

TP SHELL UNIX 2

Rechercher des fichiers

But : rechercher des fichiers avec `find`, `whereis` et `locate`.

1. Affichez tous les fichiers ayant une taille inférieure à 400 octets et ayant les droits 644.
2. Affichez tous les fichiers dans votre répertoire personnel ayant une taille inférieure à 400 blocs.
3. Listez en format long tous les fichiers du système vous appartenant modifiés il a plus de 7 jours.
4. Listez et affichez en format long les fichiers dans votre répertoire personnel qui ont comme propriétaire `guest` ou qui ont une taille entre 512 et 1024 octets, inclus.
Le petit piège réside ici dans le "inclus". Si vous indiquez `+512c`, les fichiers de 512 octets sont exclus. Vous devez modifier les bornes en conséquence.
5. Recherchez tous les fichiers vides du système vous appartenant et tentez de les supprimer.
Utilisez les paramètres `-empty` et `-exec` pour exécuter un `rm` sur chaque fichier trouvé.
6. Indiquez où se situe la commande binaire `ls`.
Utilisez la commande `whereis` pour cela :

Les redirections

But : manipuler les redirections de canaux.

7. La commande `find` / retourne beaucoup d'erreurs si elle est utilisée par un simple utilisateur à cause d'un problème de droits. Évitez les messages d'erreurs en les redirigeant vers un « trou noir » :
8. Dans le cas précédent et malgré les erreurs, vous avez encore accès à beaucoup d'emplacements et la liste qui s'affiche est très longue et donc inexploitable. Placez cette liste dans un fichier appelé `résultat`.
9. Maintenant, plus rien ne s'affiche. En fin de compte, pour savoir pourquoi vous ne pouvez pas accéder à certains répertoires vous voulez aussi obtenir les messages d'erreurs dans le fichier `résultat`, avec la liste des fichiers. Faites une redirection du canal d'erreur standard dans le canal de sortie standard :
10. Plus rien ne s'affiche. Vous voulez les deux : un fichier et l'affichage des résultats sur écran. La commande `tee` s'utilise avec un tube et permet de récupérer un flux sortant, de le placer dans un fichier et de ressortir ce flux comme si de rien n'était :

Les filtres et utilitaires

But : manipuler les filtres et utilitaires sur un fichier classique.

1. Le fichier `/etc/passwd` est un grand classique sous Unix. Il se compose de sept champs séparés par des « : » : `login:passwd:UID:GID:Commentaire:homedir:shell`. Récupérez la ligne de l'utilisateur `root` dans `/etc/passwd` :
2. De cette ligne, récupérez l'UID de `root` :
3. Comptez le nombre d'utilisateurs contenus dans ce fichier à l'aide d'une redirection en entrée :
4. Un peu plus compliqué : récupérez la liste des `GID`, triez-les par ordre croissant et supprimez les doublons :
5. De là, extrapolez le nombre de groupes différents utilisés :
6. Convertissez tous les logins en majuscules :
7. Isolez maintenant la huitième ligne de `/etc/passwd`

Les processus

But : gérer les processus.

1. Lancez le processus `sleep 1000` en arrière-plan. Récupérez son PID. (le PID varie bien entendu).
2. Remplacez ce processus en avant-plan, puis stoppez-le (ne le tuez pas) et remplacez-le en arrière-plan.
3. Indiquez les détails de ce processus :
4. Modifiez la priorité de ce processus ; passez-la à un facteur 10 :
5. Listez à nouveau le détail de ce processus mais au format long. Regardez la valeur de la colonne `NI` :
6. Envoyez le signal 15 à ce processus. Ceci va le terminer.

Programmation Shell Niveau 1

But : dans un fichier texte nous avons les lignes suivantes :

1 3

5 7

12 19

...

Écrivez un script qui accepte ce fichier comme paramètre, qui le lit et pour chacune de ses lignes calcule la somme des deux nombres et l'affiche sous la forme suivante :

$$1 + 3 = 4$$

$$5 + 7 = 12$$

$$12 + 19 = 31$$

...

1. Vérifiez en début de script que le nombre de paramètres passé au script est égal à 1 et que ce paramètre est bien un fichier.
2. Initialisez une variable à 0 qui contiendra le total de chacune des lignes.
3. Le fichier doit être lu ligne à ligne ; écrivez une boucle qui lit une ligne tant que la fin du fichier n'est pas atteinte :
4. Dans la boucle, récupérez les deux valeurs des lignes, le séparateur est l'espace. Placez les deux valeurs dans les variables c1 et c2 :
5. Additionnez ces deux valeurs et placez le résultat dans result. Affichez result.