

TP2_INF203

Groupe: INF2

Binôme: Shaghayegh HAJMOHAMMADKASHI

Kaiwen ZHENG

EXO1 :

Terminal : cd INF203

mkdir TP2

./TP1/scripts/installeTP.sh 2

le repertoire courant est

/home/h/hajmohas/INF203/TP2

il contient

total 12

-rwxr-xr-x 1 hajmohas ima-nogroup 39 9 févr. 15:51 2entiers.sh

-rwxr-xr-x 1 hajmohas ima-nogroup 1097 9 févr. 15:51 instant_suivant.sh

-rwxr-xr-x 1 hajmohas ima-nogroup 165 9 févr. 15:51 max2.sh

-rwxr-xr-x 1 hajmohas ima-nogroup 73 9 févr. 15:51 somme.sh

cd TP2

ls

2entiers.sh instant_suivant.sh max2.sh somme.sh

[a]:

Lorsqu'on exécute la commande "date", elle affiche la date et l'heure actuelles sur la ligne de commande.

Ensuite, on utilise l'opérateur de redirection ">" pour rediriger la sortie de la commande "date" vers un fichier nommé "une date". Cela signifie que la date et l'heure actuelles seront écrites dans le fichier "une date" au lieu d'être affichées à l'écran.

Lorsqu'on exécute la commande "cat une date", elle affiche le contenu du fichier "une date", qui devrait être la date et l'heure auxquelles la commande "date" a été exécutée pour la première fois.

Enfin, lorsqu'on exécute la commande "ls > une date", elle redirige la sortie de la commande "ls" (qui répertorie les fichiers dans le répertoire actuel) vers le fichier "une date". Cela signifie que le contenu du fichier "une date" sera remplacé par la liste des fichiers dans le répertoire actuel au moment où la commande a été exécutée.

[b] :

Terminal : ls -l max2.sh

```
-rwxr-xr-x 1 hajmohas ima-nogroup 165 9 févr. 15:51 max2.sh
```

Dans la commande "ls -l max2.sh", la taille du fichier "max2.sh" est indiquée dans la quatrième colonne de la sortie, qui est de 165 octets.

```
Terminal : ./max2.sh
```

Saisissez deux entiers

5

10

Le plus grand des deux entiers 5 et 10 est 10

Le texte en vert représente la sortie de programme, tandis que le texte en violet correspond à l'écho de ce que nous avons saisi au clavier.

On a créé (avec l'éditeur de fichier de VSCode) un fichier deux entiers contenant uniquement deux entiers 2 et 4

```
Terminal : ./max2.sh < deux_entiers
```

Saisissez deux entiers

Le plus grand des deux entiers 2 et 4 est 4

[c] :

```
Terminal : ./deux_entiers > ./max2.sh
```

```
Error : ./deux_entiers: Permission non accordée
```

```
chmod 777 deux_entiers
```

```
./deux_entiers > ./max2.sh
```

```
Error : ./deux_entiers: ligne 1: 2 : commande introuvable
```

```
Error : ./deux_entiers: ligne 2: 4 : commande introuvable
```

Les messages d'erreur affichés indiquent que les commandes "2" et "4" ne sont pas trouvées. Car les entrées n'ont pas été fournies lors de l'exécution de la commande.

```
Terminal : ls -l max2.sh
```

```
-rwxr-xr-x 1 hajmohas ima-nogroup 0 9 févr. 16:51 max2.sh
```

la taille du fichier "max2.sh" dépendra de la longueur de la sortie produite. Si la sortie est vide, la taille sera de 0 octets. Sinon, la taille du fichier sera égale à la longueur de la sortie.

[d] :

```
Terminal : cp /Public/203_INF_Public/TP2/max2.sh ~/INF203/TP2
```

```
./max2.sh > resultat
```

[e] :

Il ne se passe-t-il "rien" car on n'avait pas le droit d'exécuter le fichier max2.sh.

```
chmod +x max2.sh
```

```
cat resultat
```

Saisissez deux entiers

Le plus grand des deux entiers est

[f] : Terminal : `max2.sh < deux_entiers > resultat1` (commande pour exécuter max2.sh en lisant les entrées dans le fichier deux entiers en écrivant le résultat dans resultat1)

```
cat resultat1
```

Saisissez deux entiers

```
./max2.sh > resultat2 < deux_entiers
```

```
cat resultat2
```

Saisissez deux entiers

Le plus grand des deux entiers 2 et 4 est 4

Il n'y a pas de différence entre les deux fichiers de résultat

Exo2 :

