

**Institut Universitaire des Sciences (IUS)**  
**FACULTÉ DES SCIENCES ET DES TECHNOLOGIES (FST)**

**RAPPORT**  
**SUR LE PROJET**

*Cours : Réseaux 1*

Étudiant : Wendy Colas  
Niveau : L3

2024

- Créez un script avec un menu interactif permettant de choisir l'opération à effectuer (racine carrée, puissance, etc.).

The screenshot shows a desktop environment with a file browser window open. In the file browser, there is a file named "Untitled-1" which contains a bash script. The script is a menu-based calculator program. It includes options for square root, power, logarithm, and quit. The script uses the bc command for calculations.

```

#!/bin/bash
while true; do
    echo "==== Menu de Calcul Avancé ==="
    echo "1. Racine carrée"
    echo "2. Puissance"
    echo "3. Logarithme"
    echo "4. Quitter"
    read -p "Choisissez une option : " choix
    case $choix in
        1)
            read -p "Entrez un nombre : " num
            echo "Racine carrée de $num : $(echo "scale=2; sqrt($num)" | bc)"
            ;;
        2)
            read -p "Entrez un nombre : " base
            read -p "Entrez la puissance : " exp
            echo "$base ^ $exp = $(echo "scale=2; $base^$exp" | bc)"
            ;;
        3)
            read -p "Entrez un nombre : " num
            echo "Logarithme naturel de $num : $(echo "scale=2; l($num)" | bc -l)"
            ;;
        4)
            echo "Au revoir !"
            exit 0
            ;;
        *)
            echo "Option invalide."
            ;;
    esac
done

```

The screenshot shows a terminal window running on a virtual machine. The user has navigated to the "/Documents" directory and run the script "calcul1.sh". The terminal output shows the execution of the script, starting with the menu options and prompting the user to enter a number for the square root calculation.

```

wendy@wendy-VirtualBox:~/Documents$ ls
calcul1.sh  essai.sh  perle.sh  pop.sh      projet.sh
calcul.sh   jean.sql  pierre.sh  Presentation.odt  projetw.sh
copie.sh    junior.sh  pierre.sql  presentation.pdf
wendy@wendy-VirtualBox:~/Documents$ chmod +x calcul1.sh
wendy@wendy-VirtualBox:~/Documents$ ./calcul1.sh
==== Menu de Calcul Avancé ===
1. Racine carrée
2. Puissance
3. Logarithme
4. Quitter
Choisissez une option : 1
Entrez un nombre :

```

```
Iments > $ calcul1.sh
wendy@wendy-VirtualBox:~/Documents
```

```
wendy@wendy-VirtualBox:~$ cd Documents
wendy@wendy-VirtualBox:~/Documents$ ls
calcul1.sh  essai.sh  perle.sh  pop.sh      projet.sh
calcul.sh   jean.sql  pierre.sh  Presentation.odt  projetw.sh
copie.sh    junior.sh pierre.sql presentation.pdf
wendy@wendy-VirtualBox:~/Documents$ chmod +x calcul1.sh
wendy@wendy-VirtualBox:~/Documents$ ./calcul1.sh
== Menu de Calcul Avancé ==
1. Racine carrée
2. Puissance
3. Logarithme
4. Quitter
Choisissez une option : 1
Entrez un nombre : 12
Racine carrée de 12 : 3.46
== Menu de Calcul Avancé ==
1. Racine carrée
2. Puissance
3. Logarithme
4. Quitter
Choisissez une option :
```

2. Créez un script Bash qui génère un graphique représentant les surfaces de figures géométriques simples (carré, trapèze, parallélogramme) à l'aide de gnuplot. Ce script permet de calculer les surfaces de ces figures et de les afficher sous forme de graphique.

second [En fonction] - Oracle VM VirtualBox

Fichier Machine Écran Entrée Périphériques Aide

Activités Visual Studio Code mas 3 09:31

Annuler Nom surface.sh Enregistrer

Dossier personnel Nom Taille Type Modifié

Bureau

Documents 10.3 ko Programme 19 fev

Images 3.5 ko Programme 14 fev

Musique 10.3 ko Programme 18 fev

Téléchargements 4.2 ko Programme 15 fev

Vidéos 3.5 ko Programme van

Autres emplacements 3.3 ko Programme 13 fev

projet.sh 9.0 ko Programme 17 fev

projetw.sh 3.7 ko Programme 14 fev

pop.sh 630 octets Programme 09:26

pierre.sh 623 octets Programme 13 fev

perle.sh

junior.sh

essai.sh

copie.sh

calcul1.sh

calcul.sh

Taille Type Modifié

10.3 ko Programme 19 fev

3.5 ko Programme 14 fev

10.3 ko Programme 18 fev

4.2 ko Programme 15 fev

3.5 ko Programme van

3.3 ko Programme 13 fev

9.0 ko Programme 17 fev

3.7 ko Programme 14 fev

630 octets Programme 09:26

623 octets Programme 13 fev

Right Ctrl

Visual Studio Code mas 3 09:50

Fichier Edition Sélection ... ← → 🔍 wendy

\$ surface.sh

```

Documents > $ surface.sh
1  #!/bin/bash
2
3  # Fonction pour afficher le menu
4  afficher_menu() {
5      echo "Choisissez une option :"
6      echo "1. Calculer la surface d'un carré"
7      echo "2. Calculer la surface d'un losange"
8      echo "3. Calculer la surface d'un triangle"
9      echo "4. Générer un graphique des surfaces"
10     echo "5. Quitter"
11 }
12
13 # Variables globales pour stocker les surfaces
14 surface_carre=0
15 surface_losange=0
16 surface_triangle=0
17
18 # Fonction pour calculer la surface d'un carré
19 calculer_carre() {
20     read -p "Entrez la longueur du côté du carré : " cote
21     if [[ "$cote" =~ ^[0-9]+(\.[0-9]+)?$ ]] && $(echo "$cote > 0" | bc -eq 1); then
22         surface_carre=$(echo "$cote * $cote" | bc -l)
23         echo "La surface du carré est : $surface_carre unités carrées."
24     else
25         echo "Entrée invalide. Veuillez entrer un nombre positif."
26     fi
27 }
```

