 10 et 20. Si la note est en dehors de ces deux plages, le script doit écrire un message d'erreur et retourner 1 au lieu de 0. On tape la commande "sudo nano 04-grades.sh". Dans ce fichier, on écrit les lignes suivantes :

```
1 read -p "grade? " GRADE
2 if [ $GRADE -ge 0 ] && [ $GRADE -le 9 ]; then
3     echo "faillure"
4 elif [ $GRADE -gt 9 ] && [ $GRADE -le 20 ]; then
5     echo "success"
6 elif [ $GRADE -lt 0 ] || [ $GRADE -gt 20 ]; then
7     echo "error : the grade is a value between 0 and 20 but you typed \"$GRADE\""
8     exit 1
9 else
10 echo "You must type a whole number !!"
```

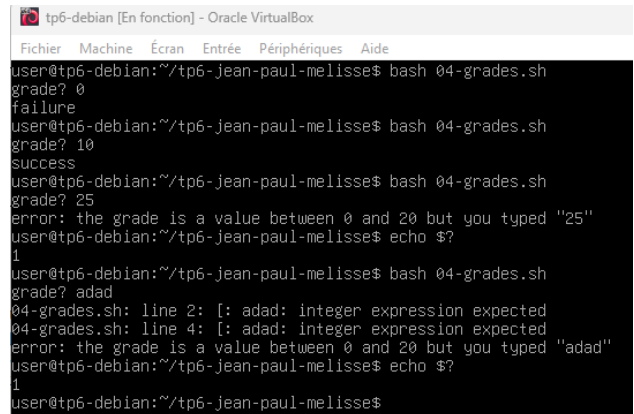
11 fi

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir l'étape précédente sur la Figure 12. On vérifie l'exécution du script en tapant la commande : "bash 04-grades.sh" (voir la Figure 13).



```
GNU nano 7.2 04-grades.sh
read -p "grade? " GRADE
if [ $GRADE -ge 0 ] && [ $GRADE -le 9 ]; then
    echo "failure"
elif [ $GRADE -gt 9 ] && [ $GRADE -le 20 ]; then
    echo "success"
elif [ $GRADE -lt 0 ] || [ $GRADE -gt 20 ]; then
    echo "error: the grade is a value between 0 and 20 but you typed \"$GRADE\""
    exit 1
else
    echo "You must type a whole number!!"
fi
```

Figure 12



```
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 04-grades.sh
grade? 0
failure
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 04-grades.sh
grade? 10
success
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 04-grades.sh
grade? 25
error: the grade is a value between 0 and 20 but you typed "25"
user@tp6-debian:~/tp6-jean-paul-melisse$ echo $?
1
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 04-grades.sh
grade? adad
04-grades.sh: line 2: [: adad: integer expression expected
04-grades.sh: line 4: [: adad: integer expression expected
error: the grade is a value between 0 and 20 but you typed "adad"
user@tp6-debian:~/tp6-jean-paul-melisse$ echo $?
1
user@tp6-debian:~/tp6-jean-paul-melisse$
```

Figure 13

11 Les expression arithmétiques en Bash

Les expressions arithmétiques permettent d'effectuer des calculs (additions, soustractions, . . .). Elles doivent être placées entre $\$(())$ (dollar, double parenthèse ouvrante et double parenthèse fermante), exemple $\$((40 + 2))$ qui sera évalué à 42. On crée un script 05-calculator.sh qui lit au clavier une chaîne de caractères représentant un opérateur, lit un entier représentant une première opérande, lit un entier représentant une seconde opérande et applique l'opérateur sur les opérandes. Si l'opérateur n'est pas supporté, le script doit écrire un message d'erreur et retourner 1 au lieu de 0. Pour cela, on tape la commande "sudo nano 05-calculator.sh". Dans ce fichier, on écrit les lignes suivantes :

```
1 echo "addition"
2 echo "substraction"
3 echo "multiplication"
4 echo "division"
5 echo "remainder"
6 echo "exponentiation"
7 read -p "operator? " OPERATOR
8 read -p "first operand? " FIRST_OPERAND
9 read -p "second operand? " SECOND_OPERAND
10
11 if [ "$OPERATOR" = "addition" ]; then
```



```

12      RESULT=$((FIRST_OPERAND + SECOND_OPERAND))
13  elif [ "$OPERATOR" = "subtraction" ]; then
14      RESULT=$((FIRST_OPERAND - SECOND_OPERAND))
15  elif [ "$OPERATOR" = "multiplication" ]; then
16      RESULT=$((FIRST_OPERAND * SECOND_OPERAND))
17  elif [ "$OPERATOR" = "division" ]; then
18      RESULT=$((FIRST_OPERAND / SECOND_OPERAND))
19  elif [ "$OPERATOR" = "remainder" ]; then
20      RESULT=$((FIRST_OPERAND % SECOND_OPERAND))
21  elif [ "$OPERATOR" = "exponentiation" ]; then
22      RESULT=$((FIRST_OPERAND ** SECOND_OPERAND))
23  else
24      echo "the operator is addition, subtraction, multiplication, division, remainder, or"
25      echo "exponentiation but you typed \"$OPERATOR\""
26      exit 1
27  fi
28
29  echo "result: $RESULT"

