

Activité : Installation Client

© 2018-2024 tv <tvaira@free.fr> - v.1.1.1

Système d'exploitation	2
GNU/Linux	4
L'interface homme-machine (IHM)	5
Distribution GNU/Linux	6
Paquet logiciel	7
Ubuntu	8
Distribution live	8
Notions de base	9
BIOS et UEFI	9
Partitions	10
Chargeur d'amorçage (<i>bootloader</i>)	12
Systèmes de fichiers	12
Installation	14
Types d'installation	14
Planification	14
Pré-installation	14
Installation	15
Post-installation	15
Questions de révision	16
Travail demandé	17
Installation	17
Post-installation	19
Obtenir des informations sur le système	19
Administrateur (super utilisateur)	19
Bonus : expérimentations	21

Activité

L'objectif de cette activité est de découvrir les principes d'une installation d'un système d'exploitation GNU/Linux : Ubuntu.

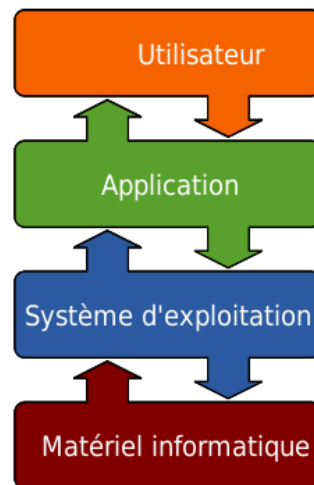


Linux (Tux) - GNU - Ubuntu

Système d'exploitation

Définition

Un système d'exploitation (*Operating System*) est _____

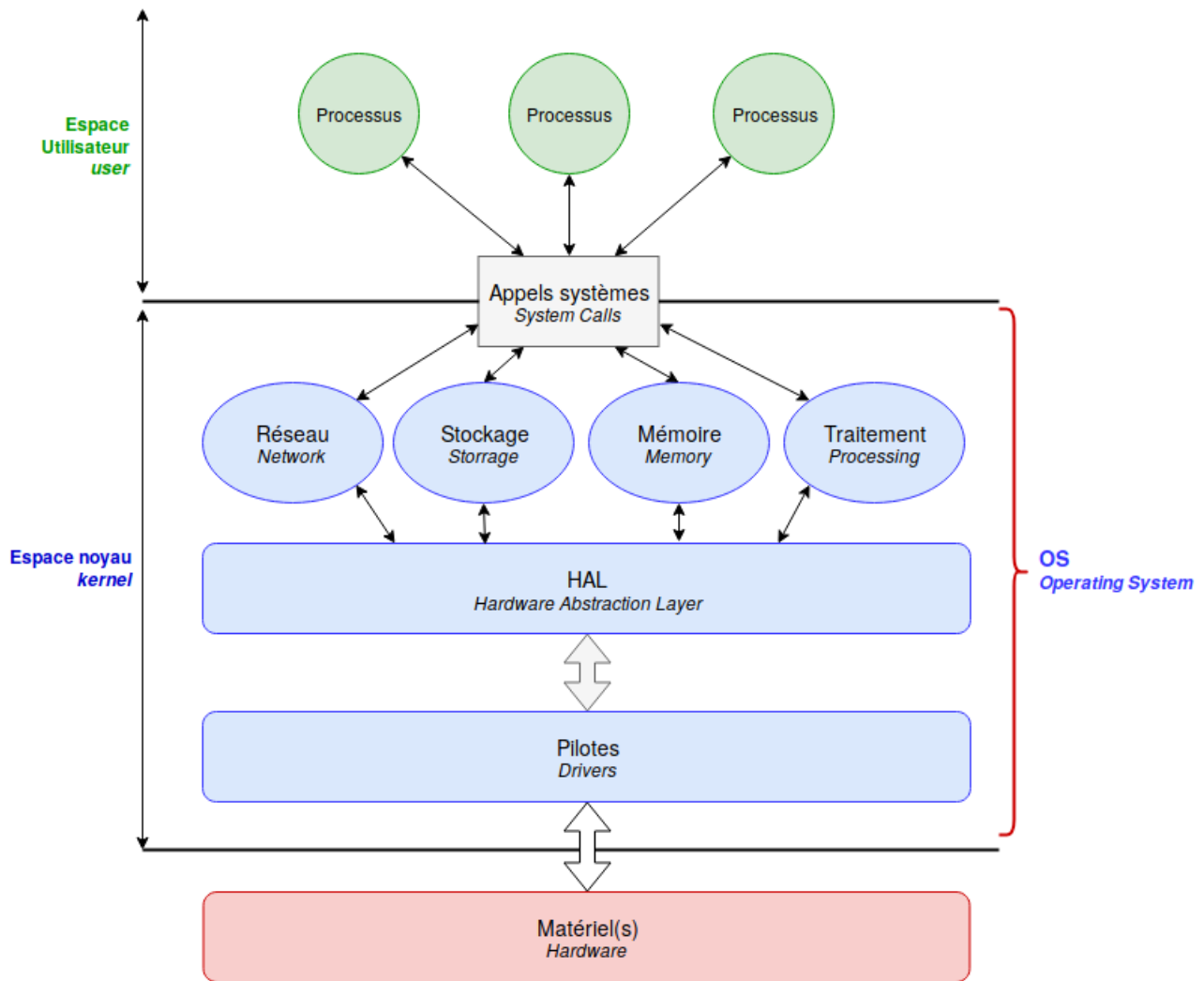


De manière générale, un système d'exploitation :

- permet l'exploitation des périphériques matériels dont il coordonne et optimise l'utilisation ;
- propose aux logiciels applicatifs des interfaces de programmation standardisées qui simplifient l'utilisation des matériels et des services qu'il offre ;
- coordonne l'utilisation du ou des processeur(s), et accorde un certain temps pour l'exécution de chaque processus (multi-tâche) ;
- gère l'espace mémoire pour les besoins des programmes ;
- organise le contenu des disques durs ou d'autres mémoires de masse en fichiers et répertoires ;
- fournit les interfaces homme-machine des différents programmes ;
- réalise enfin différentes fonctions visant à assurer la fiabilité (tolérance aux pannes, isolation des fautes) et la sécurité informatique (traçabilité, confidentialité, intégrité et disponibilité).

Définition

Un processus est _____



Structure (très) simplifiée d'un OS GNU/Linux

Ressources :

- <http://tvaira.free.fr/bts-sn/linux/cours/cours-os.pdf>
- <http://tvaira.free.fr/os/CoursOS.pdf>

GNU/Linux

Définition

GNU/Linux est le nom parfois donné à un système d'exploitation associant des éléments essentiels du projet GNU et d'un noyau Linux.

Linux est le nom couramment donné à tout système d'exploitation libre fonctionnant avec le noyau (*kernel*) Linux. C'est une implémentation libre du système **UNIX** respectant les spécifications **POSIX** (normes techniques de l'IEEE).

Le projet **GNU** (*GNU's Not UNIX*) regroupe tous les éléments nécessaires à la construction d'un système d'exploitation (un *shell*, des bibliothèques, les composants de base, les outils de développement ...).



GNU est en réalité un système d'exploitation libre créé en 1983 par Richard Stallman et maintenu par le projet GNU. Le projet GNU a tout d'abord possédé une version utilisable de tous les éléments nécessaires à la construction d'un système d'exploitation à l'exception du plus central : le noyau. Le projet GNU a initié ensuite la production d'un noyau nommé Hurd. Le développement de Hurd ayant du mal à aboutir, GNU reste donc toujours incomplet. Les logiciels du GNU sont utilisés depuis longtemps, mais habituellement avec le noyau Linux.

Le noyau Linux a été initialement écrit par **Linus Torvalds**, un étudiant finlandais au début des années 1990.

