'

Sommaire

'

Partie 1: Découvrir l'environnement Linux et installer Linux

Chapitre 1 : Généralité sur le système d'exploitation Linux

I-Notions Générales

1-Introduction

❖ **UNIX**

UNIX ou **Unix** est une famille de systèmes d'exploitation multitâche et multi-utilisateur.Son développement a commencé dans les années 1970 au centre de recherche de Bell Labs mené par Kenneth Thompson. Il repose sur un interpréteur ou superviseur(le *shell*) et de nombreux petits utilitaires, accomplissant chacun une action spécifique, commutables entre eux et appelés depuis la ligne de commande.
Les plus populaires à ce jour sont les variantes, GNU/Linux, iOS et macOS. D'une manière générale, la quasi-totalité des systèmes PC ou mobile les plus courants sont basés sur le noyau de Unix, y compris ceux commercialisés par Apple. On nomme « famille Unix », « systèmes de type Unix » ou simplement « systèmes Unix » l'ensemble de ces systèmes.

***Principales familles de systèmes Unix / Relation entre les familles de systèmes Unix (voir Annexe1)*

❖ **GNU/ Linux**

GNU est un système d'exploitation lancé en 1983 par Richard Stallman dans le but de fournir un équivalent d'Unix composé uniquement de logiciel libre. Bien que compatible avec Unix, GNU s'en démarque notamment par sa grande utilisation du Lisp.
En 1991, alors que le noyau de GNU, le Hurd traînait à être opérationnel, fut créé le noyau Linux qui fut libéré en 1992. Cela permit d'utiliser pour la première fois un système d'exploitation entièrement libre, une variante de GNU utilisant le noyau Linux connue sous le nom de GNU/Linux, ou plus couramment, simplement Linux.
GNU et GNU/Linux sont utilisés sous la forme de distributions qui les accompagnent de logiciels supplémentaires. Parmi les distributions les plus populaires, on compte notamment Debian, Ubuntu, Linux Mint, RedHat, Fedora et Arch.

❖ **La licence GPL**

Pour donner un cadre juridique aux logiciels du projet GNU, Richard Stallman écrit une licence, la GNU General Public License alias GPL. Cette licence reprend les quatre libertés fondamentales (liberté d'exécution : tout le monde a le droit de lancer le programme, quel qu'en soit le but ; liberté de modification : tout le monde a le droit d'étudier le programme et de le modifier, ce qui implique un accès au code source ; la liberté de redistribution : tout le monde a le droit de rediffuser le programme, gratuitement ou non ;liberté d'amélioration : tout le monde a le droit de redistribuer une version modifiée du programme) et impose pour la liberté d'amélioration que les versions modifiées d'un logiciel sous licence GPL ne peuvent être redistribuées que sous cette même licence.
Richard Stallman invite alors tous les logiciels libres à adopter la licence GPL.
Très tôt, Linus Torvalds adopte la licence GPL pour son noyau Linux.
Mais la licence GPL n'est pas la seule licence utilisée pour les logiciels libres. Par exemple, il existe aussi la licence BSD, qui diffère de la licence GPL par le fait qu'une version modifiée d'un logiciel sous Licence BSD peut être redistribuée sous une autre licence, même propriétaire.

'

❖ L’opensource

Un programme informatique, que ce soit un noyau ou un logiciel, est constitué de nombreuses lignes de code, écrites dans un langage de programmation (le langage C dans le cas du noyau Linux). Ce code n’est pas utilisable en tant que tel. Il faut passer par la phase de compilation qui transforme le code source en programme exécutable, souvent appelé binaire.

Il suffit d’avoir le binaire pour utiliser le programme ; on n’a pas besoin des sources du programme. Il n’existe pas de moyen de remonter aux sources du programme à partir du seul binaire. Quand on achète un logiciel (Office) ou un système d’exploitation (Windows), on a un CD qui contient le binaire, mais pas les sources. Il est donc impossible de savoir comment le programme est conçu.

Par conséquent, on ne peut pas modifier le programme. On peut seulement l’utiliser et éventuellement le copier à l’identique.

Les logiciels propriétaires sont donc les logiciels pour lesquels on n’a accès qu’aux binaires. Généralement, on doit acheter le logiciel, ce qui nous donne le droit d’utiliser les binaires dans la limite des termes du contrat de licence. Certains logiciels propriétaires sont gratuits, on les appelle freeware.

Les logiciels OpenSource sont les logiciels pour lesquels on a accès au code source, d’où le terme OpenSource.

2-Le noyau Linux

❖ Système d'exploitation

Un système d'exploitation est un logiciel qui gère un ordinateur. C'est un ensemble de données et de programmes qui gère les ressources systèmes. Il permet l'exécution de logiciels d’application en agissant comme une couche d'interface entre le matériel et les applications pour des fonctions telles que les opérations d'entrée / sortie . C'est le logiciel principal du système fonctionnant sur un ordinateur.

Les systèmes d'exploitation sont présents sur tous les types de machines (pas uniquement les ordinateurs) dotées de processeurs tels que les téléphones mobiles, les systèmes de jeu sur console, les super ordinateurs et les serveurs. Les systèmes d'exploitation les plus populaires sont Microsoft Windows, Mac OS X, UNIX, Linux et BSD.

❖ Noyau d’un système d'exploitation

Le noyau est la partie principale d’un système d’exploitation qui assure principalement la communication directe avec les ressources matérielles. Sans le noyau, un système d'exploitation ne peut pas fonctionner. Le noyau est le niveau le plus bas du système d'exploitation. C’est le cœur du système, c'est lui qui s'occupe de fournir aux logiciels une interface de programmation pour utiliser le matériel. Toutes les autres composantes du système d'exploitation (interface utilisateur graphique, gestion de fichiers, shell, etc.) dépendent de lui. Le noyau gère les tâches de base du système :

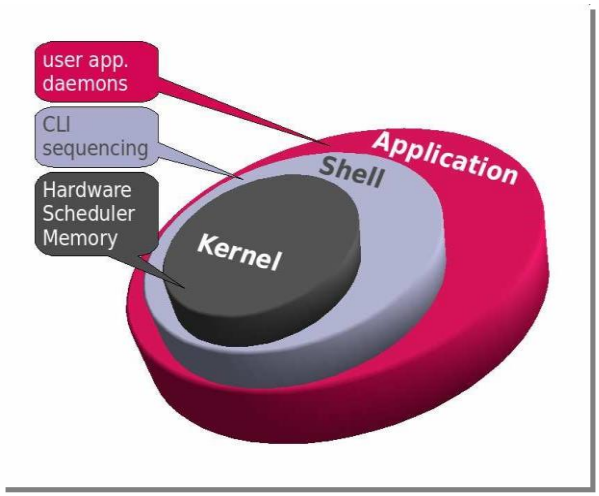
- L'initialisation du système
- La gestion des ressources
- La gestion des processus
- La gestion des fichiers
- La gestion des Entrées/Sorties

L’utilisateur communique avec le noyau par l'intermédiaire d'un SHELL. Les Shells sont aussi des langages de commandes et de programmation.

'

❖ Noyau Linux

Le noyau Linux (**kernel** en anglais) a été créé en 1991 par Linus Torvalds pour les compatibles PC. En voici une representation.



Noyau linux

❖ Système d'exploitation Linux

Linux est un système d'exploitation libre incluant GNU/Linux

Famille	UNIX
Langues	Anglais pour le noyau, multilingue pour la plupart des distributions GNU/Linux
Type de noyau	Linux
Etat du projet	En développement constant
Plates-formes	Géré par le noyau Linux : x86, x86-64, Itanium, DEC Alpha, ARM, AVR32, Blackfin, ETRAX CRIS, FR-V, H8, M32R, m68k, Microblaze, MIPS, MN103, PA-RISC, PowerPC, s390, S+core, SuperH, SPARC, TILE64, Unicore32, Xtensa (en)
Entreprise / Développeur	-Richard Stallman à l'origine du projet GNU, -Linus Torvalds à l'origine du noyau Linux -Communauté de milliers de programmeurs et d'entreprises
Licence	GNU GPL pour le noyau, licences libres pour le reste
Écrit en	C et assembleur
Première version	17 septembre 1991
Dernière version	5.00 (03/03/2019)
Environnement de bureau	Console pour le noyau, X11 (GNOME, Unity, KDE, Xfce, E16/E17, LXDE, Openbox, Awesome, etc) ou Wayland + Interface en ligne de commande
Gestionnaire de paquets	Dépendant de la distribution : dpkg, APT, Aptitude, Synaptic, apt-rpm, RPM, DNF, portage, Emerge, Pacman, etc.
Site web	-Projet GNU (www.gnu.org) -Noyau Linux (www.kernel.org)

'

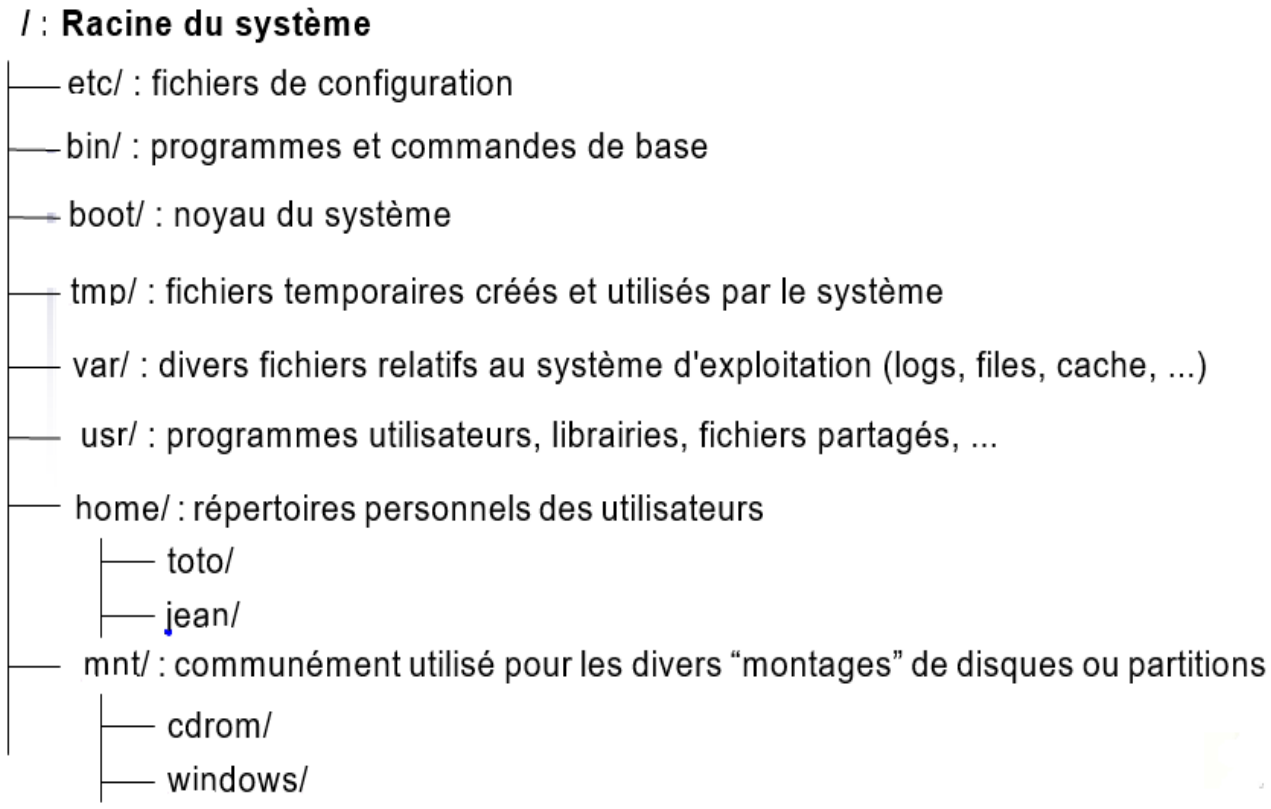
3-Les distributions Linux

Une distribution linux est un noyau linux auquel des logiciels ont été ajoutés. Il y a possibilités de créer des distributions dédiées à un usage particulier. En voici quelques unes ci-dessous citées :

II-Arborescence du système Linux et principaux les répertoires Linux

1-Arborescence du système Linux

La racine est le sommet de la hiérarchie des répertoires Linux. Il s'agit d'une arborescence logique, indépendante de l'implantation physique des divers sous-répertoires, qui peut s'étendre sur plusieurs partitions incluses sur un ou plusieurs disques, et même sur des disques réseaux.
Sa structure est standard, avec des extensions imposées par les distributions. Toute modification est de la compétence exclusive de l'administrateur, à l'exception des répertoires personnels situés dans **/home.** Il est recommandé de respecter cette architecture standard



'

2- Les principaux les répertoires Linux

Ci-dessous enumerés les principaux **répertoires linux et leur rôle :**

- ***/*** : le répertoire racine
- /bin : les fichiers exécutables (en binaire) (initialisation du système + commandes "essentielles")
- /boot : le noyau **vmlinuz** (distribution *testing* intégre le noyau vmlinuz-2.2.20) et les fichiers de démarrage
- /dev : répertoire de fichiers spéciaux, qui servent de canaux de communication avec les périphériques (disques, adaptateur réseau, cartes son etc.…)
- /etc : les fichiers de configuration du système et les principaux scripts de paramétrage
    - /etc/rc.d :*scripts de démarrage du système*
    - /etc/X11 : *scripts de configuration du serveur X*
    - /etc/init.d :*script de controle des serveurs*
    - /etc/cron.d :*description des tâches périodiques à effectuer*
    - /etc/skel : *fichiers recopiés dans le rép. personnel d'un nouvel utilisateur*
- /home : la racine des répertoires personnels des utilisateurs
- /lib : les bibliothèques et les modules du noyau
- /mnt : la racine des points de montage des périphériques. (cd, disquette, nfs ..) (sous Debian : il y a /cdrom et /floppy, placés directement à la racine).
- /opt : lieu d'installation d'applications supplémentaires (comme starOffice, java ..)
- /root : répertoire personnel du super-utilisateur root
- /sbin : les fichiers exécutables pour l'administration du système
- /tmp : stockage des fichiers temporaires
- /usr : programmes accessibles à tout utilisateur; sa structure reproduit celle de la racine /
- /var : données variables liées à la machine (fichiers d'impression, traces de connexions http, smb dans **/var/log**)
- /proc : ce pseudo-répertoire contient une "image" du système (/proc/kcore est l'image de la RAM)

'

Chapitre 2 : Création de VM et installation de Red Hat Linux Entreprise

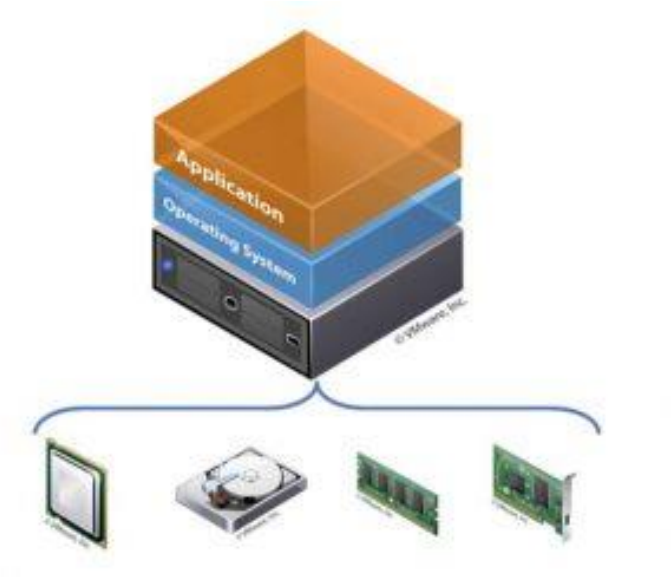
I-Presentation de la plateforme de virtualisation

1-Definition

La virtualisation une technologie qui permet à un ordinateur (physique) d'exécuter en parallèle plusieurs instances de systèmes d'exploitation différents.
Les technologies de virtualisation ont bouleversé totalement le paysage informatique aujourdhui.

2-Principe

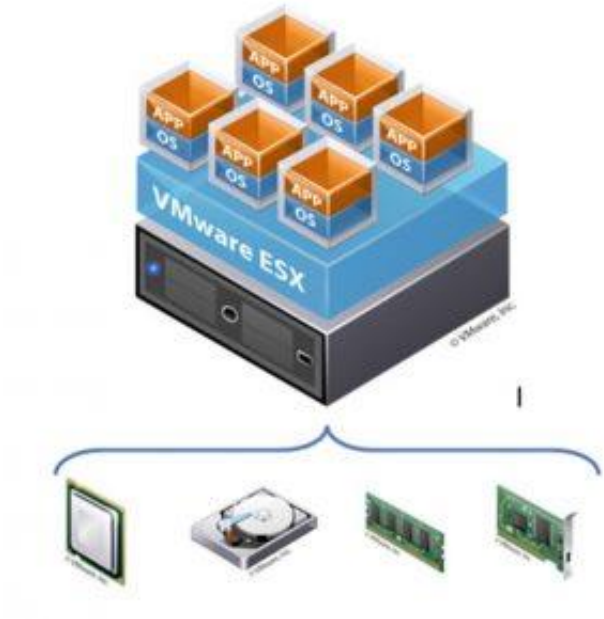
Toute machine physique est architecturée sur la base de deux couches, une matérielle (ensemble des périphériques matériels) et une logicielle (système d'exploitation et applications).



Schématisation d'une architecture classique

Afin de répondre aux multiples contraintes induites par les constructeurs de périphériques informatiques, les technologies de virtualisation ont ajouté une nouvelle couche à l'architecture classique du système d'exploitation. Cette couche est appelée Hyperviseur.
Un hyperviseur se déploie entre la couche matérielle et les différents conteneurs qui lui sont associés en lieu et place d'un véritable environnement. Un conteneur, également appelé machine virtuelle, contient toutes les caractéristiques fonctionnelles d'un système d'exploitation (une instance). La couche de virtualisation simule donc des périphériques virtuels dédiés à chaque machine virtuelle (conteneur). Le rôle principal de la couche de virtualisation est de faire la relation entre les périphériques virtuels et les périphériques physiques (ex: Carte Réseau Ethernet, Lecteur/Graveur CDROM, Carte Graphique etc.)

En conséquence, un hyperviseur est capable de faire tourner simultanément plusieurs systèmes d'exploitation (Windows, Linux, Unix…), autour d'un jeu de pilotes génériques préalablement spécifiés pour chaque machine virtuelle. De ce fait, les technologies de virtualisation permettent, quelle que soit la couche matérielle sur laquelle l'hyperviseur est installé, de déployer des machines virtuelles qui fonctionneront sans avoir besoin du pilote constructeur.



Schématisation d'une architecture virtuelle

3-Bénéfices

Les bénéfices apportés par l'utilisation des technologies de virtualisation sont nombreux, nous ciblerons ici les plus intéressants pour un utilisateur standard. Elles permettent de :

-Faire cohabiter simultanément différents systèmes d'exploitation et environnements applicatifs sur une même machine, sans avoir besoin notamment d'effectuer de nouveaux partitionnements répétitifs ou bien de faire un "dualboot" ou pire de tout reformater ;
-Installer et faire fonctionner un ou plusieurs programmes hors de l'environnement matériel et logiciel habituel (ordinateur physique / Système d'exploitation) ;
-Tester et simuler différentes applications en protégeant l'environnement habituel de travail (machine physique) de toute pollution ;
-Cloisonner et confiner toutes les menaces virales potentielles, mettant ainsi la machine physique et les données à l'abri de toute infection !
-Revenir rapidement à un état antérieur de l'OS et de la machine virtuelle associée (Snapshot)
-Cloner facilement une machine virtuelle
-Optimiser les ressources matérielles de la machine physique
-Consolider plus d'applications sur moins de matériels

4-Éditeur de solution de virtualisation

Quelques exemples d'éditeur : Hyper-V, KVM, QEMU, VirtualBox, VMWARE, VirtualPC, Xen et Bochs

L'éditeur VMWARE avec sa solution VMWARE WORKSTATION (payante) propose tout de même des alternatives gratuites qui sont VMWare vSphere Hypervisor et VMware Player Free Edition.

'

II- Installation et configuration de VMWARE WORKSTATION 12 PRO

1- Pré-requis d'installation

Processeurs compatibles (machine physique):Intel: Celeron, Pentium II, Pentium III, Pentium 4, Pentium M (incluant les ordinateurs avec la technologie mobile Centrino), Xeon (incluant "Prestonia")
AMD: Athlon, Athlon MP, Athlon XP, Duron, Opteron. Cadencé minimum à 1,3 G.Htz ou plus rapide.
Mémoire vive (RAM) installée sur l'ordinateur hôte (machine physique) : Minimum 2 Go. Il est recommandé d'avoir au moins 4 Go de mémoire vive ou plus.
Systèmes d'exploitation(machine physique): Windows 10, Windows 8, Windows 7 Windows Vista, Windows XP SP3, Windows Server 2012, Windows Server 2008, Windows Server 2003 Standard, Windows XP, Édition Familiale SP2, Ubuntu 8.04 et versions ultérieures, Red Hat Enterprise Linux 4.5 et versions ultérieures, CentOS 5.0 et versions ultérieures, Oracle Linux 5.0 et versions ultérieures, openSUSE 10.2 et versions ultérieures, SUSE Linux 10 et versions ultérieures…
Disque dur physique (machine physique)**:** IDE, SATA et SCSI de1,2 Go minimum d'espace
Lecteur/Graveur optique CDROM et DVD: IDE, SCSI, image ISO

2-Configuration du Bios

C'est à partir du menu **Bios** qu'on active la technologie de virtualisation du processeur. Une fois dans le **Bios**, à l'aide des touches de direction du clavier, on sélectionne l'onglet intitulé "**Advanced**" (ou "Advanced Features" ou "Advanced Bios Features"), puis l'onglet "**CPU**" (ou "CPU Features").
On obtient un menu à choix pour pouvoir activer l'option **"Intel VT-d" ou "AMD-V" du style "enabled" (activé) ou "desabled" (désactivé).**
Il se peut que la langue utilisée par un "Bios" récent soit en français, vous aurez donc comme choix: **"Activer" ou "Désactiver".**
Lorsque vous aurez sélectionné **"enabled" (activé)** sur le champ correspondant à la technologie de virtualisation nommée **"Intel VT-d" ou "AMD-V",** vous appuyez sur la touche de fonction "**F10**" du clavier, vous obtiendrez un message, sous forme de menu vous proposant simultanément de **sauvegarder (Save) et quitter (Exit)** le **Bios**, ce qui aura pour effet de redémarrer l'ordinateur.

