

Aide mémoire sur les connaissances informatiques minimales

Johan Ferguth¹

13 septembre 2017

1. Merci à Serge Fleury et Jean-Michel Daube pour les conseils et la relecture attentive

Résumé

Ce document se veut une aide mémoire technique pour les étudiants de Master 1 Plurital¹. Il est avant tout destiné à un public ayant peu de connaissances en informatique. Il fera un bon aide mémoire pour tout étudiant qui aura par la suite besoin d'un rappel sur une notion. Au vu du public concerné, on comprendra les choix de rédaction visant à la simplification et à la vulgarisation de concepts quitte à parfois faire des raccourcis réducteurs.

1. <http://www.tal.univ-paris3.fr/plurital/>

Sommaire

1	Qu'est ce qu'un ordinateur ?	7
1	Informatique	7
2	Hardware	9
2.1	Processeur	9
2.2	Random Access Memory	11
2.3	Hard Drive	12
3	Software	12
3.1	BIOS	13
3.2	Operating system	13
3.3	Panorama des systèmes d'exploitation	14
3.4	Unix et Linux	16
3.5	Choix du système	17
2	Utilisation d'un système Linux ou Unix	20
1	Installation	20
1.1	Fichier Image des OS, ou Image disque des VMs	21
1.2	Configuration du Bios	21
2	Virtual Machine	22
2.1	Java et VirtualBox	22
2.2	Configuration de VM	25
2.3	Installation du système d'exploitation virtuel	29
3	Dual Boot	38
4	Alternative 1 : Cygwin	39
5	Alternative 2 : Bash sous windows 10	47
3	Utilisation du système en ligne de commande	52
1	Rudiment du Bash	52
1.1	Qu'est ce que le Bash ?	52
1.2	Qu'est ce qu'une commande ?	54
1.3	Déplacement dans l'arborescence	59
1.4	Création, suppression, déplacement/copie de fichiers/- répertoires	62
1.5	Flux, redirection, lecture de fichier	62

1.6	Calculs	62
1.7	Premier programme, variables	62
1.8	Instruction de contrôle 1 : structure conditionnelle . .	62
1.9	Instruction de contrôle 2 : boucle	62
2	Programmation Bash avancée	62
2.1	Filtres, Joker en Bash	62
2.2	Fonctions	62
2.3	Insertion de commandes	62
2.4	Variables complexes : Tableaux et Tables de hash . . .	62
2.5	Sous-processus et expressions parenthésées	62
2.6	Processus en arrière plan et redirection sur plusieurs flux	62
3	Introduction à l'administration système	62
3.1	Droits d'accès, groupe/utilisateur et super-utilisateur .	62
3.2	Gestion des packages	62
3.3	Connexion à distance avec ssh	62

Figures

1.1	Architecture de John Von Neumann	8
1.2	Exemple de processeur	9
1.3	Barrettes de RAM 512 Mio	11
1.4	Disque Dur	12
1.5	Rapport entre matériel, OS, application et utilisateur	13
1.6	Utilisation des OS : à gauche sur PC en 2014, à droite sur Smartphones en 2011	15
1.7	Histoire des Unix	18
1.8	Quelques distributions sur Linux	19
2.1	Exemple d'écran principal de bios	22
2.2	Site d'oracle pour le téléchargement de virtualbox	23
2.3	Téléchargement de virtualbox V5.1.26	23
2.4	Procédure d'installation de virtual box	24
2.5	Configurer le bios pour permettre le plein usage de la virtualisation	24
2.6	Démarrage de VirtualBox	25
2.7	Importation de VM	25
2.8	Nommer la machine virtuelle et définir le type de système d'exploitation	26
2.9	Attribution de la RAM allouée à la VM	26
2.10	Création d'un disque dur virtuel de type VDI	27
2.11	Disque dur type statique ou dynamique	27
2.12	Emplacement et taille du disque virtuel	28
2.13	Machines Virtuelles disponibles dans le menu de virtualbox	28
2.14	Création d'un disque dur virtuel de type VDI	29
2.15	Démarrage de l'installateur d'Ubuntu 16.04 et information de contrôle de VirtualBox	30
2.16	Choix de la langue de l'installateur	30
2.17	Options de l'installateur	31
2.18	Choix de la langue de l'OS	31
2.19	Choix des options d'installation	32
2.20	Paramétrage du support d'installation	32

2.21	Confirmation et écriture sur le disque	33
2.22	Zone géographique	34
2.23	Clavier français	35
2.24	Configuration du compte utilisateur	36
2.25	Finalisation de l'installation	36
2.26	Fin de l'installation	37
2.27	Utilisation de la VM	37
2.28	Sélection du boot device dans l'écran du Bios pour démarrer sur une clé USB	39
2.29	Téléchargement de l'installateur de Cygwin	40
2.30	Lancement de l'installateur sur une architecture 64 bits	40
2.31	Sélection des options d'installation : sources internet	41
2.32	Sélection des options d'installation : multi utilisateurs, dossier dans disque C	42
2.33	Sélection des options d'installation : Configuration internet . .	42
2.34	Sélection des options d'installation : Sélection du miroir de téléchargement	43
2.35	Sélection des options d'installation : Sélection des packages . .	44
2.36	Installation : autorisation pour les dépendances non-satisfaites	45
2.37	Installation en cours	46
2.38	Fin de l'installation	46
2.39	Cygwin en action	47
2.40	Mode développeur	48
2.41	Activation des fonctionnalités	48
2.42	Sous-système Windows pour Linux (bêta)	49
2.43	Recherche de fichiers avant de redémarrer.	49
2.44	Initialisation du bash	50
2.45	Paramétrages supplémentaires	50
2.46	Bash an action sous Windows 10	51
3.1	Lancement du terminal	53
3.2	Lancement du terminal	54
3.3	Appel du navigateur firefox par le terminal bash	55
3.4	Arborescence sous Unix	59

Tableaux

2.1	Proposition de sélection de package	44
-----	---	----

Introduction

Dans le domaine du *TAL* ou Traitement Automatique des Langues vous serez amenés à traiter des données langagières et, souvent, à interagir avec elles par le biais d'un traitement informatique (programmes, scripts, logiciels spécialisés, etc). Une des ambitions de ce master est de vous former à cette interaction. Pour cela, la compréhension de certains concepts, matériels ou logiciels, sont autant de clés qu'il vous faudra acquérir.

Le présent document fait un rapide survol de notions essentielles, de façon à faciliter l'apprentissage pratique qui suivra. Pour cela comprendre un ordinateur qui représente « votre outil de travail » nous semble un aspect non négligeable. C'est pourquoi nous aborderons tour à tour les éléments matériels qui le composent dans la partie 1. Le matériel n'allant pas sans son pendant logiciel, nous aborderons cette question en éclaircissant les points clés tels que le système d'exploitation et ses relations avec le matériel.

Nous passerons ensuite à un apprentissage pratique sur les possibilités d'interaction avec un système d'exploitation en ligne de commandes. Nous verrons pour cela comment installer une machine virtuelle, un émulateur de système Unix (Cygwin), ou encore comment utiliser les possibilités offertes par Windows 10.

Enfin après cette séquence d'installation nous aborderons les bases d'un langage de programmation en ligne de commande, le *Bash*.

Chapitre 1

Qu'est ce qu'un ordinateur ?

Nous définissons dans ce chapitre quelques notions essentielles nécessaires pour mieux appréhender la seconde partie du document. On s'y reportera pour rappel le cas échéant. Nous commençons par un rapide historique de l'informatique qui permet de comprendre l'évolution du couple matériel/logiciel. Nous éclairons ce duo dans les sections qui suivent ¹.

1 (Très) Bref Historique de l'Informatique

L'apparition des ordinateurs (nommés ainsi en 1956 comme équivalant de « *data processing machine* ») répond au besoin initial de la résolution de calculs complexes trop long en traitement ou avec une marge d'erreur significative pour l'humain. Très vite les premiers calculateurs se voient dotés de la capacité de traiter des lettres en plus des chiffres. On dépasse alors le cadre du calcul pour un périmètre plus large de gestion de l'information.

1. La plupart des informations présentées ici sont des compilations de morceaux choisis de :

- Mooc(s) sur le langage C de Rémi Sharrock, maître de conférences à Télécom ParisTech,
 - [Sharrock, 2016a]
 - [Sharrock, 2016b]
- Mooc(s) sur le langage C et Java de François Barthélemy, maître de conférences en informatique au CNAM,
 - [Barthélemy, 2017a]
 - [Barthélemy, 2017b]
- Wikipédia :
 - [Wikipédia, b]
 - [Wikipédia, a]
- Un explication complète sur les microprocesseur de la marque Intel dans [lecrabeinfo.net, 2016]

L'évolution ultérieure de l'informatique est très liée à un certain nombre d'avancées théoriques, matérielles et corrélée à l'apparition de langages de programmation visant à faciliter l'interaction homme/machine. On peut par exemple citer² :

- les découvertes physiques sur les semi-conducteurs et la miniaturisation des transistors ;
- les découvertes mathématiques sur la calculabilité et les propriétés des algorithmes ;
- l'invention de la théorie de l'information, de la sémiotique et de la cybernétique ;
- les transformations techniques avec l'introduction de machines ou de composants informatiques (des machines à laver aux automobiles et aux avions, des banques à la santé, de l'imprimerie à la documentation en ligne) ;
- transformations sociales avec l'organisation des entreprises et des administrations autour de leur système d'information automatisé, avec la circulation de l'information dans les réseaux sous forme numérique.

Parmi ces évolutions, une avancée significative est atteinte avec la possibilité d'enregistrer des séquences d'instructions qui peuvent être transmises à l'ordinateur. Cette réalisation est conceptualisée dans l'architecture dite de Von Neumann³ (dès 1945) avec un modèle encore utilisé aujourd'hui dans la fabrication d'ordinateurs.

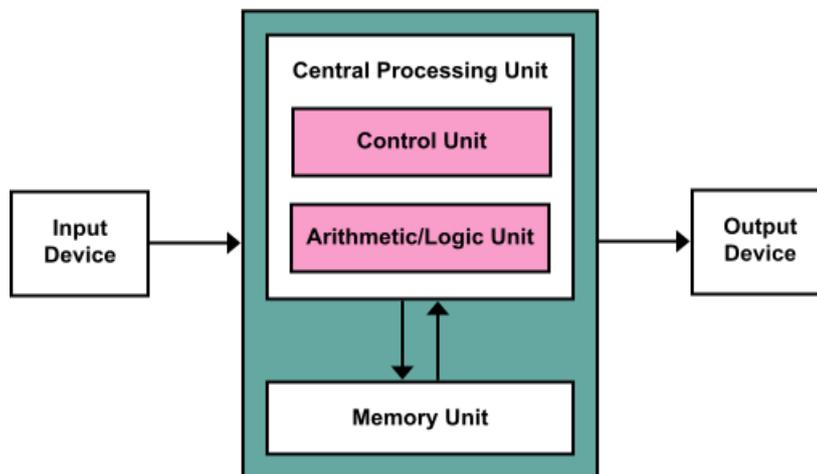


FIGURE 1.1 – Architecture de John Von Neumann

2. Toujours dans [Wikipédia, b].

3. [Wikipédia, a].

L'architecture de Von Neumann décompose l'ordinateur en quatre parties distinctes :

- l'unité arithmétique et logique ou unité de traitement, souvent incluse dans le processeur et qui a pour but d'effectuer les calculs ;
- l'unité de contrôle ou processeur, qui permet de séquencer les opérations ;
- la mémoire qui contient essentiellement des données se divise entre mémoire volatile pour les données de fonctionnement (RAM qui s'efface lorsque l'on éteint sa machine) et mémoire « permanente » (dont les données sont stockées sur un disque dur) ;
- les dispositifs d'entrée-sortie pour la communication entre l'ordinateur et l'extérieur.

Avec ces exemples on comprend bien que la conceptualisation d'un système informatique est très dépendante des innovations technologiques et du type de matériel utilisé. Ce matériel (hardware) conditionne les capacités des logiciels (software). S'il n'est pas nécessaire d'en connaître tous les détails, avoir une idée générale de son fonctionnement facilite l'utilisation des langages de programmation enseignés durant ce master.

2 Hardware

C'est souvent au moment de l'achat d'un ordinateur que l'on voit apparaître sur les étiquettes récapitulatives des produits un certain nombre d'éléments visant à nous permettre d'orienter notre choix. Au delà du prix, les comprendre permet d'évaluer la performance future de notre achat. Cette performance conditionne les limites de ce que l'on pourra faire ou pas avec ce nouveau matériel.

Nous verrons dans cette section les quelques points matériels permettant de mieux appréhender votre environnement de travail. Nous aborderons ainsi successivement les éléments que sont le processeur, la RAM et le disque dur.

2.1 Processeur



FIGURE 1.2 – Exemple de processeur

Le processeur ou unité centrale de traitement (UCT) (en anglais « *central processing unit* », CPU) représente un peu le moteur de l'ordinateur. Comme nous l'avons vu les processeurs sont initialement théorisés dans l'architecture de Von Neumann conçue pour répondre à des problèmes de reprogrammation⁴. L'apparition des transistors en 1948 permet, entre autre, de miniaturiser ce type de matériel⁵, on parle d'ailleurs aujourd'hui de microprocesseurs.

Le rôle principal du processeur est d'exécuter les instructions que lui envoie le système d'exploitation que nous verrons dans la section suivante (par exemple Windows). Cela signifie qu'il traite une à une les instructions qui lui sont soumises. Ainsi que l'on regarde une vidéo, écoute de la musique ou encore lance un logiciel, on fait appel au processeur.

Qu'est ce qu'un cœur ?

Initialement les séries d'instructions envoyées au processeur étaient traitées les unes à la suite des autres (moyennant une gestion des plus prioritaires aux moins prioritaires). Pour garantir l'accélération du traitement de ces instructions sont apparus les processeurs multi-cœurs. Là où un processeur traitait une instruction à la fois, un *dual-core* (deux cœurs) en traite deux, un *hexa-core* (six cœurs) en traite six ... etc. L'intérêt est donc de permettre à un utilisateur d'exécuter plusieurs tâches en même temps sans subir de ralentissements. Spécifiquement on verra surtout un gain significatifs dans le cadre de l'utilisation de plusieurs logiciels simultanément ou encore de logiciels conçus pour être répartis sur plusieurs cœurs car trop gourmands en ressources (comme les jeux vidéos, les logiciels de montage audio/vidéo ou encore de retouche photos).

Le nombre de cœurs d'un processeur est donc un premier indice de la puissance du processeur mais ... pas le seul !

Autres indicateurs de performances : Fréquence, Mémoire cache, Hyper-Treading et Turbo Boost

Fréquence

La fréquence permet d'évaluer la rapidité du processeur, mesurée en gigahertz (Ghz), plus elle est élevée plus l'exécution d'une instruction est rapide. À nombre de cœurs égal, l'indicateur de vitesse est donc la fréquence.

Mémoire cache

4. Avec le recâblage entier de gigantesques calculateurs à chaque nouveau programme.

5. Puisqu'à ce moment là les calculateurs/ordinateurs occupent parfois une pièce entière.

La mémoire cache est l'équivalent de la mémoire vive (RAM) mais directement intégrée au processeur et donc plus rapide. Quand une opération va être réalisée plusieurs fois, plutôt que de reproduire cette opération, le résultat sera stocké dans cette zone de mémoire pour être réutilisé. À la différence de la mémoire vive, la mémoire cache est de taille relativement réduite.

Hyper-Threading (Intel)⁶

Au lancement d'un logiciel, plusieurs tâches sont effectuées, on appelle ces tâches des processus. Chacun des processus envoyés au processeur comprend une ou plusieurs instructions sur ce qu'il doit faire. Ces séquences sont appelées des « *threads* ». Comme nous l'avons vu rapidement un cœur ne peut exécuter qu'un thread à la fois (deux cœurs = deux threads, ... etc). Avec la technologie Hyper-Thread, les cœurs physiques peuvent traiter deux threads à la fois. On parle alors de cœurs logiques et, bien qu'inférieurs aux cœurs physiques, on évalue leur gain de performance au minimum à 30%.

Turbo Boost (Intel)⁷

La technologie turbo boost offre des outils de surveillance incorporés au processeur qui lui permettent d'augmenter sa fréquence quand les demandes de l'utilisateur nécessitent un traitement plus rapide⁸.

2.2 Random Access Memory



FIGURE 1.3 – Barrettes de RAM 512 Mio

6. L'hyper-thread est l'implémentation sur des processeurs Intel du Simultaneous Multi Threading (SMT), d'autres implémentations peuvent exister sur des processeurs de marques différentes.

7. Comme l'hyper-thread le turbo boost est un exemple d'amélioration de performance issu de la technologie Intel.

8. Nous n'aborderons pas la question ici mais on pourrait aussi citer la finesse de gravure du microprocesseur comme indice de performance (aujourd'hui autour de 14nm).

Nous avons vu un certain nombre d'éléments relatifs au processeur qui a pour but d'effectuer les calculs et de séquencer les opérations (cf Von Neumann). Nous avons aussi vu que le processeur peut avoir besoin de stocker des résultats pour les réutiliser ultérieurement dans sa mémoire cache. Néanmoins ce cache est relativement petit, c'est pourquoi la RAM ou mémoire vive entre en jeu : elle permet de stocker plus de données. De manière générale la mémoire cache plus rapide mais plus petite réduit les interactions avec la RAM de la même manière que cette dernière réduit les interactions avec le disque dur que nous allons voir maintenant.

