|  |  |  |
| --- | --- | --- |
|  | Examen  Semestre : 1 2    Session : Principale Rattrapage | |
| ETUDIANT(e)  Nom et Prénom : romdhani eya allah  Classe : 2a38 | | Code : |
| Module : Systèmes & Réseaux  Enseignant(s) : UP Systèmes & UP Réseaux  Classe(s) : 2A22A42  Documents autorisés : OUI NON Nombre de pages : 7  Calculatrice autorisée : OUI NON Internet autorisée : OUI NON  Date : 28/05/2024 Heure : 11H Durée :1H30 | | |

\*\*✂

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Code | Note    /20 | Nom et Signature du Surveillant | Nom et Signature du Correcteur | Observations |

# Exercice 1 : ( 8points)

Choisir la bonne réponse :

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | Quel signe indique l’état de sortie de la dernière commande exécutée ?  A. $? \* vrai   1. $# 2. $$ 3. $! | 2. | La commande Env ou printenv permet d’ :   1. Afficher les arguments 2. Afficher les variables locales 3. Afficher les variables d’environnement vrai\* 4. Aucune de ces réponses |
| 3. | Pour soustraire l’entier b de l’entier a, le script à utiliser est :   1. echo $a-$b 2. echo a-b 3. echo ($a-$b) 4. echo $(($a-$b)) \*vrai | 4. | Dans un script Shell, la ligne #!/bin/bash :   1. Indique le chemin vers le fichier de sortie du script. 2. Définit les permissions du script. 3. Indique l'interpréteur à utiliser pour exécuter le script \*vrai 4. Définit la variable d'environnement du script. |
| 5. | Pour définir une variable utilisateur var1, quelle commande doit-on exécuter ?   1. set var1 valeur1 2. var1=valeur1 3. var1 = valeur1 4. var var1=valeur1 \*vrai | 6. | Quelle séquence de caractères est utilisée pour fournir le cas par défaut dans une instruction case : A. default:   1. =) 2. .) 3. \*) \*vrai |

|  |  |  |
| --- | --- | --- |
|  | NE RIEN ECRIRE |  |
|  |

|  |  |  |
| --- | --- | --- |
| 7. | Dans un script Shell, $1et $0 représentent respectivement :   1. Le premier argument passé lors de l’exécution et le nom du script à exécuter. 2. Le nom du script à exécuter et le premier argument passé lors de l’exécution. 3. Le premier et le deuxième argument passé lors de l’exécution La réponse A et B. 4. Le deuxième et Le premier argument passé lors de l’exécution. \*vrai | 8. Laquelle des commandes suivantes permettent d’afficher les lignes du fichier « fichier1 » qui commencent par « Robuste » ou « Rabuste»?   1. grep r{ a,o}buste$ fichier1 2. grep ^R[ao] buste fichier1 \*VRAI 3. grep ^r[ao] buste fichier1 4. grep ^R[a-o] buste fichier1 |
| 9. | Quelle est la ligne de commande qui donne les permissions suivantes au fichier hello.txt  : Lecture, écriture et exécution pour le propriétaire, écriture et exécution pour les utilisateurs du groupe et lecture pour les autres utilisateurs.   1. chmod 774 hello.txt 2. chmod 764 hello.txt 3. chmod a= rw- hello.txt 4. chmod u=rwx,g=wx,o=r hello.txt \*VRAI | 10. Comment obtenir la longueur du tableau « array»?   1. Utiliser ${#array[@]} \*VRAI 2. Utiliser $array[length] 3. Utiliser $array[#] 4. Utiliser ${array[length]} |
| 11. Que signifie 2>&1 dans une ligne d'un script Shell?   1. Cela permet de rediriger la sortie standard vers la sortie d’erreur. 2. Cela permet de simplifier le passage de l’entrée standard vers la sortie standard 3. Cela permet de rediriger à la fois la sortie standard et les erreurs vers le même fichier.   \*VRAI | | 12. Quelle commande permet de vérifier si un fichier existe?   1. test -e fichier \*VRAI 2. verify fichier 3. file\_exists fichier 4. check -f fichier |
| 13. Quelle est la commande qui permet de déterminer la longueur d’une chaine de caractères nommé « chaine » :     1. ${#!chaine} 2. expr length $chaine \*VRAI 3. expr "$chaine\*" : ‘\* !.’ | | 14. Quel est le résultat de l'exécution du script suivant ?        A.    10    B. 15 \*VRAI  C. 5  D. 25 |
| 15. Le script suivant permet de :       1. afficher la liste des fichiers du répertoire / accessibles en lecture 2. afficher la liste des fichiers du répertoire /bin accessibles en exécution 3. afficher la liste des fichiers du répertoire /bin accessibles en lecture et écriture. 4. afficher la liste des fichiers du répertoire / accessibles en écriture. \*VRAI | | 16. le résultat d’exécution du programme suivant :       1. chance2A \*VRAI 2. echance2 3. chance2A-2 4. cha |

