

CAHIER DE TRAVAUX PRATIQUES

Linux, Mise en œuvre

TABLE DES MATIÈRES

Travaux Pratiques

- TP0 – **Optionnel** : si le système hôte n'est pas Linux – Mise en route de la VM Debian
- TP1 – Le système de fichiers
- TP2 – La gestion des utilisateurs
- TP3 – Les droits sur les fichiers
- TP4 – Le shell
- TP5 – La gestion des paquets
- TP6 – Les services
- TP7 – La chaîne de démarrage
- TP8 – Le réseau
- TP9 – SSH

Corrigés des Travaux Pratiques

- TP1 – Le système de fichiers
- TP2 – La gestion des utilisateurs
- TP3 – Les droits sur les fichiers
- TP4 – Le shell
- TP5 – La gestion des paquets
- TP6 – Les services
- TP7 – La chaîne de démarrage

CAHIER DE TRAVAUX PRATIQUES

TUTORIEL

Mise en route de la machine virtuelle Debian

TABLE DES MATIÈRES

1. Introduction et rappels.....	1
1.1. A propos de ce TP.....	1
1.2. Rappel : Qu'est-ce qu'une machine virtuelle ?.....	1
1.3. L'architecture réseau pour la création de la machine virtuelle.....	2
1.4. Les « snapshots ».....	2
2. Mise en route de la machine virtuelle Debian.....	3
2.1. Utilisateurs root et tlinux.....	3
3. Rappel : les bases Linux.....	4
3.1. Structure d'une ligne de commande.....	4
3.2. L'auto-complétion et l'historique.....	4
3.3. Les éditeurs.....	4
4. Personnaliser votre système.....	6
4.1. Paramétrage IP.....	6
4.2. Le magasin d'applicatif Debian : apt.....	7
4.3. Le multiplexeur de terminal tmux.....	7

1. INTRODUCTION ET RAPPELS

1.1. A propos de ce TP

Objectif : Mettre en route une machine Virtuelle Linux qui servira à la réalisation des TP sous Linux.

Conditions : Les ordinateurs sont équipés du logiciel de virtualisation VirtualBox. La machine virtuelle a été préparée par votre formateur.

Déroulement : Ce TP consiste en une série de manipulations sur le logiciel VirtualBox.

1.2. Rappel : Qu'est-ce qu'une machine virtuelle ?

La virtualisation en informatique consiste à simuler (ou émuler) des composants physiques d'une machine (processeur, mémoire, disque, réseau, vidéo, son, etc.). Cela permet alors de créer un ordinateur « virtuel » (simulé ou émulé) sur un vrai ordinateur physique.

La virtualisation repose sur les principes suivants :

- Un système d'exploitation principal, appelé « **système hôte** », est installé sur l'ordinateur physique. Il sert à piloter les composants matériels physiques (à l'aide de pilotes), et de gérer les programmes (ordonnancement, etc.).
- Un logiciel de virtualisation, appelé « **hyperviseur** », est installé sur le système hôte. Il permet la création de machines virtuelles. L'hyperviseur garantit l'indépendance et l'isolation des machines virtuelles, entre elles, mais aussi vis à vis du système hôte.

- Un « **système invité** » est le système d'exploitation installé dans une machine virtuelle. Pour pouvoir fonctionner il dispose d'un accès aux ressources du système hôte (mémoire, espace disque, réseau, etc.), gérées par l'hyperviseur.

La virtualisation est donc une abstraction entre les composants d'une machine physique et ceux d'un système invité. Cela apporte certains intérêts :

- La facilité de déplacement d'une machine virtuelle d'une machine physique à une autre, indépendamment de son architecture matérielle physique, de la nature de ses composants et de ses ressources.
- Une machine virtuelle peut être entièrement sauvegardée, à chaud et sans interruption, facilitant la remise en service d'un état fiable en cas de panne ou de compromission, sans devoir tout réinstaller.
- Les machines virtuelles peuvent être clonées et installées sur de nouveaux serveurs physiques, facilitant le dimensionnement en fonction du besoin (évolution des performances des machines physiques, répartition de charges).
- Les composants matériels d'une machine virtuelle peuvent être émulés, permettant par exemple de simuler une architecture ARM sur un x86, un 64 bits sur un 32 bits, un MacOS sur un Windows, etc.

1.3. Les « snapshots »

VirtualBox offre la possibilité de réaliser des instantanés (« snapshots ») de machines virtuelles. Un instantané est un enregistrement de l'état d'une machine virtuelle à un moment donné. Ainsi, après avoir pris un instantané, il est possible de revenir à cet état à n'importe quel moment, même si la machine virtuelle a été complètement modifiée, formatée, endommagée, compromise, etc.

Les instantanés sont particulièrement utiles pour disposer d'un « point de retour arrière » en cas de problème(s) consécutif(s) à une manipulation de la machine (ajout, suppression ou modification de fichiers ou de logiciels comme lors de migration).

En administration des systèmes nous nous servons très souvent de cette fonctionnalité.

ATTENTION : en général (cela dépend des hyperviseurs) un snapshot n'est pas une sauvegarde. Il contient (en général) un différentiel entre l'état au moment du snapshot et l'état actuel.

Il n'est pas recommandé de conserver un état « snapshot » trop longtemps. Une fois que la manipulation nécessitant son emploi est terminée il est recommandé (dans un délais de quelques minutes à quelques semaines) de fusionner le snapshot avec l'état actuel (valider et accepter les changements) ou de revenir à l'état du snapshot.

Dans VirtualBox, un snapshot se fait en allant dans le menu « Machine » et cliquant sur « Prendre un instantané ».

2. MISE EN ROUTE DE LA MACHINE VIRTUELLE DEBIAN

2.1. Si votre hôte est une machine Windows

- **IMPORTANT :** Afin d'éviter des problèmes de conflits de cartes réseau, allez dans le gestionnaire des cartes réseau de votre ordinateur physique et désactivez toutes les cartes réseaux (sauf votre carte réseau principale).
- **IMPORTANT :** Désactivez l'Hyper-V de windows en tapant les commandes suivantes dans une console de commandes de Windows ouverte en administrateur :

```
dism /Online /Disable-Feature:Microsoft-Hyper-V
```

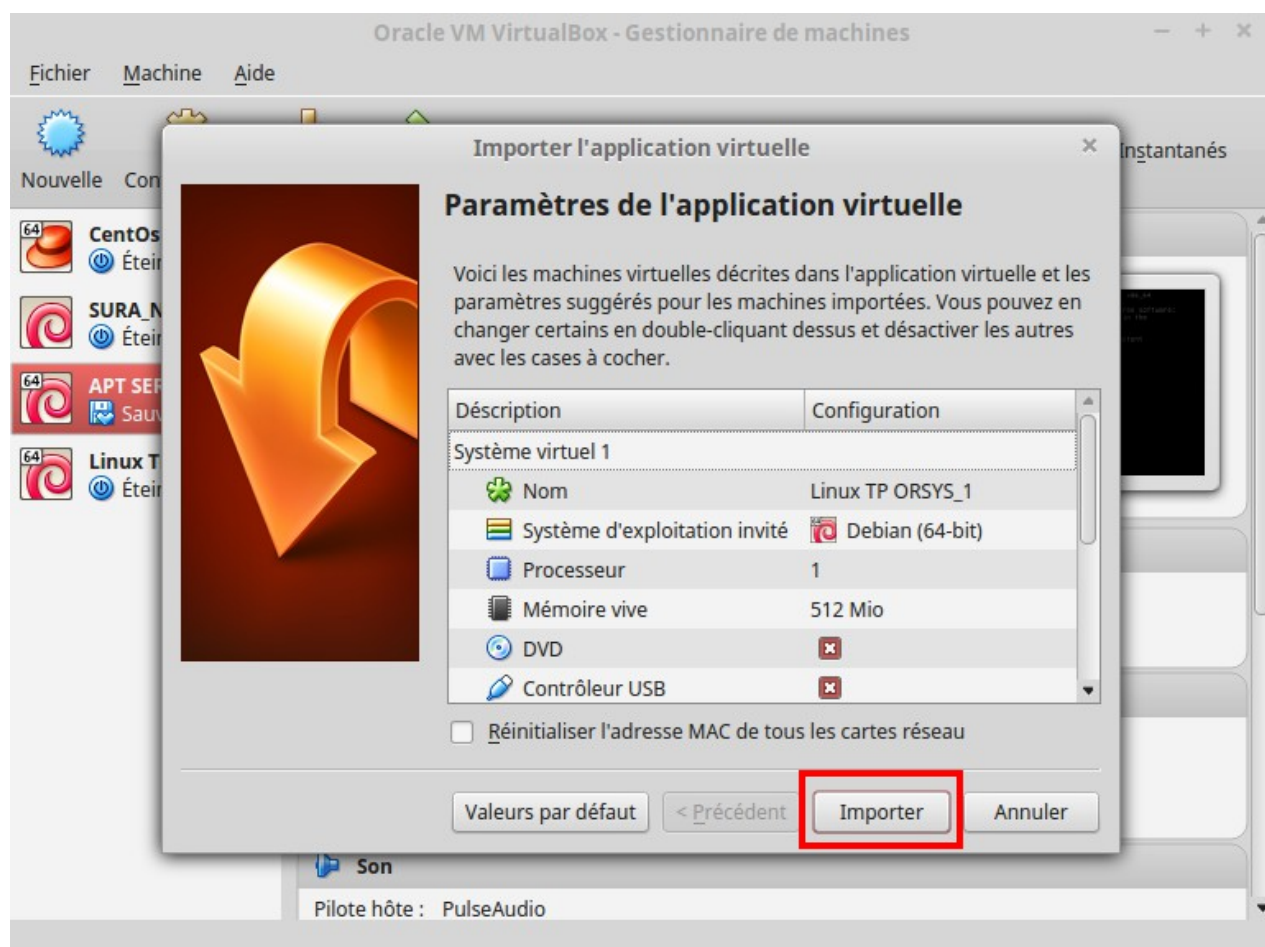
Le système va demander de redémarrer, ne pas le faire de suite. Saisissez la commande:

```
bcdedit /set hypervisorlaunchtype off
```

Redémarrez ensuite votre PC.

2.2. Importation de la VM

- Récupérez l'image de Debian (.ova) proposée par votre formateur.
- Cliquez deux fois sur cette image, **décochez la case « Contrôleur USB »**
- **Cochez la case « Réinitialiser l'adresse MAC de toutes les cartes réseau »**
- Cliquez sur le bouton Importer



- Vous pouvez ensuite démarrer votre VM et vous connecter avec l'utilisateur tpinux.

2.3. Utilisateurs root et tlinux

Lors de l'installation de Linux, deux utilisateurs ont été créés, root et tlinux. Afin d'effectuer la plupart des manipulations qui vont suivre, il faudra la plupart du temps être connecté en utilisateur root. Pour se faire, deux solutions :

- Au démarrage de votre Linux, lorsque le système vous demande le login, entrez **root**, validez (touche entrée) et saisissez le mot de passe root ("*root*" par défaut).
- Si vous êtes connecté en tlinux, tapez la commande `$ su` pour saisissez votre mot de passe root.
- Pour sortir du mode « utilisateur root », tapez la commande `# exit`.

➔ Pour connaître l'utilisateur avec lequel vous vous êtes connecté, tapez la commande

```
$ whoami
```

➔ Notez que le dernier caractère de votre prompt (chaîne de caractères invitant l'utilisateur à taper une commande) est différent selon votre utilisateur connecté : `$` pour tlinux, `#` pour root.

Le mot de passe de tlinux est par défaut "*tlinux*"

Les parties 3 et 4 ne sont pas à réaliser si vous êtes dans un cours ayant pour thème « Linux ».

3. RAPPEL : LES BASES LINUX

3.1. Structure d'une ligne de commande

La structure d'une ligne de commande linux est la suivante :

Nom + [options] + <argument(s)>

Par exemple :

```
$ cp L -R L repertoire1 L repertoire2
```

\$: fin de votre ligne de prompt

L : espace

-R : option signalée avec le signe -

repertoire1 : argument 1, répertoire source

repertoire2 : argument 2, répertoire destination

A noter :

- Entre chaque partie de la commande, il y a une espace ;
- Dans les manuels de TP qui suivent, les commandes sont toujours signalées avec le dernier caractère du prompt (\$ ou #). Ne pas ressaisir ce caractère lorsque vous tapez la commande !
- Les arguments sont notés entre <>. Par exemple : `# ifconfig eth0 <ip_wan>`.

Vous devez alors remplacer cet argument en fonction de votre environnement en enlevant les <>

Par exemple : `# ifconfig eth0 192.168.1.1`

3.2. L'auto-complétion et l'historique

La fonction d'auto-complétion, très pratique, complète le nom d'un fichier, d'un répertoire ou d'une commande. Elle s'utilise en tapant la touche [TAB]. Si un seul fichier, dossier ou commande commence par ce que vous avez commencé à saisir, alors [TAB] complètera le nom pour vous. Si plusieurs fichiers, dossiers ou commandes commencent par ce que vous avez saisi, alors en tapant une nouvelle fois sur [TAB] le shell vous affichera les choix possibles.

➔ Essayez : `cd /ho[TAB]`. Observez que le shell a ajouté le 'me' à la fin de home. En effet il n'existe qu'un seul répertoire à la racine et commençant par « ho », c'est « home ».

➔ Essayez : `cd /b[TAB]`. Observez que rien ne se passe. Tapez une nouvelle fois [TAB] et observez que le shell vous affiche les solutions possibles : « bin » et « boot ».

En tapant sur **les flèches du haut ou du bas**, vous avez accès aux commandes précédemment tapées durant la session. Cela peut s'avérer très pratique pour retrouver rapidement une commande précédemment saisie.

La commande history permet de visualiser un historique des commandes de l'utilisateur saisies dans le terminal courant.

➔ Essayez `$ history`

Vous verrez ainsi les commandes tapées avec un numéro devant. On peut donc utiliser ce numéro pour relancer des commandes sans avoir à les retaper en utilisant le caractère spécial Par exemple, pour appeler la commande numéro 10 de l'historique : `$!10`

3.3. Les éditeurs

Lors de l'installation de Debian, deux éditeurs sont installés par défaut : vi et nano.

Si vous êtes habitué à l'utilisation de vi, nous vous conseillons d'utiliser l'éditeur vim, une version plus ergonomique (et plus lourde) de l'éditeur vi.

Vi étant un peu délicat à prendre en main, si vous n'êtes pas un habitué, nous vous conseillons d'utiliser **nano** pour l'édition de fichier.

➔ Pour prendre en main cet éditeur, éditez un nouveau fichier avec la commande `# nano new_file`.

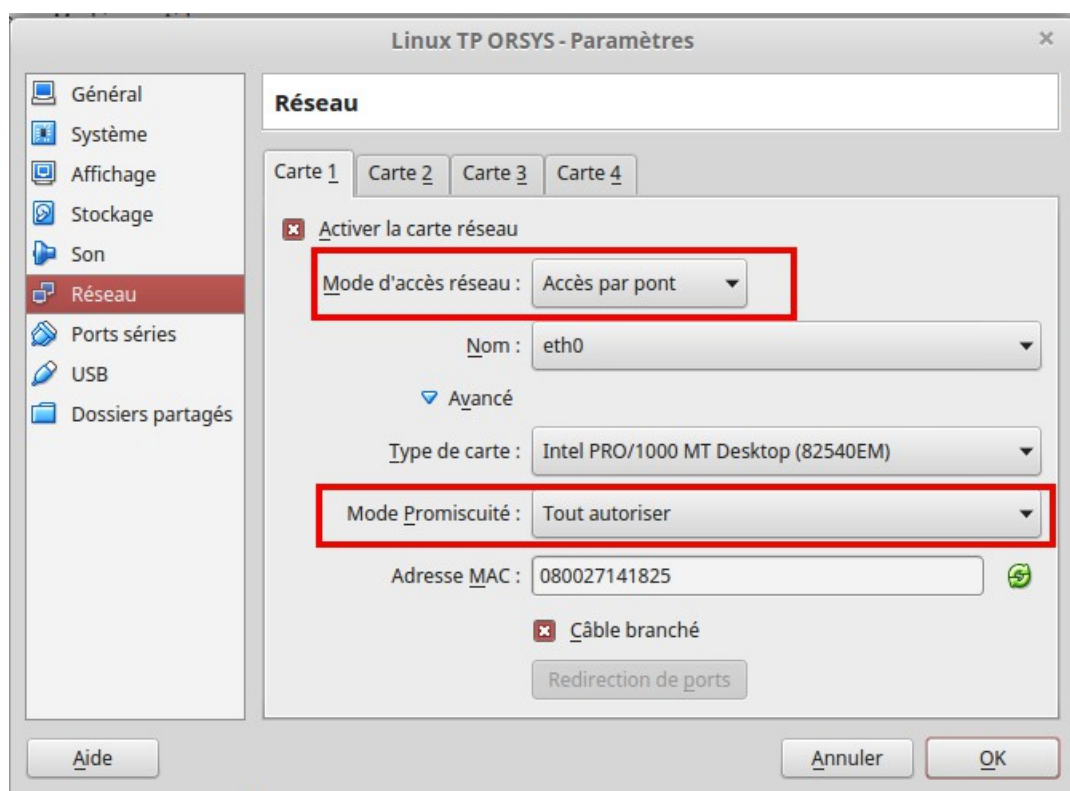
- En bas de nano, vous avez les raccourcis des commandes de base. Le ^ signifie la touche Ctrl. Pour sauvegarder votre fichier, tapez simultanément sur les touches Ctrl et O (Ctrl et X pour quitter). Nano vous demandera alors de confirmer l'enregistrement (tapez O pour oui) et le nom du fichier lors de l'enregistrement (tapez Entrée pour confirmer).

4. PERSONNALISER VOTRE SYSTÈME

4.1. Paramétrage IP

Si votre système doit être accessible depuis votre poste physique par le réseau (SSH, HTTP, autre) et afin que votre machine virtuelle ait la même IP en fixe après redémarrage :

- Sélectionnez votre machine virtuelle tlinux dans VirtualBox, puis cliquez sur le bouton « Paramètres ».
- Cliquez sur le sous-menu **Réseau**. Sur l'onglet Carte 1, cochez « Activer la carte » puis choisissez les options comme ci-dessous pour passer la carte en Accès par pont :



- Sur la machine Linux, ouvrez avec un éditeur de votre choix le fichier /etc/network/interfaces. Le fichier doit être du format suivant :

Interface loopback

```
auto lo
iface lo inet loopback
```

Interface à adresse statique

```
allow-hotplug eth0
auto eth0
iface eth0 inet static
    address <ip>
    netmask <mask>
    gateway <ip_gw>
```

- Modifiez-le si nécessaire.

Les DNS se trouvent dans le fichier /etc/resolv.conf.

→ Ouvrez avec un éditeur de votre choix le fichier `/etc/resolv.conf` et vérifiez les IP des deux DNS. Modifiez-les si nécessaire en vérifiant que la première ligne du fichier est **nameserver 1.1.1.1**

→ Pour prendre en compte les nouveaux paramètres réseaux, tapez la commande :

```
# /etc/init.d/networking restart
```

4.2. Le magasin d'applicatif Debian : apt

Le magasin applicatif facilite le téléchargement et l'installation de programmes Linux, en résolvant par exemple automatiquement les problèmes de « dépendance ». Sous Debian les outils de gestion du magasin sont principalement « apt » et « aptitude ».

La configuration des dépôts des paquets Debian se trouve dans le fichier `/etc/apt/sources.list`.

→ Éditez ce fichier, et modifiez la valeur `<ip_serveur_apt>`

→ Commentez les 2 lignes avec la source <http://security.debian.org> si votre formateur vous le demande.

→ Sauvegardez le fichier et mettez à jour l'index des paquets APT avec la commande :

```
# aptitude update
```

Ne prenez pas en compte les erreurs.

Pour installer les outils pratiques, la commande d'installation est :

```
# aptitude install <nom_paquet>
```

4.3. Le multiplexeur de terminal tmux

→ Installez le paquet tmux.

Ce multiplexeur de terminal permet de :

- Détacher / rattacher / fusionner des sessions.
- Partager des sessions entre plusieurs personnes.
- Renommer des écrans ou des sessions.

Les multiplexeurs de session sont souvent utilisés en administration et supervision distante car :

- en cas de perte de lien la session reste active en fond de tâche, les processus en cours ne sont donc pas sauvagement tués,
- il est possible de travailler à plusieurs sur une session distante,
- Les sessions sont subdivisables, évitant ainsi la multiplication des instances de terminal.

→ Lancez-le :

```
# tmux
```

Un bandeau vert apparaît alors en bas :

- Divisez horizontalement l'écran : `Ctrl+b` puis `"`
- Divisez verticalement l'écran : `Ctrl+b` puis `%`
- Naviguez entre régions : `Ctrl+b` puis flèches
- Fermez une région : `Ctrl+d`
- Créez un écran : `Ctrl+b` puis `c` (un nouvel nom apparaît en bas)
- Revenez à l'écran précédent : `Ctrl+b` puis `p`
- Revenez à l'écran suivant : `Ctrl+b` puis `n`
- Affichez l'aide : `Ctrl+b` puis `?`
- Sortez de l'aide : `q`
- Détachez cet environnement : `Ctrl+b` puis `d`

- Réattachez-vous : `tmux a`

Lorsqu'on affiche le contenu d'un fichier très long (par ex. `/etc/services`) :

```
# cat /etc/services
```

- Remontez l'écran : `Ctrl+b` puis `pageup` (la position est indiquée en haut de l'écran)
- Quittez le navigateur de page : `q`

➔ Créez un nouveau Snapshot de votre machine virtuelle.

TRAVAUX PRATIQUES

Le système de fichiers

TABLE DES MATIÈRES

1.Introduction et rappels.....	1
1.1.A propos de ce TP.....	1
1.2.Remarque.....	1
1.3.Commandes utiles dans le cadre de ce TP.....	1
2.La manipulation de fichiers.....	2
2.1.La navigation dans l'arborescence.....	2
2.2.Les liens.....	3
2.3.Les archives TAR.....	3
2.4.Les caractères spéciaux du Bash.....	4
3.Les éditeurs.....	4
3.1.Nano.....	4
3.2.Vi et Vim.....	5
4.Annexe : commandes utiles vi.....	6
4.1.Mode immédiat.....	6
4.2.Mode insertion.....	6
4.3.Mode commande ligne.....	6
4.4.Autres.....	6

1. INTRODUCTION ET RAPPELS

1.1. A propos de ce TP

Objectif : Manipuler et maîtriser les concepts de base de la gestion des fichiers sous Linux.

Conditions : Les ordinateurs sont équipés d'un système d'exploitation Linux Red Hat, Cent Os ou Debian.