3- Installation de VMWARE WORKSTATION 12 PRO

Faire un clic droit avec la souris sur le fichier exécutable VMWARE WORKSTATION 12 PRO (VMware Workstation Pro 12.1.1 Build 3770994 + serial), et sélectionner "**Exécuter en tant qu'administrateur**". Faire "Suivant", "Suivant"… et validez par défaut toutes les étapes d'installation affichées par l'installateur.

4 - Configuration de l'interface réseau VMWARE

Abordons quelques notions simples de configuration réseau à partir de la barre d'outils "**VMWARE**". Sous "**VMWARE WORKSTATION**".Il existe quatre types de configuration de l'interface réseau:

'

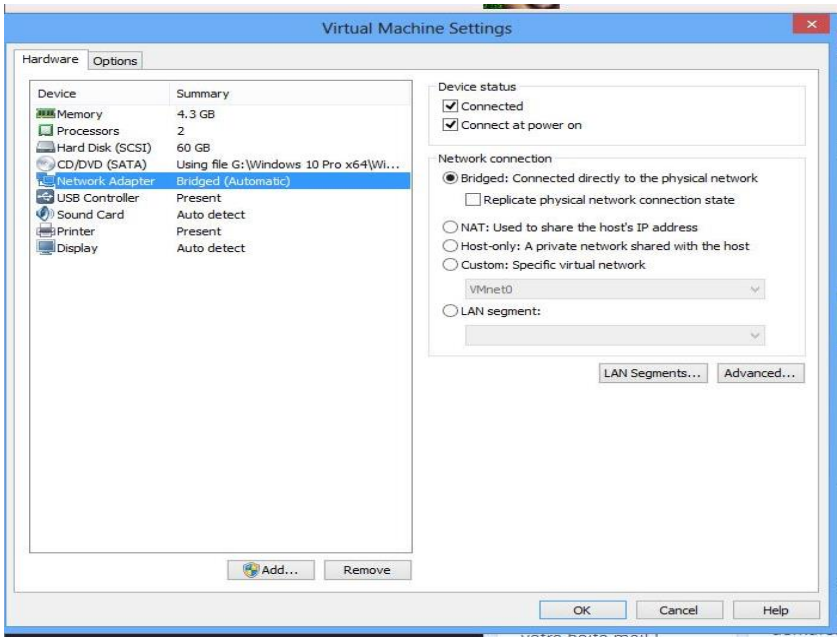
a-« **Bridged** »: la machine virtuelle aura sa propre adresse IP interne délivrée par le réseau physique.

b-« **HostOnly** »: la machine virtuelle communiquera à l'aide de l'IP de la machine physique.

c-« **NAT** »: les machines virtuelles communiquent entre elles et la machine physique.

d-« **Custom »:** carte réseau virtuelle spécifique.

Dans notre cas, nous opterons pour une configuration "**Bridged**" de l'interface réseau. Pour ce faire, cliquer sur le bouton "**VM**" situé sur la barre d'outils de "VMWARE WORKSTATION" puis sélectionner l'onglet "**Settings**" suivi de "**Network Adaptater**", et cocher "**Bridged**" .



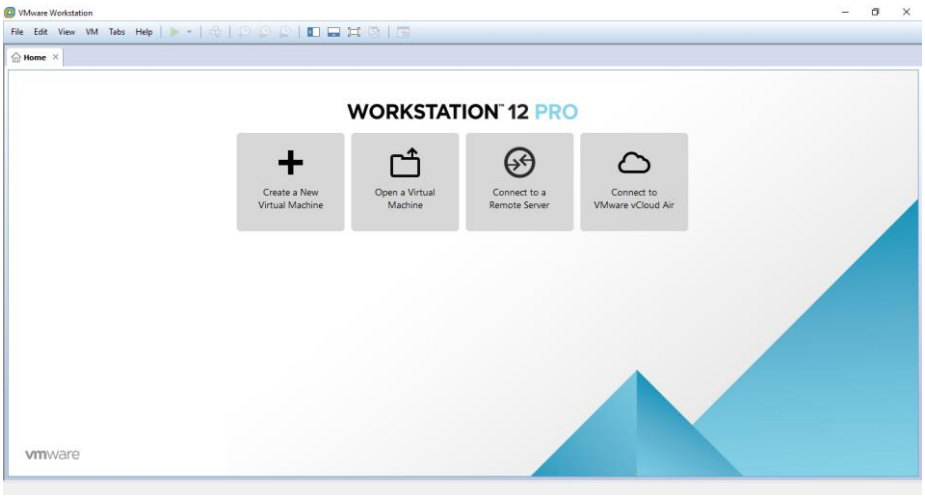
Il est recommandé de toujours configurer l'interface réseau sous "Vmware Worksation" avant de lancer l'installation d'une machine virtuelle (Windows, Linux, Unix…).

III- Installation et Configuration de Red Hat Entreprise Linux 6 dans VMWARE WORKSTATION 12

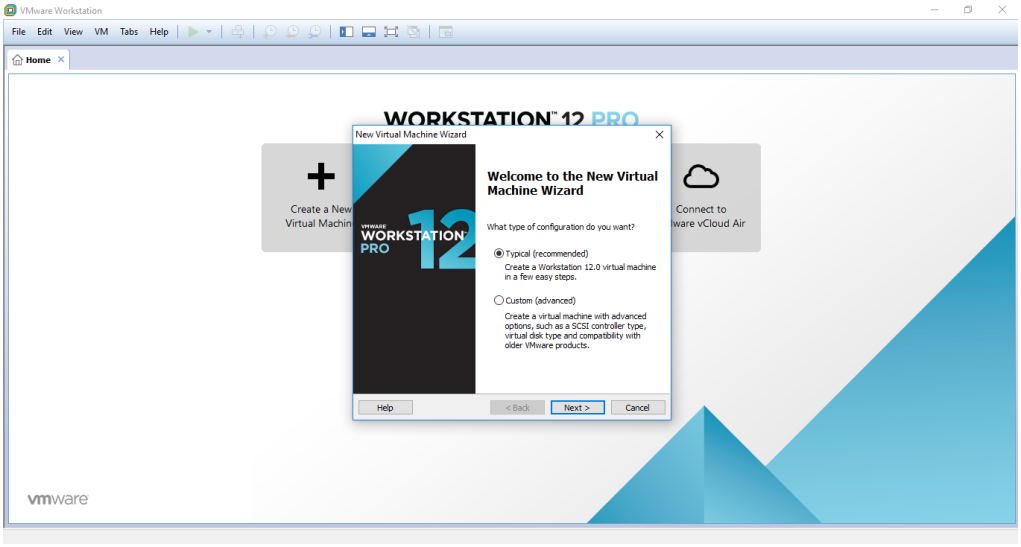
1-Installation

Etape 1 - Disposer de l'image ISO de Redhat Linux 6
Etape 2 - Localiser le fichier dans un dossier sur notre bureau par exemple
Etape 3 - Ouvrir VMWare Workstation à partir du menu Démarrer de Windows ou de votre bureau si vous avez l'icône VMware Workstation.
Étape 4 - Lancez l'assistant d'installation de la nouvelle machine virtuelle VMware Workstation. *Cliquez sur Créer une nouvelle machine virtuelle ou un nouveau fichier -> Nouvelle machine virtuelle.*

'

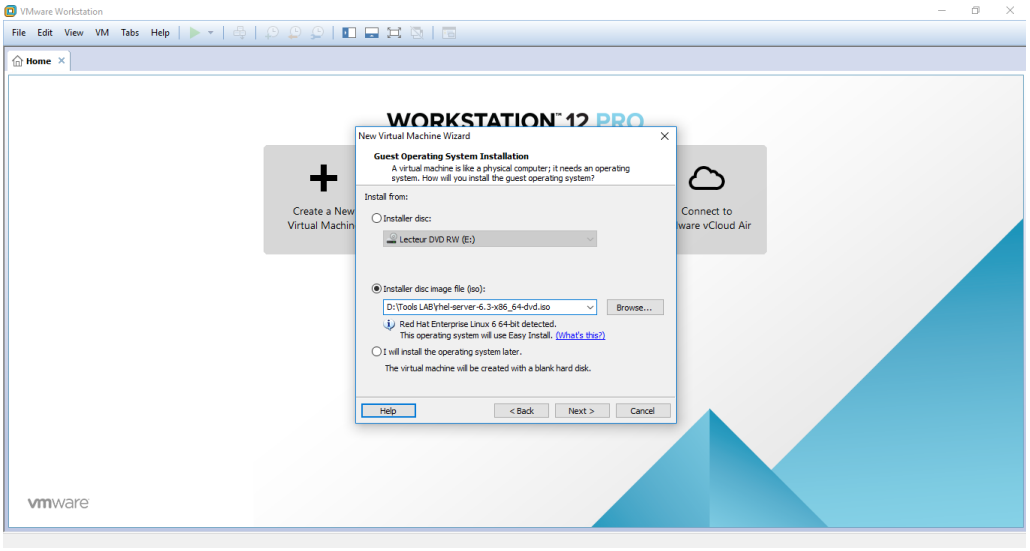


Étape 5 - Bienvenue dans la nouvelle boîte de dialogue Assistant de la machine virtuelle s'affiche.
C'est ici que vous avez la possibilité de choisir la façon dont la machine virtuelle sera créée. **Typique** est prédéfini ou surtout les valeurs par défaut. **Custom** est l'endroit où vous pouvez définir des options avancées telles que la compatibilité avec les anciennes machines virtuelles, le type de contrôleur SCSI, etc. Nous choisirons les options par défaut. Sélectionnez Typique et cliquez sur Suivant. Bienvenue dans la nouvelle boîte de dialogue Assistant de la machine virtuelle qui s'ouvre. Sélectionnez typique et cliquez sur Suivant.



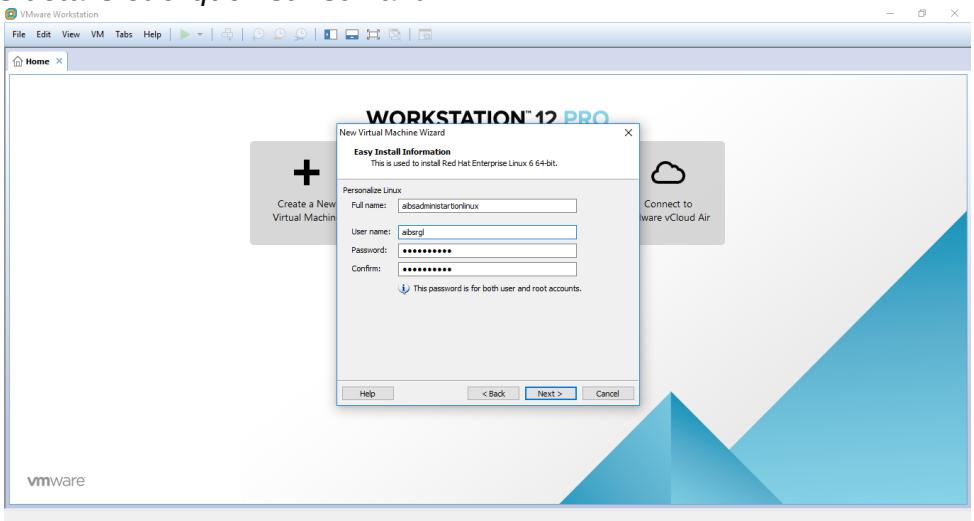
Étape 6- Sélectionner le support d'installation ou la source
Dans cette boîte de dialogue, vous devrez naviguer jusqu'au fichier ISO et cliquer sur Suivant.

'



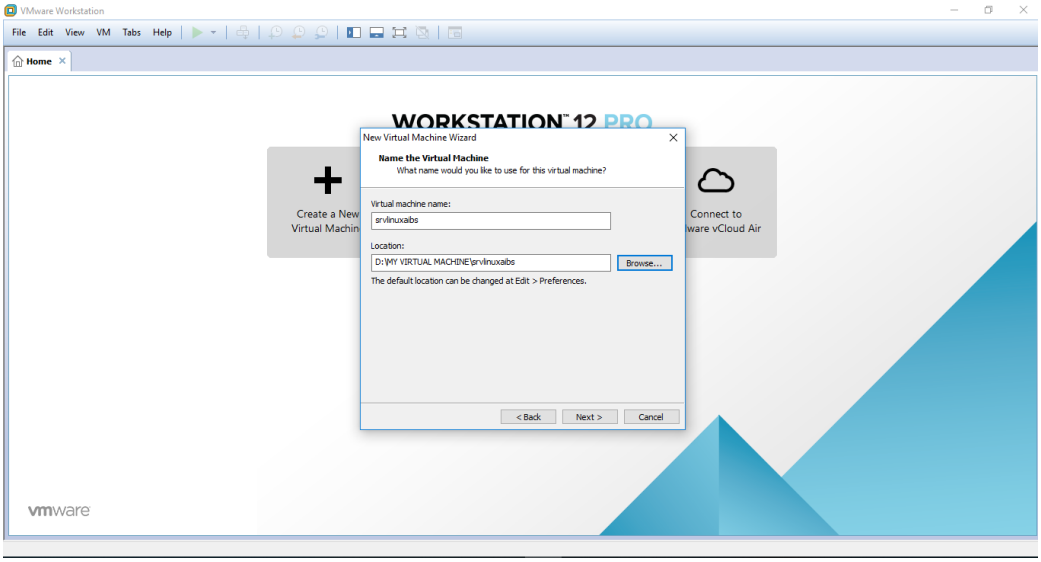
Étape 7 - Information sur l'installation facile
Dans cette boîte de dialogue, il vous sera demandé d'entrer le nom complet (**aibsadministrationlinux*)***, le nom d'utilisateur (**aibsrgl**) et le mot de passe(**1234567890**). Rappelez-vous le mot de passe, il sera utilisé pour vous connecter à Red Hat Linux une fois l'installation terminée. Ce mot de passe est également pour le **root**

Entrez les détails et cliquez sur suivant.



Étape 8- Entrez le nom de la machine virtuelle
Dans cette boîte de dialogue, il vous sera demandé de fournir le nom de la machine virtuelle (**srvlinuxaibs**). Vous pouvez fournir votre propre nom (***yao***).Vous pouvez également changer l'emplacement de la machine virtuelle en creant un dossier sur votre bureau pour vos VM. Par défaut, il est placé dans le dossier Documents/Virtual Machine. Ici nous choisisssons notre dossier personnalisé (faites en de même).

'



Etape 9- Spécifier la capacité du disque
Cette boîte de dialogue vous demande de spécifier la capacité du disque. C'est la quantité maximale d'espace disque qu'il utilisera une fois la machine virtuelle créée. L'assistant vous invite à définir la taille du disque virtuel et à spécifier s'il faut diviser le disque en plusieurs fichiers disque virtuel (.vmdk).

Un disque virtuel est constitué d'un ou plusieurs fichiers de disque virtuel. Les fichiers de disque virtuel stockent le contenu du disque dur de la machine virtuelle. Presque tout le contenu du fichier est constitué de données de la machine virtuelle. Une petite partie du fichier est allouée aux frais généraux de la machine virtuelle.

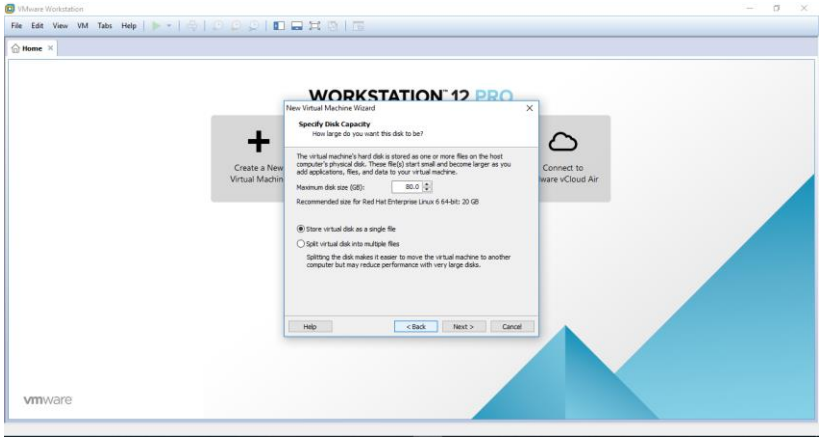
Sélectionnez Diviser le disque virtuel en plusieurs fichiers si le disque virtuel est stocké sur un système de fichiers dont la taille est limitée.

Lorsque vous divisez un disque virtuel de moins de 950 Go, une série de fichiers de disque virtuel de 2 Go sont créés. Lorsque vous fractionnez un disque virtuel de plus de 950 Go, deux fichiers de disque virtuel sont créés. La taille maximale du premier fichier de disque virtuel est de 1,9 To et le deuxième fichier de disque virtuel stocke le reste des données.

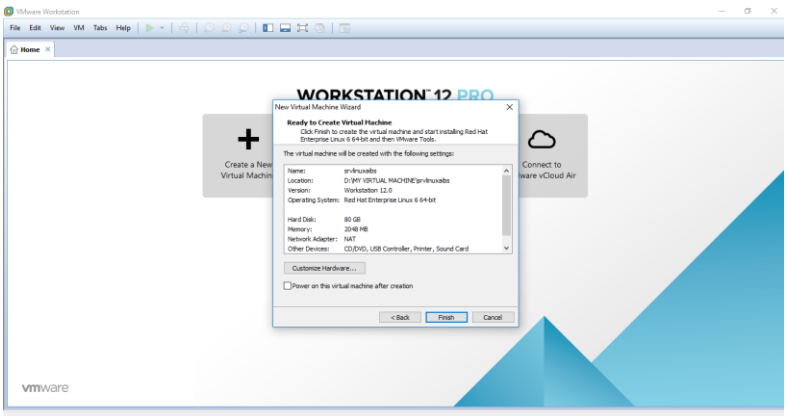
Pour les configurations personnalisées, vous pouvez sélectionner Attribuer tout l'espace disque maintenant pour allouer tout l'espace disque immédiatement plutôt que de permettre à l'espace disque d'augmenter progressivement jusqu'au maximum. L'allocation immédiate de tout l'espace disque peut fournir de meilleures performances, mais c'est une opération qui prend beaucoup de temps et qui nécessite autant d'espace disque physique que vous le spécifiez pour le disque virtuel.
Après avoir créé une machine virtuelle, vous pouvez modifier les paramètres du disque virtuel et ajouter des disques virtuels supplémentaires.

Attribuez 80 Go d'espace et Cochez la case **store virtual disk as a single file** (stocker le disque virtuel dans un seul fichier) et faire suivant

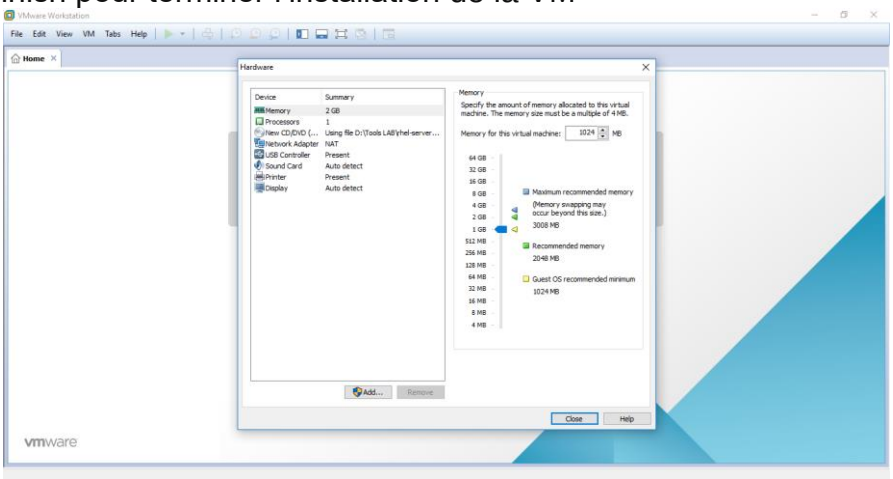
'



Étape 10- Boîte de dialogue Prêt à créer une machine virtuelle
Il s'agit de la boîte de dialogue finale et vous voyez toutes les options que vous avez sélectionnées dans les boîtes de dialogue précédentes.



Si vous avez suffisamment de RAM et de CPU sur votre machine (physique)hôte Windows, vous pouvez modifier la RAM à et le CPU en cliquant sur **customize hadware** (personaliser hadware),**.**
Cliquez sur finish pour terminer l'installation de la VM



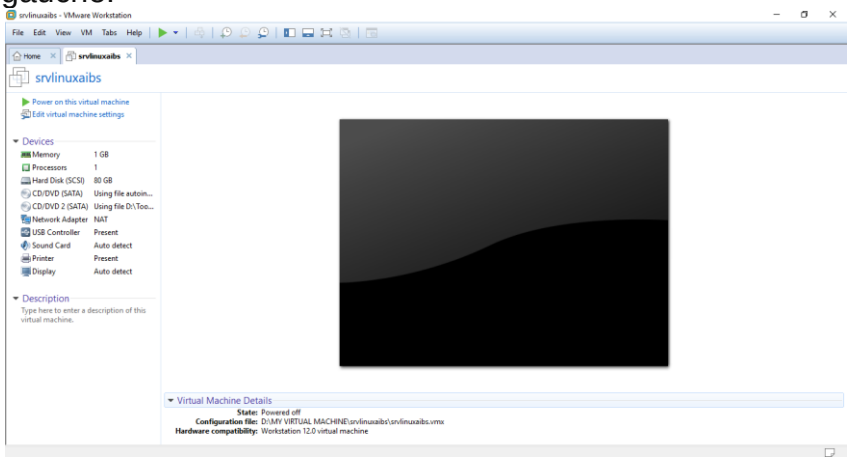
Faire **close,** puis finish pour terminer l'installation de la VM

Decochez l'option Power on this virtual machine after creation (Mise sous tension de cette machine virtuelle après la création). Par défaut, elle est cochée. Ceci créera automatiquement le démarrage de la VM.

