

Classe:.....

Nom:.....

Groupe:

Prénom:.....

**CONTRÔLE : SYSTÈME D'EXPLOITATION 4
3IIR**

Barème : 2 points par question

A- Soit un script bash monscript.sh dans le répertoire ~/bin/ qui a les droits d'accès suivants 444. Ce script contient le code suivant :

```
-----  
#!/bin/bash  
i=0  
while [[ $i -lt 24 ]]; do  
    i=$((++i))  
    echo $(date +"%F %R") >> date.log  
    sleep $((60*60))  
done  
-----
```

#rappel: le format de date "%F %R" correspond à "YYYY-MM-DD HH-mm"

Ce script est planifié avec une tâche cron en utilisant la ligne suivante :

```
0 0 * * * cd /var/log/test/ && ~/bin/monscript.sh
```

1. Quel est le nombre de lignes dans le fichier date.log après deux jours de sa première exécution?

48.....

2. Quel est l'emplacement du fichier date.log (répertoire absolu) ?

/var/log/test/.....

3. On veut effectuer le même travail en supprimant la boucle et la commande sleep. Donnez la nouvelle programmation de la tâche Cron:

0 * * * *

4. Quelle est la principale différence entre la répétition avec la boucle et la répétition planifiée par Cron?

Dans le premier cas, le programme reste toujours en execution, donc il consomme les ressources. Dans le deuxième cas, il s'exécute instantanément et libère les ressources

.....
.....
.....
.....

B- Soit un script shell appelé de la manière suivante :

```
./script.sh a b c d e
```

1. Donnez l'expression qui permettra de remplacer l'argument \$4 par "D" au lieu de "d" tout en gardant le reste des arguments inchangés.

```
set $1 $2 $3 D $5
```

-
2. Le script contient la commande suivante: `echo ${@}`
Quel est le résultat affiché par cette commande ?

```
a b c d e (ou bien a b c D e)
```

3. Quel serait le résultat affiché par la commande suivante si elle est ajoutée à la fin de notre script `monscript.sh` ?
`echo "Le premier argument est\t $1"`

```
Le premier argument est\t a (\t ne sera pas interprété)
```

C- A quoi sert la commande "tr -s" ?

```
à supprimer les répétition du même caractère
```

D- Écrire une fonction "multiplication" qui prend en argument deux entiers a et b et affiche leur résultat (a x b).

rappe! pour faire une multiplication on peut utiliser les commandes `let`, `expr` ou la syntaxe `$((...))`

1. Comment on fait l'appel à une fonction sous Bash?
comme une commande. Exemple :

```
multiplication 5 3
```

-
2. Donner le code complet de la fonction "multiplication"

```
multiplication() {  
    echo "Le résultat de $1 multiplié par $2 est: " $((($1 * $2));  
}
```