

Veille technologique

L'évolution des systèmes Linux et de l'open source dans le milieu des systèmes d'exploitations

Antoine Lonné

École d'ingénieurs CESI Pau
antoine@antoinelonne.dev

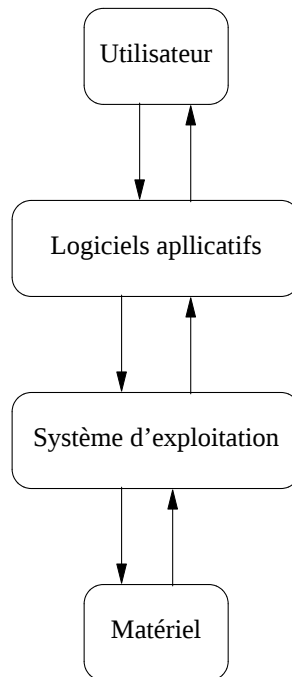
Linux, ce système d'exploitation à la communauté infinie, si répandu dans le milieu geek et des serveurs mais si peu connu du grand public, quelles sont ses origines, ses inspirations, sa communauté qui l'a fait naître ?

Janvier - Février 2021

1. Introduction

Avant tout chose, il me semble important de définir qu'est ce qu'un système d'exploitation, quel est son rôle. Un système d'exploitation (ou OS pour Operating System en anglais) est un ensemble de logiciels qui, comme son nom l'emploie, permet d'exploiter les ressources matérielles il est aussi un support pour des logiciels qui lui sont propres. Ces ressources matérielles peuvent être par exemple la mémoire vive, les périphériques de stockage, le processeur, les périphériques externes ou carte d'extension (carte graphique en PCIe par exemple). Le noyau du système d'exploitation (que l'on appellera kernel pour Linux) est chargé d'interfacer les appels reçus des applications et le matériel.

Les systèmes d'exploitations contiennent souvent divers logiciels applicatifs permettant l'utilisation du système par l'utilisateur final, on peut penser un tels système représentés par des couches comme sur la figure suivante.



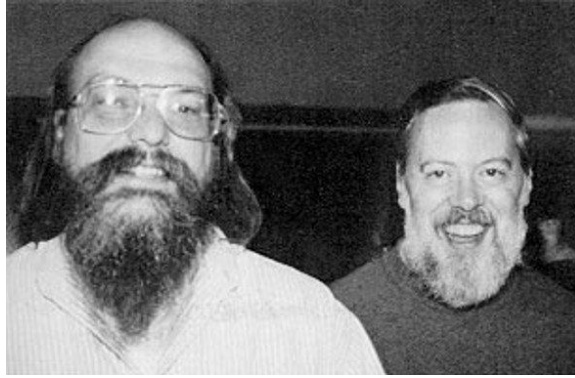
2. Historique

2.1. Unix

2.1.1. Les origines d'Unix

Unix est un système d'exploitation créé en 1969 par Ken Thompson et Dennis Ritchie au sein du centre de recherches "Bell Labs" pour AT&T (opérateur de télécommunications américain). Dans les années 60 AT&T avait commencé à développer un système d'exploitation nommé Multics pour MULTi-plexed Information and Computing Service. De son côté, Ken Thompson développa un système d'exploitation mono-utilisateur et réalisant une seule action à la fois. Brian Kernighan suggéra donc le nom "Unics" en opposition au préfixe multi- de Multics, il fut plus tard contracté en Unix.¹

¹ "Unix" in *Wikipédia*. <https://fr.wikipedia.org/w/index.php?title=Unix&oldid=178193023>.