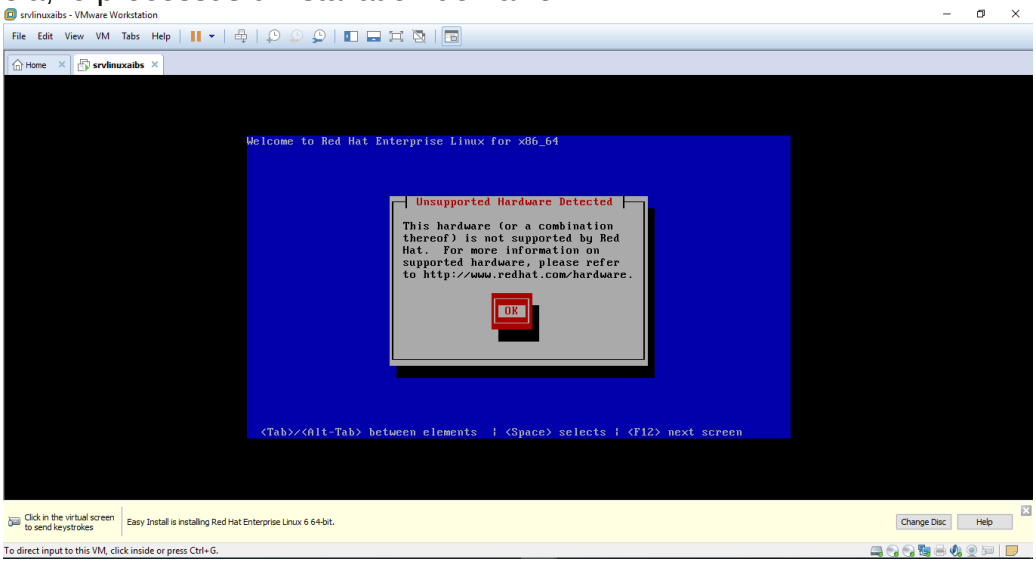
Étape 11 - Mise sous tension de la machine virtuelle

'

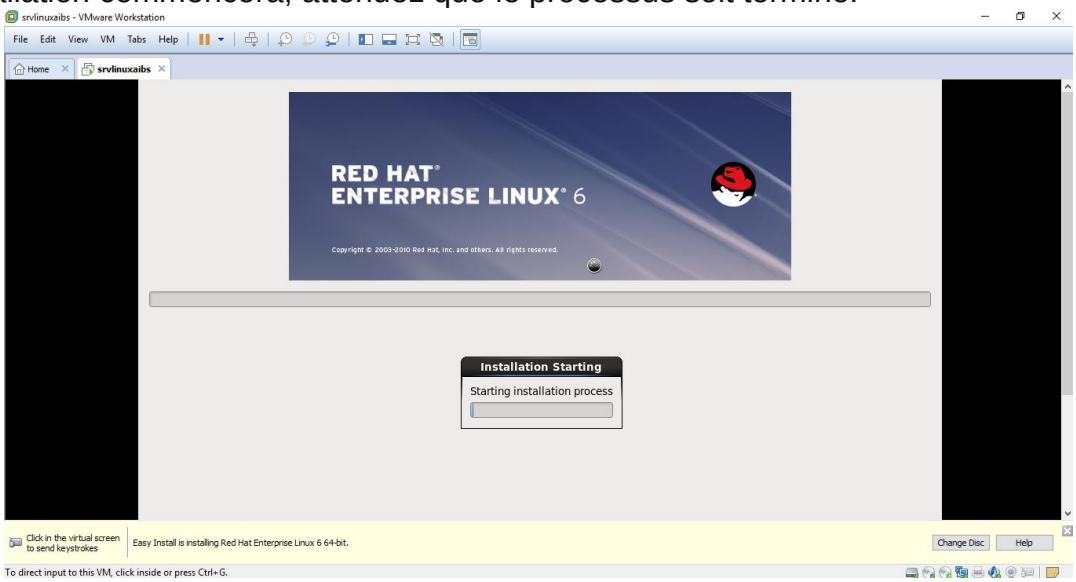
Démarrez manuellement la VM. Vous pouvez voir l'option de mise sous tension de la VM en haut à gauche.



Étape 12 - Début de l'installation de Red hat Enterprise Linux 6
Après le démarrage de la VM, vous verrez Red Hat Enterprise Linux 6 démarrer.
Après cela, le processus d'installation démarre.

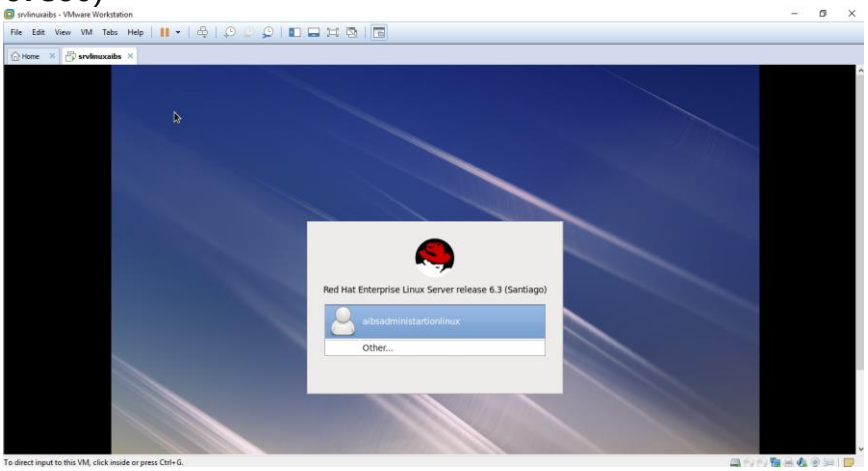


Faire OK (entrer)
L'installation commencera, attendez que le processus soit terminé.

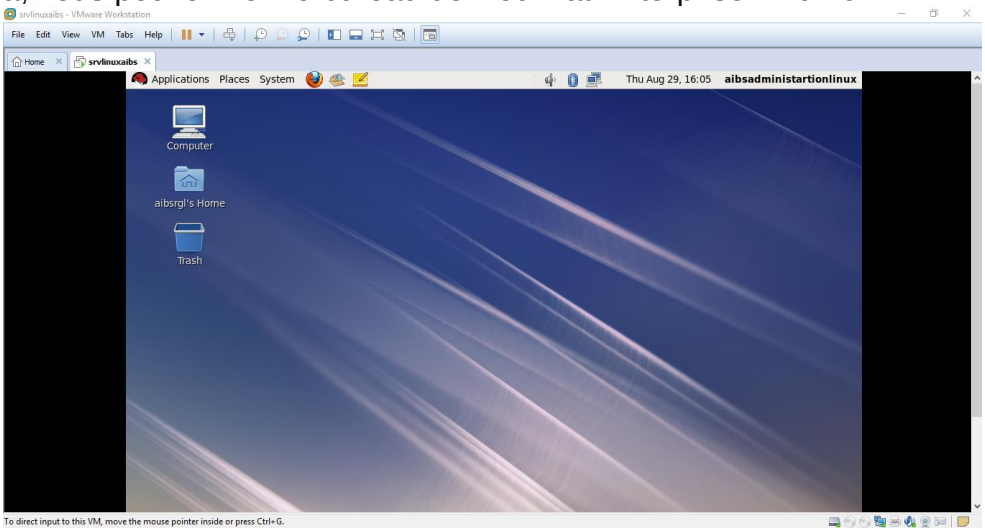


'

Une fois l'installation terminée, vous voyez l'écran de connexion Red Hat Enterprise Linux 6 avec votre nom d'utilisateur entré dans les étapes précédentes.
Cliquez sur le nom d'utilisateur complet **(aibsadministrationlinux)** et entrez le mot de passe (**1234567890**)

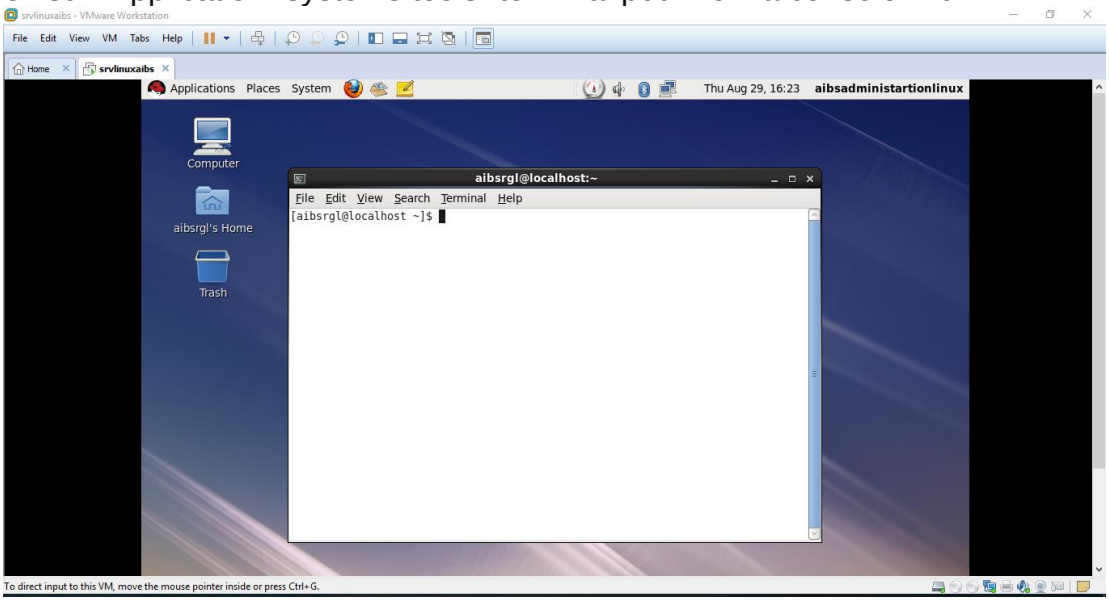


Après cela, vous pouvez voir le bureau de Red Hat Enterprise Linux 6.



2-Decouvrir la console linux

Cliquez sur Application- systems tools- terminal pour voir la console linux



'

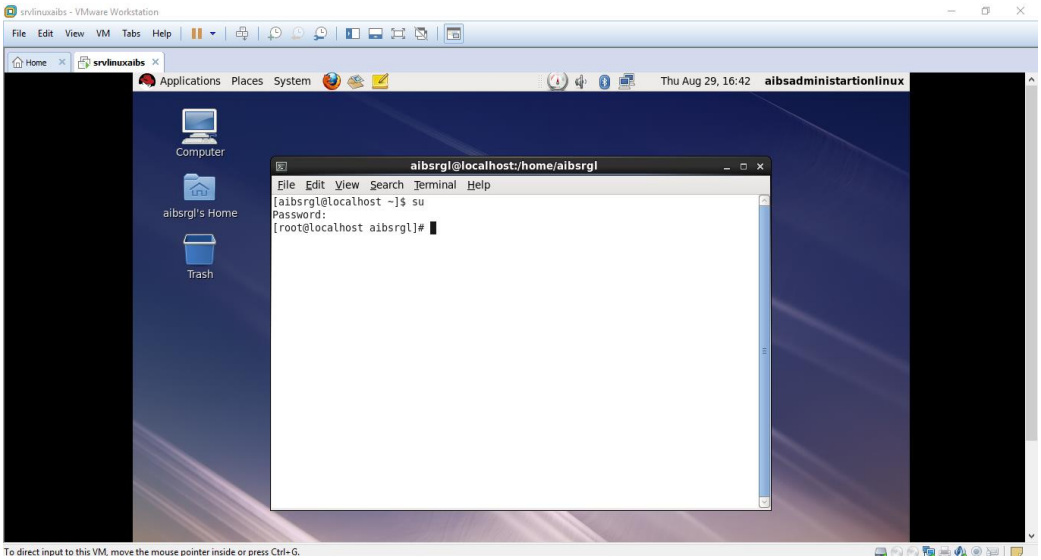
aibsrgl@localhost~**$** signifie que le compte utiliser est le compte *user* **aibsrgl** créer a l'installation de RHEL 6.
Deux types de comptes sont à distinguer.
Les comptes utilisateurs, créés pour chaque *user* présent sur le système.
Lors d'un accès en utilisateur classique *(user),* l'invite de commande est généralement composé du nom de l'utilisateur courant, du nom de la machine et se termine par le symbole **$**

Ex : aibsrgl@localhost~**$**

Le compte *root*, l'administrateur du système qui a tous les droits également appelé *super-utilisateur*. Lors d'un accès avec le compte *root* l'invite de commande est généralement composé de nom de l'utilisateur courant, du nom de la machine et se termine par le symbole #

Ex : root@localhost~#

Le mode *root* donne plus de privilèges que le mode *user*, comme etteindre la machine linux par exemple.
Pour se passer en root, taper la commede
Su et entrer le **mot de passe root (1234567890)**, rien se s'affiche mais saisir tout de même puis, taper entrer



Su et entrer le **mot de passe root (1234567890)**, rien se s'affiche mais saisir tout de même puis, taper entrer Eteindre sa machine linux en saisissant la commande
int 0.
La commande init permet de stopper le système : # Arrêt : init 0 ; # Reboot : init 6

'

Partie 2: Manipuler la console et les fichiers

Chapitre 3 : Les systèmes de fichiers sous Linux

1- Les systèmes de fichiers

Un système de fichiers est la méthode utilisée pour stocker de l'information sur un disque dur. Des systèmes d'exploitation différents utiliseront des systèmes de fichiers différents, rendant ainsi difficile le partage de l'information.
Toutefois Linux supporte plusieurs systèmes de fichiers rendant, par exemple, possible le partage d'information avec une partition dédiée à Windows.
Il existe de nombreux systèmes de fichiers sous linux : ext2, ext3, ext4,...

****Les qualifications d'un système de fichiers**

De nombreux attributs differents sont nécessaires afin de définir un système de fichiers. Ils incluent la taille maximale que peut avoir un fichier dans ce système de fichiers, la taille maximale d'une partition et la journalisation ou non du système de fichiers ainsi que la gestion ou non des droits d'accès aux fichiers et repertoires.

****Comparaison de systèmes de fichiers**

Nom du syst De fichiers	Taille max d'un fichier	Taille max d'une partition	Journalisé ou non	Gestion des droits d'accès
FAT	2 GB	2GB	Non	Non
FAT32	4 GB	4GB	Non	Non
NTFS	Limitee par la taille de la partition	2 TO	Oui	Non
ext2fs	2 To	4 To	Non	Oui
ext3fs	2 To	4 To	Oui	Oui
Ext4fs	16 To	1 Eo	Oui	Oui

2-Les types de fichiers sous Linux (répertoire, ordinaire, spéciaux)

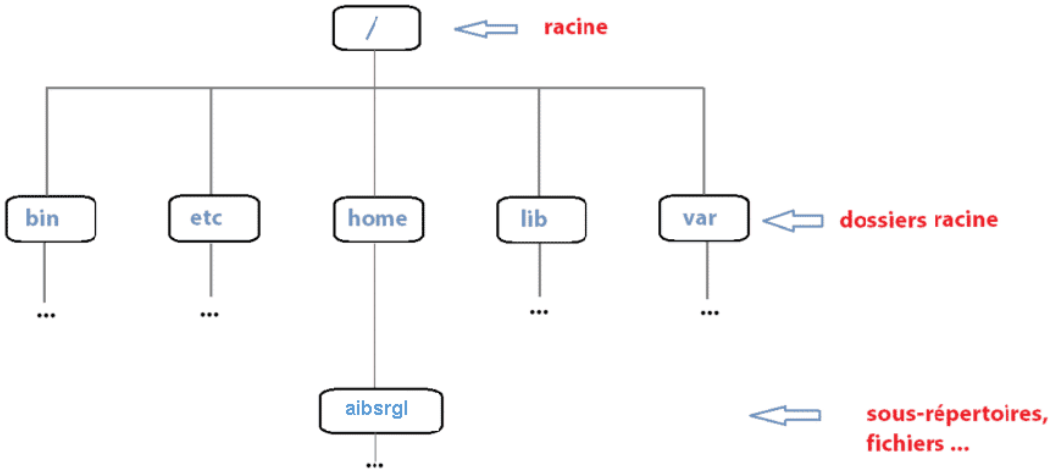
Il existe differentes catégories de fichiers sous Linux. Il s'agit notament de :
-**Fichiers normaux** : symbole (**-**)
*texte : courrier, sources des programmes, scripts, configuration ...
*exécutables : programmes en code binaire.
-**Répertoires ou Fichiers répertoires :** symbole (**d**): ce sont des fichiers conteneurs qui contiennent des références à d'autres fichiers. Véritable charpente de l'arborescence, ils permettent d'organiser les fichiers par catégories.
-**Fichiers spéciaux** : situés dans **/dev**, ce sont les points d'accès préparés par le système aux périphériques. Le montage va réaliser une correspondance de ces fichiers spéciaux vers leur répertoire "point de montage".
Par exemple, le fichier /dev/hda permet l'accès et le chargement du 1er disque IDE.
-**Liens symboliques** ou **Fichiers liens symboliques** : Ce sont des fichiers qui ne contiennent qu'une référence (un pointeur) à un autre fichier.Cela permet d'utiliser un même fichier sous plusieurs noms sans avoir à le dupliquer sur le disque.
Etc…

'

La plupart des types de fichiers sont catalogués et identifiés par un numéro magique dans "**/usr/share/magic** ". *

3-La structure des dossiers et fichiers sous Linux

Sous Linux, tous les dossiers sont centralisés sur une base que l'on appelle « **racine** » et qui est symbolisé par un *slash*, soit ce symbole « **/** ».
En réalité, dans tous les systèmes de fichiers il y a une racine, le gros dossier de base qui contient tous les répertoires, sous-repertoires et fichiers…
Mais à la grande différence de Windows qui organise les choses sur plusieurs racines, en classant le disque dur principal dans la racine **c:**, le lecteur usb sur **e:**, un disque dur secondaire sur **z:** …
En clair, toutes les racines sous Windows (c:\, d:\, e:\ …) sont équivalentes à l'unique et seul racine slash, soit « **/** » sous linux.
Cette racine « **/** » sous Linux va contenir tous les dossiers racines, qui eux-mêmes vont contenir dossiers et fichiers …



Structure des dossiers Linux qui représente qu'une partie des dossiers pour illustrer l'exemple d'arborescence.

**Sur Windows pour me rendre dans mon dossier personnel, le chemin étant: c:\utilisateurs\aibsrgl
**Sur Linux, et en se référant sur la photo ci-dessus, cela sera: /home/aibsrgl

Le symbole slash indique la racine mais il sert aussi de séparateur entre les dossiers et fichiers sous linux. C'est pour cela qu'entre **'home'**et **'aibsrgl'** il y a un slash.
Attention de ne pas confondre le slash ' / ' sous Linux et l'antislash ' \ ' (backslash) sous Windows.

****Les dossiers sous la racine**
Les dossiers justes après la racine, plus communément appellés **dossiers de la racine**, sont destinés à un processus, on a en occurence:

-/bin (binaires): contient les Exécutables, programmes utilisiés par tous les utilisateurs.
-/boot *(initialisation):* contient les fichiers permettant le démarrage du système.
-/dev *(périphérique):*contient les fichiers spéciaux des périphériques.

'

-/etc *(configuration):* contient les fichiers de configuration.
-/home : contient les données propres à chaque utilisateur.
-/lib (*bibliothèques*): contient des bibliothèques partagées essentielles au système
-/media: contient les fichiers de montage pour les périphériques amovibles (clés usb, disque dur…).
-/mnt *(montage*): identique à /média, mais a un usage plus temporaire.
-/opt *(optionnel):* contient les emplacements pour les applications installées hors gestionnaire de paquets.
-/proc (*processus*):contient les informations du système.
-/root: Répertoire du super-utilisateur.
-**/run** (*exécution système*): contient les informations des processus en cours.
-/sbin (*binaires système*): contoient les exécutables du système 'very important'!!!
-/srv (*services*): contient les données pour les services du système.
-/tmp (*temporaire*): contient les fichiers temporaires.
-/usr (*ressources système*): gros dossier comprenant les programmes installés par l'utilisateur.
-/var (*variable*): contient des données variables.

Inutile de tous les retenir, reviendrons très souvent comme *média, var, etc* …

3-Les utilisateurs et les droits

****Droits.**
Les systèmes d'exploitation inspirés d'Unix dont Linux possèdent la capacité de définir de façon poussée la gestion de droits d'accès aux divers fichiers de votre Système d'exploitation.

Les droits d'accès définissent la possession d'un fichier ou d'un répertoire à un utilisateur et à un groupe d'utilisateurs.

Ils gèrent aussi quelles actions les utilisateurs ont le droit d'effectuer sur les fichiers, selon qu'ils sont propriétaire du fichier, membre du groupe propriétaire du fichier ou ni l'un ni l'autre.

Chaque fichier ou répertoire est doté d'une structure appelé i-node où sont stockées les droits sous la forme de trois bits d'accès (lecture **r (read)**, écriture **w (write)** et exécution **x (execute)**) pour chacune des catégories d'utilisateurs

Il y a trois types de droits d'accès:
-read (r) pour l'accès en lecture au fichier (permet l'impression, l'affichage et la copie d'un fichier, et permet la traversée du répertoire ou l'affichage des fichiers d'un répertoire)
-write (w) pour l'accès en écriture au fichier (permet la modification d'un fichier, et permet l'effacement d'un fichier ou l'enregistrement d'un fichier dans un répertoire)
-execute (x) pour la possibilité d'exécuter le fichier (permet l'execution d'un programme, d'un executable, et permet d'accéder aux informations de gestions des fichiers du répertoire, comme l'inode, la table des droits,...).