La terminologie GNU/Linux a été créée par le projet **Debian** et reprise notamment par **Richard Stallman**, à l'origine du projet de travail collaboratif GNU et de la licence libre GPL (*General Public Licence*).



Android est un système basé sur Linux mais pas sur GNU.

Une des différences essentielles de Linux par rapport à d'autres systèmes d'exploitation concurrents (comme Mac OS, Microsoft Windows et Solaris) est d'être un système d'exploitation libre, apportant quatre libertés aux utilisateurs définies par la licence GNU GPL :

- « utiliser le logiciel sans restriction »
- « étudier le logiciel »
- « modifier pour l'adapter à ses besoins »
- « redistribuer sous certaines conditions précises »



Un logiciel libre n'est pas nécessairement gratuit, et inversement un logiciel gratuit n'est pas forcément libre.

Le système avec les applications est le plus souvent fourni sous la forme de **distributions Linux** comme Slackware, Debian, Red Hat, Fedora, Mandriva, Ubuntu ...

Ressources :

- <http://tvaira.free.fr/bts-sn/linux/cours/cours-linux.pdf>
- <http://tvaira.free.fr/bts-sn/linux/cours/cours-gnu-linux.pdf>

L'interface homme-machine (IHM)

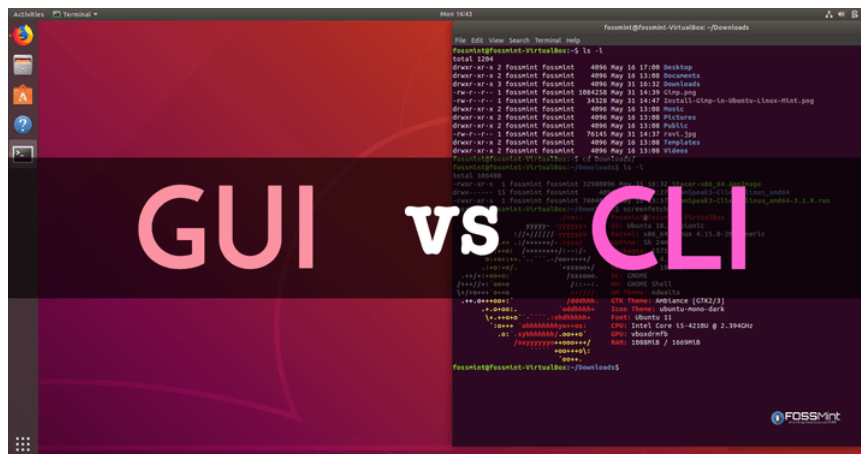
Définition


L'interface homme-machine (**IHM**) permet à un utilisateur de dialoguer avec la machine.

Deux types : **GUI** (*Graphical User Interface*) et **CLI** (*Command Line Interface*).

On distingue deux types d'IHM fournies par un système d'exploitation :


- **GUI** (*Graphical User Interface*) ou « interface utilisateur graphique » : les parties les plus typiques de ce type d'environnement sont le pointeur de souris, les fenêtres, le bureau, les icônes, les boutons, les menus, les barres de défilement, ... Les systèmes d'exploitation grand public (Windows, MacOS, GNU/Linux, etc.) sont pourvus d'une interface graphique qui, dans un souci d'ergonomie, se veut conviviale, simple d'utilisation et accessible au plus grand nombre pour l'usage d'un ordinateur personnel.
- **CLI** (*Command Line Interface*) ou « interface en ligne de commande » est encore utilisée en raison de sa puissance, de sa grande rapidité, son uniformité, sa stabilité et du peu de ressources nécessaires à son fonctionnement. Le système d'exploitation permet cette possibilité par l'intermédiaire d'un interpréteur de commandes (le *shell*). Beaucoup de serveurs ne s'administrent qu'en ligne de commande.



 Un terminal est le nom d'un logiciel d'émulation de terminal. Un émulateur de terminal, aussi appelé console virtuelle ou terminal virtuel, est un logiciel qui émule le fonctionnement d'un terminal informatique en mode CLI ce qui permet d'interagir avec les commandes lorsqu'on utilise un environnement graphique. Il peut être local ou distant (telnet, ssh, ...).

Tous les exemples d'exécution des commandes sont précédés d'une **invite** utilisateur ou **prompt** spécifique au niveau des droits utilisateurs nécessaires sur le système :

- toute commande précédée de l'invite **\$** ne nécessite aucun privilège particulier et peut être utilisée au niveau utilisateur simple ;
- toute commande précédée de l'invite **#** nécessite les privilèges du super-utilisateur (*root*). Le compte *root* est le compte de l'administrateur du système.

 Évidemment, il ne faudra jamais taper l'invite (**\$** ou **#**) lorsque vous testez par vous même les commandes indiquées.

Distribution GNU/Linux

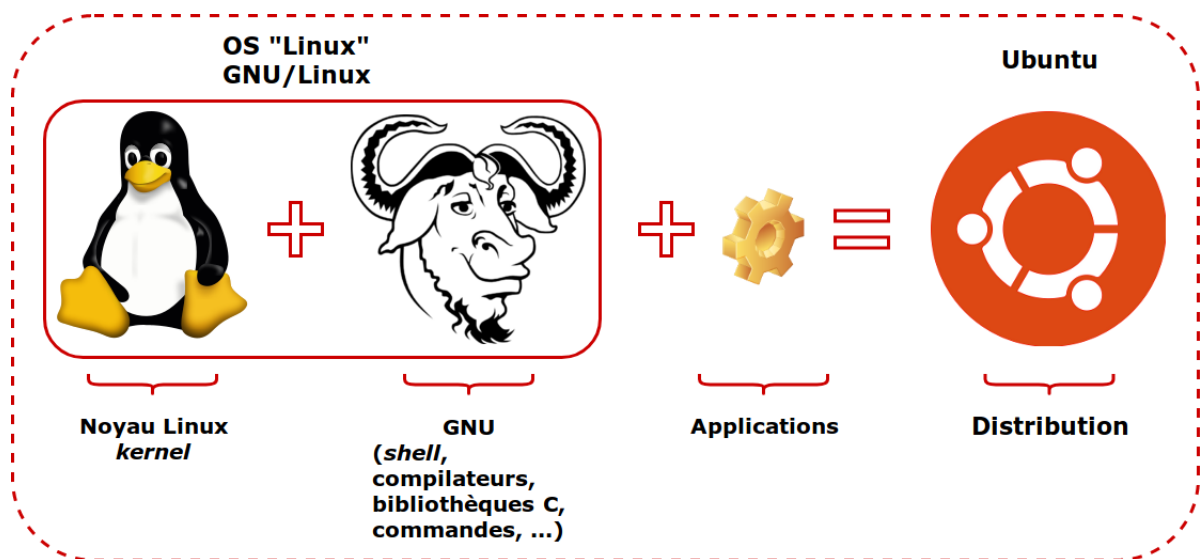
Définition

Une **distribution GNU/Linux** (ou distribution Linux) est un ensemble cohérent de logiciels, la plupart étant des logiciels libres, assemblés autour du noyau Linux.



Le terme « distribution » est calqué sur l'anglais *software distribution* qui signifie « collection de logiciels » en français.

On appelle donc **distribution GNU/Linux** : une solution prête à être installée par l'utilisateur final comprenant le système d'exploitation GNU, le noyau Linux, des programmes d'installation et d'administration de l'ordinateur, un mécanisme facilitant l'installation et la mise à jour des logiciels comme RPM ou APT (gestionnaires de paquets) ainsi qu'une sélection de logiciels produits par d'autres développeurs.