1.2. Remarque

Dans le cadre de ce TP nous allons être amenés à utiliser des commandes faisant appel à des arguments que vous choisirez vous-même. Ces arguments sont signalés entre <>. Par exemple : `# touch <nom_fichier>`. La commande que vous devrez taper ne contiendra donc pas les <> et deviendra par ex `# touch mon-premier-fichier`

1.3. Commandes utiles dans le cadre de ce TP

Dans le cadre de ce TP nous allons être amenés à installer des paquets logiciels.

Pour se faire, sous **Debian**, vous devrez utiliser la commande de gestion de paquets « apt ». Cette commande s'utilise ainsi : `# apt install <paquet>`, où « <paquet> » est le nom du paquet à installer (sans les signes '<' et '>' évidemment).

Sous **Red Hat** ou **CentOs**, vous devrez utiliser la commande de gestion de paquets « yum ». Cette commande s'utilise ainsi : `# yum install <paquet>`, où « <paquet> » est le nom du paquet à installer (sans les signes '<' et '>' évidemment).

2. LA MANIPULATION DE FICHIERS

2.1. La navigation dans l'arborescence

→ Authentifiez-vous sur la machine virtuelle Linux avec l'utilisateur tlinux ou usera

[1] Dans quel répertoire êtes-vous ? Quelle commande vous permet de le savoir ?

[2] Allez à la racine du système. Quelle commande avez-vous utilisé ?

[3] Listez le contenu du répertoire courant. Quelle commande avez-vous utilisé ?

Note : il est également possible d'énumérer le contenu d'un répertoire en utilisant l'auto complétion du shell. Cette fonction, très pratique, complète le nom d'un fichier ou d'un répertoire. Elle s'utilise en tapant la touche [TAB] lors de l'appel d'une commande. Si un seul fichier ou donnée existe et commençant par ce que vous avez commencé à saisir, alors [TAB] complètera le nom pour vous. Si plusieurs fichiers ou dossiers commencent par ce que vous avez saisi, alors en tapant une nouvelle fois sur [TAB] le shell vous affichera les choix possibles.

→ Essayez : `cd /hom[TAB]`. Observez que le shell a ajouté le 'e' à la fin de home. En effet il n'existe qu'un seul répertoire à la racine et commençant par « hom », c'est « home ».

→ Essayez : `cd /b[TAB]`. Observez que rien ne se passe. Tapez une nouvelle fois [TAB] et observez que le shell vous affiche les solutions possibles : « bin » et « boot »

[4] Quelle commande utilisez-vous pour aller dans /root ?

[5] Depuis là, quelle commande utilisez-vous pour aller dans /etc, en relatif ?

[6] Votre touche '/' est cassée ! Quelle commande utilisez-vous pour aller dans votre home ?

Le principal avantage de cette notation est qu'elle permet de manipuler un home directory courant sans avoir à connaître le login utilisé. Ceci permet la création de scripts génériques qui, quelque soit l'utilisateur qui l'appelle, sauront toujours manipuler le home courant.

[7] Quelle commande crée un répertoire « tmp » dans le home de l'utilisateur qui l'appelle, quelque-soit son identité ?

→ Passez en utilisateur root avec la commande `$ su`. Installez l'utilitaire **tree** avec la commande `# apt install tree` (Debian) ou `# yum install tree` (CentOs ou RedHat). Repassez en utilisateur normal avec la commande `# exit`. Testez l'outil tree.

→ Restez en root. En utilisant l'option `--parent` de la commande `mkdir`, la commande `touch` et dès que possible l'auto-complétion (touche [TAB]), créez l'arborescence suivante. Vérifiez avec la commande `tree` votre arborescence.

Remarque : usera est à remplacer par tlinux en fonction de votre installation.

```
usera
├── mon-fichier
├── mon-rep-1
│   ├── mon-premier-fichier
│   └── mon-ssrep-11
├── mon-rep-2
│   └── mon-ssrep-21
│       └── mon-fichier-21
```

[8] Quelle suite de commandes avez-vous utilisé ?

2.2. Les liens

→ Créez un lien symbolique appelé « mon-fichier-lien » dans le dossier mon-rep-2 du fichier mon-fichier avec la commande `ln` et l'option `s`.

[9] Quelle commande avez-vous utilisé ? Quels sont les i-nodes des deux fichiers ?

→ Écrivez dans le fichier mon-fichier en utilisant une redirection

`echo "bonjour" > mon-fichier` et lisez le fichier « mon-fichier-lien » avec la commande

`more mon-rep-2/mon-fichier-lien`

[10] Que constatez-vous ?

→ Supprimez « mon-fichier ».

[11] Que constatez-vous lorsque vous listez les fichiers de « mon-rep-2 » ?

2.3. Les archives TAR

→ Allez dans mon-rep-1 et créez le fichier fichier.txt

→ Créez une archive tar non compressée et mettez-y le fichier « fichier.txt » :

`tar cvf monarchive.tar fichier.txt`

→ Listez le contenu de l'archive : `$ tar tvf monarchive.tar`

→ Comparez le fichier « fichier.txt » avec celui de l'archive :

`tar df monarchive.tar fichier.txt`

→ Modifiez le fichier « fichier.txt » : `$ date > fichier.txt`

→ Comparez le fichier « fichier.txt » avec celui de l'archive :

`tar df monarchive.tar fichier.txt`

→ Mettez à jour l'archive :

`tar uvf monarchive.tar fichier.txt`

→ Listez le contenu de l'archive : `$ tar tvf monarchive.tar`

→ Ajoutez un autre fichier à l'archive : `$ tar rvf monarchive.tar <autrefichier>`

→ Listez le contenu de l'archive

→ Extrayez le fichier « fichier.txt » : `$ tar xvf monarchive.tar fichier.txt`

(Notez que c'est le dernier qui est restauré)

→ Extrayez le fichier « fichier.txt » avec confirmation, pour ne restaurer que le premier :

`$ tar xvfw monarchive.tar fichier.txt`

2.4. Les caractères spéciaux du Bash

Les commandes suivantes sont à réaliser dans votre répertoire home. Repassez en utilisateur normal # `su usera` ou # `su tpinux`

→ Tapez la commande : `$ cp ~/mon-rep-1/m* ~/titi/`

[12] Que s'est-il passé ? Pourquoi ?

→ Tapez la commande : `$ cp -R ~/mon-rep-1/m* ~/mon-rep-2/`

[13] Qu'a réalisé cette commande ? Quelle autre commande aurait-on pu taper ?

→ Tapez la commande : `$ mkdir ~/toto`

[14] Qu'a réalisé cette commande ? Comment faire pour supprimer le dossier créé ?

[15] Donnez les commandes qui représentent les énoncés suivants :

- liste (ordonnée) des fichiers du répertoire courant commençant par la lettre a ou A et terminée par n'importe quoi ?
- liste des fichiers qui se terminent par l'extension .cxx du répertoire mon-rep-1 contenu dans le répertoire d'accueil de usera ?
- liste des fichiers du répertoire courant dont le nom contient une étoile ?
- liste des fichiers cachés contenus dans le répertoire d'accueil de tous les utilisateurs ?

[16] Donnez quelques combinaisons de caractères de protection qui permettent d'obtenir les détails du fichier nommé : un nom bizarre*

(Source des questions 15 et 16 : Cyril Pain-Barre, IUT Aix-en-Provence, sous licence CC)

3. LES ÉDITEURS

Un **éditeur de texte** est un programme qui permet de modifier des fichiers de texte brut, sans mise en forme (gras, italique, souligné...). Sous Windows, on dispose d'un éditeur de texte très basique : le Bloc-Notes. Sous Linux, on a le choix entre Nano, Vim, Emacs et bien d'autres, sachant qu'au moins un de ceux-là est installé par défaut sur la plupart des distributions.

Un **traitement de texte** est fait pour rédiger des documents mis en forme. Sous Windows, Word est le plus célèbre traitement de texte ; sous Linux, on possède l'équivalent : Open Office Writer. Ces programmes ne peuvent être utilisés qu'en mode graphique, la console ne permettant pas vraiment de faire de la mise en forme.

3.1. Nano

Nano est un programme très simple comparé à Vim et possède assez peu de fonctions par rapport à vi. Il conviendra néanmoins pour éditer rapidement des fichiers de configuration.

- ➔ Pour prendre en main cet éditeur, éditez un nouveau fichier avec la commande `# nano new_file`.
- ➔ En bas de nano, vous avez les raccourcis aux commandes de base. Le ^ signifie la touche Ctrl. Pour sauvegarder votre fichier, tapez simultanément sur les touches Ctrl et O (Ctrl et X pour quitter). Nano vous demandera alors de confirmer l'enregistrement (tapez O pour oui) et le nom du fichier lors de l'enregistrement (tapez Entrée pour confirmer).
- ➔ Consulter l'aide de Nano `# nano --help`

3.2. Vi et Vim

Vi est un des éditeurs de texte les plus populaires sous Linux malgré son ergonomie très limitée.

Vi possède 3 modes de fonctionnement :

- **Le mode normal**: celui dans lequel vous êtes à l'ouverture du fichier. Il permet de taper des commandes
- **Le mode insertion**: Ce mode permet d'insérer les caractères que vous saisissez à l'intérieur du document. Pour passer en mode insertion, il suffit d'appuyer sur la touche *Insert* de votre clavier, ou à défaut sur la touche *i* quand vous êtes dans le mode normal.
- **Le mode de remplacement**: Ce mode permet de remplacer le texte existant par le texte que vous saisissez. Il vous suffit de réappuyer sur *insert* (ou *ï*) pour passer du mode insertion au mode remplacement, et d'appuyer sur la touche *Echap* pour revenir en mode normal.

Nous allons préférer **vim** à **vi**, qui est une version améliorée de ce dernier mais qui s'utilise de la même manière.

→ Installez vim en étant root `# apt install vim`

→ éditez un nouveau fichier avec les commandes suivantes : `# vim montest`

Bienvenue dans l'éditeur de texte vi ! Lorsque vi s'ouvre, il est en mode normal. Pour passer en mode Insert :

- tapez `[i]` ou `[Insert]` pour insérer du texte à l'endroit où se trouve le curseur,
- tapez `[A]` pour ajouter du texte à la fin d'une ligne.

En mode Insert, vous pouvez taper du texte et effacer avec la touche `[Suppr]` ou `[Bkspc]`.

Pour quitter le mode Insert, tapez `[Esc]`.

→ Tapez quelques lignes de texte au hasard.

→ Repassez en mode normal en tapant `[Esc]`. Placez-vous sur n'importe quelle ligne en utilisant les flèches de direction (haut-bas).

→ Tapez `[y][y]` et `[p]`.

[17] Que se passe-t-il ?

→ Revenez à la première ligne du texte. Tapez `[2][y][y]`. Allez à la dernière ligne du texte et tapez `[p]`.

[18] Que se passe-t-il ?

→ Revenez à la première ligne du texte. Tapez `[4][y][y]`. Allez à la dernière ligne du texte et tapez `[p]`.

[19] Que se passe-t-il ?

→ Allez sur un caractère au hasard. Tapez `[x]`.

[20] Que se passe-t-il ?

→ Placez-vous sur n'importe quelle ligne et tapez `[d][d]`.

[21] Que se passe-t-il ?

→ Tapez `[u]`.

[22] Que se passe-t-il ?

[23] Dans la documentation *vi* placée en annexe de ce TP, trouvez comment afficher les numéros des lignes.

[24] Trouvez comment remplacer tous les caractères **a** par la chaîne **ab**.

→ Tapez `:[w][q][Entrée]` pour enregistrer votre document et quitter *vi*.

4. ANNEXE : COMMANDES UTILES VI

Ci-dessous, quelques commandes utiles dans les différents modes de vi.

4.1. Mode immédiat

Toute lettre est immédiatement interprétée comme (le début d')une commande.
Tout chiffre est pris comme facteur de répétition de la commande suivante.

Principales commandes :

- U Undo : annuler les modifications de la dernière commande.
- . Répéter sur la position courante la dernière commande qui ne soit pas un positionnement.

Modification :

- DD Supprimer la ligne courante et la sauver dans un tampon (n lignes si chiffre n avant)
- P Insérer le tampon devant la position courante
- X Supprimer le caractère sous le curseur
- D Supprimer la fin de ligne.
- R Remplacer caractère par caractère à partir de la position courante.
- J Joindre la ligne qui suit à la ligne courante.

Positionnement :

- Flèches
- G Se positionner à la ligne de numéro n ou n est le nombre tape avant G. Si aucun nombre n'a été tape avant, se positionne en fin de fichier.
- 0 Se positionner en début de ligne
- \$ Se positionner en fin de ligne.
- <ctrl>f Avancer d'un écran (Forward)
- <ctrl>b Reculer d'un écran (Backward)
- M a Marquer la position courante avec le nom "a"
- ' a Se positionner à la position marquée "a"

4.2. Mode insertion

On entre et on sort du mode insertion en tapant le caractère ESC. Tout ce qui est tape est entré :

- devant le curseur (commande immédiate "i")
- derrière le curseur (commande immédiate "a")
- à la fin de la ligne (commande immédiate "A")

4.3. Mode commande ligne

Les principales à connaître :

- :w Sauvegarder le texte dans le fichier courant (le nom utilise pour lancer vi).
- :w nom_fichier Écrire le texte dans nom_fichier
- :q Terminer la session (refuse si pas de sauvegarde)
- :q! Termine la session même si pas de sauvegardé.
- :x Termine la session et sauvegarder le texte.

Pour remplacer un texte :

- :s/caractere_a_remplacer/caractere_replacement/g Remplace sur la ligne courante
- :%s/caractere_a_remplacer/caractere_replacement/g Remplace dans tous le texte

Pour paramétrer les options de vi :

- :set Affiche toutes les options et leurs valeurs possibles
- :set nu Affiche les numéros de lignes
- :set nonu Ne plus afficher les numéros de lignes

4.4. Autres

La commande `# vi -r` fait la liste des fichiers récupérables après un plantage.

TRAVAUX PRATIQUES

La gestion des utilisateurs

TABLE DES MATIÈRES

1. Introduction et rappels.....	1
1.1. A propos de ce TP.....	1
1.2. Remarque.....	1
1.3. Commandes utiles dans le cadre de ce TP.....	1
2. La gestion des utilisateurs et des groupes.....	2
2.1. Les outils de base.....	2
2.2. Sudo.....	2
3. Création d'un environnement multi-utilisateurs.....	3
3.1. Création de l'environnement.....	3
3.2. Manipulations.....	3

1. INTRODUCTION ET RAPPELS

1.1. A propos de ce TP

Objectif : Manipuler et maîtriser les concepts de base des utilisateurs sous Linux.

Conditions : Les ordinateurs sont équipés d'un système d'exploitation Linux Red Hat, Cent Os ou Debian.

1.2. Remarque

Dans le cadre de ce TP nous allons être amenés à utiliser des commandes faisant appel à des arguments que vous choisirez vous-même. Ces arguments sont signalés entre <>. Par exemple : `# touch <nom_fichier>`. La commande que vous devrez taper ne contiendra donc pas les <> et deviendra par ex `# touch mon-premier-fichier`

1.3. Commandes utiles dans le cadre de ce TP

Dans le cadre de ce TP nous allons être amenés installer des paquets logiciels.

Pour se faire, sous **Debian**, vous devrez utiliser la commande de gestion de paquets « apt ». Cette commande s'utilise ainsi : `# apt install <paquet>`, où « <paquet> » est le nom du paquet à installer (sans les signes '<' et '>' évidemment).

Sous **Red Hat** ou **CentOs**, vous devrez utiliser la commande de gestion de paquets « yum ». Cette commande s'utilise ainsi : `# yum install <paquet>`, où « <paquet> » est le nom du paquet à installer (sans les signes '<' et '>' évidemment).

Si pour une quelconque raison, certaines commandes ne fonctionnent pas, il est impératif de le signaler à votre formateur.

2. LA GESTION DES UTILISATEURS ET DES GROUPES

2.1. Les outils de base

➔ Lors de l'installation de vim, vous êtes repassé en utilisateur root.

Vérifiez votre identité : `# whoami`

- Listez les utilisateurs grâce à la commande suivante :

```
# cat /etc/passwd | awk -F: '{print $ 1}'
```

Note : le fichier **/etc/passwd** contient tous les utilisateurs du système. Nous utilisons ici une commande qui fait appel à la notion de **|** que nous verrons dans les futurs chapitres et un filtre avancé qui permet d'afficher uniquement le premier mot de chaque ligne.

- Obtenez des informations sur l'utilisateur « tlinux » ou « usera » (ou autre) en fonction de l'affichage de la commande précédente avec la commande : `# id <nom_user>`
- Listez les groupes grâce à la commande suivante :

```
# cat /etc/group | awk -F: '{print $ 1}'
```

- Créez un nouveau groupe « mongroupe » : `# groupadd mongroupe`
- Si le groupe tlinux n'existe pas, ajoutez « tlinux » : `# groupadd tlinux`
- Définissez « mongroupe » par défaut de l'utilisateur choisi précédemment : `# usermod -g mongroupe <nom_user>`
- Vérifiez : `# id <nom_user>`
- Observez également les modifications dans le home directory de votre utilisateur avec la commande `# ls -al /home/<mon_user>`
- Ajoutez le groupe « tlinux » à ses groupes : `# usermod -a -G tlinux <nom_user>`
- Changez le nom du groupe : `# groupmod -n groupeamoi mongroupe`
- Créez l'utilisateur « monuser » dans le groupe « groupeamoi » :
- ```
useradd -g groupeamoi monuser
```
- Vérifiez si son home directory est créé.
- Observez son mot de passe dans **/etc/shadow**.
- Changez d'identité vers cet utilisateur : `# su monuser`
- Vérifiez votre identité : `$ whoami`
- Déconnectez-vous de cette identité : `$ exit`
- Définissez un mot de passe pour monuser : `# passwd monuser`
- Vérifiez le mot de passe de monuser dans **/etc/shadow**.

## 2.2. Sudo

- Installez le paquet sudo
- Éditez le fichier **/etc/sudoers** et, en remplaçant `<user_choisi>` par tlinux, usera ou autre, écrivez en bas :

```
<user_choisi> ALL=(ALL) NOPASSWD:/usr/bin/whoami
```

Attention : le fichier est en lecture seule. Avec Vi, forcez d'abord l'enregistrement avec « `:w !` » puis quittez avec « `:q` »

- Ouvrez un terminal en étant l'utilisateur que vous avez choisi, et tapez : `$ sudo whoami` puis `$ sudo bash`
- Re-éditez **/etc/sudoers** et remplacez votre ajout par :

```
<user_choisi> ALL=(ALL) NOPASSWD:ALL
```

- Ouvrez un terminal en étant l'utilisateur que vous avez choisi et tapez : `$ sudo whoami` puis `$ sudo bash`

[1] Qu'avez-vous observé ?

## 3. CRÉATION D'UN ENVIRONNEMENT MULTI-UTILISATEURS

---

### 3.1. Création de l'environnement

- Créez 4 utilisateurs en suivant les règles suivantes :
- Création de 2 groupes, groupe1 et groupe2
  - Les premier et deuxième utilisateurs (user1 et user2) sont membres du premier groupe.
  - Les troisième et quatrième utilisateurs (user3 et user4) sont membres du second groupe.
  - Le deuxième utilisateur est aussi membre du second groupe.
  - Le quatrième utilisateur est aussi membre du premier groupe.
  - Outre leur répertoire de travail, les utilisateurs ont accès à un répertoire commun `/home/groupe1` et/ou `/home/groupe2` suivant leur groupe.

[2] Quelles commandes avez-vous saisies pour créer cet environnement ?

### 3.2. Manipulations

- Activez les utilisateurs en leur créant un mot de passe avec la commande `passwd` :

```
passwd user1
passwd user2
passwd user3
passwd user4
```

- Connectez-vous sur 4 terminaux (ou consoles) différents avec les 4 utilisateurs différents.
- Pour chaque utilisateur, vérifiez à quel groupe vous appartenez avec la commande `$ groups`. Le premier groupe qui est affiché est votre groupe principal.
- Pour chaque utilisateur, vérifiez vos droits dans les répertoires home des autres utilisateurs, dans les répertoires groupe1 et groupe2 en créant, modifiant et supprimant des fichiers.

[3] Que constatez-vous ?

- Avec le second utilisateur, changez votre groupe primaire de connexion avec la commande `$ newgrp groupe2`. Tapez la commande `$ groups` pour vérifier que la modification a été prise en compte.
- Toujours avec le second utilisateur, créez un fichier dans son répertoire home et regardez les droits avec la commande `$ ls`

[4] Quel est le groupe propriétaire de ce fichier ? Est-ce que user3 a le droit de lire, modifier et supprimer ce fichier ?

- Avec le second utilisateur, rechangez votre groupe primaire de connexion avec la commande `$ newgrp groupe1`. Tapez la commande `$ groups` pour vérifier que la modification a été prise en compte.
- Avec l'utilisateur root, ajoutez un mot de passe sur le groupe2 avec la commande `# gpasswd groupe2`
- Avec le second utilisateur, rechangez votre groupe primaire de connexion avec la commande `$ newgrp groupe2`
- Avec le premier utilisateur, changez votre groupe primaire de connexion avec la commande `$ newgrp groupe1`

[5] Que constatez-vous ? Est-ce que l'user3 et l'user4 peuvent modifier le mot de passe du groupe ?

- Désignez, en root, l'utilisateur user3 en tant qu'administrateur du groupe2 avec la commande `# gpasswd -A user3 groupe2`

[6] Est-ce que l'user3 et l'user4 peuvent modifier le mot de passe du groupe ?

# TRAVAUX PRATIQUES

## Les droits sur les fichiers

---

### TABLE DES MATIÈRES

---

|                                                                             |   |
|-----------------------------------------------------------------------------|---|
| 1. Introduction et rappels.....                                             | 1 |
| 1.1. A propos de ce TP.....                                                 | 1 |
| 1.2. Remarque.....                                                          | 1 |
| 1.3. Commandes utiles dans le cadre de ce TP.....                           | 1 |
| 2. La gestion des droits des fichiers.....                                  | 2 |
| 2.1. Savoir lire les droits.....                                            | 2 |
| 2.2. Modifier les droits sur un fichier.....                                | 2 |
| 3. Umask.....                                                               | 3 |
| 4. Comprendre la propagation des droits.....                                | 3 |
| 5. Création d'un environnement multi-utilisateurs – Suite.....              | 4 |
| 5.1. Mise en place de règles de gestion des droits sur les répertoires..... | 4 |

### 1. INTRODUCTION ET RAPPELS

---

#### 1.1. A propos de ce TP

**Objectif** : Manipuler et maîtriser les concepts de base de la gestion des fichiers sous Linux.

**Conditions** : Les ordinateurs sont équipés d'un système d'exploitation Linux **Red Hat**, **Cent Os**, **Alma Linux** ou **Debian**.