Ces droits n'ont, toutefois, pas la même signification lorsqu'ils s'appliquent à un fichier ou à un répertoire

'

	pour un fichier	pour un répertoire
read	lecture autorisée	Droit de lire le contenu, d'afficher la liste des fichiers (avec ls)
write	modification autorisée	Modification du contenu : droits d'ajouter et desupprimer des fichiers (avec cp, mv, rm)
execute	exécutable	Droit d'accéder aux fichiers du répertoire et de se déplacer à l'intérieur (avec cd)

***Note** : si on attribue **w** sur un répertoire, il faut lui attribuer aussi **x**.*

****Utilisateurs.**

Pour chaque fichier, les droits d'accès sont fixés pour trois catégories d'utilisateurs:
-user (u) : le propriétaire du fichier
-group (g) : le groupe propriétaire du fichier
-other (o) : les autres utilisateurs

Les droits se combinent en 3 blocs rwx. Le système de droits est spécifié symboliquement par 9 attributs, correspondants aux 3 catégories d'utilisateurs du fichier.
...|...|...
u g o
En cas d'absence de droit la lettre est remplacée par un tiret.

****La notation octale.**

En octal, chaque « groupement » de droits (pour user, group et other) sera représenté par un chiffre et à chaque droit correspond une valeur :

- r (read) = 4
- w (write) = 2
- x (execute) = 1
- - = 0

Par exemple,

- Pour **rwx**, on aura : 4+2+1 = 7
- Pour **rw-**, on aura : 4+2+0 = 6
- Pour **r--**, on aura : 4+0+0 = 4

Ce qui permet de faire toute les combinaisons :

- 0 : **- - -** (aucun droit)
- 1 : **- - x** (exécution)
- 2 : **- w -** (écriture)
- 3 : **- w x** (écriture et exécution)
- 4 : **r - -** (lecture seule)
- 5 : **r - x** (lecture et exécution)
- 6 : **r w -** (lecture et écriture)
- 7 : **r w x** (lecture, écriture et exécution)

'

Par exemple, la combinaison de tous les droits cumulés pour les trois types d'utilisateurs (rwx rwx rwx) est équivalente à la valeur octale 777. Et 700 donne tous les droits au propriétaire de fichier

Petit recap : par convention la présence d'un droit est noté «1», l'absence «0». On a la correspondance suivante :

Binaire	**Droit**	**Octal**
000	(---)	0
001	(--x)	1
010	(-w-)	2
011	(-wx)	3
100	(r--)	4
101	(r-x)	5
110	(rw-)	6
111	(rwx)	7

****Droits spéciaux SUID et SGID** (*nous ne ferons pas d'exemple sur ce cas ici*)

Ils remplacent le droit d'executer (x) pour un utilisateur ou un groupe, et permettent à l'utilisateur ou au groupe de devenir le propriétaire d'un fichier executable le temps de son execution.
Ainsi, le fichier "/usr/bin/passwd" est un fichier executable qui appartient à root, mais le droit (s) permet à tous les utilisateurs de lancer ce programme avec le droit root le temps de son execution, et ainsi de pouvoir changer eux même leur propre mot de passe.

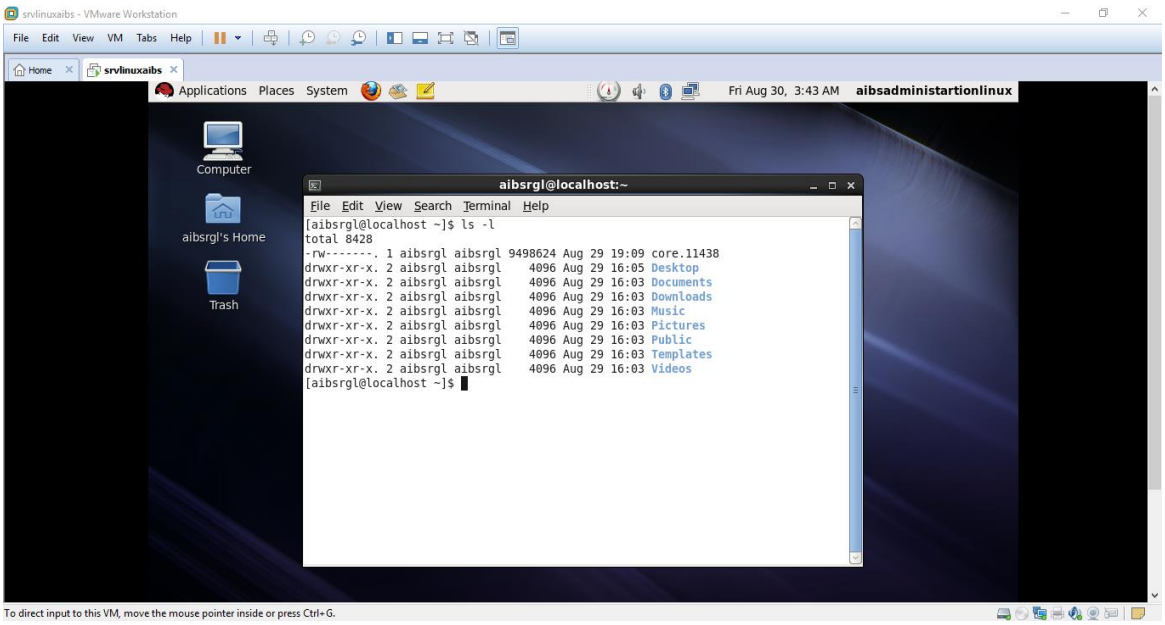
****Lire et attribuer des droits**

Lire ou afficher des droits

-Commande ls –l

Les droits sont consultables par la commande **«ls -l».**
En plus des droits cette commande affiche egalement un autre caractère : "-" pour un fichier, "d" pour un répertoire, "l" pour un lien symbolique.

'

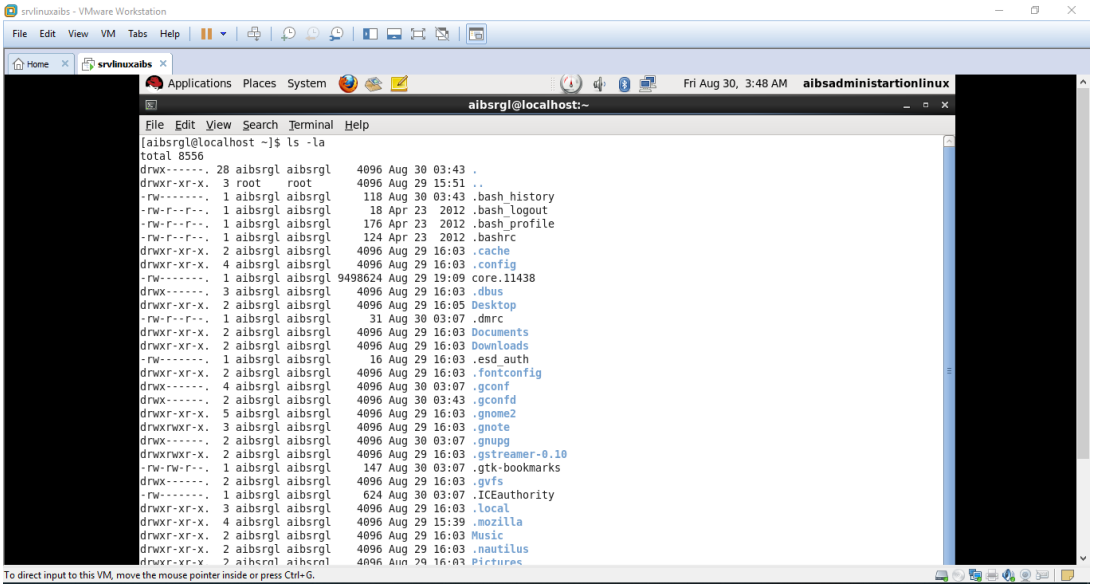


drwxr-xr-x se traduit de la manière suivante :

- **d** : c'est un répertoire.
- **rwx** pour le 1er groupe de 3 symboles : son propriétaire **(u)** peut lire, écrire et exécuter.
- **r-x** pour le 2nd groupe de 3 symboles : le groupe **(g)** peut uniquement lire et exécuter le fichier, sans pouvoir le modifier.
- **r-x** pour le 3ème groupe de 3 symboles : le reste du monde **(o)** peut uniquement lire et exécuter le fichier, sans pouvoir le modifier.

-Commande ls –la

La commande **«ls -la»** permet d'afficher les droits de tous les fichiers du répertoire courant linux.



'

***Attribuer ou modifier des droits**

La commande **«chmod »** (***ch****ange* ***mod****e*, changer les permissions) permet de modifier les permissions sur un fichier. Il peut s'employer de deux façons :

+soit en ajoutant ou en retirant des permissions à une ou plusieurs catégories d'utilisateurs à l'aide des symboles r w et x comme ce qui suit :

Par exemple :
chmod o-w fichier3 :enlèvera le droit d'écriture pour les autres.
chmod a+x : ajoutera le droit d'exécution à tout le monde.
On peut aussi combiner plusieurs actions en même temps :
chmod u+rwx,g+rx-w,o+r-wx fichier3 : On ajoute la permission de lecture, d'écriture et d'exécution sur le fichier fichier3 pour le propriétaire ; On ajoute la permission de lecture et d'exécution au groupe propriétaire, on retire la permission d'écriture ;On ajoute la permission de lecture aux autres, on retire la permission d'écriture et d'exécution.

+soit en précisant les permissions de manière octale, à l'aide de chiffres, exemple
chmod 755

Prenons un répertoire les permissions sont :
drwxr-x---
En octal, on aura **750** :

rwx	**r-x**	**---**
7(4+2+1)	**5(4+0+1)**	**0(0+0+0)**

Pour mettre ces permissions sur le répertoire on taperait donc la commande :
chmod 750 aibsfile1

Sachez seulement que les deux méthodes sont équivalentes, c'est-à-dire qu'elles affectent toutes deux les permissions de la même manière.

-La commande **«chown »** (***ch****ange* ***own****er*, changer le propriétaire) permet de changer le propriétaire du fichier. Seuls le super-utilisateur ou le propriétaire actuel d'un fichier peut utiliser chown. Et ce, pour éviter des chmod 666 ou pire encore des chmod 777. La commande s'utilise de la façon suivante : ***sudo chown aibsrgl* aibsfile1**

NB : Les permissions standards sont : 666 pour les fichiers et 777 pour les dossiers

Nous y reviendrons à la Partie 2- Manipuler la console et les fichiers----Chapitre 4- Les commandes Linux----- 4) Gestion des droits (lab)

5-Editer un texte

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

'

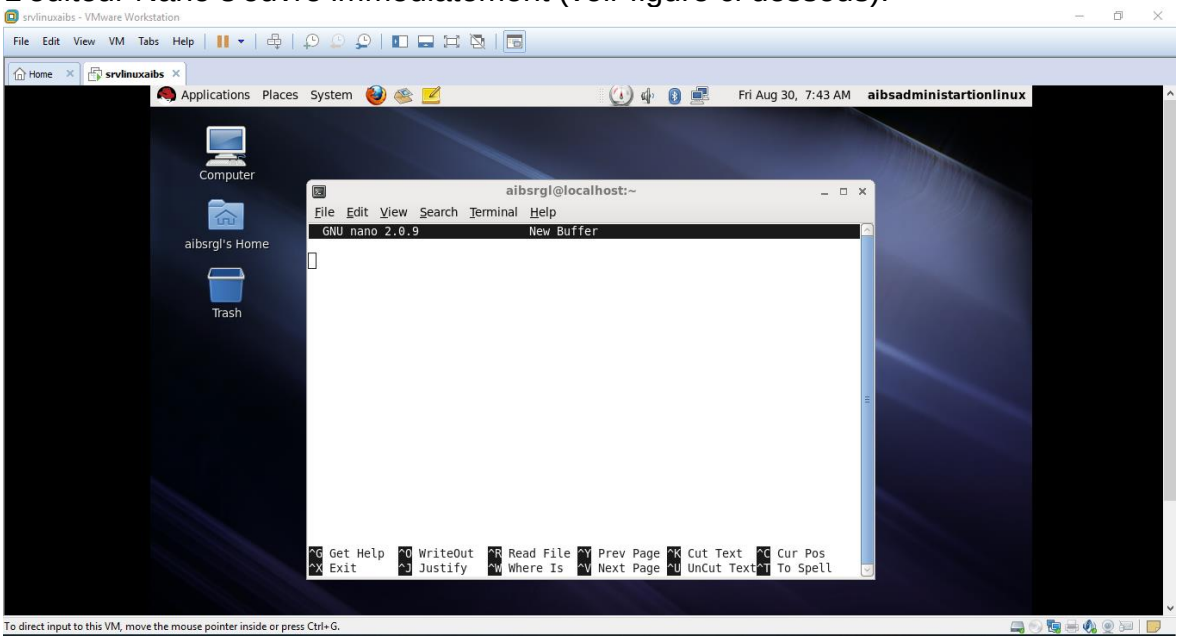
l'équivalent : Open Office Writer. Ces programmes ne peuvent être utilisés qu'en mode graphique, la console ne permettant pas vraiment de faire de la mise en forme.

**L’editeur Nano

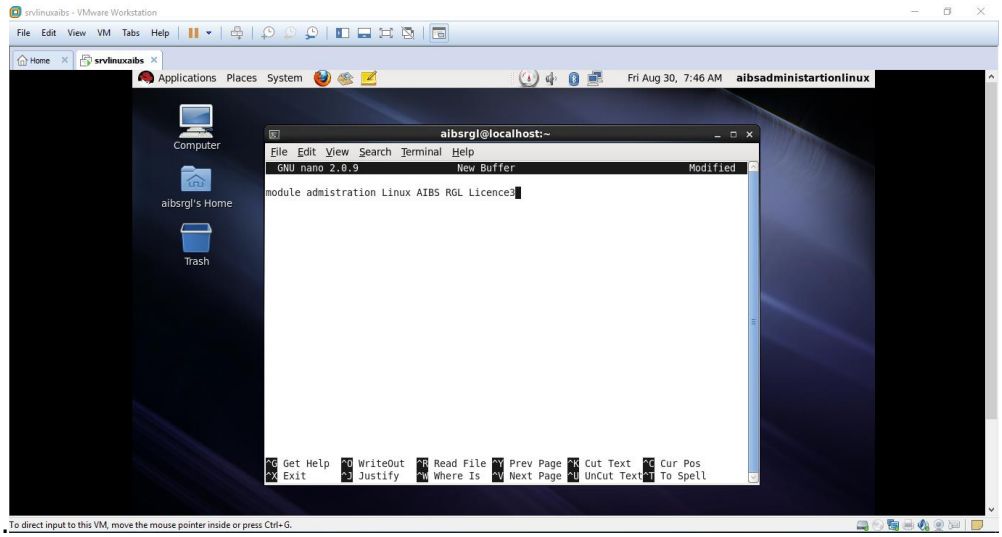
Le nom complet de Nano est « GNU nano », en référence au projet GNU. Il s'agit d'un logiciel qui s'inspire de « pico », un éditeur de texte plus ancien qui se voulait lui aussi très simple d'utilisation.

Pour démarrer le logiciel, il vous suffit simplement de taper **nano** dans la console :

L'éditeur Nano s'ouvre immédiatement (voir figure ci dessous).



Dès lors, vous pouvez commencer à taper du texte (exemple sur la figure suivante)

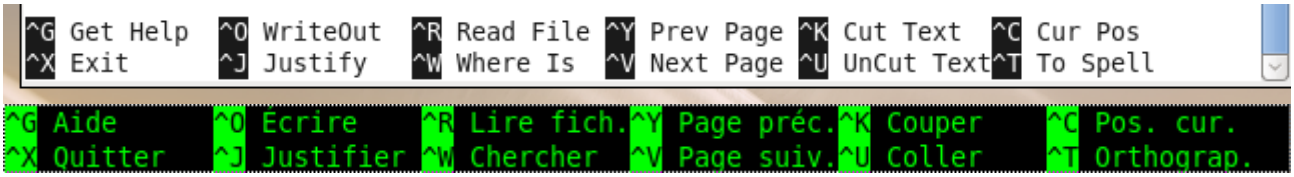


Certe il suffit de taper du texte mais cela n'est pas aussi simple sous d'autres éditeurs, comme Vim par exemple.

+Les raccourcis clavier de Nano

'

En bas de votre écran, vous pouvez voir un espace d'aide (figure suivante). Il s'agit d'un aide-mémoire pour vous rappeler à tout moment les commandes principales que vous pouvez lancer sous Nano.



Le symbole **^** signifie **Ctrl** . Ainsi, pour quitter Nano, il suffit de taper **Ctrl + X**.

Voici les raccourcis les plus importants :

Ctrl + G : afficher l'aide ;
Ctrl + K : couper la ligne de texte (et la mettre dans le presse-papier) ;
Ctrl + U : coller la ligne de texte que vous venez de couper ;
Ctrl + C : afficher à quel endroit du fichier votre curseur est positionné (numéro de ligne…) ;
Ctrl + W : rechercher dans le fichier ;
Ctrl + O : enregistrer le fichier (écrire) ;
Ctrl + X : quitter Nano.

+Enregistrer et quitter

Pour enregistrer, faites **Ctrl + O. Ctrl + X** pertmet de quitter sans enregistrer auparavant, un message vous demandera si vous voulez sauvegarder. Appuyez sur la touche o ou y (anglais),pour passer en mode enregistrement. Si vous appuyez sur la touche n, Nano quittera sans enregistrer
En appuyant sur **o**, vous vous retrouvez en mode enregistrement. Tapez juste le nom du fichier que vous voulez créer **aibsfile1** puis pressez Entrée.
Après ça, l'editeur nano se ferme.
Et vous revenez dans la ligne de commandes.
Pour ouvrir a nouveau le fichier crée, taper nano indiquer en paramètre le nom du fichier qu'on veut ouvrir. Ainsi on a :
nano aibsfile1.txt
ceci ouvrira le fichier **aibsfile1.txt** que l'on vient de créer.

À part ça, la commande nano accepte de nombreux paramètres. En voici trois essentielles

nano -m : autorise l'utilisation de la souris en console sous Nano.

'

nano -i : indentation automatique. L'alinéa (tabulations) de la ligne précédente sera respecté lorsque vous irez à la ligne. Très utile lorsque vous éditez un fichier de code source.

nano -A : active le retour intelligent au début de la ligne. Normalement, lorsque vous appuyez sur la touche origine (aussi connue sous le nom de Home) située à côté de la toucheFin, le curseur se repositionne au tout début de la ligne. Avec cette commande, il se positionnera après les alinéas. Comme nano-i, il s'agit d'une option utile avant tout pour les programmeurs.

On peut egalement lancer nano avec toutes ces options à la fois. On écrira :

nano -miA aibsfile1.txt dans la ligne de commande

D'autres editeurs existes, tels que **Sublime Text, Gedit, KWrite, Vim, GNU Emacs**…

6-Installer des programmes

Concept

L'installation de programmes fonctionne différemment d'une distribution Linux à une autre. C'est justement une des différences majeures entre les distributions Linux.

Sous Windows les programmes d'installation sont en général, des fichiers **.exe** à lancer et s'exécutent et extraient les fichiers du programme dans un dossier Program Files.

Sous Linux on n'a pas de programmes d'installation ; on a ce qu'on appelle des **paquets**. Un paquet est une sorte de dossier zippé qui contient tous les fichiers du programme.

Sous Linux, au lieu d'aller chercher des programmes sur internet, on se procure les paquets auprès de sa distribution, ce qui garantit que les paquets ont été testés pour la version de votre distribution : c'est un gage de stabilité et de sécurité.

En pratique, on installe un paquet (appelé aussi paquetage, en anglais, package) qui contient non seulement le programme, mais aussi la documentation, les fichiers de configuration, et les bibliothèques associées.

Presque chaque distribution possède son gestionnaire de paquets. Avant d'installer un logiciel, il faut donc savoir quelle est sa distribution et quel est le gestionnaire de paquets de sa distribution.

Ci-dessous voici des informations et des explications sur l'installation de logiciels par type de paquets les plus courants.

Type de paquets

Type RPM (Fedora, RHEL, CentOS, Mageia, SuSE ...)

RPM (Red Hat Package Manager) est un gestionnaire de paquets inventé en 1995 par Red Hat, et adopté par la suite par d'autres distributions telles que SuSE, Mageia ,Fedora, CentOS...)
Plusieurs programmes permettent d'automatiser les résolutions de dépendance et le téléchargement des paquets logiciels RPM :
chez Fedora : **dnf** (ligne de commande) ;

'

chez RHEL et CentOS : **yum** (ligne de commande), autrefois **up2date** ;
chez Mageia : **urpmi** (ligne de commande), **rpmdrake** (graphique) ;
chez SuSE : **zypperv** (ligne de commande), **YaST** (graphique).

Type APT (Debian, Ubuntu, Mint, Kali ...)
APT (Advanced Packaging Tool) est un gestionnaire de paquets inventé pour Debian, et utilisé par ses dérivés tels que Knoppix ou Ubuntu.
apt-get et **dpkg** sont les programmes de gestion de paquets en ligne de commande.

plusieurs interfaces graphiques à APT sont disponibles,
notamment **aptitude, kpackage, synaptic**.

Type TXZ (Slackware)
Les paquets Slackware sont des tarballs zippés sous forme de fichiers **.txz** (anciennement .tgz).
installpkg est le programme d'installation officiel en ligne de commande.

Type Portage (Gentoo)
Portage est le nom du gestionnaire de paquets de la distribution Gentoo.
emerge est le programme d'installation officiel en ligne de commande.