Liste des distributions Linux :

- https://fr.wikipedia.org/wiki/Liste_des_distributions_Linux
- https://commons.wikimedia.org/wiki/File:Linux_Distribution_Timeline.svg?uselang=fr



Les logiciels libres sont développés de manière collaborative, souvent indépendamment les uns des autres, et peuvent être librement redistribués. Il s'ensuit une particularité du monde GNU/Linux : la séparation fréquente entre ceux qui développent les logiciels et ceux qui les distribuent.

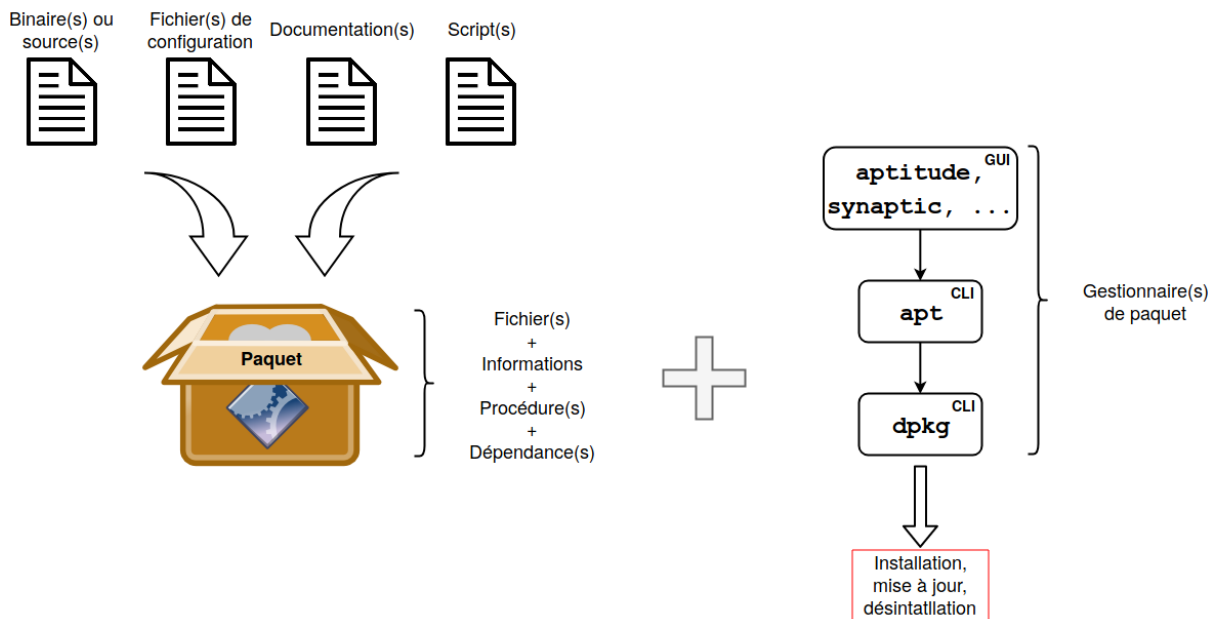
Paquet logiciel

Définition

Un **paquet** ou parfois paquetage (*package*) est une archive (souvent compressée) comprenant les fichiers, les informations et procédures nécessaires à l'installation d'un logiciel sur un système d'exploitation en s'assurant de la cohérence fonctionnelle du système ainsi modifié.

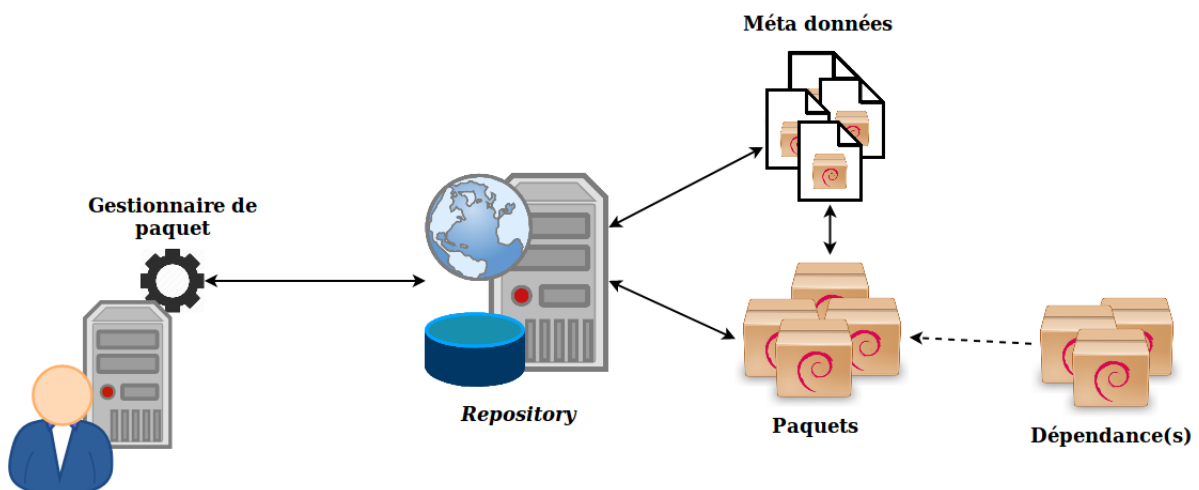


L'utilisation d'un paquetage logiciel est un élément constitutif d'une bonne pratique d'intégration logicielle.



Les types de paquets les plus répandus sous Linux sont deb (DEbian) et rpm (*Red Hat Package*). Voir aussi : tgz (fichier archive créé avec tar et ensuite compressé généralement avec gzip), msi (pour les fichiers d'installation pour *Windows Installer*).

Pour les distributions Linux, les paquets sont souvent centralisés sur des **sites dépôts** (*repository*) ce qui facilite l'installation des applications pour les utilisateurs.



Ubuntu

Définition

Ubuntu est un système d'exploitation GNU/Linux basé sur la distribution Linux Debian.



Il est développé et maintenu par la société Canonical.

Chaque version a un nom de code et un numéro. La numérotation des versions d'Ubuntu est chronologique (contrairement à la numérotation ordinale habituelle : 1.0, 1.1, ... ; 2.0 ; 3.0 ...) :

- le premier nombre correspond à l'unité des années (12 pour 2012, 14 pour 2014, ...) et,
- le second correspond au mois de la publication de la version stable (04 pour avril et 10 pour octobre)



Tous les deux ans (en avril des années paires), la version est une version avec support à long terme dit LTS (*Long Term Support*) qui est maintenue pendant cinq ans.

Exemple : 16.04 LTS (Nom de code : Xenial Xerus), 18.04 LTS (Nom de code : Bionic Beaver)

Sites : <http://ubuntu.com/> et <https://ubuntu-fr.org/>

Distribution live

Définition

Une **distribution live** est un système d'exploitation informatique exécutable depuis un support amovible, le plus souvent à partir d'une clé USB ou d'un CD-ROM (ou un disque dur externe).



La tendance actuelle est au développement de distribution live sur mémoire *flash*, c'est-à-dire des clés USB ou des cartes mémoires. On appelle ces versions des *liveUSB*.

On distingue :

- les généralistes : elles donnent accès à toutes les fonctions de base d'un système d'exploitation (internet, bureautique, multimédia, etc).
- les compactes : elles ont l'avantage de pouvoir tenir sur une petite clé USB ou un CD de format réduit et peuvent également faire fonctionner un ordinateur ayant peu de mémoire vive ou un disque dur de petite capacité.
- les spécialisés : ils sont dédiés à un domaine particulier : multimédia, maintenance, sécurité, serveurs
 - SystemRescueCd : distribution compacte, récupération de donnée et maintenance.
 - Ultimate Boot CD : nombreux outils de dépannage informatique pour Linux ou Windows.
 - BackTrack : outils pour les tests de sécurité d'un réseau (*wardriving*, outils WiFi, etc).