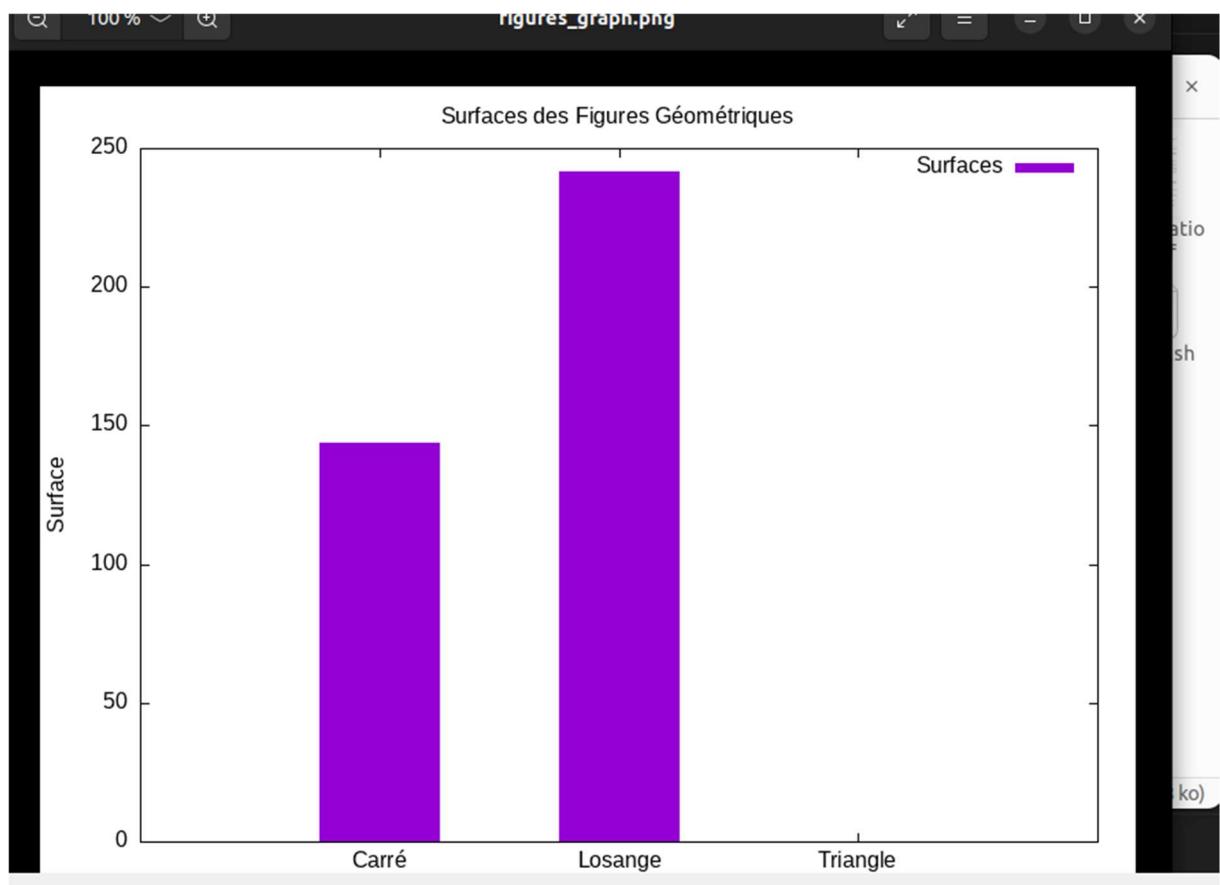
A screenshot of the Visual Studio Code interface. The title bar says "Activités" and "Visual Studio Code". The status bar shows "mas 3 09:50". The main area displays a shell script named "surface.sh". The code includes functions for generating graphs of shapes and a main loop for calculating and displaying surface areas.

```
$ surface.sh
Documents > $ surface.sh
54     generer_graphique() {
55         echo -e "Losange\tsurface_losange" >> graph.gnuplot
56         echo -e "Triangle\t$surface_triangle" >> graph.gnuplot
57         echo "e" >> graph.gnuplot
58
59         gnuplot graph.gnuplot
60         rm graph.gnuplot
61         echo "Le graphique a été enregistré sous le nom 'figures_graph.png' ."
62     }
63
64     # Boucle principale
65     while true; do
66         afficher_menu
67         read -p "Votre choix : " choix
68         case $choix in
69             1)
70                 calculer_carre
71                 ;;
72             2)
73                 calculer_losange
74                 ;;
75             3)
76                 calculer_triangle
77                 ;;
78             4)
79                 generer_graphique
80                 ;;
81             5)
82                 ;;
83         esac
84     done
```

A screenshot of a terminal window titled "wendy@wendy-VirtualBox: ~/Documents". The terminal displays the execution of the "surface.sh" script. It starts with a menu of options (1-5), then calculates the area of a square with side length 12, and finally generates a graph of the calculated surfaces.

```
wendy@wendy-VirtualBox: ~/Documents
1. Calculer la surface d'un carré
2. Calculer la surface d'un losange
3. Calculer la surface d'un triangle
4. Générer un graphique des surfaces
5. Quitter
Votre choix : 1
Entrez la longueur du côté du carré : 12
La surface du carré est : 144 unités carrées.
Choisissez une option :
1. Calculer la surface d'un carré
2. Calculer la surface d'un losange
3. Calculer la surface d'un triangle
4. Générer un graphique des surfaces
5. Quitter
Votre choix : 4
Génération de graphiques pour les surfaces calculées...
Le graphique a été enregistré sous le nom 'figures_graph.png'.
Choisissez une option :
1. Calculer la surface d'un carré
2. Calculer la surface d'un losange
3. Calculer la surface d'un triangle
4. Générer un graphique des surfaces
5. Quitter
Votre choix : 
```