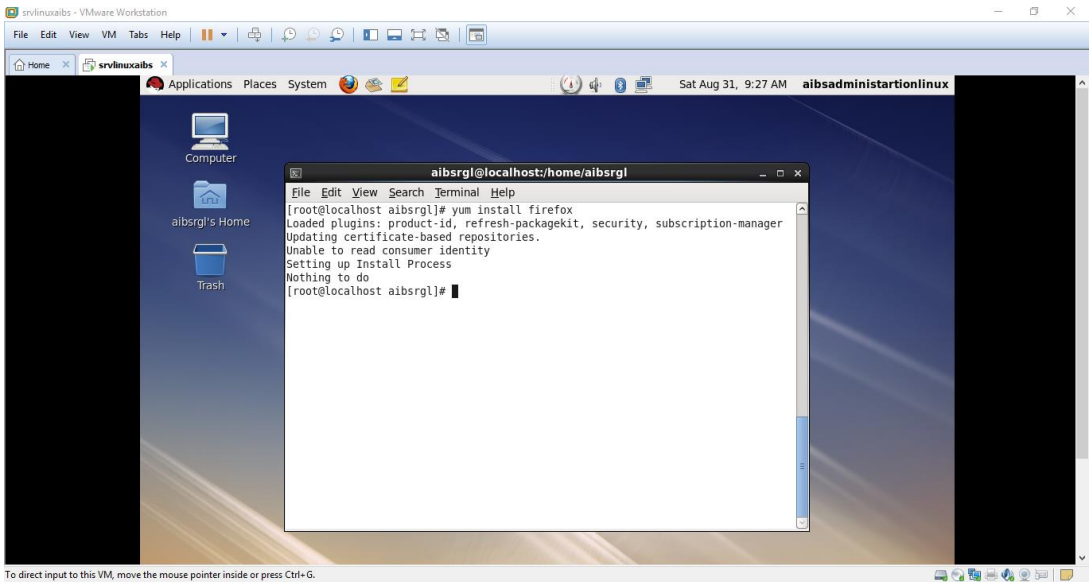
Type Pacman (Arch, Manjaro)
Pacman est le nom du gestionnaire de paquets de la distribution Arch.
pacman est le programme d'installation officiel en ligne de commande.

Type APK (Android)
Normalement, on installe un logiciel via le Google Play Store.
Mais on peut aussi installer un paquet APK à la main.

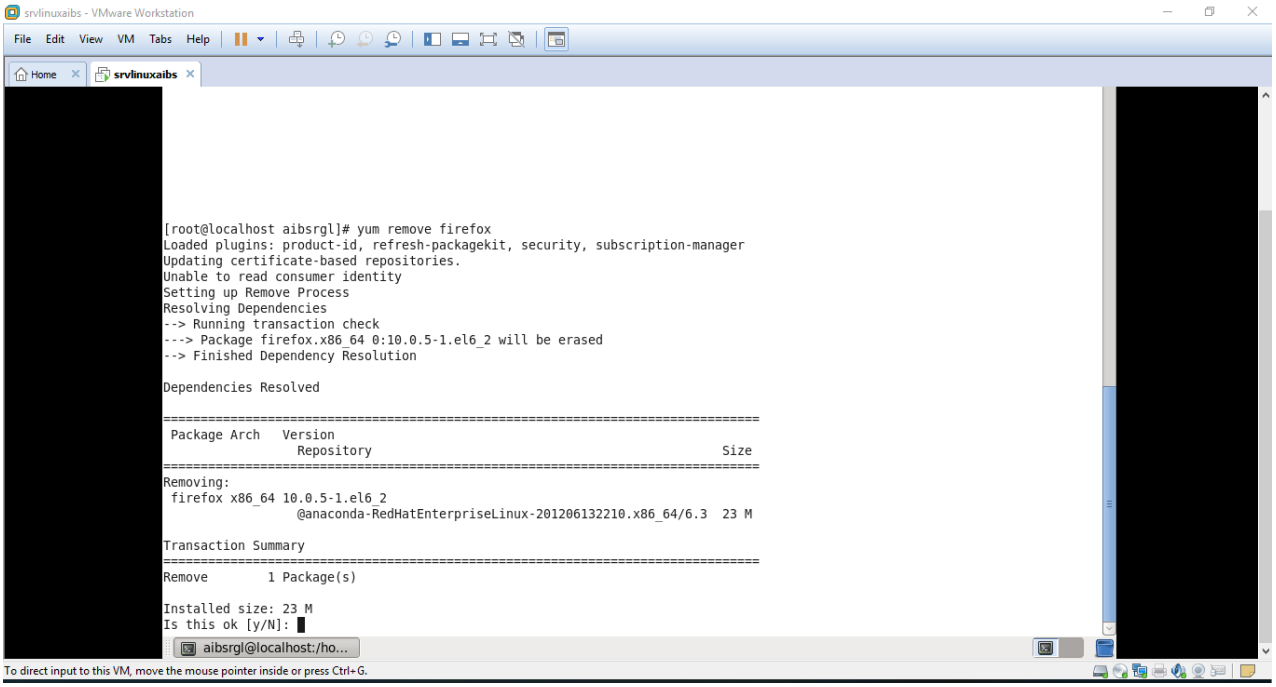
Installer un logiciel sous Red Hat en ligne de commande

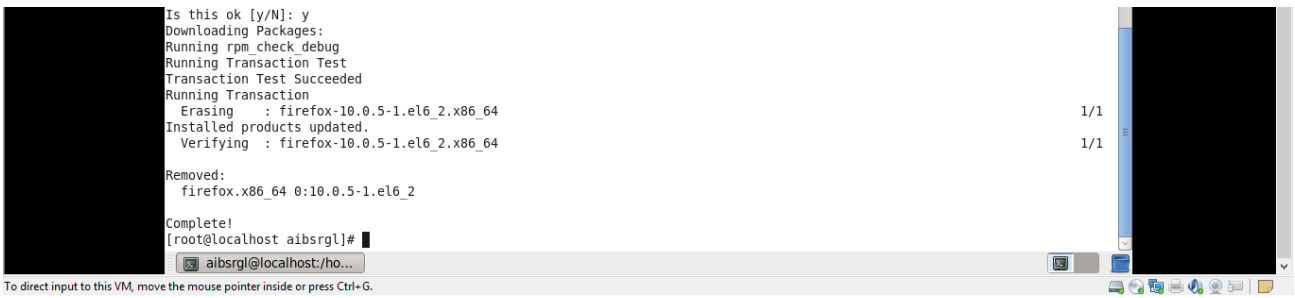
YUM, Il installe des paquets qui sont des fichiers du type **<nomdupaquet>.rpm**
Pour installer un programme, ou plutôt un paquet, ouvrez un terminal, Loguez vous en **root** avec la commande "**su**" et tapez : **yum install <nomduprogramme>**
Acceptez (en tapant "y") si YUM vous demande d'installer d'autres paquets en plus de celui que vous avez choisi. En effet, certains paquets ont besoin d'autres paquets (on appelle cela des dépendances) pour fonctionner.
Exemple : yum install firefox
Dans notre cas, firefox étant dejà installé voici ce que nous obtenons après avoir entré la commande :

'



Pour désinstaller un programme en ligne de commande
tapez **: yum remove <nomduprogramme>**
Encore une fois, acceptez si YUM vous demande de désinstaller d'autres paquets en plus de celui que vous avez choisi.
Exemple : yum remove firefox
NB : ne pas taper cette commande sur votre machine (desinstaller que les programme dont vous n'avez plus besoin)





'

Mettre à jour un programme
La commande est la suivante : **yum update <nomduprogramme**>
Rechercher un programme
Il peut être utile de rechercher un programme à installer, si on est pas sûr du nom ou pas sûr qu'il est dans les dépôts : **yum list <nomduprogramme>**
yum list affiche la liste des paquets du système

Mettre à jour tout le système
yum update met à jour tout le système d'un seul coup.
Attention, cette commande est périlleuse. En effet, beaucoup de paquets (éventuellement des centaines) vont être mis à jour. N'éteignez jamais votre ordinateur avant la fin de cette opération.

Chapitre 4 : Les commandes Linux

1-Les commandes de base

Quelques commandes de base à connaître

- ✓ **man**
  la commande **man**, permet de visionner le manuel d'une commande (son aide), en invoquant, dans une console, la commande suivante :
  man <nom_de_comande> ou **man [-s <section>] <nom_de_comande>**
  Chaque argument donné à **man** est généralement le nom d'un programme, d'un utilitaire ou d'une fonction.
  - Exemples d'utilisation : **man man :** affiche les informations pour l'utilisation de **man**
  - Tapez 'q' pour quitter

- ✓ **ls**
  - Équivalent MS-DOS/MS Windows : **dir**
  - Signification : *list segment*
  - Permet de lister un répertoire
  - Options les plus fréquentes :

  -l : Permet un affichage détaillé du répertoire (permissions d'accès, le nombre de liens physiques, le nom du propriétaire et du groupe, la taille en octets, et l'horodatage)
  -h : Associé avec **-l** affiche la taille des fichiers avec un suffixe correspondant à l'unité (K, M, G)
  -a : Permet l'affichage des fichiers et répertoires cachés (ceux qui commencent par un . (point))
  - Exemples d'utilisation :

  ls -a : affiche tous les fichiers et répertoires cachés du répertoire courant
  ls /etc/ : affiche le contenu du répertoire /etc/

- ✓ **mkdir**
  - Équivalent MS-DOS/MS Windows : **mkdir** ou **md**
  - Signification : *make directory*
  - Crée un répertoire vide
  - Options les plus fréquentes :

  -p : Crée les répertoires parents s'ils n'existent pas
  - Exemples d'utilisation :

  mkdir photos : Crée le répertoire *photos*

'

mkdir -p photos/2005/noel : Crée le répertoire *noel* et s'ils n'existent pas les répertoires *2005* et *photos*

✓ **rmdir**

- Équivalent MS-DOS/MS Windows : **rmdir** ou **rd**
- Signification : *remove directory*
- Supprime un répertoire (vide)
- Options les plus fréquentes :

-p : Supprime les répertoires parents s'ils deviennent vides

- Exemples d'utilisation :

rmdir LeRep : Supprime le répertoire *LeRep*

✓ **pwd**

- Équivalent MS-DOS/MS Windows : **chdir**
- Signification : *print working directory*
- Affiche le répertoire en cours

✓ **cat**

- Équivalent MS-DOS/MS Windows : **type**
- Signification : *concatenate*
- Affiche le contenu d'un fichier
- Options les plus fréquentes :

-n : Affiche les numéros de ligne
-v : Affiche les caractères de contrôles

- Exemple d'utilisation :

cat -n monFichier : Affiche le contenu de *monFichier* en numérotant les lignes à partir de 1

✓ **cd**

- Équivalent MS-DOS/MS Windows : **cd**
- Signification : *change directory*
- Permet de se promener dans les répertoires
- Exemples d'utilisation :

cd : permet de revenir au répertoire /home/utilisateur (identique à cd ~)
cd - : permet de revenir au répertoire précedent
cd .. : permet de remonter au répertoire parent
cd / : permet de remonter à la racine de l'ensemble du système de fichiers
cd /usr/bin/ : se place dans le répertoire /usr/bin/

✓ **mv**

- Équivalent MS-DOS/MS Windows : **move** ou **ren**
- Signification : *move*
- Permet de déplacer ou renommer des fichiers et des répertoires
- Options les plus fréquentes :

-f : Ecrase les fichiers de destination sans confirmation
-i : Demande confirmation avant d'écraser
-u : N'écrase pas le fichier de destination si celui-ci est plus récent

- Exemples d'utilisation :

mv monFichier unRep/ : Déplace *monFichier* dans le répertoire *unRep*
mv unRep/monFichier : Déplace le fichier *monFichier* du répertoire *unRep* là où on se trouve

'

mv unRep monRep : Renomme *unRep* en *monRep*

✓ **cp**

- Équivalent MS-DOS/MS Windows : **copy**
- Signification : *copy*
- Permet de copier des fichiers ou des répertoires
- Options les plus fréquentes :

-a : Archive. Copie en gardant les droits, dates, propriétaires, groupes, etc.
-i : Demande une confirmation avant d'écraser
-f : Si le fichier de destination existe et ne peut être ouvert alors le détruire et essayer à nouveau.
-r : Copie un répertoire et tout son contenu
-u : Ne copie que les fichiers plus récents ou qui n'existent pas
-v : permet de suivre les copies réalisées en temps réel

Exemples d'utilisation :

cp monFichier sousrep/ : Copie *monFichier* dans *sousrep*
cp -r monRep/ ailleurs/ : Copie le répertoire *monRep* vers *ailleurs* en créant le répertoire s'il n'existe pas.

✓ **rm**

- Équivalent MS-DOS/MS Windows : **del**
- Signification : *remove*
- Permet d'effacer des fichiers
- Options les plus fréquentes :

-f : Ne demande pas de confirmation avant d'effacer
-r : Efface récursivement les fichiers ainsi que les répertoires

- Exemples d'utilisation :

rm CeFichier : Efface le fichier *CeFichier*
rm -rf /tmp/LeRep : Efface le répertoire */tmp/LeRep* ainsi que tous ses fichiers sans demander de confirmation

2-Les commandes d'administration

Quelques commandes système

✓ **adduser**

- Signification : *add user*
- Ajoute un utilisateur, ou un groupe, au système.
- Options les plus fréquentes :

–disabled-login : Empêche l'utilisateur de se connecter.
–disabled-password : Un peu comme **</del>disabled-login** sauf qu'il est possible de se connecter via une clé RSA SSH, pratique pour créer un utilisateur qui ne se connectera que via SSH.
–system : Crée un utilisateur système.
–group : Avec **–system** crée un groupe avec le même ID que l'utilisateur système, sans un groupe avec le nom donné sera créé
–home : Permet de fixer le répertoire HOME de l'utilisateur.
–no-create-home : Ne crée pas de répertoire HOME.

- Exemples d'utilisation :

adduser CyberSDF : Crée l'utilisateur CyberSDF

'

adduser –disabled-password –no-create-home CyberSSH : Crée un utilisateur CyberSSH sans mot de passe qui ne pourra pas se connecter directement sur la machine et sans lui créer de répertoire home.

adduser –disabled-password –home /home/CyberSDF CyberSDF : Même chose qu'au dessus sauf qu'on lui donne le même répertoire HOME qu'à l'utilisateur CyberSDF créé en premier.

✓ **deluser**

- Signification : *delete user*
- Supprime un utilisateur du système.
- Option la plus fréquente :

–system : Ne supprime l'utilisateur que si c'est un utilisateur système.

–remove-home : Supprime l'utilisateur ainsi que son répertoire dans le home.

- Exemple d'utilisation :

deluser CyberSSH : Supprime l'utilisateur CyberSSH

deluser –remove-home bob : Supprime l'utilisateur bob ainsi que le répertoire /home/bob

✓ **chmod**

- Équivalent MS-DOS/MS Windows : **cacls**
- Signification : *change mode*
- Modifie les permissions d'accès à un fichier ou à un répertoire.

Type d'autorisations (une autorisation d'exécution sur un répertoire autorise son ouverture) :

+ : Ajoute une permission

- : Enlève une permission

= : Autorise uniquement l'autorisation indiquée

r : Lecture ; Valeur octale **4**

w : Ecriture ; Valeur octale **2**

x : Exécution ; Valeur octale **1**

s : Utilise les droits du propriétaire ou du groupe lors de l'exécution

u : Propriétaire du fichier

g : Groupe propriétaire du fichier

o : Tous les autres utilisateurs

- Options les plus fréquentes :

-R : Récursif, modifie les autorisation d'un répertoire et tout ce qu'il contient

-c : Ne montrer que les fichiers ayant été réellement modifiés

-f : Ne pas afficher les messages d'erreur

- Exemples d'utilisation :

chmod ugo+x monRep : Ajoute l'exécution (ouverture) du répertoire *monRep* à tous (propriétaire, groupe, autres)

chmod go-wx monRep : Supprime l'autorisation de lecture et d'écriture de *monRep* au groupe et aux autres

chmod u=rw,go=r MonFichier : Fixe l'autorisation de lecture et d'écriture au propriétaire de *MonFichier* et une autorisation de lecture au groupe et aux autres.

chmod 644 MonFichier : Exactement la même chose que ci-dessus mais en utilisant les valeurs octales (Nota : 6 = 4+2 = lecture + écriture)

chmod u=rw,g=r,o= MonFichier : Fixe l'autorisation d'ouverture et de lecture de *MonFichier* au propriétaire, uniquement la lecture au groupe et interdit tout accès aux autres.

chmod 640 MonFichier : Exactement la même chose que ci-dessus mais en utilisant les valeurs octales

'

✓ **chown**

- Équivalent MS-DOS/MS Windows : **cacls**
- Signification : *change owner*
- Change le propriétaire et le groupe propriétaire d'un fichier
- Options les plus fréquentes :

-R : Modifie récursivement un répertoire et tout ce qu'il contient

- Exemples d'utilisation :

chown autreUtilisateur MonFichier : Change le propriétaire de *MonFichier* en *autreUtilisateur*

chown -R lui:nous monRep : Change le propriétaire en *lui* et le groupe propriétaire en *nous* du répertoire *monRep* ainsi que tout ce qu'il contient

✓ **chgrp**

- Signification : *change groupe*
- Change le groupe propriétaire d'un fichier
- Options les plus fréquentes :

-R : Change récursivement un répertoire et tout ce qu'il contient

-h : Change le groupe propriétaire d'un lien symbolique et seulement lui (ne touche pas à la destination du lien)

-L : Si fournie avec **R**, change le groupe propriétaire d'un répertoire et des fichiers qu'il contient s'il est pointé par un lien symbolique rencontré lors de l'exécution.

- Exemples d'utilisation :

chgrp unGroupe MonFichier : Change le groupe propriétaire du fichier *MonFichier* en *unGroupe*

chgrp -R unGroupe monRep : Change le groupe propriétaire du répertoire *monRep* ainsi que tout ce qu'il contient en *unGroupe.*

✓ **sudo**

- Équivalent MS-DOS/MS Windows : **runas**
- Signification : *super user - do*
- Permet d'exécuter des commandes en tant qu'un autre utilisateur, donc avec d'autres privilèges que les siens.
- Options les plus fréquentes :

-s : Importe les variables d'environnement du shell

-k : Lorsque l'on utilise **sudo**, il garde en mémoire le mot de passe ; cette option déconnecte l'utilisateur et forcera à redemander un mot de passe si **sudo** est exécuté avant le timeout défini.

- Exemples d'utilisation :

$ sudo reboot : Lance la commande **reboot** avec les droits de l'utilisateur root

✓ **passwd**

- Signification : *password*
- Permet de modifier le mot de passe d'un utilisateur
- Options les plus fréquentes :

-S : Affiche l'état d'un compte (nom du compte, bloqué (L), si l'utilisateur n'a pas de mot de passe (NP) ou a un mot de passe utilisable (P), date de dernière modification du mot de passe, durée minimum avant modification, durée maximum de validité, durée d'avertissement, durée d'inactivité autorisée).

- Exemple d'utilisation :

p**asswd :** Demande à changer le mot de passe

'

✓ **groups**

- Signification : *groups*
- Affiche les groupes auxquels appartient un utilisateur
- Exemples d'utilisation :

groups : Affiche la liste des groupes auxquels appartient l'utilisateur ayant tapé la commande.

groups CyberSDF : Affiche tous les groupes auxquels appartient l'utilisateur CyberSDF.

✓ **usermod**

- Signification : *user modification*
- Modifie le groupe d'appartenance d'un utilisateur.
- Options les plus fréquentes :

-G, –groups GROUPE1 [,GROUPE2,…[,GROUPEN]]] : Ajouter l'utilisateur aux groupes précédents. Si l'utilisateur fait actuellement partie d'un groupe qui n'est pas listé, l'utilisateur sera supprimé du groupe. Ce comportement peut être changé avec l'option **-a**, qui permet d'ajouter l'utilisateur à une liste de groupes supplémentaires.

- Exemples d'utilisation :

usermod -aG toto machin : Ajoute l'utilisateur machin au groupe toto sans supprimer machin de son groupe originel.

sudo usermod -d /home/nouveau_login -m -l nouveau_login ancien_login : Permet de renommer le répertoire (dossier) utilisateur et de changer son nom. Pratique lorsque le pc change de mains.

3-Gestion des comptes et des groupes d'utilisateurs sur linux

Même si on est la seule personne à utiliser son système Linux, la compréhension et la gestion des comptes utilisateurs constituent un sujet très important pour l'administration du système. Et encore plus si la machine héberge plusieurs comptes de plusieurs personnes.

Les comptes utilisateur servent à beaucoup de choses sur les systèmes UNIX et Linux :

- Ils permettent de distinguer les différents utilisateurs qui ont accès au système, pour des raisons de sécurité. Chacun d'eux possède un compte personnel, auquel il accède par un identifiant et un mot de passe secret.
- Ces utilisateurs peuvent définir des permissions d'accès à leurs données, afin d'en autoriser ou d'en interdire l'exploitation par les autres.
- Les comptes permettent l'authentification de chaque utilisateur accédant au système, ce qui permet bien d'autres actes de gestion tels que gérer les courriers électroniques des utilisateurs par exemple ou savoir qui a fait une bêtise en examinant les fichiers de trace du système.

En dehors des comptes personnels, il existe des utilisateurs qui ne sont pas forcément des personnes physiques. Ces utilisateurs remplissent des fonctions administratives. C'est le cas du compte root utilisé par l'administrateur pour effectuer la maintenance par exemple.

