

Travaux Pratiques 0 Système d'Exploitation - Linux

A la découverte de Linux

Contents

1 Premiers pas	2
1.1 Démarrage sur l'interface graphique	2
1.2 Lancer et arrêter un terminal	2
1.3 A la découverte du shell	2
2 Quelques commandes	3
2.1 Quelques commandes utiles	3
2.2 les pages de manuel	4
3 Quelques aides pour taper les commandes	4
3.1 Quelques exemples et applications	4

Pour passer des commandes en mode texte au shell, il faut lancer un terminal ou une console virtuelle. Sur les systèmes Unix/Linux, le terminal et la console virtuelle désignent plus couramment des fenêtres d'invite de commande contenant un shell Unix. Pour cette activité, nous utiliserons le terminal.

1.2 Lancer et arrêter un terminal

L'accès depuis l'interface graphique à un terminal peut se faire

- avec les combinaisons de touches suivantes CTRL + ALT + T.
- en cliquant sur l'entrée "Terminal" du menu ou le raccourci offert par l'interface graphique.

Vous avez la possibilité de lancer plusieurs terminaux. Pour quitter un terminal :

- faire la combinaison de touche CTRL+D.
- cliquer sur le bouton de fermeture.
- ou taper la commande "exit" ou "logout" puis valider.

1.3 A la découverte du shell

Lorsque vous lancez un terminal, vous êtes accueillis par une invite de commande(prompt) du genre:

- utilisateur@nommachine:/home/utilisateur\$

Utilisateur	Applications		
	Gestionnaire de fenêtres	Gestionnaire de bureau	Commandes Shell
Server X			
Noyau			
Matiel			

L'invite de commande diffère en fonction du système d'exploitation et peut être modifiée grâce à la variable d'environnement PS1.

Pour savoir le shell que vous utilisez, vous pouvez exécuter la commande :

\$ echo \$SHELL

Résultat : **/bin/bash**.
Si vous avez pour résultat /bin/bash, vous utilisez bash . Bash, acronyme de "Bourne-again shell," est le shell par défaut sur la plupart des systèmes Linux. Le rôle du Shell est d'exécuter vos commandes, il vous permet d'interagir avec le système.

2 Quelques commandes

2.1 Quelques commandes utilitaires

Exécuter sur un terminal les commandes suivantes :

\$ whoami → Affiche mon nom ; tapé qui
\$ echo bonjour → Affiche bonjour
\$ date → La date du jour
\$ echo Au revoir → Au revoir
\$ clear → Supprime tous les commandes précédentes affichées sur l'écran du terminal
\$ cal → Affiche le calendrier du mois.
\$ exit → quitte le terminal.

- Que font ces commandes?

Lancer un terminal et exécuter les commandes suivantes :
\$ who → Qui est connecté au terminal
\$ cal 2005 → Calendrier 2005 au format
\$ echo bonjour → Bonjour bonjour
\$! → !
\$ cal -m 7 2011 → Mois du calendrier précédent exécuté du terminal
\$ history → Les commandes précedentes.
\$ xeyes → Contrôle la souris.

Si une commande ne rend pas la main, on peut arrêter le programme correspondant en tapant CTRL+C.

- Que font ces commandes?
- Quel est votre jour de naissance?

2.2 les pages de manuel

Exécuter les commandes

\$ echo bonjour
\$!!
\$ clear
\$ man clear

- Que fait cette dernière commande?
- Pour fermer la page de manuel, appuyer sur Q.
- afficher la page de manuel de la commande echo, puis celle de cal.