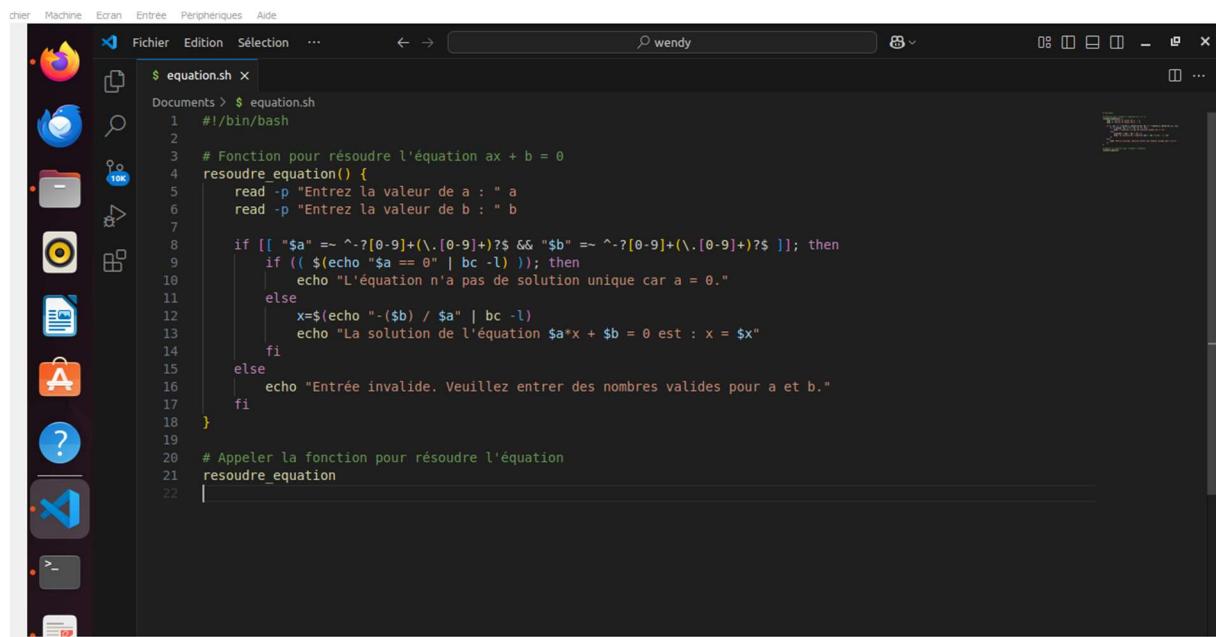


3. Créez un script Bash pour comparer deux nombres en utilisant ces opérateurs et affichez un message à la fin du script.

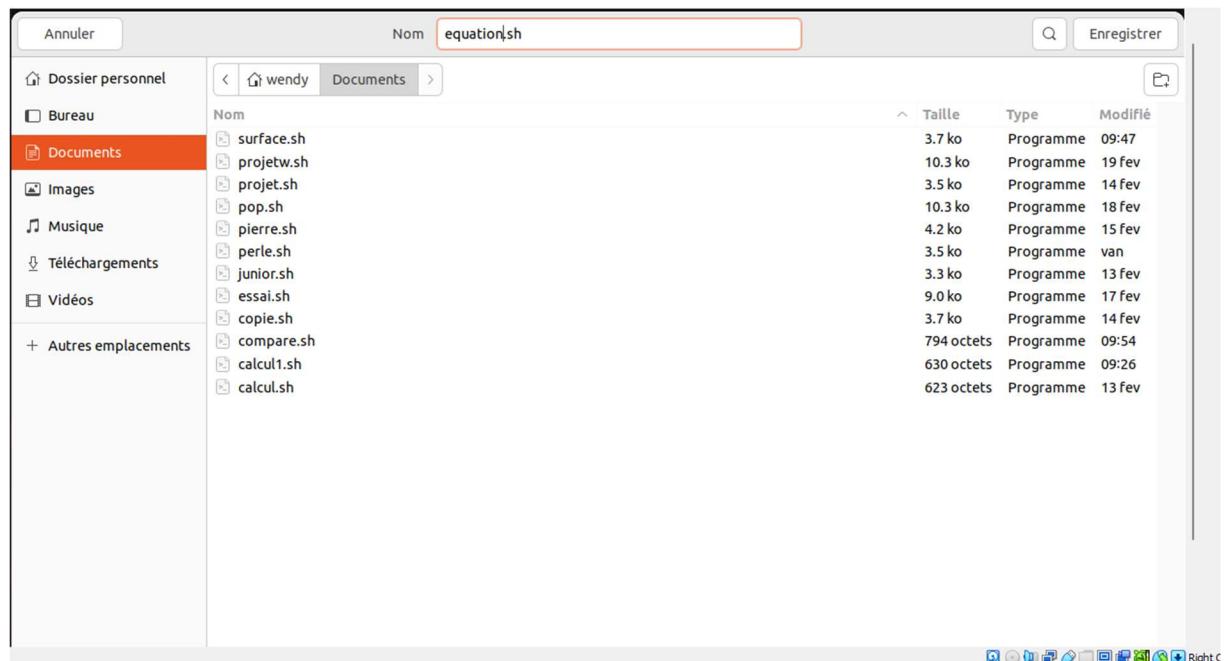
```
Fichier Edition Sélection ... ← → 🔍 wendy
$ #!/bin/bash Untitled-1 ●
1  #!/bin/bash
2  # Demander à l'utilisateur d'entrer deux nombres
3  read -p "Entrez le premier nombre : " num1
4  read -p "Entrez le deuxième nombre : " num2
5  # Comparer les nombres en utilisant différents opérateurs
6  if [ "$num1" -eq "$num2" ]; then
7      echo "$num1 est égal à $num2."
8  else
9      echo "$num1 n'est pas égal à $num2."
10 fi
11 if [ "$num1" -ne "$num2" ]; then
12     echo "$num1 est différent de $num2."
13 fi
14 if [ "$num1" -lt "$num2" ]; then
15     echo "$num1 est inférieur à $num2."
16 elif [ "$num1" -gt "$num2" ]; then
17     echo "$num1 est supérieur à $num2."
18 fi
19 if [ "$num1" -le "$num2" ]; then
20     echo "$num1 est inférieur ou égal à $num2."
21 else
22     echo "$num1 est supérieur à $num2."
23 fi
24 if [ "$num1" -ge "$num2" ]; then
25     echo "$num1 est supérieur ou égal à $num2."else
26     echo "$num1 est inférieur à $num2."
27 fi
```