2.3 Hard Drive

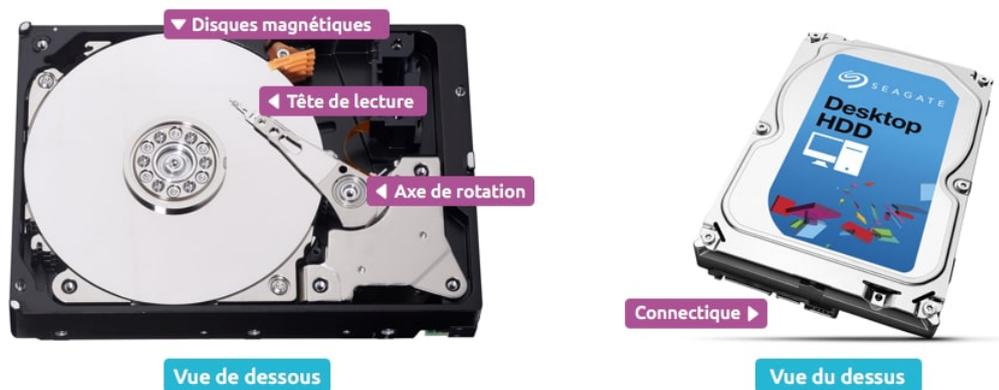


FIGURE 1.4 – Disque Dur

Quand l'on souhaite sauvegarder des données sur le long terme ou simplement que ces données soient gardées en mémoire après l'extinction de l'ordinateur on lance une opération d'écriture sur le disque dur. Ces opérations sont plus longues que l'accès au cache, ou même à la RAM puisqu'on écrit sur un disque au sens propre⁹.

3 Software

Nous abordons maintenant la question du logiciel en explorant la notion de système d'exploitation et de BIOS avant de présenter les systèmes Unix.

⁹. Par modification du champ magnétique du disque.

3.1 BIOS

Le BIOS ou « *Basic Input Output System* » est un firmware (micrologiciel) installé sur la carte mère et qui permet à l'ordinateur d'effectuer des opérations de base (mise sous tension, d'identification du matériel d'entrée sortie, lecture d'un secteur de disque dur, etc). Il s'effectue de manière transparente pour l'utilisateur avant de laisser la place au chargement du système d'exploitation. Il peut néanmoins être utile d'accéder à la configuration du BIOS en cas de panne ou si l'on souhaite évoluer avec différents systèmes d'exploitation sur une même machine comme nous le verrons dans la partie installation.

3.2 Operating system

Un système d'exploitation (ou OS pour « *Operating system* ») est un programme servant d'intermédiaire entre le matériel physique que nous avons présenté auparavant et les applications¹⁰. Ainsi en tant qu'utilisateur nous utilisons les services rendus par les applications de la même façon que ces dernières utilisent le système d'exploitation. Enfin lui-même exploite le matériel physique.

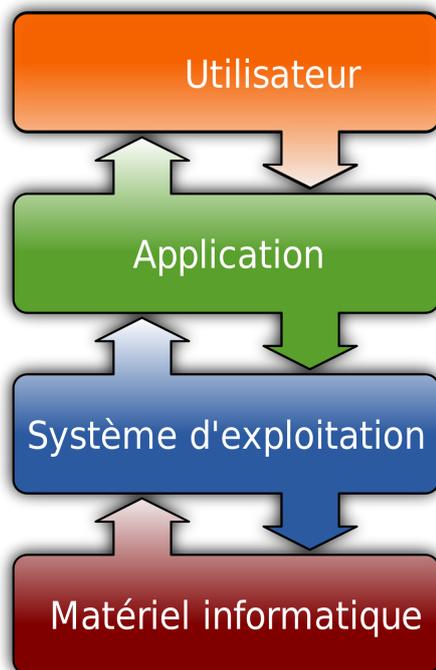


FIGURE 1.5 – Rapport entre matériel, OS, application et utilisateur

10. Ou autrement dit les programmes.

L'OS s'occupe par exemple de la gestion :

- des « file system » (FS) : Création et gestion d'un système de fichiers¹¹ qui garantit la possibilité de lire et d'écrire des données sur le disque dur et les organise avec une cohérence que l'on appelle l'arborescence de fichier.¹²
- de la mémoire vive : La RAM peut ainsi être partagée entre plusieurs applications en même temps.
- des applications : La gestion des processus en cours d'utilisation (Lancer une application, la stopper, etc)
- des entrées/sorties : envoi/réception de données pour l'écran, le clavier, la carte son/vidéo, la carte réseaux (pour internet), etc.

La création des systèmes d'exploitation vise ainsi à simplifier les interactions entre l'homme et la machine. Mais quels sont les systèmes d'exploitation que l'on peut utiliser ?

3.3 Panorama des systèmes d'exploitation

L'observation d'un rapide sondage¹³ sur l'utilisation des systèmes d'exploitation nous apporte les chiffres suivants :

11. On parle de formatage.

12. Les fichiers et dossiers forment un arbre logique qui correspond à leur emplacement physique sur le disque dur. On comprend ici pourquoi un disque organisé selon un standard imposé par un OS ne garantit pas forcément sa reconnaissance si l'on change d'OS. Par exemple un disque dur externe formaté par MacOS n'est pas forcément reconnu sur un ordinateur Windows sans les bons outils logiciels.

13. Source : [\[Wikipédia, c\]](#)

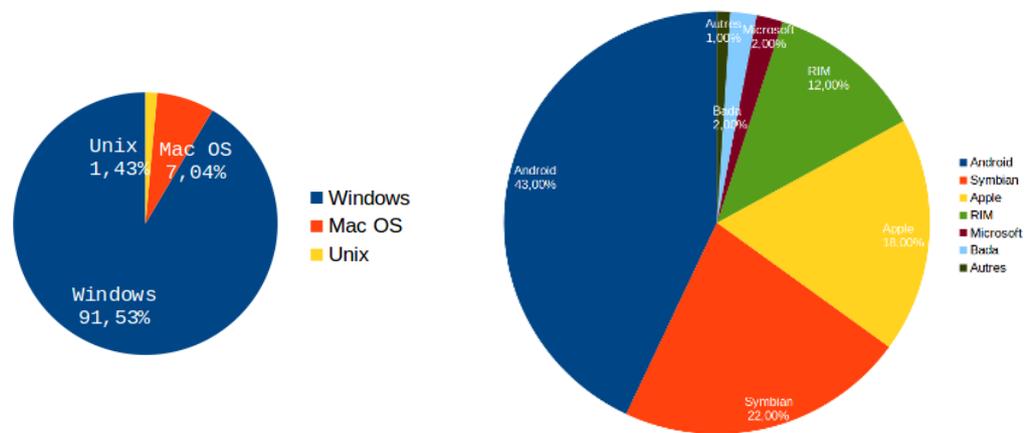


FIGURE 1.6 – Utilisation des OS : à gauche sur PC en 2014, à droite sur Smartphones en 2011

Ce sondage, bien qu’ancien, nous livre un panorama des OS utilisés ces dernières années. Si Windows ou encore MacOS sont des systèmes d’exploitation bien connus du grand public, il faut savoir que bien que moins connus, les systèmes de type Unix représentent une part de marché plus conséquente dès que l’on parle de serveurs web¹⁴ ou encore de cloud services¹⁵.

Bien que Windows reste leader sur le marché PC (90.7%), plus récemment¹⁶ on a observé une évolution de plus de 3% d’utilisation de Linux sur les PC juste derrière MacOS (5.94%)

Mais qu’est ce qu’Unix et quel est son rapport avec Linux ?

Unix

Projet MAC¹⁷

Unix est un peu le « père » des systèmes d’exploitation modernes. Il doit son origine au système MULTICS. En effet, dans les années 60/70 après les transistors¹⁸, on voit apparaître les circuits intégrés¹⁹ et les disques magnétiques (début des disques durs). Ces innovations technologiques donnent

14. Notamment le serveur Apache d’après [developpez.com, 2015] en 2015.

15. Linux à plus de 90% en 2012 selon [journal dunet.com, 2012].

16. Voir [developpez.com, 2017] de septembre 2017.

17. Toute la section historique d’Unix est une synthèse de l’excellent l’historique de [Sharrock, 2016a].

18. Dans les années 1950, qui permettent la réduction de matériel.

19. La mise en relation de plusieurs transistors contribue encore davantage à la miniaturisation.

naissance à des familles d'ordinateurs compatibles entre eux. L'avancée matérielle permet de conceptualiser le projet MAC (Man And Computer) qui débute à l'institut de technologie du MIT et est financé par l'organisme de défense des États Unis, la DARPA. Dans ce projet, on imagine un système pouvant partager ses ressources matérielles entre applications et utilisateurs simultanément.

Le système baptisé MULTICS voit finalement le jour en 1969 grâce à deux entreprises : les laboratoires Bell pour la partie logiciels et General Electric pour la réalisation matérielle. Malheureusement la réalisation concrète est loin d'égaliser l'idée initiale du projet et les laboratoires Bell se retirent du projet.

Unics

Quelques ingénieurs des Bell labs²⁰, forts de leur expérience de travail sur MULTICS, décident sans soutien et pour leur propre compte de réaliser un système minimal qui serait moins lourd et moins complexe. Le système Unics²¹ est né. En 1969 il est mono-utilisateur et devient mutli-utilisateurs en 70. Pour l'occasion il est rebaptisé Unix. Le système est complètement réécrit en langage C en 1972 après la création de ce dernier. Ceci explique que les deux soient encore très liés puisque, encore aujourd'hui, la plupart des fonctions des systèmes d'exploitation Unix sont encore écrites en C. En 1975, Unix est offert aux universités et à des centres de recherche à des fins éducatives. Il devient alors très populaire, et rassemble une très large communauté dans les années 80.

Unix aujourd'hui

Aujourd'hui le système Unix est omniprésent : on citera par exemple MacOS (Mac), IOS (pour Iphone) ou encore Android (la plupart des smartphones), qui sont des dérivés d'Unix²². Enfin, il en est aussi de même pour Linux qui, comme nous l'avons vu est présent sur la grande majorité des serveurs et objets connectés.

3.4 Unix et Linux

Alors qu'Unix est en plein boom dans le monde de la recherche dans les années 80, en 83 est lancé le projet GNU²³ visant à fournir une suite de logiciels compatibles avec Unix. Les logiciels très nombreux de ce projet (éditeur de texte, compilateur, système de fenêtrage, etc) sont entièrement

20. Avec entre autre Ken Thompson et Dennis Ritchie les inventeurs du langage C.

21. Jeu de mots sur Multics.

22. On notera aussi que beaucoup de protocoles issus d'Unix sont repris chez Windows, notamment le protocole TCP/IP qui est à la base d'Internet.

23. Initié par Richard Stallman du MIT. GNU signifie « *GNU's not Unix* », les jeux de mots récursifs étant à l'honneur dans le monde informatique.

gratuits et libres de modifications²⁴ mais utilisés avec le système d'exploitation Unix qui est lui propriétaire. Cette possession le rend soumis aux choix d'exploitation de son détenteur, les Bell labs. De plus, les licences propriétaires limitent aussi l'usage du système d'exploitation puisqu'elles (entre autre) ne permettent pas de modification.

Un étudiant finlandais, Linus Torvalds, entre alors en scène : il se propose de développer un noyau de système d'exploitation qui serait aussi gratuit. Une autre communauté se crée et le projet GNU associe ses logiciels à cet OS : Linux est né. Les premières distributions (Le noyau/système d'exploitation + certains des logiciels du projet GNU) Linux apparaissent dans les années 90. C'est le plus grand projet collaboratif humain jamais réalisé. A la fin des années 90 tous les grands constructeurs de matériels informatiques annoncent la compatibilité avec Linux. A partir de 2010 la plupart des matériels (hors PC) tournent sous Linux ou Unix. Quelques exemples actuels :

- . Serveurs Web (+ de 70%)
- . Smartphones (+ de 90%)
- . Ordinateurs de calculs
- . Serveurs de l'industrie du film ou effets spéciaux (environ 99%)
- . Consoles de jeux (ex : PlayStation)
- . Box Internet
- . Routeurs wifi
- . Objets connectés

3.5 Choix du système

Reste à choisir le système à utiliser parmi la miriade d'OS et de distributions disponibles. En effet nous avons présenté l'histoire d'Unix comme un ensemble uniforme qui aboutissait à la création de Linux, cependant, les choix technologiques et querelles idéologiques ont donné naissance à un grand nombre de système Unix²⁵.

24. Voir la licence GPL qui donne une liberté totale sur un programme.

25. On parle même d'Unix Wars [blank] à compléter



FIGURE 1.8 – Quelques distributions sur Linux

A retenir :

Concernant l'installation de ce type de système avant d'aborder la partie pratique on peut noter que :

- Les élèves possédant un PC de type Mac ont déjà un environnement Unix. Il n'est donc pas utile de passer par la phase d'installation. Néanmoins ces élèves devront résoudre les problèmes de différence entre un système Linux et le leur (gestion des packages et installation de certains programmes par exemple, commandes différentes (wget vs curl), etc)
- Les élèves possédant un PC fonctionnant sous Windows auront à effectuer une installation telle que décrite dans le chapitre 2 avec un choix à faire entre 4 options :
 - L'installation d'une machine virtuelle : Virtualbox + machine virtuelle avec une distribution Ubuntu ou Lubuntu
 - L'installation d'un émulateur de système Unix dans leur Windows : Cygwin
 - L'utilisation d'un environnement Unix natif sous Windows : bash + Windows 10
 - Installation en dual boot des deux OS (voir abandon de Windows) : implique le lancement soit de Windows, soit de l'OS alternatif choisi

Dans tous les cas, des problèmes viendront se poser pendant la période de prise en main du système (transfert de fichiers entre les OS, installation de logiciels avec procédures différentes selon les OS, arborescences différentes notamment pour Cygwin). Chaque solution d'installation possède ses avantages et ses inconvénients, il appartiendra aux élèves d'en faire mention et d'essayer de trouver des solutions sachant que tous les problèmes ne pourront être abordés en cours.

Chapitre 2

Utilisation d'un système Linux ou Unix

Nous allons maintenant aborder la partie pratique en commençant avec quelques explications permettant d'installer un système Linux ou Unix.

Remarque : Ce chapitre concerne les étudiants ayant un pc Windows souhaitant installer un OS Unix. Les étudiants utilisant MacOS¹ ou Linux peuvent passer au chapitre suivant.

1 Installation

Aujourd'hui les solutions pour utiliser différents environnements sur un même poste de travail ne manquent pas.

Historiquement on a d'abord vu apparaître le « *Dual Boot* ». Cette solution permettait d'avoir deux systèmes d'exploitation sur un même disque dur. Malheureusement, dans ce cas, on ne peut utiliser qu'un seul de ces OS à la fois. Cela oblige à éteindre et rallumer son ordinateur pour passer de l'un à l'autre. De plus la gestion du système de fichier n'étant pas uniforme entre OS (certains FS n'étant pas exploités²) il fallait aussi prévoir un espace d'échange avec un FS commun pour les deux systèmes d'exploitation. Cela revenait donc à diviser son disque dur (partitionner) en trois morceaux distincts (un pour chaque OS et un pour l'espace d'échange).

Pour pallier cela sont apparues les machines virtuelles (ou VM). Il s'agit ici, grâce à un programme, d'installer un OS « dans » un autre OS. On parle

1. Il peut être utile pour les Mac d'installer un gestionnaire de paquets comme Homebrew. Voir https://brew.sh/index_fr.html

2. On pourra ici citer en exemple et de manière non exhaustive les FS :

- ext2/3/4, btrfs pour Linux
- NTFS, NFS pour Windows/Linux/MacOS X
- HFS+ pour MacOS X/Linux, APFS pour MacOS X

alors de machine hôte (par exemple windows) qui contient une machine invitée (par exemple Linux). L'inconvénient ici réside dans le partage des ressources systèmes : si l'ordinateur hôte est trop faible, son invité ne pourra pas fonctionner pleinement. De plus le partage de fichiers entre hôte et invité demande quelques ajustements.

Les alternatives pour pouvoir malgré tout utiliser un système Unix réside dans l'installation de programmes d'émulation (Cygwin) ou en débloquent des fonctionnalités sur Windows (Bash uniquement pour Windows 10)

1.1 Fichier Image des OS, ou Image disque des VMs

On peut considérer deux options :

- Dans le premier cas on installe un système complet en partant de zero. Ce cas est valable pour le dual boot ou dans le cadre d'utilisation des machines virtuelles. Avant tout il faut récupérer l'image du système d'exploitation que l'on souhaite installer (le suffixe du fichier est en .iso). Il suffit par exemple de taper dans un moteur de recherche : ubuntu+iso+download+16.04³ pour trouver le lien suivant : <http://releases.ubuntu.com/16.04/> où l'on choisira la version 32 ou 64 bits en fonction de l'architecture de son ordinateur.
- Dans le second cas on utilise une machine virtuelle déjà constituée par l'équipe enseignante. C'est cette option qui sera privilégiée et un lien de téléchargement sera fourni en classe pour télécharger la VM.