```

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir l'étape précédente sur la Figure 14. On vérifie l'exécution du script en tapant la commande : "bash 05-calculator.sh" (voir la Figure 15).

```

tp6-debian [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
GNU nano 2.2.0 05-calculator.sh
echo "addition"
echo "subtraction"
echo "multiplication"
echo "division"
echo "remainder"
echo "exponentiation"
read -p "operator? " OPERATOR
read -p "first operand? " FIRST_OPERAND
read -p "second operand? " SECOND_OPERAND

if [ "$OPERATOR" = "addition" ]; then
    RESULT=$((FIRST_OPERAND + SECOND_OPERAND))
elif [ "$OPERATOR" = "subtraction" ]; then
    RESULT=$((FIRST_OPERAND - SECOND_OPERAND))
elif [ "$OPERATOR" = "multiplication" ]; then
    RESULT=$((FIRST_OPERAND * SECOND_OPERAND))
elif [ "$OPERATOR" = "division" ]; then
    RESULT=$((FIRST_OPERAND / SECOND_OPERAND)) # Here the result gives only whole number
elif [ "$OPERATOR" = "remainder" ]; then
    RESULT=$((FIRST_OPERAND % SECOND_OPERAND))
elif [ "$OPERATOR" = "exponentiation" ]; then
    RESULT=$((FIRST_OPERAND ** SECOND_OPERAND))
else
    echo "error: the operator is addition, subtraction, multiplication, division, remainder, or"
    echo "exponentiation but you typed \"$OPERATOR\""
    exit 1
fi

echo "result: $RESULT"

```

Figure 14

```

tp6-debian [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 05-calculator.sh
addition
subtraction
multiplication
division
remainder
exponentiation
operator? addition
first operand? 4
second operand? 95
result: 99
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 05-calculator.sh
addition
subtraction
multiplication
division
remainder
exponentiation
operator? remainder
first operand? 9
second operand? 4
result: 1
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 05-calculator.sh
addition
subtraction
multiplication
division
remainder
exponentiation
operator? adad
first operand? 4
second operand? 6
error: the operator is addition, subtraction, multiplication, division, remainder, or
exponentiation but you typed "adad"
user@tp6-debian:~/tp6-jean-paul-melisse$ echo $?
1
user@tp6-debian:~/tp6-jean-paul-melisse$ _

```

Figure 15

12 Les itérations en Bash

Bien que les instructions d'un script soient habituellement exécutées de façon séquentielle, il est possible d'en exécuter à l'intérieur d'une structure itérative (boucle).