3 Quelques aides pour taper les commandes

- Edition de ligne: Si on se trompe en tapant une commande, et que l'on s'en aperçoive avant d'appuyer sur "entrée", on peut utiliser les touches flèches gauche et droite pour déplacer le curseur à l'emplacement où se trouve l'erreur.
- Historique: Si on ne s'aperçoit de l'erreur qu'après avoir lancé la commande, on veut souvent lancer une autre commande corrigée. Au lieu de tout recréer, on peut utiliser la flèche vers le haut, qui rappelle la commande précédente (puis la commande d'avant, etc., si on appuie plusieurs fois).
- Complétion: Lorsque l'on veut taper le nom d'un fichier existant, ou d'une commande, on peut taper le début du nom du fichier puis appuyer sur la touche tabulation (marquée Tab ou par deux flèches en sens opposés). Le Shell insère alors la fin du nom (s'il y a plusieurs possibilités, le Shell complète seulement le plus long préfixe commun). La complétion a deux avantages : elle permet de moins taper, et elle assure que le nom complété existe.
- Sélection-copier-coller: Lorsqu'on sélectionne du texte avec la souris, celui-ci est mis dans une mémoire tampon. Pour coller ce texte ailleurs, il suffit de s'y positionner avec le curseur et d'appuyer sur (cliquer avec) la touche du milieu (scroll). Le copier-coller classique (sélection, bouton droit, copier - bouton droit, coller) reste bien sûr valable.
- !!: Rappelle la dernière commande.

• !num: Rappelle la commande numéro num (la commande history affiche la liste des commandes conservées en mémoire, chacune précédée d'un numéro).

- !chaine: Relance la dernière commande saisie commençant par chaine

3.1 Quelques exemples et applications

saisir au clavier les débuts de commande suivant et appuyer sur tabulation (=) avant de valider.

\$ his puis appuyer sur tabulation (=)
\$ ec puis appuyer sur tabulation (=)
\$ cd /u puis appuyer sur tabulation (=)
\$ cd /usr/s puis appuyer sur tabulation (=)

Exercice : Helloworld

1. Écrire un script shell helloworld qui affiche le message « Helloworld »
2. écrire un script shell hello qui demande à l'utilisateur son nom « nom » et affiche «Hello Nom »
3. modifier hello en hello2 pour qu'il prenne son paramètre en ligne de commande

Exercice : somme, produit

1. Écrire un script operation qui affiche la somme et le produit de deux nombres lus au clavier
2. Modifier opération pour qu'il prenne ses deux paramètres en ligne de commande
3. Modifier le script pour que les deux opérations soient des fonctions.

Exercice : Maximum

- 1.écrire un script max_ab qui affiche le maximum de deux nombres lus au clavier
- 2.écrire un script shell max_ab_par qui affiche le max de deux nombres passés en paramètres
- 3.modifier le script max_ab_par pour qu'il affiche un texte qui décrit ce qu'il fait et sa syntaxe lorsqu'on lui passe le paramètre --help.

Exercice : Nombre paramètres

4. écrire un script nb_param qui renvoie le nombre de ses paramètres
5. modifier nb_param pour qu'il affiche le nom du script, le nombre de paramètres et les paramètres (avec leur position)

Exercice : équation degré 1

- 4.Écrire un script shell equation1 qui permet de résoudre une équation du premier degré
- 5.Modifier le script equation1 pour qu'il résolve l'équation avec les paramètres passés en ligne de commande. Si le nombre de paramètre passé en ligne de commande est incorrect, il demande à l'utilisateur de les saisir au clavier.
- 6.Ajouter une documentation à ce script qui s'affichera avec l'option -help,
- 7.Ajouter un numéro de version qui s'affichera avec l'option --version

Exercice : équation degré 2

8. Écrire un script shell **equation2** qui permet de résoudre une équation du premier degré
9. Modifier le script **equation2** en **equation2_par** pour qu'il prenne ses paramètres en ligne de commande

Exercice : Table de multiplication

1. Créer un script nommé "**table**" permettant d'afficher des tables de multiplication.
"**table 5**" aura pour résultat l'affichage:

```
1 x 5 = 5  
2 x 5 = 10  
3 x 5 = 15  
4 x 5 = 20  
5 x 5 = 25  
6 x 5 = 30  
7 x 5 = 35  
8 x 5 = 40  
9 x 5 = 45  
10 x 5 = 50
```

2. Améliorer le script **table** pour qu'il puisse prendre en paramètre plusieurs nombres et afficher leurs tables de multiplication les uns à la suite des autres.

Exercice : calculatrice

Nous voulons mettre en place un script **calcul** qui prend en paramètre une chaîne de caractère la décompose en opération et opérande et affiche le résultat.

Exemple :

```
$ calcul 5+8
```

```
$ 13
```

```
$ calcul 10*3
```

```
$ 30
```

Exercice : Conversion

1. Écrire un programme **conversion** qui convertit un nombre n saisi au clavier en binaire et en hexadécimal (vous ferez l'algorithme sans utiliser les commandes natives du système).
2. Améliorer le programme **conversion** pour qu'il affiche les codes binaires et hexadécimaux des nombres de 1 en n, n saisi au clavier.