1.2 Configuration du Bios

Dans tous les cas, il faut comprendre que le système d'exploitation lancé après le BIOS l'a été avec un certain nombre d'options configurées par ce même BIOS. Certaines de ces options (par sécurité ou pour d'autres raisons) limitent les possibilités de votre machine ou conditionnent son comportement. Il vous faudra sans doute changer certaines de ces options pour permettre certaines installations. Chaque constructeur garantit un accès au Bios de manière différente. Afin de faciliter votre travail, vous pouvez rechercher comment accéder à celui de votre machine en tapant dans un navigateur internet : « *[la marque de votre matériel]+Bios* »

La manière d'y accéder dépend de la marque du matériel mais revient souvent à presser une touche au moment de l'allumage de la machine (la plupart du temps une des touches comprises entre F1 et F12, Echap ou encore Del).

3. Pour les ordinateurs n'ayant pas plus d'un cœur et avec une RAM inférieure ou égale à 2Go il vaudrait mieux considérer la version allégée d'ubuntu : lubuntu. On tape alors lubuntu+iso+download+16.04 dans un moteur de recherche pour obtenir le lien <http://cdimage.ubuntu.com/lubuntu/releases/16.04/release/> et y télécharger ensuite l'image.

Ce faisant, vous devriez avoir accès à un écran comme celui ci ⁴ :

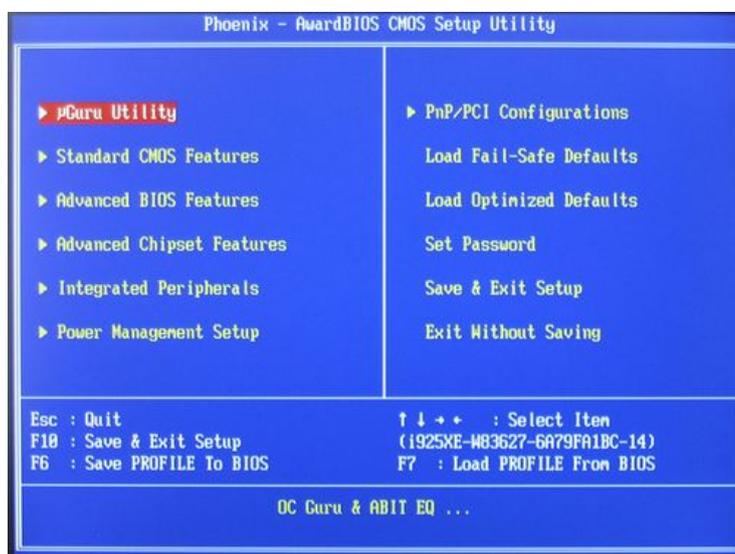


FIGURE 2.1 – Exemple d'écran principal de bios

Nous exposerons dans les sous-sections suivantes les cadres d'utilisation du Bios selon le type d'installation choisi.

2 Virtual Machine

2.1 Java et VirtualBox

virtualbox est un programme permettant l'installation d'une VM invitée dans votre machine hôte. Il s'appuie sur un autre programme, *java*. Nous allons donc installer successivement java puis virtualbox. Java se décompose en plusieurs logiciel selon l'utilité que l'on peut en avoir. Le plus complet est celui de développement ou JDK (pour Java Development Kit). En tapant « *jdk+download* » on obtient par exemple le lien suivant : <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> où on peut télécharger la version 8 du JDK et l'installer.

Une fois l'installation de java effectuée on peut faire de même avec virtual box. On commence par trouver le site officiel d'oracle qui est le fournisseur de virtual box :

4. Chaque Bios étant différent, celui-ci est fourni à titre d'exemple.

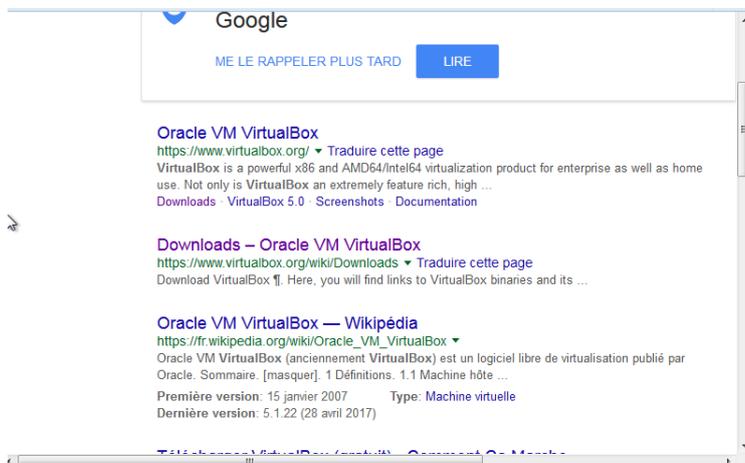


FIGURE 2.2 – Site d’oracle pour le téléchargement de virtualbox

On télécharge ensuite le programme d’installation :

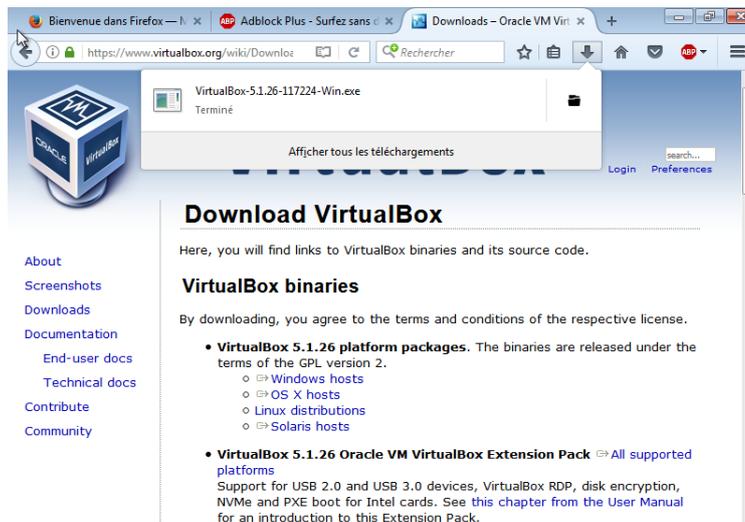


FIGURE 2.3 – Téléchargement de virtualbox V5.1.26

Il ne reste qu’à installer le programme :

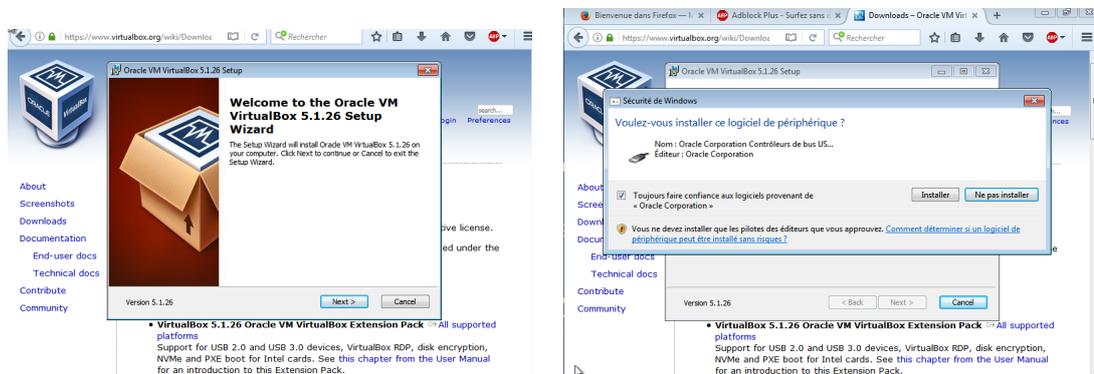


FIGURE 2.4 – Procédure d’installation de virtual box

A ce stade, avant de démarrer l’installation d’une VM, il peut être utile de vérifier que le Bios ne possède pas une configuration restrictive concernant les VMs. On peut permettre la virtualisation comme dans l’exemple suivant :

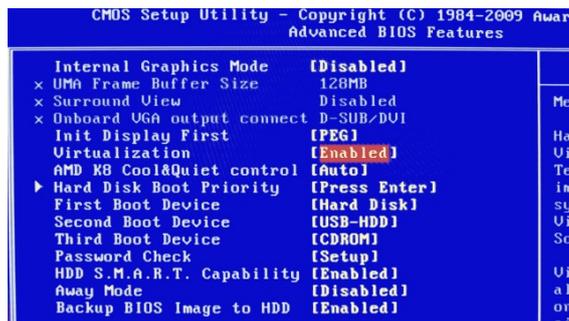


FIGURE 2.5 – Configurer le bios pour permettre le plein usage de la virtualisation

On démarre ensuite VirtualBox⁵ :

5. La plupart des captures d’écran sont réalisées sous système Linux, outre les différences visuelles, quelques différences de syntaxe peuvent apparaître : par exemple un chemin sous Linux s’écrit « /<dossierLinux>/<fichierLinux> » contre « C:\<dossierWindows>\<fichierWindows> » pour Windows.

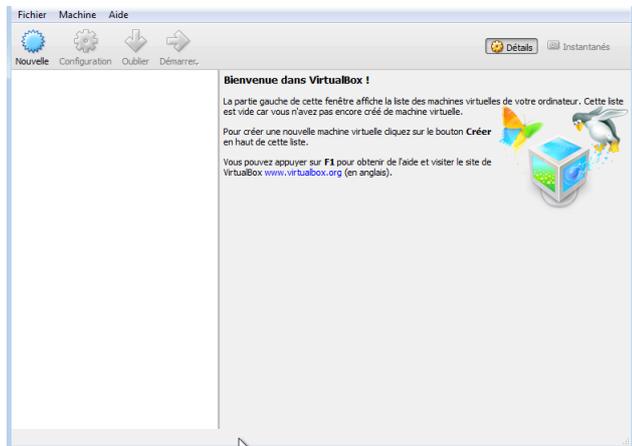


FIGURE 2.6 – Démarrage de VirtualBox

2.2 Configuration de VM

Si on possède déjà une VM on peut l'importer (fichier/importer) en renseignant le chemin :



FIGURE 2.7 – Importation de VM

Dans ce cas l'importation est réalisée en quelques instants et on peut utiliser le bouton démarrer immédiatement pour accéder à la VM nouvellement importée.

Dans le cas contraire on peut cliquer sur « Nouvelle » pour installer une nouvelle machine virtuelle. Pour cela il faudra en créer une à partir d'un fichier.iso (cf supra) comme nous le verrons plus bas. Pour commencer on doit donner un nom au nouveau système :

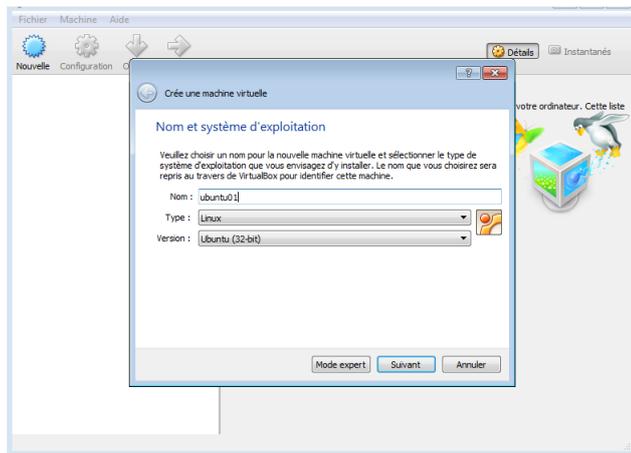


FIGURE 2.8 – Nommer la machine virtuelle et définir le type de système d'exploitation

On doit ensuite définir la RAM que l'on accorde à la VM en prenant garde de ne pas atteindre le maximum pour éviter de ralentir le système hôte ou de le saturer :

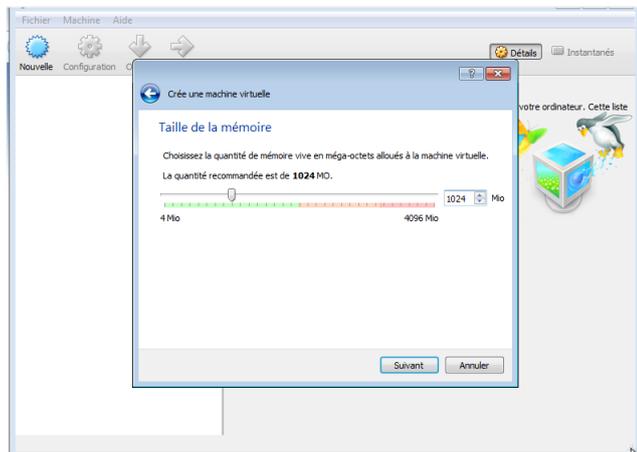


FIGURE 2.9 – Attribution de la RAM allouée à la VM

On peut sélectionner 1024 ou 2048 ce qui correspondra respectivement à 1Go ou 2Go de mémoire vive allouée à votre VM.
Attention de bien prendre en compte les spécificités de votre système. Ce

paramètre peut néanmoins être changé selon les besoins après la création de la VM.

Il faut ensuite créer un disque dur virtuel pour la VM :



FIGURE 2.10 – Création d'un disque dur virtuel de type VDI

Ce disque dur pourra être de deux types soit fixe soit dynamique :



FIGURE 2.11 – Disque dur type statique ou dynamique

Chaque élève devra faire un choix selon l'espace dont il dispose en gardant à l'esprit les points suivants :

- La taille fixe réserve exactement la taille que l'on renseigne sur le disque dur réel (ce qui crée, par exemple, un fichier d'exactly 10 Go si c'est la taille que l'on souhaite). Ce choix prend tout l'espace alloué, est un peu plus long à la création mais plus rapide à l'utilisation.
- La taille dynamique s'ajuste en fonction des besoins mais ne dépasse pas la taille renseignée qui est son maximum autorisé (le fichier peut

donc être de taille relativement modeste à la création). Ce mode est plus rapide à la création, prend juste l'espace nécessaire à la VM pour fonctionner mais est un peu plus lent à l'utilisation.

Après ce réglage on passe à l'allocation de l'espace disque proprement dit et à l'emplacement où sera stocké le disque virtuel :

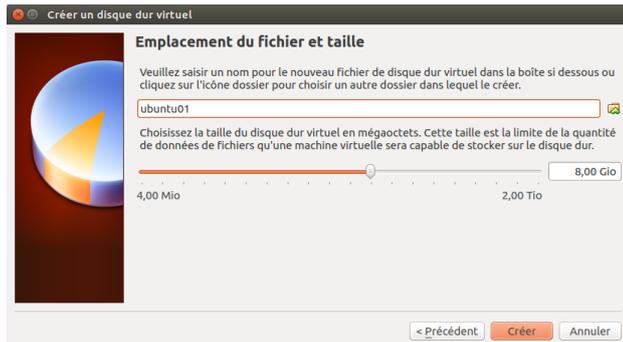


FIGURE 2.12 – Emplacement et taille du disque virtuel

Plus le disque dur sera grand, plus vous pourrez stocker d'informations dans votre VM. 50Go semble un bon compromis, mais si votre système manque de place, 10 Go peuvent être largement suffisants pour faire fonctionner votre système ⁶.

La machine virtuelle est enfin créée et on peut la sélectionner en cliquant sur démarrer :

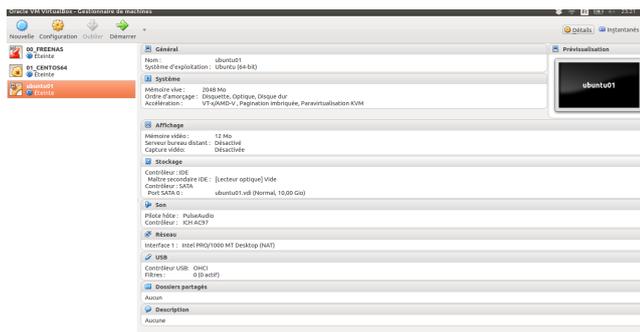


FIGURE 2.13 – Machines Virtuelles disponibles dans le menu de virtualbox

6. À condition de limiter l'installation de programmes complémentaires

A ce stade, commence la partie installation de l'OS. En effet, la machine virtuelle est opérationnelle mais elle est vide. Pour pouvoir utiliser ce nouvel ordinateur il faut lui installer un système d'exploitation. Pour cela on sélectionne l'image (.iso) téléchargée de l'OS que l'on a choisi et on lance la procédure d'installation :

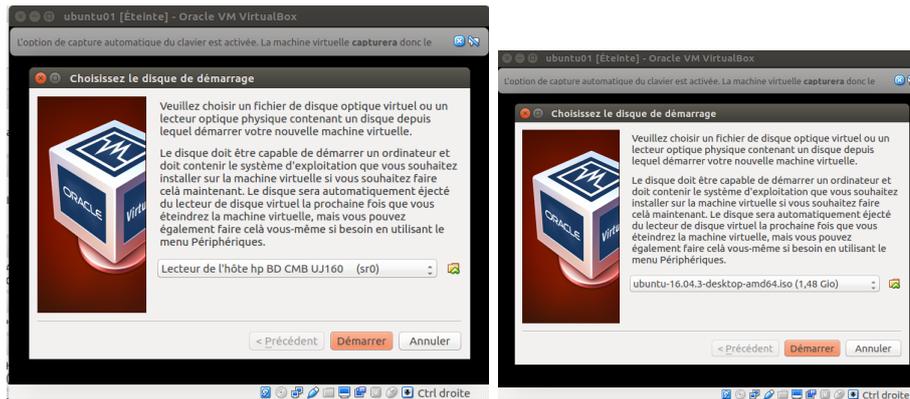


FIGURE 2.14 – Création d'un disque dur virtuel de type VDI

2.3 Installation du système d'exploitation virtuel

On considérera maintenant le paramétrage uniquement du point de vue de l'OS que l'on est en train d'installer en faisant abstraction de virtualbox ou du matériel réel (machine hôte). Un seul élément est à prendre en compte, lorsque l'on sélectionne la VM avec la souris, les interactions avec les entrées sorties (clavier, souris, ...) sont redirigées vers la VM. Par exemple la souris est captive de la VM. Pour libérer les interactions vers le système hôte il faut utiliser la touche « ctrl » droite du clavier. Cette touche est changeable et est toujours indiquée en bas de la VM :

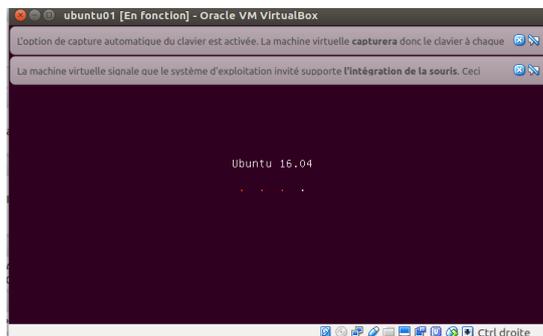


FIGURE 2.15 – Démarrage de l’installateur d’Ubuntu 16.04 et information de contrôle de VirtualBox

Remarque : L’ordre et l’agencement des séquences suivantes peut varier en raison de la disparité des OS et logiciels de virtualisation testés pendant la réalisation de ce guide.