- Syntaxe :

```

1 while CONDITION-1; do
2     STATEMENT-1
3 done

```

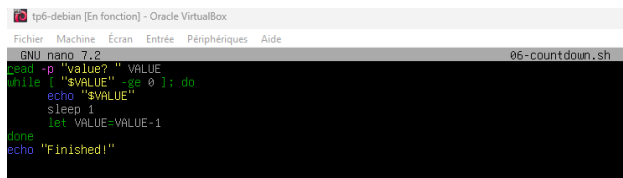
On va donc créer un script 06-countdown.sh qui lit au clavier un entier représentant le début d'un compte à rebours, écrit cette valeur à l'écran, lui retire un, attend une seconde et recommence jusqu'à atteindre zéro. On tape la commande "sudo nano 06-countdown.sh". Dans ce fichier, on écrit les lignes suivantes :

```

1 read -p "value? " VALUE
2 while [ "$VALUE" -ge 0 ]; do
3     echo "$VALUE"
4     sleep 1
5     let VALUE=VALUE-1
6 done
7 echo "Finished!"

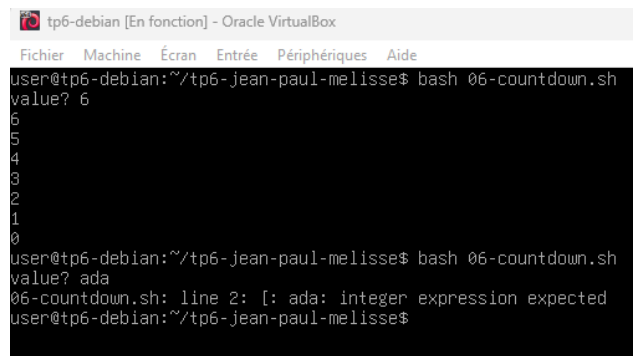
```

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir l'étape précédente sur la Figure 16. On vérifie l'exécution du script en tapant la commande : "bash 06-countdown.sh" (voir la Figure 17).



```
GNU nano 7.2 06-countdown.sh
read -p "value? " VALUE
while [ "$VALUE" -gt 0 ]; do
  echo "$VALUE"
  sleep 1
  let VALUE=VALUE-1
done
echo "Finished!"
```

Figure 16



```
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 06-countdown.sh
value? 6
6
5
4
3
2
1
0
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 06-countdown.sh
value? ada
06-countdown.sh: line 2: [: ada: integer expression expected
user@tp6-debian:~/tp6-jean-paul-melisse$
```

Figure 17

13 Les fonctions en Bash

Les fonctions ont un nom, des paramètres et une valeur de retour. Elles permettent de décomposer le code en plusieurs calculs indépendants. Une fonction peut être définie une fois et appelée autant de fois que nécessaire.

- Syntaxe pour définir une fonction :

```
1 NAME ()
2 {
3     STATEMENT-1
4 }
5 function NAME
6 {
7     STATEMENT-2
8 }
```

- Syntaxe pour appeler une fonction :

```
1 NAME PARAMETER-1
```

Les paramètres peuvent être obtenus grâce à \$1 pour le premier paramètre et ainsi de suite pour les autres paramètres. La valeur de retour peut être obtenue grâce à \$? (dollar et point d'interrogation). On va créer un script "07-even.sh" qui définit la fonction "even" à un paramètre "NUMBER" et qui retourne 0 si "NUMBER" est pair, sinon 1. Ce script lit au clavier un entier, appelle la fonction "even" en lui passant en paramètre l'entier précédemment lu au clavier et écrit à l'écran le résultat obtenu. Pour cela, on tape la commande "sudo nano 07-even.sh". Dans ce fichier, on écrit les lignes suivantes :

```
1 read -p "number? " NUMBER
2 function even {
3     if [ $((("$1"%2)) -eq 0 )]; then
4         echo "the number \"$1\" is even"
```

```

5          RESULT=0

6      else

7          echo "the number \"\$1 \" is odd"

8          RESULT=1

9      fi

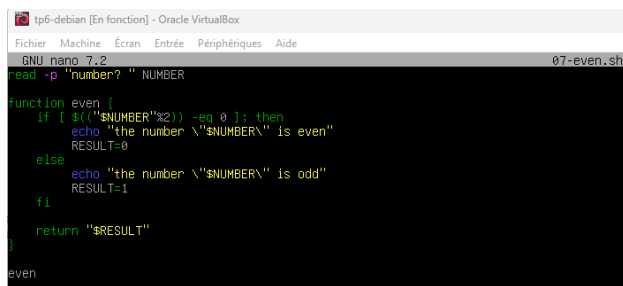
10     return "$RESULT"

11 }

12 even "$NUMBER"