Ken Thompson (à gauche) et Dennis Ritchie (à droite)

Peu de temps après, le problème de la maintenance apparut, il deviens difficile de maintenir ce système écrit en assembleur. Ken Thompson a l'idée du Fortran qu'il abandonne rapidement, il commence ensuite à le réécrire en langage B, une amélioration du langage BCPL, qui est en fait un précurseur au langage C. Unix en langage B ne sera pas achevé et le langage C sera créé par Dennis Ritchie en 1971. Ce langage a permis la réécriture du noyau en 1972, cela a permis de rendre Unix portable : il suffit "simplement", de compiler le code C pour une certaine plateforme (type d'architecture, type de processeur).²

2.1.2. La philosophie Unix

Dans son nom, Unix implique quelque chose d'unique : Unix est composé de plusieurs programmes permettant de réaliser chacun une tâche unique. Et c'est ce principe qui a guidé tout le développement d'Unix, c'est la base de la "philosophie Unix", ils tentèrent de créer un système d'exploitation petit et simple, "pas seulement un environnement de programmation mais un système autour duquel une communauté peut se former, ce n'est pas juste taper sur un terminal mais aussi encourager la communication proche" a énoncé Dennis Ritchie.

Unix propose de décomposer les tâches de programmation complexes en sous-tâches simples, on peut décomposer Unix en trois couches : le **noyau** qui gère l'interaction entre les programmes et le matériel, le **shell** qui gère la communication entre le système et l'utilisateur et les **utilitaires** qui réalisent des tâches diverses (éditer un fichier, trouver des mots, trier des nombres...). Ces utilitaires simples s'interfaçent entre eux grâce aux "pipes" du shell, ces tuyaux redirigent la sortie d'un programme vers un autre programme, un fichier, ou bien l'affichent à l'utilisateur.³

Enfin, on peut résumer la philosophie Unix à 2 des 4 principes qu'a documenté Doug McIlroy :⁴

- Faites en sorte que chaque programme fasse une chose et bien. Faire un programme par tâche plutôt que de compliquer d'autres programmes
- Pensez que la sortie de chaque programme peut être l'entrée d'un autre. Seulement du texte sans information supplémentaire, éviter les formats binaires

Cette citation de Dennis Ritchie résume plutôt bien Unix : *UNIX is very simple, it just needs a genius to understand its simplicity*. Cette citation peut se traduire en français par "Unix est très simple, il a juste besoin d'un génie pour comprendre sa simplicité.

² Bellevue Linux Users Group, "Ken Thompson: developed UNIX at Bell Labs" in *The Linux Information Project* (14 janvier 2006). <http://www.linfo.org/thompson.html>.

³ "Ken Thompson and Dennis Ritchie Explain UNIX (Bell Labs)" (en), Bell Labs (1980). <https://www.youtube.com/watch?v=JoVQTPbD6UY>.

⁴ McIlroy, Doug, "Bell System Technical Journal" (en), pp. 1902-1093, AT&T USA (1978). https://ia802701.us.archive.org/12/items/bstj57-6-1899/bstj57-6-1899_text.pdf.

⁶ “POSIX” in *Wikipédia* (2020-12-01T20:20:08Z). <https://fr.wikipedia.org/w/index.php?title=POSIX&oldid=177173779>.

2.2. Linux

2.3. Introduction générale

Commençons par donner un bref aperçu de Linux avant de rentrer dans ses détails. Linux dans son terme premier est un noyau de système d'exploitation créée par le finlandais Linus Torvalds en 1991, il était étudiant à l'école d'Helsinki quand il a commencé le développement du noyau. Le système d'exploitation que l'on connaît aujourd'hui sous le nom de Linux est aussi dénommé "GNU/Linux", ce terme indique qu'il s'agit du noyau Linux ayant en plus les utilitaires GNU (compilateurs, shell...), cela compose un système d'exploitation utilisable.

Pour des soucis de simplicité, dans ce document j'utilise le terme "Linux" pour qualifier le système d'exploitation GNU/Linux. Le terme "Linux" viens de la contraction de "Linus' Unix" ("Unix de Linus" en français).



Linus Torvalds, créateur de Linux, "Aalto Talk", 2012.

Cette photo plutôt polémique est tirée d'un enregistrement d'une conférence lorsqu'un personne lui demande un avis le fait que NVidia ne propose aucune aide ou collaboration avec Linux. Linus affirme que c'est la pire marque avec qui il a travaillé, il s'exclame donc "NVidia f*ck you!" par ras-le-bol.