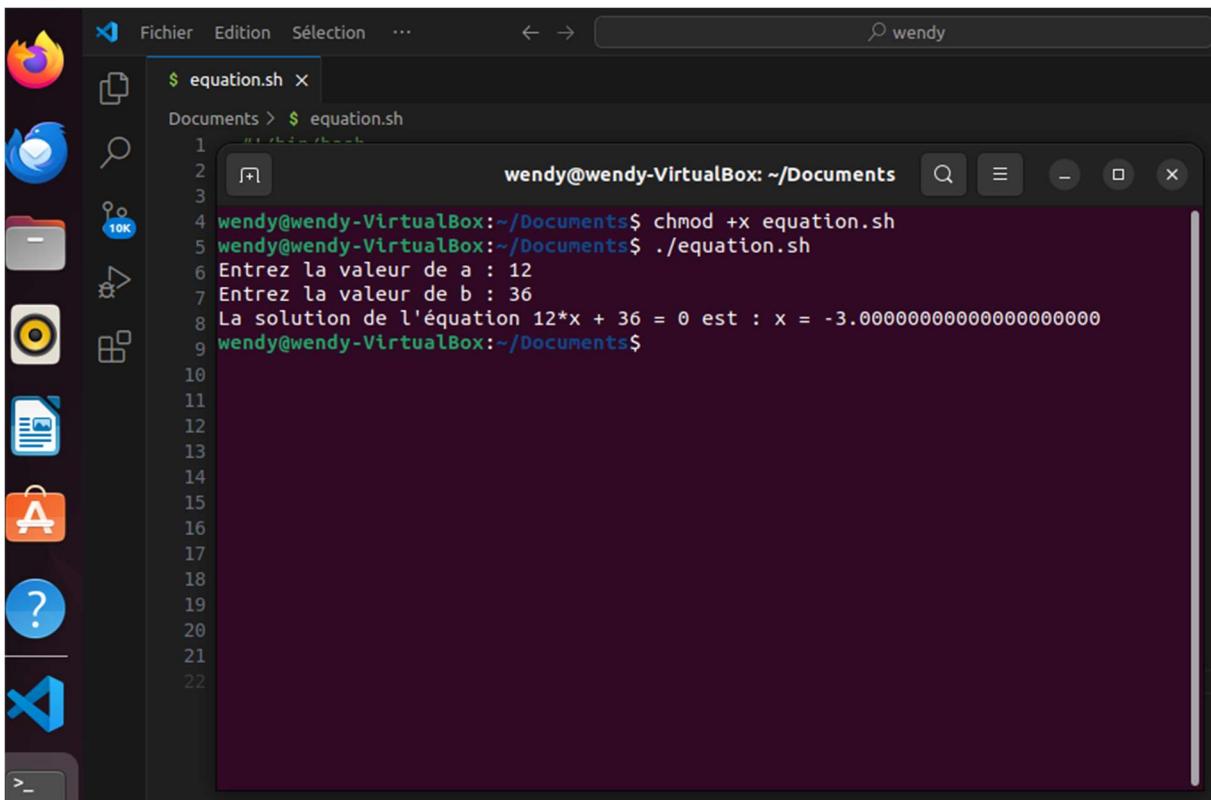
```
Documents > $ compare.sh x
Documents > $ compare.sh
1  Wendy@wendy-VirtualBox:~/Documents$ ls
2  calcul1.sh  essai.sh          perle.sh    Presentation.odt  surface.sh
3  calcul.sh   figures_graph.png pierre.sh   presentation.pdf
4  compare.sh  jean.sql         pierre.sql  projet.sh
5  copie.sh    junior.sh        pop.sh     projetw.sh
6  Wendy@wendy-VirtualBox:~/Documents$ chmod +x compare.sh
7  Wendy@wendy-VirtualBox:~/Documents$ ./compare.sh
8
9  Entrez le premier nombre : 13
10 Entrez le deuxième nombre : 32
11 13 n'est pas égal à 32.
12 13 est différent de 32.
13 13 est inférieur à 32.
14 13 est inférieur ou égal à 32.
15 Wendy@wendy-VirtualBox:~/Documents$
```

4. Créez un script Bash pour résoudre une équation sous la forme  $ax + b = 0$ .



```
$ equation.sh
Documents > $ equation.sh
1  #!/bin/bash
2
3  # Fonction pour résoudre l'équation ax + b = 0
4  resoudre_equation() {
5      read -p "Entrez la valeur de a : " a
6      read -p "Entrez la valeur de b : " b
7
8      if [[ "$a" =~ ^-[0-9]+(\.[0-9]+)?$ && "$b" =~ ^-[0-9]+(\.[0-9]+)?$ ]]; then
9          if (( $(echo "$a == 0" | bc -l) )); then
10              echo "L'équation n'a pas de solution unique car a = 0."
11          else
12              x=$(echo "-($b) / $a" | bc -l)
13              echo "La solution de l'équation $a*x + $b = 0 est : x = $x"
14          fi
15      else
16          echo "Entrée invalide. Veuillez entrer des nombres valides pour a et b."
17      fi
18  }
19
20  # Appeler la fonction pour résoudre l'équation
21  resoudre_equation
22
```





```
$ equation.sh
Documents > $ equation.sh
1
2
3
4 wendy@wendy-VirtualBox:~/Documents$ chmod +x equation.sh
wendy@wendy-VirtualBox:~/Documents$ ./equation.sh
5 Entrez la valeur de a : 12
6 Entrez la valeur de b : 36
7 La solution de l'équation 12*x + 36 = 0 est : x = -3.00000000000000000000000000000000
wendy@wendy-VirtualBox:~/Documents$
```