Un fois lancé on peut choisir la langue de l’installateur avec le clavier qu’on valide avec « entrée » :

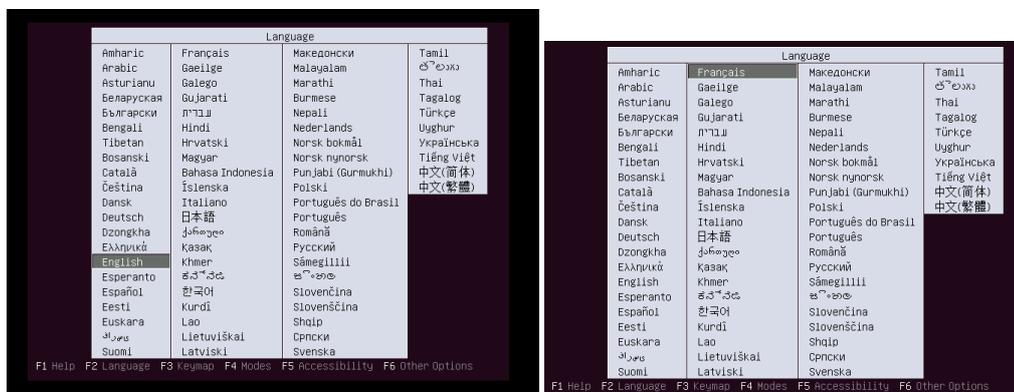


FIGURE 2.16 – Choix de la langue de l’installateur

L’installateur permet plusieurs options⁷, sans surprise on choisit « installer » :

7. Notamment des tests utiles en cas de difficultés pendant l’installation.



FIGURE 2.17 – Options de l'installateur

On peut aussi choisir d'installer l'OS dans la langue de son choix :

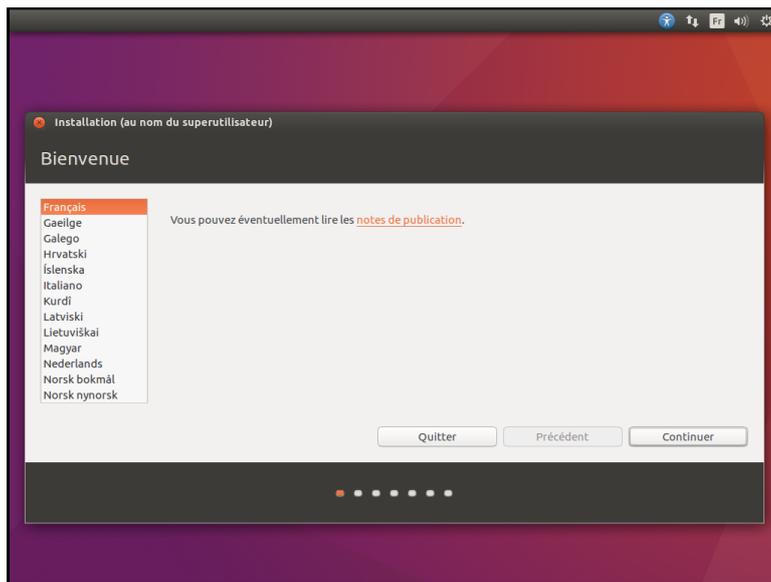


FIGURE 2.18 – Choix de la langue de l'OS

Il vaut mieux choisir de télécharger⁸ le maximum d'éléments durant l'installation :

8. Avec connexion internet bien sûr !

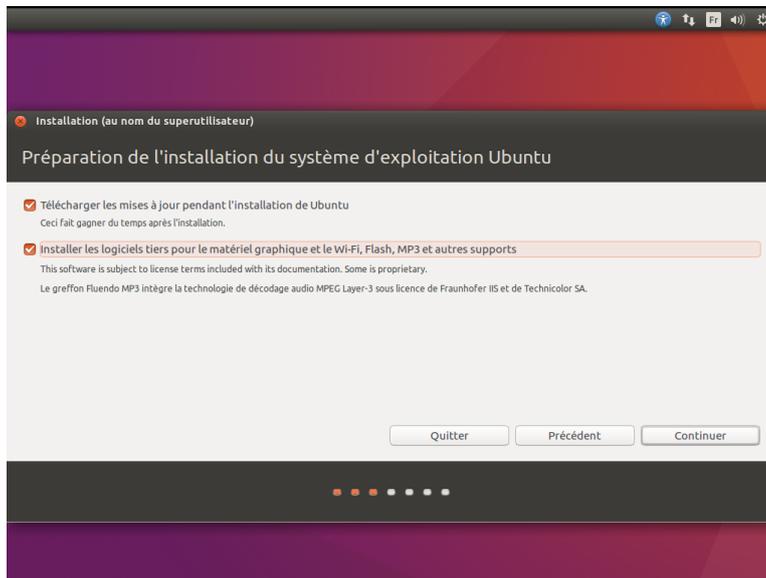


FIGURE 2.19 – Choix des options d'installation

On peut ensuite choisir les conditions d'écriture sur le disque dur. Dans le cas de la VM il s'agit du disque dur virtuel qui ne nécessite pas de configuration particulière :

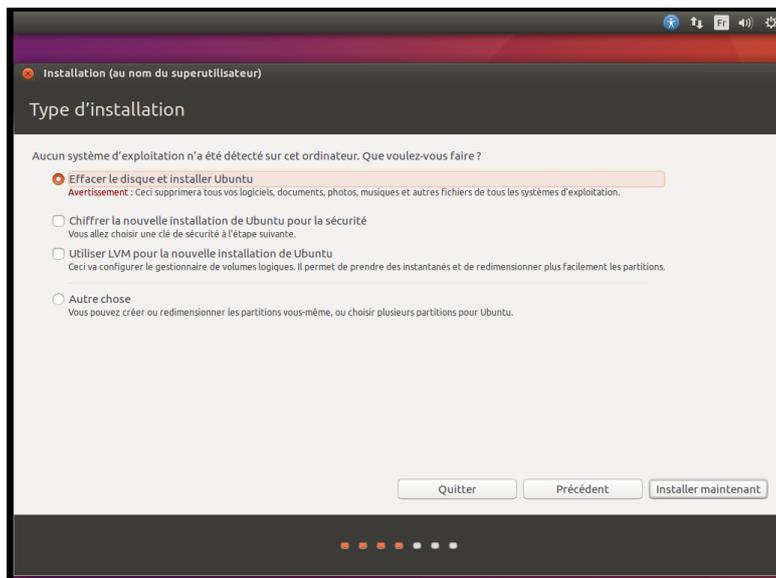


FIGURE 2.20 – Paramétrage du support d'installation

On ne détaillera pas les autres cas d'installation ici mais à titre indicatif :

- Chiffrer la nouvelle installation : correspond au cryptage du disque, plus longue cette méthode est surtout utile sur des machines réelles afin protéger des données sensibles.
- LVM : méthode de gestion de l'espace disque permettant de gérer l'espace disque par des volumes logiques. Utile pour sécuriser et optimiser le stockage sur un disque dur. De cette façon on facilite le déplacement des données et la configuration de l'espace. Ici aussi, on l'utilisera surtout sur des machines réelles.
- Autre chose : permet la configuration manuelle et la personnalisation de l'installation. Avec l'aide d'un assistant graphique on peut facilement partitionner, formater et réorganiser l'espace du disque dur selon ses besoins. Cette option permet une personnalisation assez fine mais nécessite la compréhension de plusieurs concepts que nous n'aborderons pas ici.

Une fois l'option sélectionnée, l'écriture sur le disque va avoir lieu, cette écriture demande confirmation :

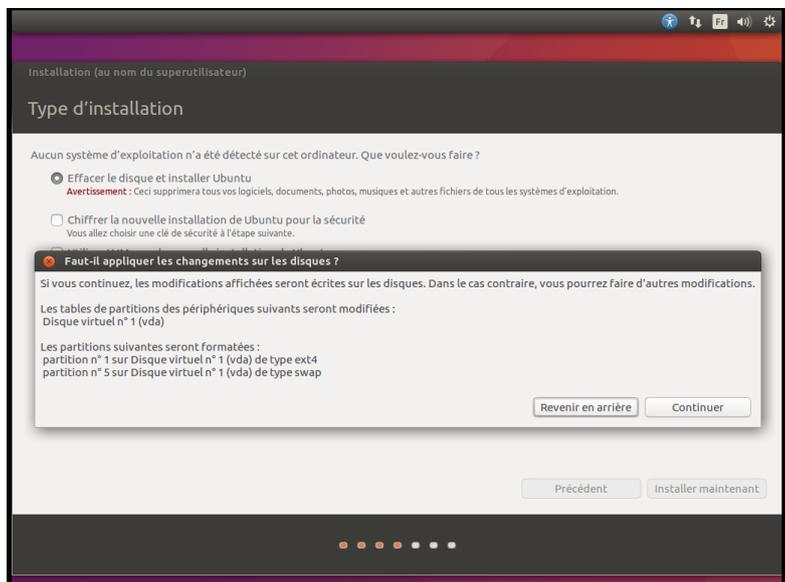


FIGURE 2.21 – Confirmation et écriture sur le disque

On remarquera d'ailleurs que cette installation crée deux partitions, une en « ext4 » un format commun chez linux et très efficace (pas besoin de

défragmentation comme sur Windows par exemple). L'autre partition dite « Swap » est un moyen intéressant d'augmenter la RAM⁹ en utilisant de l'espace disque réservé. Dans ce choix de configuration la Swap est calculée automatiquement.

On sélectionne ensuite la zone d'installation (gestion de la timezone, etc) :

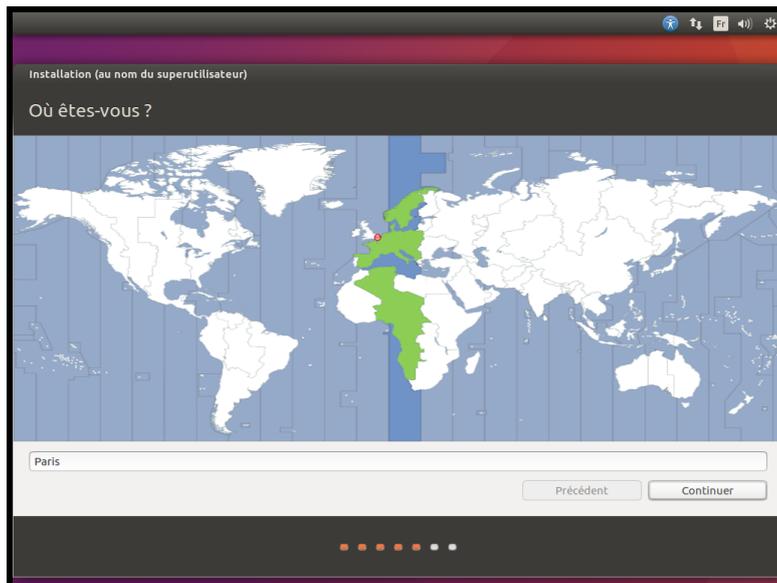


FIGURE 2.22 – Zone géographique

On configure ensuite la disposition du clavier :

9. Si par le passé on réservait une taille de Swap équivalente à deux fois la RAM cette pratique est aujourd'hui dépassée et inefficace. Sur une machine réelle, une swap d'une taille comprise entre 5 et 10 Go est globalement suffisant.

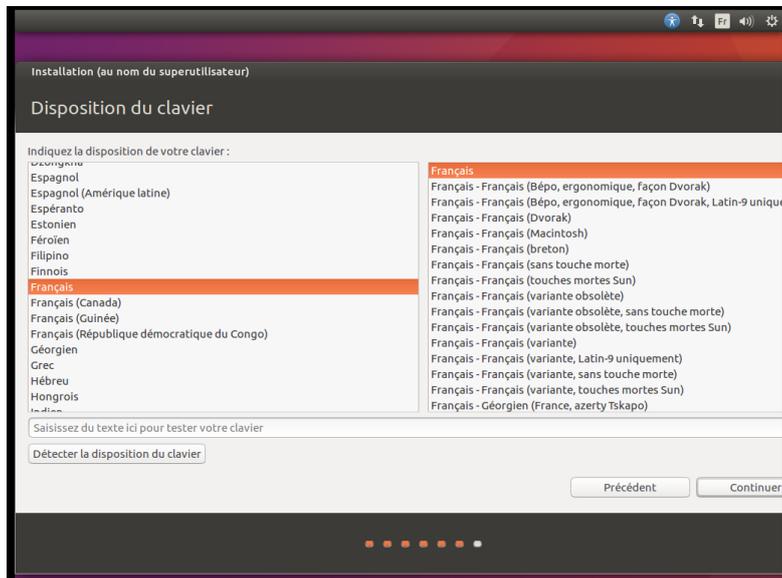


FIGURE 2.23 – Clavier français

On prendra garde ici de ne pas sélectionner la détection automatique qui n'est pas toujours bien gérée par l'utilisation de la virtualisation. On fera aussi attention de ne pas sélectionner la variante obsolète du clavier français.