Exercice : chiffrement par Nombres Premiers

Écrire un script **chiffrement** qui renvoie le chiffrement en nombre premier d'une chaîne de caractère saisi au clavier. Le chiffrement en nombre premier consiste à établir une correspondance entre caractères et nombre premier. Par défaut vous remplacerez les 26 lettres de l'alphabet par les 26 premiers nombres premiers dans l'ordre (A=2, B=3, C=5, D=7, ..., Z=101) et l'espace par le nombre premier suivant.

Linux - Travaux Dirigés 2 - Système de Fichier

Exercice 1

1. Donnez deux exemples de formats de fichier propriétaires. Donnez deux exemples de formats de fichiers Libre.
2. Supposons que deux utilisateurs U1 et U2 partagent le même fichier F à travers un lien dur. L'utilisateur U1 n'ayant plus besoin de F le supprime de son répertoire (en supposant qu'il a les droits nécessaires pour le faire). L'utilisateur U2 a-t-il encore la possibilité d'utiliser F? Pourquoi?
 - o Même question, mais cette fois-ci U1 accède à F par un lien dur, tandis que U2 y accède par un lien symbolique.
 - o Même question mais cette fois U1 accède à F par un lien symbolique, tandis que U2 y accède par un lien dur.
3. Rappeler la différence entre un lien symbolique et un lien dur (physique). Quelle commande shell permet de reconnaître des liens durs et symboliques dans une arborescence ?
4. Quelle différence faites vous entre le formatage physique et le formatage logique d'un disque.
5. Vous connectez un périphérique de stockage externe (clé USB, disque externe) à votre ordinateur, en quoi consiste le montage du système de fichier pour ce périphérique?
6. Dans la norme Filesystem Hierarchy Standard (FHS), quel est le nom du répertoire racine du système de fichier? quel est le contenu des répertoires : /etc, /bin, /sbin, /dev et /home?

Exercice 2

Supposons que dans votre répertoire personnel, vous avez un répertoire TP avec l'arborescence donnée en annexe. Donnez les commandes qui permettent de faire les tâches suivantes :

1. Créer le répertoire TP11 dans le répertoire TP1
2. Créer le fichier fic21 dans le répertoire rapport
3. créer un lien dur lien11 dans TP1 qui pointe sur le fichier fic1
4. créer un lien symbolique lien12 dans rep11 qui pointe sur le fichier fic5
5. supprimer le fichier fic4
6. supprimer le répertoire rep11
7. supprimer le répertoire scripts
8. renommer le fichier fic3 en fichier3
9. déplacer le fichier fic1 vers le répertoire rep11.

Linux – TD/TP 2 - Complément find et mount (==>manpages pour plus d'info)

Exercice 3 : montage/démontage

Nous allons voir comment monter et démonter un système de fichier.

1. Vérification des périphériques usb.

Avant d'insérer une clé usb dans votre machine, exécuter les commandes suivantes :

1. lsusb, elle vous permet de lister les périphériques usb du système.
2. dmesg, elle vous permet d'afficher les dernier messages du noyau.
3. df -H, elle vous permet de voir l'espace libre dans chaque SF mais aussi les points de montage

2. Automontage.

Insérer une clé usb dans votre ordinateur. Avec le système d'automontage, la clé est automatiquement montée sur un répertoire. Exécuter les commandes lsusb, dmesg et df -H.

1. Constatez vous un changement dans les résultats de ces commandes ?
2. Quel fichier de périphérique représente votre clé usb ?
3. Dans quel répertoire votre clé usb est elle montée ?

4. Déplacer vous avec cd dans le répertoire qui représente votre clé usb et lister son contenu.

3. Démontage.

1. Pour démonter la clé usb, exécuter la commande umount suivie du nom du fichier qui représente votre clé usb. Si la clé est en usage, le démontage ne marche pas.
2. Essayer de vous déplacer à nouveau dans le répertoire qui représente votre clé usb.
3. Que constatez vous.