2.4. Les premiers pas de Linux

Début des années 90, les microprocesseurs Intel 80386 32 bits sont répandus sur le marché des ordinateurs personnels (l'architecture de ces processeurs est aussi connue sous le nom compatible PC ou bien x86). Les systèmes et les programmes à cette époque, ne profitaient pas pleinement des 32 bits et utilisaient des couches de compatibilité 16 bits. C'est donc en 1991 que notre Linus, travaillant sur Minix (système d'exploitation micro-noyau similaire à Unix, cf. diagramme ci dessus), s'amuse à créer un système d'exploitation fonctionnant sur x86 et implémentant les normes POSIX (cf. 2.1.3 La suite d'Unix).⁷ Au début Linus n'avait aucune intention pour Linux mis à part de s'amuser et apprendre le fonctionnement d'un système d'exploitation : Minix était plus avancé. Linux publie librement le code source de son projet afin que n'importe qui puisse consulter et apporter sa pierre à l'édifice.

⁷ Naissance de Linux - Wikisource. https://fr.wikisource.org/wiki/fr:Naissance_de_Linux?oldid=2143806.

2.5. GNU, la montée

Parallèlement, Richard Stallman, leader et créateur de la communauté open-source mène le projet GNU ("GNU's Not Unix", "GNU n'est pas Unix" en français) qui a pour but de créer un système d'exploitation entièrement libre (cf. 4.2 Le rôle de l'open source). Commencé en 1990, GNU a pour but de créer un système de type Unix (respectant les normes POSIX) librement distribuable. GNU possède déjà des utilitaires open-source de remplacement tels qu'un compilateur C (GCC), un shell (gnu-bash, version libre du "bourne again shell")... Mais il manque une pièce maîtresse pour composer un système d'exploitation, un noyau, GNU possède un noyau en développement : Hurd. Malheureusement après plusieurs années de développement, le noyau est plutôt médiocre, le projet est inachevé à ce moment.

Linux étant distribué sous la même licence (GPL, cf. 4.1.1 GPL : GNU Public Licence), il est facile du point de vue légal d'obtenir un système composant les 2, en effet, le noyau Linux est venu combler le trou de Hurd pour former le système Linux (GNU/Linux) que l'on connaît aujourd'hui. Ce système, au fil des années suscite l'engouement de la communauté qui ne cesse de grandir.

3. Linux

3.1. Le noyau, les plateformes

Afin de comprendre l'idée du fonctionnement du noyau Linux, il faut d'abord comprendre les deux familles plus ou moins différents de noyaux de systèmes d'exploitation, en effet, même si dans un système d'exploitation le noyau remplit globalement toujours la même fonction, il peut le faire de différentes manières.⁸ On peut donc différencier 4 types principaux de noyaux.

3.1.1. Les noyaux monolithiques

C'est le format le plus simple et le plus ancien, il apparaît sur d'anciennes versions de Linux, Unix, BSD. Le noyau monolithique répond simplement au besoin d'un noyau : il interface le matériel d'un côté, de l'autre il gère les appels systèmes. Il se compose d'un seul (d'où l'étymologie "mono") bloc de code compilé contenant tout le nécessaire (et parfois plus) pour l'exécution, c'est-à-dire que toutes les fonctions système et l'ensemble des pilotes sont contenus dans le noyau et chargés au démarrage.

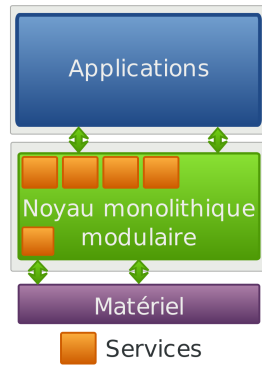
Cela présente l'avantage de la simplicité de mise en place ainsi qu'une vitesse d'exécution excellente. Néanmoins, plus le noyau grossit et devient populaire sur différents supports, moins cette méthode devient optimale, en effet, le code est difficilement maintenable car c'est un seul bloc important, de plus s'il supporte du matériel différent, il doit contenir un tas de pilotes. Ces pilotes sont chargés en mémoire, donc la gestion de cette dernière n'est pas optimale. C'est pour cela que ce type de noyau a vite été abandonné au profit des suivants (abandonné après la version 1.2 de Linux).