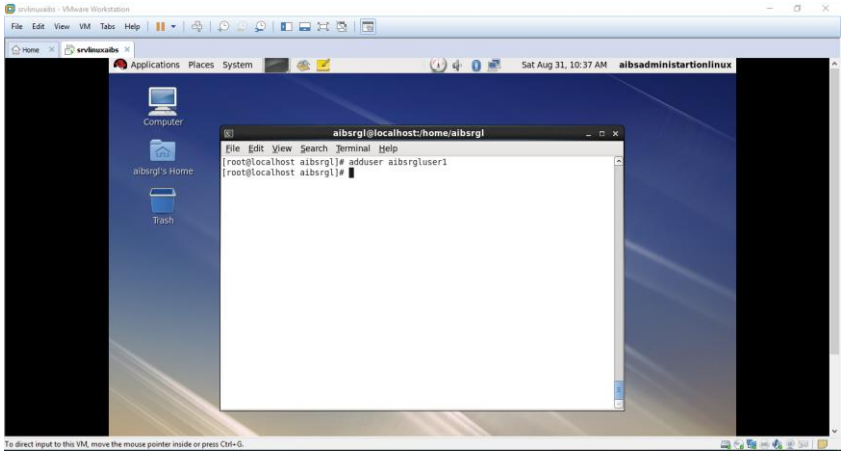
Les utilisateurs

Ajout d'un utilisateur :

Cette opération se fait via la commande **adduser.** Syntaxe **adduser nom_utilisateur.**

Exemple 1 : ajouter l'utilisateur "**aibsrgluser1**" ; on lance la commande **# adduser aibsrgluser1.**

'



Par défaut son répertoire utilisateur sera **/home/aibsrgluser1**
Exemple 2 : Ajouter aussi l'utilisateur "**aibsrgluser2**" avec la commande suivante : # **adduser aibsrgluser2**

Créer ou changer le mot de passe d'un utilisateur
Cette opération se fait via la commande **passwd.** Syntaxe **passwd nom_utilisateur**
Exemple 1 : **# passwd aibsrgluser2**
Entrez le nouveau mot de passe UNIX
Retapez le nouveau mot de passe UNIX :

Attention ! Si vous appelez **passwd** sans préciser de compte user en paramètre, c'est le mot de passe de root que vous changerez !

Changer d'utilisateur
Lorsque l'on est connecté à la machine et que l'on souhaite changer d'utilisateur, il suffit de lancer la commande : **# su autre_utilisateur** où **autre_utilisateur** est le nom d'un utilisateur existant sur le système. Son mot de passe sera alors demandé.
Pour revenir à l'utilisateur précédent il suffira ensuite de lancer la commande **:# exit**
Pour devenir super-utilisateur, la commande **su** seule suffit

supprimer un compte
deluser permet de supprimer un utilisateur.
 # deluser aibsrgluser1. Aucune confirrmation ne vous sera demandée !
Toutefois, cette commande seule ne supprime pas le répertoire personnel de **aibsrgluser1**.
Si vous voulez supprimer aussi son home et tous ses fichiers personnels, utilisez le paramètre **--remove-home** . exemple **:# deluser --remove-home aibsrgluser1**

Les groupes
Un groupe est, aussi pour Linux, un ensemble d'utilisateurs qui partagent les mêmes fichiers et répertoires. Les fichiers accordent des droits d'accès réglables à ces groupes. Chaque utilisateur doit faire partie au moins d'un groupe, son ***groupe primaire***. Celui-ci est défini au moment de la création du compte, et *par défaut*, l'utilisateur appartient à un nouveau groupe créé, portant son nom.

Ainsi, dans /etc/passwd chaque utilisateur possède un groupe par défaut, précisé par son identifiant gid dans ce fichier. L'appartenance au groupe primaire n'étant pas exclusive, tout utilisateur peut faire partie de plusieurs autres groupes, appelés ses *groupes secondaires*.

'
Mais le rôle joué par le groupe primaire demeure prépondérant. Pour lister tous les groupes (primaire et secondaires) d'un utilisateur : **# groups aibsrgluser1**

Créer un groupe
addgroup : La commande addgroup crée un nouveau groupe. Vous avez juste besoin de spécifier le nom de celui-ci en paramètre :
 # addgroup aibsrglclass
Super, mais personne ne fait encore partie de ce groupe

Modifier un utilisateur
usermod : La commande usermod permet d'éditer un utilisateur. Elle possède plusieurs paramètres ; nous allons en retenir deux :
-l : renomme l'utilisateur (mais le nom de son répertoire personnel ne sera pas changé);
-g : change de groupe.

Pour mettre **aibsrgluser1** dans le groupe **aibsrglclass,** faire :
#**usermod -g aibsrglclass aibsrgluser1**
Pour remettre **aibsrgluser1** dans le groupe **aibsrgluser1** comme il l'était avant :
#usermod -g aibsrgluser1 aibsrgluser1

Il est aussi possible de faire en sorte qu'un utilisateur appartienne à plusieurs groupes. Pour ce faire, utilisez le paramètre -G (majuscule).

Exemple: **usermod -G aibsrglclass,aibsfcclass,aibstlclass**. Séparez les noms des groupes par une virgule, sans espace entre chaque nom de groupe.

Faites très attention en utilisant usermod ! Lorsque vous avez recours à **-G**, l'utilisateur change de groupe et ce peu importe les groupes auxquels il appartenait auparavant.
Si vous voulez ajouter des groupes à un utilisateur (sans perdre les groupes auxquels il appartenait avant cela), utilisez **-a** :
#usermod -aG aibsrglclass aibsrgluser1

Supprimer un groupe
delgroup : permet de supprimer un groupe: #delgroup **aibsrglclass**

4-Les commandes pour la gestion des droits.

*Ref.. Partie 2- Manipuler la console et les fichiers-----Chapitre 3- Le système de fichiers linux -----4)-Les utilisateurs et les droits ------*Attribuer ou modifier des droits*

Trois commandes servent à modifier les droits : **chown (change owner), chgrp (change group) et chmod (change mode).** Elles sont utilisables par **root** ou le propriétaire du fichier.
chown : changer le propriétaire d'un fichier, d'un répertoire ou d'un ensemble de ces éléments. Syntaxe : **chown [options] nouvel_utilisateur [fichier(s)] [répertoire(s)]**
Exemple: **chown aibsrgluser2 aibsfile1**

chgrp : changer le groupe propriétaire.
Utilisable par root ou le propriétaire, à condition que celui-ci soit membre du nouveau groupe. Syntaxe **: chgrp [options] nouveau_groupe [fichier(s)] [répertoire(s)]**

'

La commande chgrp est redondante car chown permet de changer le groupe seul, ainsi que le propriétaire et le groupe en même temps.
Pour les deux commandes l'option «-R» permet de changer propriétaire et/ou groupe récursivement.

chmod : changer les permissions sur les fichiers ou répertoires (effectué par root ou le propriétaire). Syntaxe **: chmod [options] droits [fichier(s)] [répertoire(s)]**
La commande «chmod» peut s'écrire de plusieurs manières. Les droits peuvent se définir de façon relative, par ajout ou retrait par rapports aux droits existants ou bien de façon absolue, en remplaçant les anciens droits par les nouveaux droits.

Notation relative (aux droits existants).
Syntaxe : chmod [options] [ugoa] [+-=] [rwx] [fichier(s)] [répertoire(s)]

où «u, g et o» désignent les 3 catégories d'utilisateurs et «a» pour all, «r,w,x» les attributs, « +, -, = » l'action d'ajouter, de retirer ou de fixer un droit s'appliquant à une catégorie.

Exemples déjà vu
chmod o-w fichier3 :enlèvera le droit d'écriture pour les autres.
chmod a+x : ajoutera le droit d'exécution à tout le monde.
On peut aussi combiner plusieurs actions en même temps :
chmod u+rwx,g+rx-w,o+r-wx fichier3 : On ajoute la permission de lecture, d'écriture et d'exécution sur le fichier fichier3 pour le propriétaire ; On ajoute la permission de lecture et d'exécution au groupe propriétaire, on retire la permission d'écriture ;On ajoute la permission de lecture aux autres, on retire la permission d'écriture et d'exécution.

Exemple1 :
[root@linux phil]# ls -l
drwx------ 5 phil phil 1024 oct 6 19:24 Desktop/
[root@linux phil]# **chmod go+rwx Desktop**
[root@linux phil]# ls -l
drwxrwxrwx 5 phil phil 1024 oct 6 19:24 Desktop/

Exemple2 :
[root@linux phil]# ls -l essai2
-rw-rw-r-- 1 phil phil 8 oct 22 10:13 essai2
[root@linux phil]# **chmod go-r essai2**
[root@linux phil]# ls -l essai2
-rw--w---- 1 phil phil 8 oct 22 10:13 essai2

Notation absolue.
Syntaxe : chmod [options] u=...,g=...,o=... [fichier(s)] [répertoire(s)]
Pour chaque catégorie d'utilisateur, elle permet de fixer les nouveaux droits qui remplacent les anciens. Si une catégorie n'est pas présente, ses anciens droits s'appliquent. L'option «-R» permet de changer les droits récursivement

Exemple3 :
[root@linux phil]# ls -l
drwxr--r-- 5 phil phil 1024 oct 6 19:24 Desktop/
[root@linux phil]# **chmod g=r,o=rx Desktop**

'

[root@linux phil]# ls -l
drwxr--r-x 5 phil phil 1024 oct 6 19:24 Desktop/

Exemple4 :
[root@linux phil]# **chmod g=r,o=rwt Desktop**
[root@linux phil]# ls -l
drwxr--rwT 5 phil phil 1024 oct 6 19:24 Desktop/

Exemple5
[root@linux phil]# ls -l
drwxrwxr-t 2 phil phil 1024 oct 23 14:33 essai1/
[root@linux phil]# ls -l essai1
total 2
-rwSr--r-- 1 phil phil 42 oct 23 14:33 fichier1.txt
-rw-r-sr-x 1 phil phil 275 oct 23 14:33 fichier2.txt*
[root@linux phil]# **chmod -R u=rwx,g=r,o= essai1**
[root@linux phil]# ls -l
drwxr----- 2 phil phil 1024 oct 23 14:33 essai1/
[root@linux phil]# ls -l essai1
total 2
-rwxr----- 1 phil phil 42 oct 23 14:33 fichier1.txt*
-rwxr----- 1 phil phil 275 oct 23 14:33 fichier2.txt*

Permissions de manière octale, à l'aide de chiffres, exemple **chmod 755**
Prenons un répertoire les permissions sont :
drwxr-x---
En octal, on aura **750** :

rwx	**r-x**	**---**
7(4+2+1)	**5(4+0+1)**	**0(0+0+0)**

Pour mettre ces permissions sur le répertoire on taperait donc la commande :
chmod 750 aibsfile1

chmod 644 MonFichier = **chmod u=rw,g=r,o= MonFichier :** Exactement la même chose mais en utilisant les valeurs octales (Nota : 6 = 4+2 = lecture + écriture) **:** Fixe l'autorisation d'ouverture et de lecture de *MonFichier* au propriétaire, uniquement la lecture au groupe et interdit tout accès aux autres.
Autre exemple

$ ls -l monscript.sh
-rw-rw-r--. 1 francois francois 51 17 jan 05:02 monscript.sh
$ ./monscript.sh
-bash: ./monscript.sh: Permission non accordée
N'accorder les droits qu'au seul propriétaire
$ chmod 700 monscript.sh

'

Partie 3: Manipuler des données à travers le réseau

Chapitre 5 : Manipuler les données

1-commande de manipulation de fichier linux

Après avoir vu comment étaient organisés les fichiers sous Linux, nous allons maintenant les manipuler. Nous allons créer un fichier à l'aide d'une commande, afficher le contenu, déplacer, copier, supprimer un fichier.

Créer des fichiers et dossiers : touch & mkdir

créer un fichier : touch

En général, on se contente d'ouvrir un éditeur de texte et d'enregistrer, ce qui provoque la création d'un fichier sous linux comme sous Windows.

La commande **touch** est à la base faite pour modifier la date de dernière modification d'un fichier. D'où son nom : on « touche » le fichier pour faire croire à l'ordinateur qu'on vient de le modifier alors que l'on n'a rien changé. Ça peut se révéler utile dans certains cas précis qu'on ne verra pas ici.

L'intérêt de **touch** pour nous ici, c'est que si le fichier n'existe pas, il sera créé. On peut donc aussi utiliser touch pour créer des fichiers. **Syntaxe Touch monfichier**
Exemple : touch aibsrglfile1.txt

créer un dossier : mkdir

La commande mkdir, est faite pour créer un dossier. Elle fonctionne de la même manière que touch. **Syntaxe mkdir mondossier**
On peut créer deux dossiers (ou plus !) en même temps en les séparant par des espaces : mkdir mondossier1 mondossier2
Exemple : mkdir aibsrglfolder1
mkdir aibsrglfolder1 aibsrglfolder2 aibsrglfolder3

faire ls –l pour voir les dossiers et fichiers ont bien été créés.

mkdir –p permet de créer tous les dossiers intermédiaires.

Par exemple : mkdir -p animaux/vertebres/chat créera le dossier animaux, puis à l'intérieur le sous-dossier vertebres, puis à l'intérieur encore le sous-dossier chat .

Afficher un fichier : cat & less

Pour afficher le contenu d'un fichier il ya deux commandes basiques sous Linux : cat et less. Aucune de ces commandes ne permet d'éditer un fichier, elles permettent juste de le voir.

cat : afficher tout le fichier

La commande cat permet d'afficher tout le contenu d'un fichier dans la console d'un coup. Il vous suffit d'indiquer en paramètre le nom du fichier que vous voulez afficher

```
[aibsrgl@localhost ~]$ cat aibsfile1
module admistration Linux AIBS RGL Licence3
[aibsrgl@localhost ~]$
```

'

Le contenu de notre fichier céer avec l'éditeur nano

less : afficher le fichier page par page

La commande cat est rapide. Trop rapide. Tout le fichier est lu et affiché d'un coup dans la console, ce qui fait que l'on n'a pas le temps de le lire s'il est très gros.

C'est là qu'une autre commande comme **less** devient vraiment indispensable. La grosse différence entre less et cat, c'est que less affiche progressivement le contenu du fichier, page par page. Ça vous laisse le temps de le lire dans la console.

head & tail : afficher le début et la fin d'un fichier

Ces deux commandes sont un peu à l'opposé l'une de l'autre : la première permet d'afficher le début du fichier, la seconde permet d'afficher la fin.

head : afficher le début du fichier

La commandehead(« tête » en anglais) affiche seulement les premières lignes du fichier. Elle ne permet pas de se déplacer dans le fichier comme less, mais juste de récupérer les premières lignes.

tail : afficher la fin du fichier

Très intéressante aussi (voire même plus), la commande tail vous renvoie la fin du fichier, donc les dernières lignes.
Mais ce n'est pas tout ! Il y a un autre paramètre à côté duquel vous ne pouvez pas passer :-f (f pour follow, « suivre » en anglais).
Ce paramètre magique ordonne à tail de « suivre » la fin du fichier au fur et à mesure de son évolution. C'est extrêmement utile pour suivre un fichier de log qui évolue souvent.

cp & mv : copier et déplacer un fichier

Parmi les opérations de base que l'on veut pouvoir faire avec les fichiers, il y a la copie et le déplacement de fichier. C'est un peu le genre de chose que l'on fait tous les jours, il est donc important de savoir s'en servir.

cp : copier un fichier

La commande cp (abréviation de CoPy, « copier » en anglais) vous permet comme son nom l'indique de copier un fichier… mais aussi de copier plusieurs fichiers à la fois, et même de copier des dossiers !

Exemple : cp abisfile aibsfilecopie

Le premier paramètre est le nom du fichier à copier, le second le nom de la copie du fichier à créer. En faisant cela, on aura donc deux fichiers identiques dans le même répertoire.

Copier des dossiers

Avec l'option-R(un « R » majuscule !), vous pouvez copier un dossier, ainsi que tous les sous-dossiers et fichiers qu'il contient !

'

Tout à l'heure, on a créé un dossieranimauxqui contenait un autre dossiervertebres, qui lui-même contenait le dossierchat. Si vous tapez cette commande :

cp -R animaux autresanimaux

cela aura pour effet de copier animaux ainsi que tous ses sous-dossiers sous le nom autresanimaux.
Faites des ls –l après pour vérifier que les sous-dossiers sont bien.

Utiliser le joker*

Le symbole*est appelé joker, ou encore wildcard en anglais sous Linux.
Il vous permet de copier par exemple tous les fichiers image.jpg dans un sous-dossier : cp *.jpg mondossier/
Vous pouvez aussi vous en servir pour copier tous les fichiers dont le nom commence par « so » : cp so* mondossier/

mv : déplacer un fichier

Très proche de **cp**, la commande **mv** (MoVe, « déplacer » en anglais) a en fait deux utilités : déplacer un fichier (ou un dossier) ; renommer un fichier (ou un dossier).

Vous allez comprendre pourquoi.

Déplacer un fichier : La commande mv s'utilise pratiquement comme cp :

mv animaux/ mondossier/

Déplacera le dossier animaux (et tous ses sous-dossiers) dansmondossier.

Vous pouvez aussi utiliser les jokers :

mv *.jpg mondossier/

Renommer un fichier

La commande mv permet de faire quelque chose d'assez étonnant : renommer un fichier. En effet, il n'existe pas de commande spéciale pour renommer un fichier en console sous Linux, c'est la commandemvqui est utilisée pour ça.

Par exemple :mv fichierbidon superfichier renommerafichierbidonensuperfichier. Après cette commande,fichierbidon n'existe plus, il a été renommé.

.

Déplacer et renommer un fichier à la fois

Vous pouvez aussi déplacerfichierbidondansmondossiertout en lui affectant un nouveau nom : mv fichierbidon mondossier/superfichier

rm : supprimer des fichiers et dossiers

On attaque la commande qui fâche :rm.
Pourquoi est-ce qu'elle fâche ? Parce qu'il n'existe pas de corbeille dans la console de Linux : le fichier est directement supprimé sans possibilité de récupération !

rm : supprimer un fichier

'

La commanderm (pour ReMove, « supprimer » en anglais) peut supprimer un fichier, plusieurs fichiers, des dossiers, voire même votre ordinateur entier si vous le voulez.

Il faut donc l'utiliser avec précaution. supprimons ce **fichierbidon** :

rm fichierbidon

Normalement, on ne vous demande pas de confirmation, on ne vous affiche rien. Le fichier est supprimé sans autre forme d'avertissement.

Vous pouvez aussi supprimer plusieurs fichiers en séparant leurs noms par des espaces : rm fichierbidon fichiercopie

-i : demander confirmation

La commande-i permet de vous demander une confirmation pour chacun des fichiers :
mateo21@mateo21-desktop:~$ rm -i fichierbidon
rm: détruire fichier régulier vide `fichierbidon'?
Lorsqu'on vous demande une confirmation de type oui/non comme ici, vous devez répondre par une lettre :
o : signifie « Oui ». Sur certains systèmes anglais, il faudra peut-être utiliseryde Yes ;
n : signifie « Non ».
Tapez ensuite surEntrée pour valider.
-f : forcer la suppression, quoi qu'il arrive
-f, c'est un peu le contraire de-i : c'est le mode des gros bourrins.
Ce paramètre force la suppression, ne demande pas de confirmation, même s'il y a un problème potentiel.
En raison des risques que cela comporte, utilisez-le aussi rarement que possible.

rm -f fichierbidon

-r : supprimer un dossier et son contenu

Le paramètre-r peut être utilisé pour supprimer un dossier (au lieu d'un fichier) ainsi que tout ce qu'il contient : fichiers et dossiers !

C'est un paramètre assez dangereux, faites donc bien attention de l'utiliser sur un dossier dont vous ne voulez vraiment plus, car tout va disparaître à l'intérieur :

rm -r animaux/ supprime le dossieranimauxainsi que tout ce qu'il contenait (sous-dossiers vertebres et chat).

Notez qu'il existe aussi la commande **rmdir**. La grosse différence avec rm -r, c'est que rmdir ne peut supprimer un dossier que s'il est vide ! Il faudra y avoir fait le ménage auparavant.

2- rechercher des fichiers

Sous Linux, les fichiers sont organisés d'une façon assez particulière. Nous l'avons vu en affichant la liste des répertoires à la racine avec un ls /, il y a une foule de dossiers aux noms assez variés :var, opt, etc, bin, dev…

'

Une partie de ces répertoires est là pour des raisons historiques, depuis l'époque d'Unix. Le problème, c'est qu'il peut être difficile de retrouver le fichier dont on a besoin dans cette foule de répertoires.

On dispose heureusement sous Linux d'outils très puissants pour rechercher un fichier sur le disque dur. Certains d'entre eux sont très rapides, d'autres plus lents mais aussi plus complets.

locate : une recherche rapide

La première façon d'effectuer une recherche que nous allons voir est de loin la plus simple. La commande s'appelle locate(« localiser »). Elle est très rapide.

Son utilisation est intuitive, il suffit d'indiquer le nom du fichier que vous voulez retrouver. Par exemple :

mateo21@mateo21-desktop:~$ locate notes.txt

/home/mateo21/notes.txt

La commande a retrouvé notre fichiernotes.txtqui était situé dans/home/mateo21.

find : une recherche approfondie

find est la commande de recherche par excellence pour retrouver des fichiers, mais aussi pour effectuer des opérations sur chacun des fichiers trouvés. Elle est très puissante, permet donc de faire beaucoup de choses.

Utilisation basique de la commande findb :Recherche à partir du nom

Je me place dans mon répertoire home et je vais essayer de retrouver un fichier appelélogo.png que j'ai égaré. Je dois écrire :

find -name "logo.png"

Le-name "logo.png"est un paramètre qui demande de retrouver un fichier qui s'appelle très exactement logo.png.

3-Extraire, trier, filtrer des données

grep : filtrer des données

La commande grep est essentielle.Son rôle est de rechercher un mot dans un fichier et d'afficher les lignes dans lesquelles ce mot a été trouvé.

grep texte nomfichier

Le premier paramètre est le texte à rechercher, le second est le nom du fichier dans lequel ce texte doit être recherché.

Essayons par exemple de rechercher le mot « alias » dans notre fichier de configuration.bashrc. Rendez-vous dans votre répertoire personnel (en tapant cd) et lancez la commande suivante :

'

grep alias .bashrc

Cette commande demande de rechercher le mot « alias » dans le fichier.bashrcet affiche toutes les lignes dans lesquelles le mot a été trouvé.

Résultat :

$ grep alias .bashrc

/.bash_aliases, instead of adding them here directly.