```

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir l'étape précédente sur la Figure 18. On vérifie l'exécution du script en tapant la commande : "bash 07-even.sh" (voir la Figure 19).



```

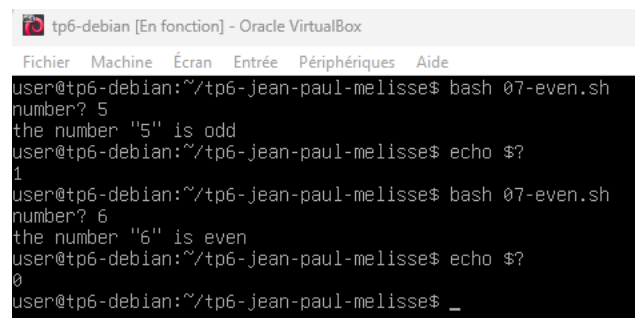
tp6-debian [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
GNU nano 7.2 07-even.sh
read -p "number? " NUMBER

function even {
    if [ $((${NUMBER%*}) -eq 0) ]; then
        echo "the number \"${NUMBER}\" is even"
        RESULT=0
    else
        echo "the number \"${NUMBER}\" is odd"
        RESULT=1
    fi
    return "$RESULT"
}

even

```

Figure 18



```

tp6-debian [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 07-even.sh
number? 5
the number "5" is odd
user@tp6-debian:~/tp6-jean-paul-melisse$ echo $?
1
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 07-even.sh
number? 6
the number "6" is even
user@tp6-debian:~/tp6-jean-paul-melisse$ echo $?
0
user@tp6-debian:~/tp6-jean-paul-melisse$ _

```

Figure 19

14 Les paramètres en Bash

Des paramètres peuvent être passés aux scripts en ligne de commande de la même manière que des paramètres peuvent être passés aux fonctions. Les paramètres peuvent être obtenus de la même manière qu'avec les fonctions. On crée un script 08-parameters.sh qui écrit à l'écran le nom du script (paramètre zéro) et les paramètres passés. On tape la commande "sudo nano 08-parameters.sh". Dans ce fichier, on écrit les lignes suivantes :

```

1 echo "program: $0"

2 while [ "$#" -gt 0 ]; do

3     echo "parameter: $1"

4     shift

5 done

```

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir l'étape précédente sur la Figure 20. On vérifie l'exécution du script en tapant la commande : "bash 08-parameters.sh" (voir la Figure 21).

```

tp6-debian [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
GNU nano 7.2 08-parameters.sh
echo "program: $0"
while [ "$#" -gt 0 ]; do
  echo "parameter: $1"
  shift
done

```

Figure 20

```

tp6-debian [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 08-parameters.sh
program: 08-parameters.sh
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 08-parameters.sh ad
program: 08-parameters.sh
parameter: ad
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 08-parameters.sh ad zdza zadaz
program: 08-parameters.sh
parameter: ad
parameter: zdza
parameter: zadaz
user@tp6-debian:~/tp6-jean-paul-melisse$

```

Figure 21

15 Obtenir des informations système en Bash

Dans cette partie du TP, on va créer un script "09-sysinfo.sh" qui écrit à l'écran des informations sur le système. Pour cela, on tape la commande "sudo nano 09-sysinfo.sh". Dans ce fichier, on écrit les lignes suivantes :

- 1 echo "System information:"
- 2 CPU=\$(grep "model name" /proc/cpuinfo | sed 's/\^.*: //')
- 3 echo "* CPU: \$CPU"
- 4 echo "* Cores: \$(grep -c 'core id' /proc/cpuinfo | sort -u | wc -l)"
- 5 echo "* Threads: \$(grep -c '^processor' /proc/cpuinfo)"
- 6 RAM=\$(grep "MemTotal" /proc/meminfo | sed -E 's/^[^0-9]//g')
- 7 RAM_GB=\$((RAM/1024/1024))
- 8 echo "* RAM: \$RAM_GB GB"
- 9 ROM=\$(grep 'sda\$' /proc/partitions | sed 's/^[]*[0-9]*[]*[0-9]*[]*\([0-9]*\) .*\/1/')
- 10 ROM_GB=\$((ROM/1024/1024))
- 11 echo "* ROM: \$ROM_GB GB"