#### 1.2. Remarque

Dans le cadre de ce TP nous allons être amenés à utiliser des commandes faisant appel à des arguments que vous choisirez vous-même. Ces arguments sont signalés entre <>. Par exemple : `# touch <nom_fichier>`. La commande que vous devrez taper ne contiendra donc pas les <> et deviendra par ex `# touch mon-premier-fichier`

#### 1.3. Commandes utiles dans le cadre de ce TP

Dans le cadre de ce TP nous allons être amenés à installer des paquets logiciels.

Pour se faire, sous **Debian**, vous devrez utiliser la commande de gestion de paquets « apt ». Cette commande s'utilise ainsi : `# apt install <paquet>`, où « <paquet> » est le nom du paquet à installer (sans les signes '<' et '>' évidemment).

Sous **Red Hat** ou **CentOs**, vous devrez utiliser la commande de gestion de paquets « yum ». Cette commande s'utilise ainsi : `# yum install <paquet>`, où « <paquet> » est le nom du paquet à installer (sans les signes '<' et '>' évidemment).

## 2. LA GESTION DES DROITS DES FICHIERS

---

### 2.1. Savoir lire les droits

(Source : Cyril Pain-Barre, IUT Aix-en-Provence, sous licence CC)

Supposons qu'un utilisateur tape la commande suivante `$ ls -al` et obtienne le résultat suivant :

```
drwxr---wx 3 truc bidule 12288 18 oct. 16:21 .
drwxr-xr-x 4 root root 4096 23 août 2010 ..
-r-xrw-rwx 1 truc machin 333 22 mars 08:52 fic
drw-r---wx 2 machin bidule 4096 8 sep. 10:10 rep
```

- [1] Qui est le propriétaire du répertoire de travail ?
- [2] Quelles sont les permissions de l'utilisateur truc appartenant aux groupes machin et bidule sur fic ?
- [3] Quelles sont les permissions de l'utilisateur bidule appartenant aux groupes machin et truc sur fic ?
- [4] Quelles sont les permissions de l'utilisateur bidule appartenant aux groupes machin et truc sur rep ?
- [5] Que peut-on dire des utilisateurs qui peuvent supprimer fic ?
- [6] Est-il possible que rep ait pu être créé tel quel dans le répertoire de travail ?
- [7] Quels sont les droits effectifs de l'utilisateur machin appartenant aussi au groupe machin sur rep ?

### 2.2. Modifier les droits sur un fichier

(Inspiration : Cyril Pain-Barre, IUT Aix-en-Provence, sous licence CC)

Dans votre répertoire **/home** :

➔ Utilisez le mode absolu de `chmod` pour fixer les permissions de mon-fichier à `rwX-----` et vérifiez en demandant les informations détaillées sur ce fichier.

[8] Quelles commandes avez-vous tapé ?

[9] Quelle commande auriez-vous tapé en utilisant le mode symbolique ?

➔ Créez un fichier `essai.txt` et modifiez la protection du fichier en interdisant l'écriture pour le propriétaire et en ajoutant la lecture aux autres (pas au groupe). Vérifier en demandant les informations détaillées sur ce fichier.

[10] Quelles commandes avez-vous tapé ?

➔ Éditez le fichier `essai.txt` sous `vi`. Insérer une ligne de texte et sauver sans quitter (avec `:w`). Remarquez la réaction de `vi`.

### 3. UMASK

---

**umask** est un outil de gestion des droits sur la création des fichiers et des répertoires qui permet de gérer une certaine sécurité et une restriction des droits par défaut.

Par exemple, si votre umask est 244, pour un fichier, vous partez des droits maximum **666** et vous retranchez **244**, on obtient donc **422** et par défaut les fichiers auront comme droit **422**. Pour un dossier, vous partez des droits maximum **777** et vous retranchez **244**, on obtient donc **533** et par défaut les fichiers auront comme droit **533**.

[11] A quoi correspondent les droits 422 et 533 ?

[12] Quel serez l'umask pour qu'à la création d'un fichier ou d'un répertoire, l'utilisateur obtienne les droits maximaux ?

➔ Tapez la commande `$ umask` et vérifiez que les droits des fichiers et dossiers de l'arborescence précédemment créée correspondent à la valeur de l'umask.

### 4. COMPRENDRE LA PROPAGATION DES DROITS

---

(Source : Cyril Pain-Barre, IUT Aix-en-Provence, sous licence CC)

Supposons qu'un utilisateur **toto** du groupe **users** tape les commandes suivantes et obtient les résultats suivants :

```
$ umask
0023
```

```
$ ls -al . rep1/fic8
```

```
total 12
dr-xrwxr-x 2 titi users 4096 sep 8 09:09 .
drwxr-x--x 3 titi titi 4096 sep 10 09:33 ..
drwxr-xrwx 1 titi users 4096 sep 4 17:30 rep1
--wxr-xrwx 1 truc chose 362 sep 4 17:31 fic1
-rw-r-x-w- 1 titi tata 151 sep 4 17:32 fic2
-rwxr----- 1 truc chose 512 sep 4 17:33 fic3
-r--r--r-- 1 toto titi 1024 sep 4 17:34 fic4
-rwxrw--w- 1 titi chose 421 sep 4 17:35 fic5
-rwxrw-r-x 1 titi titi 1421 sep 4 17:36 fic6
-rw-rw-rw- 1 titi chose 1421 sep 4 17:37 fic7
-rwxrw-r-x 1 toto users 8763 sep 4 17:38 rep1/fic8
```

[13] Que se passera-t'il si l'utilisateur tape les commandes suivantes :

```
$ cp fic1 fic2
```

```
$ touch fic10
```

```
$ mv fic3 fic4
```

```
$ vi fic5
```

```
$ mv -f fic7 rep1/fic8
```

```
$ cp -f fic1 fic6
```

```
$ cp -f fic1 fic9
```

[14] Compléter le tableau suivant en indiquant quels sont le propriétaire, le groupe et les permissions (complètes et en utilisant la forme symbolique avec des r, w, x et -) des fichiers de la première colonne à l'issue de ces commandes.

| Fichier | Propriétaire | Groupe | Permissions |
|---------|--------------|--------|-------------|
| fic2    |              |        |             |
| fic10   |              |        |             |
| fic4    |              |        |             |
| fic5    |              |        |             |
| fic6    |              |        |             |
| fic9    |              |        |             |

## 5. CRÉATION D'UN ENVIRONNEMENT MULTI-UTILISATEURS – SUITE

---

Cet exercice fait suite à l'exercice 3 du TP « Gestion des utilisateurs ».

Rappel : Nous avons créé un environnement de 4 utilisateurs avec 2 groupes en respectant les règles suivantes :

- Création de 2 groupes, groupe1 et groupe2
- Les premier et deuxième utilisateurs (user1 et user2) sont membres du premier groupe.
- Les troisième et quatrième utilisateurs (user3 et user4) sont membres du second groupe.
- Le deuxième utilisateur est aussi membre du second groupe.
- Le quatrième utilisateur est aussi membre du premier groupe.
- Outre leur répertoire de travail, les utilisateurs ont accès à un répertoire commun /home/groupe1 et/ou /home/groupe2 suivant leur groupe.

### 5.1. Mise en place de règles de gestion des droits sur les répertoires

➔ Modifiez les droits sur les fichiers en respectant les contraintes suivantes :

- Aucun autre utilisateur ne peut lire le contenu du home d'un autre utilisateur n'appartenant pas à son groupe.
- Les dossiers /home/groupe1 et /home/groupe2 doivent être accessibles en écriture aux membres du groupe. Les utilisateurs des autres groupes n'ont pas d'accès (ni même en lecture).

[15] Quelles commandes avez-vous saisi pour respecter ces règles ?

➔ Connectez-vous sur 4 terminaux (ou consoles) différentes avec les 4 utilisateurs différents.

➔ Pour chaque utilisateur, vérifiez vos droits dans les répertoires home des autres utilisateurs, dans les répertoires groupe1 et groupe2 en créant, modifiant et supprimant des fichiers.

[16] Quelle est la valeur conseillée pour l'umask pour échanger des fichiers seulement avec les membres des groupes dont on est membre ?

➔ Ajustez vos droits avec le sticky-bit pour que seul l'utilisateur propriétaire d'un fichier puisse supprimer un fichier dans son répertoire de groupe.

# TRAVAUX PRATIQUES

## Le shell

---

### TABLE DES MATIÈRES

---

|                                                                               |          |
|-------------------------------------------------------------------------------|----------|
| <b>1. Le shell.....</b>                                                       | <b>1</b> |
| 1.1. L'historique des commandes.....                                          | 1        |
| 1.2. Le copier/coller en mode console.....                                    | 2        |
| 1.3. Personnalisation de votre environnement.....                             | 2        |
| 1.4. Traitement d'une ligne de commande avec le backquoting.....              | 3        |
| <b>2. Les outils du shell.....</b>                                            | <b>4</b> |
| 2.1. Les redirections.....                                                    | 4        |
| 2.2. Les pipes.....                                                           | 4        |
| 2.3. Les filtres.....                                                         | 5        |
| 2.4. Les filtres et les pipes.....                                            | 5        |
| 2.5. Grep et les expressions régulières.....                                  | 6        |
| 2.6. Sed et les expressions régulières.....                                   | 6        |
| 2.7. Expression régulière de vérification de validité d'une adresse mail..... | 6        |
| <b>3. Les scripts.....</b>                                                    | <b>7</b> |
| 3.1. Les tests.....                                                           | 7        |
| 3.2. Exemple de script.....                                                   | 7        |
| <b>4. Création d'un environnement multi-utilisateurs – Suite.....</b>         | <b>8</b> |

### 1. LE SHELL

---

#### 1.1. L'historique des commandes

La commande history permet de visualiser un historique des commandes de l'utilisateur saisies dans le terminal courant.

➔ Essayez `$ history`

Vous verrez ainsi les commandes tapées avec un numéro devant. On peut donc utiliser ce numéro pour relancer des commandes sans avoir à les retaper en utilisant le caractère spécial !

[1] Quelle commande est appelée en exécutant `$ !1`

[2] Quelle commande est appelée en exécutant `$ !!`

[3] Comment appelle-t-on la dernière commande commençant par la lettre **c** ?

[4] Comment effacez-vous vos traces ?

➔ La commande history garde par défaut les 50 dernières commandes tapées, même si on a déconnecté la session entre temps.

[5] Quelle est la variable du shell à modifier pour définir le nombre de commandes gardées en mémoire ?

[6] Comment initialiser la variable du shell `$HISTTIMEFORMAT` pour que la commande history affiche la date et l'heure devant chaque commande gardée en mémoire ?



En tapant sur **les flèches du haut ou du bas**, vous avez accès aux commandes précédemment tapées durant la session.

L'utilisation de la fonctionnalité de recherche « **reverse search** » vous permet d'accéder à la fois aux toutes dernières commandes que vous venez de taper tout comme à des commandes que vous pourriez avoir tapé il y a suffisamment longtemps pour que la touche haut soit plus qu'inadaptée.

- ➔ Tapez `[Ctrl] + [R]` pour passer dans le mode « **reverse search** ».
- ➔ Tapez la lettre `[C]` : la dernière commande contenant la lettre c s'affiche.
- ➔ Retapez `[Ctrl] + [R]` : vous accédez à une autre commande contenant la lettre c, et ainsi de suite si on tape plusieurs fois `[Ctrl] + [R]`
- ➔ Pour exécuter la commande, tapez sur Entrée. Pour sortir du mode « **reverse search** », tapez sur la touche `[Esc]`

## 1.2. Le copier/coller en mode console

- ➔ Testez les deux manière d'effectuer un copier/coller sous linux en mode console :
  - ➔ `[CTRL]+[SHIFT]+[C]` pour copier, et `[CTRL]+[SHIFT]+[V]` pour coller
  - ➔ ou Sélectionnez le texte pour copier et cliquez sur la molette de la souris pour coller
  - ➔ Note que ces deux techniques utilisent deux buffers différents, donc mémorisent deux textes différents en même temps.

## 1.3. Personnalisation de votre environnement

- ➔ La variable `$PS1` permet de personnaliser votre prompt. Affichez la valeur de cette variable pour visualiser le format initial de votre prompt avec la commande `$ echo $PS1`
  - ➔ Essayez `$ PS1="[#]-\u(\h)w: "`
- [7] Quel sera le format de votre prompt ?
- [8] Personnalisez votre prompt pour afficher l'heure, le répertoire courant et #
- ➔ Visualisez les alias créés sur votre environnement.
- [9] Créez l'alias `cl` qui fait appel à la commande `clear`
- ➔ Visualisez la valeur de `$PATH` pour votre utilisateur courant ainsi que pour l'utilisateur root (`$ su`).
  - ➔ Des portions utiles pour bash sont disponibles sur le fichier `exemple_perso_bash.txt` fourni par le formateur ou ici :

```
export PS1='${debian_chroot:+($debian_chroot)}\[\033[01;31m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '

export LS_OPTIONS='--color=auto'
eval "`dircolors`"
alias ls='ls $LS_OPTIONS'
alias ll='ls $LS_OPTIONS -l'
alias l='ls $LS_OPTIONS -lA'

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

alias la='ls -alh'
```

- Mettez ces portions dans votre fichier profile (`~/.bash_profile` sur Red Hat, `~/.profile` sur Debian). Pour se faire, vous pouvez utiliser `vi`, `nano`, installer `mousepad`, une redirection `>>`, ....
- Nous souhaitons ajouter au `PATH` les répertoires `/usr/games` et `/usr/local/games`. Pour se faire, il faut concaténer (caractère `'.'`) ces deux répertoires à la variable `PATH`. On initialise alors la variable `PATH` de la manière suivante : `PATH=$PATH.:/usr/games:/usr/local/games`
- Dans votre fichier profile (`~/.bash_profile` sur Red Hat, `~/.profile` sur Debian), ajoutez l'initialisation de la variable `PATH` en ajoutant les répertoires `/usr/games` et `/usr/local/games`.
- Redémarrez votre machine et vérifiez que les modifications ont été prises en compte.

## 1.4. Traitement d'une ligne de commande avec le backquoting

(Source des questions 1.4 : Cyril Pain-Barre, IUT Aix-en-Provence, sous licence CC)

[10] Donnez la commande qui représentent l'énoncé suivant en utilisant le backquoting :

- liste des fichiers du répertoire courant dont le nom contient une étoile ?

[11] Donnez quelques combinaisons de caractères de protection qui permettent d'obtenir les détails du fichier nommé en utilisant le backquoting : un nom bizarre\*

[12] Comparez et expliquez le résultat des commandes suivantes (respectez les espaces qui sont représentés ici par le caractère `␣` :

- `$ echo␣Je␣fais␣␣␣partie␣␣␣␣␣des␣␣arguments.`
- `$ echo␣"Je␣fais␣␣␣partie␣␣␣␣␣des␣␣arguments."`
- `$ echo␣Je␣fais␣"␣␣␣part'ie␣␣d'es"␣␣argum␣"␣ents.`

[13] Comme les espaces, `bash` traite spécialement les caractères spéciaux non protégés. Notamment, il remplace les motifs par la liste des fichiers qui correspondent, remplace la séquence `$PATH` (substitution de variable) par le contenu de la variable `PATH`, et remplace la séquence `!!` (substitution de l'historique) par la commande précédente. Plusieurs protections sont possibles mais n'ont pas le même effet.

Comparez et expliquez le résultat des commandes suivantes :

- `$ echo --- * --- $PATH --- !!`
- `$ echo --- \* --- \$PATH --- \!!`
- `echo '--- * --- $PATH --- !!'`
- `echo "--- * --- $PATH --- !!"`
- `echo "--- * --- \$PATH ---" \!!`

[14] Utilisez les 3 protections dans une seule commande `echo` pour faire afficher le texte suivant :

~ C'est \* bon \* pour tout ? \" \' !!! \* ~ \$PATH " ' ??

## 2. LES OUTILS DU SHELL

### 2.1. Les redirections

- [15] Enregistrez la date dans un fichier date.txt à l'aide d'une redirection de sortie et vérifiez le contenu du fichier.
- [16] Écrasez le contenu du fichier date.txt à l'aide d'une redirection avec une chaîne de caractère de votre choix et vérifiez le contenu du fichier.
- [17] Concaténez la date au contenu du fichier date.txt à l'aide d'une redirection et vérifiez le contenu du fichier.

### 2.2. Les pipes

→ **Sur Debian** : Installez les paquets cowsay, figlet, boxes avec apt (Debian)

```
apt install cowsay figlet boxes
```

→ **Sur AlmaLinux** : Installez les paquets cowsay, figlet, boxes avec yum et en ajoutant la repository EPEL :

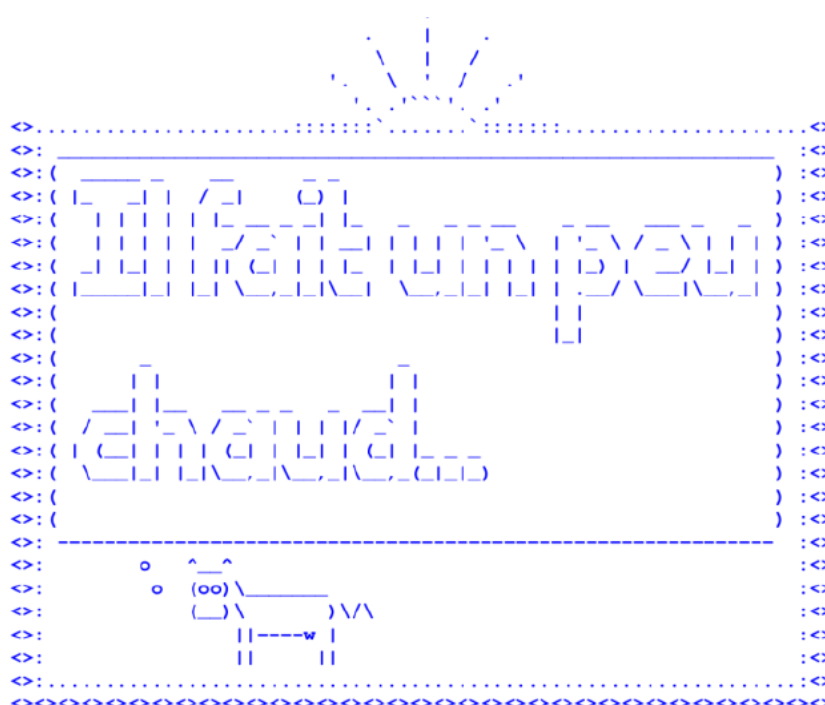
```
dnf config-manager --set-enabled crb
```

```
dnf install epel-release
```

```
yum install cowsay figlet boxes
```

- Vérifiez que cowsay est disponible pour l'utilisateur root (prise en compte de la modification du \$PATH dans le paragraphe 1.3).
- Essayez `# cowsay 'hello kids'`
- Essayez `# figlet 'hello kids'`
- Essayez `# figlet -f slant 'hello kids'`
- Essayez `# echo 'hello kids' | boxes -d boy`

[18] Reproduisez l'image suivante :



## 2.3. Les filtres

→ Créez un fichier liste\_caractere.txt et écrivez dans ce fichier les chaînes de caractères suivantes :

```
aaa
bbb
bbb
bbb
aaa
bbb
ccc
ccc
```

[19] Qu'auriez-vous saisi pour remplir le fichier liste\_caractere.txt en une seule commande à l'aide d'un echo et d'une redirection ?

A l'aide de la commande uniq :

[20] Affichez uniquement les lignes dupliquées ainsi que leur nombre d'occurrence ;

[21] Afficher les lignes uniques.

[22] Quel filtre permet de découper un fichier en plusieurs fichiers ? Quelle est la commande pour découper le fichier liste\_caractere.txt en plusieurs fichiers de 3 lignes chacun terminés par le numéro de séquence de découpage (liste\_car01, liste\_car02, liste\_car03) ?

## 2.4. Les filtres et les pipes

→ Créez un fichier matrice.txt et écrivez dans ce fichier les valeurs de l'exemple suivant d'une matrice carrée de 4\*4. Les colonnes seront séparées par des tabulations.

Exemple :

|   |   |   |   |
|---|---|---|---|
| 1 | 5 | 9 | 8 |
| 4 | 7 | 6 | 3 |
| 1 | 3 | 2 | 7 |
| 7 | 8 | 7 | 8 |

```
$ echo -e "1\t5\t9\t8\n4\t7\t6\t3\n1\t3\t2\t7\n7\t8\t7\t8" > matrice.txt
```

→ Affichez les 3 premières lignes de la matrice en tapant la commande :

```
$ head -n 3 matrice.txt
```

→ Affichez la dernière ligne de la matrice en tapant la commande :

```
$ tail -n 1 matrice.txt
```

→ Affichez la 2ème colonne de la matrice en tapant la commande :

```
$ cut -f 2 matrice.txt
```

Remarque : cut -f permet de récupérer une colonne que si les colonnes sont séparées par des tabulations.

[23] Affichez la 3<sup>ème</sup> ligne de la matrice avec un pipe et les filtres précédemment testés.

[24] Affichez l'élément 32 (ligne 3, colonne 2) de la matrice.

[25] En utilisant des redirections dans des fichiers temporaires, affichez la matrice de façon à ce que les valeurs de chaque colonne soient triées en ordre croissant.

[26] Analysez ce qu'affiche la commande suivante :

```
$ ls -R -F -1 $HOME/* | grep ':' | wc -l
```

## 2.5. Grep et les expressions régulières

➔ Récupérez le fichier « `exo_regex.txt` » fourni par le formateur ou créez un fichier `exo_regex.txt` comportant les lignes suivantes :

```
Perl
perle
perlant
parlant
Parlant
parler
perl Parler parle perl langage
alpha qwertz exe language

exe qwertz
qwertz alpha

exe qwertz
qwertz alpha

Langage
Language
Le langage c ou perl
le le le langage gage ge pppppperl ppeeerrlllll peeeerrlllll peerrlllll
le langage C ou Perl
02.03.05.03
03.236.5696
zÃ©ro un deux trois quatre cinq six sept huit neuf
0 1 2 3 5 6 7 8 9
400000 401000 402000 403000 404000 405000
400 401 402 403 404 405
4 41 42 43 44 45
4 1 2 3 4 5
ami tel age
joe 05-55-33-22-55 23 ans
bar 10-23-22-55-63 35 ans
team 03-02-21-12-25 99 ans
```

➔ Affichez ce fichier.

