

Pratique, Installation et Utilisation des machines

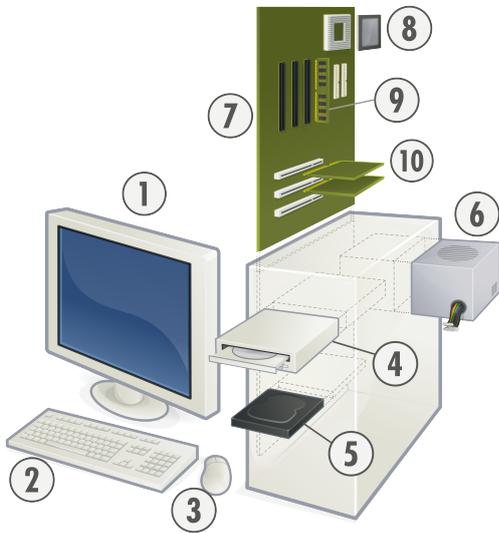
TP 10 - Révisions (Linux/Bash)

02 décembre 2016

Complétez les lignes en indiquant chaque fois la commande à taper ou la réponse appropriée.

1 - Connaitre sa machine

Les différents éléments visibles sur ce schéma sont :



2. _____

3. _____

4. _____

5. _____

6. _____

7. _____

8. _____

9. _____

1. _____

10. _____

Quand je veux installer Linux je fais les opérations suivantes (complétez et numérotez dans l'ordre) :

(___) Je démarre le PC avec la _____

(___) Je crée une clef _____ d'installation GNU/Linux.

(___) J'installe _____.

(___) Je configure le _____ pour démarrer sur une clef USB

(___) Je _____ mes données sous Windows.

(___) Je télécharge une _____ Linux

Linux est un s_____ d'_____ m_____.

Quand la mémoire est utilisée à 100%, pour pouvoir continuer à fournir de la mémoire aux nouvelles applications,

Linux recopie une partie de la mémoire dans le _____, c'est une _____ spéciale située à la fin du _____.

2 - Se déplacer, manipuler les fichiers

Pour me déplacer dans mon répertoire personnel : \$ _____

Pour créer un dossier nommé `tp10` : \$ _____

Pour me déplacer dans le répertoire `tp10` : \$ _____

Pour créer un chemin `tp10/script/data/tmp` : \$ _____

Pour me déplacer dans `tp10/script/data/tmp` : \$ _____

Pour créer les fichiers `un`, `deux` et `trois` : \$ _____

Pour supprimer le fichier `trois` : \$ _____

Pour revenir au niveau précédent : \$ _____

Pour savoir où je suis : \$ _____

Pour lister les fichiers présents dans ce répertoire : \$ _____

Pour lister les fichiers en affichant les fichiers cachés : \$ _____

Pour supprimer le répertoire `tmp` qui contient des fichiers : \$ _____

Pour remonter de deux niveaux d'un coup : \$ _____

Pour effacer le répertoire vide `script/data` : \$ _____

Pour créer le fichier `"mon texte"` : \$ _____

Pour renommer ce fichier `mon_texte` : \$ _____

Pour visualiser le contenu du fichier `texte` `~/bashrc` : \$ _____

Pour connaître le type du fichier `/bin/date` : \$ _____

3 - Permissions, propriété, chemins d'accès

Pour lister les fichiers en affichant le détail des droits : \$ _____

Sous Linux, le superutilisateur ou administrateur se nomme _____ et son groupe est _____.

Cette ligne : `lrwxrwxrwx 1 guss users 10 jan. 1 12 :24 cours -> Documents/fac/python/`

indique que le fichier nommé _____ est un _____ qui appartient à _____
et au groupe _____.

Cette ligne : `-rw-r r 1 usr1 usr1 19554 dec. 30 14 :22 exercices.odt`

Indique que le fichier est lisible par _____ modifiable par _____ et exécutable par _____.

Les informations concernant les différents groupe sont stockées dans le fichier _____, les informations concernant les utilisateurs dans le fichier _____ et les mots de passe dans le fichier _____.