Ressources :

- https://fr.wikipedia.org/wiki/Liste_de_live_CD
- https://fr.wikipedia.org/wiki/Live_USB
- https://fr.wikipedia.org/wiki/Mini_Linux

Notions de base

BIOS et UEFI

Définition

Le **BIOS** (*Basic Input Output System*) est à l'origine un ensemble de fonctions, fournies par la carte mère, permettant d'effectuer des opérations de base sur le matériel. Maintenant, le terme BIOS fait plutôt référence au micrologiciel (*firmware*) de contrôle de la carte mère.

Le BIOS permet l'**amorçage** (*boot*) de l'ordinateur :

- La première phase est l'auto-configuration à l'allumage POST (*Power-On Self-Test*) qui teste la présence des divers périphériques et leur attribue les ressources nécessaires à un fonctionnement sans conflit.
- La deuxième phase (une fois le POST terminé) sert à trouver un périphérique amorçable disponible puis de charger et exécuter le chargeur d'amorçage (*bootloader*) dont le rôle est de démarrer le système d'exploitation.



Il est voué à être remplacé par l'UEFI (*Unified Extensible Firmware Interface*).

Définition

Le standard UEFI (*Unified Extensible Firmware Interface*) définit un logiciel intermédiaire entre le micrologiciel (*firmware*) de la carte mère et le système d'exploitation (OS) d'un ordinateur. Cette interface succède sur certaines cartes-mères au BIOS.



L'UEFI intègre maintenant la fonctionnalité « lancement sécurisé » (*secure boot*) qui n'autorise le démarrage qu'aux systèmes d'exploitation reconnus (par vérification de signatures numériques). Cela a posé des problèmes pour l'installation de Linux et des logiciels libres.

Ressources :

- https://fr.wikipedia.org/wiki/Basic_Input_Output_System
- https://fr.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface

Partitions

Définitions

Une **partition** est une région (une fraction) d'un espace d'un support de stockage (disque dur, SSD, carte-mémoire ...).

Le **partitionnement** d'un support de stockage est l'opération qui consiste à découper son espace en partitions.

Une partition est généralement destinée à accueillir un **système de fichiers**.

Les informations (les adresses de début et de fin, son type, sa taille, ...) sur les partitions sont conservées sur le support lui-même dans des zones appelées **tables de partitions**.

Formater (une partition) est l'opération qui consiste à installer un système de fichiers sur une partition.



L'outil « historique » pour créer des partition est `fdisk`.



Même si on installe qu'un système sur un disque dur, il faudra créer une partition. Si on installe deux systèmes, il faudra donc créer au moins deux partitions. Il y a un intérêt concret et des limites à créer des partitions.

On nomme « partition d'amorçage » (partition active), celle qui prend le contrôle au démarrage, qu'elle contienne ou non le système d'exploitation.



Le *swap* sert à étendre la mémoire physique (RAM) utilisable par un système d'exploitation, par un fichier d'échange (Windows) ou une partition dédiée (GNU/Linux). La taille de la partition de swap est habituellement fixée à une valeur égale (ou le double) à la quantité de RAM. Pour installer un système Linux, il faudra au moins 2 partitions : une pour le système de fichiers racine (*root*) et une autre pour le *swap*.

Un support de stockage peut être partitionné suivant différentes architectures :

- MBR (partitionnement Intel) longtemps employé sur la majorité des PC pour les supports d'une capacité inférieure ou égale à 2 To. MBR (*Master Boot Record*) est le nom donné au premier secteur (512 octets) adressable d'un disque dur. Il contient la table des partitions (4 partitions primaires max.) du disque dur et une routine d'amorçage. Cette routine d'amorçage cherche dans la table de partitions la partition active afin de charger le système d'exploitation ou le chargeur d'amorçage (*boot loader*).
- GPT (*GUID Partition Table*) pour des architectures plus récentes (Macintosh, Linux, et PC depuis les années 2010) conçues pour exploiter pleinement les supports de capacité supérieure à 2 To. GPT (*GUID Partition Table*) est un standard (de l'UEFI) pour décrire la table de partitionnement d'un disque dur.

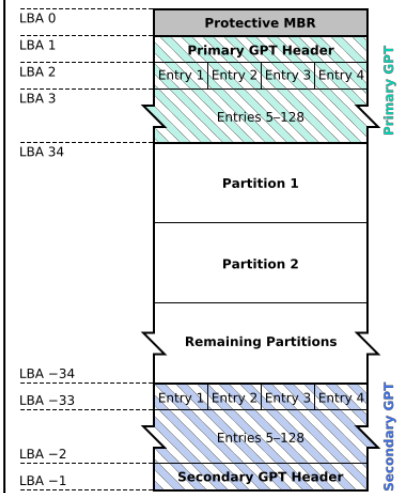
Structure du Master Boot Record :

Adresse		Description	Taille en octets
Hex	Déc		
0000	0	Routine	max. 444
01B8	440	Signature optionnelle	4
01BC	444	Habituellement nul ; 0x0000	2
01BE	446	Table des partitions primaires (Quatre entrées de 16 octets, (IBM Partition Table scheme))	64
01FE	510	MBR signature; 0xAA55	2
01FF	511		
Taille totale du MBR : 444 + 2 + 64 + 2 =			512

Structure d'une entrée dans la table de partition

Adresse	Contenu	Taille (octets)
0x00	État de la partition : - 00 : partition non active - 80 : partition active	1
0x01	N° de tête où commence la partition	1
0x02	N° de secteur et cylindre où commence la partition	2
0x04	Type de partition	1
0x05	N° de tête où finit la partition	1
0x06	N° de secteur et cylindre où finit la partition	2
0x08	Distance en secteurs entre secteur de partition et secteur de boot de la partition	4
0x0C	Taille de la partition en nombre de secteurs de 512 octets	4

GUID Partition Table Scheme



Dans le MBR, on peut créer au plus 4 partitions, qui sont soit 4 partitions primaires, soit 3 partitions primaires et une partition étendue. Cette partition étendue peut contenir à son tour des partitions secondaires (souvent appelées partitions logiques).

Identification des partitions :

- Adressage matériel (/dev/sda1, ...)
- Adressage par étiquette (« label »)
- Adressage par UUID (*Universally Unique Identifier*)



Un GUID (*Globally Unique Identifier*) sert habituellement d'identifiant unique pour un composant logiciel. Sa taille est de 16 octets, soit 128 bits. Ce terme est utilisé à la fois dans le monde Microsoft et dans le monde Unix. Sous Unix, on utilise le UUID.

```
$ sudo blkid
/dev/sda1: UUID="2a31d147-5559-477c-8a99-f9660b21c55b" TYPE="ext4"
/dev/sda5: UUID="5eeb1799-3aa4-4dae-b216-f1664aeed450" TYPE="swap"
/dev/sdb1: LABEL="disk1" UUID="dbe734d4-ab29-4ca1-b994-ab780a695c05" TYPE="ext4"
/dev/sde1: LABEL="TVaira" UUID="5ADEC4FBDEC4D085" TYPE="ntfs"
$
```

Chargeur d'amorçage (*bootloader*)

Définition

Un **chargeur d'amorçage** (*bootloader*) est un logiciel permettant de lancer un système d'exploitation, même parmi plusieurs (*multi-boot*). Cela permet d'utiliser plusieurs systèmes, à des moments différents, sur la même machine.

Dans le cas le plus simple, le micrologiciel BIOS charge les 512 premiers octets de ce disque, ces 512 octets constituant le MBR. À partir des informations du MBR, le micro-logiciel (routine) de 442 octets détermine l'emplacement de la partition d'amorçage.