# Exercice 2 : ( 5points)

1. Remplir les espaces vides :

#!/bin/bash

# Déclaration d'un tableau d'entiers nommé tableau qui contient les valeurs suivants 10, 20, 30, 40, 50. (0,5pt)

REPONSE :

declare -a tab=(10 20 30 40 50)

# Fonction pour afficher le contenu du tableau (0,5pt)

afficher\_tableau() {

echo "Contenu du tableau : echo ${tab[@]} "

}

# Fonction pour déterminer la longueur du tableau (0,5pt) longueur\_tableau() {

echo "Longueur du tableau : echo ${#tab[\*]} " }

# Fonction pour calculer la somme de tous les éléments du tableau (1pt) somme\_tableau() {

somme=0

for i in "${tableau[@]}"; do

somme=$((somme + i))

done

echo "Somme des éléments du tableau : $somme  " }

# Fonction pour ajouter un élément au début du tableau. (0,5pt)

ajouter\_debut() {

nouvel\_element=$1

tableau==( $nouvel\_element ${tab[\*]})

echo "Ajout de $nouvel\_element au début du tableau."

}

# Fonction pour ajouter un élément à la fin du tableau. (0,75pt) ajouter\_fin() {

nouvel\_element= $1 ;

tableau ==( ${tab[\*]}$nouvel\_element)

echo "Ajout de $nouvel\_element à la fin du tableau." }

2. Comment faire l’appel de ces fonctions pour les cas suivants :

# Appel de la fonction ajouter\_debut pour ajouter la valeur 5. (0,5pt)

ajouter\_debut "5 "

# Appel de la fonction ajouter\_fin pour ajouter la valeur 60. (0,5pt)

ajouter\_fin "60"

# Affichage du tableau mis à jour (0,25pt)

afficher\_tableau

|  |  |
| --- | --- |
| Nom et Prénom : …………………………………………………...  Classe : ………………………………………………….................... | Code : |

# Exercice 3: ( 7points)

L’objectif de cet exercice est d’écrire un script Shell qui agit comme un gestionnaire de répertoires simplifié nommé « Gestion\_rep ». L'utilisateur devrait pouvoir naviguer dans les répertoires et atteindre les fonctionnalités suivantes :

1. : Afficher le contenu du répertoire passé en argument.
2. : Afficher la liste des fichiers exécutables pour tous les utilisateurs dans le répertoire passé en argument.
3. : Déplacer tous les sous répertoires qui dépassent la taille 100 ko dans le répertoire passé en argument vers /tmp.
4. : compter le nombre de fichiers image d’extension « .jpeg» qui se trouve dans le répertoire passé en argument.

1. Ecrire la fonction afficher\_usage qui affiche dans la sortie standard l’usage du script comme suit : (1pt)

USAGE : Gestion\_rep : [-a] [-b] [-c] [-d] Répertoire

1. Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

   Le contenu généré par l’IA peut être incorrect.Compléter la création des fonctions appelées pour chaque option tapée (2pts)

#!/bin/bash

# Fonction pour afficher le contenu du répertoire passé en argument

afficher\_contenu() {

if [-d "$1"]; then #tester que l’argument passé en paramètre est un répertoire.

echo "Contenu du répertoire $1 :"

ls -l "$1" #affiche le contenu de la liste détaillé

else

echo "Erreur: Le chemin spécifié n'est pas un répertoire valide."

Exit 1

fi

}

# Fonction pour afficher la liste des fichiers exécutables pour tous les utilisateurs dans le répertoire passé en argument

afficher\_executables() {

if [ -d "$1" ]; then #tester que l’argument passé en paramètre est un répertoire.

echo "Fichiers exécutables dans le répertoire $1 :"

find -x"$1"

else echo "Erreur: Le chemin spécifié n'est pas un répertoire valide."

exit 1

fi

# NE RIEN ECRIRE

# Fonction pour déplacer tous les sous-répertoires qui dépassent la taille de 100 ko vers /tmp deplacer\_grands\_sous\_repertoires() {

if [ -d "$1" ]; then

echo "Déplacement des sous-répertoires de plus de 100 ko dans $1 vers /tmp :"

find "$1" -mindepth 1 -maxdepth 1 -type d -exec du -k {} \; | awk '$1 > 100 {print $2}' | while read rep; do

mv "$rep" /tmp/

else

echo "Erreur: Le chemin spécifié n'est pas un répertoire valide."

exit 1

fi

argument ……………………………………………………………………………………………………………………………………………………… compter\_images\_jpeg() {

if [ ……………………………… ]; then #tester que l’argument passé en paramètre est un répertoire. echo "Nombre de fichiers image .jpeg dans le répertoire $1 :"