3.1.2. Les noyaux monolithiques modulaires

C'est une des architectures de noyau qui permet de répondre aux problèmes énoncés précédemment. Ce noyau monolithique est modulaire comme son nom l'emploie, c'est à dire qu'il contient quand même une partie monolithique insécable qui regroupe les fonctions fondamentales dans un seul bloc de code. Tous les autres fonctions noyau telles que les pilotes sont séparées en modules pouvant être chargés individuellement en mémoire.

Tous les défauts du monolithique simple sont donc résolus : le code est composé de plusieurs blocs, le nombre de pilotes disponibles n'affecte pas les performances mémoire, en effet, on peut donc grandement augmenter la portabilité du système. C'est d'ailleurs ce choix que Linus Torvalds a retenu, ce type de noyau est utilisé par Linux (et la plupart des BSD) actuellement, on remarque en effet que Linux est très portable, on peut l'installer sur quasiment n'importe quel matériel. Néanmoins cette solution n'a pas été retenue pour toutes les architectures de noyaux, d'autres solutions sont implémentées.

⁸ "Noyau Linux" in *Wikipédia*. https://fr.wikipedia.org/w/index.php?title=Noyau_Linux&oldid=177848772.



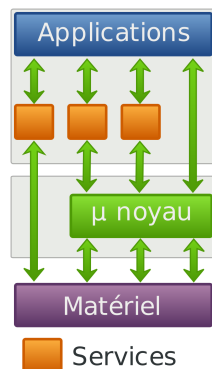
Représentation schématique des noyaux monolithiques modulaires

3.1.3. Les micro-noyaux

Ce type de noyaux est en opposition avec ceux monolithiques, les micro-noyaux ont une part inséparable très petite : la plupart des fonctionnalités du système sont donc des programmes (appelés services) exécutées dans l'espace utilisateur. C'est à dire que ces fonctions systèmes interagissent avec le noyau via des appels systèmes, cela permet de réduire les fonctionnalités dépendantes du noyau ainsi que de réduire la taille et l'occupation mémoire de celui ci.

Les systèmes à micro-noyaux possèdent l'avantage de séparer les services, si l'un d'entre eux est défaillant (problème de fonctionnement, faille de sécurité) il ne peut pas influencer et compromettre le noyau car il est dans un espace d'exécution différent (il ne peut pas interagir avec la mémoire du noyau). De plus, les services peuvent être démarrés indépendamment selon les moments auxquels ils sont nécessaires, ainsi seuls les services utiles sont lancés au démarrage. On peut aussi redémarrer des services ayant plantés.

L'ensemble micro-noyau + services se nome micro-noyau enrichi. Néanmoins ces avantages ont un coût, en effet le système se compose de plusieurs services effectuant continuellement des appels systèmes qui alourdissent la charge du micro-noyau. Ces communications importantes à l'intérieur du micro-noyau ralentissent la vitesse d'exécution. Ces types de noyaux sont utilisés pour Minix, une des sources d'inspiration de Linux, par le noyau GNU Hurd inachevé à l'époque qui souffrait de problème de performance liés à cette dite architecture.



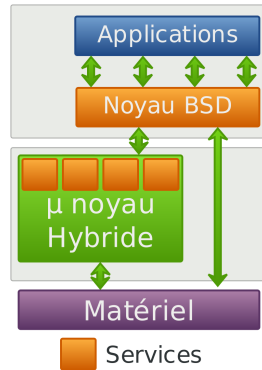
Représentation schématique des micro-noyaux

3.1.4. Les noyaux hybrides

Ces noyaux désignent un type reprenant les deux types précédemment vus : les noyaux monolithiques et les micro-noyaux. Ils tentent de combiner les avantages de ces deux architectures de noyau. Ils permettent de résoudre les faiblesses des micro-noyaux en implémentant un bloc monolithique pour les