Terminal : `./2entiers.sh`

30015

10357

```
./2entiers.sh
```

18462

2173

```
./2entiers.sh > deux_entiers_bis
```

```
cat deux_entiers_bis
```

206

25003

```
./max2.sh < deux_entiers_bis
```

Saisissez deux entiers

Le plus grand des deux entiers 206 et 25003 est 25003

```
./2entiers.sh | ./max2.sh
```

Saisissez deux entiers

Le plus grand des deux entiers 31962 et 13864 est 31962

Exo3 :

[g] :

La syntaxe $\$(...)$ est appelée la substitution de commande dans Bash. Elle permet d'exécuter une commande à l'intérieur d'une autre commande et de remplacer la sortie de cette commande dans la commande principale. Cela permet de combiner les résultats de différentes commandes en une seule.

Quant à la commande "expr" en Bash, elle est utilisée pour évaluer des expressions arithmétiques et de chaînes de caractères.

Terminal :

```
./somme.sh
2
4
2+4=6
cp ./somme.sh difference.sh
cp ./somme.sh produit.sh
cp ./somme.sh quotient.sh
cp ./somme.sh reste.sh
ls
2entiers.sh deux_entiers_bis instant_suivant.sh produit.sh reste.sh resultat1 somme.sh
deux_entiers difference.sh max2.sh quotient.sh resultat resultat2 une_date
./reste.sh
10
7
10%7=3
./quotient.sh
9
2
9/2=4
./produit.sh
8
7
8*7=56
./difference.sh
```

9

4

9-4=5

```
ls -l
total 36
-rwxr-xr-x 1 hajmohas ima-nogroup 39 9 févr. 15:51 2entiers.sh
-rwxrwxrwx 1 hajmohas ima-nogroup 3 9 févr. 16:43 deux_entiers
-rw-r--r-- 1 hajmohas ima-nogroup 10 9 févr. 18:17 deux_entiers_bis
-rwxr-xr-x 1 hajmohas ima-nogroup 73 17 févr. 19:21 difference.sh
-rwxr-xr-x 1 hajmohas ima-nogroup 1097 9 févr. 15:51 instant_suivant.sh
-rwxrwxrwx 1 hajmohas ima-nogroup 165 17 févr. 13:38 max2.sh
-rwxr-xr-x 1 hajmohas ima-nogroup 74 17 févr. 19:21 produit.sh
-rwxr-xr-x 1 hajmohas ima-nogroup 73 17 févr. 19:21 quotient.sh
-rwxr-xr-x 1 hajmohas ima-nogroup 73 17 févr. 19:21 reste.sh
-rw-r--r-- 1 hajmohas ima-nogroup 61 17 févr. 13:49 resultat
-rw-r--r-- 1 hajmohas ima-nogroup 67 9 févr. 18:06 resultat1
-rw-r--r-- 1 hajmohas ima-nogroup 67 9 févr. 18:09 resultat2
-rwxr-xr-x 1 hajmohas ima-nogroup 73 9 févr. 15:51 somme.sh
-rw-r--r-- 1 hajmohas ima-nogroup 61 9 févr. 16:01 une_date
-d '-' -f 2
```

36

1

1

1

1

1

1

1

1

1

1

1

1

1

1

```
ls -l | tr -s ' ' | cut -d ' ' -f 5
```

39

3

10

73

1097

165

74

73

73

61

67

67

73

61

Exo4 :

[h] :

1.Test avec une heure valide (00-23), une minute valide (00-59) et une seconde valide (00-59).

Exemple : 09:30:25

2.Test avec une heure invalide (supérieure à 23), une minute valide et une seconde valide.

Exemple : 25:30:25

3.Test avec une heure valide, une minute invalide (supérieure à 59) et une seconde valide.

Exemple : 09:75:25

4.Test avec une heure valide, une minute valide et une seconde invalide (supérieure à 59).

Exemple : 09:30:75

5.Test avec une heure invalide, une minute invalide et une seconde invalide.

Exemple : 25:75:75

6. Test avec une heure valide, une minute valide et une seconde égale à 59 (pour tester la gestion du passage à la minute suivante).

Exemple : 09:30:59

7. Test avec une heure valide, une minute égale à 59 et une seconde égale à 59 (pour tester la gestion du passage à l'heure suivante).

