EXAMEN

Élément INF3B1 - Système

L2 Informatique

13 décembre 2021 - 1H30

Documents autorisés

Université de Caen Normandie

- Donner aux réponses le maximum de précision possible.
- Les réponses aux questions sont à donner uniquement sur les emplacements prévus dans ce document. Pour chaque question, le nombre de points (approximatif) est donné dans le cadre situé à droite et au début de la question.
- Le candidat indiquera sur ce document (voir emplacement en bas de cette page) son numéro de place. En outre, il devra en début d'épreuve remplir une copie sur laquelle il indiquera ***aussi*** son numéro de place. Il écrira également son nom dans le coin de la copie qu'il cachera par collage après signature de la feuille d'émargement.
- Cet examen comporte 19 questions.

Bon travail.

Numéro de place :

1 Bash (5 points)

Dans un répertoire de travail que l'on nommera *workspace*, on dispose d'un certain nombre de fichiers de différents types : texte (avec l'extension .txt), pdf (avec l'extension .pdf) et images (avec les trois extensions .png, .jpg et .gif).

Soit le script `script.sh` suivant :

```
#!/bin/bash

if [ $# -ne 1 ] ; then
    echo "error usage : $0 <input>" 1>&2
    exit 0
fi
```

Question 1. Que fait ce script ?

Réponse:

1

Question 2. Si le fichier `script.sh` se situe dans le répertoire *workspace* et que la commande `./script.sh pdf` est exécutée dans le terminal, celui-ci indique :

1

```
bash: ./script.sh: Permission non accordée
```

Que signifie ce message d'erreur ?

Réponse:

Donnez la commande permettant d'éviter ce message d'erreur.

Réponse:

Question 3. Écrire une boucle, que l'on pourrait ajouter à ce script, afin d'afficher tous les noms des fichiers du répertoire *workspace* ayant l'extension donnée en paramètre (par exemple la commande `./script.sh pdf` permettra d'afficher tous les fichiers du répertoire *workspace* ayant l'extension .pdf). 1

Réponse:

Question 4. Supposons que l'on ajoute les lignes suivantes au script : 1

```
mkdir $1
for i in *.$1; do
        cp $i $1/$(basename $i .$1)_copie.$1
done
```

Que fait le script si l'on exécute la commande `./script.sh pdf`

Réponse:

Question 5. Par quoi remplacer `X` dans ce qui suit afin de supprimer les répertoires *png*, *jpg* et *gif* du répertoire courant ? 1

```
for i in X; do rm -rf $i; done
```

Réponse:

---

2 Expressions régulières (5 points)

Question 6. Dans un répertoire contennant plusieurs fichiers *.txt* on cherche les lignes qui contiennent le mot *maison* mais qui ne contiennent pas *maisons* ou *maisonnette*. Donner la ligne de commande correspondante. 1

Réponse:

Question 7. Dans un fichier *formulaire.txt*, certaines lignes contiennent **uniquement** une adresse mail se terminant par **.fr**. Donner la ligne de commande qui, appliquée au fichier *formulaire.txt*, ne conserve que le nom de domaine des adresses mail (la chaîne de caractères entre **@** et **.fr**). 1

Réponse:

Question 8. Soit un fichier *adn.txt* contenant une séquence d'ADN par ligne (c'est à dire des mots écrits avec l'alphabet A-C-G-T). 2

- Écrivez une ligne de commande permettant de **ne conserver** que les morceaux de séquences d'ADN compris entre une occurrence de **CGGGCAA** et une occurrence de **TAAAAAAA**.

  Réponse:

- Écrivez une ligne de commande permettant de **supprimer de la ligne** les morceaux de séquence d'ADN compris entre une occurrence de **CGGGCAA** et une occurrence **TAAAAAAA**.

  Réponse:

Question 9. Lorsque l'on effectue la commande **ls -l**, on obtient un résultat similaire à celui-ci.

