contrTP3

Exercices:

sed:

1 - afficher la premiere ligne d'un fichier1 avec la commande sed

sed -n '1p' fichier1

2 - afficher la ligne 3 et 5 d'un fichier

sed -n '3p;5p' fichier1

3 - quelle est la différence entre -n, -i et sans option

l'option **-n** est utilisée pour désactiver l'affichage par défaut de **sed**, l'option **-i** est utilisée pour modifier directement le fichier d'entrée et sans option, **sed** affichera toutes les lignes du fichier par défaut sans apporter de modifications au fichier.

4 effacer toutes les lignes qui commencent par le mot password

5 faire un script qui prend en argument un fichier et remplacer password=true avec password=false et inversement

filename=$1

sed -i 's/password=true/password=false/g' "$filename"

sed -i 's/password=false/password=true/g' "$filename"

echo "Password value switched in file: $filename"

sed -i 's/password=true/password=false/g;s/password=false/password=true/g' "fichier1"

echo "Done."

**awk:**

1 afficher la deuxième et 3ème colonne uniquement du fichier2 avec awk

awk '{print $2,$3}' fichier2

2 faire la même chose avec le fichier3

awk '{print $2,$3}' fichier2

3 en quoi consiste les variables NR, NF et END

NF Number of fields in the current record.

NR Ordinal number of the current record.

END { final command(s) }' file

4 ajouter le numero de ligne au fichier2

awk '{print NR, $0}' "$1" > "$1.new"

mv "$1.new" "$1"

5 en tête de page, afficher le nombre de colonnes et en fin de page afficher le nombre de ligne

* Vérifier si un nom de fichier a été fourni en argument

if [ $# -eq 0 ]; then

echo "Usage: $0 filename"

exit 1

fi

(permet d’afficher un message d’erreur – utile pour la sécurité )

* **Réponse juste :**

**Echo « number of columns : $(awk ‘NR==1{print NF}’ « fichier2 »)**

**Echo « number of lines : $(awk ‘END{print «number of lines », NR}’ « fichier2 »)**

crontab

1 Utiliser une commande crontab pour lancer un de vos scripts à une heure prédéfinie.

Pour utiliser une commande **crontab** pour lancer un de vos scripts à une heure prédéfinie, suivez les étapes suivantes :

1. Ouvrez une session terminal et tapez la commande **crontab -e** pour ouvrir le fichier de configuration de la crontab.
2. Si c'est la première fois que vous ouvrez la crontab, il vous sera demandé de choisir un éditeur de texte par défaut. Choisissez celui avec lequel vous êtes le plus à l'aise (par exemple, Nano, Vim, Emacs).
3. Une fois le fichier de configuration ouvert, ajoutez une nouvelle ligne à la fin pour définir le moment où vous souhaitez que votre script soit exécuté. Par exemple, pour lancer votre script chaque jour à 17h00, ajoutez la ligne suivante :

javascriptCopy code

**0 17 \* \* \*** /chemin/vers/votre/script.sh

* **0** représente les minutes (dans cet exemple, votre script sera exécuté à la première minute de l'heure)
* **17** représente l'heure
* **\*** représente tous les jours du mois, tous les mois et tous les jours de la semaine
* **/chemin/vers/votre/script.sh** représente le chemin absolu vers votre script (pensez à rendre votre script exécutable en utilisant la commande **chmod +x /chemin/vers/votre/script.sh**).

1. Enregistrez le fichier de configuration et fermez l'éditeur de texte.
2. La commande **crontab** va alors automatiquement prendre en compte les nouvelles entrées dans la configuration. Vous pouvez vérifier que votre entrée a bien été ajoutée en tapant **crontab -l**, qui liste les tâches actuellement programmées dans la crontab.

Votre script sera désormais exécuté automatiquement à la date et l'heure prédéfinies.

2 Utiliser une commande crontab pour lancer un de vos scripts à intervalle de 2 mi

**\*/2 \* \* \*** /chemin/vers/votre/script.sh