…………………………………………………………………………………………………….. else

echo "Erreur: Le chemin spécifié n'est pas un répertoire valide."

exit ………………

fi

}

1. Une fois le transfert est effectué, il a reçu un email pour la confirmation.
2. Quels sont les protocoles de la couche application utilisés pour cette réception ? (0.25 pt)

Les protocoles utilisés dans la réception des mails sont : POP3 / IMAP

2.} Quelles sont les couches du modèle OSI qui ont le même rôle que la couche application du

Couche Session / présentation

# Fonction pour compter le nombre de fichiers image .jpeg dans le répertoire passé en modèle TCP/IP ? (0.25 pt)

3. Le script principal teste la présence d’au moins un argument, sinon il affiche l’usage sur la sortie d’erreur et échoue. Puis exécute l’une des fonctions selon l’option donnée lors de l’exécution du script. Compléter les champs manquants. (2,5pts)

if [ $# -eq 0 ] || ! `echo $1 | grep -q "-"` then #ligne 1

………………………………………………………………………………………………..

……………………………………………………………………………………………….. else

while

getopts …………………………………………………………………………… do

…………………… …… $option in

…..) ……………………………………………………………………………. $OPTARG;; …..) ………………………………………………………………………………………….;;

…..) ………………………………………………………………………………………….;;

…..) ………………………………………………………………………………………….;;

\*) ……………………………………………………………………………………………;; esac

………………………………………………………………………………………………….

…………………………………………………………………………………………………

1. Expliquer la ligne 1 en se référant à l’annexe1 pour comprendre la commande grep –q : (0,5pts)

…………………………………………………………………………………………………………

…………………………………………………………………………………………………………

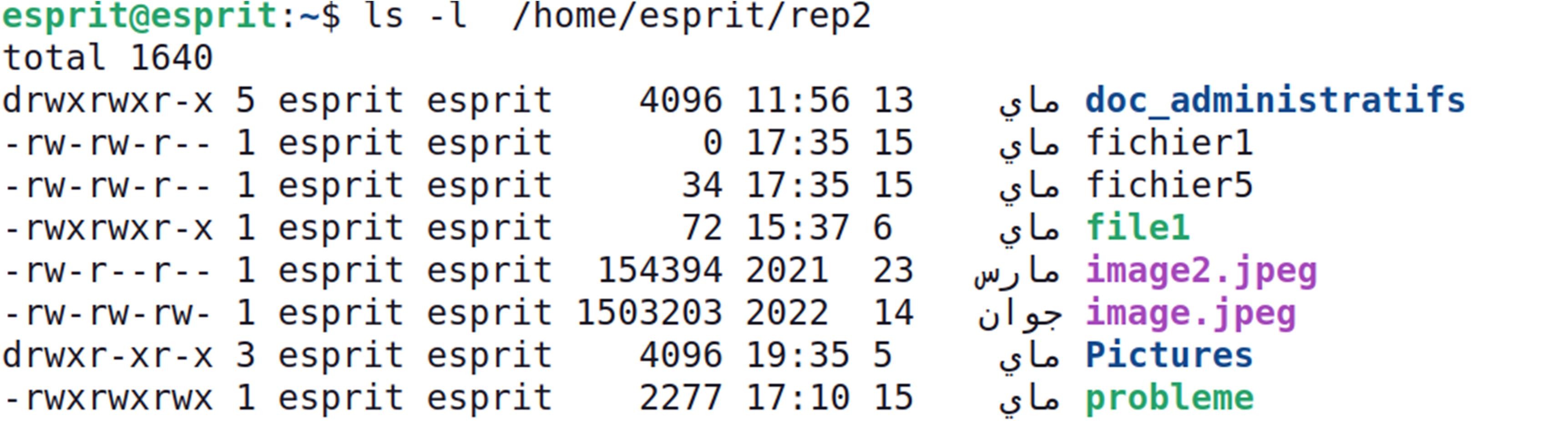
…………………………………………………………………………………………………………

…………………………………………………………………………………………………………

1. Donner la commande qui permet de compter à la fois tous les fichiers images d’extension « .jpeg ». (0,25pt)

…………………………………………………………………………………………………………

1. Le résultat de la commande ls –l /home/esprit/rep2 est la suivante :



Donner les résultats de l’exécution des commandes suivantes : (0,75pt)

a. ./ Gestion\_rep -b /home/esprit/rep2

……………………………………………………………………………………………………….

……………………………………………………………………………………………………. … b. ./ Gestion\_rep -d /home/esprit/rep2

……………………………………………………………………………………………………….. c. ./ Gestion\_rep

……………………………………………………………………………………………………….

## Bon Travail