4. Montage.

Nous allons monter la clé usb dans un répertoire se trouvant dans votre home

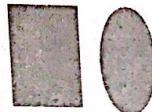
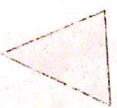
1. créer un répertoire monrep dans votre home et lister son contenu.
1. Executer la commande (remplacer /dev/sdb1 par le fichier qui représente votre clé)
\$ mount -t vfat /dev/sdb1 monrep

2. lister le contenu de monrep.
3. Que constatez vous ?

Si vous souhaitez monter automatiquement un périphérique au démarrage du système, il faut ajouter une ligne le représentant dans le fichier /etc/fstab. Visualisez le fichier /etc/fstab.

Exercice 4 : recherche de fichiers

1. Localisez la commande man ? Localisez la commande adduser ?
2. Localisez la commande et les manpages des commandes cal, man et mount.
3. Cherchez dans toute l'arborescence les fichiers dont le nom se termine par .c, redirigez les erreurs vers le fichier poubelle /dev/null
4. Trouvez tous les fichiers core dans l'arborescence et supprimez les après confirmation.

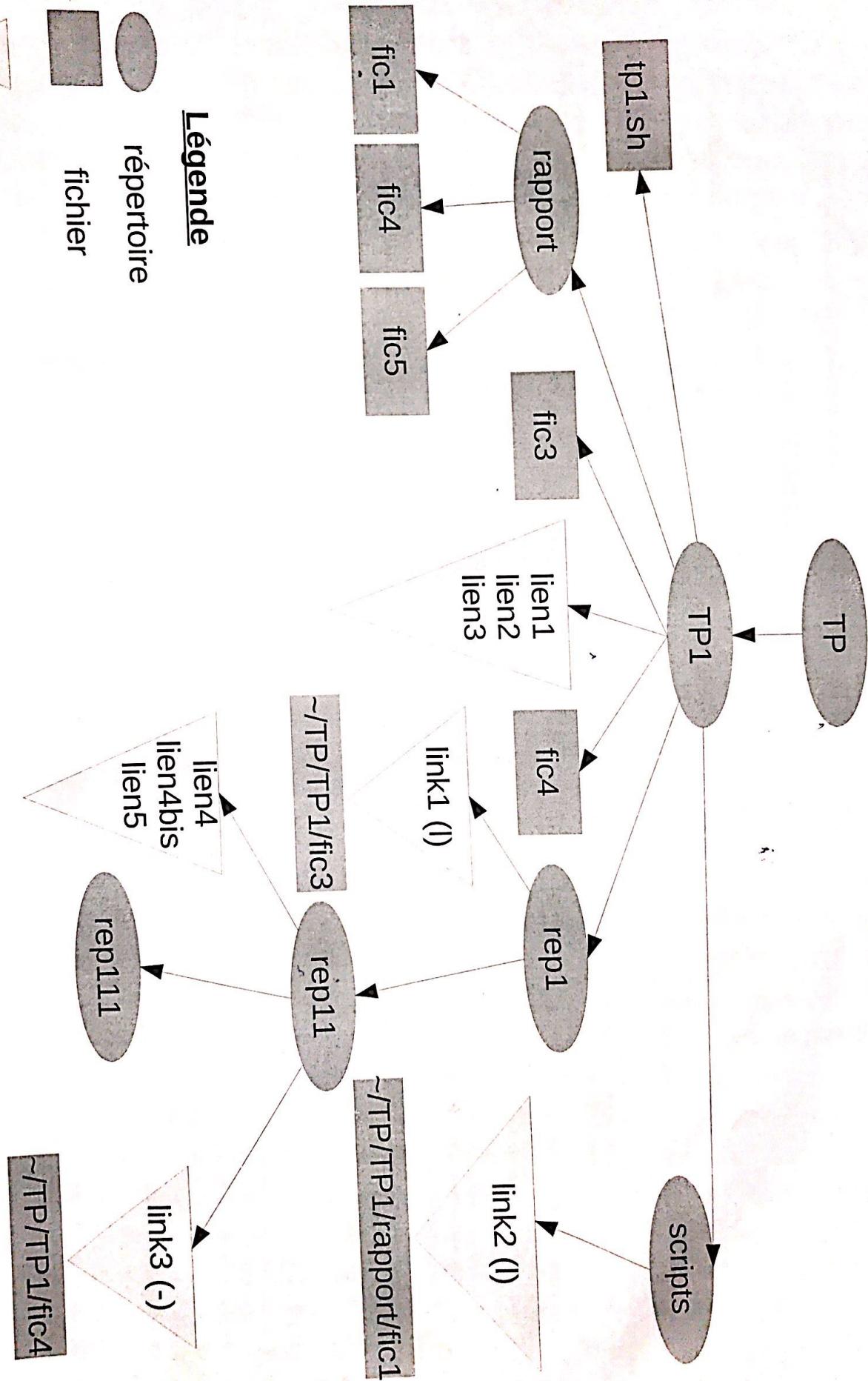


Légende

répertoire

fichier

lien



Université Assane Seck de Ziguinchor

UFR Sciences et Techniques

Département d'informatique

Licence 3 Informatique

Linux - Travaux Dirigés

Fiche 3 – Sécurité, mécanismes de droit

Exercice 1

1. Donner la commande qui permet de créer un nouvel utilisateur **adama** sur votre système avec pour home **/home/l3info/adama**, pour groupe principal **l3info** et qui expire le **31/12/2021**
2. Quel seront les droits d'un fichier créé dans un système où le umask est égal à 056 ? et 033 ?
3. Considérons la situation ci-dessous et répondez aux questions:
[mirian@home2 Test]\$ ls -l
drwxrwxr-x 2 mirian mirian 4096 Sep 22 10:58 Temp
-rw-rw-r- 1 mirian mirian 14 Sep 20 21:15 titi
-rw-rw-r- 1 mirian mirian 69 Sep 20 21:18 toto
 1. En sachant qu'aucun changement de droit particulier a été fait pour les fichiers du répertoire Test, quelle est la valeur de umask ?
 2. Proposer une valeur de umask permettant de supprimer pour les groupes le droit d'écriture sur les fichiers et sur les répertoires. Comment faire le changement ? Quelle sera la valeur numérique correspondant aux droits des fichiers et des répertoires ?
