

Évaluation

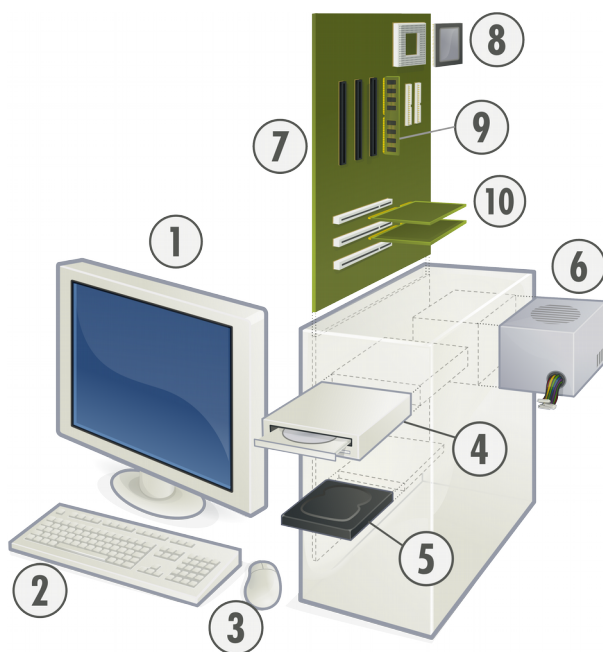
Durée : 45 minutes

Répondez directement dans ce fichier en complétant les blancs (signalés par un surlignage jaune).
Envoyez ce document complété en pièce jointe à aline.hufschmitt@iedparis8.net en précisant
« REGROUPEMENT IED JUIN 2017 » comme sujet du mail.

Nom	Prénom	Numéro étudiant

Connaitre sa machine (1 point)

Les différents éléments visibles sur ce schéma sont :



1. écran
2. clavier
3. souris
4. lecteur CD / DVD
5. disque dur
6. alimentation
7. carte mère
8. processeur
9. barrette mémoire (RAM)
10. cartes PCI

Connaitre son système (1 point)

Compléter ce texte avec le vocabulaire approprié :

Linux est un **système** d'**exploitation** **multi utilisateurs**. Si je souhaite conserver Windows sur ma machine, je peux installer une **distribution** linux en **double boot** ou bien la tester dans une **machine virtuelle**. Si j'utilise une clef LiveUSB, il sera peut-être nécessaire de configurer le **BIOS** pour **booter** dessus. 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 **RAM** dans le **SWAP** c'est une **partition** spéciale située à la fin du **disque dur**. Sous Linux, les deux **disques durs** SATA de ma machine se nomment **sda** et **sdb**. Le premier est divisé en trois **partitions** nommées **sda1** **sda2** et

sda3. Le second contient une unique **partition étendue** qui est divisée en trois **partitions logiques**. Contrairement à Linux, Windows ne s'installe que sur une **partition primaire**.

Les licences (1 point)

Si j'utilise un logiciel libre je peux jouir des 4 libertés fondamentales suivantes (complétez) :

- la liberté d'**utiliser** le logiciel
- la liberté de **copier** le logiciel
- la liberté d'**étudier** le logiciel
- la liberté de **modifier** le logiciel et de **redistribuer** les versions **modifiées**

Une des licences libres les plus connues est la licence **GNU-GPL**. Pour les documentations on peut utiliser la licence **GFDL**. Si on veut être en accord spécifiquement avec la loi française on peut utiliser la licence **CeCILL**. Une licence **copyleft** garantit que, même modifié, un morceau du programme protégé par elle ne pourra pas être redistribué sous une licence propriétaire. Le terme **open source** signifie juste que l'on peut voir comment un programme est fait, et ne doit pas être confondu avec le terme "libre".

Se déplacer, manipuler les fichiers (4 points)

Indiquez les commandes à saisir pour effectuer les opérations suivantes :

Pour me déplacer dans mon répertoire personnel : **\$ cd ~**

Pour créer un dossier nommé tp10 : **\$ mkdir tp10**

Pour créer un chemin tp10/script/data/tmp : **\$ mkdir -p tp10/script/data/tmp**

Pour créer les fichiers un, deux et trois : **\$ touch un deux trois**

Pour supprimer le fichier trois : **\$ rm trois**

Pour revenir dans le répertoire parent : **\$ cd ..**

Pour savoir où je suis : **\$ pwd**

Pour lister les fichiers du répertoire courant en affichant les fichiers cachés : **\$ ls -a**

Pour supprimer le répertoire tmp qui contient des fichiers : **\$ rm -r tmp**

Pour effacer le répertoire vide script/data : **\$ rmdir script/data**

Pour créer un fichier nommé "mon texte" : **\$ > "mon texte"**

Pour renommer le fichier "mon texte" en "mon_texte" : **\$ mv "mon texte" "mon_texte"**

Pour visualiser le contenu du fichier texte ~/.bashrc : **\$ cat ~/.bashrc** ou **\$ less ~/.bashrc**

Pour connaître le type du fichier bin/date : **\$ file bin/date**

Pour comparer deux fichiers version1.py et version2.py : **\$ diff version1.py et version2.py**

Permissions, propriété, chemins d'accès (4 points)

Complétez les blancs en indiquant les commandes à saisir ou les réponses appropriées :

Pour lister les fichiers en affichant le détail des droits : `$ ls -l`

Sous Linux, le super-utilisateur ou administrateur se nomme `root` et son groupe est `root`.

