CSC 3001: examen 1ère session 2008-2009

Consignes :

- Toute entorse à ces consignes donnera lieu à des pénalités.
- Documents autorisés : uniquement ceux fournis dans le cadre du module : polycopié papier, pages web du cours en ligne, et illustrations interactives et QCM.
- Répondez aux questions sur la copie et exécutez les commandes sur votre compte.
- En plus des règles usuelles des contrôles, il est *strictement interdit* de :
  - Commencer le contrôle avant le signal de démarrage de l'enseignant
  - Continuer le contrôle après le signal d'arrêt de l'enseignant
  - Naviguer sur le web ailleurs que dans les pages du module
  - Communiquer de l'information par tout moyen électronique : courrier, fichier...
- Un enseignant est présent pour vous aider dans la compréhension du sujet.

Objectif général: comparer les caractéristiques de différentes machines comme la durée de fonctionnement, versions de logiciels...

1/ Préparation

Objectif: préparation de l'environnement de travail.

1. Créez un repertoire `Private` dans votre répertoire de connexion.
   *** Donnez la commande.

   ```
   mkdir Private
   ```

2. Supprimez les droits d'accès pour toute autre personne que vous sur ce répertoire `Private`.
   *** Donnez la commande.

   ```
   chmod go= Private
   ```

3. Déplacez-vous dans ce répertoire `Private`.
   *** Donnez la commande.

   ```
   cd Private
   ```

2/ Durée de fonctionnement

Objectif: le but de cette section est de récupérer la durée de fonctionnement (*uptime* en anglais), d'une machine distante.

1. Accédez à la documentation de la commande `uptime`.
   *** Donnez la commande.

   ```
   man uptime
   ```

2. Exécutez la commande `uptime`.

   ```
   uptime
   ```

3. Affichez uniquement le champ indiquant la durée de fonctionnement de la machine en une seule commande.
   *** Donnez la commande.

   ```
   uptime | cut -c14-18
   ```

   Remarquez que les informations qui nous intéressent occupent les caractères 14 à 18. Aidez-vous des pages du manuel.

4. Quelle est la durée de fonctionnement de votre machine?
   *** Votre réponse.

   ```
   3:48
   ```

5. En une seule commande, exécutez la commande `uptime` sur la machine indiquée au tableau.
   *** Donnez la commande.

   ```
   ssh b03-02 uptime
   ```

6. Créez la variable machine ayant pour valeur le nom de la machine indiquée au tableau.
   *** Donnez la commande.

   ```
   machine=b04-03
   ```

7. Ré-écrivez la commande précédente permettant d'exécuter la commande `uptime` à distance en utilisant la variable `machine`
   *** Donnez la commande.

   ```
   ssh $machine uptime | cut -c14-18
   ```

   ATTENTION: si vous n'arrivez pas à faire cette question, demandez de l'aide à l'encadrant qui notera sur votre copie qu'il vous a aidé(e).

3/ Automatisation

Objectif: le but de cette section est d'automatiser par un script la récupération de la durée de fonctionnement de machines distantes.

1. Copiez le fichier `~silber/Public/UX/Examen_2008/Scripts/script1.bash` dans votre répertoire de travail.
   *** Donnez la commande.

   ```
   cp ~silber/Public/UX/Examen_2008/Scripts/script1.bash .
   ```

2. Affichez le fichier `script1.bash`.
   *** Donnez la commande.

   ```
   cat script1.bash
   ```

3. Remplacez la chaîne de caractères MACOMMANDE_A_REMPLACER par la dernière commande de la partie précédente (question 2.7). Expliquez la marche à suivre.
   *** Votre réponse.

   ```
   Je lance mon éditeur de texte préféré, par exemple emacs. J'édite le
   fichier script1.bash pour remplacer la chaîne de caractères. Je le
   sauvegarde et quitte l'éditeur de texte.
   ```

4. Ce script exécute cette commande pour une liste de machines. En lisant le script `script1.bash`, donnez les machines concernées.
   *** Votre réponse.

   ```
   ssh1 b02-02 b03-03 b04-04
   ```

5. Quels sont les droits nécessaires pour que vous exécutiez le script `script1.bash`? Précisez lesquels et pour qui.
   *** Votre réponse.

   ```
   x pour le propriétaire du fichier script1.bash
   ```

6. Positionnez ces droits sur le fichier `script1.bash`.
   *** Donnez la commande.

   ```
   chmod u+x script1.bash
   ```

7. Exécutez le script.
   *** Donnez la commande.

   ```
   script1.bash
   ```

8. L'exécution de ce script vous demande t-elle des informations supplémentaires? Si oui, précisez lesquelles. Si non, expliquez pourquoi.
   *** Votre réponse.

   ```
   Oui, l'exécution du script nécessite de saisir mon mot de passe pour
   chacune des machines sur laquelle je me connecte. Oui, j'ai mis
   l'authentification par clé publique et l'exécution du script
   nécessite de saisir la phrase d'authentification. Non, j'ai mis
   l'authentification par clé publique et je n'utilise pas de phrase
   d'authentification.
   ```

9. Re-exécutez le script en conservant le résultat dans le fichier `resultat_uptime` dans votre répertoire de travail.
   *** Donnez la commande.