Pour se donner le droit d'exécution sur le fichier `script.sh` : \$ _____

Symbole pour donner un droit à l'utilisateur ____, le groupe ____, les autres ____, tout le monde ____

Symbole pour ajouter un droit ____, pour supprimer un droit ____, donner que le droit indiqué ____

Symboles correspondants aux droits (en chiffre) : lecture __ (___) écriture __ (___) exécution __ (___)

Équivalent de `chmod u=rwx script.sh ; chmod go=rx script.sh` en chiffres :

\$ _____

Pour donner `script.sh` à l'utilisateur `martin` et au groupe `users` :

\$ _____

Pour lancer `script.sh` qui est dans le répertoire courant : \$ _____

Pour lancer `date` qui est dans le répertoire `/bin` : \$ _____

Je n'ai pas besoin de préciser le chemin pour les commandes du système parce que _____

L'arborescence d'un système de fichier GNU/Linux débute par _____

Exemple de chemin relatif : _____

Exemple de chemin absolu : _____

4 - Liens, redirections, pipes

Pour créer un lien vers `/home/arthur/Document` : \$ _____

Pour rediriger la sortie standard : `echo "bonjour" _____ fichier`

Pour rediriger la sortie d'erreur : `apt-get update _____ fichier`

Pour rediriger les deux sorties : `find / -name "*log*" _____ fichier`

Pour rediriger les erreurs au même endroit : `./script.sh > fichier _____`

Pour rediriger la sortie standard au même endroit : `date 2> fichier _____`

Pour rediriger la sortie standard sans écraser le contenu du fichier : `echo "export PS1" _____ fichier`

Pour rediriger la sortie d'erreur sans écraser le contenu du fichier : `./prog.py _____ fichier`

Pour rediriger la sortie d'une commande vers la suivante : `sort listing.txt _____ uniq _____ wc -l`

5 - Réseau

Le service réseau qui attribue les _____ comme 192.168.3.129 se nomme _____.

Le service réseau qui permet de savoir que l'_____ de Langoustine est 192.168.3.129 se nomme _____.

Pour savoir si Langoustine est accessible sur le réseau : \$ _____

Pour me connecter à distance sur langoustine avec le nom d'utilisateur usr1 :

\$ _____

6 - Processus

Pour stopper proprement un processus qui demande une saisie de texte : _____

Pour stopper en force un processus qui tourne en boucle : _____

Pour mettre un processus en pause à l'arrière plan : _____

Pour récupérer le PID d'un processus : \$ _____

Pour tuer ce processus (PID=23452) avec le signal SIGTERM : \$ _____

Pour tuer violemment ce processus (PID=23452) (perte de données, pas de nettoyage de la mémoire) :

\$ _____

Pour tuer tous les processus nommés prog : \$ _____

Le seul processus qui n'a pas de parents est _____

Un processus dont le parent ne récupère pas la valeur de retour devient un _____.

Pour examiner les ressources consommées par les processus : \$ _____

7 - Environnement

La variable PATH est une variable _____, je peux afficher sa valeur avec :

\$ _____

La variable PS1 est une variable _____, je peux afficher sa valeur avec :

\$ _____

Les processus enfants reçoivent les variables _____ du processus parent.

Pour créer une variable VAR qui contient 10 : \$ _____

Pour placer la variable VAR dans l'environnement : \$ _____

Pour détruire la variable VAR : \$ _____

8 - Script Shell

Complétez les commentaires et les caractères manquants remplacés par des `_`. Ajoutez les instructions pour créer le répertoire de sauvegarde si il n'existe pas.

```
-----
# Basic script bash to backup files in a given directory
# Usage backup.sh [PATH of dir. you want backup] [PATH of dir. where to backup]
# John Doe september 1981

# Version : 0.1
DATE='_____ +%d-%m-%y'

#
if [ __ != 2 ]; then
    _____ "Usage : backup.sh [origin PATH] [backup PATH]"
    exit 1
--

#
__ [ ! -d $1 ]; _____
    echo "It seems __ is not a directory"
    _____ 1
-----
    _____ "Ok directory found ; we can go further"
fi

#
_____ -n 1 __ "Do i have to compress : [y/n]? " COMPRESS ; echo
if [[ $_____ =~ ^[Yy] ]]_ then
    echo "Ok compress with name backup.$DATE.tar.gz "
    _____ [[ ! _____ != ^[Nn]$ ]]; then
        _____ "I beg your pardon i did not understand, you must answer by y or n"
        _____ 2
--
```

```
#
---- "You've got few seconds to cancel it by ctrl^C"
---- i __ {1..5}; do echo __; sleep 1; done
---- "so let's go"
```

```
# create the backup directory if it does not exist
```

```
#
if [[ _____ =~ ^[Yy]$ ]]; then
    archive="backup.$DATE.tar.gz"
    tar ___ $archive $1
    mv $archive __
else
    mkdir $2/$DATE
    cp -a __ __/$DATE
--
exit _
```