1

```
-rw-rw-r-- 1 david david  111 nov.   4  2021 adn.txt
drwxr-xr-x 2 david david 4096 nov.   7  2021 Examen
-rw-rw-r-- 1 david david 6876 mai   20 15:32 examen21.txt
-rw-rw-r-- 1 david david   94 nov.   4  2021 formulaire.txt
drwxrwxr-x 2 david david 4096 mai   24 09:14 Rattrapage
```

Donner une ligne de commande permettant de filtrer le résultat d'un **ls -l** afin d'extraire uniquement le **nom des répertoires**.

Réponse:

---

3 Awk (5 points)

Nous possédons un fichier nommé *owl.txt*, ce fichier regroupe pour toute une saison de l'Overwatch League les statistiques des joueurs. La première ligne de ce fichier indique la signification des colonnes.

```
player_name team_name role damage_done damage_taken
Doha Dallas_Fuel DPS 7181.49 5083.61
FEARLESS Dallas_Fuel Tank 8177.83 17382.85
Fielder Dallas_Fuel Support 5178.88 4707.15
kevster Los_Angeles_Gladiators DPS 10583.61 3365,22
shu Los_Angeles_Gladiators Support 5600.69 3436.66
skewed Los_Angeles_Gladiators Support 4684.58 3673.22
Dallas_Fuel Paris_Eternal Support 3498.41 5572.05
Onigod Paris_Eternal DPS 4350.56 3607.19
ELLIVOTE Paris_Eternal Tank 3824.65 9691.57
```

Le caractère de séparation des champs du fichier *owl.txt* est l'espace " ".

Question 10. Expliquer ce que fait la commande `awk "/Paris\_Eternal/ { print }" owl.txt` ?

1

Réponse:

Question 11. Expliquer ce que fait la commande `awk "/^[A-Z]/ { print }" owl.txt`

Réponse:

1

Question 12. Quelle commande AWK permet d'afficher **uniquement le nom des joueurs** appartenant à l'équipe *Dallas_Fuel* dans le fichier *owl.txt*?

1

Réponse:

Question 13. Quelle commande AWK permet d'afficher le nom des joueurs ayant fait des dégâts supérieurs à 6400 et un nombre de dégâts reçus inférieurs à 3000 dans le fichier *owl.txt*?

1

Réponse:

Question 14. Quelle commande AWK permet de compter le nombre de joueurs de rôle Support dans le fichier *owl.txt*?

1

Réponse:

---

4 Processus (5 points)

On précise les éléments suivants :

- la commande `sleep` déclenche une attente *passive* du système en endormant le processus courant. Pendant cette attente, les signaux reçus sont mis en attente de traitement. À l'issue de l'attente, seul le dernier signal reçu est traité.
- la commande `wait` attend la terminaison d'une tâche en arrière plan. Pendant ce temps d'attente, les signaux reçus à l'aide de `trap` peuvent être traités.
- `date +%H:%M:%S-%N` affiche la date courante au format heures:minutes:secondes-nanosecondes. Par exemple,

```
$ date +%H:%M:%S-%N
14:55:09-180115609
$ date +%H:%M:%S-%N
14:55:36-147274915
```

On donne le script `time.sh` suivant :

```
1  #!/bin/bash

   function printTime {
       date +%H:%M:%S-%N
5  }

   trap 'printTime' 16

   echo "l'horloge a pour identifiant " $$
10
   while true; do
       sleep 1 &
       wait $!
   done
15
```

Question 15. À quoi sert la ligne 1 du script `time.sh` ?

Réponse:

1

Question 16. À quoi sert la ligne 7 du script `time.sh` ?

Réponse:

1

Question 17. À quoi sert le double dollar `$$` de la ligne 9 du script `time.sh` ?

Réponse:

1

Question 18. En vous appuyant sur la signification de l'expression `$!` à la ligne 13 du script `time.sh`, expliquer ce que le script effectue dans le corps de la boucle `while`.

Réponse:

1

Question 19. Quel est l'intérêt d'avoir écrit le corps de la boucle comme cela ? Quelle différence cela aurait-il fait si on avait simplement écrit

1

```
while true; do
    sleep 1
done
```

Réponse:
