

Survivre devant une machine Linux

Table des matières

Objectifs	3
Introduction	4
I - Cours	5
1. Introduction à Linux	5
1.1. Linux en deux mots	5
1.2. Utiliser Linux	6
1.3. Le terminal	6
1.4. Lancer des applications sous Linux	7
1.5. Commandes de bases sous Linux : cd, ls, mkdir, rm, find, cat, nano... ..	7
2. Utiliser une machine Linux	9
2.1. Connexion à un serveur Linux avec SSH	9
2.2. Affichage des droits sous Linux (ls -al)	10
2.3. Gestion des droits sous Linux	11
2.4. Gestion des processus sous Linux (ps et kill)	13
2.5. Gestion de mémoires amovibles sous Linux (df -h et umount)	13
2.6. Quelques utilitaires sous Linux	14
II - Exercices	16
1. Découverte de la ligne de commande sous Linux	16
1.1. Exercice : man man	16
1.2. Exercice	16
1.3. Exercice : ls	17
1.4. Exercice : cd pwd mkdir touch	17
III - Défi	19
1. Exercice : Fichier caché	19
IV - Complément	20
1. Administrer une machine Linux (introduction)	20
1.1. Agir en tant que root (su / sudo)	20
1.2. Installation d'applications sous Debian et Ubuntu (apt-get)	21
1.3. Gestion des utilisateurs sous Linux	21
1.4. Éteindre sa machine	22
2. school.picasoft.net	22
Glossaire	23
Références	24

Objectifs



- Savoir survivre devant un terminal Linux
- Savoir utiliser un PC sous Linux (en complément)

Introduction



J'ai été trop souvent confronté à des étudiants qui se trouvaient totalement tétanisés devant les terminaux Linux qu'ils devaient utiliser dans le cadre de mon cours, parfois en situation d'examen sur machine. Ils auraient pu survivre s'ils avaient su ouvrir un terminal, faire un `scp` pour récupérer des fichiers, lancer un éditeur, changer les droits d'un fichier réticent à se publier sur le web...

Ce petit module est destiné à accompagner ses premiers pas sur un système Linux déjà installé sur un poste de travail, sans accès super-utilisateur.

En complément, on trouvera quelques éléments pour ceux qui se trouvent sur leur propre machine avec un accès super-utilisateur ainsi que quelques pistes pour aller plus loin.

Le parcours Démarrer sous Linux permet de prendre plus de temps pour domestiquer une machine Linux : <https://librecours.net/parcours/lin01>



1. Introduction à Linux

1.1. Linux en deux mots

- Linux est un *système d'exploitation** né en 1991.
- C'est le **premier système utilisé pour les serveur informatiques** (devant Windows et d'autres UNIX).
- C'est le troisième système utilisé pour les ordinateurs personnels (derrière Windows et MacOSX)
- C'est un système *libre** et gratuit.
- Il fait partie de la famille des Unix, à l'instar d'Android, MacOSX ou FreeBSD.

Distribution Linux



Définition

On appelle distribution Linux un ensemble de logiciels composé d'un système Linux et de logiciels complémentaires pré-installés et pré-paramétrés, typiquement : une interface graphique de gestion de fichiers, des suite bureautiques, des outils internet, des logiciels multimédia...

Exemple de distributions :

- Debian
- Ubuntu
- Xubuntu
- Fedora
- ...

Linux ou GNU/Linux



Remarque

GNU/Linux est la réunion de deux parties, le projet GNU de Richard Stallman et le projet Linux de Linus Torvalds.

Le nom Linux désigne en général le système d'exploitation dont le nom complet est GNU/Linux. On utilisera l'un pour l'autre dans le cadre de ce cours.

1.2. Utiliser Linux

Pour utiliser Linux, il y a plusieurs possibilités :

- Avoir accès à un PC sur lequel Linux est déjà installé (il suffit de disposer d'un compte utilisateur sur cet ordinateur).
- Installer Linux sur son ordinateur :
 - En téléchargeant en installant une distribution (il faut savoir préalablement graver un DVD ou créer une clé USB *bootable*) ;
 - En commandant un DVD ou une clé USB (coût de quelques euros) ;
 - En participant à une *install party* (organisées par des associations, elles permettent de se faire aider dans le processus d'installation et la prise en main initiale de l'environnement).
- Installer Linux sur son ordinateur en *double-boot* (à côté de son OS initial, on choisit au démarrage quel système on utilise).
- Installer Linux sur son ordinateur dans une machine virtuelle (par exemple : on peut utiliser Linux dans une fenêtre Windows).
- Installer Linux sur une clé en version *live USB persistant*.

Installer Xubuntu sur son PC



<http://xubuntu.fr/>

Installer Ubuntu dans une machine virtuelle sous VirtualBox sous Windows



<https://openclassrooms.com/courses/reprenez-le-contrôle-a-l'aide-de-linux/installez-linux-dans-une-machine-virtuelle>

Installer Debian dans une machine virtuelle sous VirtualBox



<https://school.picasoft.net/modules/deb01>

1.3. Le terminal

Les distributions Linux comportent souvent un mode graphique, pratique pour de nombreuses opérations.

Savoir utiliser un terminal en lignes de commandes n'est donc pas indispensable, mais cela présente des avantages comme :

- connaître des opérations qui seront reproductibles sur toutes les distributions (voire sur d'autres Unix) ;
- savoir utiliser un serveur à distance (via SSH) ;
- savoir échanger ou reproduire des procédures (sans avoir besoin de refaire une succession de manipulations à la souris).



Dès lors qu'on se connecte à un système Linux on peut ouvrir un terminal et faire de nombreuses opérations, telles que :

- lancer des applications,
- manipuler des fichiers,
- se connecter à un autre ordinateur.

1.4. Lancer des applications sous Linux



Le plus générique pour lancer une application sous Linux consiste à :

1. ouvrir un terminal,
2. écrire le nom de l'application, par exemple `firefox`,
3. saisir le caractère `&` et appuyer sur entrée.

```

Terminal - stc@pclf001215: ~
Fichier  Éditer  Affichage  Terminal  Onglets  Aide
stc@pclf001215:~$ firefox&
[1] 13174
stc@pclf001215:~$
  
```

La touche TAB



Il n'est pas en général nécessaire de saisir l'intégralité du nom du programme : on saisit le début, par exemple `fir` et on appuie sur la touche `TAB`.

Dès que le système identifie l'application recherchée, il complète automatiquement avec les caractères manquants.

1.5. Commandes de bases sous Linux : `cd`, `ls`, `mkdir`, `rm`, `find`, `cat`, `nano`...

Le manuel !

La commande `man` permet d'afficher le manuel d'une autre commande.

Par exemple `man ls` permet d'afficher les options de la commande `ls`, et `man man` est l'affichage du manuel du manuel.

Organisation des fichiers

Les fichiers sont organisés sous Linux dans des dossiers (ou répertoires) arborescents (il n'y a pas de notion de disques).

Le premier de ces dossiers, appelé racine, est /.

Des fils courants de racines sont :

- /bin qui contient des programmes du systèmes ;
- /home qui contient les données des utilisateurs ;
- /tmp qui contient des données volatiles accessibles à tous.

Gestion de fichiers

- `pwd` savoir où je me situe dans l'arborescence
- `cd` aller quelque part dans l'arborescence
 - `cd /home` aller dans /home (déplacement absolu)
 - `cd ..` aller dans le répertoire me fils de mon répertoire courant (déplacement relatif)
 - `cd ..` remonter dans l'arborescence (déplacement vers son père)
 - `cd ~` permet de retourner dans son dossier initial (retour au domicile)
- `ls` voir les fichiers et dossiers dans mon dossier actuel (sauf les fichiers cachés commençant par un .)
- `ls -al` voir tous les fichiers et dossiers dans mon dossier actuel avec leurs informations associées (vue détaillée)
- `touch file` créer un fichier file
- `rm file` supprimer le fichier file dans mon dossier actuel
- `mkdir dir` créer un nouveau dossier dir dans mon dossier actuel
- `rm *` supprimer tous les fichiers de mon dossier actuel
- `rm -R dir` supprimer le dossier dir dans mon dossier actuel
- `cat file` afficher le contenu du fichier file
- `more` afficher le contenu du fichier file en mode paginé
- `less` afficher le contenu du fichier file en mode défilement

Édition de fichiers

- `nano file`
 - Éditeur dans le terminal (simple d'utilisation)
 - Les commandes sont indiqués en bas de l'éditeur
- `gedit file &`
 - Éditeur graphique
 - Utiliser une extension de fichier standard ou le menu `Affichage > Mode de coloration` pour obtenir une visualisation adapté au type de fichier édité (par exemple `.sql` pour un fichier SQL)

Rechercher un fichier

- `find / -name '*test*'` permet de rechercher un fichier contenant la chaîne test sur tout le disque
- `find ~ -name '*test*'` permet de rechercher un fichier contenant la chaîne test dans son espace personnel



Agir en tant que root (su / sudo) (cf. p.20)

Installer des applications sous Linux (cf. p.21)

2. Utiliser une machine Linux

Cette section a pour objectif de présenter une partie des commandes utiles pour utiliser un ordinateur sous Linux. Elle suppose l'usage du terminal et des commandes de base (`cd`, `ls`, `cat`, `nano`...) acquise.

On s'appuiera sur une distribution Ubuntu ou Debian.

<https://www.debian.org/doc/manuals/debian-reference/ch01.fr.html>

2.1. Connexion à un serveur Linux avec SSH



Le terminal (cf. p.6)

L'usage de Linux en ligne de commande permet de travailler sur une machine à distance.

On utilise pour cela le protocole SSH. Il permet à un client SSH de se connecter à un serveur SSH.

ssh : ce connecter à un ordinateur



`ssh user@server` permet de se connecter depuis n'importe quel terminal à une machine distante qui autorise les connexions SSH et sur laquelle on dispose d'un compte *user*.



`ssh nf17@apollon.utc.fr` permet de se connecter à la machine *apollon.utc.fr* en tant qu'utilisateur *nf17*.

Syntaxe complète



`ssh user@server -p num_port` si la machine n'utilise pas le port standard (port 22).

hostname



Si l'on veut savoir sur quelle machine est actuellement connecté notre terminal, on utilise la commande `hostname`.

Putty sous Windows



*Putty** est un client SSH sous Windows, qui permet donc de se connecter à distance à une machine Linux

scp : copie distante de fichier



```
scp /local/path/file user@server:/dist/path/file
scp -r /local/path/dir user@server:/dist/path/dir
scp user@server:/dist/path/file /local/path/file
scp -r user@server:/dist/path/dir /local/path/dir
```



`scp monfichier.txt nf17@apollon.utc.fr:~/mondir` permet de copier le fichier *monfichier.txt* vers le répertoire *mondir* situé dans le dossier personnel de *nf17* sur la machine *apollon.utc.fr*.

sftp : transfert de fichier



SFTP est un protocole de transfert de fichiers alternatif à *scp* qui peut être utilisé avec un logiciel graphique comme *Filezilla*.

2.2. Affichage des droits sous Linux (ls -al)



Commandes de bases sous Linux : *cd, ls, mkdir, rm, find, cat, nano...* (cf. p.7)

Sous Linux, chaque dossier et chaque fichier :

- appartient à un utilisateur **et** à un groupe ;
- possède des droits.

Ces droits se divisent en trois :

1. Les droits pour l'utilisateur propriétaire
2. Les droits pour les utilisateurs membres du groupe propriétaire
3. Les droits pour tous les autres utilisateurs

ls -al



La commande `ls -al` affiche tous les fichiers et dossiers d'un répertoire, avec leurs propriétaires et leurs droits.

```
1 drwxr-x--- 3 stc friends 4096 mars 31 19:11 .dir
```

- Le premier caractère indique que l'élément est un dossier (*d*) ou un fichier (*-*).
- Les trois caractères suivants indiquent les droits de l'utilisateur propriétaire (ici *stc* a les droits *rwx* : lire, écrire, parcourir)
- Les trois caractères suivants indiquent les droits des utilisateur membres du groupe propriétaire (ici les membres de *friends* ont les droits *r-x* : lire et parcourir, mais pas écrire)

- Les trois caractères suivants indiquent les droits des autres utilisateurs (ici --- : aucun droit)
- Le nombre suivant indique le nombre de sous-dossiers dans le dossier listé (ici `.dir` a 3 sous-dossiers, dont `.` et `..` donc un seul vrai sous-dossier)
- La chaîne suivante indique l'utilisateur propriétaire (ici l'utilisateur `stc`)
- La chaîne suivante indique le groupe propriétaire (ici le groupe `friends`)
- Le nombre suivant indique la taille de l'élément en octets (ici 4096 octets)
- La chaîne suivante indique la date et l'heure de dernière modification du fichier (ici le 31 mars à 19h11)
- La dernière chaîne est le nom du fichier (ici `.dir`)

? Exemple

```
1 -rw----- 1 root  root   2048 mars  31 19:11 .config
2 drwx----- 2 stc   stc   4096 avril  3 20:25 firefox
3 -rw-rw-r-- 1 stc   www   31691 avril  2 10:11 test.jpg
```

- Le fichier `.config` appartient à l'utilisateur `root` et au groupe `root` (il y a souvent un utilisateur et un groupe de même nom) ; l'utilisateur `root` peut le lire et le modifier, personne d'autre ne peut y accéder.
- Le dossier `firefox` appartient à l'utilisateur `stc` et au groupe `stc` ; l'utilisateur `stc` peut y accéder, le lire et le modifier, personne d'autre ne peut y accéder.
- Le fichier `test.jpg` appartient à l'utilisateur `stc` et au groupe `www` ; l'utilisateur `stc` ainsi que les membres du groupe `www` peuvent le lire et le modifier, les autres utilisateurs peuvent seulement le lire.

Les droits rwx

💡 Fondamental

- **r** (*read*) : lire le contenu d'un fichier (exécuter `more` par exemple) ou d'un dossier (`ls`)
- **w** (*write*) : écrire un fichier (`nano`), écrire dans un dossier (`touch`)
- **x** (*execute*) : exécuter un fichier, entrer dans un dossier (`cd`)

📄 Complément

<http://www.linuxcertif.com/doc/keyword/ls/>

<http://www.linux-france.org/article/sys/fichiers/>

2.3. Gestion des droits sous Linux

📖 Rappel

Affichage des droits sous Linux (cf. p.10)

chmod

La commande `chmod` permet de modifier l'attribution des droits d'un fichier ou d'un dossier.

Pour cela on attribue 3 nombres de 0 à 7 correspondant aux 3 droits : utilisateur propriétaire (u), utilisateurs membres du groupe propriétaire (g), autres utilisateurs (o), sachant que :

1. **r** vaut 4, **w** vaut 2, **x** vaut 1 ;
2. on somme les droits que l'on veut cumuler.

? Exemple

- `chmod 640 file` donne les droits **rw** (4+2) à *user*, **r** (4) à *group* et aucun droit (0) à *others*.
- `chmod 775 file` donne les droits **rw**x (4+2+1) à *user* et *group* et **rx** (4+1) à *others*.