A ce stade on peut renseigner les différentes informations du compte utilisateur :

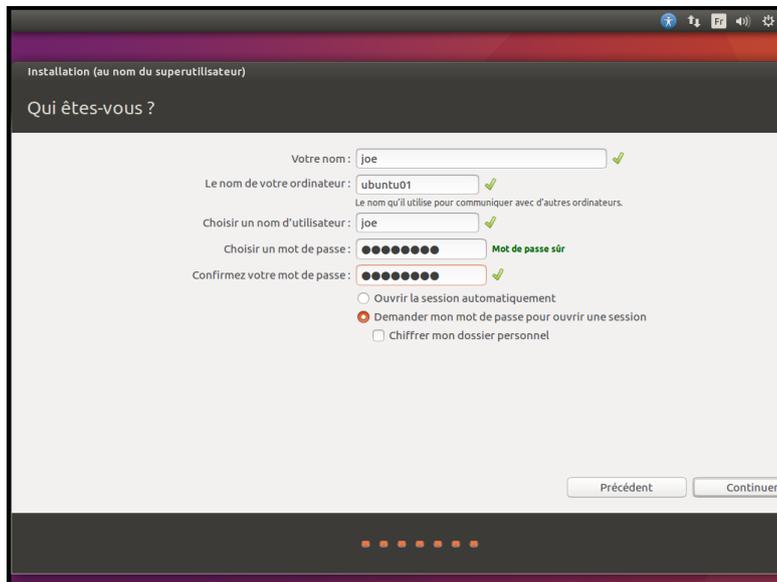


FIGURE 2.24 – Configuration du compte utilisateur

A ce stade la procédure est presque terminée, il reste à attendre que le système procède aux téléchargements et installations divers ¹⁰ :

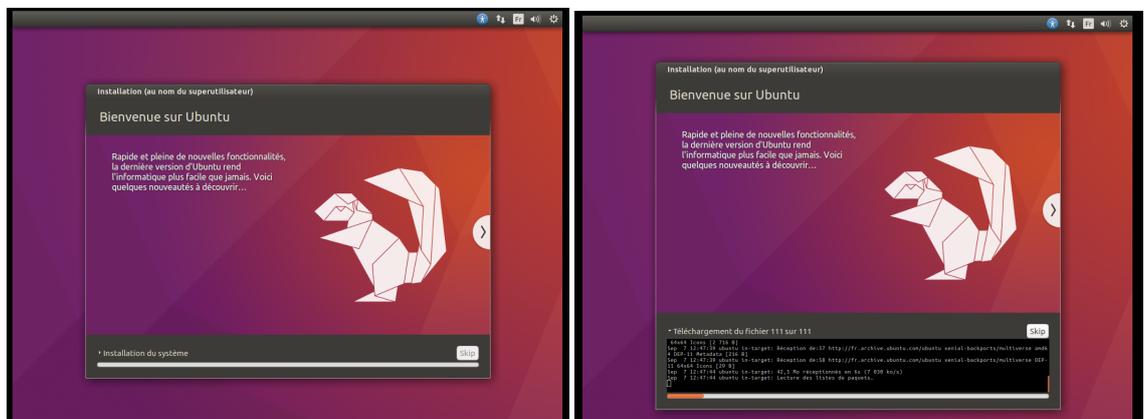


FIGURE 2.25 – Finalisation de l'installation

À l'issue des différentes opérations le système doit redémarrer ¹¹ :

10. Cliquer sur le menu déroulant permet de voir l'évolution l'installation.
11. Si le système redémarre en repassant par l'installateur, éteindre la machine et désactiver la lecture du fichier.iso dans les options de virtualbox

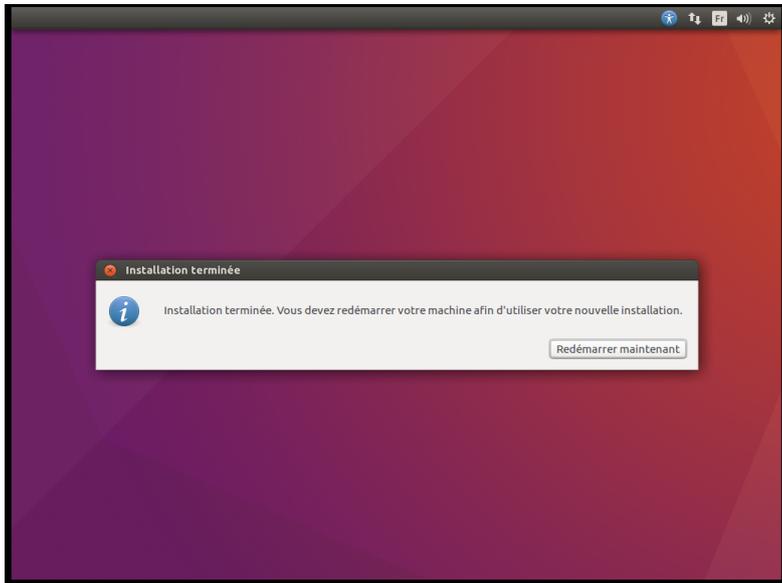


FIGURE 2.26 – Fin de l'installation

On accède enfin au menu de notre VM et on peut l'utiliser selon les besoins :

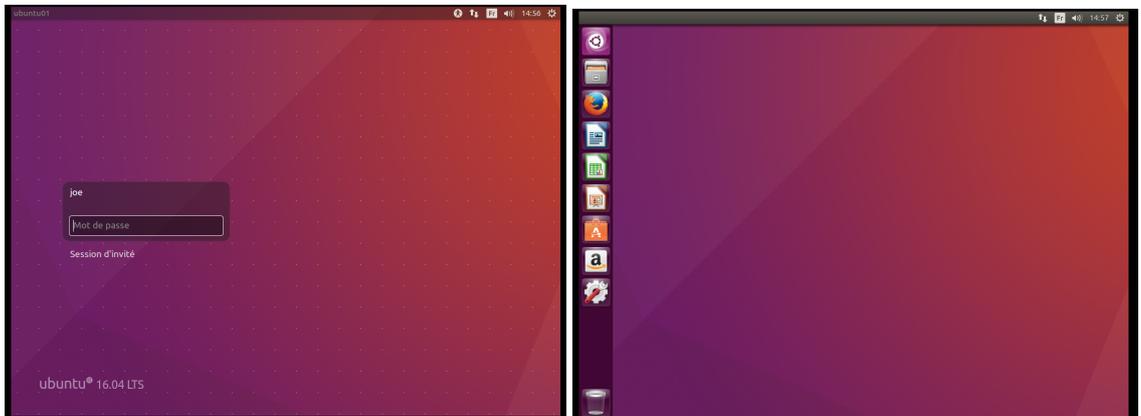


FIGURE 2.27 – Utilisation de la VM

Félicitations, vous venez d'installer votre première VM.

3 Dual Boot

Remarque : Compte tenu du nombre de concepts à connaître et des inconvénients que nous avons mentionnés plus haut, nous ne conseillons pas cette méthode. Nous ne fournissons ici que les grandes lignes à titre indicatif pour ceux qui souhaiteraient la tester. Cette méthode est singulièrement la même si vous souhaitez remplacer votre système d'exploitation par un autre.

Il est plus prudent de créer un espace sur son disque dur (partitionner) pour recevoir la nouvelle installation car nous avons vu que les FS varient selon les OS ¹².

Après la récupération de l'image (type fichier.iso voir plus haut) qui contient tous les éléments nécessaires à l'installation de l'OS on utilise en général un programme pour mettre en place cette image sur une clé USB. Le programme unetbootin ¹³, par exemple, permet la détection de la clé USB au lancement du Bios et garantit qu'elle soit utilisable (on dit « bootable ») pour lancer l'installateur. De cette façon le système pourra démarrer sur un nouvel OS directement via cette clé.

L'installation en dual boot se heurte en général à cette seule difficulté : arriver à détecter et démarrer sur un autre périphérique que le disque dur sur lequel est installé l'OS courant de l'ordinateur. Puisque l'installation d'un nouveau système d'exploitation sur un disque dur présuppose que celui-ci n'est pas en cours d'utilisation, en démarrant sur un autre OS on garantit que l'OS courant n'effectue aucune modification sur ce disque. On évite ainsi les conflits entre le nouvel OS qui intervient directement sur le matériel et son homologue déjà installé. C'est en général le même type d'opération que l'on utilise quand on démarre sur un cd-rom pour réparer une panne système.

On peut configurer la reconnaissance et le fait de démarrer sur un périphérique matériel autre que le disque dur (la clé USB) directement dans le BIOS :

12. Dans un navigateur tapez `partitionner+disque dur+windows+[numéro de version de votre OS]`.

13. Pour télécharger Unetbootin : <https://unetbootin.github.io/>.

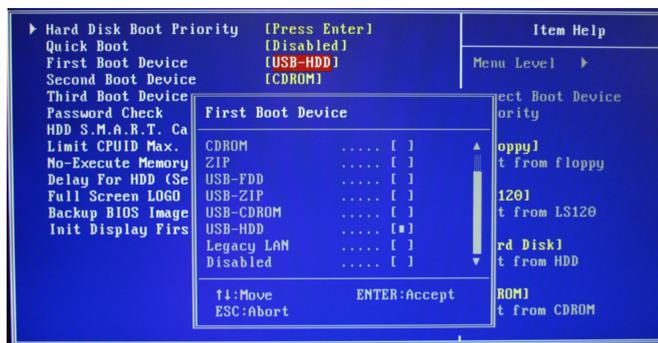


FIGURE 2.28 – Sélection du boot device dans l'écran du Bios pour démarrer sur une clé USB

Une fois l'installateur lancé, on en revient à la procédure d'installation déjà abordée plus haut.

Attention : Dans le cas présent il convient de faire attention et de ne pas effacer la partition utilisée par Windows. Le partitionnement ayant été effectué auparavant dans ce dernier, vous devrez être sûr de l'endroit où vous lancerez l'installation¹⁴. De plus, pour répondre aux problématiques d'échange de données il convient de créer deux partitions supplémentaires (trois en tout) sur le disque : une pour Windows déjà existante, une pour l'installation d'Ubuntu et une comme espace d'échange qui devra être partitionné en ntfs.

4 Alternative 1 : Cygwin

Cygwin est un émulateur qui regroupe un certain nombre de logiciels libres permettant d'émuler un système Unix.

Son point fort est donc qu'il suffit d'installer un programme pour permettre l'utilisation de fonctionnalités Unix dans un environnement Windows. En revanche, on perd certaines capacités d'un système natif puisque Cygwin n'est pas un OS à part entière.

Comparativement aux autres installations, celle de Cygwin peut sembler aisée (pas de configuration BIOS par exemple). Il convient néanmoins de faire attention à certains points, notamment concernant la partie installation des packages que nous verrons plus bas.

Pour toutes les opérations suivantes une connexion internet avec un débit suffisant est nécessaire.

On commence par télécharger l'installateur de Cygwin :

¹⁴. La désignation des partitions n'étant pas la même sur Linux, on ne retrouve pas la mention disque C. Un bon indice concernant l'identité d'une partition réside dans sa taille, celle-ci étant la même quel que soit l'OS considéré.



FIGURE 2.29 – Téléchargement de l'installateur de Cygwin

Pour cela on sélectionnera dans la rubrique « install » le « setup » qui correspond à l'architecture de l'ordinateur 32 ou 64 bits. Sur un ordinateur 64 bits par exemple, on lance ensuite le programme :

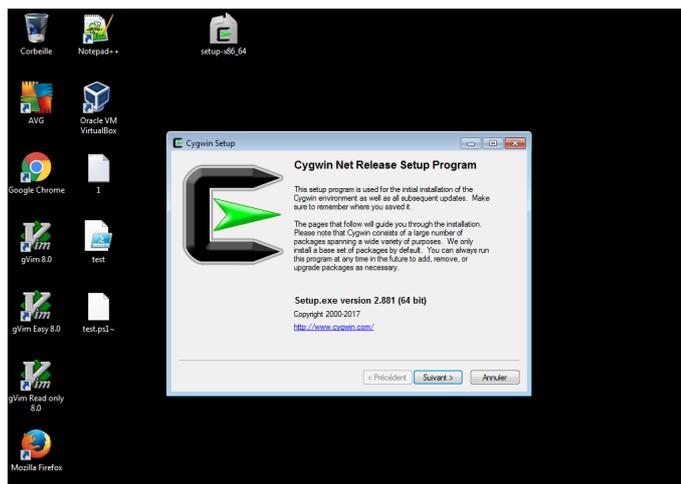


FIGURE 2.30 – Lancement de l'installateur sur une architecture 64 bits

Après plusieurs informations habituelles lors de l'installation de programmes à renseigner, on fera attention de sélectionner l'installation via internet qui permet de récupérer les fichiers sources sur un serveur distant :

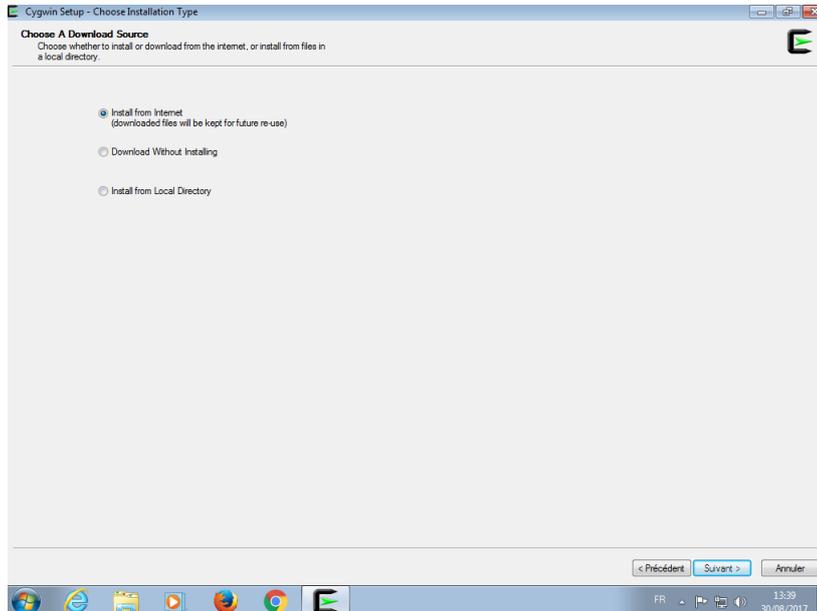


FIGURE 2.31 – Sélection des options d'installation : sources internet

On installe aussi une version de Cygwin pour tous les utilisateurs directement dans le disque C :

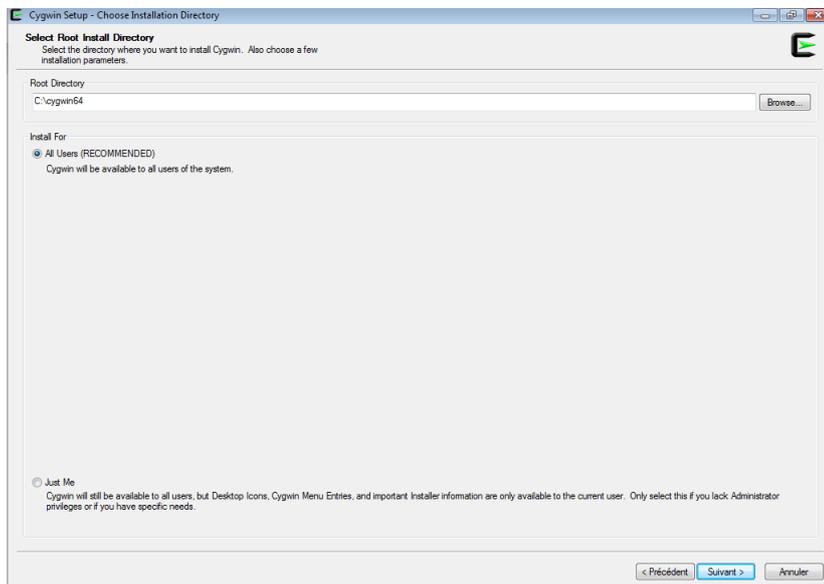


FIGURE 2.32 – Sélection des options d’installation : multi utilisateurs, dossier dans disque C

La configuration internet est normalement standard :

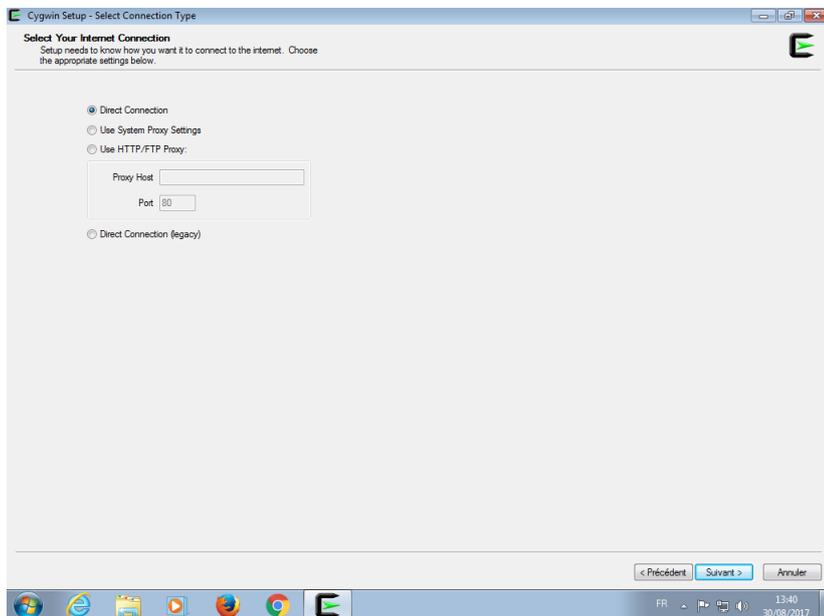


FIGURE 2.33 – Sélection des options d’installation : Configuration internet

On peut ensuite choisir le serveur (et donc la zone géographique la plus proche) où seront téléchargées les sources. Cela permet de réduire le temps de traitement ou, si beaucoup de personnes téléchargent en même temps, d'éviter de saturer un seul serveur :

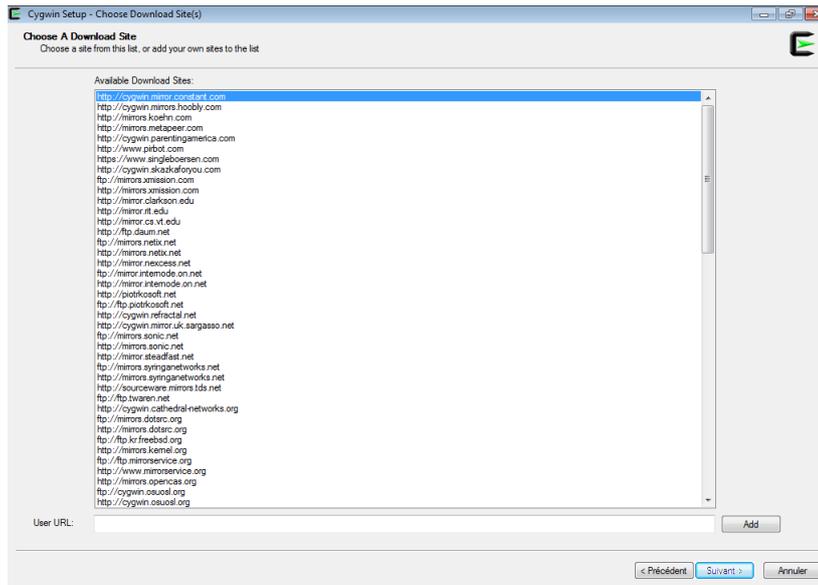


FIGURE 2.34 – Sélection des options d'installation : Sélection du miroir de téléchargement

Enfin, c'est sans doute l'étape la plus importante, on sélectionne le type d'installation et les fonctionnalités qui seront intégrées dans celle-ci. Pour cela il faut choisir des « packages » :

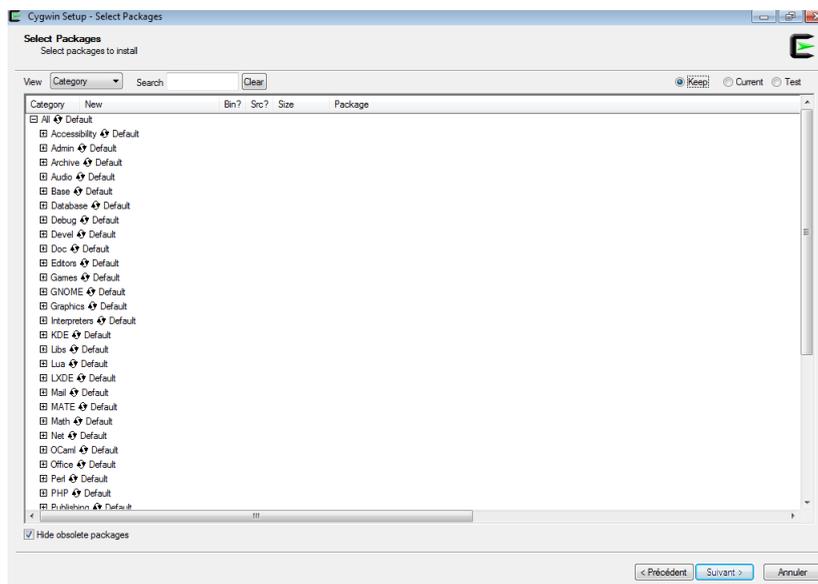


FIGURE 2.35 – Sélection des options d’installation : Sélection des packages

Remarque : Il convient de cliquer sur All de manière à faire apparaître Install (et non Default : qui oublie des paquets importants).