#if [-f /.bash_aliases]; then

. /.bash_aliases

enable color support of ls and also add handy aliases

    alias ls='ls --color=auto'

    #alias dir='ls --color=auto --format=vertical'

    #alias vdir='ls --color=auto --format=long'

some more ls aliases

alias ll='ls -lArth'

#alias la='ls -A'

#alias l='ls -CF'

Les oprions avec grep

-i : ne pas tenir compte de la casse (majuscules / minuscules)

$ grep -i alias .bashrc

-n : connaître les numéros des lignes

$ grep -n alias .bashrc

-v : inverser la recherche : ignorer un mot. Pour connaître toutes les lignes qui ne contiennent pas un mot donné, utilisez-v :

$ grep -v alias .bashrc

-r : rechercher dans tous les fichiers et sous-dossiers

grep -r "aibs school" code/

Recherchera la chaîne « aibs school » dans tous les fichiers du répertoire code, y compris dans les sous-dossiers.

Notez que le « / » à la fin n'est pas obligatoire. Sans cela Linux comprendra tout de même très bien qu'il s'agit d'un répertoire.

'

sort : trier les lignes

La commande sort se révèle bien utile lorsqu'on a besoin de trier le contenu d'un fichier.

Pour nos exemples, je vous propose de créer un nouveau fichier (avec nano par exemple) appelé noms.txt et d'y placer le texte suivant :

François

Marcel

Albert

Jean

Stéphane

patrice

Vincent

jonathan

Ensuite, exécutez la commande sort sur ce fichier :

$ sort noms.txt

Albert

François

Jean

jonathan

Marcel

patrice

Stéphane

Vincent

Le contenu du fichier est trié alphabétiquement et le résultat est affiché dans la console. Vous noterez que sort ne fait pas attention à la casse (majuscules / minuscules).

-o : écrire le résultat dans un fichier

Le fichier en lui-même n'a pas été modifié lorsque nous avons lancé la commande. Seul le résultat était affiché dans la console.

Vous pouvez faire en sorte que le fichier soit modifié en précisant un nom de fichier avec l'option-o :

sort -o noms_tries.txt noms.txt

Ecrira la liste de noms triés dans noms_tries.txt.

'

-r : trier en ordre inverse

L'option-r permet d'inverser le tri :

$ sort -r noms.txt

-R : trier aléatoirement

Cette option permet de trier aléatoirement les lignes d'un fichier. C'est assez amusant et ça peut se révéler utile dans certains cas :

$ sort -R noms.txt

-n : trier des nombres

Le tri de nombres est un peu particulier. En effet, la commande sort ne distingue pas si les caractères sont des nombres et va donc par défaut les trier par ordre alphanumérique, en prenant en compte le premier chiffre uniquement. Par conséquent, le « mot » 129 précèdera 42 alors que ça devrait être l'inverse !

Prenons un exemple. Créez un nouveau fichiernombres.txtet placez-y les nombres suivants :

36

16

42

129

27

364

Triez-les comme vous avez appris à le faire :

$ sort nombres.txt

129

16

27

36

364

42

Comme vous voyez, tout ce qui commence par 1 est en premier, puis vient ce qui commence par 2 et ainsi de suite.
Bien sûr, quand on veut trier des nombres, c'est n'importe quoi.

C'est là que l'option-n intervient. Elle permet de trier en considérant le texte comme des nombres. Cette fois, le nombre 42 sera bien placé avant 129 !

'

$ sort -n nombres.txt

16

27

36

42

129

364

wc : compter le nombre de lignes

La commande wc signifie word count. C'est donc a priori un compteur de mots mais en fait, on lui trouve plusieurs autres utilités : compter le nombre de lignes (très fréquent) et compter le nombre de caractères.

Comme les précédentes, la commande wc travaille sur un fichier.

Sans paramètre, les résultats renvoyés parwcsont un peu obscurs. Voyez plutôt :

$ wc noms.txt

 8 8 64 noms.txt

Ces trois nombres signifient, dans l'ordre :

le nombre de lignes,le nombre de mots,le nombre d'octets.

-l : compter le nombre de lignes

Pour avoir uniquement le nombre de lignes, utilisez-l :

$ wc -l noms.txt

8 noms.txt

-w : compter le nombre de mots

Combien de mots différents y a-t-il dans le fichier ?

$ wc -w noms.txt

8 noms.txt

-c : compter le nombre d'octets

Combien d'octets comporte le fichier ?

$ wc -c noms.txt

64 noms.txt

'

-m : compter le nombre de caractères

L'option-m renvoie le nombre de caractères :

$ wc -m noms.txt

62 noms.txt

Comme vous pouvez le voir, le nombre de caractères est différent du nombre d'octets.

uniq : supprimer les doublons

Parfois, certains fichiers contiennent des lignes en double et on aimerait pouvoir les détecter ou les supprimer. La commande uniq est toute indiquée pour cela.

Nous devons travailler sur un fichier trié. En effet, la commande uniq ne repère que les lignes successives qui sont identiques.
Je vous propose de créer un fichierdoublons.txtcontenant les noms suivants :

Albert

François

François

François

Jean

jonathan

Marcel

Marcel

patrice

Stéphane

Vincent

Il y a des noms en double (et même en triple) dans ce fichier. Appliquons un petit coup de uniq là-dessus pour voir ce qu'il en reste :

$ uniq doublons.txt

Albert

François

Jean

jonathan

Marcel

patrice

Stéphane

'
Vincent

La liste de noms sans les doublons s'affiche alors dans la console !

Vous pouvez demander à ce que le résultat sans doublons soit écrit dans un autre fichier plutôt qu'affiché dans la console :

uniq doublons.txt sans_doublons.txt

La liste sans doublons sera écrite dans sans_doublons.txt.

-c : compter le nombre d'occurrences

Avec-c, la commande uniq vous affiche le nombre de fois que la ligne est présente dans le fichier :

$ uniq -c doublons.txt

1 Albert

3 François

1 Jean

1 jonathan

2 Marcel

1 patrice

1 Stéphane

1 Vincent

On sait ainsi qu'il y a trois fois « François », une fois « Jean », deux fois « Marcel », etc.

-d : afficher uniquement les lignes présentes en double

L'option-ddemande à afficher uniquement les lignes présentes en double :

$ uniq -d doublons.txt

François
Marcel
Comme seuls François et Marcel avaient des doublons, on les voit ici s'afficher dans la console.

4 -Archiver et Compresser

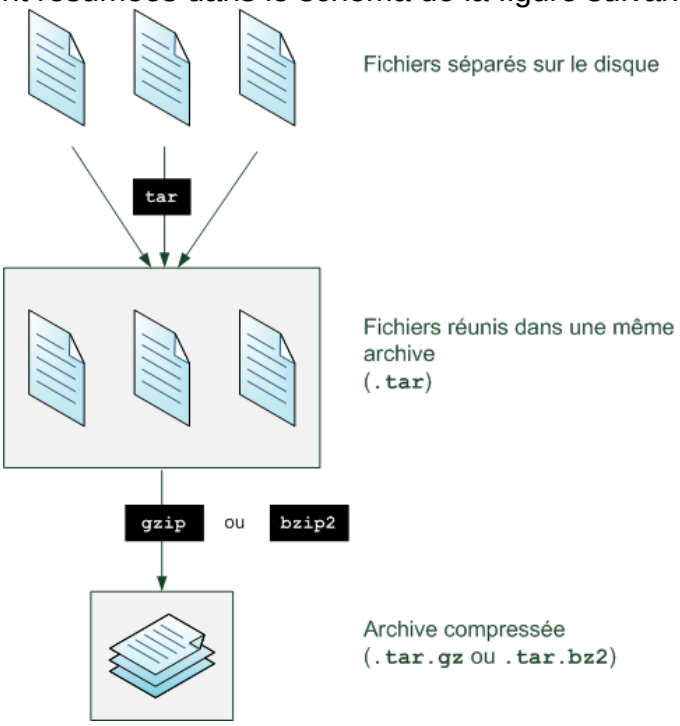
Si desirez envoyer un ou plusieurs fichiers par le réseau (que ce soit par mail, FTP ou autre), il est toujours préférable de commencer par les compresser afin de réduire leur taille.

tar : assembler des fichiers dans une archive

Sous Linux, on a pris l'habitude de procéder en deux étapes :
-réunir les fichiers dans un seul gros fichier appelé archive. On utilise pour cela le programme tar ;
-compresser le gros fichier ainsi obtenu à l'aide de gzip ou de bzip2.

'

Ces deux étapes sont résumées dans le schéma de la figure suivante.



Nous allons dans un premier temps apprendre à manipuler tar, puis nous verrons la compression avec gzip et bzip2.

Comme vous le voyez, sous Linux il y a donc une méthode à suivre dans un ordre précis. Voyons ensemble comment faire !

Regrouper d'abord les fichiers créer au par avant dans un même dossier

Vous avez plusieurs fichiers que vous souhaitez compresser.
Il est recommandé de placer d'abord les fichiers à archiver dans un seul et même dossier. Créons-le et déplaçons-y tous nos .txt :

```
$ mkdir tutoriels
$ mv *.txt tutoriels/
$ ls
```

Voilà, nos fichiers sont réunis dans le dossier tutoriels.

-cvf : créer une archive tar

Nous allons maintenant créer une archive tar de ce dossier et de ses fichiers. La procédure à suivre pour créer une archive est :

tar -cvf nom_archive.tar nom_dossier/

on utilise trois options :
-c : signifie créer une archive tar ;
-v : signifie afficher le détail des opérations ;
-f : signifie assembler l'archive dans un fichier.
Essayons de faire cela sur notre dossier tutoriels :

```
$ tar -cvf tutoriels.tar tutoriels/
```

Ici on archive le dossier tutoriels et donc son contenu. Grâce à -v, on voit bien la liste des fichiers qui ont été archivés.

Est-on obligé de mettre systématiquement nos fichiers dans un même dossier pour archiver ensuite ce dossier ? On ne pourrait pas archiver directement les fichiers ?
Si, c'est possible. Imaginons que nous soyons toujours dans notre home avec nos fichiers .tuto. On pourrait très bien faire :

tar -cvf archive.tar fichier1 fichier2 fichier3

'
C'est possible et ça fonctionne. Toutefois, il est de coutume sous Linux de placer d'abordles fichiers dans un dossier avant de les mettre dans le tar. Cela permet d'éviter, lorsqu'on extrait les fichiers de l'archive, que ceux-ci aillent se mêler à d'autres fichiers..

-tf : afficher le contenu de l'archive sans l'extraire

Vous venez de recevoir une archive tar qu'on vous a envoyée. Bien. Mais que contient-elle ?

Avant d'extraire quoi que ce soit, vous aimeriez peut-être voir son contenu. C'est possible avec -tf : **$ tar -tf tutoriels.tar**

Quand on fait cela, on voit que tous les fichiers sont réunis dans un même dossier tutoriels, et ça c'est très pratique.

-rvf : ajouter un fichier

Si vous avez oublié un fichier, vous pouvez toujours l'ajouter par la suite avec -rvf :

$ tar -rvf tutoriels.tar fichier_supplementaire.tuto

-xvf : extraire les fichiers de l'archive

Pour extraire les fichiers, on va utiliser les options -xvf (-x pour eXtract) :

$ tar -xvf tutoriels.tar

Les fichiers s'extraient dans le répertoire dans lequel vous vous trouvez. Vérifiez donc avant de les extraire que ceux-ci sont réunis dans un même dossier (avec -tf) si vous ne voulez pas que ces fichiers aillent se mélanger à d'autres !

gzip & bzip2 : compresser une archive

Vous avez maintenant créé une belle archive tar. Tous vos fichiers sont réunis là-dedans. Voyons comment compresser cela.

Nous disposons de deux programmes de compression bien répandus dans le monde Linux :

gzip : c'est le plus connu et le plus utilisé ; **bzip2** : il est un peu moins fréquemment utilisé. Il compresse mieux mais plus lentement que gzip.

Ces programmes sont simples à utiliser. Ils prennent comme paramètre le nom du fichier à compresser. Ils le compressent et modifient ensuite son nom.

Concrètement, ils ajoutent un suffixe pour indiquer que l'archive a été compressée :

.tar.gz : si l'archive a été compressée avec gzip ;

.tar.bz2 : si l'archive a été compressée avec bzip2.

Concrètement, le programme gzip s'utilise de la manière la plus simple qui soit :

gzip tutoriels.tar

L'archive est compressée et gagne ensuite le suffixe .gz. Elle s'appelle donc désormais tutoriels.tar.gz. Voilà pourquoi vous voyez circuler sur l'internet des fichiers .tar.gz : cela signifie que ce sont des archives compressées !

Pour décompresser l'archive ensuite, il suffit d'utiliser gunzip :

gunzip tutoriels.tar.gz

L'archive retrouve son état non compressé en .tar. Vous pouvez maintenant extraire les fichiers de l'archive comme vous avez appris à le faire un peu plus tôt avec tar -xvf.

bzip2 : la compression la plus puissante

Le fonctionnement de bzip2 est le même que celui de gzip :

bzip2 tutoriels.tar

Une archive compressée tutoriels.tar.bz2 sera alors créée. Pour la décompresser, utilisez bunzip2 :

bunzip2 tutoriels.tar.bz2

Vous retrouvez un .tar que vous pouvez extraire avec tar -xvf.

'

Archiver et compresser en même temps avec tar

-zcvf : archiver et compresser en gzip

Vous connaissez tar -cvf pour créer une archive tar. Si vous rajoutez l'option -z, l'archive sera automatiquement compressée avec gzip.

tar -zcvf tutoriels.tar.gz tutoriels/

Voilà comment on obtient une archive compressée en une seule commande.
Pour décompresser, c'est pareil, sauf que le -c est remplacé par un -x comme tout à l'heure :

tar -zxvf tutoriels.tar.gz

-jcvf : archiver et compresser en bzip2

Le principe est le même avec -j à la place de -z :
tar -jcvf tutoriels.tar.bz2 tutoriels/
Et pour extraire :
tar -jxvf tutoriels.tar.bz2 tutoriels/

unzip & unrar : décompresser les .zip et .rar

Les .tar.gz et .tar.bz2 ont beau être courants dans le monde Linux, vos amis utilisant Windows ne les connaissent pas et risquent tôt ou tard de vous envoyer un superbe .zip ou .rar… que vous ne pouvez pas décompresser avec gunzip.
Heureusement, il existe des utilitaires de décompression pour ces formats. Ils ne sont pas toujours installés par défaut, il faudra donc les installer si vous ne les avez pas.

unzip : décompresser un .zip

Vous venez de recevoir un .zip ?
Pas de panique ! Le programme unzip est capable de l'extraire. Il est peut-être installé par défaut, mais si vous ne l'avez pas, vous savez ce qu'il vous reste à faire :

sudo apt-get install unzip

Ceci étant fait, l'utilisation d'unzip est très simple :

unzip archive.zip

Les fichiers vont s'extraire dans le dossier dans lequel vous vous trouvez ! Le problème est le même qu'avec les .tar.gz et .tar.bz2. Avant de décompresser, vérifiez si les fichiers sont réunis dans un même dossier.
Pour voir le contenu d'une archive zip sans l'extraire, utilisez -l :**$ unzip -l tutoriels.zip**

unrar : décompresser un .rar

Il vous faut installer le paquet **unrar** pour pouvoir utiliser cette commande :

sudo apt-get install unrar

Ensuite, pour extraire :**unrar e tutoriels.rar**
NB :on ne met pas ici de tiret devant l'option e.
Pour lister le contenu avant décompression, utilisez l'option l :
$ unrar l tutoriels.rar

'

Chapitre 6 : Configuration et administration reseau

1-Surveiller l'activité du système

Comme tous les OS actuels, Linux est un système **multi-tâches** : il est capable de gérer plusieurs programmes tournant en même temps.
Mieux encore, Linux est un système **multi-utilisateurs** : plusieurs personnes peuvent utiliser la même machine en même temps (en s'y connectant via Internet).

Tous ces programmes et ces personnes qui sont sur votre PC peuvent vite donner le tournis. Parfois, l'ordinateur peut se retrouver surchargé à cause d'un programme. Qui a lancé ce programme ? Depuis quand ? Comment arrêter un programme qui ne répond plus ?

Sous Windows, vous avez probablement entendu parler de la commande magique Ctrl + Alt + Suppr qui peut parfois vous sortir de bien des situations embarrassantes. Sous Linux, on utilise d'autres outils et d'autres techniques.

Voici queleques commandes qui nous permettront de savoir ce qui se passe dans notre ordinateur

- **w : qui fait quoi ?**

Première commande qu'on tape en général quand on se connecte à un serveur surchargé et qu'on veut essayer de comprendre ce qui se passe. Cela me permet de voir d'un seul coup d'oeil si la machine est vraiment surchargée (et si oui, à quel point) et si quelqu'un d'autre est en train d'intervenir sur la machine.

Si vous utilisez Linux sur votre ordinateur personnel, tranquillement chez vous, vous êtes seuls à l'utiliser en ce moment. Pour que d'autres personnes puissent se connecter à votre ordinateur via Internet, il faut avoir configuré Linux pour ça.

On en a principalement besoin sur les serveurs. Voici un exemple de message renvoyer par un serveur en tapant **: $ w**

```
 16:50:30 up  8:50,  2 users,  load average: 0,08, 0,34, 0,31
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
mateo21  :0       -                19Apr08 ?xdm?   3:38m  1.18s /usr/bin/gnome-
mateo21  pts/0    :0.0             16:49    0.00s  0.33s  0.03s w
```

Bon : à première vue, c'est court mais dense, ça n'a pas l'air très clair.

Pourtant, cette commande nous donne un condensé d'informations très utiles que je vais vous présenter dans l'ordre, de gauche à droite et de haut en bas.

L'heure (aussi accessible via date)

Ici, l'heure qui nous est donnée est 16:50:30 (16 h 50 mn 30 s).

Essayez aussi les commandes suivantes :.

$ date

'

$ uptime

- **ps & top : lister les processus**

La commande w nous a permis de faire rapidement le point sur l'état du système. Listons maintenant les processus qui tournent sur votre machine.

Pour faire simple, dites-vous qu'un processus est un programme qui tourne en mémoire. La plupart des programmes ne font tourner qu'un processus en mémoire (une seule version d'eux-mêmes). C'est le cas d'OpenOffice par exemple. D'autres lancent des copies d'eux-mêmes, c'est le cas du navigateur Google Chrome qui crée autant de processus en mémoire que d'onglets ouverts.

Sur un serveur web, on utilise en général le logiciel Apache qui délivre les pages web aux internautes. Ce logiciel crée beaucoup de processus pour séparer ses activités. Il en va de même pour les systèmes de gestion de bases de données, comme MySQL et PostgreSQL.
Il ne faut pas s'inquiéter si un programme génère beaucoup de processus, cela n'est pas anormal.

Pour lister les processus qui tournent sous Windows, on utilise Ctrl + Alt + Suppr et on va dans l'onglet « Processus ». Sous Linux, on peut utiliser deux commandes différentes : ps et top.

ps : liste statique des processus

ps vous permet d'obtenir la liste des processus qui tournent au moment où vous lancez la commande. Cette liste n'est pas actualisée en temps réel, contrairement à ce que fait top et qu'on verra plus tard.

Essayons d'utiliser ps sans paramètre :

$ ps

```
  PID TTY          TIME CMD
23720 pts/0    00:00:01 bash
29941 pts/0    00:00:00 ps
```

On distingue quatre colonnes.

PID : c'est le numéro d'identification du processus. Chaque processus a un numéro unique qui permet de l'identifier. Ce numéro nous sera utile plus tard lorsque nous voudrons arrêter (tuer) le processus.

TTY : c'est le nom de la console depuis laquelle a été lancé le processus.

TIME : la durée d'exécution du processus. Plus exactement, cela correspond à la durée pendant laquelle le processus a occupé le processeur depuis son lancement.

'
CMD : le programme qui a généré ce processus. Si vous voyez plusieurs fois le même programme, c'est que celui-ci s'est dupliqué en plusieurs processus (c'est le cas de MySQL, par exemple).

Dans mon cas, on distingue deux processus : bash (qui correspond à l'invite de commandes qui gère les commandes) et ps que je viens de lancer.

Quand on utilise ps sans argument comme on vient de le faire, il affiche seulement les processus lancés par le même utilisateur dans la même console .Cela limite énormément les processus affichés, car beaucoup sont lancés par root (l'utilisateur administrateur de la machine) et ne sont pas lancés depuis la même console que la vôtre.

La commande ps vous permet d'utiliser énormément d'options.

ps -ef : lister tous les processus

Avec ps -ef, vous pouvez avoir la liste de tous les processus lancés par tous les utilisateurs sur toutes les consoles **:$ ps –ef**

Vous noterez l'apparition de la colonne UID (User ID) qui indique le nom de l'utilisateur qui a lancé la commande. Il y en a beaucoup, lancés par root automatiquement au démarrage de la machine, dont vous n'avez jamais entendu parler.

ps -ejH : afficher les processus en arbre

Cette option intéressante vous permet de regrouper les processus sous forme d'arborescence. Plusieurs processus sont des « enfants » d'autres processus, cela vous permet de savoir qui est à l'origine de quel processus. **$ ps –ejH**

.**ps -u UTILISATEUR : lister les processus lancés par un utilisateur**

Pour filtrer un peu cette longue liste, on peut utiliser -u afin d'obtenir par exemple uniquement les processus que l'on a lancés nous-mêmes. $ ps -u aibsrgl

Ici, j'obtiens uniquement les processus lancés par l'utilisateur « aibsrgl », ce qui filtre déjà pas mal les autres processus système lancés par root.

top : liste dynamique des processus

La liste donnée par ps a un défaut : elle est statique (elle ne bouge pas). Or, votre ordinateur, lui, est en perpétuel mouvement. De nombreux processus apparaissent et disparaissent régulièrement.
Comment avoir une liste régulièrement mise à jour ? Avec la commande top! Essayez-la : $ top
NB: Faites CRTL+C pour revenir sur linvite de commande ou q

Cette liste est interactive et régulièrement mise à jour.