Sur certains PC actuels, c'est le micrologiciel EFI (et non pas le BIOS) qui est utilisé pour lancer le chargeur d'amorçage : l'EFI lit la GPT du disque (*GUID Partition Table*) pour déterminer l'emplacement de la routine d'amorçage. Le standard UEFI (*Unified Extensible Firmware Interface*), qui fait suite à EFI, définit un logiciel intermédiaire entre le micrologiciel (*firmware*) et le système d'exploitation (OS) d'un ordinateur. Une des fonctions d'UEFI est l'amorçage d'un système d'exploitation, auparavant assurée par le BIOS.

Les chargeurs d'amorçage les plus usuels sont :

- Microsoft : NTLDR (*NT LoaDeR* ou Chargeur d'amorçage de Windows NT), IA86ldr.efi et IA64ldr.efi avec l'EFI, winload.exe (Vista)
- Open source : GRUB (*GRand Unified Bootloader*), LILO (*Linux loader*) pour le BIOS et elilo pour EFI, IsoLinux pour booter à partir d'un DVD ISO 9660 et PXELinux pour booter à partir d'une carte réseau
- Apple : Boot Camp



Sur les machines de BTS, on utilisera GRUB.

Systèmes de fichiers

Définitions

Un **système de fichiers** (*file system*) définit l'organisation d'une partition.

C'est une structure de données permettant de stocker les informations et de les organiser dans des **fichiers**. Le fichier est la plus petite entité logique de stockage sur un système.

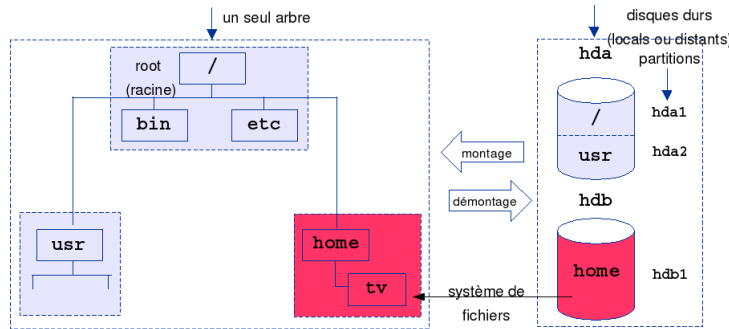
Il offre à l'utilisateur une vue abstraite sur ses données et permet de les localiser à partir d'un **chemin d'accès** (*path*).

Le **formatage** est l'action de formater, c'est-à-dire de préparer une partition en y inscrivant un système de fichiers, de façon à ce qu'il soit reconnu par le système d'exploitation de l'ordinateur.

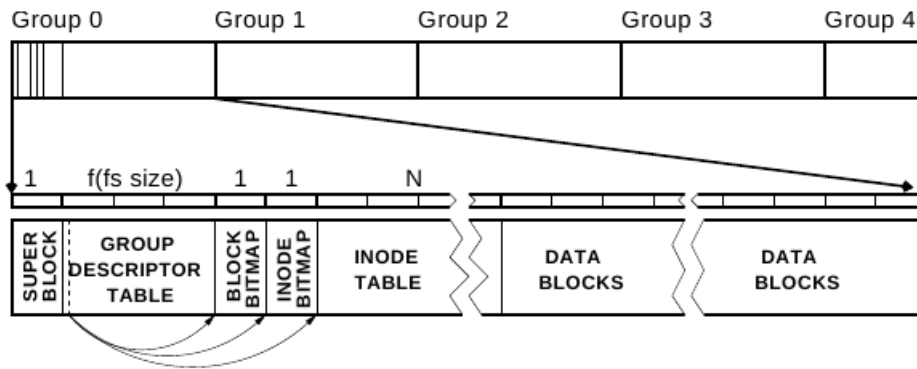
Il existe de nombreux systèmes de fichiers différents : FAT, NTFS, HFS, ext4, UFS, etc.



Sous GNU/Linux, les fichiers sont « vus » dans une arborescence unique possédant une racine /. Cet arbre est en fait l'unification de plusieurs systèmes de fichiers réalisé par des opération de montage (mount).



Un système de fichiers définit l'organisation permettant le stockage de fichiers sur une partition. Dans le cas d'un système de fichiers *ext2*, on aura l'organisation suivante :



Exemple de structure d'un système de fichiers (*ext2*)



ext2 est un système de fichiers historique de GNU/Linux. Il a été créé à l'origine par Rémy Card, un développeur français. Sur la version d'Ubuntu à installer, on utilisera le système de fichiers *ext4*. `mkfs.type` est une commande qui permet de formater une partition avec un type de système de fichiers donné.

Chaque fichier est décrit par des **métadonnées** (conservées dans l'**inode** sous GNU/Linux), alors que le contenu du fichier est écrit dans un ou plusieurs blocs du support de stockage, selon la taille du fichier. Sur la plupart des systèmes Unix, la commande `stat` permet d'afficher le contenu de l'inode.

<pre>\$ stat .profile Fichier : .profile Taille : 807 Blocs : 8 Blocs d'E/S : 4096 fichier Périphérique : 801h/2049d Inodeud : 25165827 Liens : 1 Accès : (0644/-rw-r--r--) UID : (1026/ tv) GID : (65536/ tv) Accès : 2018-08-20 16:41:16.724499879 +0200 Modif. : 2018-07-17 13:21:11.283040393 +0200 Changt : 2018-07-18 16:17:22.141874816 +0200 Créé : -</pre>	UN INODE
<p>Informations :</p> <ul style="list-style-type: none"> Mode (= type de fichier) Droits d'accès et type de fichier UID du propriétaire et GID du groupe du fichier Taille en octets Date et heure de création Date et heure de la dernière modification Date et heure de suppression Nb de liens physiques Nb de blocs ... 	↑ ↓ 128 octets
<p>12 adresses de blocs de données</p> <p>Adresse du bloc d'indirection de niveau 1 <small>contient l'adresse d'un bloc d'adresse</small></p>	
<p>Adresse du bloc d'indirection de niveau 2 <small>contient l'adresse d'un bloc d'adresse de blocs d'adresse</small></p> <p>Adresse du bloc d'indirection de niveau 3</p>	

Exemple d'inode dans un système de fichiers (*ext2*)

Installation

Types d'installation

L'installation d'un système d'exploitation peut se réaliser de différentes manières :

- à partir d'un CD/DVD
- à partir d'un CD/DVD personnalisé (on utilise parfois le terme de *master*)
- à partir du réseau (installation partielle ou complète mais dans ce cas il faut régler le problème du démarrage initial cf. PXE)
- à partir d'une image (ISO par exemple)
- depuis un disque dur (Linux ou Windows)
- depuis une clé USB
- depuis un boot sur disquette ;)

De manière générale, on peut réaliser une installation depuis un CD/DVD ou clé USB, un dépôt local, un disque dur, le réseau ...

Planification

Il est aussi important de planifier son installation car celle-ci est une opération qui peut s'avérer longue et complexe à déployer.

Il faut généralement être capable de répondre aux questions suivantes :

- Quel système ? Quelle version ?
- Combien de postes ?
- Quelle est la source de l'installation (CD/DVD, USB, image, réseau, ...) ?
- Installation automatique, semi-automatique ou manuelle ?
- Les postes sont-ils identiques au niveau matériel ?
- L'utilisation de ces postes sera-t-elle la même pour tous ? Quel profil ? Quelles logiciels déployer ?
- La configuration post-installation est-elle automatique ou nécessite-t-elle une intervention manuelle ?
- etc ...



Dans le cas d'un serveur (ou plus) cela peut s'avérer encore plus délicat : prévoir une phase de test avant basculement, prévoir le basculement, prévoir un retour en cas d'anomalie, etc ...

Pré-installation

Avant d'installer un système d'exploitation, il est important d'avoir le maximum d'informations sur le matériel de la machine cible. Pour cela, on peut utiliser une **distribution live** ou parfois le système existant.