On trouvera ci-dessous une proposition des packages à sélectionner en plus de la version de base proposée par l’installateur dans la catégorie « current ».

Attention : Ne pas désélectionner les éléments de base proposés par le programme d’installation car ils garantissent le bon fonctionnement d’un système de manière cohérente.

Catégorie	Package supplémentaire	Fonction
Admin	Sélectionner l’ensemble de la catégorie	fonctions supplémentaires d’administration possibles
Devel	cygwin32-gcc-core cygwin32-gcc-g++ libgcc1 gdb make	base du compilateur gcc pour langage C/C++ ajout supplémentaire pour C++ Librairies C debugger pour C/C++ utilitaire make pour création de programmes
Libs	libmpfr4	précision pour les nombres de type float
Perl	mention descriptive : perl programming language	permet d’utiliser le langage perl grâce à son interpréteur
Python	mention descriptive : langage interpreter pour python2 mention descriptive : langage interpreter pour python3	permet d’utiliser le langage Python2 grâce à son interpréteur permet d’utiliser le langage Python3 grâce à son interpréteur
Util	pwget	Utilitaire de téléchargement
Net	inetutils openssh	Utilitaire réseaux gestion de connexion à distance via serveur ssh

TABLE 2.1 – Proposition de sélection de package

Remarque : Dans tous les cas, si l’on souhaite installer de nouveaux packages, il suffit de relancer le programme de setup et d’ajouter la nouvelle sélection.

Un écran de résolution des dépendances¹⁵ peut alors s'afficher selon la configuration choisie. Il suffit de mettre « suivant » sans décocher la case d'autorisation :

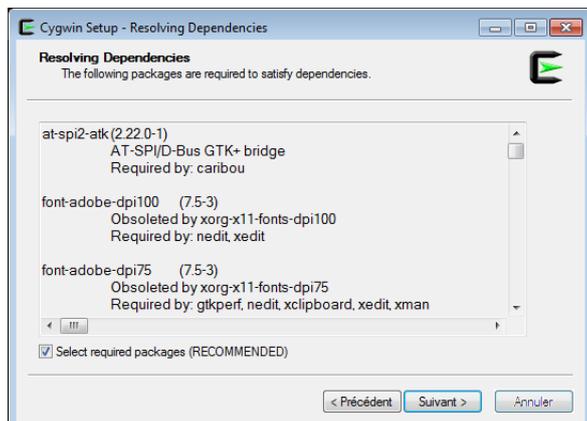


FIGURE 2.36 – Installation : autorisation pour les dépendances non-satisfaites

- L'installation proprement dite commence, elle se découpe en trois phases :
- téléchargement des sources dans un dossier ;
 - vérification que les programmes téléchargés sont bien ceux que l'on attend (sécurité, SHA) ;
 - installation proprement dite.

Ces trois phases peuvent prendre du temps (selon le nombre de packages sélectionnés), et de l'espace disque (pour les mêmes raisons). Dans tous les cas l'utilisateur voit l'avancée de l'installation grâce à un écran d'information :

15. Il s'agit de programmes supplémentaires nécessaires pour le bon fonctionnement d'autres programmes.

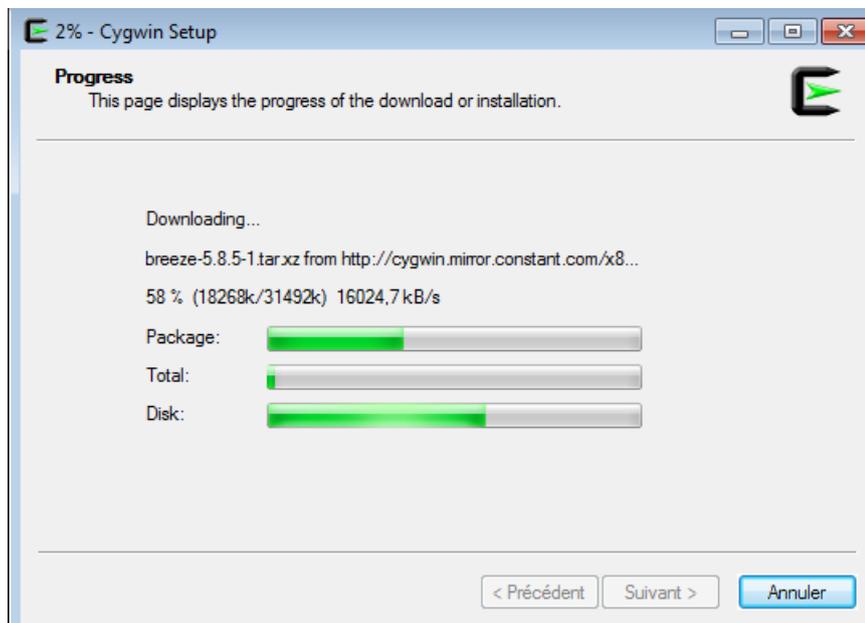


FIGURE 2.37 – Installation en cours

Normalement après le déroulement de l'installation on obtient :

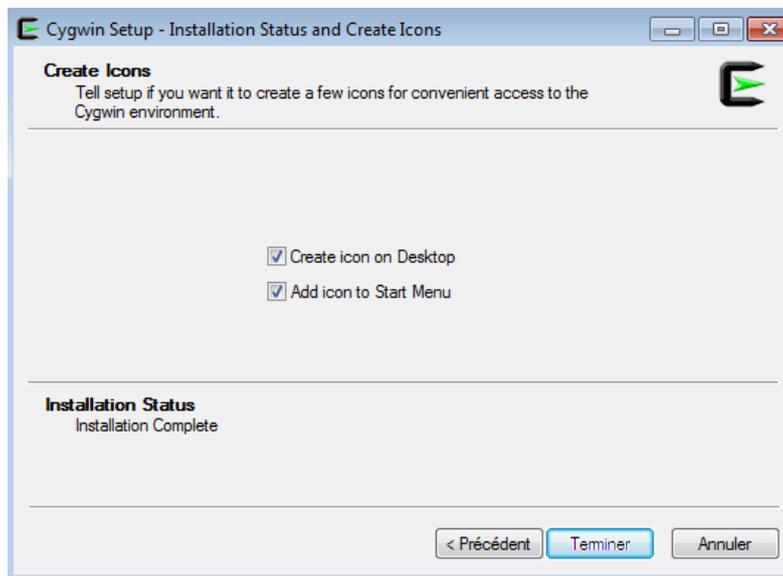
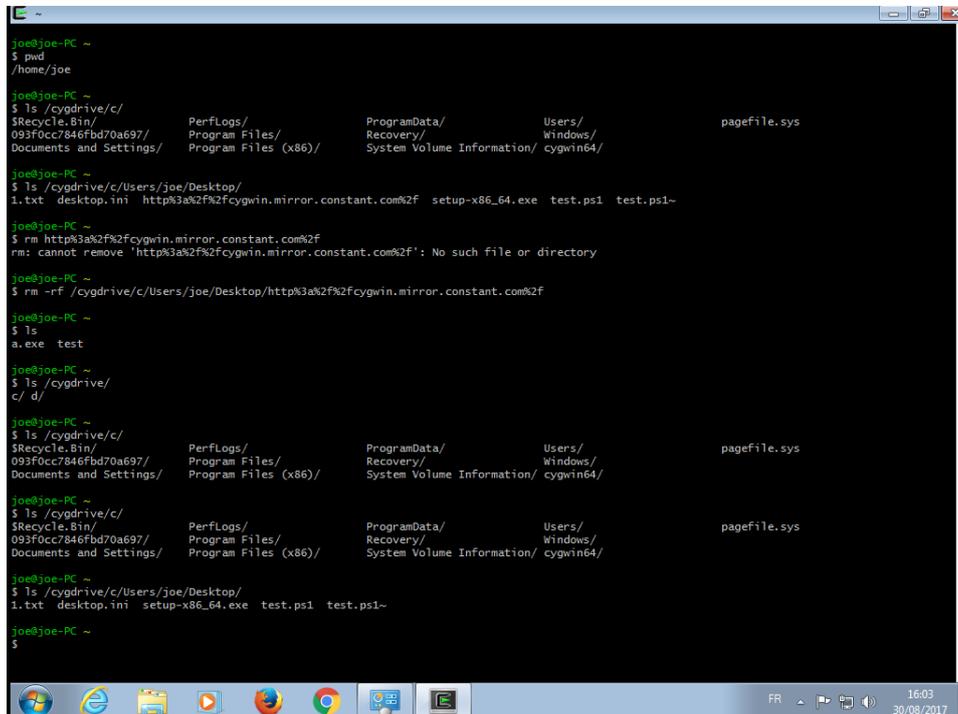


FIGURE 2.38 – Fin de l'installation

Et l'on peut enfin utiliser Cygwin :



```
joe@joe-PC ~
$ pwd
/home/joe

joe@joe-PC ~
$ ls /cygdrive/c/
$Recycle.Bin/          PerfLogs/          ProgramData/        Users/              pagefile.sys
093F0cc7846Fbd70a697/ Program Files/      Recovery/           Windows/
Documents and Settings/ Program Files (x86)/ System Volume Information/ cygwin64/

joe@joe-PC ~
$ ls /cygdrive/c/Users/joe/Desktop/
1.txt  desktop.ini  http%3a%2f%2fcygwin.mirror.constant.com%2f  setup-x86_64.exe  test.ps1  test.ps1~

joe@joe-PC ~
$ rm http%3a%2f%2fcygwin.mirror.constant.com%2f
rm: cannot remove 'http%3a%2f%2fcygwin.mirror.constant.com%2f': No such file or directory

joe@joe-PC ~
$ rm -rf /cygdrive/c/Users/joe/Desktop/http%3a%2f%2fcygwin.mirror.constant.com%2f

joe@joe-PC ~
$ ls
a.exe  test

joe@joe-PC ~
$ ls /cygdrive/
c/  d/

joe@joe-PC ~
$ ls /cygdrive/c/
$Recycle.Bin/          PerfLogs/          ProgramData/        Users/              pagefile.sys
093F0cc7846Fbd70a697/ Program Files/      Recovery/           Windows/
Documents and Settings/ Program Files (x86)/ System Volume Information/ cygwin64/

joe@joe-PC ~
$ ls /cygdrive/c/
$Recycle.Bin/          PerfLogs/          ProgramData/        Users/              pagefile.sys
093F0cc7846Fbd70a697/ Program Files/      Recovery/           Windows/
Documents and Settings/ Program Files (x86)/ System Volume Information/ cygwin64/

joe@joe-PC ~
$ ls /cygdrive/c/Users/joe/Desktop/
1.txt  desktop.ini  setup-x86_64.exe  test.ps1  test.ps1~

joe@joe-PC ~
$
```

FIGURE 2.39 – Cygwin en action

Remarques : Le dossier de téléchargement des sources n'étant plus nécessaire une fois les packages installés, on pourra l'effacer afin de gagner de l'espace sur le disque.

5 Alternative 2 : Bash sous windows 10

Ces dernières années, Windows inclut des fonctionnalités se rapprochant du monde Unix. Sous Windows 10 il est désormais possible d'activer un shell bash correspondant (moyennant quelques pertes de fonctionnalités) à une distribution Ubuntu.

Pour cela il faut commencer par activer les fonctionnalités et passer en « mode développeur ». Dans les paramètres :

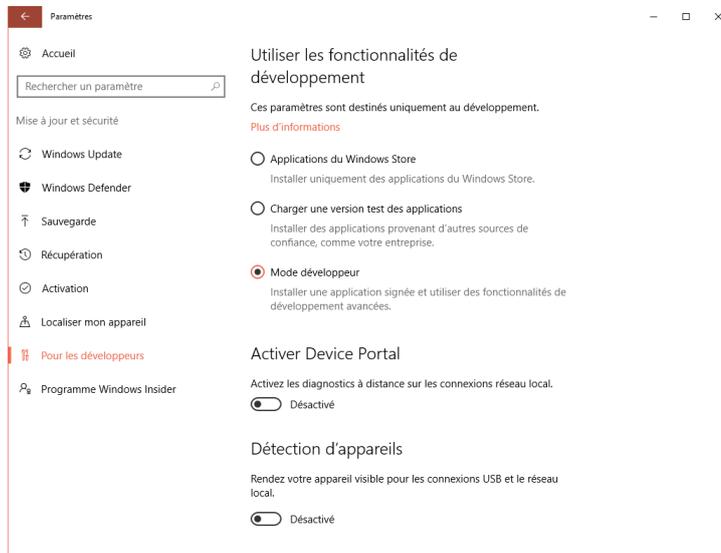


FIGURE 2.40 – Mode développeur

On va ensuite dans l'onglet « programmes et fonctionnalités » (Panneau de configuration/paramètres) pour cliquer sur « Activer ou désactiver des fonctionnalités windows » :

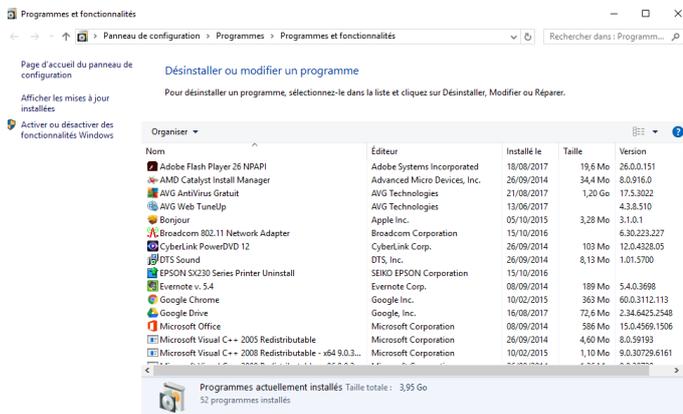


FIGURE 2.41 – Activation des fonctionnalités

On coche ensuite la case correspondante aux « Sous-système Windows pour Linux (bêta) » :

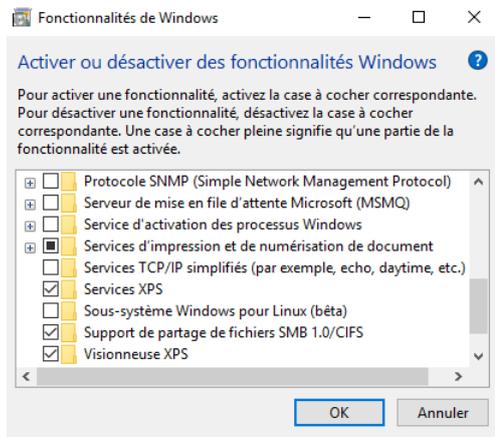


FIGURE 2.42 – Sous-système Windows pour Linux (bêta)

Windows télécharge¹⁶ alors les programmes nécessaires. Après quoi, pour activer ces éléments il faut redémarrer :

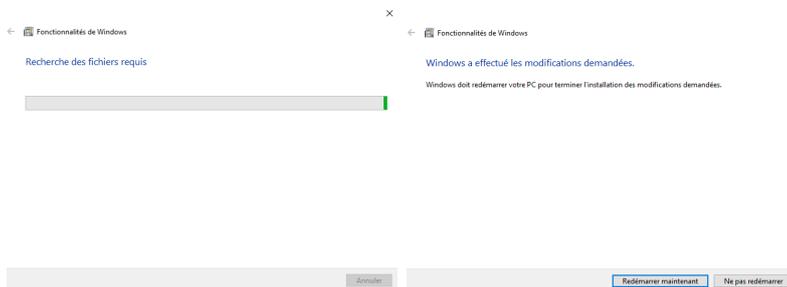


FIGURE 2.43 – Recherche de fichiers avant de redémarrer.