➔ Pour trouver les lignes vides du fichier, tapez la commande : `$ grep ^$ exo_regex.txt`

➔ Pour trouver les chaînes contenant au minimum 3 « p » : `$ grep -E "p{3,}" exo_regex.txt`

➔ Pour trouver les mots terminés par "age" : `$ grep "age\>" exo_regex.txt`

➔ Pour trouver les mots terminés par "guage" ou "gage" : `$ grep "gu\?age\>" exo_regex.txt`

A vous de jouer ! Trouvez les commandes pour trouver les motifs suivants :

[27] Trouvez les chaînes contenant "rla"

[28] Trouvez les lignes qui se terminent par « ans »

[29] Trouvez les chaînes contenant les sous-chaînes "alpha" ou "exe"

[30] Trouvez les lignes ne commençant pas par un chiffre

[31] Trouvez les lignes commençant par une voyelle ou un chiffre

[32] Trouvez les lignes ne commençant pas par une voyelle

[33] Trouvez les lignes commençant ou se terminant par un chiffre

[34] Que recherche cette commande :

```
$ grep -E "^.{0,9}$" exo_regex.txt
```

## 2.6. Sed et les expressions régulières

➔ Pour afficher le cinquième caractère de chaque ligne du fichier *exo\_regex.txt*, tapez la commande ci-dessous qui utilise l'éditeur sed :

```
$ cat exo_regex.txt | sed -e "s/^\.{4}\(. \).* /\1/"
```

Le `^\.{4}` du début sert à filtrer les 4 premiers caractères de chaque ligne. Ensuite, `\(. \)` filtre n'importe quel caractère, le fait de mettre des parenthèses enregistre le caractère en question dans une variable que l'on peut utiliser dans la chaîne de remplacement avec `"\1"`. Ensuite, `.*` filtre le reste de la ligne.

[35] Sur le même principe, trouvez la commande pour afficher les caractères 5 à 10 et le treizième.

[36] Trouvez la commande pour afficher tous les caractères à partir du quinzième.

[37] A l'aide de sed et de redirections, quelle commande pouvez-vous utiliser pour créer un fichier à partir du fichier */etc/passwd* ne contenant que les noms des utilisateurs de votre système ?

[38] A l'aide de sed, quelle commande pouvez-vous utiliser pour faire afficher la liste détaillée des fichiers d'extension *.txt* (uniquement) de votre répertoire *home*, en ne gardant que les permissions (avec le type) et le nom des fichiers, séparés par un espace. Pour simplifier, on suppose que les noms des fichiers ne contiennent pas d'espace.

## 2.7. Expression régulière de vérification de validité d'une adresse mail.

La RFC 3696 décrit le format d'une adresse mail. Vous pourrez constater que la RFC autorise l'utilisation de caractères tels que `! ? ...` Mais usuellement, la plupart des vérifications du formatage des adresses emails se basent sur les règles suivantes :

- Le pseudo formant la première partie du mail est formé d'au minimum de 2 caractères et au maximum de 64 : lettres minuscules (pas de majuscules), des chiffres, des points, des tirets et des underscores « `_` ».
- Le pseudo est ensuite suivi d'un `@` et ne doit pas commencer par un caractère spécial.
- Le nom de domaine est formé d'au minimum de 2 caractères et au maximum de 63 : lettres minuscules (pas de majuscules), des chiffres, des tirets et des underscores « `_` ».
- Le nom de domaine et l'extension sont séparés par un « `.` ».
- L'extension comprend entre 2 et 4 lettres minuscules.
- Deux caractères spéciaux ne peuvent se suivre (ex : `email.@toto.fr` et `email._nom@toto.fr` sont invalides).

[39] Trouvez et testez l'expression régulière qui permettra de vérifier le maximum de règles. Pour tester, vous utiliserez un grep sur un fichier d'email (avec des adresses bien formées et d'autres mal formées).

---

## 3. LES SCRIPTS

### 3.1. Les tests

[40] Testez les commandes suivantes (attention aux espaces) et commentez le résultat :

```
$ [12 == 13] ; echo $?
```

```
$ [12 -eq 13] ; echo $?
```

```
$ [12 -eq 12] ; echo $?
```

[41] Quel est le test pour vérifier l'existence d'un fichier ?

[42] Expliquez le test suivant :

```
$ test $[RANDOM%2] -eq 1 && echo "Pile" || echo "Face"
```

### 3.2. Exemple de script

➔ Récupérez le script fourni par le formateur « exemple\_script.txt » ou éditez un fichier exemple\_script.txt pour écrire les données suivantes dedans :

```
#!/bin/bash

BEGIN INIT INFO
Provides: nom_de_mon_script
Required-Start:
Required-Stop:
Default-Start: 2 3 4 5
Default-Stop: 0 1 6
Short-Description: Description de mon script
Description:
END INIT INFO

case "$1" in
 start)
 echo "Le script démarre ici"
 ;;
 stop)
 echo "Le script s'arrête ici"
 ;;
 restart)
 $0 stop
 sleep 2s
 $0 start
 ;;
 status)
 echo "Statut de mon script"
 ;;
 *)
 echo "Usage: {start|stop|restart|status}"
 exit 1
 ;;
esac
exit 0
```

➔ Renommez-le et rendez-le exécutable

➔ Affichez-le et essayez-le

➔ Personnalisez l'entête LSB.

[43] Dans la partie « start », ajoutez une option pour lancer le test de la question 42. Testez cette nouvelle option.

[44] Dans la partie « start », ajoutez une deuxième option pour créer 20 fichiers numérotés de 0 à 20 (ex : mon\_fic01, mon\_fic02 ...) en utilisant une boucle for. Testez votre option.

[45] Enfin, ajoutez une troisième option pour réaliser un dialogue interactif avec votre script. Et testez votre option.

➔ Vous pouvez trouver d'autre inspiration de personnalisation sur <http://www.commandlinefu.com>

## 4. CRÉATION D'UN ENVIRONNEMENT MULTI-UTILISATEURS – SUITE

---

Cet exercice fait suite à l'exercice 3 du TP « Gestion des utilisateurs ».

Rappel : Nous avons créé un environnement de 4 utilisateurs avec 2 groupes en respectant les règles suivantes :

- Création de 2 groupes, groupe1 et groupe2
  - Les premier et deuxième utilisateurs (user1 et user2) sont membres du premier groupe.
  - Les troisième et quatrième utilisateurs (user3 et user4) sont membres du second groupe.
  - Le deuxième utilisateur est aussi membre du second groupe.
  - Le quatrième utilisateur est aussi membre du premier groupe.
  - Outre leur répertoire de travail, les utilisateurs ont accès à un répertoire commun /home/groupe1 et/ou /home/groupe2 suivant leur groupe.
- ➔ Modifiez les profils des utilisateurs pour qu'au login, le répertoire d'accueil sur lequel il souhaite travailler lui soit demandé (s'il entre U, il sera sous son répertoire, s'il entre G, il sera sous le répertoire de son groupe).



# TRAVAUX PRATIQUES

## La gestion des paquets

---

### TABLE DES MATIÈRES

---

|                                                            |   |
|------------------------------------------------------------|---|
| 1. Introduction.....                                       | 1 |
| 1.1. A propos de ce TP.....                                | 1 |
| 2. La gestion des paquets sous Debian.....                 | 1 |
| 2.1. APT et les licences de paquets.....                   | 1 |
| 2.2. Dpkg.....                                             | 2 |
| 2.3. APT-GET.....                                          | 2 |
| 2.4. apt.....                                              | 2 |
| 2.5. Aptitude.....                                         | 3 |
| 2.6. Installation par les sources.....                     | 3 |
| 3. La gestion des paquets sous AlmaLinux (ou Red Hat)..... | 4 |

### 1. INTRODUCTION

---

#### 1.1. A propos de ce TP

**Objectif :** Manipuler les outils d'installation des paquets sous **Debian** et **Red Hat / AlmaLinux**

**Conditions :** Les ordinateurs sont équipés d'un système d'exploitation **Debian** ou **Red Hat / AlmaLinux** ou d'une machine virtuelle sous l'un des deux systèmes d'exploitation.

**Remarque :** n'hésitez pas à tester les outils inutiles (donc indispensables) comme `cmatrix` et `sl`.

### 2. LA GESTION DES PAQUETS SOUS DEBIAN

---

#### 2.1. APT et les licences de paquets

APT est un projet relativement vaste, qui prévoyait à l'origine une interface graphique. Il repose sur une bibliothèque contenant le cœur de l'application et **apt-get** est la première interface — en ligne de commande — développée dans le cadre du projet. **apt** est une deuxième interface en ligne de commande fournie par APT qui corrige quelques erreurs de conception de `apt-get`. De nombreuses interfaces graphiques sont ensuite apparues en tant que projets extérieurs : `synaptic` (interface graphique), **aptitude** (qui inclut à la fois une interface en mode texte et une interface graphique, bien que pas encore complète), `wajig`, etc.

La configuration des dépôts des paquets Debian se trouve dans le fichier **/etc/apt/sources.list**

➔ Éditez ce fichier, et ajoutez les licences « contrib » et « non-free » à la liste des licences acceptées. Vous avez à votre disposition, en standard, les éditeurs `vi` et `nano`.

(Astuce avec `vi` : vous pouvez chercher « main » et remplacer par « main contrib non-free » grâce à la commande « `:%s/main/main contrib non-free/g` »)

- Profitez de l'édition de ce fichier pour repérer les sources CD-Rom. Pour éviter trop de téléchargements sur internet, laissez cette ligne.
- Mettez à jour l'index des paquets APT avec la commande : `# apt update`

## 2.2. Dpkg

Dpkg est un outil de manipulation de fichiers paquet Debian. Pour illustrer son utilisation nous nous apprêtons à installer le paquet locate, contenant la commande de même nom servant à indexer et retrouver des fichiers par leur nom. La première étape consiste alors à récupérer le fichier paquet Debian, reconnaissable à son extension « .deb ».

- Récupérez le paquet «locate» en HTTP avec wget sur le serveur web du formateur (Outils Linux) ou à l'adresse [ftp.fr.debian.org/debian/pool/main/f/findutils/locate\\_4.9.0-4\\_amd64.deb](http://ftp.fr.debian.org/debian/pool/main/f/findutils/locate_4.9.0-4_amd64.deb) :  
`# wget <url>`
- Obtenez des informations sur le paquet : `# dpkg -I locate_4.9.0-4_amd64.deb`
- Listez les fichiers contenus dans le paquet : `# dpkg -c locate_4.9.0-4_amd64.deb`
- Extrayez le contenu : `# mkdir tmp ; dpkg -x locate_4.9.0-4_amd64.deb tmp/`
- Installez le paquet avec dpkg : `# dpkg -i locate_4.9.0-4_amd64.deb`
- Listez les paquets installés : `# dpkg -l`

## 2.3. APT-GET

### 2.3.a) Utilisation de apt-get

Apt-get sert à manipuler les paquets situés dans le dépôt Debian. Pour illustrer son utilisation nous nous apprêtons à installer un navigateur internet en mode console : lynx.

- Mettez à jour le cache de apt-get : `# apt-get update`
- Installez lynx avec apt-get : `# apt-get install lynx`

### 2.3.b) Utilisation de apt-rdepends

- Installez `# apt-get install apt-rdepends`
- Lister les dépendances pour lynx : `# apt-rdepends lynx`
- Lister les paquets dépendant de lynx : `# apt-rdepends -r lynx`

(on comprend mieux l'intérêt des outils comme apt, aptitude et autres ^^ ).

### 2.3.c) Utilisation de apt-cache

- Recherchez tous les paquets contenant le mot apache `# apt-cache search apache`

## 2.4. apt

### 2.4.a) Utilisation de apt

Depuis Debian 7, le frontal le plus recommandé est apt .

- Mettez à jour le cache de apt : `# apt update`
- Installez cmatrix avec apt : `# apt install cmatrix`

### 2.4.b) Utilisation de l'option list

- Listez les paquets installés : `# apt list --installed`
- Listez les paquets pouvant être mis à jour : `# apt list --upgradable`

### 2.4.c) Utilisation de l'option search

→ Recherchez tous les paquets contenant le mot apache : `# apt search apache`

Plus lisible que apt-cache non ?

Vous remarquerez aussi que l'outil apt permet l'auto-complétion.

## 2.5. Aptitude

→ Installez l'outil aptitude s'il n'est pas disponible

→ Mettez à jour le cache Aptitude : `# aptitude update`

→ Cherchez les paquets liés à sl : `# aptitude search sl`

→ Installez sl : `# aptitude install sl`

## 2.6. Installation par les sources

→ Installez l'outil de décompression bzip2 : `# apt install bzip2`

→ Récupérez la dernière archive nmap depuis le serveur du formateur ou depuis Internet (<https://nmap.org/dist/nmap-7.94.tar.bz2>) avec lynx :

```
lynx <url>
```

→ Décompressez l'archive bzip (sans supprimer l'archive d'origine):

```
bzip2 -d -k -v nmap-7.94.tar.bz2
```

→ Décompressez l'archive tar extraite du bzip2 : `# tar xvf nmap-7.92.tar`

→ Installez les outils standard de compilation : `# apt install build-essential`

→ Allez dans le répertoire décompressé et générez le fichier de compilation : `# ./configure`

Une erreur vous signale qu'il manque des bibliothèques pour compiler correctement nmap.

→ Installez la bibliothèque libpcap-dev et retapez la commande `# ./configure`

→ Compilez : `# make`

→ Installez : `# make install`

→ Testez : `# nmap -v -n -T4 -A <ip_serveur_formateur>`

### 3. LA GESTION DES PAQUETS SOUS ALMALINUX (OU RED HAT)

---

[1] À l'aide des pages de manuel de rpm et yum, déterminez comment effectuer les actions suivantes et testez-les :

- (a) Rechercher le paquet nmap par mots-clés (scanner, security)
- (b) Afficher des informations sur le paquet nmap
- (c) Lister les paquets installés sur le système (avec RPM)
- (d) Installer le paquet nmap et ses dépendances (avec YUM)
- (e) Lister les fichiers du paquet nmap précédemment installé (avec RPM)
- (f) Installer manuellement un fichier RPM (ne pas tester)
- (g) Supprimer le paquet nmap installé

Yum se configure dans /etc/yum.conf, et ses repositories sont configurés dans /etc/yum.repos.d/.

→ Regardez si le repo centosplus est activé avec la commande `# yum repolist all`

Ce dépôt contient des applications qui changent quelques composants bases de AlmaLinux. Il ne sera donc plus complètement compatible avec le contenu du 'Fournisseur en amont du système d'exploitation servant de référence' (comprendre RedHat) bien que l'équipe de développement AlmaLinux ait fait des tests avec toutes les applications de ce dépôt.

Ce dépôt peuvent être appelé ponctuellement pour une recherche ou une installation avec une option de Yum : `# yum search --enablerepo=centosplus <nom_paquet>`

Supposons que vous sachiez que la nouvelle version de coreutils (qui contient les outils de base comme ls, mkdir, etc.), vers laquelle une mise-à-jour serait faite, est buggée.

→ Affichez le fichier /etc/redhat-release et notez la version de votre AlmaLinux (ou Red Hat)

→ Excluez coreutils des mises-à-jour en rajoutant une ligne dans /etc/yum.conf.  
`exclude=coreutils`

→ Effectuez un update de votre système `# yum update`

→ Affichez le fichier /etc/redhat-release et notez que la version n'a pas été modifiée.

→ Supprimez la ligne exclude=coreutils de votre fichier yum.conf.

→ Effectuez de nouveau un update de votre système `# yum update`

S'il y a des mises à jour disponibles, alors votre version de AlmaLinux a été modifiée.

# TRAVAUX PRATIQUES

## Les services

---

### TABLE DES MATIÈRES

---

|                                                  |          |
|--------------------------------------------------|----------|
| <b>1. Les processus – Commandes de base.....</b> | <b>1</b> |
| 1.1. La surveillance des processus.....          | 1        |
| 1.2. La commande systemctl.....                  | 2        |
| 1.3. Les jobs.....                               | 2        |
| 1.4. Gérer les jobs.....                         | 2        |
| <b>2. Les processus en arrière plan.....</b>     | <b>3</b> |
| 2.1. Nohup et les binaires GNU.....              | 3        |
| 2.2. Nohup et vos binaires.....                  | 3        |

---

### 1. LES PROCESSUS – COMMANDES DE BASE

---

#### 1.1. La surveillance des processus

➔ Testez la commande **ps**

[1] Qu'affiche cette commande ? Qu'affichent les options **ax** de la commande **ps** ?

[2] Quel est le PID du démon init (utilisez un **grep**) ? Quelle commande avez-vous tapé ? Pourquoi la commande que vous avez tapé retourne deux processus ? Comment faire pour n'afficher que le pid du processus init ?

Pour certains traitements automatisés, il peut être intéressant d'afficher le résultat de la commande **ps** de manière personnalisée.

[3] Avec la commande **sed**, comment supprimer les tabulations pour les remplacer par des espaces lorsqu'on veut afficher le PID d'init ?

➔ Utilisez **top** avec un rafraîchissement toutes les secondes des processus classés par ordre décroissant d'utilisation de la mémoire.

[4] Comment avez-vous fait ?

➔ Utilisez la commande **pstree** et retrouvez votre démon init. (**pstree** est dans le paquet **psmisc**, pour l'installer : **# apt install pstree**)

[5] Que peut-on dire de ce démon ?

Installez si nécessaire et essayez les outils :

➔ **iostat** qui est fournit par le paquet **sysstat**

➔ **nmon**

➔ **ac**, **lastcomm**, **sa** qui sont fournis par le paquet **acct**

#### 1.2. La commande systemctl

Attention : cette commande ne fonctionne que si vous êtes sur un Linux avec **systemd**.

➔ Pour lister les services actifs, tapez la commande **# systemctl list-units --type=service**

→ Pour vérifier si le service réseau est actif (networking), tapez la commande :

```
systemctl is-active networking
```

→ Pour lister les dépendances du service réseau, tapez la commande :

```
systemctl list-dependencies networking
```

### 1.3. Les jobs

→ Tapez la commande `# sleep 100`

[6] Que fait cette commande ? Comment tuer ce processus ?

→ Retapez la commande précédente en ajoutant un `&` à la fin.

[7] Quelle est la différence avec la commande précédente ?

→ Utilisez la commande **jobs** pour trouver le PID du processus `sleep` ainsi que son numéro de job.

### 1.4. Gérer les jobs

Si vous avez lancé une tâche et oublié de la lancer en arrière plan (oubli du `&`), il est possible de mettre une commande en arrière plan à posteriori, sans avoir besoin d'arrêter la tâche (signal `QUIT`).

Il faut, tout d'abord, envoyer le signal **STOP** via la séquence de touches **CTRL + Z**. Cela aura pour effet de stopper la commande en cours et de rendre la main pour exécuter une autre commande. Il est alors possible de la passer en arrière plan avec **bg** (background). La commande `sleep` vous permettra d'appréhender ce mécanisme.

→ Testez ce mécanisme en tapant les commandes suivantes :

```
sleep 100
^Z
[1]+ Stoppé sleep 100
bg
[1]+ sleep 100 &
jobs -l
[1]+ <num_pid> En cours d'exécution sleep 100 &
```

Il est possible de revenir à l'exécution de notre première commande à tout moment avec **fg** (foreground).

```
fg
```

A noter que ce qui concerne l'entrée standard, si une commande attend sur celle-ci, `bash` suspendra la commande. Il faudra alors remettre la commande en avant plan (`fg`), ce qui l'autorisera à lire sur l'entrée standard :

```
read message &
fg
```

## 2. LES PROCESSUS EN ARRIÈRE PLAN

---

### 2.1. Nohup et les binaires GNU

L'option `&`, bien qu'assez couramment utilisée, a ce défaut non négligeable : le processus reste attaché à la console, ce qui veut dire que si la console est fermée ou que l'utilisateur se déconnecte, le processus sera automatiquement arrêté.

Si on veut que le processus continue, il faut lancer la commande via **nohup**.

La plupart des programmes GNU (et le shell lui même) ont cependant intégré la gestion du signal HUP dans leur code. En effet, en lançant un processus puis en fermant la console dans laquelle il a été exécuté, le processus reste actif tant qu'il n'est pas fini et devient le fils direct de init.

Pour visualiser ce mécanisme, réalisez les étapes suivantes :

- ➔ Dans une console, lancez un ping vers 8.8.8.8 redirigé dans un fichier de manière détachée (avec un &) : `$ ping 8.8.8.8 > test &`
- ➔ Récupérez le PID de l'exécution du ping. Ouvrez une nouvelle console (Sur la machine virtuelle, [Ctrl-Droit]+[F2]) et repérez ce PID dans l'arborescence de pstree (option -p pour afficher les PIDs, option -H pour surligner le PID recherché).
- ➔ Revenez sur la première console ([Ctrl-Droit]+[F1]) et fermez ( `$ exit` ). Dans la deuxième console, repérez ce PID dans l'arborescence de pstree.
- ➔ Arrêtez le ping.

[8] Qu'avez-vous observé ?

- ➔ Vous pouvez tester ces manipulations sur des terminaux (émulateurs de console) et étudier le comportement de ce mécanisme.

## 2.2. Nohup et vos binaires

Pour visualiser le mécanisme de la commande nohup, il faut écrire un programme en C qui ne gère pas le signal HUP.

- ➔ En utilisateur root, installez au préalable les paquets gcc et build-essential.
- ➔ Écrivez un programme nommé test.c exécutant une pause infinie.

```
// Modèle de boucle infinie avec pause :
include <stdlib.h>
include <unistd.h>
void main() {
 while(1) { sleep(100); }
}
```

- ➔ Puis compilez ce programme en utilisant gcc : `# gcc test.c -o test`
- ➔ Exécutez le binaire et récupérez le PID de son exécution dans un autre terminal. Repérez ce PID dans l'arborescence de pstree (option -p pour afficher les PIDs, option -H pour surligner le PID recherché).
- ➔ Fermez le terminal où a été exécuté le script.

[9] Que constatez-vous ? Est-ce que votre binaire s'exécute encore ?

- ➔ Exécutez le binaire avec **nohup** et récupérez le PID de son exécution. Ouvrez un nouveau terminal et repérez ce PID dans l'arborescence de pstree (option -p pour afficher les PIDs, option -H pour surligner le PID recherché).
- ➔ Fermez le terminal où a été exécuté le script. Re-lancez la commande pstree.

[10] Que constatez-vous ?