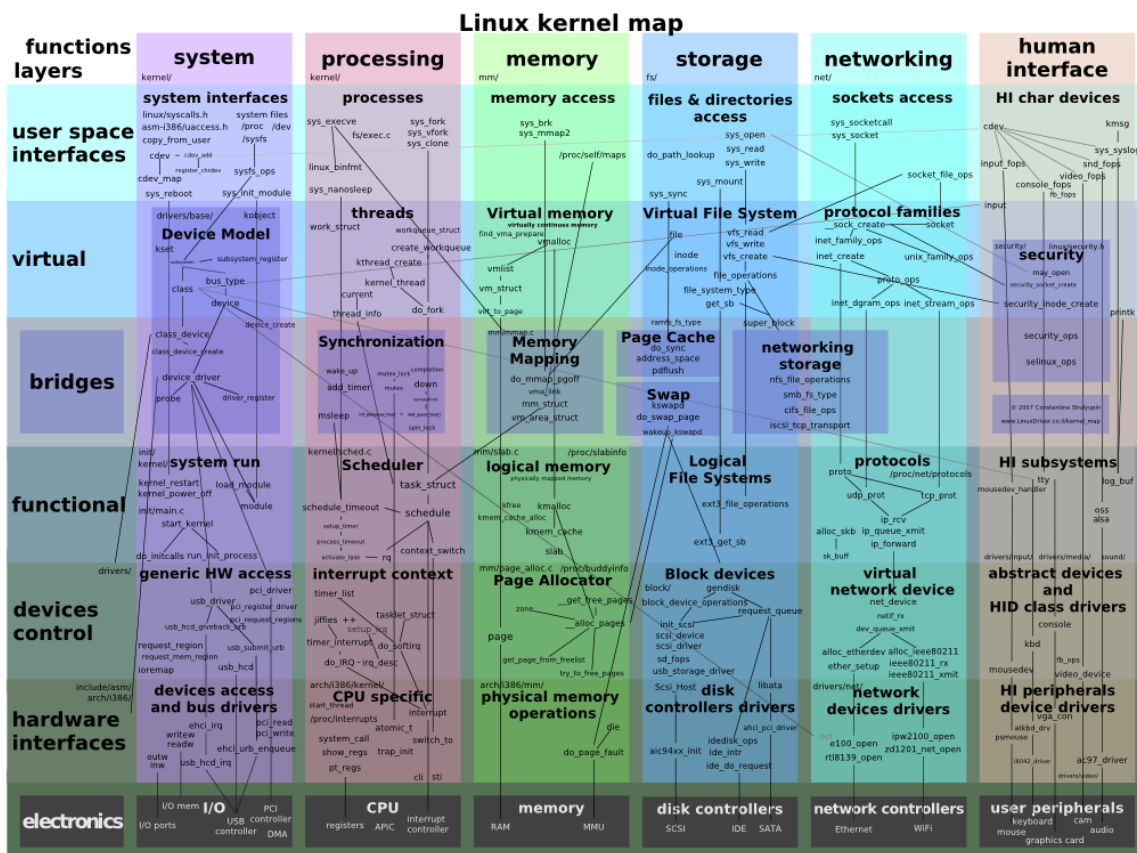
fonctions fondamentales du système d'exploitation.

Ce compromis d'architecture permet de résoudre les problèmes de performance existants avec les micro-noyaux. C'est ce type de noyau retenu par Apple pour son noyau XNU utilisé notamment pour Mac OS X, il contient un micro-noyau Mach (autrefois aussi utilisé par GNU Hurd) et une partie monolithique BSD. Ce type est aussi celui Windows NT, noyau qui fait suite à DOS (qui lui était monolithique).



Représentation schématique du noyau XNU utilisé par Apple

Au final, ces noyaux offrent différentes solutions au même problème en posant chacun leur lots de d'avantages et inconvénients.



Cette représentation schématique impressionnante montre les différents composants du noyau Linux, comment ils interagissent entre eux.

En bas se trouve le matériel, on voit toutes les couches d'abstraction que procure le noyau.

3.2. Les distributions

3.2.1. L'origine des distributions

À l'origine, l'installation d'un système Linux utilisable requérait des connaissances poussées en informatique car il fallait composer son système en trouvant manuellement les logiciels un par un. Dès 1992, les premiers ensembles de logiciels donnant un système complet sont apparus, ce sont les **distributions**. On peut noter Debian et Slackware sorties toutes deux en 1993, cette première est encore connue aujourd'hui et a donné naissance à une très grande quantité de distributions actuelles (Ubuntu et ses dérivés pour ne citer que la plus connue), cette deuxième a aussi donné quelques noms remarquables encore existants aujourd'hui (SUSE et OpenSUSE pour ne citer qu'eux).