 3. Que se passe t-il si nous décidons de mettre la valeur de umask à 0 ?

Exercice 2

1) Quelle est la version numérique de la commande chmod u=rwx,g=rw toto.txt ?

2) Quelle est la version symbolique de la commande chmod 700 test.sh ?

3) Soit l'exemple de session linux suivant:

```
[mirian@home2 Teste]$ ls -l  
drwxrwxr-x 2 mirian mirian 4096 Sep 18 23:17 OS  
-rw-rw-r- 1 mirian mirian 0 Sep 18 23:17 titi  
-rw-rw-r- 1 mirian mirian 0 Sep 18 23:17 toto
```

Expliquez ce que fait chacune des commandes ci-dessous, dans l'ordre présenté:

1. chmod 400 *
2. chgrp l3info OS
3. chmod a+x *
4. touch OS/tata
5. chown bineta;l3info toto

Université Assane Seck de Ziguinchor
UFR Sciences et Techniques
Département d'informatique
Licence 3 Informatique

Linux - Travaux Pratiques
Fiche 3 – Sécurité, mécanismes de droit

Exercice 1

1. Créez les répertoires licence et master dans le répertoire /home (penser à utiliser sudo).
2. Créez les répertoires tapha et adama dans le répertoire /home/licence.
3. Créez le répertoire bineta dans /home/master.
4. Créez les utilisateurs bineta et youssou avec la commande **adduser** (alias).
5. Créez l'utilisateur tapha avec pour home /home/licence/tapha en utilisant **useradd**. Initialiser son mot de passe avec **passwd**. Rendre tapha propriétaire de son home.
6. Créez les groupes l3in, m2gl et informatique.
7. Affichez les groupes de l'utilisateur tapha. Affectez l'utilisateur tapha au groupe l3in comme groupe primaire.
8. Créez l'utilisateur adama avec pour home /home/licence/adama et pour groupe primaire l3in en utilisant **useradd**. Initialisez son mot de passe. Rendre adama propriétaire de son home.
9. Affichez les groupes de bineta. Affectez l'utilisateur bineta au groupe m2gl comme groupe primaire et informatique comme groupe secondaire.
10. Modifiez l'actuel home de bineta (/home/bineta) pour qu'il soit /home/master/bineta.
11. Affectez les utilisateurs bineta, tapha et adama au groupe informatique comme **groupe secondaire** tout en conservant les groupes auxquels ils appartiennent déjà.