# TRAVAUX PRATIQUES

## La chaîne de démarrage

### TABLE DES MATIÈRES

|                                     |   |
|-------------------------------------|---|
| 1. Script Initd.....                | 1 |
| 2. Avec Systemd (By Snorky).....    | 2 |
| 3. Le grub.....                     | 3 |
| 4. Autres options au démarrage..... | 3 |

### 1. SCRIPT INITD

- ➔ Récupérez le fichier exemple\_script\_initd.txt fourni par le formateur ou éditez un nouveau fichier exemple\_script\_initd.txt avec dedans les instructions suivantes :

```
#!/bin/bash

BEGIN INIT INFO
Provides: nom_de_mon_script
Required-Start:
Required-Stop:
Default-Start: 2 3 4 5
Default-Stop: 0 1 6
Short-Description: Description de mon script
Description:
END INIT INFO

case "$1" in
 start)
 ..
 echo "Le script démarre ici"
 ;;
 stop)
 ..
 echo "Le script s'arrête ici"
 ;;
 restart)
 $0 stop
 sleep 2s
 $0 start
 ;;
 status)
 ..
 echo "Statut de mon script"
 ;;
 *)
 ..
 echo "Usage: {start|stop|restart|status}"
 exit 1
 ;;
esac
exit 0
```

- ➔ Renommez-le en /etc/init.d/script\_initd
- ➔ Rendez-le exécutable et personnalisez-le en écrivant dans la partie start la date et l'heure dans un fichier de la racine du système.
- ➔ N'oubliez pas de modifier le nom du fichier dans l'entête LSB (Provides : script\_initd)
- ➔ Ajoutez-le dans la chaîne de démarrage du runlevel de la machine (2,3 ou 5 en fonction du système que vous utilisez).

[1] Quelles sont les deux méthodes que vous pouvez utiliser ?

- ➔ Redémarrez la machine et vérifiez que cela a fonctionné.



## 2. AVEC SYSTEMD (BY SNORKY)

→ Sur Debian, installez le paquet deb mis à disposition par Snorky de la manière suivante :

```
apt get install git
git clone https://gogs.sincrone.fr/Snorky/Rick.git
cd Rick/
dpkg -i rick.deb
```

→ Une fois le .deb installé :

- Vérifiez que /bin/rick est exécutable
- Éditez le fichier rick présent dans /etc/default pour l'étudier
- Éditez les fichiers présents dans /etc/rick et étudiez-les.
- Enfin, étudiez le fichier /bin/rick

### [2] Que fait ce script ?

→ Créez un fichier rick.service et éditez-le

→ En vous aidant du tableau suivant, créez les configuration du service rick. Le fichier /etc/default/rick sera le fichier d'environnement du service rick. Toutes les conditions ne sont pas forcément nécessaires.

| SECTION | NAME                                                     | DESCRIPTION                                                      |
|---------|----------------------------------------------------------|------------------------------------------------------------------|
| Unit    | permet de décrire le service, de vérifier des prérequis. |                                                                  |
|         | Description                                              | Description du service                                           |
|         | After                                                    | Doit se lancer après un processus donné                          |
|         | ConditionPathExists                                      | Sera utilisé si le chemin existe                                 |
| Service | définit le type de service, et ce qu'il va faire         |                                                                  |
|         | Type                                                     | Le type de service                                               |
|         | ExecStart                                                | Commande de service lorsque start est invoqué                    |
|         | ExecStopPost                                             | Commande à exécuter après l'arrêt du service                     |
|         | Restart                                                  | Permet de redémarrer le service                                  |
|         | User                                                     | Permet de lancer le processus sous le nom d'un utilisateur donné |
|         | EnvironmentFile                                          | Fichier contenant les configuration global de l'application.     |
|         | Group                                                    | Permet de lancer le processus sous le nom d'un groupe donné      |
| Install | circonstances et les déclencheurs                        |                                                                  |
|         | WantedBy                                                 | Dans quel cadre le service s'exécute                             |

→ Une fois le fichier service créé, copiez-le dans le répertoire /etc/systemd/system/rick.service

→ Démarrez le service et faites un status sur le service rick.

[3] Avez-vous le résultat escompté de la lecture du script ?

[4] Comment faire pour que le fichier pid ne soit pas créé au démarrage du service ?

→ Ajoutez le service rick au boot de la machine

[5] Comment avez-vous fait ?

→ Enfin, redémarrez votre machine et vérifiez que le service rick est bien up and running.

### 3. LE GRUB

---

→ Éditez votre grub.cfg (dans /boot) ou votre menu.lst (en fonction de la distribution).

→ Personnalisez le nom de votre menuentry, nom qui s'affichera au démarrage lors de l'écran du grub.

[6] Quelle est le temps d'affichage de cet écran ?

→ Redémarrez la machine et vérifiez que les modifications effectuées ont été prises en compte.

[7] Comment modifier lors du démarrage les options du grub ?

**Pour aller plus loin :** Trouvez sur Internet comment modifier l'image de fond de votre menu grub et faire en sorte que le grub demande un mot de passe lors de l'édition manuelle des options du grub lors du démarrage.

### 4. AUTRES OPTIONS AU DÉMARRAGE

---

→ Dans les fichiers /etc/profile et /etc/login.defs, modifiez, en utilisateur root, la valeur du \$PATH en ajoutant les chemins /usr/games et /usr/local/games pour l'utilisateur root (id = 0).

→ Redémarrez votre machine et vérifiez les valeurs du \$PATH et que cowsay est accessible à l'utilisateur root.

→ Faites en sorte qu'à chaque lancement de console, l'image de la vache précédemment créée s'affiche à l'écran (petit indice : recherchez l'emplacement du fichier « motd.dynamic » avec la commande find).

→ Créez au démarrage vos alias préférés pour l'utilisateur root et l'utilisateur tlinux.

→ Personnalisez le prompt de l'utilisateur tlinux.

→ Redémarrez votre machine et vérifiez que les modifications ont été prises en compte. Vérifiez que cowsay est accessible à l'utilisateur root.

# TRAVAUX PRATIQUES

## Le réseau

---

### TABLE DES MATIÈRES

---

|                                                                  |          |
|------------------------------------------------------------------|----------|
| <b>1. Configuration réseau.....</b>                              | <b>1</b> |
| 1.1. A propos de ce TP.....                                      | 1        |
| 1.2. Remarques sur l'utilitaire IP.....                          | 1        |
| 1.3. Paramétrage de la carte réseau de la machine virtuelle..... | 2        |
| 1.4. Paramétrage du réseau.....                                  | 3        |
| <b>2. Quelques outils réseau.....</b>                            | <b>5</b> |
| 2.1. ping.....                                                   | 5        |
| 2.2. arp.....                                                    | 5        |
| 2.3. traceroute.....                                             | 6        |
| 2.4. Résolution de nom de domaine.....                           | 6        |
| 2.5. Option -s de la commande IP.....                            | 6        |
| 2.6. La commande netstat et ss.....                              | 6        |
| <b>3. Pour aller plus loin : Le superserveur XINETD.....</b>     | <b>7</b> |
| 3.1. Le superserveur xinetd.....                                 | 7        |
| 3.2. Personnalisation de telnet.....                             | 7        |
| 3.3. Pour aller plus loin : Création d'un service.....           | 7        |
| <b>4. Encore plus loin : Juste pour le fun.....</b>              | <b>8</b> |
| 4.1. Xhost.....                                                  | 8        |

### 1. CONFIGURATION RÉSEAU

---

#### 1.1. A propos de ce TP

**Conditions** : Votre Debian est installé sur une machine virtuelle. A ce stade, la carte réseau de votre machine virtuelle est configurée en mode NAT. L'IP attribuée à votre machine virtuelle est une IP en DHCP n'appartenant pas au réseau général. Vous ne pouvez pas atteindre la machine virtuelle de votre voisin.

#### 1.2. Remarques sur l'utilitaire IP

Depuis 2015, l'outil `iproute2` (dont la commande `IP`) a remplacé l'outil `net-tools` (commandes `ifconfig`, `route`, `netstat` ...).

Cependant, il est intéressant de connaître l'utilisation des différentes commandes réseau, que cela soit via l'outil `iproute2` ou l'outil `net-tools`.

➔ Pour appréhender les différences entre les différents outils, installez l'outil `net-tools` :

```
apt install net-tools
```

### 1.3. Paramétrage de la carte réseau de la machine virtuelle

→ Récupérez le nom de votre interface réseau principale avec la commande :

```
ip link show
```

(équivalent `# ifconfig -a`). On appellera **iface** la variable de ce nom d'interface réseau (eth0, ens33, enp0s3 etc).

→ Afin de désactiver la configuration en DHCP de votre interface réseau, éditez le fichier `/etc/network/interfaces`.

→ Remplacez la ligne **iface <iface> inet dhcp** par **iface <iface> inet static**. Sauvegardez le fichier.

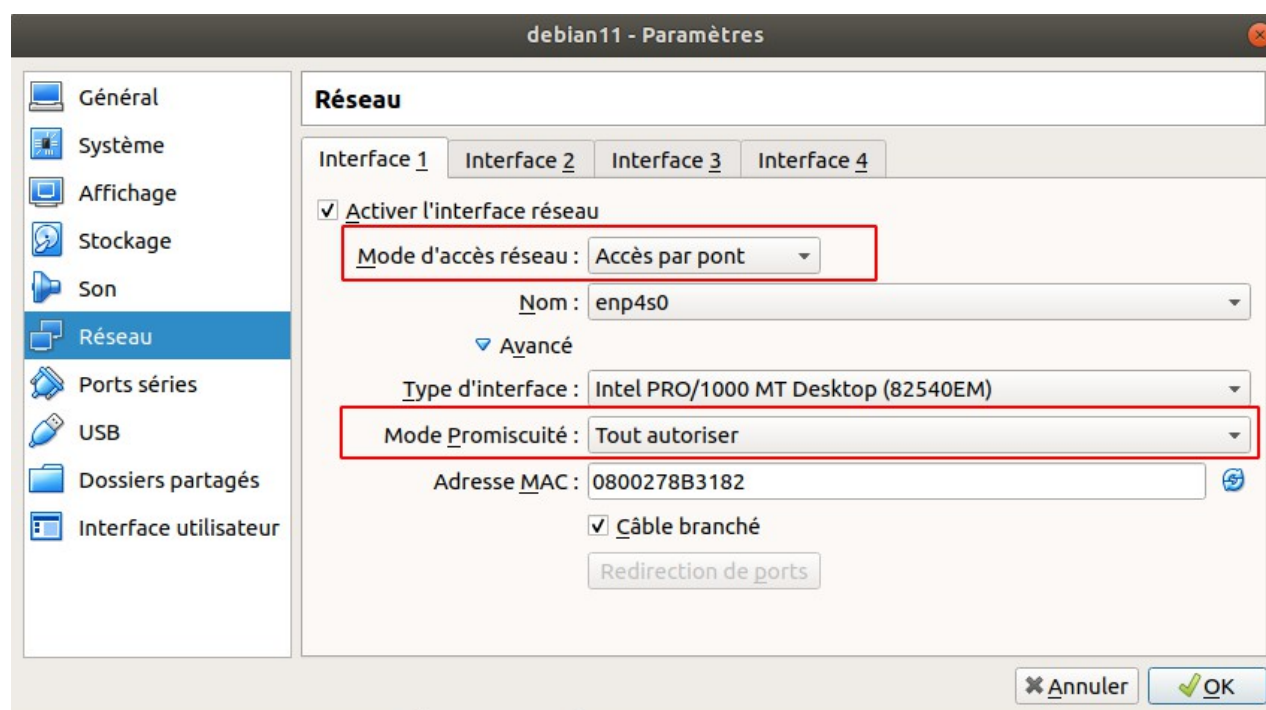
→ Arrêtez votre machine virtuelle (ne pas enregistrer l'état de votre machine). Nous allons reparamétrer la carte réseau sur la machine virtuelle connectée en mode « Accès par pont ».

L'accès "**accès par pont**", qui correspond au mode "Bridged" de VMWare, permet à la machine virtuelle de communiquer vers l'extérieur via la machine hôte. La carte réseau de la machine virtuelle accède au LAN via la carte réseau physique. En créant deux cartes réseaux virtuelles en mode « accès par pont », la carte réseau physique a ainsi un rôle de switch, via laquelle sont connectés trois IP : deux IP pour la machine virtuelle, une IP pour la machine physique.

#### Sur Virtual Box :

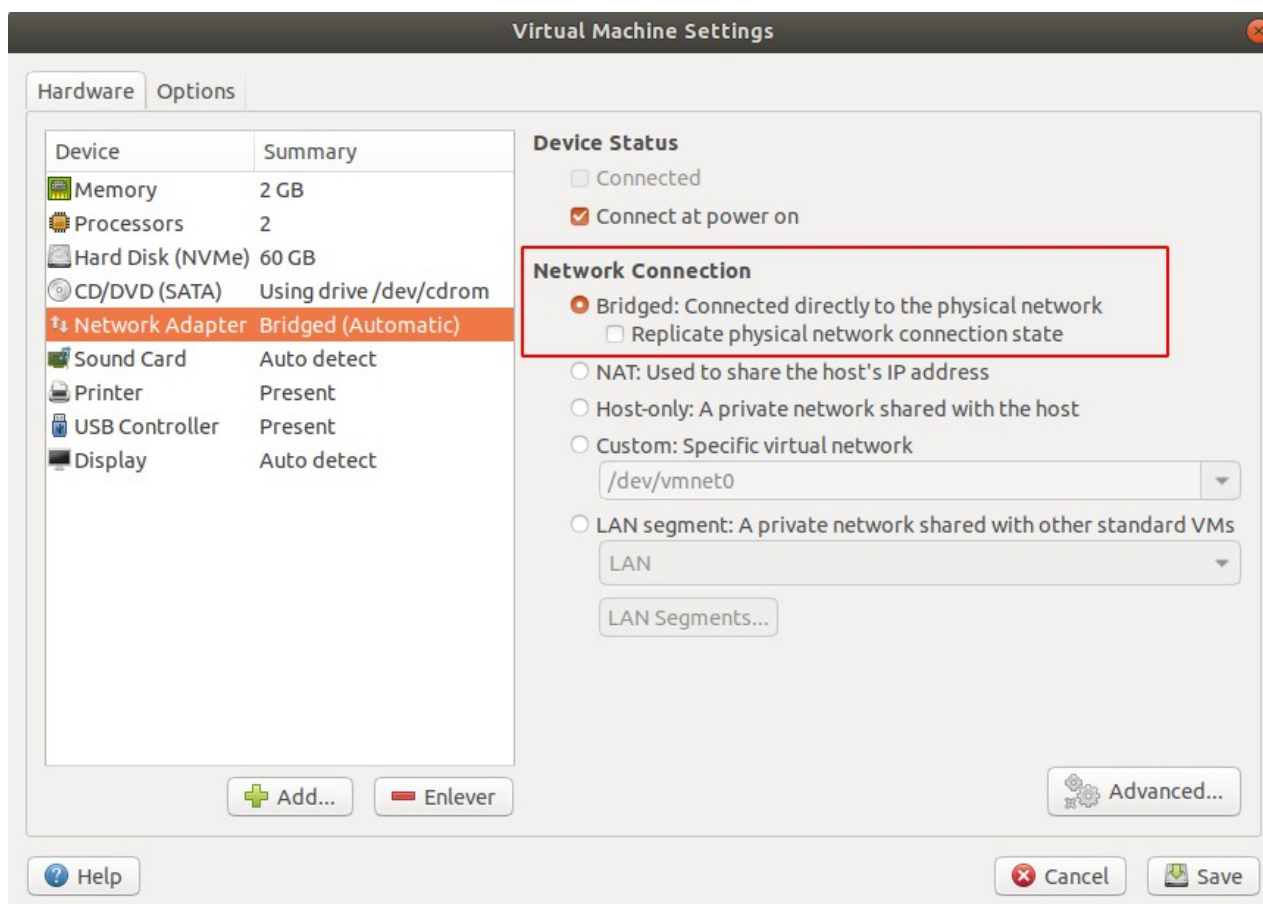
→ Sélectionnez votre machine virtuelle, puis cliquez sur le bouton « Paramètres ».

→ Cliquez sur le sous-menu **Réseau**. Sur l'onglet « Interface 1 », cochez « Activer l'interface réseau » puis choisissez les options comme ci-dessous :



**Sur VMWARE:**

- ➔ Sélectionnez votre machine virtuelle, puis cliquez sur le bouton « Paramètres ».
- ➔ Cliquez sur l'onglet Hardware. Sur la carte réseau (Network Adapter), cochez « Activer la carte » puis choisissez les options comme ci-dessous :



Une fois la modification faite sur la configuration de votre VM :

- ➔ Redémarrez votre machine virtuelle ou le daemon réseau. Vérifiez vos paramètres réseau avec les commandes suivantes :

```
ifconfig -a
```

```
ip -4 address show
```

Vous noterez qu'il n'y a plus d'IP attribuée à la carte réseau *iface*.

Remarque : la commande IP propose des raccourcis et options très intéressantes. L'option `-4` affiche que les adresses ipv4.

- ➔ Testez la commande : `# ip -4 a s`

(et oui, le Linuxien est un flemmard du clavier...)

## 1.4. Paramétrage du réseau

L'interface réseau principale et filaire de votre machine virtuelle aura une IP appartenant au réseau de la salle, donc une IP dans la même plage que celle actuellement attribuée à la carte physique de votre ordinateur.

- ➔ Afin d'éviter les conflits réseaux, ajoutez 100 à l'IP de votre ordinateur. Par exemple, si l'IP de votre ordinateur est 192.168.0.21, l'IP WAN de votre machine virtuelle sera 192.168.0.121. On appellera *ip\_lan* la variable définissant l'IP de votre VM et définie dans ce paragraphe.

La passerelle (***ip\_passerelle\_wan***) et le masque (***masque\_lan***) seront les mêmes que ceux attribués à votre machine physique.

➔ Indiquez à votre machine virtuelle les serveurs DNS en remplissant le fichier ***/etc/resolv.conf*** de la façon suivante :

```
echo "nameserver 1.1.1.1" > /etc/resolv.conf
echo "nameserver 208.67.220.220" >> /etc/resolv.conf
```

➔ Faites des recherches sur Internet sur les deux IP indiquées précédemment comme DNS.

Pour vérifier votre fichier DNS de configuration :

```
more /etc/resolv.conf
```

La commande ***ip*** va nous permettre d'attribuer l'IP aux interfaces réseaux de la machine virtuelle.

➔ Tapez la commande suivante avec les IPs notées dans votre plan d'adressage :

```
ip a add <ip_lan>/<masque_lan> dev <iface>
```

Remarque : avec la commande IP, le masque peut être écrit en mode CIDR ou non : /24 ou /255.255.255.0 par exemple.

L'équivalent avec net-tools est :

```
ifconfig <iface> <ip_lan> netmask <masque_lan>
```

Remarque : avec ifconfig et l'option netmask, le masque doit être écrit en entier (255.255.255.0). Vous pouvez aussi spécifier le masque directement derrière l'IP : ***# ifconfig <iface> <ip\_lan>/<masque\_lan\_CIDR>***

**Attention** : si vous vous souhaitez modifier votre IP, contrairement à la commande ifconfig, la commande « ip a add » ajoute une IP à l'interface (cela crée un alias). Pour modifier votre IP avec la commande IP, il faudra d'abord supprimer l'IP qui n'est pas correcte avant d'ajouter l'IP correcte : ***# ip a del <ip\_lan\_erronée> netmask <masque\_lan\_erronée>***

➔ Vérifiez vos paramètres réseau en retapant la commande ip link show (mode raccourci) :

```
ip l sh
```

**Important** : Si vous avez utilisé la commande IP, vous constaterez que l'interface *iface* est notée DOWN. Alors que si vous avez utilisé la commande ifconfig, elle est notée UP.

➔ Assurez-vous que votre interface est UP avec la commande précédente ; si ce n'est pas le cas, passez l'interface en UP avec la commande :

```
ip link set <iface> up
```

➔ Vérifiez votre table de routage avec la commande :

```
ip route show
(raccourci : # ip r s)
```

L'équivalent avec net-tools est :

```
route -n
```

Vous remarquerez que la route vers votre sous-réseau a été ajoutée automatiquement.

➔ Vérifiez que vous pinguez votre passerelle et que vous ne pinguez pas l'internet (ce qui est jusque là normal car vous n'avez pas spécifié de passerelle) :

```
ping <ip_passerelle_lan> -c4
ping 8.8.8.8 -c4
```

Remarque : l'option `-c4` permet de spécifier que vous envoyez seulement 4 paquets. En effet, contrairement à Windows, sans cette option `-c`, la commande `ping` de Linux continuera d'envoyer des packages ICMP jusqu'à ce qu'il reçoive un signal d'interruption (Ctrl+C).

→ Ajoutez la passerelle par défaut de l'interface WAN :

```
ip route add default via <ip_passerelle_wan>
```

L'équivalent avec `net-tools` est :

```
route add default gw <ip_passerelle_wan> dev <iface>
```

→ Vérifiez vos routes :

```
ip r s
```

→ Enfin, avec un `ping`, vérifiez que vous accédez à Internet.

→ Modifiez le fichier `/etc/network/interfaces` pour que ces paramètres soient fixés au démarrage de votre machine. Testez en redémarrant le service `networking` :

```
service networking restart
```

Remarque : si vous utilisez une autre distribution que Debian, pensez à modifier les bons fichiers des paramètres réseaux.

→ Vérifiez que vous accédez à l'interface Webmin de votre voisin (si Webmin a été précédemment installé) en allant sur l'URL **`https://<ip_Linux_voisin>:10000`**

Source intéressante : <https://memo-linux.com/ip-la-commande-linux-pour-gerer-son-interface-reseau/>

## 2. QUELQUES OUTILS RÉSEAU

---

### 2.1. ping

**Ping** permet de tester l'accessibilité d'une autre machine à travers un réseau IP. Ping utilise une requête ICMP *Request* et attend une réponse *Reply*. L'envoi est répété pour des fins statistiques : déterminer le taux de paquets perdus et le délai moyen de réponse. Si d'autres messages ICMP sont reçus de la part de routeurs intermédiaires (comme *TTL exceeded*, *Fragmentation needed*, *administratively prohibited...*), ils sont affichés à l'écran.

Dans la réponse d'un ping, le paramètre TTL (Time to Live) permet donc de déterminer combien de routeurs ont été traversés pour effectuer le retour. Si on a un TTL de 53, cela signifie que l'on a probablement traversé 11 routeurs (sur un Linux, le TTL initial est de 64 et en moyenne un paquet traverse environ 10-15 routeurs pour la plupart des destinations, sur un Windows, le TTL initial est de 128).

→ Envoyez 4 requêtes ICMP sur l'IP 8.8.8.8 avec la commande `# ping -c4 8.8.8.8`

→ En déduire avec le TTL le nombre de routeurs traversés.

### 2.2. arp

Dans les principaux systèmes d'exploitation (Linux, Windows, MacOS), la commande permettant de manipuler la table ARP s'appelle : `arp`

→ Lisez le man de la commande `arp` et affichez votre table ARP.