Les sources d'informations de base sur le matériel sous GNU/Linux sont :

- `/etc` : répertoire contenant les fichiers (ASCII) de configuration de la machine
- `/proc` : pseudo-système de fichiers d'informations sur le noyau (interface avec les données systèmes du noyau de l'OS)
- `/var/log` : répertoire contenant les fichiers de journalisation de la machine
- des commandes : `dmesg`, `uname`, `hostname`, `ifconfig`, `lsusb`, `lspci`, `df`, `free`, `hdparm`, `dmidecode`, `lshw`, ...

Installation

Quelque soit le système d'exploitation à installer, on rencontrera les étapes suivantes :

- démarrage de l'installation
- partitionnement du disque et formatage de(s) partition(s)
- copie des fichiers
- configuration du système
- finalisation de l'installation


 Il est parfois nécessaire de passer par le BIOS de la machine pour définir les périphériques de démarrage et leur ordre. L'accès au BIOS se fait généralement par l'appui sur une touche (Suppr., F1, F2, F8 à F12, ...) du clavier. Parfois, un message indiquant l'accès au BIOS s'affiche au démarrage. D'autre part, il est parfois nécessaire de désactiver *secure boot* pour installer GNU/Linux avec l'UEFI.

Il faut insérer un disque contenant l'ISO et *booter* sur le lecteur CD/DVD. L'utilisation d'une clé USB est aussi possible.

Post-installation

Une fois le système installé et redémarré, il faut s'assurer que :

- le nouveau système installé se lance et s'exécute correctement
- les autres systèmes installés se lancent et s'exécutent toujours correctement
- il est possible d'ouvrir une session locale (compte *iris*) et distante (votre compte)
- les logiciels nécessaires sont présents et s'exécutent correctement

 Il est souvent nécessaire de faire certaines mises à jour.

Questions de révision

Question 1. Qu'est-ce qu'un système d'exploitation ?

Question 2. Qu'est-ce Linux ?

Question 3. Qu'est-ce qu'un noyau (*kernel*) ?

Question 4. Qu'est-ce qu'une distribution GNU/Linux ?

Question 5. Qu'est-ce qu'un paquet (*package*) sous GNU/Linux ?

Question 6. Qu'est-ce qu'une GUI ? Qu'est-ce qu'une CLI ?

Question 7. Qu'est-ce qu'un chargeur d'amorçage (*bootloader*) ?

Question 8. Qu'est-ce qu'une partition ?

Question 9. Qu'est-ce qu'un système de fichiers ?

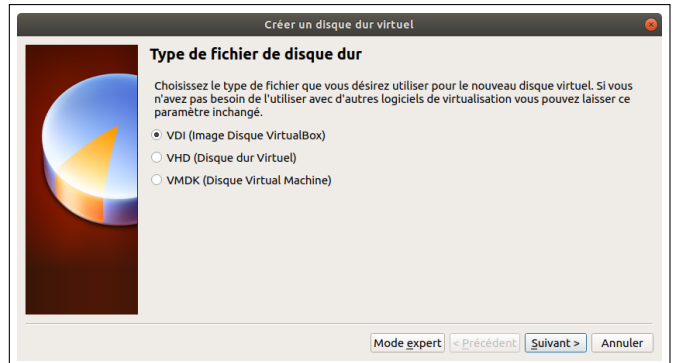
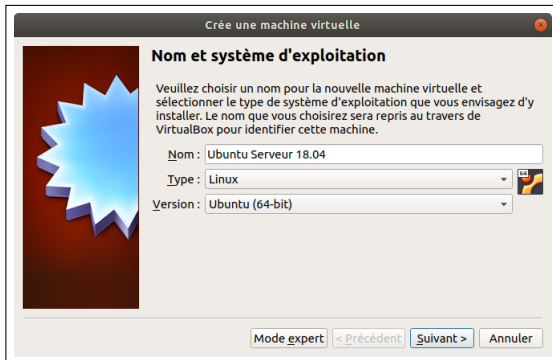
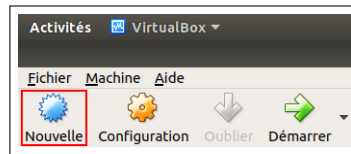
Question 10. Qu'est-ce qu'un fichier ? Qu'est-ce qu'un inode ?

Travail demandé

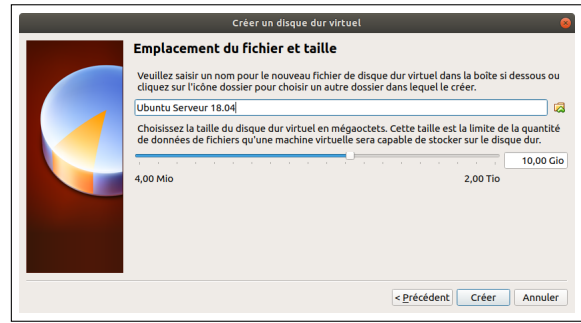
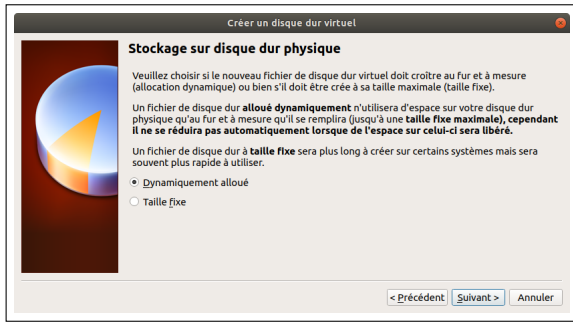
Installation

Vous devez réaliser l'installation d'une version Desktop (client) LTS d'Ubuntu (une 22.04 par exemple) sur une machine virtuelle (VirtualBox). Le compte Utilisateur sera **bts2** avec le mot de passe **lasalle84**.

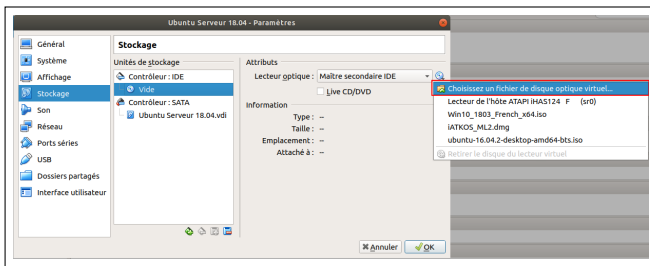
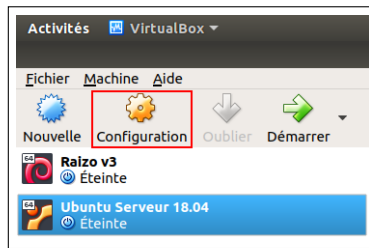
Démarrer VirtualBox et créer une nouvelle machine virtuelle.



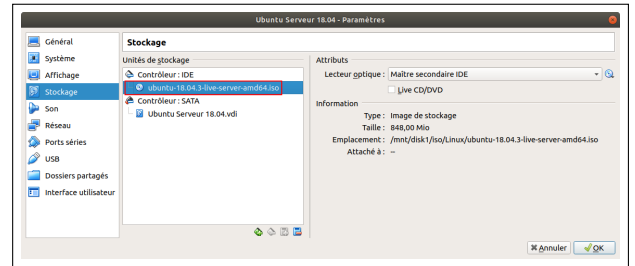
Une installation minimale nécessite au moins 4 GO d'espace disque.



Configurer votre machine virtuelle.