Exercice 2

1. Connectez vous comme tapha sur un terminal.
 1. Donnez le chemin du home de tapha et ses groupes d'appartenance.
 2. Créez les fichiers fic1 et fic2 dans le home de tapha en y mettant du contenu.
 3. Qui est propriétaire de ces fichiers (utilisateur et groupe) ? Quels sont les droits par défaut sur ces fichiers ?
 4. Ajoutez le droit de modification (w) pour le groupe propriétaire (g) sur le fichier fic1. Enlever le droit de lecture (r) aux autres (o) sur le fichier fic2
2. Connectez vous comme bineta sur un deuxième terminal.
 1. Donnez le chemin du home de bineta et ses groupes d'appartenance.
 2. Créez les répertoires rep1 et rep2 dans le home de bineta, créez le fichier fic11 dans rep1 et le fichier fic21 dans rep2, tous les deux avec du contenu.

3. Qui est propriétaire de ces répertoires ? Quels sont les droits par défaut sur ces répertoires ?
 4. Ajouter le droit de modification (w) pour le groupe propriétaire sur le répertoire rep1. Enlever le droit de lecture (r) et d'exécution (x) au groupe propriétaire (g) et aux autres(o) sur le répertoire rep2
3. Connectez vous comme adama sur un troisième terminal.
1. adama peut il lister le contenu du home de tapha? Pourquoi ?
 2. adama peut il créer des fichiers et répertoires dans le home de tapha ? Pourquoi ?
 3. Idrssa peut il afficher le contenu des fichiers fic1 et fic2? Pourquoi ?
 4. adama peut il modifier le contenu des fichiers fic1 et fic2? Pourquoi ?
 5. adama peut il lister le contenu du home de bineta ? Pourquoi ?
 6. adama peut il lister le contenu des répertoires rep1 et rep2 ? Pourquoi ?
 7. adama peut il créer des fichiers dans rep1 et rep2 ? Pourquoi ?
 8. adama peut il accéder (avec cd) à rep1 et rep2 ? Pourquoi ?

Exercice 1

Le répertoire /usr/include contient les fichiers d'entête standards en langage C (stdlib.h, ...).

1. Créer un répertoire nommé inc dans le répertoire TP2.
2. En utilisant une seule commande, y copier les fichiers du répertoire /usr/include dont le nom commence par std.
3. Afficher la liste des fichiers de /usr/include dont le nom commence par a, b ou c.
4. Modifier la commande de la question précédente pour qu'au lieu d'afficher le résultat, celui-ci soit placé dans un fichier nommé "abc.list" situé dans TP2.
5. Afficher le contenu de ce fichier en utilisant la commande cat.
6. Afficher le nombre de ligne de ce fichier.
7. En une seule ligne de commande, afficher le nombre de fichiers de /usr/include dont le nom contient la lettre t. *ls -a /bin/*t* | wc -l*

Exercice 2

1. Soit le script bash maboucle.sh suivant, éditer le script dans TP2. Que fait ce script ?
2. La commande kill permet d'envoyer un signal à un processus. **kill -l** permet d'afficher les signaux possibles. Quels sont les numéros des signaux KILL, CONT, STOP, INT ?
3. Lancer le script maboucle dans un terminal.
4. Dans un second terminal, déterminer le numéro du processus (PID) maboucle.
5. Envoyer le signal STOP (19) au processus ma boucle avec la commande :
kill -19 NUMRPOC
ou NUMPROC désigne le numéro du processus maboucle. Que constatez vous ?
6. Envoyer le signal CONT (18) au processus avec la commande :
kill -s CONT NUMPROC
Que constatez vous
7. Dans un troisième terminal, lancer le script maboucle. Quel est le PID de ce processus.
8. Envoyer le signal KILL (9) à ce processus, que constatez vous ?

```
i=0
while true
do
    i=${i+1}
    echo "valeur : $i"
    sleep 2
done
```