[1] ★ Voyez-vous des entrées ARP correspondant à l'IP 8.8.8.8 ? Pourquoi ?

La commande `# ip neighbour show` permet également d'afficher la table ARP et d'ajouter des entrées.

La commande `# ip monitor` permet d'afficher en temps réel les nouvelles entrées ARP. Pour avoir une nouvelle entrée ARP, vous pouvez depuis votre machine hôte faire un ping sur la VM.

## 2.3. traceroute

Traceroute est une commande permettant d'identifier les différentes machines traversées lorsque l'on cherche à atteindre une destination donnée. Elle identifie un itinéraire. Sous Windows cette commande s'appelle « tracert ».

- ➔ Utilisez `tracert` pour identifier l'itinéraire emprunté pour joindre 8.8.8.8 :  
`# tracert 8.8.8.8`
- ➔ Explicitiez l'utilisation de la méthode ICMP avec l'option `-I` et lisez le man de la commande `tracert` pour comprendre l'intérêt du choix de la méthode utilisée.

## 2.4. Résolution de nom de domaine

- ➔ Installez le paquet `dnsutils`.
- ➔ Testez les commandes `host`, `nslookup` et `dig` sur [cyberwings.fr](http://cyberwings.fr) et notez les différences dans les résultats affichés.

## 2.5. Option -s de la commande IP

Si vous avez testé la commande `ifconfig -a`, vous avez dû remarquer que le résultat affichait des statistiques sur le nombre de paquets envoyés/reçus (depuis le dernier reboot de la VM), ainsi que les éventuelles erreurs de trafic. Cela peut être très pratiques pour de la supervision réseau ou lors d'incidents réseau.

L'option `-s` de la commande `IP` permet d'obtenir les statistiques.

- ➔ Testez la commande suivante :

```
ip -s a s
```

## 2.6. La commande netstat et ss

`Netstat` permet d'afficher les connexions réseau (entrantes et sortantes), les tables de routage et un certain nombre de statistiques d'interface réseau.

Il est disponible sous Linux, de type Unix et Windows les systèmes d'exploitation. `Netstat` est puissant et peut être un outil pratique pour résoudre les problèmes liés au réseau et vérifier les statistiques de connexion.

Pour lister les services et ports en écoute sur votre machine, la commande `netstat` Linux peut prendre de nombreux paramètres. Mais voici une bonne base :

```
netstat -paunt
```

- `-a` : Tous les ports
- `-t` : Tous les ports TCP
- `-u` : Tous les ports UDP
- `-l` : Tous les ports en écoute
- `-n` : Affiche directement les IP. Pas de résolution de nom.
- `-p` : Affiche le nom du programme et le PID associé.

`Netstat` est considérée comme obsolète : la commande `ss` qui la remplace obtient ses informations directement à partir de l'espace noyau (donc déjà plus rapide que l'agrégation de données depuis des fichiers). Les options utilisées avec ses commandes sont très semblables à celles de *netstat*, la remplaçant donc avantageusement.

- ➔ Testez `netstat` et `ss` avec les mêmes options.



### 3. POUR ALLER PLUS LOIN : LE SUPERSERVEUR XINETD

**Objectif de l'exercice :** Manipulation des fichiers de configuration de xinetd pour :

- personnaliser le service existant telnet pour qu'un script soit exécuté lors d'une connexion telnet ;
- créer votre propre service réseau.

#### 3.1. Le superserveur xinetd

xinetd sert à piloter l'accès à un ou plusieurs services réseaux. Il reçoit des requêtes de clients, extérieurs pour la plupart, qui demandent un accès à un service réseau déterminé (ex : ftp, telnet, ssh...). Le super démon va, en fonction des instructions qu'on lui aura données (fichiers de configuration) transmettre ou rejeter la connexion.

xinetd permet d'affiner les paramétrages d'accès aux services :

- possibilité d'affiner les logs des services gérés
- paramétrage d'accès par service et non global
- paramétrage des plages horaires de disponibilité des services
- possibilité de chrooter les services (ex : ftp)
- possibilité de limiter les attaques de type deny of service (contrôle de la priorité d'un serveur, contrôle de la charge CPU, contrôle du nombre de connexions par service, ...)
- redirection de ports

➔ Installez le paquet xinetd.

➔ Créez un fichier script en bash qui affiche « bonjour » (ne pas oublier le sheebang, les droits .... testez le script !).

#### 3.2. Personnalisation de telnet

➔ Dans le dossier `/etc/xinetd.d/`, créez un fichier nommé **telnet** qui définira le service *telnet* en écoute sur le port 23 (telnet) et qui exécutera votre script. Remplacez le paramètre `<path_script>` par la localisation de votre script.

```
service telnet
{
 disable = no
 port = 23
 socket_type = stream
 protocol = tcp
 wait = no
 user = root
 server = <PATH_script>/votre_script
}
```

➔ Redémarrez le service xinetd.

➔ Vérifiez que votre service est bien en écoute : `# netstat -antupx | grep 23`

➔ Connectez-vous à ce service avec netcat (installez ce binaire si nécessaire) :

```
nc 127.0.0.1 23
```

➔ A partir d'un autre poste, connectez-vous à ce service avec netcat : `# nc <ip> 23`

#### 3.3. Pour aller plus loin : Création d'un service

Le fichier `/etc/services` listent les services pris en charge par xinetd.

➔ Dans le fichier `/etc/services`, ajoutez « votreservice » en TCP sur le port 730.

- Créez un nouveau fichier « votreservice » dans le dossier xinetd.d en écoute sur le port 730, redémarrez le service xinetd et testez.

## 4. ENCORE PLUS LOIN : JUSTE POUR LE FUN...

---

### 4.1. Xhost

Pour l'utiliser, il est nécessaire d'activer explicitement l'activation des communications TCP/IP pour le serveur X. Ce qui n'est pas le cas par défaut dans la plupart des environnements graphiques modernes :

- Pour KDE: vérifiez que /etc/sddm.conf existe, sinon exécuter `# sddm --example-config > /etc/sddm.conf` puis chercher la ligne "ServerArguments" et retirez les caractères "no" de "-nolisten tcp".
- Pour GDM (GNOME): dans /etc/gdm/gdm.conf, mettre "DisallowTCP=false" sous "[security]". Vérifiez néanmoins que lightdm n'est pas installé, le cas échéant ; suivre l'étape pour XFCE4.
- Pour XFCE4: dans /etc/lightdm/lightdm.conf, mettre "xserver-allow-tcp" à true

- Affichez les autorisations pour le serveur X:0 : `# DISPLAY=:0 xhost`

- Regardez la valeur de la variable DISPLAY : `# echo $DISPLAY`

- Affichez les autorisations pour ce serveur en fonction de la variable récupérée (X:0.0 ou X:1) :  
`# DISPLAY=:1 xhost`  
ou

`# DISPLAY=:0.0 xhost`

- Essayez de lancer xeyes sur l'adresse IP locale : `# DISPLAY=<votre_ip>:1 xeyes`

ou `# DISPLAY=<votre_ip>:0.0 xeyes`

- Autorisez votre adresse IP locale à se connecter au serveur X:1 ou X:0.0 :

`# DISPLAY=:1 xhost +<votre_ip>`

- Affichez les autorisations pour le serveur X:1 ou X:0.0 : `# DISPLAY=:1 xhost`

- Essayez de lancer xeyes sur l'adresse IP locale : `# DISPLAY=<votre_ip>:1 xeyes`

- Autorisez toutes les adresses IP à se connecter au serveur X:1 : `# DISPLAY=:1 xhost +`

- Affichez les autorisations pour le serveur X:1 ou X:0.0 : `# DISPLAY=:1 xhost`

- Installez xterm

- Affichez un message sur le serveur X:1 ou X:0.0 de votre voisin :

`# DISPLAY=<ip_du_voisin>:1 xeyes`

`# DISPLAY=<ip_du_voisin>:1 xterm -e 'cowsay "coucou c moi" && read'`

- Affichez chez votre voisin une instance de firefox-esr affichant une image située sur un site web pris au hasard. (astuce : ne lancez pas l'instance de firefox en root)

# TRAVAUX PRATIQUES

## OpenSSH

---

### TABLE DES MATIÈRES

---

|                                                                              |          |
|------------------------------------------------------------------------------|----------|
| <b>1. Introduction.....</b>                                                  | <b>1</b> |
| 1.1. A propos de ce TP.....                                                  | 1        |
| 1.2. Commandes Linux utiles dans le cadre de ce TP.....                      | 2        |
| 1.3. Pré-requis sur votre serveur LINUX.....                                 | 2        |
| <b>2. SSH.....</b>                                                           | <b>2</b> |
| 2.1. Client SSH sous Linux.....                                              | 2        |
| 2.2. Serveur OpenSSH sous Linux.....                                         | 2        |
| 2.3. Client SSH Sous Windows.....                                            | 2        |
| 2.4. Serveur OpenSSH sous Windows.....                                       | 3        |
| 2.5. Serveur SSH sur un équipement Cisco.....                                | 5        |
| 2.6. Envoi de documents.....                                                 | 6        |
| 2.7. Paramétrage d'OpenSSH Serveur.....                                      | 6        |
| 2.8. Étude du flux SSH.....                                                  | 6        |
| <b>3. Pour aller plus loin : redirection SSH simple.....</b>                 | <b>7</b> |
| 3.1. Objectif du Tunnel.....                                                 | 7        |
| 3.2. Pré-requis : Webmin sur le poste Linux.....                             | 7        |
| 3.3. Utilisation de Bitwise Tunnelier pour établir le tunnel.....            | 7        |
| 3.4. Utilisation d'OpensSSH en ligne de commande pour établir le tunnel..... | 9        |
| 3.5. Étude du flux SSH.....                                                  | 9        |

### 1. INTRODUCTION

---

#### 1.1. A propos de ce TP

**Objectif :** Mettre en œuvre de SSH.

**Conditions :**

Nous vous proposons de réaliser la série de manipulation soit à partir d'un système d'exploitation Windows11 soit à partir d'un système Linux.

**Déroulement :** Ce TP consiste en une série de manipulations et de questions. Les réponses sont facilement accessibles en réalisant la(les) bonne(s) manipulation(s), et, au besoin, en recherchant de l'information complémentaire sur Internet. En cas de doute ou de difficulté : nous consulter.

#### 1.2. Commandes Linux utiles dans le cadre de ce TP

Dans le cadre de ce TP nous allons être amenés installer des paquets logiciels.

Pour se faire, sous **Debian**, vous devrez utiliser la commande de gestion de paquets « apt ». Cette commande s'utilise ainsi : `# apt install <paquet>`, où « <paquet> » est le nom du paquet à installer (sans les signes '<' et '>' évidemment).

Sous Red Hat ou **CentOs**, vous devrez utiliser la commande de gestion de paquets « yum ». Cette commande s'utilise ainsi : `# yum install <paquet>`, où « <paquet> » est le nom du paquet à installer (sans les signes '<' et '>' évidemment).

### 1.3. Pré-requis sur votre serveur LINUX

➔ Afin de pouvoir utiliser la commande netstat, installez le paquet net-tools.

## 2. SSH

---

### 2.1. Client SSH sous Linux

Le client OpenSSH utilisable via la commande **ssh** est installé par défaut sur les systèmes Linux ou MAC OS.

### 2.2. Serveur OpenSSH sous Linux

➔ Installez le paquet openssh-server sur votre machine Linux (serveur d'accès distant SSH).

➔ Pour lancez le binaire : `# service sshd restart`

➔ À partir d'un autre poste, connectez-vous sur votre serveur SSH :

```
ssh -p 22 root@<IP_Machine_Distante_Linux>
```

Vous êtes maintenant connecté à un terminal SSH de votre machine virtuelle. Le logiciel vous demande votre mot de passe root.

Si votre machine distante est une machine virtuelle, cela peut vous permettre de réaliser plus simplement des copier/coller par rapport à l'utilisation de l'interface directe de la machine virtuelle.

**Remarque** : Selon la distribution utilisée, le compte root n'a pas accès à SSH par défaut. Il faut alors modifier le fichier de configuration. Si tel est le cas :

➔ Éditez le fichier `/etc/ssh/sshd_config`.

➔ Modifiez la valeur de la variable `PermitRootLogin` en mettant `yes`.

➔ Modifiez la valeur de la variable `PasswordAuthentication` en mettant `yes`.

### 2.3. Client SSH Sous Windows

Dans les nouvelles versions de Windows10, le client OpenSSH est activé par défaut et accessible par le PowerShell.

Si ce n'est pas le cas, vous pouvez l'installer soit depuis le powerShell soit depuis l'interface graphique de Windows.

#### 2.3.a) Activation du client SSH depuis PowerShell :

➔ Lancez d'abord PowerShell en tant qu'administrateur et tapez la commande suivante :

```
Get-WindowsCapability -Online | ? Name -like 'OpenSSH.Client*'
```

La commande doit renvoyer la sortie suivante :

Name : OpenSSH.Client~~~~0.0.1.0

State : NotPresent (ou **Installed** dans le cas où il est installé)

➔ Pour installer la fonctionnalité OpenSSH Client, tapez la commande suivante :

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

### 2.3.b) Activation du client SSH depuis l'environnement graphique :

- ➔ Rendez-vous dans les « Paramètres Windows » via menu Démarrer ou le raccourci clavier Win+I, puis cliquez sur « Applications ».
- ➔ Cliquez sur « Applications et fonctionnalités »
- ➔ Cliquez ensuite sur « Fonctionnalités facultatives », puis sur le bouton « Ajouter une fonctionnalité ».
- ➔ Cherchez et localisez « Client OpenSSH » sur la liste et enfin, cliquez sur « Installer » pour l'activer sur votre machine.

### 2.3.c) Test de la connexion :

- ➔ Une fois votre client SSH windows activé, testez la connexion vers votre serveur Linux.

## 2.4. Serveur OpenSSH sous Windows

### 2.4.a) Installation :

Depuis PowerShell :

- ➔ Lancez d'abord PowerShell en tant qu'administrateur et tapez la commande suivante :

```
Get-WindowsCapability -Online | ? Name -like 'OpenSSH.Server*'
```

- ➔ Pour installer la fonctionnalité OpenSSH serveur :

```
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0
```

Depuis l'environnement graphique :

- ➔ Suivre les mêmes étapes que pour le Client OpenSSH jusqu'à la fenêtre « Ajouter une fonctionnalité ».
- ➔ Cherchez et localisez « serveur OpenSSH » sur la liste et enfin, cliquez sur « Installer » pour l'activer sur votre machine.

**Note :**

- Les fichiers binaires sont installés sur `%WINDIR%\System32\OpenSSH`
- `%WINDIR%` est généralement `C:\Windows`

### 2.4.b) Activation des services :

OpenSSH Serveur installe deux services qui ne sont pas encore en cours d'exécution après l'installation et dont le type de démarrage est manuel et désactivé.

- ➔ Pour vérifier l'état des services tapez la commande :

```
Get-Service -Name *ssh* |Select DisplayName, Status, StartType, Name
```

```
Get-Service -Name *ssh* |Select * (pour afficher toutes les propriétés)
```

- ➔ Pour une utilisation régulière, vous pouvez activer ces services automatiquement au démarrage avec les commandes :

```
Set-Service sshd -StartupType Automatic
```

```
Set-Service ssh-agent -StartupType Automatic
```

➔ Pour démarrer les deux services manuellement :

```
Start-Service sshd
```

```
Start-Service ssh-agent
```

➔ Vérifiez l'état après activation :

```
Get-Service -Name *ssh*
```

```
PS C:\Windows\system32> Get-Service -Name *ssh* |Select DisplayName, Status, StartType, Name

DisplayName Status StartType Name

OpenSSH Authentication Agent Running Automatic ssh-agent
OpenSSH SSH Server Running Automatic sshd

PS C:\Windows\system32>
```

### 2.4.c) Pare-feu

➔ Vérifiez que la règle de pare-feu pour les connexions SSH entrantes a été correctement activée avec la commande suivante :

```
Get-NetFirewallRule -Name *SSH*
```

```
PS C:\Windows\system32> Get-NetFirewallRule -Name *SSH*

Name : OpenSSH-Server-In-TCP
DisplayName : OpenSSH SSH Server (sshd)
Description : Inbound rule for OpenSSH SSH Server (sshd)
DisplayGroup : OpenSSH Server
Group : OpenSSH Server
Enabled : True
Profile : Any
Platform : {}
Direction : Inbound
Action : Allow
EdgeTraversalPolicy : Block
LooseSourceMapping : False
LocalOnlyMapping : False
Owner :
PrimaryStatus : OK
Status : La règle a été analysée à partir de la banque. (65536)
EnforcementStatus : NotApplicable
PolicyStoreSource : PersistentStore
PolicyStoreSourceType : Local

PS C:\Windows\system32>
```

➔ Si ce n'est pas le cas, ajoutez une règle à partir de l'interface graphique ou bien en tapant la commande suivante sous PowerShell :

```
New-NetFirewallRule -Protocol TCP -LocalPort 22 -Direction Inbound -Action Allow
-DisplayName SSH
```

### 2.4.d) Connexion à votre serveur

A partir d'une machine distante, vous pouvez maintenant vous connecter à votre serveur SSH windows en utilisant votre utilisateur courant comme login.

➔ Pour voir plus d'options tapez :

```
>ssh -h
```

## 2.5. Serveur SSH sur un équipement Cisco

### 2.5.a) Activation du SSH sur un switch

Nous allons configurer SSH sur un Switch. Les commandes sont quasiment identiques sur un routeur, à l'exception de l'attribution d'une IP qui ne se paramètre pas de la même façon.

→ Sur l'interface Console (mode « CLI »), en mode « privilège », paramétrez une IP sur votre switch :

```
vlan 1
interface vlan 1
ip address « ip » « mask »
no shutdown
```

→ Entrez ensuite un nom pour l'équipement :

```
hostname <nom_equipement>
```

→ Puis entrez un domaine :

```
ip domain-name <domain.com>
```

→ Créez un utilisateur et un mot de passe pour la connexion SSH :

```
username <nom_utilisateur> password <mot_de_passe>
```

→ Générez les clés asymétriques RSA d'une taille de 4096 bits

```
crypto key generate rsa general-keys modulus 4096
```

→ Activez le protocole SSHv2, la version 1 de SSH étant obsolète et vulnérable à des attaques :

```
ip ssh version 2
```

→ Sélectionnez les lignes d'accès à distance 0 à 4

```
line vty 0 4
```

→ Forcez la connexion uniquement par SSH :

```
transport input ssh
```

→ Activez l'authentification par utilisateur précédemment créé :

```
login local
```

→ Contrôlez votre configuration avec la commande suivante :

```
show run
```

### 2.5.b) Accès à la console du switch via SSH

## 2.6. Envoi de documents

- ➔ Si le client est sur Windows, installez WinSCP ou Filezilla Client et transférez un fichier que vous aurez préalablement créé sur le poste distant.
- ➔ Si vous êtes sur Linux, tapez la commande suivante pour envoyer un fichier que vous aurez préalablement créé sur le poste distant :

```
scp mon-fichier <votre_user>@<IP_Machine_Distante>:/<dossier_user>
```

- ➔ Vérifiez que le poste distant a bien reçu votre fichier dans le dossier root. Si votre Linux possède un environnement graphique, à partir d'un explorateur de fichier, tapez dans la barre d'adresse l'URL suivante : `sftp://<votre_user>@<IP_Machine_Distante>`

Vous pouvez naviguer dans l'arborescence de la machine distante.

## 2.7. Paramétrage d'OpenSSH Serveur

Dans la vraie vie, l'administrateur d'un serveur Linux ne laissera pas l'accès SSH sur le port 22 et surtout, l'utilisateur root ne pourra pas se connecter directement en SSH.

### 2.7.a) Sous Linux

- ➔ Si l'utilisateur tlinux n'existe pas, ajoutez le groupe et l'utilisateur tlinux : `# adduser tlinux`

Le fichier de configuration de SSH est `/etc/ssh/sshd_config`.

- ➔ Retrouvez dans le fichier de configuration les lignes qui permettent :

- de spécifier le port de connexion ;
- d'interdire la connexion à l'utilisateur root ;
- autoriser l'accès ssh qu'à certains groupes et utilisateurs (l'utilisateur tlinux par exemple).

- ➔ Modifiez ces lignes, relancez SSH et testez.

### 2.7.b) Sous Windows

- ➔ Le fichier de configuration de SSH est `%programdata%\ssh\sshd_config`

- ➔ Retrouvez dans le fichier de configuration les lignes qui permettent :

- de spécifier le port de connexion ;
- autoriser l'accès ssh qu'à certains groupes et utilisateurs.

PermitRootLogin n'est pas applicable sous Windows. Pour empêcher les administrateurs de se connecter via SSH, il faut ajouter une directive « DenyGroups » :

```
DenyGroups Administrators
```

- ➔ Modifiez ces lignes, relancez SSH `Restart-Service sshd` et testez.

## 2.8. Étude du flux SSH

Note : Sur Linux, OpenSSH supporte uniquement SSHv2 (SSHv1 ayant été proscrit).

- ➔ Sur le poste à partir duquel vous établissez la connexion SSH distante, lancez Wireshark ou la commande `tcpdump` et analysez votre connexion SSH. Vérifiez que tous les flux sont chiffrés, y compris la saisie des mots de passe.

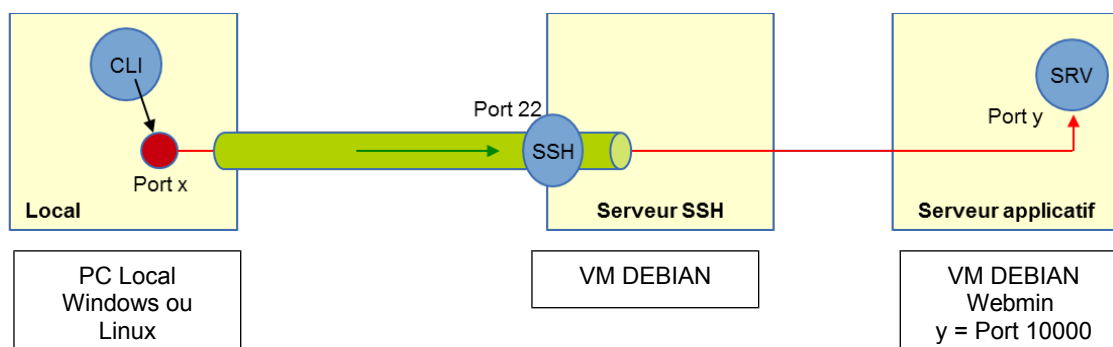


### 3. POUR ALLER PLUS LOIN : REDIRECTION SSH SIMPLE

#### 3.1. Objectif du Tunnel

Le but de cet exercice est d'accéder, à partir de votre PC windows (via Bitvise Tunnelier), à l'interface Webmin de votre machine Debian au travers d'un tunnel SSH.