Après avoir redémarré, on peut taper « Bash » dans l'explorateur pour pouvoir lancer le programme une première fois, il faut alors répondre « o » pour confirmer l'installation :

16. Il faut bien sûr une connexion internet active.

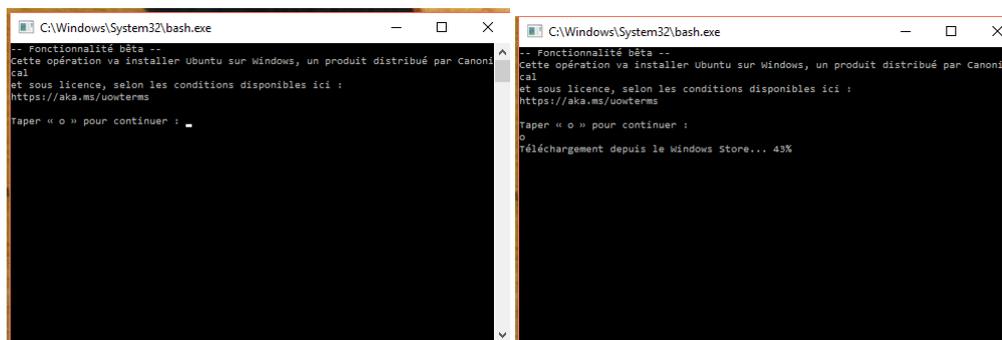


FIGURE 2.44 – Initialisation du bash

Après un certain temps de téléchargements et d’installations, il faut répondre à quelques questions servant au paramétrage du système tel que le nom d’utilisateur ou le mot de passe¹⁷ :

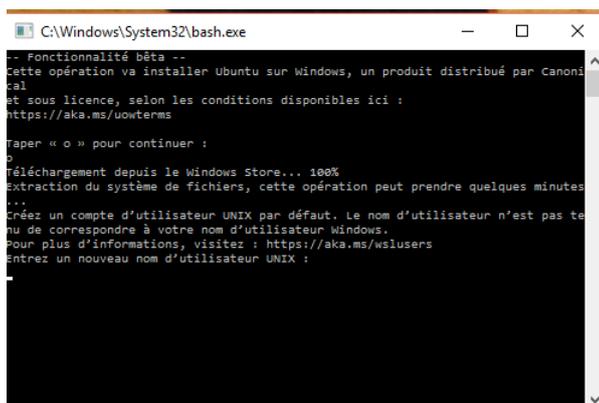


FIGURE 2.45 – Paramétrages supplémentaires

On peut ensuite utiliser le programme en tapant bash dans l’explorateur :

17. Attention : les mots de passe n’apparaissent pas en clair sous unix mais sont pris en compte.

```
joe@PCLESIE ~  
sudo: impossible de déterminer le nom de l'hôte PCLESIE  
Paramétrage de libfreetype6:amd64 (2.5.2-1ubuntu2.8) ...  
Traitement des actions différées (« triggers ») pour man-db (2.6.7.1-1ubuntu1) ...  
Paramétrage de libcrypt11:amd64 (1.5.3-2ubuntu4.5) ...  
Paramétrage de libtasn1-6:amd64 (3.4-3ubuntu0.5) ...  
Paramétrage de libdrm2:amd64 (2.4.67-1ubuntu0.14.04.2) ...  
Paramétrage de libroken18-heimdal:amd64 (1.6-git20131207+dfsg-1ubuntu1.2) ...  
Paramétrage de python3-update-manager (1:0.196.24) ...  
Paramétrage de libwind0-heimdal:amd64 (1.6-git20131207+dfsg-1ubuntu1.2) ...  
Paramétrage de libxpat1:amd64 (2.1.0-4ubuntu1.4) ...  
Paramétrage de libicu52:amd64 (52.1-3ubuntu0.6) ...  
Traitement des actions différées (« triggers ») pour ureadahead (0.100.0-16) ...  
Paramétrage de libgnutls26:amd64 (2.12.23-12ubuntu2.8) ...  
Paramétrage de libheimbase1-heimdal:amd64 (1.6-git20131207+dfsg-1ubuntu1.2) ...  
Paramétrage de libasn1-8-heimdal:amd64 (1.6-git20131207+dfsg-1ubuntu1.2) ...  
Paramétrage de ntpdate (1:4.2.6.p5+dfsg-3ubuntu2.14.04.12) ...  
Installation de la nouvelle version du fichier de configuration /etc/network/if-  
up.d/ntpdate ...  
Paramétrage de update-manager-core (1:0.196.24) ...  
Paramétrage de libcrypto4-heimdal:amd64 (1.6-git20131207+dfsg-1ubuntu1.2) ...  
Paramétrage de librtmp0:amd64 (2.4+20121230.gitdf6c518-1ubuntu0.1) ...  
Paramétrage de libgnutls-openssl127:amd64 (2.12.23-12ubuntu2.8) ...  
Paramétrage de update-notifier-common (0.154.1ubuntu3) ...  
Paramétrage de libhx509-5-heimdal:amd64 (1.6-git20131207+dfsg-1ubuntu1.2) ...  
Paramétrage de libkrb5-26-heimdal:amd64 (1.6-git20131207+dfsg-1ubuntu1.2) ...  
Paramétrage de libheimt1m0-heimdal:amd64 (1.6-git20131207+dfsg-1ubuntu1.2) ...  
Paramétrage de libgssapi3-heimdal:amd64 (1.6-git20131207+dfsg-1ubuntu1.2) ...  
Paramétrage de libldap-2.4-2:amd64 (2.4.31-1+nmu2ubuntu8.4) ...
```

FIGURE 2.46 – Bash an action sous Windows 10

Chapitre 3

Utilisation du système en ligne de commande

Après la phase d'installation nous allons maintenant aborder la partie programmation en Bash en utilisant une interface en ligne de commande (ou « CLI » pour « *command line interface* »).

1 Rudiment du Bash

1.1 Qu'est ce que le Bash ?

Le Bash est un interpréteur de commandes qui permet d'exécuter les instructions que l'on saisit au clavier ou que l'on a écrites dans un script. Les interpréteurs de commande de ce type sont appelés « *shell* », qui signifie coquille. Cette image vient du fait que le système d'exploitation possède un élément central qui lui permet de dialoguer avec le matériel à sa disposition, le noyau ou « *Kernel* ». Ce noyau est entouré d'une partie supérieure, le Shell donc, qui est un moyen de dialoguer avec lui par le biais de commandes. C'est une interface textuelle entre l'utilisateur et les possibilités offertes par le noyau qui exploite le matériel.

Le Bash ou *Bourne Again Shell*¹ est sans doute le shell de référence² sous Linux. Il est présent sous MacOS et est aussi exécutable sous Windows.

Pour l'utiliser sous Ubuntu³ on lancera le programme « terminal ». Pour cela plusieurs possibilités :

- Cherchez le terminal dans l'explorateur
- Taper un raccourci avec les touches : ctrl gauche+alt gauche+t

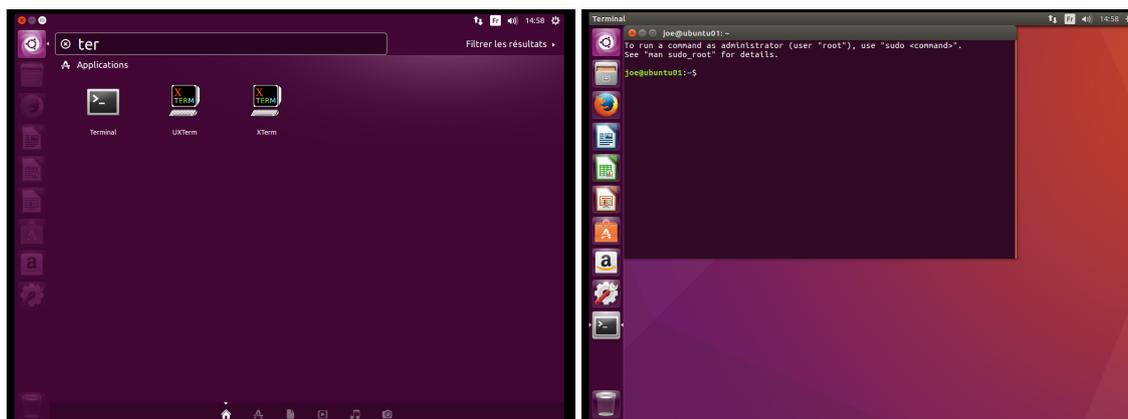


FIGURE 3.1 – Lancement du terminal

On peut ensuite garder le terminal comme raccourci dans le lanceur et le paramétrer selon ses envies (couleurs, degré de transparence, etc) :

1. Jeux de mots avec le *Bourne Shell* (*sh*) dont il est issu.
2. La liste des Shell est nombreuse, on pourra citer de manière non exhaustive :
 - sh : Thompson Shell qui n'existe plus
 - sh : Bourne Shell a remplacé le précédent
 - sh : Bourne Again Shell évolution de sh avec des fonctionnalités csh et ksh
 - ksh : Korn Shell évolution du Bourne Shell
 - csh : C Shell avec une structuration du langage C
 - zsh : Z Shell évolution du Korn Shell
 - bat, cmd, btm : shell Dos pour Windows NT
 - ps1, psm1, psd1 : powershell, shell novateur actuel de Windows orienté objet
3. Dans les autres cas d'installation il suffit
 - pour Windows selon les choix : de lancer Cygwin ou le Bash Ubuntu
 - pour Mac de lancer un terminal

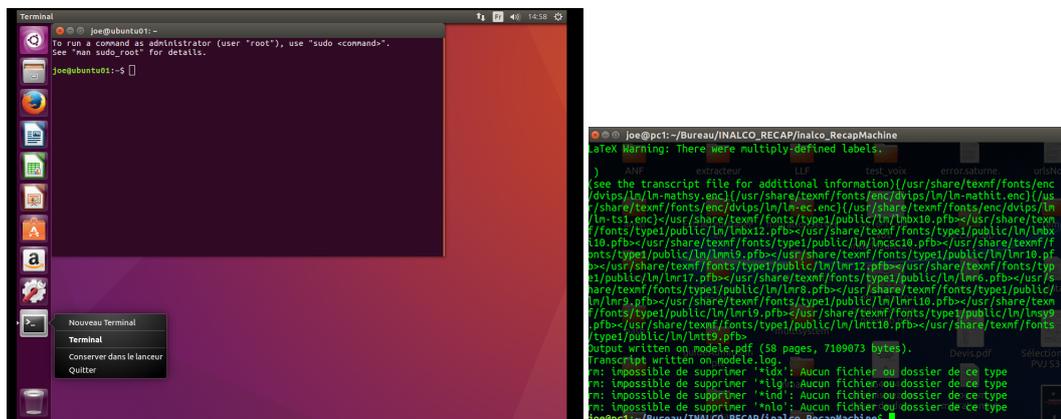


FIGURE 3.2 – Lancement du terminal

On interagira ensuite sur le système en tapant directement des commandes dans ce terminal.

1.2 Qu'est ce qu'une commande ?

Les interactions textuelles entre l'utilisateur et l'ordinateur forment un ensemble de mots clés que l'on appelle des commandes. Elles se présentent sous la forme suivante, les éléments entre < > sont optionnels :

commande unix

commande <-option> <argument(s)>

La commande se trouvant à gauche en rouge peut-être un mot clé du langage bash ou un programme installé dans l'OS. On peut tester avec firefox :

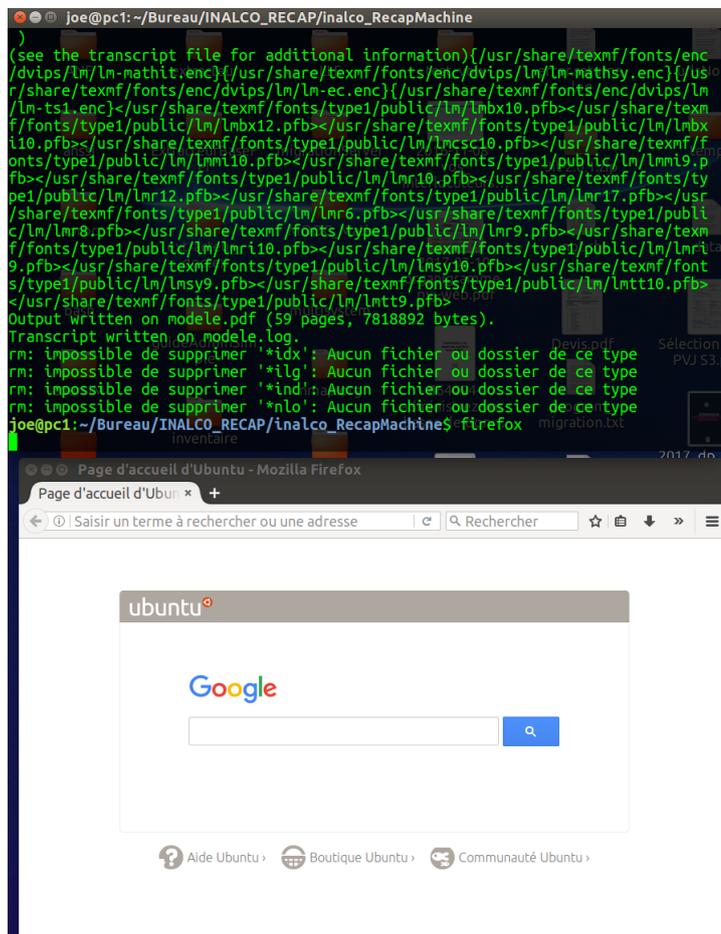


FIGURE 3.3 – Appel du navigateur firefox par le terminal bash

Globalement et grossièrement, on peut classer les commandes en deux types distincts⁴ :

- les commandes actives : pour effectuer un changement, lancer un programme, etc
- les commandes informatives : qui permettent d’afficher des informations dans le terminal ou de filtrer des informations

La commande « *pwd* » (*Print Working Directory*) par exemple, permet de connaître l’endroit où on se situe sur le disque :

4. On effectue ici une simplification excessive uniquement dans un but explicatif, la frontière entre commandes étant difficilement traçable et très perméable.

```
1 joe@pc:~$ pwd
2 /home/joe
```

On observe dans la séquence précédente du terminal que l'utilisateur « joe » est connecté sur la machine « pc »⁵ et qu'il lance la commande « pwd ». La réponse obtenu est « /home/joe » qui est le dossier dans lequel nous nous trouvons. La commande « ls » (*LiSt*) permet de connaître le contenu de ce dossier :

```
1 joe@pc:~$ ls
2 BaseXData
3 Bibliothèque calibre
4 binBash
5 binC
6 binPerl
7 binPython
8 Bureau
9 Calibre Library
10 Diagramme1.dia.autosave
11 Documents
12 Dropbox
13 Erreurlog
14 exemples.desktop
15 hs_err_pid5617.log
16 hs_err_pid5653.log
17 Images
18 ISO
19 JAVASCRIPT
20 [...]
```

Par défaut, la commande « ls » considère que le dossier sur lequel on l'interroge est le dossier actuel, celui dans lequel on se situe. On peut lui fournir un autre dossier comme argument. Cet argument optionnel apparaissait en gris, plus haut, lors de l'explication de ce qu'est une commande.

```
1 joe@pc:~$ ls /home/
2 joe lost+found
```

Ici, on lui donne par exemple le dossier « /home » en argument, la réponse de la commande « ls » indique que ce dossier ne contient que deux éléments : le dossier « joe » et le dossier « lost+found ».

Enfin, on peut renseigner des options qui changent le comportement de la commande (les options apparaissaient en bleu dans la définition de commande Unix). Par exemple l'option « -l » permet d'obtenir plus d'informations en utilisant un format d'affichage plus long :

5. Le format : utilisateur@machine rappellera sans doute à certains le format des adresses e-mail ...

```

1  joe@pc:~$ ls -l
2  total 45164
3  drwxr-xr-x 12 joe vboxusers 4096 déc. 13 2016 BaseXData
4  drwxr-xr-x 2 joe vboxusers 4096 janv. 27 2017 Bibliothèque calibre
5  drwxrwxr-x 7 joe vboxusers 4096 déc. 30 2016 binBash
6  drwxr-xr-x 4 joe vboxusers 4096 janv. 2 2017 binC
7  drwxrwxr-x 7 joe vboxusers 4096 juin 28 2016 binPerl
8  drwxr-xr-x 6 joe vboxusers 4096 juin 12 11:43 binPython
9  drwxr-xr-x 47 joe vboxusers 4096 sept. 11 18:04 Bureau
10 drwxr-xr-x 3 joe vboxusers 4096 janv. 27 2017 Calibre Library
11 -rw-rw-r-- 1 joe vboxusers 1081 mai 24 2016 Diagramme1.dia.autosave
12 drwxr-xr-x 3 joe vboxusers 4096 mai 12 17:10 Documents
13 drwx----- 37 joe vboxusers 4096 août 25 10:37 Dropbox
14 -rw-r--r-- 1 joe vboxusers 1069 oct. 10 2016 Erreurlog
15 -rw-r--r-- 1 joe vboxusers 8980 avril 13 2016 exemples.desktop
16 -rw-r--r-- 1 joe vboxusers 82446 juin 28 2016 hs_err_pid5617.log
17 -rw-r--r-- 1 joe vboxusers 82519 juin 28 2016 hs_err_pid5653.log
18 drwxr-xr-x 3 joe vboxusers 4096 avril 25 11:14 Images
19 drwxr-xr-x 2 joe vboxusers 4096 oct. 10 2016 ISO
20 drwxr-xr-x 9 joe vboxusers 4096 sept. 6 2016 JAVASCRIPT
21 [...]