   ```
   script1.bash > resultat_uptime
   ```

ATTENTION: si vous n'arrivez pas à faire cette question, demandez de l'aide à l'encadrant qui notera sur votre copie qu'il vous a aidé(e).

4/ Analyse du fichier resultat_uptime

Objectif: le but de cette section est d'analyser les durées de fonctionnement des machines distantes.

1. Affichez le contenu du fichier `resultat_uptime`.
   *** Donnez la commande.

   ```
   cat resultat_uptime
   ```

2. Y a t-il des machines démarrées depuis plus d'un jour? Si oui, combien?
   *** Votre réponse.

   ```
   Peut-être une ou deux.
   ```

3. Affichez uniquement les durées de fonctionnement inférieures à un jour. Enregistrez ce résultat dans un fichier appelé `moins1jour` dans votre répertoire de travail.
   *** Donnez la commande.

   ```
   grep -v d resultat_uptime > moins1Jour s'il existe des machines
   démarées depuis plus d'un jour
   cp resultat_uptime moins1Jour sinon
   ```

4. Affichez le fichier `moins1jour` trié par ordre croissant par rapport à la durée de fonctionnement des machines en heure(s).
   *** Donnez la commande.

   ```
   sort -t: -n moins1jour
   ```

5/ Processus gourmand

Objectif: identifier un processus monopolisant les ressources.

1. Lancez le programme `~silber/Public/UX/Examen_2008/Programmes/ProcProg` en arrière-plan.
   *** Donnez la commande.

   ```
   ~silber/Public/UX/Examen_2008/Programmes/ProcProg &
   ```

2. Donnez son numéro de processus et son numéro de job et les commandes correspondantes
   *** Donnez la commande.

   ```
   ps u | grep ProcProg
   jobs
   ```

3. À partir des commandes précédentes, indiquez les pourcentages de CPU et de mémoire utilisés par ce processus.
   *** Votre réponse.
4. Affichez sur une seule ligne le processus consommant le plus de CPU.
   *** Donnez la commande.

   ```
   ps u | sort -r -k 3,3 | head -2  | tail -1
   ```

   Suggestion : faire un tri sur le champ CPU dans le résultat de la commande précédente

5. Détruisez ce processus
   *** Donnez la commande.

   ```
   killall ProcProg
   ```

6/ Récupération d'autres informations

Objectif: le but de cette section est d'utiliser le script précédent pour récupérer d'autres informations sur les machines distantes.

6.1/ Version de navigateur Web

1. En utilisant la page manuelle, déterminez la commande permettant de connaître la version de Firefox utilisée
   *** Donnez la commande.

   ```
   firefox --version
   ```

2. Quelle est la version de firefox?
   *** Votre réponse.

   ```
   Mozilla Firefox 3.0.2
   ```

3. Seules les trois premières informations du résultat de la commande nous intéressent (Mozilla Firefox XX.XX.XX). Donnez la commande permettant d'afficher uniquement ces trois informations.
   *** Donnez la commande.

   ```
   firefox --version | cut -d',' -f1
   ```

4. Remplacez dans le script `script1.bash` la commande `uptime` par la commande précédente.
5. Exécutez le script `script1.bash` et récupérez le résultat dans le fichier `resultat_firefox`.
   *** Donnez la commande.

   ```
   script1.bash > resultat_firefox
   ```

6. Pour chaque version différente, affichez le nombre de machines correspondant.
   *** Donnez la commande.

```
sort resultat_firefox | uniq -c
```

7. Affichez le nombre de versions différentes.
   *** Donnez la commande.

```
sort resultat_firefox | uniq | wc -l
```

6.2/ Nom de la machine

8. Quelle est la commande permettant de connaître le nom de la machine?
   *** Donnez la commande.

```
hostname
```

9. Remplacez dans le script `script1.bash` la commande par la commande précédente.
10. Exécutez le script `script1.bash` et récupérer le résultat dans le fichier `resultat_hostname`.
    *** Donnez la commande.

```
script1.bash > resultat_hostname
```

11. Les fichiers `moins1jour` et `resultat_hostname` ont chacun une colonne de resultat. Concaténez ces deux fichiers en un seul fichier appelé `moins1jour_complet` possédant 2 colonnes avec, sur chaque ligne, le nom de la machine suivi de la durée de fonctionnement.
    *** Donnez la commande.

```
paste resultat_hostname moins1jour > moins1jour_complet
```

12. Triez le fichier `moins1jour_complet` par durée de fonctionnement décroissante.
    *** Donnez la commande.

```
sort -nr -k 2 moins1jour_complet
```

13. En reprenant la commande précédente, affichez la ligne correspondant à la durée de fonctionnement la plus élevée.
    *** Donnez la commande.

```
sort -nr -k 2 moins1jour_complet | head -1
```

14. A quelle machine cela correspond t-il?
    *** Votre réponse.

```
b03-08
```