Dans notre cas, le serveur SSH et le serveur applicatif se trouve donc sur le même poste (la machine virtuelle Debian).



#### 3.2. Pré-requis : Webmin sur le poste Linux

→ Sur votre poste Linux, avec la commande `wget`, téléchargez la dernière version de webmin (prendre le fichier **.deb**) sur le serveur web du formateur (dans OUTILS/LINUX) ou à cette adresse : <https://sourceforge.net/projects/webadmin/files/webmin/>

→ Avec `aptitude` installez les paquets `apt-show-versions`, `libapt-pkg-perl`, `libauthen-pam-perl`, `libio-pty-perl`, `libnet-ssleay-perl` :

```
apt install libnet-ssleay-perl apt-show-versions libapt-pkg-perl libauthen-pam-perl libio-pty-perl
```

→ Installez le paquet webmin avec `dpkg` ( `# dpkg -i webmin_XXXX_all.deb` )

→ S'il y a des erreurs lors de l'installation des paquets, utilisez les commandes

```
apt --fix-missing update
```

```
apt install -f
```

Webmin est alors en écoute sur le port 10000 en SSL/TLS.

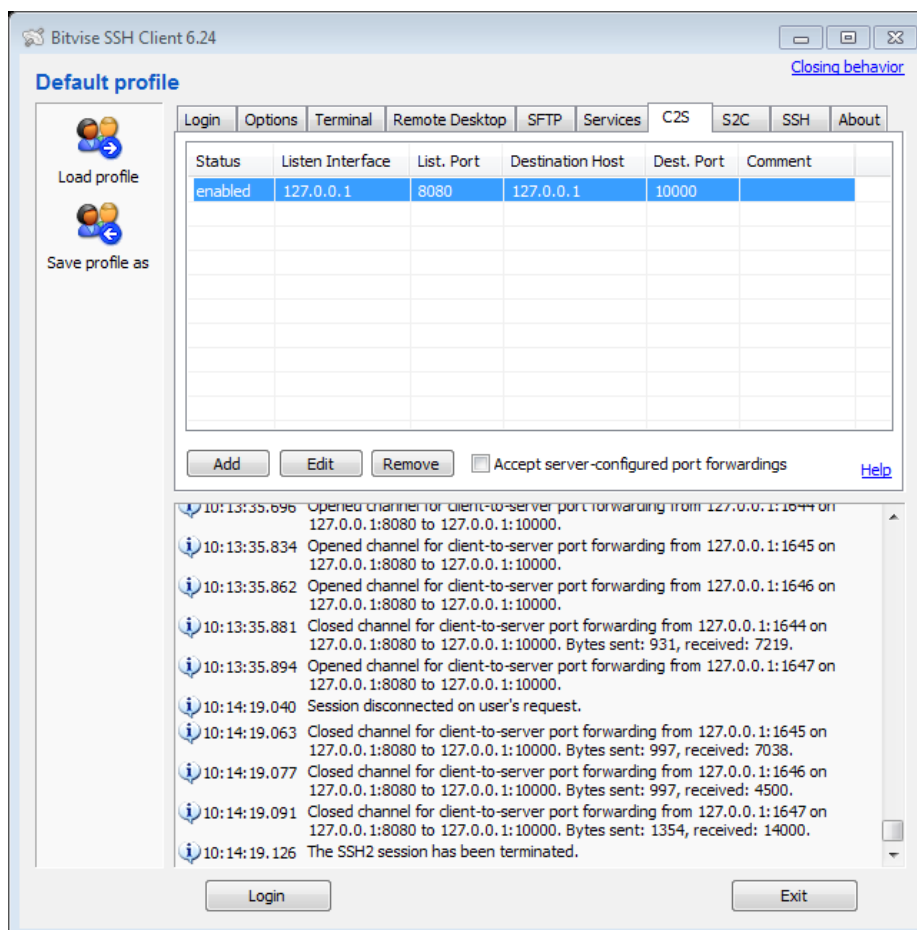
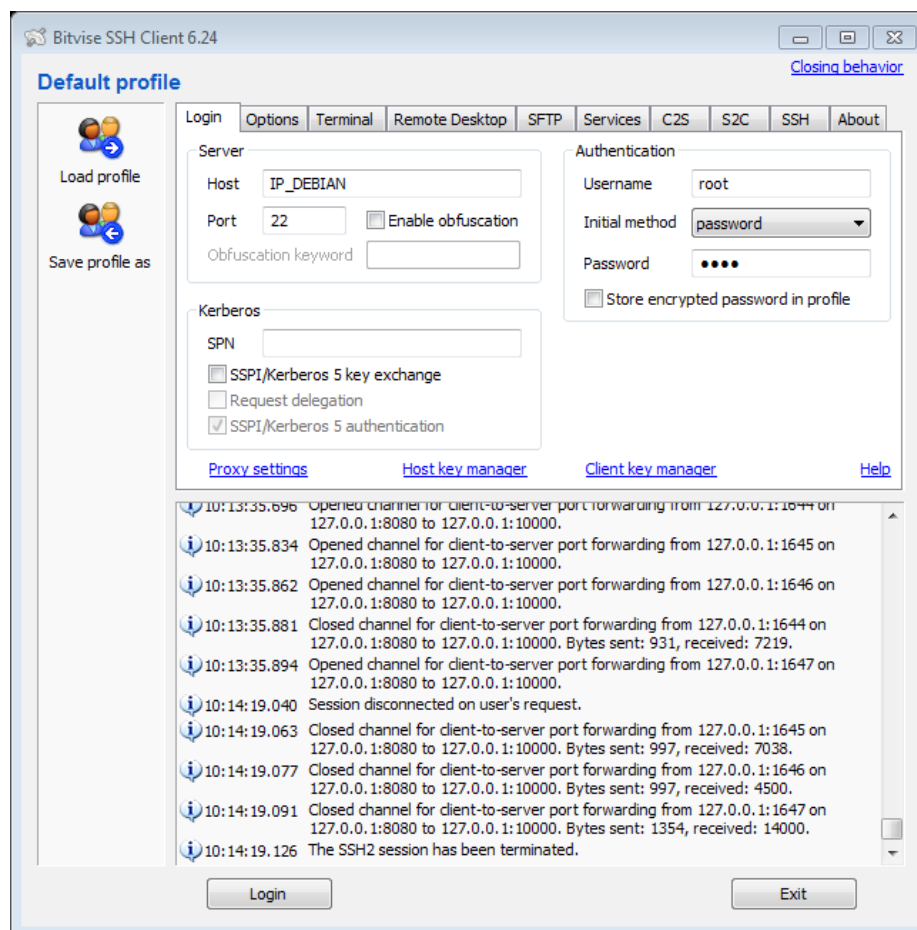
→ Sur un poste distant et avec un navigateur Web, connectez-vous à webmin par l'adresse `ip` de votre machine Linux (`http://<ip_linux>:10000`).

#### 3.3. Utilisation de Bitvise Tunnelier pour établir le tunnel

→ Installez Bitvise Tunnelier sur votre PC Windows.

→ Tapez la commande `netstat -a -n` dans la console DOS de Windows et vérifiez qu'aucun service ne tourne en écoute sur le port 8080.

→ Ouvrez Bitvise Tunnelier et entrez les paramètres de connexion suivants :



L'onglet C2S permet de créer une redirection d'un client vers un serveur. « List. Port » correspond à notre port x du schéma, il est choisi au hasard. « Destination Host » est 127.0.0.1 car le serveur applicatif se trouve sur la même machine que le serveur web. « Destination Port » est le port de webmin 10000.

➔ Cliquez sur le bouton « Login » et vérifiez avec la commande `netstat -an` dans la console DOS de Windows que le port 8080 est ouvert en écoute.

➔ Dans le navigateur web de votre poste local (Windows), tapez l'url : `https://127.0.0.1:8080`

Vous êtes connecté à votre webmin via un tunnel SSH.

Remarque : L'onglet S2C vous permet de faire une redirection SSH distante.

### 3.4. Utilisation d'OpensSSH en ligne de commande pour établir le tunnel

➔ Tapez la commande suivante sur votre poste local (Linux ou Windows via PowerShell) en remplaçant les paramètres IP par l'IP de votre serveur SSH (le serveur applicatif étant sur le serveur SSH) :

```
ssh -L 8080:<IP_serveur_Webmin>:10000 <IP_serveur_SSH>
```

➔ Une fois votre tunnel établi, dans le navigateur web de votre poste local, tapez l'url : `https://127.0.0.1:8080`.

Vous êtes connecté à votre webmin via un tunnel SSH.

### 3.5. Étude du flux SSH

➔ En lançant Wireshark lors de votre connexion SSH, vérifiez que le tunnel SSH est bien établi et analysez vos échanges avec le serveur Webmin.

# CAHIER DE TRAVAUX PRATIQUES

**- RÉPONSES -**

# TRAVAUX PRATIQUES

## Le système de fichiers - **REPONSES**

---

### TABLE DES MATIÈRES

---

|                                             |   |
|---------------------------------------------|---|
| 1. Rappels.....                             | 1 |
| 2. La manipulation de fichiers.....         | 1 |
| 2.1. La navigation dans l'arborescence..... | 1 |
| 2.2. Les liens.....                         | 1 |
| 2.3. Les archives TAR.....                  | 2 |
| 2.4. Les caractères spéciaux du Bash.....   | 2 |
| 3. Les éditeurs.....                        | 2 |
| 3.1. Nano.....                              | 2 |
| 3.2. Vi et Vim.....                         | 2 |

## 1. RAPPELS

## 2. LA MANIPULATION DE FICHIERS

---

### 2.1. La navigation dans l'arborescence

[1] Nous sommes dans notre home : /home/monlogin. La commande à utiliser est `# pwd`

[2] `$ cd /`

[3] `$ ls ou $ ls .`

[4] `$ cd /root`

[5] `$ cd ../etc`

[6] `$ cd ~`

[7] `$ mkdir ~/tmp`

[8] `# mkdir --parent mon-rep-1/mon-ssrep-11 mon-rep-2/mon-ssrep-21`

`# touch mon-fichier mon-rep-1/mon-premier-fichier mon-rep-2/mon-ssrep-21/mon-fichier-21`

L'auto-complétion peut être utilisée pour compléter le nom des répertoires et des sous-répertoires lors de la saisie de la commande touch.

### 2.2. Les liens

[9] `# ln -sr mon_fichier mon-rep-2/mon-fichier-lien`

L'option r permet de faire des liens lorsque la TARGET est spécifiée avec un chemin relatif. Les i-nodes des fichiers sont différents (option -i de ls pour afficher les i-nodes). Le lien est indiqué par la lettre l au début de la chaîne indiquant les droits (avec l'option -la de ls).

[10] Le contenu des deux fichiers est le même.

[11] Lorsqu'on nous supprimons mon-fichier, le listing sur le répertoire mon-rep-2 indique que le lien est cassé en surlignant en rouge le fichier.

## 2.3. Les archives TAR

## 2.4. Les caractères spéciaux du Bash

[12] L'utilisateur titi n'existe pas, donc le répertoire /home/titi non plus. La commande a donc retourné une erreur sur la destination finale.

[13] Cette commande a copié tous les fichiers et dossiers commençant par la lettre m dans le sous-répertoire mon-rep-2 de mon home.

La commande équivalente serait : `# cp -R /home/usera/mon-rep-1/mon-premier-fichier /home/usera/mon-rep-1/mon-ssrep-11 /home/usera/mon-rep-2`

Notez que la commande cp a pris 3 arguments. En effet, la commande cp peut prendre plusieurs fichiers en source, le dernier argument est considéré comme la destination.

[14] La commande a créé un dossier ~toto dans notre /home. Pour supprimer la commande, on doit protéger le caractère ~ : `# rm -fr \~toto`

[15]

```
$ ls [aA]*
```

```
$ ls ~usera/mon-rep-1/*.cxx
```

```
$ ls *[*]* mais aussi $ ls *'*'* ou même $ ls *'"'*
```

```
$ ls /home/*/*.*
```

[16]

```
$ ls -l 'un nom bizarre*'
```

```
$ ls -l "un nom bizarre*"
```

Pour protéger une chaîne de caractères, on utilisera de préférence les apostrophes. Les guillemets ne doivent être utilisés que si l'on veut que bash traite quand même les caractères spéciaux \$, ' ou !.

**Source des questions 15 et 16 : Cyril Paint-Barre**

## 3. LES ÉDITEURS

---

### 3.1. Nano

### 3.2. Vi et Vim

[17] L'action a copié la ligne sur laquelle nous étions placé et l'a collé en dessous de cette ligne.

[18] L'action a copié la ligne sur laquelle nous étions placé ainsi que la suivante (2 lignes copiées) et l'a collé à la fin du texte.

[19] Idem mais avec 4 lignes.

[20] L'action a supprimé le caractère sur lequel nous étions placé.

[21] L'action a supprimé la ligne sur laquelle nous étions placé. Remarque : la ligne est palcé dans le buffer, avec la commande p effectuée immédiatement après le dd, nous pouvons donc coller cette ligne n'importe où dans le fichier (équivalent du couper/coller).

[22] L'action a effectué un « undo ».

[23] Pour afficher les numéros de ligne, en mode commande, tapez la ligne suivante : **:set nu**

[24] La syntaxe de la commande à utiliser est la suivante :

```
:%s/texte_à_trouver/nouveau_texte/g
```

# TRAVAUX PRATIQUES

## La gestion des utilisateurs - **REPONSES**

---

### TABLE DES MATIÈRES

---

|                                                       |   |
|-------------------------------------------------------|---|
| 1.Introduction et rappels.....                        | 1 |
| 2.La gestion des utilisateurs et des groupes.....     | 1 |
| 2.1.Les outils de base.....                           | 1 |
| 2.2.Sudo.....                                         | 1 |
| 3.Création d'un environnement multi-utilisateurs..... | 1 |
| 3.1.Création de l'environnement.....                  | 1 |
| 3.2.Manipulations.....                                | 2 |

## 1. INTRODUCTION ET RAPPELS

## 2. LA GESTION DES UTILISATEURS ET DES GROUPES

---

### 2.1. Les outils de base

### 2.2. Sudo

#### [1] Qu'avez-vous observé ?

Lors de la première manipulation, en étant connecté avec votre utilisateur, lorsque vous avez tapé `$ sudo whoami` le système ne vous a pas demandé de mot de passe contrairement à la commande `$ sudo bash`.

Lors de la deuxième manipulation, les deux commandes n'ont pas demandé de mot de passe.

## 3. CRÉATION D'UN ENVIRONNEMENT MULTI-UTILISATEURS

---

### 3.1. Création de l'environnement

#### [2]

➔ Création de 2 groupes :

```
groupadd groupe1; groupadd groupe2
```

Pour vérifier : `# cat /etc/group`

➔ Création des 4 utilisateurs avec création de leur répertoire home, en spécifiant le shell utilisé ainsi que le placement dans leur groupe primaire et secondaire :

```
useradd -m user1 -g groupe1 -s /bin/bash ;
useradd -m user2 -g groupe1 -G groupe2 -s /bin/bash ;
useradd -m user3 -g groupe2 -s /bin/bash ;
useradd -m user4 -g groupe2 -G groupe1 -s /bin/bash ;
```

Pour vérifier : `# ls -al /home/ ; cat /etc/group`

➔ Création des répertoires communs :

```
mkdir /home/groupe1
```

```
mkdir /home/groupe2
```

- Mise en place des permissions pour permettre aux utilisateurs d'écrire dans le répertoire de leur groupe :

```
chgrp groupe1 /home/groupe1
```

```
chgrp groupe2 /home/groupe2
```

### 3.2. Manipulations

[3] On constate que tous les utilisateurs peuvent lire les fichiers des autres utilisateurs. Par exemple, user1 peut lire le fichier de **/home/user2**.

En revanche, user1 ne peut pas écrire dans **/home/user2**.

On remarque aussi qu'aucun utilisateur n'a le droit d'écriture dans le dossier de son groupe, de son groupe secondaire et de l'autre groupe.

Nous solutionnerons ces problèmes dans le TP « Gestion des droits ».

[4] Le groupe propriétaire du fichier est maintenant groupe2. L'utilisateur 3 a le droit de lire le fichier mais toujours pas les droits d'écriture. L'utilisateur 1 a aussi le droit de lire ce fichier...

[5] La commande newgrp permet à un utilisateur de changer de groupe durant sa session, même s'il ne fait pas partie du groupe en question. Si on configure un mot de passe sur le groupe, celui-ci est demandé si l'utilisateur fait un newgrp sans faire partie de ce groupe. Si l'utilisateur appartient à ce groupe (même en groupe secondaire), alors le mot de passe n'est pas demandé.

user3 et user4, même en faisant partie du groupe2 ne peuvent pas modifier le mot de passe.

[6] L'user3 peut maintenant modifier le mot de passe du groupe, user4 toujours pas.



# TRAVAUX PRATIQUES

## Les droits sur les fichiers - **REPONSES**

---

### TABLE DES MATIÈRES

---

|                                                                             |   |
|-----------------------------------------------------------------------------|---|
| 1. Rappels.....                                                             | 1 |
| 2. La gestion des droits des fichiers.....                                  | 1 |
| 2.1. Savoir lire les droits.....                                            | 1 |
| 2.2. Modifier les droits sur un fichier.....                                | 1 |
| 3. Umask.....                                                               | 2 |
| 3.1. Umask.....                                                             | 2 |
| 4. comprendre la propagation des droits.....                                | 2 |
| 5. Création d'un environnement multi-utilisateurs – Suite.....              | 3 |
| 5.1. Mise en place de règles de gestion des droits sur les répertoires..... | 3 |

### 1. RAPPELS

### 2. LA GESTION DES DROITS DES FICHIERS

---

#### 2.1. Savoir lire les droits

**Source de l'exercice : Cyril Paint-Barre**

- [1] La première ligne affichée par `ls` indique que `truc` est le propriétaire de `.` (répertoire de travail).
- [2] `truc` est le propriétaire de `fic`. Il possède donc les droits `r-x`.
- [3] `bidule` est membre du groupe de `fic`. Il possède donc les droits `rw-`.
- [4] `bidule` n'est ni le propriétaire de `fic`, ni membre de son groupe. Il a donc les droits `-wx`.
- [5] Pour supprimer `fic`, il faut au minimum les droits d'écriture et d'exécution sur le répertoire de travail. Ceux qui les possèdent sont `truc` et les utilisateurs qui ne sont pas membre du groupe `bidule`.
- [6] Non, s'il n'y a pas eu intervention de `root`, `rep` n'a pu être créé qu'avec des permissions différentes du répertoire de travail (ou alors par déplacement mais ce n'est pas une création). En effet, le propriétaire (et créateur) de `rep` est `machin`. Puisque le groupe de `rep` est `bidule` et que seul `machin` (ou `root`) peut le changer, on en déduit que `machin` est membre du groupe `bidule`. Or, pour créer un fichier il faut les droits `wx` sur le répertoire mais `machin` a les droits du groupe, soit uniquement `r`.
- [7] Aucun droit ! En effet, en tant que membre du groupe `bidule`, `machin` n'a pas les droits d'exécution sur le répertoire de travail et ne peut donc pas accéder à `rep` !

#### 2.2. Modifier les droits sur un fichier

- [8] `$ chmod 700 mon-fichier` puis `$ ls -l mon-fichier`.
- [9] Avec le mode symbolique, on aurait pu écrire `$ chmod u+x,go-r mon-fichier`
- [10] `$ touch essai.txt ; $ chmod 404 essai.txt` ou `$ chmod u-w,o+r essai.txt`

### 3. UMASK

---

#### 3.1. Umask

[11] 422 correspond à un droit **r--w-w-** et 533 à **r-x-wx-wx**

[12] un umask de 000 !!

### 4. COMPRENDRE LA PROPAGATION DES DROITS

---

**Source de l'exercice : Cyril Paint-Barre**

[13]

**Commande cp :**

Lorsqu'un fichier (au sens large) est créé en effectuant une copie, celui-ci a pour propriétaire et pour groupe ceux de l'utilisateur effectif qui réalise la copie. Ses permissions sont celles du fichier d'origine moins celles du masque de création de fichier.

Si la destination est un fichier qui existe, cp ne le supprimera pas mais tentera de remplacer son contenu (l'écraser), ce qui nécessite le droit d'écriture sur la destination. Dans ce cas, même si la copie réussit, la destination garde son propriétaire, son groupe et ses permissions. Seuls son contenu et sa date de modification changent. Si ce droit est absent (ce qui provoquerait une erreur), l'option **-f** tente de supprimer destination avant de procéder à la copie. Si la suppression réussit et que l'utilisateur a les droits d'écriture sur le répertoire de copie, tout se passe comme si la copie créait un nouveau fichier.

Les options **-p** et **-a** sont principalement destinées à root : **-p** demande que la copie garde les mêmes propriétaire/groupe, permissions et dates que la source. L'option **-a** va plus loin et demande de préserver tout ce qui est possible du fichier source, tout en activant les options **-d** et **-r**.

**Commande mv :**

Pour déplacer/renommer un fichier dans une même partition, il faut avoir le droit d'accéder aux répertoires source et destination et le droit de les modifier (écriture). Aucun droit sur le fichier lui-même n'est nécessaire.

Lorsque le déplacement/renommage se fait d'une partition à l'autre, un droit supplémentaire est nécessaire : celui de lecture du fichier source, car il faut copier son contenu sur la nouvelle partition avant de le supprimer.

Les droits du fichier source sont conservés dans le cas du déplacement dans une même partition.

Le déplacement de fichiers peut conduire à la suppression de fichiers existants. Il ne s'agit pas vraiment d'écrasement, car pour que le déplacement réussisse, les permissions du fichier qui sera supprimé n'ont aucune importance. Seules comptent les permissions sur le répertoire de destination qui doivent être l'écriture et l'exécution.

À noter que le déplacement d'un répertoire peut conduire de la même manière à la suppression d'un répertoire existant, mais celui-ci doit être vide. D'autre part, le déplacement d'un fichier ne peut pas conduire à la suppression d'un répertoire, et inversement.

---

1/ Lors d'une copie d'un fichier vers un fichier existant, les droits du fichier de destination sont conservés. On a les droits de lecture sur fic1 et les droits d'écriture sur fic2, donc la copie réussie.

2/ Lors de la création d'un fichier, celui-ci est créé avec des droits 777 – umask – les droits d'exécution. Bien sûr, il faut avoir les droits d'écriture dans le dossier.

Par exemple, lorsque un utilisateur crée un fichier, le fichier créé a les droits 777-023= 753. Mais comme un fichier en création n'a jamais les droits x, on a les droits 642 (soit rw-r---w-).