Droits courant pour un fichier (non exécutable)

✂ Méthode

- `chmod 666 file` (fichier non exécutable public)
- `chmod 644 file` (fichier non exécutable public, modifiable par soi uniquement)
- `chmod 600 file` (fichier privé)
- `chmod 400 file` (fichier privé en lecture seule)
- `chmod 444 file` (fichier public en lecture seule)

Droits courant pour un dossier (et de son contenu)

`chmod 777 dir` (dossier public)

`chmod 755 dir` (dossier public, modifiable par soi uniquement)

`chmod 700 dir` (dossier privé)

`chmod 500 dir` (dossier privé en lecture seule)

`chmod 555 dir` (dossier public en lecture seule)

chmod -R Changer les droits de tout un répertoire

📁+ Complément

`chmod -R XXX dir` affecte les droit XXX à *dir* et à tous les fichiers et sous-dossiers de *dir* (récursivement).

chown : Changer les propriétaires d'un fichier

📁+ Complément

`chown user:group target` permet de donner la propriété de *target* à l'utilisateur *user* et au groupe *group*.

umask : Changer les droits par défaut

📁+ Complément

- `umask -S` permet de voir les droits par défaut.
- `umask XYZ` enlève X, Y et Z à 666 pour les fichiers et 777 pour les dossiers lors de la création
- `umask 022` enlève les droits par défaut en écriture pour *group* et *others*;
- `umask 077` enlève tous les droits par défaut pour *group* et *others*.

2.4. Gestion des processus sous Linux (ps et kill)

Processus



Définition

Un processus est un programme en cours d'exécution.

ps : afficher les processus



Syntaxe

La commande `ps` permet d'afficher la liste des processus associés au terminal courant.

La commande `ps -aux` permet d'afficher la liste de tous les processus.

La commande `ps tree -p` permet d'afficher l'arbre de tous les processus.

ps aux | grep : chercher un processus



Fondamental

La commande `ps aux | grep xxx` permet d'afficher tous les processus qui ont été lancé par `xxx` ou qui contiennent `xxx` dans leur nom.

kill : arrêter les processus



Fondamental

Pour arrêter un processus, trouver son numéro (*PID*) puis exécuter : `kill PID`.

Si le processus est bloqué et refuse de s'arrêter, exécuter `kill -9 PID` (cet arrêt forcé peut entraîner des perte de données).



Complément

`pkill xxx` (`pkill -9 xxx`) permet d'arrêter le processus de nom `xxx`.

`killall xxx` (`killall -9 xxx`) permet d'arrêter tous les processus de nom `xxx`.

top et htop



Complément

Les commandes `top` et `htop` permettent d'afficher les processus qui consomment le plus de ressources sur la machine.

2.5. Gestion de mémoires amovibles sous Linux (df -h et umount)

Montage et point de montage



Définition

Le **montage** d'une mémoire est le fait d'associer à cette mémoire un emplacement dans le système de fichier, appelé **point de montage**, pour pouvoir y accéder.

Montage automatique (/media)

Sous les Linux récents les mémoires amovibles (clés ou disques USB, CD-ROM...) se **montent** automatiquement. Les points de montage dépendent des distributions et des configurations, sous Ubuntu et Debian, il s'agit du dossier `/media` (on trouve aussi souvent le dossier `/mnt`).

Liste des disques et mémoires montées



Fondamental

Pour avoir des informations sur la liste des mémoires montées (et sur l'occupation de l'espace d'un disque ou d'une mémoire amovible) : `df -h`

Démontage

**Attention**

Une mémoire doit être démontée afin d'être débranchée, cela permet d'être certain que le système n'est plus en train d'agir sur la mémoire. Une déconnexion sans démontage peut entraîner la corruption des données stockées sur la mémoire : `umount point-de-montage`.

Par exemple : `umount /media/stc/MYUSB01`.

Montage manuel (et remontage)

**Complément**

Pour monter manuellement une mémoire qui ne l'aurait pas été automatiquement ou remonter une mémoire démontée (il faut être *root*) :

```
mount memoire point-de-montage
```

Par exemple : `sudo mount /dev/sdd1 /media/stc/MYUSB01`

Analyse d'occupation d'espace disque

**Complément**

- `du -h -d0 dir` espace occupé par *dir*.
- `du -h -d1 dir` espace occupé par chaque dossier de *dir*.
- `du -ha -d1 dir` espace occupé par chaque dossier et fichier de *dir*.
- `du -ha -d1 dir | sort -h` espace occupé par chaque dossier et fichier de *dir*, avec un tri par espace.

gnome-disks, baobab

**Complément**

gnome-disks est un utilitaire graphique qui permet de monter, démonter et formater des mémoires amovibles (ainsi que les disques durs du système).

baobab est un utilitaire graphique permettant de voir l'occupation des espaces sur les disques.