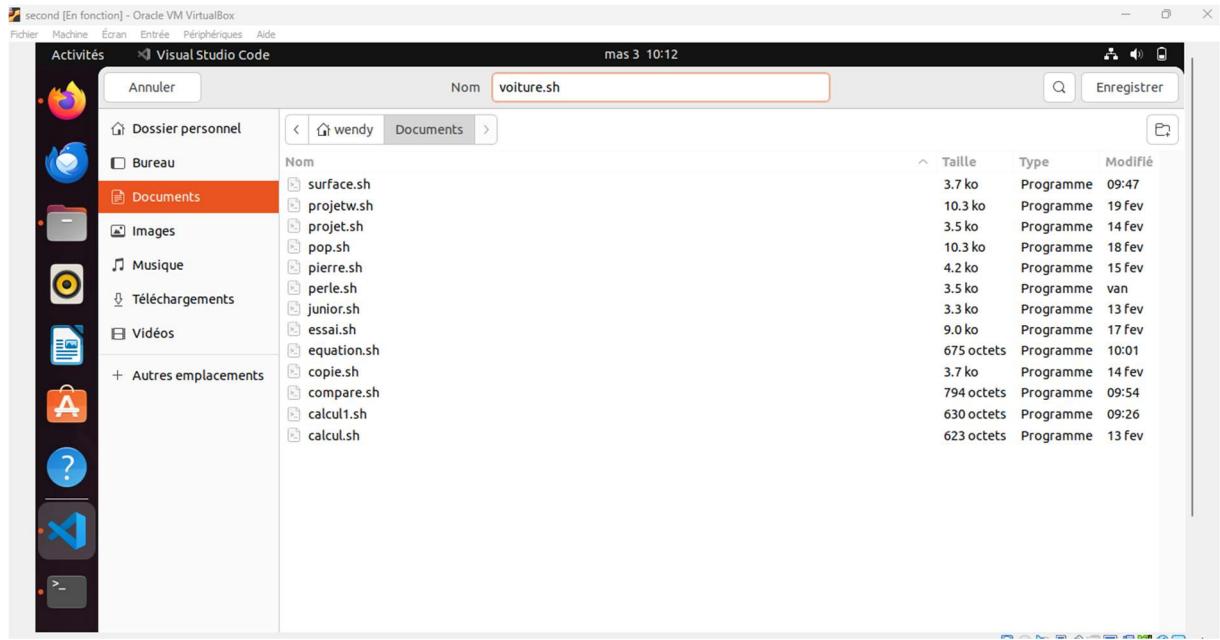
5. Créez une base de données contenant des informations sur les voitures, y compris les marques, les numéros de plaques d'immatriculation et les informations sur les propriétaires. Vous pouvez utiliser SQL, PostgreSQL ou SQLite.

Visual Studio Code window showing a bash script named Untitled-1. The script handles car management operations like creation, update, and deletion of cars and their owners.

```
$ #!/bin/bash Untitled-1
1 #!/bin/bash
2 # paramètres de connexion mysql
3 DB_NAME="gestion_voitures"
4 TABLE_VOITURES="voitures"
5 TABLE_PROPRIETAIRES="proprietaires"
6 DB_USER="root"
7 DB_PASS=""
8
9 # pour afficher le menu principal
10 menu() {
11     echo "Gestion des voitures"
12     echo "1. Créer la base de données et les tables"
13     echo "2. Ajouter une nouvelle voiture"
14     echo "3. Ajouter un nouveau propriétaire"
15     echo "4. Mettre à jour les informations d'une voiture"
16     echo "5. Supprimer une voiture"
17     echo "6. Afficher la liste des voitures"
18     echo "7. Afficher la liste des propriétaires"
19     echo "8. Quitter"
20 }
21
22 # pour créer la base de données et les tables
23 creer_db_et_tables() {
24     mysql -u $DB_USER -p$DB_PASS -e "CREATE DATABASE IF NOT EXISTS $DB_NAME;"
25     mysql -u $DB_USER -p$DB_PASS -e "
26         USE $DB_NAME;
27         CREATE TABLE IF NOT EXISTS $TABLE_PROPRIETAIRES (
28             id INT AUTO_INCREMENT PRIMARY KEY,
29             nom VARCHAR(100) NOT NULL,
30             adresse VARCHAR(255) NOT NULL
31     ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
32     mysql -u $DB_USER -p$DB_PASS -e "
33         USE $DB_NAME;
34         CREATE TABLE IF NOT EXISTS $TABLE_VOITURES (
35             id INT AUTO_INCREMENT PRIMARY KEY,
36             marque VARCHAR(50) NOT NULL,
37             immatriculation VARCHAR(100) NOT NULL,
38             proprietaire_id INT,
39             FOREIGN KEY (proprietaire_id) REFERENCES $TABLE_PROPRIETAIRES(id)
40     ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
41 }
42
43 # pour ajouter une nouvelle voiture
44 ajouter_voiture() {
45     echo "Entrez la marque :"
46     read marque
47     echo "Entrez l'immatriculation :"
48     read immatriculation
49     echo "Entrez l'ID du propriétaire :"
50     read proprietaire_id
51     if [ ! -z "$marque" ]; then
52         mysql -u $DB_USER -p$DB_PASS -e "USE $DB_NAME; INSERT INTO $TABLE_VOITURES (marque, immatriculation, proprietaire_id) VALUES ('$marque', '$immatriculation', '$proprietaire_id');"
53     fi
54 }
55
56 # pour mettre à jour une voiture existante
57 mettre_a_jour_voiture() {
58     echo "Entrez l'ID de la voiture à mettre à jour :"
59     read id
60     echo "Entrez la nouvelle marque :"
61     read marque
62     echo "Entrez le nouveau numéro de plaque d'immatriculation :"
63     read immatriculation
64     echo "Entrez le nouvel ID du propriétaire :"
65     read proprietaire_id
66     if [ ! -z "$marque" ]; then
67         mysql -u $DB_USER -p$DB_PASS -e "USE $DB_NAME; UPDATE $TABLE_VOITURES SET marque='$marque' WHERE id=$id;"
```