3.2.2. Les composants d'une distribution

Aujourd'hui, une distribution Linux contient plusieurs composants plus ou moins essentiels qui permettent l'utilisation d'un système, ces composants peuvent varier d'une distribution à l'autre. En consultant le livre "Linux From Scratch" on peut trouver une liste recommandée de paquets permettant de créer sa propre distribution Linux depuis des sources.⁹ La liste est longue et détaillée car elle permet de créer son propre système, néanmoins, nous pouvons quand même lister les composants principaux :

- Le noyau Linux, composante essentielle, probablement le plus important.
- Un amorçeur de démarrage, le plus utilisé est GRUB, il permet de charger le système d'exploitation. Il est lancé par le BIOS depuis la partition d'amorçage du disque. Ce programme permet aussi à l'utilisateur de choisir quel système il souhaite lancer si plusieurs sont présents.
- Un système d'initialisation, est le premier processus lancé sur la machine par le noyau. Chaque processus lancé ensuite est un enfant de ce processus parent. Il existe plusieurs systèmes d'initialisation :
 - sysVinit, le système originellement présent sur Unix System V, un répertoire contient des scripts qui indiquent à "init" quels processus démarrer. Ce programme est léger et concis, il respecte la philosophie Unix.
 - systemd, pour System Daemon, est le système d'initialisation le plus utilisé sous Linux depuis 2015 et encore aujourd'hui il est présent sur une majorité de distributions. Il propose beaucoup de fonctionnalité et rends plus accessible la gestion des services en ajoutant une couche d'abstraction. C'est une grosse suite logicielle complète effectuant une tonne de tâches, une "usine à gaz", ce qui n'est pas en accord avec la philosophie Unix "faire un logiciel à fonction unique, interfaçable".
 - Trop lourd à comprendre et modifier il fait débat au cœur de la communauté. Il existe donc des alternatives modernes à systemd telles que OpenRC, runit ou r6 qui sont plus légères, il sont implémentés par exemple dans Artix Linux.
- Des programmes tels que GCC (GNU Compiler Collection), Glibc, binutils, automake, make... Ils servent à compiler, linker, interfacier avec le système les différents programmes que vous pourrez compiler et exécuter.
- Des utilitaires simples pouvant être des dépendances pour d'autres programmes tels que grep (pour rechercher des termes dans des fichiers), gzip et tar (pour compresser et décompresser des archives) ou bien perl, awk ou python qui permettent d'exécuter des scripts.
- Un shell, c'est-à-dire un interpréteur de commandes, le plus utilisé est bash mais il en existe d'autres similaires avec plus ou moins les mêmes fonctions (zsh, fish, dash...). Il permet à l'utilisateur de communiquer avec le système par une interface en ligne de commande (CLI pour "command line interface")

⁹ Beekmans, Gerard, Editor, Managing, and Dubbs, Bruce, "Linux From Scratch - Version 10.0-systemd" (en), p. 316.

en anglais) ou bien de réaliser des scripts pour effectuer des actions plus complexes

- Un gestionnaire de paquets : présent sur toutes les distributions modernes, il permet de récupérer des paquets ou des mises à jour (source ou binaires déjà compilés) depuis internet. Chaque famille de distribution en a un différent. Il facilite la récupération de programmes.
- Éventuellement pour un ordinateur de bureau, notre distribution va posséder un serveur X tel que X.org, il s'occupe de l'affichage graphique. Il peut être couplé à un environnement de bureau (GNOME, KDE, LXDE, Cinnamon...) ou un gestionnaire de fenêtres (dwm, i3, bspwm...) qui eux vont gérer l'affichage et l'interaction des fenêtres... Chacun de ces programmes va proposer un niveau différent de possibilités de réglages et personnalisations graphiques.