2.6. Quelques utilitaires sous Linux

tar et zip

tar est un format d'archivage qui permet de représenter une arborescence complète sous la forme d'un seul fichier.

gzip est un utilitaire de compression qui peut être utilisé pour compresser une archive Tar.

zip est un utilitaire d'archivage et de compression, équivalent fonctionnellement à Tar+Gzip ; il est moins utilisé sous Linux, mais plus sur Windows.

- `tar -cvf file.tar dir` crée un fichier *file.tar* avec le dossier *dir*.
- `tar -czvf file.tgz dir` crée un fichier *file.tgz* compressé avec *gzip*.
- `tar -tf file.tar` permet de voir le contenu du fichier archive *file.tar*.
- `tar -xvf file.tar` permet de décompresser le contenu du fichier *file.tar* dans l'emplacement courant.
- `zip file.zip dir` permet de zipper le dossier *dir* dans le fichier *file.zip*.
- `unzip file.zip -d destdir` permet de dézipper le fichier *file.zip* dans le dossier *destdir*.

cups-pdf et pdftk

pdftk permet de manipuler des fichiers PDF. Par exemple :

- `pdftk input.pdf cat 1-3 5 output output.pdf` permet de créer un fichier *output.pdf* avec les pages 1,2,3 et 5 du fichier *input.pdf*
- `pdftk t1.pdf t2.pdf output output.pdf` permet de créer un fichier *output.pdf* par concaténation des fichiers *t1.pdf* et *t2.pdf*.

cups-pdf permet d'imprimer dans un fichier PDF.

Exercices



1. Découverte de la ligne de commande sous Linux

Cet exercice consiste à expérimenter quelques commandes de base sous Linux.

1. Connectez vous à une machine Linux ;
2. Ouvrez un terminal.

1.1. Exercice : man man

Exécutez la commande `man man`.

Exercice

Comment s'appelle la section qui résume la syntaxe d'une commande dans le manuel.

Exercice

Quelle touche doit-on utiliser pour convoquer l'aide de *man* ?

Exercice

Quelle touche doit-on utiliser pour sortir de *man* ?

1.2. Exercice

Trouvez ce que signifient dans le manuel :

- les termes en gras
- les termes soulignés ou italiques
- les termes entre crochets [...]
- la barre |
- les trois petits points ...

[]

|

italique ou souligné

gras

...

à taper exactement comme indiqué	à remplacer par l'argument approprié	arguments facultatifs	options qui ne peuvent pas être utilisées simultanément	argument ou expression qui peut être répété
----------------------------------	--------------------------------------	-----------------------	---------------------------------------------------------	---------------------------------------------

1.3. Exercice : ls

Exercice

Quelle commande devez-vous exécuter pour visualiser le manuel de la commande `ls` ?

```
_____
```

Exercice

En lisant le manuel, trouvez comment afficher le contenu du répertoire courant en mode *listing*.

```
_____
```

1.4. Exercice : cd pwd mkdir touch ...

L'exercice suivant consiste à manipuler le système de fichier. On vous demande de saisir la commande nécessaire pour répondre à chaque question posée.

Exercice

Déplacez-vous dans le répertoire `tmp` situé à la racine du système de fichier.

```
_____
```

Exercice

Vérifiez que vous êtes bien au bon endroit.

```
_____
```

Exercice

Créez un répertoire *hello*.

```
_____
```

Exercice

Déplacez-vous dans ce répertoire.

```
_____
```

Exercice

Créez un fichier *world* dans ce répertoire.

```
_____
```

Exercice

Afficher la liste des fichiers de ce répertoire en mode *listing*.

```
_____
```

Exercice

Éditer le fichier *world* avec l'éditeur *nano*, et saisissez le texte de votre choix.

