

<b>Matière :</b> Système d'Exploitation 4		<b>Filière :</b> IIR	<b>Niveau :</b> 3
<b>Date de l'épreuve :</b> 16/06/2022		<b>Durée de l'épreuve :</b> 2H	
<b>Documents autorisés</b> Oui <input type="checkbox"/> Non <input checked="" type="checkbox"/>	<b>Matériels autorisés :</b> Oui <input type="checkbox"/> : (Préciser).....AUCUN..... Non <input checked="" type="checkbox"/>		
Pour ne pas oublier, inscrivez votre nom et prénom avant de commencer:	<b>Nom :</b> ..... <b>Prénom :</b> ..... <b>Classe :</b> .....		
<b>- Répondre directement sur le questionnaire</b> - Répondre aux questions telles qu'elles, aucune explication supplémentaire ne sera donnée. - Les réponses doivent être brèves, précises et clairement présentées - Calculatrices, téléphones portables et objets connectés <b>NON</b> autorisés - Interdiction de toute forme d'échange verbal, écrit ou matériel entre candidats			

Rappels:

- L'option -s de la commande tr permet d'annuler les répétitions.

## 1<sup>e</sup> partie : compréhension

(8pts)

1.1. Quel est le travail de chacun des opérateurs de redirection suivants

>&2 .....  
&> .....

1.2. Quels sont les paramètres positionnels ?

- Variables spéciales pour assigner des arguments à partir de la ligne de commande
- Paramètres pour correspondre un modèle
- Variables spéciales pour lire l'entrée de l'utilisateur
- Variables et modèles spéciaux

1.3. Considérons la sortie suivante de ce script :

```
$ ./report.sh  
$ echo $?  
0
```

1.3.(suite) Que peut-on dire :

- Le script a mal tourné.
- Le script s'est bien exécuté.
- Un argument est passé à l'environnement durant l'exécution du script.
- Pas d'arguments passés au script.

1.4. Lors de l'écriture d'un script bash, vous devez quitter plus tôt en raison d'une condition d'erreur. Laquelle des commandes suivantes devez-vous utiliser ?

- die
- break
- exit 1
- raise

1.5. Dans un script shell qui accepte des paramètres dans la ligne de commande :

Comment vous assurer que vous avez plus de trois paramètres ?

- a. `if [[ $# -ge 3 ]]`
- b. `if [[ $ARGV > 3 ]]`
- c. `if [[ 3 -eq $# ]]`
- d. `if [[ $! = 3 ]]`

1.6. Quelle commande est utilisée pour rendre le script shell interactif

- a. `input`
- b. `ip`
- c. `read`
- d. `output`

1.7. Quelle commande affiche un affichage d'écran fréquemment mis à jour des processus en cours d'exécution sur le système.

- a. `threads`
- b. `free`
- c. `procinfo`
- d. `top`

1.8. Les commandes suivantes permettent de lancer le script bash: `myscript` se trouvant dans le répertoire courant. l'une d'elles exige que le fichier `myscript` ait le droit d'accès `x`

- a. `$ source ./myscript`
- b. `$ bash myscript`
- c. `$ ./myscript`
- d. `$ . myscript`

1.9. Parmi les commandes suivantes laquelle ne permet pas d'afficher la valeur de la variable `x`

- a. `$ echo $x`
- b. `$ printf '%s\n' $x`
- c. `$ print $x`

1.10. Pour afficher un message d'erreur dans un script shell on écrit :

- a. `$ echo "message" >&2`
- b. `$ echo "message" &>2`
- c. `$ echo "message" > stderr`
- d. `$ echo stderr "message"`

1.11. Laquelle des commandes suivantes, si elles sont tapées dans un shell bash, créera une variable d'environnement appelée `MYVAR` avec le contenu `mystuff` qui sera accessible aux programmes lancés ultérieurement ?

- a. `$ MYVAR="mystuff"`
- b. `$ $MYVAR==mystuff`
- c. `$ export MYVAR="mystuff"`
- d. `$ echo $MYVAR mystuff`

1.12. Pour déclarer un tableau qui s'appelle `myarray` on peut utiliser la commande :

- a. `$ array myarray[20]`
- b. `$ myarray=array[20]`
- c. `$ myarray=( bonjour mes amis )`
- d. `$ array[20]`

1.13. Laquelle des lignes suivantes, si elle est utilisée dans une tâche cron, exécutera le script `cleanup.sh` chaque nuit à 9:30?

- a. `30 9 * * * /usr/bin/cleanup.sh`
- b. `30 21 * * * /usr/bin/cleanup.sh`
- c. `9 30 * * * /usr/bin/cleanup.sh`
- d. `21 30 * * * /usr/bin/cleanup.sh`

1.14. Quelle commande supprimera toutes les lignes vides du fichier `old.txt`?

- a. `sed '/d/' old.txt`
- b. `sed '/^/d/' old.txt`
- c. `sed '/^$/d/' old.txt`
- d. `sed '/^*/d/' old.txt`

1.15. Un script se voit transmettre 10 paramètres. Comment récupérer le dixième paramètre ?

- a. Par \$10
- b. Par un shift puis \$9
- c. Par A=10 puis \${A}.

1.16. Pour incrémenter une variable x sous bash, on peut écrire :

- a. x++
- b. let x++
- c. x = \$x + 1

## 2<sup>e</sup> partie : interprétation de scripts

(8pts)

2.1. Quel serait le résultat des commandes suivantes sous bash:

```
$ t=2.8; t=${t#*.*}; echo "10-$t"
```

```
.....  
$ s="Tout passe, tout lasse, sauf la bonne glasse"
```

```
$ echo -n "${s:0:38}"; echo "${s:38}" | tr -s "s" "c"
```

```
.....  
$ var1=`(var2=101; echo "5*$var2")`; echo ":$var1:$var2:"
```

2.2. Soit la commande suivante utilisée en boucle dans un script bash :

```
getopts ":df:" option
```

- Quelles sont les options qu'on peut lire dans ce script?

2.3. Nous avons un dossier "rep" qui contient 20 fichiers nommés f01, f02, ... ,f19, f20; Nous avons aussi un script shell **script.sh** executable et contient le code suivant :

```
#!/bin/bash
```

```
echo "$1 , $2"
```

- Quel serait le contenu du fichier **fich** après l'exécution de la ligne suivante :

```
$ ls rep | tail -n 4 | tee fich | xargs -n 2 script.sh >> fich
```

2.4. Supposons que la variable `myNumber` contient une chaîne composée de 10 chiffres. Que va produire cette commande ?

```
$ echo $myNumber | sed -e 's/^[0-9][0-9][0-9]/(&)/g'
```

.....

**3<sup>e</sup> partie : production de scripts** **(4pts)**

3.1. Écrire un script AWK qui fait le même travail que la commande suivante :

```
cut -d: -f 1,3 monfichier
```

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

3.2. On considère la sortie de la commande `ps -aux` ayant l'entête suivant:

```
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
```

- Sachant que les colonnes sont séparées par un ou plusieurs espaces, écrivez le script (bash ou awk au choix) qui donne les 3 programmes qui consomment le plus grand pourcentage de cpu.

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