'

En haut, vous retrouvez l'uptime et la charge, mais aussi la quantité de processeur et de mémoire utilisée. Nous n'entrerons pas dans les détails à ce niveau car cela demanderait un peu trop d'explications avancées sur le fonctionnement du système d'exploitation. Néanmoins, si vous savez lire la charge et la mémoire disponible, vous pouvez déjà vous faire une idée de ce qui se passe.

En dessous, vous avez la liste des processus.

Par défaut, les processus sont triés par taux d'utilisation du processeur (colonne %CPU). Les processus que vous voyez tout en haut de votre liste sont donc actuellement les plus gourmands en processeur. Ce sont peut-être eux que vous devriez cibler en premier si vous sentez que votre système est surchargé.

On navigue à l'intérieur de ce programme en appuyant sur certaines touches du clavier. En voilà au moins deux à connaître : **q** : ferme top ; **h** : affiche l'aide, et donc la liste des touches utilisables.

Voici quelques commandes à connaître au sein de top qui peuvent vous être utiles.
B : met en gras certains éléments.
f : ajoute ou supprime des colonnes dans la liste.
F : change la colonne selon laquelle les processus sont triés. En général, laisser le tri par défaut en fonction de %CPU est suffisant.
u : filtre en fonction de l'utilisateur que vous voulez.
k : tue un processus, c'est-à-dire arrête ce processus. Ne vous inquiétez pas, en général les processus ne souffrent pas. On vous demandera le numéro (PID) du processus que vous voulez tuer. Nous reviendrons sur l'arrêt des processus un peu plus loin.
s : change l'intervalle de temps entre chaque rafraîchissement de la liste (par défaut, c'est toutes les trois secondes).

- **Ctrl + C & kill : arrêter un processus**

Parfois, rien ne va plus. Un processus s'emballe et ne veut pas s'arrêter. Cela arrive partout, même sous Linux. À la différence de Windows toutefois, vous ne devriez pas avoir le réflexe de redémarrer « pour que ça aille mieux ». Tout peut être résolu en arrêtant les processus qui vous gênent et en les relançant au besoin.

Il y a plusieurs façons d'arrêter un processus

Ctrl + C : arrêter un processus lancé en console

La combinaison de touches Ctrl + C est à connaître. Cela demande (gentiment) l'arrêt du programme console en cours d'exécution à l'écran. Ce raccourci se comporte ainsi en mode console seulement. En effet, en mode graphique, le comportement est le même que sous Windows : cela permet d'effectuer une copie dans le presse-papier. Notez que pour copier-coller sous Linux, on utilise souvent une autre technique : on sélectionne du texte et on clique avec la molette de la souris pour le coller ailleurs.

Prenez une commande qui n'en finit plus, comme par exemple un find sur l'ensemble du disque. Celui-ci va analyser tout votre disque dur à la recherche du fichier demandé. Si vous trouvez cela trop long et que vous voulez arrêter le programme en cours de route, il vous suffit de taper **Ctrl + C :**

'

find / -name "*log*"

/dev/log

/bin/login

/sys/module/scsi_mod/parameters/scsi_logging_level

/sys/module/ehci_hcd/parameters/log2_irq_thresh

La liste aurait dû être beaucoup plus longue. Mais j'ai demandé l'arrêt du programme avec Ctrl + C, ce qui fait que j'ai pu « retrouver » l'invite de commandes rapidement et facilement.

Taper Ctrl + C ne coupe pas le programme brutalement, cela lui demande gentiment de s'arrêter, comme si vous aviez cliqué sur la croix pour fermer une fenêtre.

kill : tuer un processus

Ctrl + C ne fonctionne que sur un programme actuellement ouvert dans la console. De nombreux programmes tournent pourtant en arrière-plan, et Ctrl + C n'aura aucun effet sur eux.
C'est là que vous devez utiliser kill si vous voulez les arrêter (on dit aussi « tuer »)
Pour vous en servir, il faudra auparavant récupérer le PID du ou des processus que vous voulez tuer. Pour cela, deux solutions :
ps et top.
Ces deux commandes que nous venons de voir vous indiquent le PID (numéro d'identification) de chaque processus. Par exemple avec ps :

```
$ ps -u aibsrgl
  PID TTY          TIME CMD
 5012 ?        00:00:01 gnome-session
 5057 ?        00:00:00 ssh-agent
 ...
25227 pts/1    00:00:00 bash
```

Supposons qu'on souhaite arrêter Firefox. On peut filtrer cette longue liste avec grep et un pipe que nous avons appris à utiliser.

```
$ ps -u mateo21 | grep firefox
32678 ?        00:00:03 firefox-bin
```

on a filtré Firefox de cette longue liste et on a même récupéré son PID. Il ne nous reste plus qu'à le tuer, avec la commande suivante :

```
kill 32678
```

Si tout va bien, la commande ne renvoie rien. Sinon, une erreur devrait s'afficher dans la console.

Vous pouvez aussi tuer plusieurs processus d'un seul coup en indiquant plusieurs PID à la suite :

```
kill 32678 2768 33071
```

Attention : même si kill est par défaut une commande « gentille » qui demande simplement au processus de s'arrêter, évitez de tuer des processus que vous ne connaissez pas. Beaucoup d'entre eux sont essentiels au bon fonctionnement de votre système, surtout ceux qui ont été lancés par root.

Vous voulez tuer un processus sans lui laisser le choix ?
Avec **kill -9** , vous demandez à Linux de tuer le processus sans lui laisser le temps de s'arrêter proprement. Cela peut faire le ménage quand rien ne va plus.

'

kill -9 32678

… tuera le processus n°32678 (Firefox, dans mon cas) immédiatement sans lui laisser le temps de finir.

killall : tuer plusieurs processus

Contrairement à kill, killall attend le nom du processus à tuer et non son PID.
Supposons que nous ayons trois processus find en cours d'exécution que nous souhaitions arrêter.

```
$ ps -u mateo21 | grep find
  675 pts/1    00:00:01 find
  678 pts/2    00:00:00 find
  679 pts/3    00:00:01 find
```

Pour tous les tuer, il faudra donc taper : $ killall find
Si la commande ne renvoie rien, c'est que tout s'est bien passé.
En revanche, si vous avez :

```
$ killall find
find: aucun processus tué
```

… cela signifie qu'il n'y avait aucun processus de ce nom à tuer. Soit le processus n'est plus là, soit vous n'avez pas écrit correctement son nom. Vérifiez ce nom à nouveau avec la commande ps.

- **halt & reboot : arrêter et redémarrer l'ordinateur**

Nous venons d'apprendre à arrêter des processus avec kill. Je pense que le moment est bien choisi pour découvrir comment arrêter et redémarrer l'ordinateur.

Comme je vous le disais plus tôt, il est assez rare que l'on soit forcé d'arrêter ou de redémarrer l'ordinateur. À moins d'avoir mis à jour le kernel (noyau) de Linux, il n'est jamais nécessaire de redémarrer.
L'arrêt et le redémarrage d'un serveur sous Linux sont réellement des opérations exceptionnelles.

halt : arrêter l'ordinateur

La commande halt commande l'arrêt immédiat de l'ordinateur. Il faut être root pour arrêter la machine ; vous devrez donc taper : $ sudo halt

Un message sera affiché dans la console pour annoncer l'arrêt de l'ordinateur.

reboot : redémarrer l'ordinateur

De même, il existe la commande reboot pour redémarrer l'ordinateur. Il faut à nouveau être root : $ sudo reboot

Le redémarrage prend effet immédiatement.

'

2-configuration carte reseau ethernet

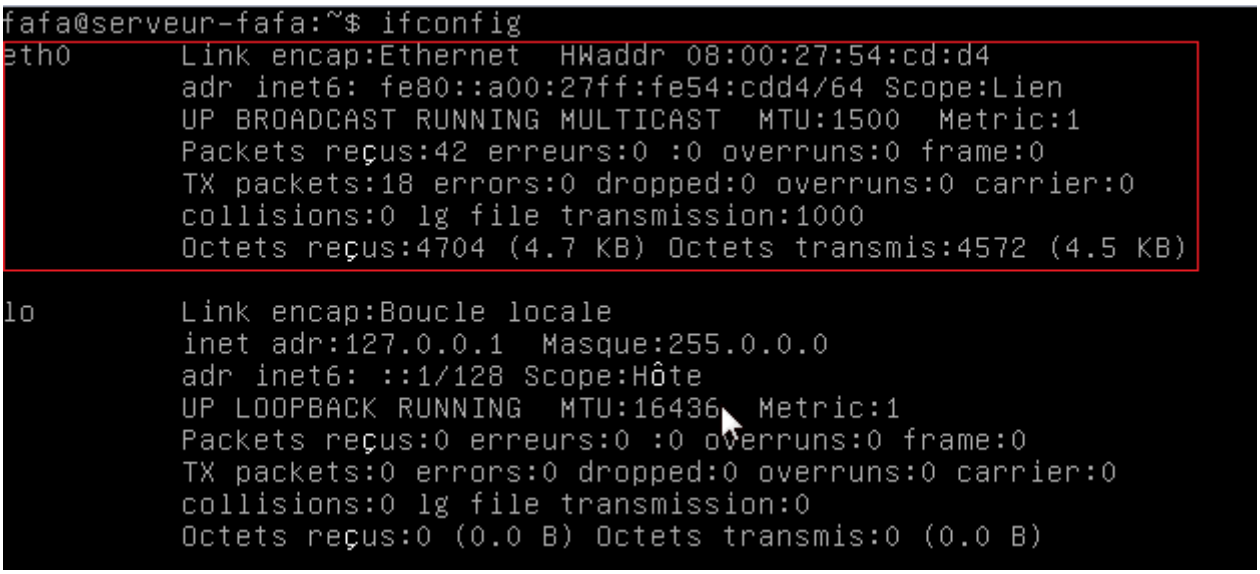
Configurer une carte réseau Ethernet dans Linux en ligne de commande est très utile pour les systèmes d'exploitation Linux n'ayant pas d'interface graphique. Par exemple, des serveurs de production pour mettre en place des services web.

L'objectif est de vous permettre de configurer une carte réseau Ethernet Linux en ligne de commande pour communiquer sur un réseau (domicile, entreprise, internet…) avec d'autres ordinateurs.

C'est indispensable si votre machine est utilisée comme serveur. Vous devez toujours avoir la même adresse IP. Si vous changez régulièrement d'IP, les autres ordinateurs auront des difficultés pour se connecter à votre serveur.

Dans un premier temps, vous pouvez vérifier la configuration actuelle des cartes Ethernet sur votre machine. Pour cela, utilisé la commande « ifconfig » qui va afficher les cartes réseau disponibles.

Sur ma machine (voir la capture ci-dessous), on peut constater qu'il y a deux cartes réseau. La première se nomme « eth0 » et la seconde « lo ». La carte « lo » est une carte local, elle n'est donc d'aucune utilité pour paramétrer notre carte réseau pour communiquer sur le réseau. La carte qui nous intéresse est donc **eth0**, pour le moment celle-ci n'est pas paramétrée, elle ne possède pas d'adresse IP. Commençons la configuration.



Pour ce faire modifier le paramètre de la carte réseau en le passant en « static ». Par la suite, nous devons lui renseigner les paramètres nécessaires à son fonctionnement.

'

Paramétrer La Carte Réseau Eth0

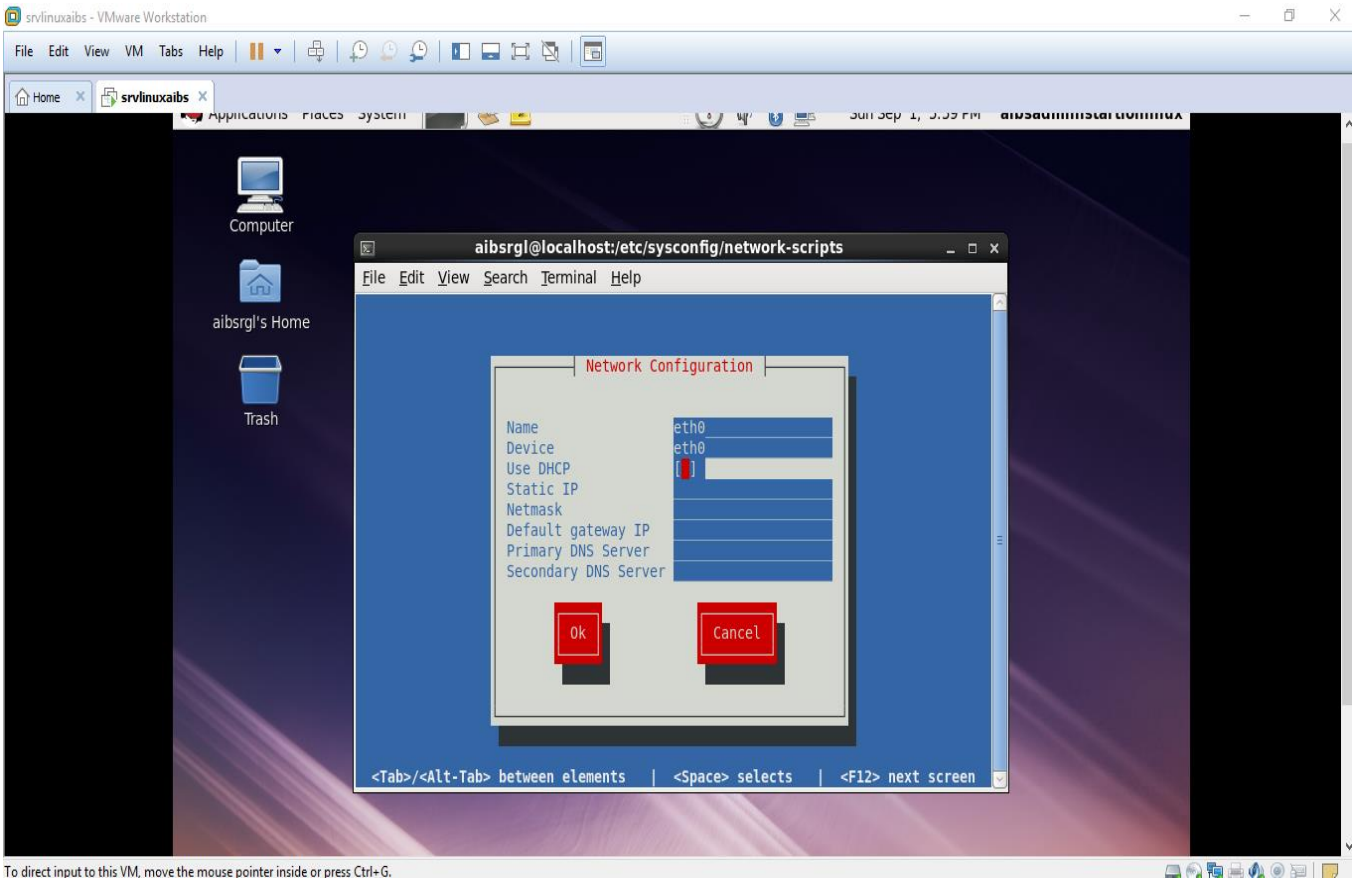
La première chose à faire est de se connecter en super-utilisateur (root)

[aibsrgl@localhost ~]$ su
Password:

Une fois connecté en super utilisateur,

[root@localhost network-scripts]# setup

Taper entrer ;puis network configuration ; puis device configuration ; puis eth0 ;
Puis utiliser la touch de tab pour aller à DHCP, uliser la touche d'espacement pour effacer son contenu ; puis entrer les informations comme ci dessous



Static Ip : 20.20.20.1
netmask 255.0.0.0
Default gateway IP 20.20.20.1
Primary DNS Server 202.83.21.12
Secondary DNS Server 202.83.21.12

'

Puis utiliser la touch de tab pour aller sur ok et faire entrer; puis save ; puis save&quit ; puis quit.

3-Configuration Serveur SAMBA (Partage de fichiers Linux et Windows)

Historiquement, les systèmes Windows utilisent leur propre protocole de partage de fichiers : SMB/CIFS. Le logiciel Samba a été développé pour permettre aux systèmes UNIX (dont Linux et Mac OS X) d'utiliser également ce protocole. Comme il est souvent plus facile d'utiliser Samba sous UNIX que de configurer NFS sous Windows, il est courant de préférer l'utilisation de Samba dans des environnements comportant des clients Windows.

Configuration en ligne de commande

Avant tout chose, vous devez installer le serveur samba par la commande selon votre distribution linux (yum install samba) pour notre cas ; Puis, pour configurer votre partage samba, il vous faudra éditer le fichier /etc/samba/smb.conf
Les changements ne prennent effet qu'une fois le démon Samba redémarré à l'aide de la commande service smb restart.

Nous preferons ne pas rentrer ici dans une configuration à prompement dite.Toutefois voici une demarche generale. Dans l'édition du fichier /etc/samba/smb.conf :

Pour spécifier le groupe de travail Windows et une brève description du serveur Samba, modifiez les lignes suivantes dans votre fichier smb.conf :

```
workgroup = WORKGROUPNAME

server string = BRIEF COMMENT ABOUT SERVER
```

Remplacez WORKGROUPNAME par le nom du groupe de travail Windows auquel cet ordinateur devrait appartenir. Le champ facultatif BRIEF COMMENT ABOUT SERVER est utilisé comme commentaire de Windows sur le système Samba.

Pour créer un répertoire de partage Samba sur votre système Linux, ajoutez la section suivante à votre fichier smb.conf (après l'avoir modifié en fonction de vos besoins et de votre système) :

```
[sharename]

comment = Insert a comment here

path = /home/share/

valid users = tfox carole

public = no
```

'

```
writable = yes

printable = no

create mask = 0765
```

Dans l'exemple ci-dessus, les utilisateurs **tfox et carole** peuvent lire et écrire dans le répertoire /home/share sur le serveur Samba, à partir d'un client Samba.

Démarrage et arrêt du serveur
Le service smb doit être en cours d'exécution sur le serveur partageant les répertoires via Samba.
Affichez le statut du démon Samba à l'aide de la commande suivante :

```
/sbin/service smb status
```

Démarrez le démon à l'aide de la commande suivante :

```
/sbin/service smb start
```

Arrêtez le démon à l'aide de la commande suivante :

```
/sbin/service smb stop
```

Pour lancer le service smb au démarrage, utilisez la commande suivante :

```
/sbin/chkconfig --level 345 smb on
```

4-Configuration Serveur DHCP

Un serveur DHCP (Dynamic Host Configuration Protocol) a pour rôle de distribuer de façon automatique, des adresses IP à des clients pour une durée déterminée.

Au lieu d'affecter manuellement à chaque hôte une adresse statique, ainsi que tous les paramètres tels que (serveur de noms, passerelle par défaut, nom du réseau), un serveur DHCP alloue à un client, un bail d'accès au réseau, pour une durée déterminée (durée du bail). Le serveur passe en paramètres au client toutes les informations dont il a besoin.

C'est quoi un bail ? Il s'agit d'un "contrat" passé entre le serveur et le client qui inclue notamment la durée de vie de l'adresse Ip qu'attribue le serveur au client.

'

Une fois le serveur démarré, un client voulant se connecter, diffuse un message dhcp-discover pour "trouver" un serveur DHCP. Lorsque le client trouve le serveur DHCP, ce dernier lui attribue automatiquement une adresse IP.

Les commandes utilisées peuvent différées quelques peut selon la distribution utilisée.

Avant de démarrer la configuration DHCP, assurez-vous que le "network manager", un outil simplifié pour la configuration d'un réseau sous Linux, soit désactivé. Par défaut, c'est bien le cas.

Voici un exemple

Configuration du serveur dhcp pour qu'il attribue un bail par défaut de 220s, d'une durée maximale de 360s, pour le nom de domaine sdz.net, avec une plage d'adresses IP allant de 192.168.21.30 à 192.168.21.70.

La configuration du serveur se fait dans un fichier où l'on va renseigner tous les paramètres nécessaires. Ce fichier est accessible en tapant cette ligne de commande (debian):

nano /etc/dhcp3/dhcpd.conf

Une fois le fichier ouvert, il suffit d'y ajouter les informations suivantes:

Option domain-name « sdz.net » ;

Default-lease-time 220 ;

Max-lease-time 360 ;

Subnet 192.168.21.0 netmask 255.255.255.0 {

Range 192.168.21.30 192.168.21.70

}

Remarquez bien les ";" il ne faut surtout pas les oublier.

Explications:

- Option domain-name « sdz.net » ;: Indiquez ici le nom de domaine que vous utiliserez.
- Default-lease-time 220 ;: Indiquez ici la durée du bail par défaut. Il s'agit du temps de vie d'une adresse IP. Il s'exprime en secondes.
- Max-lease-time 360 ;: Il s'agit du temps de vie maximum d'une adresse IP. Il s'exprime en secondes.
- Subnet 192.168.21.0 netmask 255.255.255.0 {: Indiquez sur cette ligne l'adresse réseau que vous souhaitez utiliser et son masque associé.
- Range 192.168.21.30 192.168.21.70 }: Il s'agit de la plage d'adresses IP que vous souhaitez utiliser dans le réseau 192.168.21.0

'

Pour finir il faut configurer manuellement l'adresse IP de notre serveur:

ifconfig eth0 192.168.21.2 netmask 255.255.255.0

Apres s’etre rassuré de la configuration tant du côté serveur que du côté client

La dernière étape, est la mise en service de notre serveur DHCP:

Du côté Serveur :

Il est nécessaire de démarrer notre serveur DHCP:

/etc/init.d/dhcp3-server start

Du côté client :

Il est nécessaire de redémarrer les interfaces réseaux de chaque ordinateur client afin qu'elles se voient attribuer une adresse IP.

Vous pouvez le faire de cette façon, que vous devez connaître maintenant : /etc/init.d/networking restart

Toujours du côté client, pensez à vérifier que des processus DHCP ne tournent pas. Sous Linux Debian, ceux-ci sont "tuer" automatiquement, mais sous Fedora par exemple, il est nécessaire de vérifier.

Vérification de l'adresse: Constatez par vous-même que votre adresse IP a bien été prise en compte sur chaque client : ifconfig eth0