Université Assane Seck de Ziguinchor

UFR Sciences et Techniques

Département d'informatique

Licence 3 Informatique

Linux - Travaux Dirigés

Fiche 4 – Commandes de bases, Shell, processus, filtres et redirection

Exercice 1

1. Quelle différence faites vous entre l'invite de commande (**prompt**) et l'interpréteur de commande (**shell**)?
2. Que font les commandes suivantes :

\$ ls /etc/rc???	\$ echo /etc/rc???	\$ touch '[te]*'
\$ touch [te]*	\$ rm file[1-8]	\$ ls [a-m]?*txt
3. afficher les fichiers de /dev dont le nom commence par tty suivi de deux caractères quelconque
4. afficher les fichiers de /dev dont le nom commence par tty suivi d'une lettre, suivi d'un nombre quelconque de caractère
5. créer en une seule commande les fichiers tic1, tic2, tic3, tic4 et tic5.

Exercice 2

Soit le contenu du fichier fic1.txt suivant:

Monsieur
Madame
Zen
Aristote

1. Quel est le processus ancêtre de tous les processus ? Que contient la variable d'environnement PATH ?
2. Que contiendra le fichier fic2.txt après l'exécution des commandes suivantes ?
`$ cat fic1.txt | sort > fic2.txt`
`$ cat fic1.txt | wc -l >> fic2.txt`
3. Que contiendra le fichier fic1.txt après l'exécution des commandes suivantes?
`$ cat fic1.txt | sort > fic1.txt`
`$ ls -l ~ | wc -l >> fic1.txt`
4. Que font les commandes suivantes?
`$ ls -l ~ | head -5 >> fic1.txt 2>erreur.txt`
`$ head -n -5 <liste.txt | tail -8 >resultat.txt 2>/dev/null`

Université de Ziguinchor
UFR Sciences et Technologies
Département d'informatique
Licence 3 Informatique

Contrôle continu Système d'exploitation - Linux
Durée: 2h.

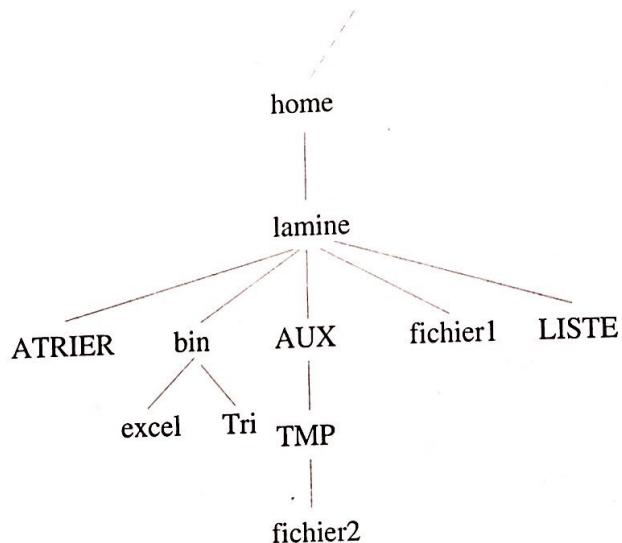
Documents autorisés

Exercice 1: Questions de cours (2 pts)

1. Dans la norme FileSystem Hierarchy Standard (FHS), quel est le contenu des répertoires suivants : /etc, /bin, /dev, /home
2. Qu'est ce qu'une distribution Linux? Donnez deux exemples?

Exercice 2: Système de fichiers (8 pts)

Pour chaque question, la position courante (Working Directory - WD) de l'utilisateur est indiquée entre parenthèses et il vous est demandé de donner les commandes sans changer de répertoire avec cd.



1. (WD : lamine) Créer un répertoire dans lamine?
2. (WD : lamine) Créer un lien physique **lien** dans ATTRIER qui pointe sur **fichier2**

3. (WD : lamine) Supprimer les fichiers **excel** et **fichier1**
4. (WD : ATRIER) Créer un fichier **linux** dans le répertoire **LISTE** ?
5. (WD : ATRIER) Déplacer le fichier **fichier2** de **TMP** à **AUX** ?
6. (WD : LISTE) Créer le fichier **fic5** dans le répertoire **home**?
7. (WD : LISTE) Créer un lien symbolique **symb** dans **LISTE** qui pointe sur **excel**
8. (WD : home) Renommer le fichier **fichier1** en **fic1** ?