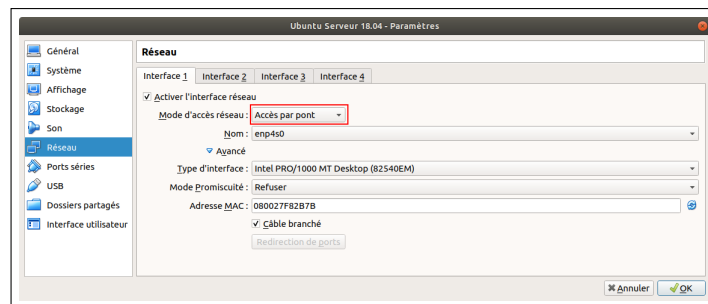


Sélectionner l'image ISO fournie

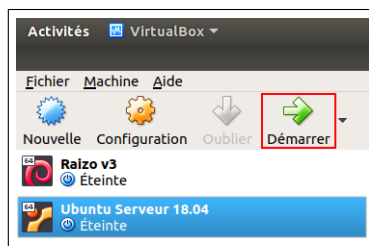


Valider

➡ Passer ensuite l'interface réseau de la machine virtuelle en mode "Accès par pont" (dans l'onglet "Réseau"). :



Démarrer l'installation.



Post-installation

➡ Après avoir redémarré, vous pouvez ouvrir une session avec le compte `bts2` et le mot de passe `lasalle84`.



Pour obtenir la **page de manuel** sur une commande, il faut taper par exemple :

```
$ man apt-get
```

Obtenir des informations sur le système

Question 11. Afficher la version du noyau Linux à partir de la commande `uname`.

Question 12. Quelle est la version de la distribution Ubuntu installée ? Utiliser la commande `cat` pour visualiser le contenu des fichiers `cat /etc/lsb-release` et `cat /etc/os-release`.

Question 13. Afficher page par page la liste des paquets installés. Utiliser la commande `dpkg -l | more`. Que permet de faire la commande `dpkg -l | grep -E "^ii" | wc -l`



Il est possible de rediriger le flux de sortie d'une commande vers une autre en utilisant un tube (*pipe*). Un **tube** (`|`) est un canal entre deux processus (redirection de la sortie d'un processus vers l'entrée d'un autre processus).



La commande `grep` permet de filtrer des lignes de texte à partir d'un motif (*pattern*). Par exemple :

```
$ dpkg -s bash | grep -i "status"
```

Question 14. Afficher les informations sur le paquet `bash` ? Utiliser la commande `dpkg -s bash`. Qu'est-ce que `bash` ? Que permet de faire la commande `dpkg -L bash` ?

Question 15. Quel est le CPU de votre machine ? Visualiser le fichier `/proc/cpuinfo` puis afficher seulement la ligne `"model name"` avec une commande.

Question 16. Quelle est la quantité de mémoire RAM installée sur votre machine ? Utiliser la commande `free` avec un affichage approprié des unités.

Question 17. Indiquer l'espace occupé par les systèmes de fichiers en limitant l'affichage aux systèmes locaux. Utiliser la commande `df` avec un affichage des unités du système international. Existe-t-il une unité spécifique à l'Informatique ?

Administrateur (super utilisateur)

L'utilisation courante ne demande pas de privilèges particuliers. Par contre pour les tâches d'administration, qui pourraient endommager le système en cas d'erreur, des droits spécifiques sont requis : les droits de **super utilisateur** (*root*).

Par défaut, sous Ubuntu, l'accès direct au compte système (*root*) est désactivé. La logique de ce système est d'utiliser `sudo` pour effectuer toutes les tâches administratives. Le premier utilisateur créé à l'installation d'Ubuntu a automatiquement les privilèges d'exécuter des commandes avec `sudo`.



Il est totalement déconseillé d'activer l'accès et d'utiliser directement le compte *root* sous Ubuntu.

Il est toutefois possible d'utiliser un « *terminal root* » :

```
$ sudo -i
```

// ou :

```
$ sudo -s
```

Pour pouvoir disposer des droits d'administration utilisables avec la commande `sudo`, il faut :

- soit appartenir au groupe `sudo` :

```
$ sudo adduser <identifiant> sudo
```

// ou :

```
$ sudo usermod -a -G sudo <identifiant>
```

- soit être déclaré dans le fichier `/etc/sudoers` :

```
$ sudo vim /etc/sudoers
```

```
# User privilege specification
```

```
root ALL=(ALL:ALL) ALL
```

```
<identifiant> ALL=(ALL:ALL) ALL
```

Documentation : https://doc.ubuntu-fr.org/utilisateurs/roschan/elevation_privileges_avances

Certaines commandes doivent donc être exécutées avec les droits de l'administrateur de la machine (*root*). Sous Ubuntu, il faut utiliser la commande `sudo`. Par exemple pour installer le programme `htop`, il faudra faire : `sudo apt install htop`

Question 18. En utilisant la commande `apt`, il faut réaliser une mise à jour des paquets disponibles puis installer les versions les plus récentes de tous les paquets présents sur le système :

```
sudo apt update
```

```
sudo apt upgrade
```

Question 19. Quelle est la configuration de votre interface réseau ? Utiliser la commande `ifconfig`.



Il vous faudra probablement installer le paquet `net-tools`.

Question 20. Afficher la table de partitions du disque sur lequel est installé Ubuntu ? Utiliser la commande `fdisk -l`.



Il existe de nombreuses commandes pour obtenir des informations techniques sur le système : `lshw`, `lsusb`, `lspci`, `lscpu`, `lsmem`, `lsblk`, `lsmod`, `lsof`, `netstat`, ...

Bonus : expérimentations

Question 21. Décrire la démarche expérimentale vous permettant de localiser la « Corbeille » associée à la session d'un utilisateur. Comment le « système » est capable de restaurer (au bon endroit) un fichier placé dans « Corbeille » ? Attention une explication technique détaillée est attendue !

On rappelle qu'il existe trois types de commande « interprétable » par le *shell* :

- les commandes externes :

```
$ type bash
bash est /bin/bash
```

- les commandes internes au *shell* :

```
$ type cd
cd est une primitive du shell
```

- les alias :

```
$ type ll
ll est un alias vers « ls -halF »
```

Question 22. Décrire la démarche expérimentale vous permettant d'identifier l'ordre de recherche d'une commande par le *shell*.

Question 23. Effectuer la démarche expérimentale décrite ci-dessous permettant de localiser le stockage des noms de fichier dans un système de fichier *ext*. Elle permet aussi de répondre à la question : quelle est la taille d'un répertoire ?

Créer un répertoire *tmp* dans votre répertoire personnel. Créer ensuite deux fichiers *essai1.txt* et *essai2.txt* dans le répertoire *tmp* précédemment créé :

```
$ mkdir ~/tmp
```

```
$ touch ~/tmp/essai1.txt ~/tmp/essai2.txt
```

Identifier le fichier de périphérique du répertoire *tmp* :

```
$ df ~/tmp
```

```
Sys. de fichiers Type Taille Utilisé Dispo Uti% Monté sur
/dev/sda1      ext4  440G   307G  111G  74% /
```

⇒ Le fichier de périphérique est */dev/sda1* (Le système de fichier est bien de type *ext*)

Vérifier si l'inode d'un fichier contient le nom du fichier :

```
$ ls -ai ~/tmp
```

```
25428124 . 25165826 .. 25428169 essai1.txt 25428170 essai2.txt
```

```
$ echo "stat <25428169>" | sudo debugfs /dev/sda1
```

```
Inode: 25428169 Type: regular Mode: 0644 Flags: 0x80000
```

```
Generation: 2199382711 Version: 0x00000000:00000001
```

```
User: 1026 Group: 65536 Project: 0 Size: 0
```

```
File ACL: 0
```

```
Links: 1 Blockcount: 0
```

```
Fragment: Address: 0 Number: 0 Size: 0
```

```
ctime: 0x5d96c304:23ebe544 -- Fri Oct 4 05:56:52 2019
```

```

atime: 0x5d96c304:23ebe544 -- Fri Oct 4 05:56:52 2019
mtime: 0x5d96c304:23ebe544 -- Fri Oct 4 05:56:52 2019
ctime: 0x5d96c304:23ebe544 -- Fri Oct 4 05:56:52 2019
Size of extra inode fields: 32
Inode checksum: 0x2498e6ee
EXTENTS:

```

➡ **Non!** L'inode d'un fichier ne contient pas le nom du fichier. Mais où est donc stockée cette information ?