Continuation of the bash script in Visual Studio Code. It includes functions for updating and deleting cars.

```
68     if [ ! -z "$immatriculation" ]; then
69         mysql -u $DB_USER -p$DB_PASS -e "USE $DB_NAME; UPDATE $TABLE_VOITURES SET immatriculation='$immatriculation' WHERE id=$id;"
```



```
$ voiture.sh x
Documents > $ voiture.sh
66
67
68
69 La commande « lls » n'a pas été trouvée, voulez-vous dire :
70   commande « rls » du snap rustup (1.27.1)
71   commande « fls » du deb sleuthkit (4.11.1+dfsg-1)
72   commande « hls » du deb hfsutils (3.2.6-15build2)
73   commande « les » du deb atm-tools (1:2.5.1-4build2)
74   commande « ls » du deb coreutils (8.32-4.1ubuntu1.2)
75   commande « lvs » du deb lvm2 (2.03.11-2.1ubuntu4)
76   commande « lrs » du deb lrslib (0.71a-1)
77   commande « llc » du deb llvm (1:14.0-55~exp2)
78   commande « ols » du deb speech-tools (1:2.5.0-12)
79   commande « jls » du deb sleuthkit (4.11.1+dfsg-1)
80   commande « ils » du deb sleuthkit (4.11.1+dfsg-1)
81   commande « llt » du deb storebackup (3.2.1-2)
82   commande « lld » du deb lld (1:14.0-55~exp2)
83   commande « als » du deb atool (0.39.0-11)
84   commande « lli » du deb llvm-runtime (1:14.0-55~exp2)
85 Voir « snap info <nomdusnap> » pour des versions supplémentaires.
86 wendy@wendy-VirtualBox:~/Documents$ ls
87 calcul1.sh equation.sh junior.sh pop.sh projetw.sh
88 calcul.sh essai.sh perle.sh Presentation.odt surface.sh
89 compare.sh figures_graph.png pierre.sh presentation.pdf voiture.sh
90 copie.sh jean.sql pierre.sql projet.sh
91 wendy@wendy-VirtualBox:~/Documents$ chmod +x voiture.sh
92 wendy@wendy-VirtualBox:~/Documents$ ./voiture.sh
```

```
. Mettre à jour les informations d'une voiture
. Supprimer une voiture
. Afficher la liste des voitures
. Afficher la liste des propriétaires
. Quitter
Votre choix : 2
Entrez la marque de la voiture :
toyota
Entrez le numéro de plaque d'immatriculation :
aa-1278
Entrez l'ID du propriétaire :

Entrez password:
Voiture ajoutée avec succès.
Gestion des voitures
. Créer la base de données et les tables
. Ajouter une nouvelle voiture
. Ajouter un nouveau propriétaire
. Mettre à jour les informations d'une voiture
. Supprimer une voiture
. Afficher la liste des voitures
. Afficher la liste des propriétaires
. Quitter
Votre choix :
Supprimer_voiture() {
    echo "Entrez l'ID de la voiture à supprimer :"
    read id
```

second [En fonction] - Oracle VM VirtualBox

Fichier Machine Écran Entrée Périphériques Aide

Activités Terminal mas 3 10:16

wendy

\$ voiture.sh x

Documents > \$ voiture.sh

66 2. Ajouter une nouvelle voiture
67 3. Ajouter un nouveau propriétaire
68 4. Mettre à jour les informations d'une voiture
69 5. Supprimer une voiture
70 6. Afficher la liste des voitures
71 7. Afficher la liste des propriétaires
72 8. Quitter
73 Votre choix : 6
74 Enter password:
75 +-----+
76 | id | marque | immatriculation | proprietaire\_id |
77 +-----+
78 | 2 | toyota | aa-1278 | 1 |
79 +-----+
80
81
82 Gestion des voitures
83 1. Créer la base de données et les tables
84 2. Ajouter une nouvelle voiture
85 3. Ajouter un nouveau propriétaire
86 4. Mettre à jour les informations d'une voiture
87 5. Supprimer une voiture
88 6. Afficher la liste des voitures
89 7. Afficher la liste des propriétaires
90 8. Quitter
91 Supprimer\_voiture() {
92 echo "Entrez l'ID de la voiture à supprimer :"

```
$ voiture.sh
Documents > $ voiture.sh
66
67
68
69 2. Ajouter une nouvelle voiture
70 3. Ajouter un nouveau propriétaire
71 4. Mettre à jour les informations d'une voiture
72 5. Supprimer une voiture
73 6. Afficher la liste des voitures
74 7. Afficher la liste des propriétaires
75 8. Quitter
76 Votre choix : 7
77 Enter password:
78 +---+-----+-----+
79 | id | nom   | adresse | telephone |
80 +---+-----+-----+
81 | 1 | wendy | 12,morne | 37693573 |
82 +---+-----+-----+
83 Gestion des voitures
84 1. Créer la base de données et les tables
85 2. Ajouter une nouvelle voiture
86 3. Ajouter un nouveau propriétaire
87 4. Mettre à jour les informations d'une voiture
88 5. Supprimer une voiture
89 6. Afficher la liste des voitures
90 7. Afficher la liste des propriétaires
91 8. Quitter
92 Votre choix :
```