████████████████████

Exercice

Après être sorti de l'éditeur, affichez le contenu de votre fichier (avec *cat*, *more* ou *less*)

████████████████████

Exercice

Rechercher dans votre répertoire courant tous les fichiers dont le nom contient *orl*.

████████████████████

Exercice

Supprimer le fichier *world*.

████████████████████

Exercice

Retournez dans votre dossier personnel initial.

████████████████████

Exercice

Supprimer le dossier *hello* que vous avez créé dans */tmp*.

████████████████████

Défi



1. Exercice : Fichier caché

Question 1

Dans un répertoire de votre choix :

1. créez le fichier *sherlock.sh* contenant la ligne suivante. Il s'agit de code *sh* qui a été volontairement (légèrement) offusqué, ne cherchez pas à l'interpréter.
2. exécutez ce fichier avec la commande *sh sherlock.sh*

Qu'affiche ce programme ?

```
1 F="Hidden";mkdir -p /tmp/b607c198/1/2/3;mkdir -p /tmp/b607c198/2/2/3;mkdir -p
/tmp/b607c198/3/2/3;mkdir -p /tmp/5f260f39da7b/a/b/c;mkdir -p
/tmp/bed0a/11e8b/59c7;mkdir -p /tmp/5f260f39da7b/a/b/c;mkdir -p
/tmp/5f260f39da7b/a/b/c;for i in `ls /tmp/b607c198`;do echo $i >
/tmp/bed0a/11e8b/59c7/$F;done;j=$((3+4));chmod $j$j$j /tmp/bed0a/11e8b/59c7/$F;chmod
500 /tmp/bed0a/11e8b/59c7;echo `date`;M="Hello $USER, I have finished, answer the
questions now.";
```

Indice :

On pourra utiliser l'éditeur *nano* et copier le code en utilisant le bouton central ou droit de la souris.

Question 2

Le script *sherlock.sh* a créé un fichier *Hidden* quelque part dans le dossier */tmp*, trouver ce fichier en utilisant la commande *find*.

Question 3

Que contient le fichier *Hidden* ?

Indice :

Utilisez la commande *cat*.

Question 4

Le fichier *Hidden* est-il accessible en écriture par n'importe quel utilisateur du système ?

Question 5

Supprimer le fichier *Hidden*.

Question 6

Supprimer le dossier */tmp/bed0a/11e8b/59c7*.

Complément



1. Administrer une machine Linux (introduction)

Cette section a pour objectif de présenter une partie des commandes utiles pour administrer un ordinateur sous Linux. Elle suppose l'accès en mode *root* à la machine.

1.1. Agir en tant que root (su / sudo)

Un système Linux est généralement utilisé par plusieurs utilisateurs (ou *user*) identifiés par un compte.

Tous les systèmes ont en commun le premier de ces utilisateurs : *root*, l'administrateur du système qui possède tous les droits.

su

§ Syntaxe

Pour se connecter en tant qu'utilisateur *root* (et pouvoir faire les opérations qui lui sont réservées) : `su root`.

- Le mot de passe de l'utilisateur *root* est demandé par le système.

sudo

§ Syntaxe

Pour exécuter une commande en tant que *root* sans changer d'*user* : `sudo commande`.

- Le mot de passe de l'utilisateur lançant `sudo` est demandé par le système.
- Seuls certains utilisateurs appelés *sudoers* ont le droit d'exécuter la commande `sudo`.
- Pour ajouter un utilisateur dans la liste des *sudoers*, exécuté en tant que *root* (ou via un autre utilisateur *sudoers*) :
 - `adduser user sudo` (en tant que *root*)
 - `sudo adduser user sudo` (avec un utilisateur déjà membre des *sudoers*)

Debian (sudo)

! Attention

Sous Debian la commande `sudo` n'est pas installée par défaut.

Ubuntu (su)

! Attention

Sous Ubuntu l'utilisateur *root* n'est pas actif et il n'est donc pas possible d'exécuter la commande `su root`. On peut en revanche utiliser `sudo`, l'utilisateur créé à l'installation est membre des *sudoers*.