Et encore tout ceci ne suffit pas à obtenir un Linux fonctionnel, mais cela donne l'idée que Linux est le résultat de l'assemblage d'une ribambelle de programmes fonctionnant en harmonie. J'ai omis les programmes utilisateurs qui sont au choix des goûts de chacun comme un éditeur de texte (même si Vim est le meilleur), un navigateur web, un explorateur de fichiers...

3.2.3. Les familles de distributions

Il existe aujourd'hui des centaines de distributions Linux différentes, il serait impossible et il est inutile de toutes les lister, car cette liste deviendrait rapidement obsolète, et beaucoup se ressemblent. Il est néanmoins intéressant de comparer les différentes familles de distributions selon des critères tels que la popularité, la facilité d'installation, la quantité de logiciels présents avec...

- La famille Debian, un des plus vieux systèmes Linux, il a donné une grosse famille de distributions. Il est aujourd'hui connu pour être stable, lui et ses enfants occupent une part importante des machines sous Linux, sa communauté est probablement la plus grande. Il a donné naissance à Ubuntu, probablement la distribution la plus populaire de ces dernières années, cette popularité fait que ces systèmes sont disponibles sous plusieurs formes avec des environnements graphiques et des logiciels pré-installés différents.

Les distributions de cette famille utilisent le gestionnaire de paquets apt, il existe des versions dites LTS (Long Term Service) qui offrent une stabilité sur plusieurs années, ces distributions sont donc appréciées sur les serveurs.

- Arch Linux est une distribution destinée aux utilisateurs avancés (ils adorent répéter qu'ils utilisent Arch Linux par fierté d'avoir réussi à l'installer). Elle se veut rester simple, elle est livrée sans rien : l'utilisateur doit deviner comment partitionner son disque, régler son système, installer un environnement graphique... Elle est très formatrice pour l'utilisateur motivé qui voudrait "composer" sa distribution à son souhait. Slackware (une des premières distributions) était aussi dans ce style : elle s'adresse à l'utilisateur avancé.

Arch Linux possède pacman, un gestionnaire de paquets qui permet d'installer des paquets binaires pré-compilés, ou de compiler un paquet depuis ses sources. De plus les utilisateurs peuvent proposer librement des paquets sur l'AUR (Arch User Repository), ce dépôt de logiciel contient des programmes publiés par les utilisateurs de Arch, n'importe qui peut les installer. De plus les mises à jour sont effectuées en publication continue ("Rolling release") : il n'y a pas de version stable spécifique, soit le système est à jour soit il est dépassé. Ce cycle continu peut avoir un coût en stabilité, les utilisateurs ont l'habitude de réparer leur distribution après certaines mises à jour.

Arch a donné naissance à Manjaro, une version simple à utiliser et installer pour l'utilisateur lambda, Manjaro est disponible avec plusieurs environnements de bureau, c'est cette distribution que j'utilise depuis que j'ai choisi de passer de Windows à Linux.

- Pour continuer sur les distributions d'utilisateurs habitants à l'abri de la lumière du soleil, il existe Gentoo, une distribution à compiler soit même. Cette distribution se compile depuis ses sources, ses logiciels aussi doivent être compilés. Cette version demande un utilisateur expérimenté mais sa complexité d'installation possède un avantage : les logiciels sont compilés spécifiquement pour l'ordinateur souhaité donc les performances ne peuvent qu'être meilleures. Bien sûr, comme les précédentes elle n'est fournie qu'avec le minimum, c'est à l'utilisateur d'installer son interface graphique.

Étonnamment, Chrome OS, le système d'exploitation de google est basé sur Gentoo, bien que l'utilisateur final obtient une expérience finale bien différente du Gentoo de base.

- Les distributions dites "professionnelles" : certaines distributions offrent des services adressés aux entreprises, les distributions sont open-source mais l'entreprise offre une version binaire compilé ainsi qu'un support, qui peut être un gain de temps pour une entreprise, donc de potentielles économies. On peut citer "Red Hat Linux" de l'entreprise Red Hat, cette version est vendue aux professionnels. Red Hat distribue aussi des versions communautés comme CentOS (destiné aux serveurs) ou Fedora, une distribution avec un environnement de bureau.