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir l'étape précédente sur la Figure 22. On vérifie l'exécution du script en tapant la commande : "bash 09-sysinfo.sh" (voir la Figure 23).

```

tp6-debian [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
GNU nano 7.2 09-sysinfo.sh
echo "System information:"

CPU=$(grep "model name" /proc/cpuinfo | sed 's/\^.*: //' )
echo "* CPU: $CPU"

echo "* Cores: $(grep -c 'core id' /proc/cpuinfo | sort -u | wc -l)"
echo "* Threads: $(grep -c '^processor' /proc/cpuinfo)"

RAM=$(grep "MemTotal" /proc/meminfo | sed -E 's/^[^0-9]//g')
echo "$RAM"

RAM_GB=$((RAM/1024/1024))
echo "* RAM: $RAM_GB GB"

ROM=$(grep 'sda$' /proc/partitions | sed 's/^[ ]*[0-9]*[ ]*[0-9]*[ ]*\([0-9]*\) .*\/1/')
ROM_GB=$((ROM/1024/1024))
echo "* ROM: $ROM_GB GB"

```

Figure 22

```

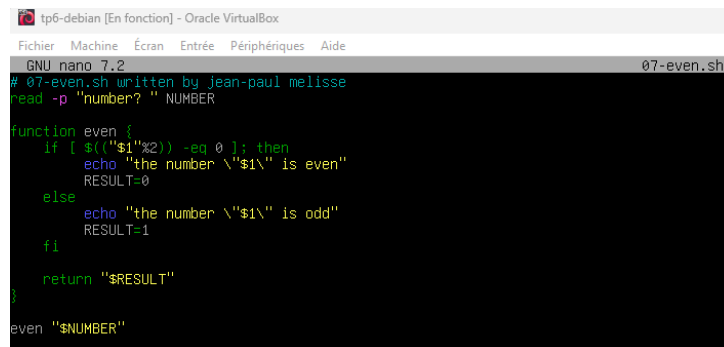
tp6-debian [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
user@tp6-debian:~/tp6-jean-paul-melisse$ bash 09-sysinfo.sh
System information:
* CPU: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz
* Cores: 1
* Threads: 1
2014488
* RAM: 1 GB
* ROM: 20 GB
user@tp6-debian:~/tp6-jean-paul-melisse$

```

Figure 23

16 Les commentaires en Bash

Pour commenter un code, il suffit de précéder le texte par # (croisillon), exemple : # this is my comment. On va ajouter un commentaire à la première ligne de tous les scripts qu'on a créés afin d'y indiquer "# <fichier> written by jean-paul melisse" (avec <fichier> à remplacer par le nom du fichier. Seuls les caractères suivants sont autorisés pour ce commentaire : 0-9 (numériques), a-z (minuscules), - (tiret) et .(point). On peut voir un exemple sur la Figure 24.



```
tp6-debian [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
GNU nano 7.2 07-even.sh
# 07-even.sh written by jean-paul melisse
read -p "number? " NUMBER

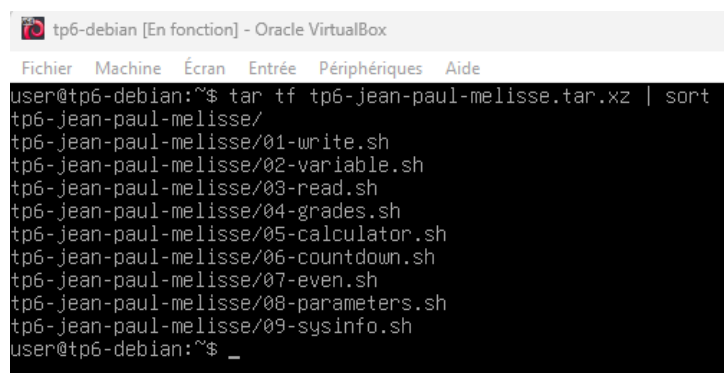
function even {
    if [ $(("$1"%2)) -eq 0 ]; then
        echo "the number \"$1\" is even"
        RESULT=0
    else
        echo "the number \"$1\" is odd"
        RESULT=1
    fi
    return "$RESULT"
}

even "$NUMBER"
```