## 6. Exécuter ces commandes.

```
wendy@wendy-VirtualBox:~$ cd ..  
wendy@wendy-VirtualBox:~$ df -h  
ys. de fichiers Taille Utilisé Dispo Uti% Monté sur  
mpfs          197M    1,6M  196M   1% /run  
dev/sda3       39G    16G   22G  42% /  
mpfs          985M      0  985M   0% /dev/shm  
mpfs          5,0M    4,0K  5,0M   1% /run/lock  
dev/sda2       512M    6,1M  506M   2% /boot/efi  
mpfs          197M   120K  197M   1% /run/user/1000  
wendy@wendy-VirtualBox:~$ du -sh  
93M .  
wendy@wendy-VirtualBox:~$ free -h  
              total        utilisé         libre      partagé tamp/cache  disponibl  
em:           1,9Gi        840Mi        453Mi        22Mi        674Mi        951Mi  
change:       4,2Gi       1,3Gi        2,9Gi  
wendy@wendy-VirtualBox:~$
```

```
wendy@wendy-VirtualBox:~$ ps aux  
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND  
root      1  0.1  0.3 176100  7620 ?      Ss  09:00  0:07 /sbin/init sp  
root      2  0.0  0.0      0     0 ?      S  09:00  0:00 [kthreadd]  
root      3  0.0  0.0      0     0 ?      S  09:00  0:00 [pool_workque  
root      4  0.0  0.0      0     0 ?      I< 09:00  0:00 [kworker/R-rc  
root      5  0.0  0.0      0     0 ?      I< 09:00  0:00 [kworker/R-rc  
root      6  0.0  0.0      0     0 ?      I< 09:00  0:00 [kworker/R-sl  
root      7  0.0  0.0      0     0 ?      I< 09:00  0:00 [kworker/R-ne  
root     12  0.0  0.0      0     0 ?      I< 09:00  0:00 [kworker/R-mm  
root     13  0.0  0.0      0     0 ?      I  09:00  0:00 [rcu_tasks_kt  
root     14  0.0  0.0      0     0 ?      I  09:00  0:00 [rcu_tasks_ru  
root     15  0.0  0.0      0     0 ?      I  09:00  0:00 [rcu_tasks_tr  
root     16  0.6  0.0      0     0 ?      S  09:00  0:28 [ksoftirqd/0]  
root     17  0.0  0.0      0     0 ?      I  09:00  0:03 [rcu_preempt]  
root     18  0.0  0.0      0     0 ?      S  09:00  0:00 [migration/0]  
root     19  0.0  0.0      0     0 ?      S  09:00  0:00 [idle_inject/  
root     20  0.0  0.0      0     0 ?      S  09:00  0:00 [cpuhp/0]  
root     21  0.0  0.0      0     0 ?      S  09:00  0:00 [kdevtmpfs]  
root     22  0.0  0.0      0     0 ?      I< 09:00  0:00 [kworker/R-in  
root     24  0.0  0.0      0     0 ?      S  09:00  0:00 [kaudittd]  
root     25  0.0  0.0      0     0 ?      S  09:00  0:00 [khungtaskd]  
root     26  0.0  0.0      0     0 ?      S  09:00  0:00 [oom_reaper]  
root     28  0.0  0.0      0     0 ?      I< 09:00  0:00 [kworker/R-wr
```

```
La commande « lspi » n'a pas été trouvée, veuillez vous écrire :
    commande « lspci » du deb pciutils (1:3.7.0-6)
    Essayez : sudo apt install <nom du deb>
wendy@wendy-VirtualBox:~$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:02.0 VGA compatible controller: VMware SVGA II Adapter
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service
00:05.0 Multimedia audio controller: Intel Corporation 82801AA AC'97 Audio Controller (rev 01)
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:0b.0 USB controller: Intel Corporation 82801FB/FBM/FR/fw/FRW (ICH6 Family) US
B2 EHCI Controller
00:0d.0 SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode] (rev 02)
wendy@wendy-VirtualBox:~$
```

```
0:0d.0 SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode] (rev 02)
wendy@wendy-VirtualBox:~$ sudo apt install traceroute
[sudo] Mot de passe de wendy :
lecture des listes de paquets... Fait
construction de l'arbre des dépendances... Fait
lecture des informations d'état... Fait
les NOUVEAUX paquets suivants seront installés :
traceroute
mis à jour, 1 nouvellement installés, 0 à enlever et 215 non mis à jour.
Il est nécessaire de prendre 45,4 ko dans les archives.
Avant cette opération, 152 ko d'espace disque supplémentaires seront utilisés.
  exception de :1 http://ht.archive.ubuntu.com/ubuntu jammy/universe amd64 traceroute
5,4 ko réceptionnés en 1s (80,8 ko/s)
```

```
wendy@wendy-VirtualBox:~
```

```
/usr/sbin/traceroute » (tcptraceroute) en mode automatique
Traitement des actions différées (« triggers ») pour man-db (2.10.2-1) ...
wendy@wendy-VirtualBox:~$ traceroute google.com
traceroute to google.com (172.217.3.78), 30 hops max, 60 byte packets
1 _gateway (10.0.2.2) 33.963 ms 1.730 ms 0.817 ms
2 * * *
3 * * *
4 * * *
5 * * *
6 * * *
7 * * *
8 * * *
9 * * *
0 * * *
1 * * *
2 * * *
3 * * *
4 * * *
5 * * *
6 * * *
7 * * *
8 * * *
9 * * *
0 * * *
```

```
* * *
* * *
* * *
* * *
* * *
* * *
* * *

ndy@wendy-VirtualBox:~$ netstat -tuln
commande « netstat » n'a pas été trouvée, mais peut être installée avec :
do apt install net-tools
ndy@wendy-VirtualBox:~$ ss -tuln
tid  State    Recv-Q   Send-Q     Local Address:Port      Peer Address:Port Process
p    UNCONN    0        0          0.0.0.0:42101        0.0.0.0:*
p    UNCONN    0        0          0.0.0.0:5353         0.0.0.0:*
p    UNCONN    0        0          0.0.0.0:631          0.0.0.0:*
p    UNCONN    0        0          127.0.0.53%lo:53       0.0.0.0:*
p    UNCONN    0        0          [::]:5353            [::]:*
p    UNCONN    0        0          [::]:48926           [::]:*
p    LISTEN    0        70         127.0.0.1:33060      0.0.0.0:*