Exercice 3: script signe du produit (4 pts)

Proposer un script shell qui demande deux nombres entiers à l'utilisateur et l'informe ensuite si le produit est négatif, positif ou nul.

NB : on ne doit pas calculer le produit.

Exercice 4: script opérations sur n (6 pts)

1. Écrire une fonction *somme* qui calcule la somme des nombres de 1 à n (2 points).
 2. Écrire une fonction *factoriel* qui calcule le factoriel de n (2 points).
 3. Écrire un script shell **operations.sh** qui permet de calculer au choix la somme des nombres de 1 à n ou le factoriel de n (Il faut juste appeler les méthodes proposées).
- Le script prend ses deux paramètres en ligne de commande : le premier paramètre est l'opération souhaitée (somme/factoriel) et le deuxième paramètre est l'entier n.

Université Assane Seck de Ziguinchor
UFR Sciences et Technologies
Département d'Informatique
L3 Informatique

Examen Système d'exploitation - Linux
Durée: 1h30.

La consultation des documents est autorisée.

Exercice 1: 10 pts

Voici une suite de commandes exécutées par l'utilisateur arnaud :

```
arnaud@mamachine:~/tmp$ pwd
/home/alioune/tmp
arnaud@mamachine:~/tmp$ ls -l
drwxrwxr-x 2 alfred labesse 36573 Nov 20 1996 Offre
-rwxrwxr-x 2 alfred almira 36573 Nov 20 1996 Offre.bash
-rw-rw-r-- 2 alfred almira 36573 Nov 20 1996 Offre.txt
-rwxrwxr-x 2 arnaud labesse 36573 Nov 20 1996 OffreDell.sh
-rwxrwxr-- 2 alfred labesse 36573 Nov 20 1996 OffreIbm.sh
arnaud@mamachine:~/tmp$
```

1. À quel utilisateur appartient le fichier `Offre.bash`? À quel groupe appartient le fichier `OffreDell.sh`?
2. Donnez la commande qui fait de alfred l'utilisateur propriétaire du fichier `OffreDell.sh` et de almira le groupe propriétaire.
3. Quels sont les droits en mode numérique sur le fichier `Offre.txt`? et sur le répertoire `Offre`?
4. Donnez la commande qui enlève le droit d'écriture au groupe et le droit d'exécution aux autres sur tous les fichiers d'extension `.sh`.
5. Quel est le répertoire de travail de arnaud (Working Directory)?
6. Donnez la commande qui permet à arnaud de supprimer les fichiers du répertoire `Offre` d'extension `.odt` et dont le nom ne commence ni par `a`, ni par `c`.
7. Donnez la valeur de masque (umask) qui permet de donner les droits par défaut suivants sur les fichiers: droit de lecture et d'écriture pour le propriétaire, droit de lecture pour le groupe et aucun droit pour les autres (la démarche est notée, merci de la consigner dans le document).
8. Donnez la commande qui définit pour alfred `13in` comme groupe primaire et `licence et informatique` comme groupes secondaires.

Exercice 2: 5 pts

Écrire un script shell **equation** qui permet de résoudre une suite d'équations du premier degré.
Les paramètres sont dans un fichier *entree.txt* sous la forme :

a:b
a:b
a:b
a:b
0:b

Ils seront ainsi lu de ce fichier (1.5) et les équations résolus jusqu'à ce que à soit égal à 0. Les résultats seront à mettre dans un fichier *resultat.txt* (1).

Exercice 3 : 5 pts

1. Donner la commande qui affiche le nombre de lignes et de caractères des fichiers d'extension *.txt* dont le nom commence par O et est composé de 8 caractères
2. Donner la commande qui ajoute le texte de la première ligne d'un fichier *fic1* à la fin d'un fichier *fic2* sans l'effacer.
3. Donner la commande qui affiche de la 20 ème à la 30 ème ligne d'un fichier *fic3* en les numérotant à partir de 1¹.
4. Donner la commande qui permet d'afficher les informations relatives aux bus USB du système et aux périphériques qui y sont connectés.
5. Donner la commande qui permet de trier ensemble le contenu de trois fichiers *fic1*, *fic2* et *fic3* puis affiche les 20 premières lignes de ce tri.

¹la commande **n1** permet de numérotter les lignes d'un fichier