Figure 24

17 Création de l'archive et son exportation

Pour cette dernière partie de ce TP, on va archiver le dossier "tp6-jean-paul-melisse" et l'exporter sur la machine physique. Pour commencer, on revient sur le dossier parent (/home/user/) avec la commande : "cd ..". Puis, on crée l'archive "tp6-jean-paul-melisse.tar.xz" en tapant la commande "sudo tar -caf tp6-jean-paul-melisse.tar.xz tp6-jean-paul-melisse/". [Attention : il est possible que les utilitaire xz ne soient pas installés. Il faut les installer avec la commande : "sudo apt install xz-utils".] On peut vérifier le contenu de l'archive en tapant la commande : "tar tf tp6-jean-paul-melisse.tar.xz | sort" (Voir la Figure 25).



```
tp6-debian [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
user@tp6-debian:~$ tar tf tp6-jean-paul-melisse.tar.xz | sort
tp6-jean-paul-melisse/
tp6-jean-paul-melisse/01-write.sh
tp6-jean-paul-melisse/02-variable.sh
tp6-jean-paul-melisse/03-read.sh
tp6-jean-paul-melisse/04-grades.sh
tp6-jean-paul-melisse/05-calculator.sh
tp6-jean-paul-melisse/06-countdown.sh
tp6-jean-paul-melisse/07-even.sh
tp6-jean-paul-melisse/08-parameters.sh
tp6-jean-paul-melisse/09-sysinfo.sh
user@tp6-debian:~$ _
```

Figure 25

La suite est de créer un dossier partagé. On retourne sur la machine physique et on crée le dossier "share" dans le chemin : "C:\Users\Jean-Paul\Desktop\tp6\". Puis, on va sur VirtualBox et on clique sur "tp6-debian" et "Configuration". Puis on clique sur "Shared Folders" et sur "Ajouter un nouveau dossier partagé". On met dans le chemin du dossier : "C:\Users\Jean-Paul\Desktop\tp6\share". On met le nom du dossier : "share" et on clique sur "Ok". On peut voir sur la Figure 26 que le dossier "share" a bien été

partagé avec la machine virtuelle "tp6-debian".

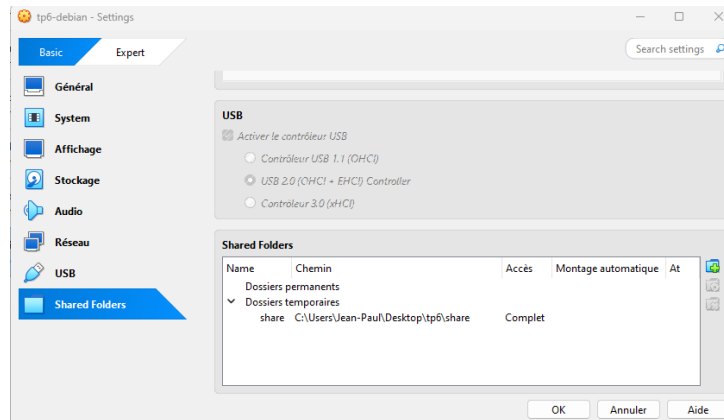


Figure 26

On retourne sur la machine "tp6-debian". On tape la commande "mkdir ~/mnt" pour créer un nouveau dossier dans le dossier personnel de l'utilisateur "user". On tape la commande : "sudo mount -t vboxsf share ~/mnt" pour lier le dossier "mnt" qui se trouve dans la machine "tp6-debian" au dossier partagé "share" venant de la machine physique. Puis, on copie l'archive "tp6-jean-paul-melisse.tar.xz" dans le dossier "mnt" en tapant la commande "sudo cp tp6-jean-paul-melisse.tar.xz mnt/". On retourne sur la machine physique et on regarde le dossier "share". On voit, comme dans la Figure 27, que l'archive est bien sur la machine physique. On peut l'exporter pour une utilisation ultérieure.

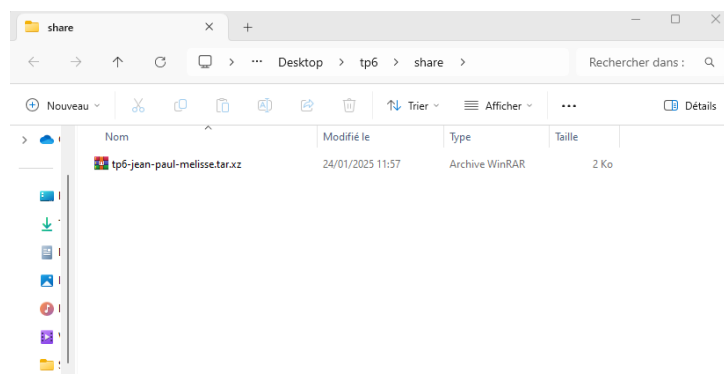


Figure 27