3/ Lors d'un déplacement (mv) d'un fichier vers un fichier existant, les droits du fichier de destination sont remplacés par les droits du fichier source.

4/ La modification d'un fichier par un autre utilisateur que le propriétaire n'entraîne pas de changement de droits.

5/ La commande `$mv -f fic7 rep1/fic8` échoue car toto ne dispose pas des droits d'écriture sur rep1 .

6/ L'utilisateur n'a pas les droits d'écriture sur le fichier f6. L'option -f de la commande cp agit de la façon suivante : les droits du fichier source sont conservés en retirant l'umask (comme si c'était une création mais à partir des droits du fichier source). Le propriétaire et le groupe propriétaire changent comme si le fichier était créé.

Ici, on a le fichier 1 qui a les droit -wxr-xrwx, en retirant l'umask (---w--wx), les droits deviennent -wxr-wr--.

7/ Lors d'une copie d'un fichier vers un fichier non existant, le fichier de destination est créé avec les droits du fichier source et l'umask qui est ici de 023. L'umask est soustrait au droit de l'utilisateur sur le dossier et le fichier.

Le fichier fic1 a les droits 357. Le fichier en création à partir du fichier fic1 pour l'user toto du groupe users aura donc un droit de 357 – 023 (l'umask), soit 354 (-wxr-xr--).

[14]

| Fichier | Propriétaire | Groupe | Permissions |
|---------|--------------|--------|-------------|
| fic2    | titi         | tata   | rw-r-x-w-   |
| fic10   | toto         | users  | rw-r---w-   |
| fic4    | truc         | chose  | rwxr-----   |
| fic5    | titi         | chose  | rwxrw--w-   |
| fic6    | toto         | users  | -wxr-xr--   |
| fic9    | toto         | users  | -wxr-xr--   |

## 5. CRÉATION D'UN ENVIRONNEMENT MULTI-UTILISATEURS – SUITE

### 5.1. Mise en place de règles de gestion des droits sur les répertoires

[15] Les commandes à taper sont les suivantes :

- pour chaque /home des utilisateurs, on enlève tous les droits aux autres :

```
chmod -R o=-r-x /home/user1
```

```
chmod -R o=-r-x /home/user2
```

```
chmod -R o=-r-x /home/user3
```

```
chmod -R o=-r-x /home/user4
```

Remarque : on utilise l'option -R pour que les droits s'appliquent sur les fichiers existants.

- pour chaque /home des groupes, on ajoute le droit d'écriture aux membres du groupes :

```
chmod -R g=+rwx /home/groupe1
```

```
chmod -R g=+rwx /home/groupe2
```

[16] La valeur 067 offre le meilleur compromis entre la sécurité et la souplesse. Elle permet le parcours des répertoires sans en autoriser l'examen avec la commande "ls" tout en autorisant l'accès à des répertoires fils dont les droits seront plus permissifs, utilisés pour échanger des fichiers.

# TRAVAUX PRATIQUES

## Le shell - **REPONSES**

---

### TABLE DES MATIÈRES

---

|                                                                       |          |
|-----------------------------------------------------------------------|----------|
| <b>1. Le shell.....</b>                                               | <b>1</b> |
| 1.1. L'historique des commandes.....                                  | 1        |
| 1.2. Le copier/coller en mode console.....                            | 2        |
| 1.3. Personnalisation de votre environnement.....                     | 2        |
| 1.4. Traitement d'une ligne de commande avec le backquoting.....      | 2        |
| <b>2. Les outils du shell.....</b>                                    | <b>3</b> |
| 2.1. Les redirections.....                                            | 3        |
| 2.2. Les pipes.....                                                   | 3        |
| 2.3. Les filtres.....                                                 | 3        |
| 2.4. Les filtres et les pipes.....                                    | 3        |
| 2.5. Grep et les expressions régulières.....                          | 3        |
| 2.6. Sed et les expressions régulières.....                           | 4        |
| 2.7. Pour aller plus loin.....                                        | 4        |
| <b>3. Les scripts.....</b>                                            | <b>5</b> |
| 3.1. Les tests.....                                                   | 5        |
| 3.2. Exemple de script.....                                           | 5        |
| <b>4. Création d'un environnement multi-utilisateurs – Suite.....</b> | <b>7</b> |

---

## 1. LE SHELL

---

### 1.1. L'historique des commandes

[1] `$ !1`

Rappel de la première commande tapée.

[2] `$ !!`

Rappel de la dernière commande tapée.

[3] `$ !c`

[4] `$ history -c`

[5] variable `$history`

[6] `$ HISTTIMEFORMAT='%F %T '`

## 1.2. Le copier/coller en mode console

## 1.3. Personnalisation de votre environnement

[7] Le format du prompt sera le suivant (**en gras et bordeaux** : les chaînes de caractères. *En italique* : des variables) :

**[ Numéro de la ligne de commande courante ]** - **Utilisateur courant**(*nom-machine*) **Répertoire courant**:

[8] `$ PS1="\t \w $"`

[9] `$ alias cl="clear"`

## 1.4. Traitement d'une ligne de commande avec le backquoting

(Source des questions 1.4 : Cyril Pain-Barre, IUT Aix-en-Provence, sous licence CC)

[10] `$ ls *\**`

[11] `$ ls -l un\ nom\ bizarre\*`

`$ ls -l un 'nom' 'bizarre\*`

[12]

- `$ echo Je fais partie des arguments.`

Le nombre d'espaces séparant les arguments n'a pas d'importance. Il y a 5 arguments à echo qui les écrits séparés par une seule espace.

- `$ echo "Je fais partie des arguments."`

Les espaces sont affichées telles quelles car elles sont protégées par les guillemets et il n'y a plus qu'un seul argument à echo.

- `$ echo Je fais "partie d'es" argum ents.`

Les quotes et les guillemets peuvent être placés n'importe où. Ici, ils protègent les espaces supplémentaires. echo reçoit les deux arguments *Je* et *fais partie des arguments*. et on obtient le même affichage qu'à la question précédente.

[13]

- `$ echo --- * --- $PATH --- !!`

echo reçoit de nombreux arguments car bash remplace \* par la liste triée des fichiers du répertoire ) ; \$PATH par /usr/local/bin:/usr/bin:/bin:... , et !! par la dernière commande ( echo )

- `$ echo --- \* --- \$PATH --- \!!`

bash supprime les backslashes protégeant les caractères spéciaux qu'il ne traite pas, et echo reçoit 6 arguments

- `echo '--- * --- $PATH --- !!'`

bash supprime les quotes protégeant les caractères spéciaux, et echo ne reçoit qu'un argument

- `echo "--- * --- $PATH --- !!"`

bash supprime les guillemets qui ne protègent ni \$PATH ni !! qui sont remplacés. Mais echo ne reçoit qu'un seul argument.

- `echo "--- * --- \$PATH --- \!!"`

bash supprime les guillemets dans lesquels \$PATH est protégé par le backslash. À l'extérieur des guillemets, !! est aussi protégé par un backslash. Au total, echo reçoit deux arguments.

[14] Voici une possibilité parmi tant d'autres :

```
$ echo "~ C'est * bon * pour tout ?" "\\" " !!! * ~ $PATH "" ' ??"
```

## 2. LES OUTILS DU SHELL

---

### 2.1. Les redirections

[15] `$ date > date.txt; more date.txt`

[16] `$ echo "coucou" > date.txt; more date.txt`

[17] `$ date >> date.txt; more date.txt`

### 2.2. Les pipes

[18] `# figlet "Il fait un peu chaud" | cowsay -n | boxes -d sunset`

### 2.3. Les filtres

[19] `$ echo -e "aaa\nbbb\nbbb\nbbb\naaa\nbbb\nccc\nccc" > liste_caractere.txt`

[20] `$ uniq -d -c liste_caractere.txt`

[21] `$ uniq -u liste_caractere.txt`

[22] La commande split permet le découpage en fichier. La commande pour découper le fichier en list\_car01 est la suivante :

```
$ split -l 3 -d liste_caractere.txt liste_car
```

### 2.4. Les filtres et les pipes

[23] `$ head -n 3 matrice.txt | tail -n 1`

[24] `$ head -n 3 matrice.txt | tail -n 1 | cut -f 2`

[25]

```
$ cut -f 1 matrice.txt | sort -n > temporaire1.txt
```

```
$ cut -f 2 matrice.txt | sort -n > temporaire2.txt
```

```
$ cut -f 3 matrice.txt | sort -n > temporaire3.txt
```

```
$ cut -f 4 matrice.txt | sort -n > temporaire4.txt
```

```
$ paste temporaire1.txt temporaire2.txt temporaire3.txt temporaire4.txt
```

[26] la commande compte le nombre de répertoires et de sous-répertoires présents dans notre \$HOME

### 2.5. Grep et les expressions régulières

[27] `$ grep "rla" exo_regex.txt`

[28] `$ grep ans$ exo_regex.txt`

[29] `$ grep -E "(alpha|exe)" exo_regex.txt`

[30] `$ grep ^[^0-9] exo_regex.txt`

[31] `$ grep ^[0-9aeiaouAEIOU] exo_regex.txt`

[32] `$ grep -E "([0-9]|[0-9]$)" exo_regex.txt`

[33] `$ grep -E "([0-9]|[0-9]$)" exo_regex.txt`

[34] La commande recherche les lignes qui font moins de 10 caractères.

## 2.6. Sed et les expressions régulières

[35] `# cat exo_regex.txt | sed -e "s/^\.{4}\(\.\{6\}\)\.\.\{1\} \2/"`

[36] `# cat exo_regex.txt | sed -e "s/^\.{14}\//"`

[37] `sed -re 's/:x:.*//' /etc/passwd | sed -re 's/^/user: /' > liste_user`

[38] Il y a plusieurs façons de faire. Soit on mémorise ce qu'on veut garder pour le réécrire (ou le réutiliser différemment) et on élimine le reste :

`ls -l *.txt | sed -e 's/^\([^ ]*\).*\([ ]*\)$/\1\2/'`

soit on élimine uniquement ce qu'on ne veut pas (en le remplaçant par un espace) en se disant que ce qu'on veut éliminer est entre le premier et le dernier espace de la ligne. En utilisant le fait que `.*` correspond à la plus grande chaîne possible, on peut simplement écrire :

`ls -l *.txt | sed -e 's/ .* / /'`

## 2.7. Pour aller plus loin

[39] Étape par étape, voici une méthode pour vérifier un email :

On ne veut QUE l'adresse e-mail ; on va donc demander à ce que ça soit un début et une fin de chaîne :

`^$`

Ensuite, on a des lettres, chiffres, tirets, points, underscores, au moins une fois. On utilise donc la classe `[a-z0-9._-]` à la suite de laquelle on rajoute le signe `+` pour demander à ce qu'il y en ait au moins un :

`^[a-z0-9._-]+$`

Vient ensuite l'arobase:

`^[a-z0-9._-]+@`

Puis encore une suite de lettres, chiffres, points, tirets, au moins deux fois. On tape donc `{2,}` pour dire « deux fois ou plus » :

`^[a-z0-9._-]+@[a-z0-9._-]{2,63}$`

Ensuite vient le point (de « .fr » par exemple). Attention, le point est un caractère spécial qui sert à indiquer « n'importe quel caractère » (même des accents). Or, ici, on veut enlever sa signification au point pour dire que l'on veut le symbole point dans notre regex. On va donc mettre un antislash devant :

`^[a-z0-9._-]+@[a-z0-9._-]{2,}\.$`

Enfin, pour terminer, il nous faut deux à quatre lettres. Ce sont forcément des lettres minuscules, et cette fois pas de chiffre ou de tiret, etc. On écrit donc :

`^[a-z0-9._-]+@[a-z0-9._-]{2,}\.[a-z]{2,4}$`

Et voilà encore la regex pour vérifier les emails :

`^[a-z0-9._-]+@[a-z0-9._-]{2,}\.[a-z]{2,4}$`

A vous de jouer pour améliorer la regex pour vérifier les règles manquantes (nombre de caractères avant l'arobase, premier caractère vérifié, caractère avant l'@ vérifié...).



## 3. LES SCRIPTS

---

### 3.1. Les tests

[40] La commande `echo $?` affiche le code retour du test, donc 1 si le test est faux (car on a une erreur), 0 si le test est vrai.

[41] `$ test -e mon_fichier`

[42] Jeu du pile ou face : on divise par deux un chiffre au hasard. Si le chiffre au hasard est paire, alors le test « `-eq 1` » est vérifié et on affiche Pile. Sinon, on affiche face.

### 3.2. Exemple de script

Ne pas oublier :

- de renommer le fichier pour enlever l'extension `.txt` qui ne représente pas un script exécutable :

`# mv exemple_script.txt exemple_script`

- d'exécuter un `# chmod +x exemple_script` pour le rendre exécutable.

Pour lancer le script en affichant l'aide à l'usage : `# ./exemple_script`

Pour lancer le script avec l'option start : `# ./exemple_script start`

[43]

Contenu du script avec la partie "Pile ou face" dans la partie start :

```
#!/bin/bash
BEGIN INIT INFO
Provides: nom_de_mon_script
Required-Start:
Required-Stop:
Default-Start: 2 3 4 5
Default-Stop: 0 1 6
Short-Description: Description de mon script
Description:
END INIT INFO

case "$1" in
 start)
 echo "Le script démarre ici"
 case "$2" in
 PileouFace)
 test $[RANDOM%2] -eq 1 && echo "Pile" || echo "Face"
 ;;
 esac
 ;;
 stop)
 echo "Le script s'arrête ici"
 ;;
 restart)
 $0 stop
 sleep 2s
 $0 start
 ;;
 status)
 echo "Statut de mon script"
 ;;
 *)
 echo "Usage: {start|stop|restart|status}"
 exit 1
 ;;
esac
exit 0
```

Pour tester votre script : # `./exemple_script start PileouFace`

[44]

```
case "$1" in
 start)
 echo "Le script démarre ici"
 case "$2" in
 PileouFace)
 test $[RANDOM%2] -eq 1 && echo "Pile" || echo "Face"
 ;;
 CreationFichier)
 for numero in `seq 1 50`
 do touch fichier"$numero"
 done
 ;;
 esac
 ;;
 ;;
```

[45] Un exemple de dialogue :

```
case "$1" in
 start)
 echo "Le script démarre ici"
 case "$2" in
 PileouFace)
 test $[RANDOM%2] -eq 1 && echo "Pile" || echo "Face"
 ;;
 CreationFichier)
 for numero in `seq 1 50`
 do touch fichier"$numero"
 done
 ;;
 Dialoguez)
 echo "Souhaitez-vous continuer ce TP ?"
 select opinion in Oui Non
 do
 case $opinion in
 "Oui"|"Non") break;;
 *) continue;;
 esac
 done
 echo "Votre réponse est $opinion"
 ;;
 esac
 ;;
 ;;
```

## 4. CRÉATION D'UN ENVIRONNEMENT MULTI-UTILISATEURS – SUITE

---

Dans /etc/profile, on ajoute par exemple :

```
echo "Bienvenue"
echo "entrez U pour travailler dans votre repertoire"
echo "entrez G pour travailler dans le repertoire de votre groupe"
read CHX
while [$CHX != 'U'] && [$CHX != 'G']
do
echo "Entrez U ou G ..."
read CHX
done
if [$CHX == "G"]
then
case $USER in
 "user1") export HOME=/home/groupe1;;
 "user2") export HOME=/home/groupe1;;
 "user3") export HOME=/home/groupe2;;
 "user4") export HOME=/home/groupe2;;
esac
cd $HOME
fi
```

# TRAVAUX PRATIQUES

## Les services **REPONSES**

### TABLE DES MATIÈRES

|                                                   |          |
|---------------------------------------------------|----------|
| <b>1. Les processus – Commandes de base</b> ..... | <b>1</b> |
| 1.1. La surveillance des processus.....           | 1        |
| 1.2. Les jobs.....                                | 2        |
| 1.3. Gérer les jobs.....                          | 2        |
| <b>2. Les processus en arrière plan</b> .....     | <b>2</b> |
| 2.1. Nohup et les binaires GNU.....               | 2        |
| 2.2. Nohup et vos binaires.....                   | 2        |

### 1. LES PROCESSUS – COMMANDES DE BASE

#### 1.1. La surveillance des processus

[1] Ps affiche le listing des process en cours d'exécution pour l'utilisateur connecté

a : all, processus des autres utilisateurs

x : affiche aussi les processus qui n'ont pas de terminal de contrôle

Remarque : ps est une commande où les options n'ont pas besoin d'être marquées d'un «-».

[2] `# ps ax | grep init`

PID = 1

Vous constatez que votre filtre retourne un deuxième processus avec l'expression « init », ce processus étant la commande ps elle-même.

Si vous ne souhaitez pas que cette ligne s'affiche, il faut utiliser le grep suivant :

`# ps ax | grep init | grep -v grep`

qui exclut le grep

[3] `# ps ax | grep init | grep -v grep | sed -re 's/ +/ /g'`

-re : expression régulière avancée

s : recherche la chaîne de caractère

+ : tabulations

g : action de remplacement

[4] Lancez top puis tapez sur la touche [s] et le chiffre [1], puis [entrée] : rafraîchissement toutes les secondes.

Tapez la touche [m] puis [entrée] : classement par utilisation de la mémoire.

[5] On constate que systemd est bien le père de tous les processus.

## 1.2. Les jobs

[6] La commande sleep met une attente de x secondes.

Pour arrêter ce processus, utilisez la combinaison de touche [CrI]+[C] ou dant un autre terminal :

```
ps ax | grep sleep
kill -9 <pid_sleep>
```

[7]

Avec le &, l'utilisateur reprend la main sur son terminal.

- ➔ \$ jobs -l
- ➔ (option -l (L minuscule)) pour afficher toutes les informations disponibles.
- ➔ Entre [], le numéro de job suivi du PID.

## 1.3. Gérer les jobs

## 2. LES PROCESSUS EN ARRIÈRE PLAN

---

### 2.1. Nohup et les binaires GNU

[8] # ping 8.8.8.8 > ping-tmp.txt &  
# ps ax | grep ping  
# pstree -p -H <valeur\_pid\_du\_ping\_récupérée\_précédemment>

Pour lancer pstree, pensez à agrandir en largeur la fenêtre de votre terminal. Le PID se trouve entre le 5ème et le 12ème niveau d'arborescence.

Après fermeture de la fenêtre, le PID se trouve maintenant en deuxième position et devient un fils direct de systemd. Vous remarquez aussi qu'avec un \$ px -ax le processus n'est plus rattaché à un terminal ou une console.

Pour arrêter le processus :

```
kill -3 <valeur_pid_du_ping_récupérée_précédemment>
```

### 2.2. Nohup et vos binaires

[9]

```
apt-get install gcc build-essential
gcc test.c -o test
chmod +x test
./test
```

Dans un autre terminal :

```
ps ax | grep test
pstree -p -H <valeur_récupérée_précédemment>
```

Le PID se trouve entre le 8ème et le 12ème niveau d'arborescence.

Dans l'autre terminal :

```
ps ax | grep test
```

Le binaire a été arrêté !

[10]

```
nohup ./test
```

Dans un autre terminal :

```
#ps ax | grep test
```

```
pstree -p -H <valeur_récupérée_précédemment>
```

Le PID se trouve entre le 8ème et le 12ème niveau d'arborescence.

Quand vous fermez le terminal où s'exécute test :

```
#ps ax | grep test
```

Le binaire tourne toujours !

```
#pstree -p -H <valeur_récupérée_précédemment>
```

Il est rattaché à systemd.

```
#ps ux | grep test
```

Vous remarquerez que le binaire n'est plus attaché à un terminal.

# TRAVAUX PRATIQUES

## La gestion des paquets - Réponses

---

### TABLE DES MATIÈRES

---

|                                               |   |
|-----------------------------------------------|---|
| 1. Introduction.....                          | 1 |
| 2. La gestion des paquets sous Debian.....    | 1 |
| 3. La gestion des paquets sous AlmaLinux..... | 1 |

### 1. INTRODUCTION

### 2. LA GESTION DES PAQUETS SOUS DEBIAN

### 3. LA GESTION DES PAQUETS SOUS ALMALINUX

---

[1] À l'aide des pages de manuel de rpm et yum, déterminez comment effectuer les actions suivantes et testez-les :

(a) Rechercher le paquet nmap par mots-clés (scanner, security)

# yum search security ou # yum search scanner

# yum list nmap (quand on connaît le nom de l'utilitaire).

(b) Afficher des informations sur le paquet nmap

# yum info nmap

(c) Lister les paquets installés sur le système (avec RPM)

# rpm -qa

(d) Installer le paquet nmap et ses dépendances (avec YUM)

# yum install nmap

(e) Lister les fichiers du paquet nmap précédemment installé (avec RPM)

# rpm -ql nmap

(f) Installer manuellement un fichier RPM (ne pas tester)

# rpm -ihv <mon\_package.rpm>

(g) Supprimer le paquet nmap installé

# yum remove nmap

# TRAVAUX PRATIQUES

## La chaîne de démarrage - RÉPONSES

---

### TABLE DES MATIÈRES

---

|                                  |   |
|----------------------------------|---|
| 1. Script Initd.....             | 1 |
| 2. Avec Systemd (By Snorky)..... | 1 |
| 3. Le grub.....                  | 1 |

### 1. SCRIPT INITD

---

[1]

Ne pas oublier de personnaliser l'entête. Ajoutez `date > /fichier_date` dans la partie start.

Méthode 1 :

Sur Debian : `# update-rc.d script_initd defaults`

Sur Red Hat : `# chkconfig --add /etc/init.d/script_initd`

Méthode 2 : `# cd /etc/rc2.d ; ln -s ../init.d/script_initd S51script_initd`

### 2. AVEC SYSTEMD (BY SNORKY)

---

[2] Que fait ce script ?

Rick prend en entrée le fichier `$RICK_CONF` (défini dans `/etc/rick/rick.conf`) et lit ligne par ligne le fichier pour l'afficher. Une fois le fichier affiché, il reste en attente (pas de break de fin).

[3] Avez-vous le résultat escompté de la lecture du script ?

Vous vous êtes fait rickrollé !

[4] Comment faire pour que le fichier pid ne soit pas créé au démarrage du service ?

En étudiant le script, et le fichier `etc/default/sysconfig/rick` contient une variable `ARGS`. Il suffit de la commenter et ou retirer le `-p` dans celle ci.

[5] Comment mettre le service rick au démarrage ?

```
systemctl enable rick.service
```

### 3. LE GRUB

---

[6] Le temps d'affichage de l'écran du grub est défini par le paramètre **timeout**. Par défaut et sur la plupart des distribution, il est de 5s.

[7] Lors de l'affichage du grub, tapez sur la touche [E]. Vous accédez alors à l'édition manuelle de certaines options du grub.