Exemple : 09:59:59

[i]:

```
#!/bin/bash
echo "Entrez une valeur pour instant (au format heure:minute:seconde):"
# lecture de la variable instant : à compléter
read instant
# instant est de la forme heure:minute:seconde
# lecture de chaque champ avec la commande cut :
heure=$(echo $instant | cut -d: -f 1)
# à compléter pour minute et seconde :
minute=$(echo $instant | cut -d: -f 2)
seconde=$(echo $instant | cut -d: -f 3)
# affichage eventuel de heure minute et seconde pour verifier que tout s'est
bien passé. à compléter
echo "Heure: $heure, Minute: $minute, Seconde: $seconde"

# ajouter 1 à seconde pour obtenir secondesuiv : à compléter
secondesuiv=$(expr $seconde + 1)
# ajouter minute et secondesuiv/60 pour obtenir minutesuiv : à compléter
minutesuiv=$(expr $minute + $secondesuiv / 60)
# ajouter heure et minutesuiv/60 pour obtenir heuresuiv : à compléter
heuresuiv=$(expr $heure + $minutesuiv / 60)

# affichage eventuel de heuresuiv minutesuiv et secondesuiv pour verifier que
tout s'est bien passé. à compléter
echo "Heuresuiv (pre-correction): $heuresuiv, Minutesuiv (pre-correction):
$minutesuiv, Secondesuiv (pre-correction): $secondesuiv"

# correction des résultats au changement de minute, d'heure et de jour :
# prendre le reste de secondesuiv par 60 (dans secondesuiv par exemple)
secondesuiv=$(expr $secondesuiv % 60)
# de même pour minutesuiv : à compléter
minutesuiv=$(expr $minutesuiv % 60)
# trouver par qui prendre le reste pour heuresuiv : à compléter
heuresuiv=$(expr $heuresuiv % 24)

# affichage de heuresuiv minutesuiv et secondesuiv pour verifier que tout
s'est bien passé.
echo "Heuresuiv: $heuresuiv, Minutesuiv: $minutesuiv, Secondesuiv:
$secondesuiv"
```

```
date +%T
```

20:48:59

```
date +%T > une_date
```

```
./instant_suivant.sh < une_date
```

Entrez une valeur pour instant (au format heure:minute:seconde):

Heure: 20, Minute: 48, Seconde: 59

Heuresuiv (pre-correction): 20, Minutesuiv (pre-correction): 49, Secondesuiv (pre-correction): 60

Heuresuiv: 20, Minutesuiv: 49, Secondesuiv: 0

[j] :

```
date +%T | ./instant_suivant.sh
```

Entrez une valeur pour instant (au format heure:minute:seconde):

Heure: 20, Minute: 56, Seconde: 48

Heuresuiv (pre-correction): 20, Minutesuiv (pre-correction): 56, Secondesuiv (pre-correction): 49

Heuresuiv: 20, Minutesuiv: 56, Secondesuiv: 49

[h] :

```
mkdir DIR
```

```
]: echo $?
```

0

```
mkdir DIR
```

mkdir: impossible de créer le répertoire « DIR »: Le fichier existe

```
echo $?
```

1

```
rmdir DIR
```

```
echo $?
```

0

```
cd DIR
```

bash: cd: DIR: Aucun fichier ou dossier de ce type

```
echo $?
```

1

```
cp ../max2 cop
```

cp: impossible d'évaluer '../max2': Aucun fichier ou dossier de ce type


```
echo $?
```

1

```
cp ../../max2 cop
```

cp: impossible d'évaluer '../../max2': Aucun fichier ou dossier de ce type

```
echo $?
```

1

```
cp cop
```

cp: opérande de fichier cible manquant après 'cop'

Saisissez « cp --help » pour plus d'informations.

```
echo $?
```

1

[k] :

Lors de l'exécution d'une commande Unix, celle-ci peut renvoyer un code entier qui indique si l'exécution s'est terminée avec succès ou avec une erreur. En général, un code de retour de 0 indique que la commande s'est terminée avec succès, tandis qu'un code différent de 0 indique une **erreur(1)**.

[l] :

NOM_A n'a pas le droit d'effectuer cette copie car les espaces de travail des deux membres du binôme sont probablement séparés et chacun a ses propres droits d'accès et de modification sur son propre espace de travail. Ainsi, NOM_A ne peut pas accéder directement à l'espace de travail de NOM_B pour y effectuer la copie du fichier.

[m] :

```
cp -r /home/premiere_lettre_NomB/NomB/INF203/TP2 TP2
```

[n] :

On peut utiliser la commande `chmod u+rw` pour donner accès à Nom A pour pouvoir effectuer des changements. Pour annuler le pouvoir de Nom B on utilise `chmod u-rw`. (avec u c'est user et on veut rajouter les paramètres de lecture et d'écriture)