Pour activer le compte *root*, il faut lui attribuer un mot de passe avec la commande : `sudo passwd root`.

Agir en tant qu'un autre utilisateur

Complément

Pour agir en tant qu'utilisateur *user*, exécuter :

`su user`, ou `su - user` pour adopter l'environnement de *user*.

1.2. Installation d'applications sous Debian et Ubuntu (apt-get)

Pour installer une application exécuter la commande suivante en tant que *root* ou via *sudo* :

```
sudo apt-get install applications
```

Installer virtualbox, lftp, cups-pdf, pdftk, dia, gimp et git

Exemple

```
1 sudo apt-get install virtualbox lftp cups-pdf pdftk dia gimp git
```

Maintenir le système à jour

Complément

Pour mettre le système à jour, exécuter les deux commandes suivantes en tant que *root* ou via *sudo* :

1. `sudo apt-get update`
2. `sudo apt-get upgrade`

1.3. Gestion des utilisateurs sous Linux

Remarque

Seul l'utilisateur *root* (ou les *sudoers*) peuvent exécuter ces commandes.

Utilisateurs

Syntaxe

`adduser user` ajouter un utilisateur *user*.

`deluser user` supprimer un utilisateur *user*.

`passwd user` changer le mot de passe de *user*.

`cat /etc/passwd` lister les utilisateurs.

Groupes

Syntaxe

`addgroup group` ajouter un groupe *group*.

`usermod -aG group user` ajouter *user* à *group*.

`deluser user group` enlever *user* de *group*

`cat /etc/group` lister les utilisateurs avec leurs groupes.

`groups user` lister les groupes de *user* (cette commande est accessible aux utilisateurs non *root*)

1.4. Éteindre sa machine



La commande `shutdown` permet d'éteindre sa machine en ligne de commande :

- `shutdown -h now` (éteindre tout de suite)
- `shutdown -h 23:00` (éteindre à 23h00)
- `shutdown -h +60` (éteindre dans 60 minutes)

2. school.picasoft.net

Install party



Installer une distribution Linux comme Ubuntu sur un ordinateur est aujourd'hui une opération simple et rapide, qui ne nécessite aucune compétence spécifique.

Je vous conseille néanmoins de participer à une *install party*, cela vous permettra de comprendre plus rapidement les notions de base dont vous aurez besoin pour bien utiliser votre nouvel OS.



Sur le site librecours.net propose une initiation à Linux plus complète: <https://librecours.net/parcours/in01>

Glossaire



Logiciel libre

Aujourd'hui, un logiciel est considéré comme libre, au sens de la *Free Software Foundation*, s'il confère à son utilisateur quatre libertés (numérotées de 0 à 3)7 :

- 0. la liberté d'exécuter le programme, pour tous les usages ;
- 1. la liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins ;
- 2. la liberté de redistribuer des copies du programme (ce qui implique la possibilité aussi bien de donner que de vendre des copies) ;
- 3. la liberté d'améliorer le programme et de distribuer ces améliorations au public, pour en faire profiter toute la communauté.

L'accès au code source est une condition d'exercice des libertés 1 et 3.

(Wikipédia ; fsf.org)

OS (Operating System, Système d'Exploitation)

Un ordinateur est une machine qui a besoin de programmes pour faire quelque chose. Lorsqu'un ordinateur démarre il exécute en général un programme interne, dit programme d'amorçage, puis le premier programme externe qu'il exécute est le **système d'exploitation**. C'est le système d'exploitation qui permet ensuite d'exécuter les applications (traitement de texte, navigateur web, client mail...).

Exemples de système d'exploitation :

- Windows
- MacOSX
- GNU/Linux
- Android
- iOS
- FreeBSD
- ...

Références



Putty

PuTTY est utile si vous êtes sur un système Windows, pour obtenir une connexion SSH et une console sur un serveur Linux.

C'est un exécutable qui ne nécessite pas d'installation, il peut donc être téléchargé sur n'importe quel disque et lancé directement.

<http://www.putty.org/>

<http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>