```

Il existe deux façon d'utiliser les options. Par exemple l'option « -s » (un tiret, format court) peut aussi s'écrire « -size » (deux tirets, format long) et donnera le même résultat :

```

1  joe@pc:~$ ls --size
2  total 45160
3      4 BaseXData
4      4 Bibliothèque calibre
5      4 binBash
6      4 binC
7      4 binPerl
8      4 binPython
9      4 Bureau
10     4 Calibre Library
11     0 content
12     4 Diagramme1.dia.autosave
13     4 Documents
14     4 Dropbox
15     4 Erreurlog
16    12 exemples.desktop
17    84 hs_err_pid5617.log
18    84 hs_err_pid5653.log
19     4 Images
20     4 ISO
21     4 JAVASCRIPT
22 [...]
23  joe@pc:~$ ls -s
24  total 45164

```

```

25     4 BaseXData
26     4 Bibliothèque calibre
27     4 binBash
28     4 binC
29     4 binPerl
30     4 binPython
31     4 Bureau
32     4 Calibre Library
33     4 content
34     0 content2
35     4 Diagramme1.dia.autosave
36     4 Documents
37     4 Dropbox
38     4 Erreurlog
39     12 examples.desktop
40     84 hs_err_pid5617.log
41     84 hs_err_pid5653.log
42     4 Images
43     4 ISO
44     4 JAVASCRIPT
45     [...]

```

Si toutes les options ne permettent pas cette double écriture, le format court présente l'avantage de cumuler rapidement plusieurs options :

```

1  joe@pc:~$ ls -larth
2  total 12M
3  drwxrwxrwx 14 joe joe 4,0K nov. 9 2009 piste jap
4  -rw-r--r-- 1 joe joe 675 févr. 25 2016 .profile
5  -rw-r--r-- 1 joe joe 8,8K févr. 25 2016 examples.desktop
6  -rw-r--r-- 1 joe joe 220 févr. 25 2016 .bash_logout
7  -rw-r--r-- 1 joe joe 25 févr. 25 2016 .dmrc
8  drwxr-xr-x 2 joe joe 4,0K févr. 25 2016 Public
9  drwxr-xr-x 2 joe joe 4,0K févr. 25 2016 Modèles
10 drwxr-xr-x 2 joe joe 4,0K févr. 25 2016 Documents
11 drwx----- 3 joe joe 4,0K févr. 25 2016 .local
12 drwx----- 3 root root 4,0K févr. 25 2016 .dbus
13 drwx----- 3 joe joe 4,0K févr. 25 2016 .compiz
14 -rw----- 1 joe joe 0 févr. 25 2016 .mysql_history
15 drwx----- 3 joe joe 4,0K févr. 25 2016 .pki
16 drwx----- 2 joe joe 4,0K févr. 25 2016 .gnome2_private
17 drwx----- 4 joe joe 4,0K févr. 25 2016 .mozilla
18 drwxr-xr-x 2 root root 4,0K févr. 27 2016 .rpmdb
19 -rw-rw-r-- 1 joe joe 82 févr. 28 2016 .gitconfig
20 drwx----- 2 root root 4,0K avril 19 2016 .gnupg
21 -rw-rw-r-- 1 joe joe 12M avril 19 2016 treetagger.tar.gz
22 drwxr-xr-x 6 root root 4,0K avril 19 2016 .cpan
23 drwxrwxr-x 7 joe joe 4,0K avril 19 2016 treetagger
24 drwxr-xr-x 4 root root 4,0K avril 19 2016 ..
25 drwxr-xr-x 7 joe joe 4,0K mai 9 2016 binPerl
26 [...]

```

On cumule dans cet exemple les options « -a (all), -l (list), -r (reverse), -t(time), -h (human readable) ».

Nous avons vu que les commandes permettent d'obtenir de l'information et nous avons rapidement parlé des commandes d'action, ainsi que du positionnement des fichiers/dossiers sur le disque. Observons à présent comment utiliser des commandes d'action pour nous déplacer sur le disque.

1.3 Déplacement dans l'arborescence

Les Unix, à la création du File System, établissent un ingénieux système d'inclusion entre les dossiers/fichiers. Ce système est un standard que l'on appelle l'arborescence⁶. On doit ce nom à sa ressemblance avec un arbre renversé. L'élément initial s'appelle d'ailleurs racine ou « *root* » et est symbolisé par « / ». Cette racine est le dossier principal qui contient lui-même un certain nombre d'autres dossiers système :

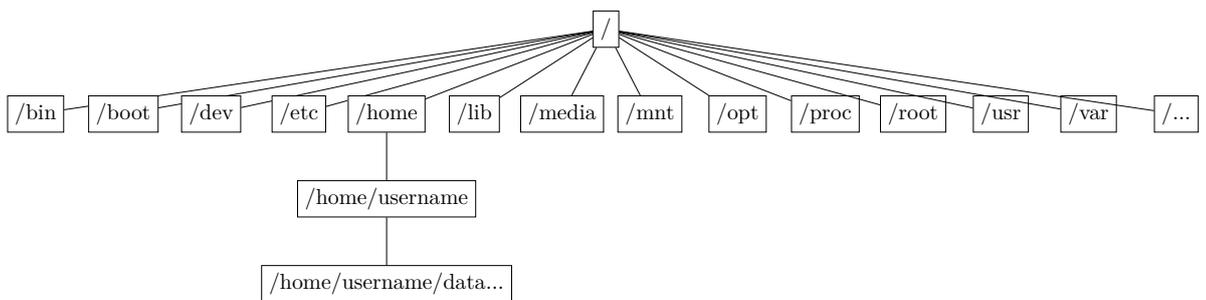


FIGURE 3.4 – Arborescence sous Unix

Parmi ces dossiers, on distingue le répertoire « home » qui contient les données de tous les utilisateurs qui ont un compte sur le système. Ces données sont conservées dans des dossiers correspondant aux noms de ces mêmes utilisateurs.

Lorsqu'on se connecte en tant qu'utilisateur dans un terminal, on arrive dans le répertoire par défaut qui est `/home/<nom_utilisateur>`.

On peut se déplacer dans l'arborescence en utilisant la commande « `cd` » (Change Directory). Par exemple puisque chaque utilisateur possède un dossier Bureau (Desktop) on peut y accéder par :

```
1 joe@pc1:~$ pwd
2 /home/joe
3 joe@pc1:~$ cd Bureau
4 joe@pc1:~/Bureau$ pwd
```

6. Ce standard est défini par la norme de la hiérarchie des systèmes de fichiers ou *Filesystem Hierarchy Standard* (FHS) voir https://fr.wikipedia.org/wiki/Filesystem_Hierarchy_Standard.

```
5 /home/joe/Bureau
6 joe@pc1:~/Bureau$
```

On retiendra ici plusieurs choses :

- Lorsque l'on se déplace dans le dossier Bureau, l'invite de commande change à « ~/Bureau », ceci car le tilde est un synonyme du répertoire par défaut /home/joe.
- Unix fait la différence entre majuscule et minuscule, on dit qu'il est sensible à la casse. Ainsi, « Bureau » et « bureau » n'indiquent pas le même répertoire.
- Quand on commence à composer un nom, on peut utiliser une aide pratique du système qui complète automatiquement le texte que l'on est en train de taper. Ainsi, l'auto-complétion, activée par la touche tabulation, permet de compléter automatiquement la commande que l'on est en train d'écrire et permet un gain de temps significatif.

Les déplacements dans l'arborescence sont possibles de deux manières « absolue » ou « relative ». Un déplacement absolu consiste à indiquer exactement tout le détail du chemin à parcourir :

```
1 # déplacement à la racine
2 joe@pc1:~$ cd /
3 # nous sommes bien dans le dossier racine ? vérification :
4 joe@pc1:/$ pwd
5 /
6 # déplacement dans /home/joe/ mais si on ne connaît plus la suite du chemin
  => auto-complétion avec tabulation => le système proposera une série
  de choix possibles, en écrivant un mot, on peut le compléter avec la
  même touche
7 joe@pc1:/$ cd /home/joe/
8 .AMC.d/ binPython/ Dropbox/ Images/ Outils/ .thunderbird/
9 .anthy/ Bureau/ .dropbox-dist/ .java/ .pki/ .tigersearch/
10 .aqbanking/ .cache/ .fltk/ .kde/ Projets-QCM/ TIGERSearchTools/
11 Bibliothèque calibre/ .compiz/ .gconf/ Latex/ Public/ tools/
12 binBash/ .config/ .gdfuse/ .local/ R/ treetagger/
13 binC/ .cpan/ .gimp-2.8/ .macromedia/ .rstudio/ .uml/
14 binHTML/ .cpanm/ .gnome/ Modèles/ .ssh/ Vidéos/
15 binJava/ .dbus/ .gnome2/ .mozc/ Téléchargements/ .vim/
16 binJavascript/ Documents/ .gnupg/ .mozilla/ .texmf-var/ .zenmap/
17 binPerl/ .dropbox/ .hplip/ Musique/ .thumbnails/
18 # Idem que précédemment placé dans le bureau mais on obtient la composition
  du chemin avec tabulation
19 joe@pc1:/$ cd /home/joe/Bureau/
20 ANF/ DOCS/ INALCO_RECAP/ migrationServer/ ressources perso/
21 anssi/ document/ inventaire/ MOOC_Math/ temp/
22 aoroc/ DSI/ Jit/ multisystem/ TestParsingYatea/
23 bash/ extract/ JourneesPerl2017/ nmap.org/ testPepperAnnis/
24 coursIM/ extracteur/ kaisersly.github.io/ Note/ testProjet/
25 COURS_INALCO/ ExtracteurBaseFer/ KALI/ partage_vm_samba/ testsql/
```

```

26 coursTrad/ extracteur_doctex/ ktouch/ PERSO/ test_voix/
27 DeveloppementExtracteur/ guideAdminSimple/ LLF/ python_google_ngram/
28 joe@pc1:/$ cd /home/joe/Bureau/extracteur
29 extracteur/ extracteur_doctex/
30 # Déplacement dans le dossier extracteur
31 joe@pc1:/$ cd /home/joe/Bureau/extracteur/
32 bin/ data/ environnements/ .git/ lib/ public/ runScript/ t/ temp/ views/
33 joe@pc1:/$ cd /home/joe/Bureau/extracteur/
34 # vérification du dossier
35 joe@pc1:~/Bureau/extracteur$ pwd
36 /home/joe/Bureau/extracteur

```

L'autre façon de se déplacer est d'utiliser un déplacement dit relatif. On va changer de répertoire relativement à celui dans lequel on se situe. Pour cela il faut connaître deux symboles qui sont « . » et « .. » :

- le point représente le dossier actuel, dans lequel on se situe ;
- le double point représente le répertoire supérieur.

```

1 # où sommes nous ?
2 joe@pc1:~/Bureau/extracteur$ pwd
3 /home/joe/Bureau/extracteur
4 # vers où peut-on se diriger ? l'option -a de la commande ls permet
   d'afficher tous les éléments du dossier y compris les fichiers cachés.
   On voit apparaître "." et ".."
5 joe@pc1:~/Bureau/extracteur$ ls -a
6 . bin cpanfile environnements .gitignore Makefile.PL MANIFEST.SKIP
   README.html runExtractor.sh t views
7 .. config.yml data .git lib MANIFEST public README.md runScript temp
8 #On remonte vers le répertoire supérieur
9 joe@pc1:~/Bureau/extracteur$ cd ..
10 joe@pc1:~/Bureau$ pwd
11 /home/joe/Bureau
12 # nous sommes dans "Bureau", auto-complétion
13 joe@pc1:~/Bureau$ cd ./
14 ANF/ DOCS/ INALCO_RECAP/ migrationServer/ ressources perso/
15 anssi/ document/ inventaire/ MOOC_Math/ temp/
16 aoroc/ DSI/ Jit/ multisystem/ TestParsingYatea/
17 bash/ extract/ JourneesPerl2017/ nmap.org/ testPepperAnnis/
18 coursIM/ extracteur/ kaisersly.github.io/ Note/ testProjet/
19 COURS_INALCO/ ExtracteurBaseFer/ KALI/ partage_vm_samba/ testsql/
20 coursTrad/ extracteur_doctex/ ktouch/ PERSO/ test_voix/
21 DeveloppementExtracteur/ guideAdminSimple/ LLF/ python_google_ngram/
22 # Déplacement vers le dossier document
23 joe@pc1:~/Bureau$ cd ./document/
24 joe@pc1:~/Bureau/document$ pwd
25 /home/joe/Bureau/document

```

Après avoir vu comment se déplacer nous allons voir comment créer, copier, supprimer et déplacer les fichiers/dossiers.

- 1.4 Création, suppression, déplacement/copie de fichiers/répertoires
- 1.5 Flux, redirection, lecture de fichier
- 1.6 Calculs
- 1.7 Premier programme, variables
- 1.8 Instruction de contrôle 1 : structure conditionnelle
- 1.9 Instruction de contrôle 2 : boucle
- 2 Programmation Bash avancée**
 - 2.1 Filtres, Joker en Bash
 - 2.2 Fonctions
 - 2.3 Insertion de commandes
 - 2.4 Variables complexes : Tableaux et Tables de hash
 - 2.5 Sous-processus et expressions parenthésées
 - 2.6 Processus en arrière plan et redirection sur plusieurs flux
- 3 Introduction à l'administration système**
 - 3.1 Droits d'accès, groupe/utilisateur et super-utilisateur
 - 3.2 Gestion des packages
 - 3.3 Connexion à distance avec ssh

Bibliographie

- [Barthélemy, 2017a] Barthélemy, F. (2017a). Premier Programme en Java. <https://www.fun-mooc.fr/courses/CNAM/01009S02/session02/about>. [En ligne sur Fun Univ accessible par période; dernier accès 21/08/2017]. – Cité page 7.
- [Barthélemy, 2017b] Barthélemy, F. (2017b). Socle en Informatique. <https://www.fun-mooc.fr/courses/course-v1:itii+119003+session01/about>. [En ligne sur Fun Univ accessible par période; dernier accès 21/08/2017]. – Cité page 7.
- [developpez.com, 2017] developpez.com (01/09/2017). Enquête sur l'évolution récente de Linux. <https://www.developpez.com/actu/158200/Linux-franchit-la-barre-des-3-pourcent-sur-les-PC-et-se-rapproche-de-macOS-5-84-> [dernier accès 07/09/2017]. – Cité page 15.
- [developpez.com, 2015] developpez.com (20/01/2015). Enquête sur les Serveurs Web. <https://www.developpez.com/actu/80364/Marche-des-serveurs-web-Apache-toujours-en-tete-Microsoft-IIS-continue-son-decli> [dernier accès 21/08/2017]. – Cité page 15.
- [journaldunet.com, 2012] journaldunet.com (20/03/2012). Enquête sur les clouds services. <http://www.journaldunet.com/solutions/expert/51207/1-ecrasante-domination-de-linux-dans-le-cloud---vers-90--de-part-de-marche.shtml>. [dernier accès 21/08/2017]. – Cité page 15.
- [lecrabeinfo.net, 2016] lecrabeinfo.net (14/05/2016). Rôle des processeurs Intel. <https://lecrabeinfo.net/le-role-des-processeurs-et-de-leurs-coeurs.html>. [dernier accès 21/08/2017]. – Cité page 7.
- [Sharrock, 2016a] Sharrock, R. (2016a). Mooc(s) sur le langage C. <https://www.fun-mooc.fr/courses/MinesTelecom/04019S02/session02/about>. [En ligne sur Fun Univ accessible par période; dernier accès 21/08/2017]. – Cité pages 7 et 15.
- [Sharrock, 2016b] Sharrock, R. (2016b). Mooc(s) sur le langage C. <https://www.fun-mooc.fr/courses/MinesTelecom/04020S02/session02/about>. [En ligne sur Fun Univ accessible par période; dernier accès 21/08/2017]. – Cité page 7.

- [Wikipédia, a] Wikipédia. Architecture de Von Neumann. https://fr.wikipedia.org/wiki/Architecture_de_von_Neumann. – Cité pages 7 et 8.
- [Wikipédia, b] Wikipédia. Histoire de l'informatique. https://fr.wikipedia.org/wiki/Histoire_de_l%27informatique. – Cité pages 7 et 8.
- [Wikipédia, c] Wikipédia. Système d'exploitation. https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation. – Cité page 14.