p    LISTEN    0        151        127.0.0.1:3306      0.0.0.0:*
p    LISTEN    0        4096       127.0.0.53%lo:53       0.0.0.0:*
p    LISTEN    0        128        127.0.0.1:631          0.0.0.0:*
p    LISTEN    0        128        [::1]:631            [::]:*
```

```
ndy@wendy-VirtualBox:~$ journalctl
-- Logs begin at Fri 2019-02-15 10:08:55 CET, end at Fri 2019-02-15 10:08:55 CET.
wendy@wendy-VirtualBox:~$ journalctl
feb 10 10:08:55 wendy-VirtualBox kernel: Linux version 6.8.0-40-generic (buildd)
feb 10 10:08:55 wendy-VirtualBox kernel: Command line: BOOT_IMAGE=/boot/vmlinuz>
feb 10 10:08:55 wendy-VirtualBox kernel: KERNEL supported cpus:
feb 10 10:08:55 wendy-VirtualBox kernel:   Intel GenuineIntel
feb 10 10:08:55 wendy-VirtualBox kernel:   AMD AuthenticAMD
feb 10 10:08:55 wendy-VirtualBox kernel:   Hygon HygonGenuine
feb 10 10:08:55 wendy-VirtualBox kernel:   Centaur CentaurHauls
feb 10 10:08:55 wendy-VirtualBox kernel:   zhaoxin Shanghai
feb 10 10:08:55 wendy-VirtualBox kernel: BIOS-provided physical RAM map:
feb 10 10:08:55 wendy-VirtualBox kernel: BIOS-e820: [mem 0x0000000000000000-0x0>
feb 10 10:08:55 wendy-VirtualBox kernel: BIOS-e820: [mem 0x000000000009fc00-0x0>
feb 10 10:08:55 wendy-VirtualBox kernel: BIOS-e820: [mem 0x000000000000f0000-0x0>
feb 10 10:08:55 wendy-VirtualBox kernel: BIOS-e820: [mem 0x00000000000100000-0x0>
feb 10 10:08:55 wendy-VirtualBox kernel: BIOS-e820: [mem 0x0000000007ffff000-0x0>
feb 10 10:08:55 wendy-VirtualBox kernel: BIOS-e820: [mem 0x00000000fec00000-0x0>
feb 10 10:08:55 wendy-VirtualBox kernel: BIOS-e820: [mem 0x00000000fee00000-0x0>
feb 10 10:08:55 wendy-VirtualBox kernel: BIOS-e820: [mem 0x00000000fffc0000-0x0>
feb 10 10:08:55 wendy-VirtualBox kernel: NX (Execute Disable) protection: active
feb 10 10:08:55 wendy-VirtualBox kernel: APIC: Static calls initialized
feb 10 10:08:55 wendy-VirtualBox kernel: SMBIOS 2.5 present.
feb 10 10:08:55 wendy-VirtualBox kernel: DMI: innotek GmbH VirtualBox/VirtualBo>
feb 10 10:08:55 wendy-VirtualBox kernel: Hypervisor detected: KVM
feb 10 10:08:55 wendy-VirtualBox kernel: kvm-clock: Using msrs 4b564d01 and 4b56
```

```
wendy@wendy-VirtualBox:~$ journalctl -b
mas 03 09:01:04 wendy-VirtualBox kernel: Linux version 6.8.0-40-generic (buildd>
mas 03 09:01:04 wendy-VirtualBox kernel: Command line: BOOT_IMAGE=/boot/vmlinuz>
mas 03 09:01:04 wendy-VirtualBox kernel: KERNEL supported cpus:
mas 03 09:01:04 wendy-VirtualBox kernel:   Intel GenuineIntel
mas 03 09:01:04 wendy-VirtualBox kernel:   AMD AuthenticAMD
mas 03 09:01:04 wendy-VirtualBox kernel:   Hygon HygonGenuine
mas 03 09:01:04 wendy-VirtualBox kernel:   Centaur CentaurHauls
mas 03 09:01:04 wendy-VirtualBox kernel:   zhaoxin Shanghai
mas 03 09:01:04 wendy-VirtualBox kernel: BIOS-provided physical RAM map:
mas 03 09:01:04 wendy-VirtualBox kernel: BIOS-e820: [mem 0x0000000000000000-0x0>
mas 03 09:01:04 wendy-VirtualBox kernel: BIOS-e820: [mem 0x000000000009fc00-0x0>
mas 03 09:01:04 wendy-VirtualBox kernel: BIOS-e820: [mem 0x00000000000f0000-0x0>
mas 03 09:01:04 wendy-VirtualBox kernel: BIOS-e820: [mem 0x0000000000100000-0x0>
mas 03 09:01:04 wendy-VirtualBox kernel: BIOS-e820: [mem 0x00000000007ffff0000-0x0>
mas 03 09:01:04 wendy-VirtualBox kernel: BIOS-e820: [mem 0x000000000fec0000-0x0>
mas 03 09:01:04 wendy-VirtualBox kernel: BIOS-e820: [mem 0x000000000fee0000-0x0>
mas 03 09:01:04 wendy-VirtualBox kernel: BIOS-e820: [mem 0x000000000fffc0000-0x0>
mas 03 09:01:04 wendy-VirtualBox kernel: NX (Execute Disable) protection: active
mas 03 09:01:04 wendy-VirtualBox kernel: APIC: Static calls initialized
mas 03 09:01:04 wendy-VirtualBox kernel: SMBIOS 2.5 present.
mas 03 09:01:04 wendy-VirtualBox kernel: DMI: innotek GmbH VirtualBox/VirtualBo>
mas 03 09:01:04 wendy-VirtualBox kernel: Hypervisor detected: KVM
mas 03 09:01:04 wendy-VirtualBox kernel: kvm-clock: Using msrs 4b564d01 and 4b5>
Lines 1-23...skipping...
mas 03 09:01:04 wendy-VirtualBox kernel: Linux version 6.8.0-40-generic (buildd>
```

```
wendy@wendy-VirtualBox:~$ journalctl -n 10
mas 03 10:26:44 wendy-VirtualBox gnome-shell[1356]: Window manager warning: W28>
mas 03 10:26:44 wendy-VirtualBox gnome-shell[1356]: Window manager warning: las>
mas 03 10:26:44 wendy-VirtualBox gnome-shell[1356]: Window manager warning: W28>
mas 03 10:26:45 wendy-VirtualBox gnome-shell[1356]: Window manager warning: las>
mas 03 10:26:45 wendy-VirtualBox gnome-shell[1356]: Window manager warning: W28>
mas 03 10:26:46 wendy-VirtualBox gnome-shell[1356]: Window manager warning: las>
mas 03 10:26:46 wendy-VirtualBox gnome-shell[1356]: Window manager warning: W28>
mas 03 10:26:52 wendy-VirtualBox systemd[1198]: Started VTE child process 7882 >
mas 03 10:27:13 wendy-VirtualBox systemd[1198]: vte-spawn-caf22c7b-1ef5-438b-9e>
mas 03 10:27:35 wendy-VirtualBox systemd[1198]: Started VTE child process 7893 >
lines 1-10/10 (END)
```

Conclusion

Pour réaliser ce travail, j'ai dû faire un effort considérable pour maîtriser les théories du cours et également approfondir mes connaissances à l'aide de tutoriels sur YouTube.