Afficher le numéro d'inode du répertoire `tmp` :

```

$ ls -id ~/tmp
25428124 /home/tv/tmp

```

➡ Le numéro d'inode du répertoire `tmp` est **25428124**

Afficher le contenu de l'inode de `tmp` :

```

$ echo "stat <25428124>" | sudo debugfs /dev/sda1
Inode: 25428124 Type: directory Mode: 0755 Flags: 0x80000
Generation: 1290512767 Version: 0x00000000:00000003
User: 1026 Group: 65536 Project: 0 Size: 4096
File ACL: 0
Links: 2 Blockcount: 8
Fragment: Address: 0 Number: 0 Size: 0
ctime: 0x5d96c304:23ebe544 -- Fri Oct 4 05:56:52 2019
atime: 0x5d96c2e6:7745fcc8 -- Fri Oct 4 05:56:22 2019
mtime: 0x5d96c304:23ebe544 -- Fri Oct 4 05:56:52 2019
ctime: 0x5d96c2e6:7745fcc8 -- Fri Oct 4 05:56:22 2019
Size of extra inode fields: 32
Inode checksum: 0x3fee5619
EXTENTS:
(0):101721931

```

À partir de l'affichage du contenu de l'inode de `tmp`, déterminer le numéro de bloc utilisé par celui-ci :

➡ Le numéro de bloc est **101721931**

Afficher le contenu du bloc utilisé par `tmp` :

```

$ sudo dd if=/dev/sda1 bs=4096 skip=101721931 count=1 | hexdump -C
00000000 9c 00 84 01 0c 00 01 02 2e 00 00 00 02 00 80 01 |.....|
00000010 0c 00 02 02 2e 2e 00 00 c9 00 84 01 14 00 0a 01 |.....|
00000020 65 73 73 61 69 31 2e 74 78 74 4c 00 ca 00 84 01 |essai1.txtL....|
00000030 c8 0f 0a 01 65 73 73 61 69 32 2e 74 78 74 00 00 |...essai2.txt..|
00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
0000ff00 00 00 00 00 00 00 00 00 0c 00 00 de 8d 6e 1a 37 |.....n.7|
00001000

```

➡ Les noms de fichier sont donc conservés dans les « répertoires » où ils sont stockés!!!

Afficher la taille du répertoire `tmp` :

```

$ ls -ld ~/tmp
drwxr-xr-x 2 tv tv 4096 oct. 4 05:56 /home/tv/tmp

```

```
$ du -sh ~/tmp
4,0K /home/tv/tmp
$ du -sh ~/tmp/*
0 /home/tv/tmp/essai1.txt
0 /home/tv/tmp/essai2.txt
```

➔ La taille du répertoire `tmp` est de **4096 octets**. C'est la taille d'un bloc.



Il y a ici un conflit de « point de vue » ! Les informations que vous obtenez ici sont celles du système. Pour le système, la taille d'un répertoire correspond aux nombres de blocs qu'il utilise pour y stocker ses « entrées » de répertoire. Les « entrées » d'un répertoire sont des fichiers et des sous-répertoires. Pour l'utilisateur, la taille d'un répertoire est la somme des tailles des fichiers et des sous-répertoires qu'il contient. Cette information n'est pas connue du système. Elle sera alors forcément calculée (cf. la commande `du`).

Question 24. Décoder « *à la main* » les entrées du répertoire `tmp`.



Le format ext4 est décrit ici : https://ext4.wiki.kernel.org/index.php/Ext4_Disk_Layout. Les explications sur la structure d'« entrée » de répertoire sont ici : https://ext4.wiki.kernel.org/index.php/Ext4_Disk_Layout#Directory_Entries. La déclaration de cette structure en langage C se trouve dans ce fichier : https://elixir.bootlin.com/linux/v2.6.25/source/include/linux/ext4_fs.h

```
/*
 * Structure of a directory entry
 */
#define EXT4_NAME_LEN 255

struct ext4_dir_entry {
    __le32 inode;      /* Inode number */
    __le16 rec_len;   /* Directory entry length */
    __le16 name_len; /* Name length */
    char name[EXT4_NAME_LEN]; /* File name */
};

/*
 * The new version of the directory entry. Since EXT4 structures are
 * stored in intel byte order, and the name_len field could never be
 * bigger than 255 chars, it's safe to reclaim the extra byte for the
 * file_type field.
 */
struct ext4_dir_entry_2 {
    __le32 inode;      /* Inode number */
    __le16 rec_len;   /* Directory entry length */
    __u8 name_len;   /* Name length */
    __u8 file_type;
    char name[EXT4_NAME_LEN]; /* File name */
};

/*
 * Ext4 directory file types. Only the low 3 bits are used. The
 * other bits are reserved for now.
 */
#define EXT4_FT_UNKNOWN 0
```

```
#define EXT4_FT_REG_FILE 1
#define EXT4_FT_DIR 2
#define EXT4_FT_CHRDEV 3
#define EXT4_FT_BLKDEV 4
#define EXT4_FT_FIFO 5
#define EXT4_FT_SOCKET 6
#define EXT4_FT_SYMLINK 7

#define EXT4_FT_MAX 8
```



Depuis la version 4 d'ext, les données sont stockées dans l'ordre des octets Intel, c'est-à-dire en *little-endian*. Il faut absolument lire les explications ici : <https://fr.wikipedia.org/wiki/Boutisme>.



Pour assurer une optimisation, la taille des « entrées » de répertoire sera alignée sur une architecture 32 bits (soit un multiple de 4 octets). Il y aura donc un bourrage avec des valeurs 0x00 pour assurer cet alignement.



Une structure factice `ext4_dir_entry` est placée à la fin de chaque bloc pour contenir une somme de contrôle (*checksum*). Cette entrée de répertoire a une taille de 12 octets. Les champs `inode` et `name_len` sont mis à zéro pour permettre une compatibilité avec les anciennes version d'ext. La somme de contrôle est stockée à l'emplacement où se trouve normalement le nom.

Exemple de décodage de la première « entrée » de répertoire soit les 12 octets du bloc :

```
9c 00 84 01 0c 00 01 02 2e 00 00 00
```

Champ	Taille	Valeur	Description
<i>Inode number</i>	4 octets	9c 00 84 01	0184009c = 25428124
<i>Directory entry length</i>	2 octets	0c 00	12 octets
<i>Name length</i>	1 octet	01	1 caractère ou 1 octet
<i>Type</i>	1 octet	02	EXT4_FT_DIR = un répertoire
<i>File name</i>	255 octets max	2e	.



Ici, il y a 3 octets (0x00) qui ont été ajoutés pour assurer un alignement 32 bits.

Question 25. Est-ce que l'extension est une métadonnée d'un fichier ? C'est quoi alors ?

Question 26. Supprimer le fichier `essai2.txt`. Est-ce que l'entrée de répertoire a été modifiée ? Est-ce que l'inode de `essai2.txt` a été supprimée ? Proposez une explication sur le mécanisme de suppression d'un fichier. Que représente le champ `dtim` d'un inode ?

Question 27. Réfléchir au mécanisme de copie, déplacement, renommage, suppression du point de vue du système. Peut-on envisager de récupérer des données même après une suppression ou une perte de données ?



TestDisk est un puissant logiciel gratuit de récupération de données : <https://www.cgsecurity.org/wiki/TestDisk>