Cette ligne :

```
rw-rw-rwx 1 guss users 10 jan. 1 12 :24 cours -> Documents/fac/python/
```

indique que le fichier nommé `cours` est un `lien symbolique` qui appartient à `guss` et au groupe `users`.

Cette ligne :

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

Indique que le fichier est lisible par `tous` modifiable par `l'utilisateur usr1` et exécutable par `l'administrateur root uniquement`.

Les informations concernant les différents groupes sont stockées dans le fichier `/etc/groups`, les informations concernant les utilisateurs dans le fichier `/etc/passwd` et les mots de passe dans le fichier `/etc/shadow`.

Pour donner à l'utilisateur le droit d'exécution sur le fichier `script.sh` sans modifier les autres droits : `$ chmod u+x script.sh`

Équivalent de `chmod u=rwx script.sh ; chmod go=rx script.sh` en chiffres :
`$ chmod 755 script.sh`

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

```
$ chown martin::users script.sh
```

Pour lancer `script.sh` qui est dans le répertoire courant : `$./script.sh` ou `sh script.sh`

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

La base de l'arborescence d'un système de fichier GNU/Linux est `/`

Je n'ai pas besoin de préciser le chemin pour lancer les commandes du système parce que `le chemin vers ces commandes est dans la variable d'environnement PATH`

Exemple de chemin relatif : `../document/toto.sh`

Exemple de chemin absolu : `/usr/bin/env`

Liens, redirections, pipes (2 points)

Complétez les blancs en indiquant les commandes à saisir ou les mots appropriés :

Pour créer un raccourcis vers /home/arthur/Document :

```
$ ln -s /home/arthur/Document doc
```

Pour rediriger la sortie standard : `$ echo "bonjour" > fichier`

Pour rediriger la sortie d'erreur : `$ apt-get update 2> fichier`

Pour rediriger les deux sorties : `$ find / -name "*log*" &> fichier`

Pour rediriger les erreurs au même endroit : `$./script.sh > fichier 2>&1`

Pour rediriger la sortie standard au même endroit : `$ date 2> fichier 1>&2`

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 2>> fichier
```

Pour rediriger la sortie d'une commande vers la suivante :

```
$ sort listing.txt | uniq | wc -l
```

Processus (2 points)

Complétez les blancs en indiquant les raccourcis claviers ou les commandes appropriées :

Pour stopper proprement un processus qui demande une saisie de texte : `Ctrl + D`

Pour stopper en force un processus qui tourne en boucle : `Ctrl + C`

Pour mettre un processus en pause à l'arrière plan : `Ctrl + Z`

Pour récupérer le PID d'un processus : `$ ps -a` ou `top`, `atop`, `htop`

Pour tuer ce processus (PID=23452) avec le signal SIGTERM : `$ kill 23452`

Pour tuer violemment un processus planté (PID=23452) (perte de données, pas de nettoyage de la mémoire) : `$ kill -9 23452` ou `$ kill -SIGKILL 23452`

Pour tuer tous les processus nommés `prog` : `$ killall prog`

Pour examiner les ressources consommées par les processus : `$ top`

Variables et contexte des processus (2 points)

Complétez les blancs avec les mots ou les commandes appropriées :

La variable `PATH` est une variable de l'`environnement` je peux afficher sa valeur avec :

```
$ echo $PATH
```

 ou

```
$ printenv PATH
```

La variable `PS1` est une variable `locale` je peux afficher sa valeur avec : `$ echo $PS1`

Les processus enfants reçoivent les variables `d'environnement` de leur processus parent.

Pour créer une variable `VAR` qui contient 10 : `$ VAR=10`

Pour placer la variable VAR dans l'environnement : `$ export VAR`

Pour détruire la variable VAR : `$ unset VAR`

Script Shell (2 points)

Écrire une boucle pour faire 3 copies de `document.txt` (situé dans le répertoire courant) nommées `doc1.txt`, `doc2.txt` et `doc3.txt`.

```
$ for i in $(seq 1 3) ; do cp document.txt doc$i.txt ; done
```

Écrire une commande avec `grep` qui va filtrer les lignes commençant par 3 lettres entre `a` et `f` dans le fichier `log.txt` :

```
$ grep -E "^[a-f]{3}" log.txt
```

Expression régulière (1 point)

Quel mot est capturé par l'expression régulière suivante ? (cocher la ou les bonnes réponses)

```
^fr[a-m]{3}[^i][^s]*$
```

- ☐ framboise.
- ☐ fromage
- ☒ frigidaire
- ☐ fricassée