Dernièrement Red Hat à "arrêté" CentOS en distribuant les mises à jours de manière continues au profit de versions stables, ce qui casse l'intérêt pour les serveurs qui ont besoin d'être stables. Le but étant d'attirer les utilisateurs vers l'achat de Red Hat Linux, mais cela ne fonctionne pas trop, en effet un groupe d'utilisateurs travaille au développement de "nouveaux CentOS", on peut citer les projets "Alma" ou "Rocky Linux". Comme quoi il ne faut pas sous estimer la puissance de la communauté open-source.

Finalement, il existe des distributions pour chaque utilisation avec ses avantages et ses inconvénients, je n'ai testé qu'une part limité de distributions et je ne suis pas capable de donner plus de précisions sur les cas d'utilisation. Il existe néanmoins des témoignages détaillés en ligne sur les différentes expériences des utilisateurs et leurs avis sur chacune des distributions.¹⁰

3.3. Les utilisations de Linux

Par sa versatilité et son nombre de modifications infinies, Linux est utilisé sur un variété de supports que ce soit des distributions GNU/Linux ou juste le noyau utilisé dans d'autres buts.

3.3.1. Bureau

Comme énoncé précédemment Linux possède un nombre et une variété importante de distributions de bureau pouvant correspondre à n'importe qui. Il existe des centaines de distributions de bureau, un nombre important environnements de bureau ou de gestionnaires de fenêtres, possiblement un infinité de programmes installables dessus.

Il existe aussi plusieurs alternatives aux logiciels propriétaires (et potentiellement payants) disponibles sur la plateforme Linux. Néanmoins, aujourd'hui Linux occupe qu'une très faible partie des ordinateurs de bureau dans le monde, cela est probablement dû au fait que très peu de personnes s'intéressent au système d'exploitation tant qu'il fonctionne. En effet Windows étant le système le plus vendu sur les ordinateurs de bureau, les utilisateurs conservent celui ci.

Pour ma part, j'ai testé plusieurs versions d'Ubuntu puis Manjaro récemment, cette dernière est celle que j'utilise au quotidien depuis plusieurs mois à la place de Windows.

3.3.2. Serveur

Le domaine des serveurs est une autre paire de manches, ici le pingouin est empereur (Tux, l'emblème de Linux est le dessin d'un manchot papou). En effet, Linux est le système le plus utilisé dans le domaine des serveurs : il est robuste, consomme peu de mémoire, gratuit, modifiable, sécurisable car transparent sur son fonctionnement. Sur un serveur on préfère utiliser une distribution avec un cycle de mises à jours lentes et possédant des versions "service long terme" (c'est à dire de mise à jour et des correctifs longtemps après la parution de la version. Les distributions les plus utilisées sur des serveurs sont Debian, Ubuntu et CentOS.

Par exemple pour la plupart des sites web auxquels vous accédez tournent sur une machine Linux. Même Microsoft propose Linux pour ses services de location de serveurs dans le cloud.

¹⁰ Nicolas Kovacs (kikinovak), "Comment j'ai choisi mon système d'exploitation" in *MicroLinux*.
<https://blog.microlinux.fr/choix-os/>.

On peut aussi noter que c'est le système d'exploitation de choix des super-calculateurs, il tourne sur 100% du top 500 des super-ordinateurs.¹¹

3.3.3. Embarqué

Le noyau Linux est disponible sur beaucoup de plateforme : x86_64 par exemple pour les ordinateurs (bureau ou serveur), mais aussi pour des architectures ARM, l'architecture des microprocesseurs de téléphones portables. En effet Android présent sur plus de la moitié des téléphones cellulaires est basé sur un noyau Linux. L'utilisation de Linux sur un raspberryPi est aussi une application de Linux sur du matériel embarqué : cet ordinateur miniature tenant dans la pomme de la main est très apprécié des bricoleurs amateurs en électronique.

Des versions de Linux sont utilisé dans certains robots, machines outils ou autres applications industrielles diverses, ils utilisent la version du temps réel du noyau qui permet de garantir le temps d'exécution des tâches.

4. Les logiciels libres

4.1. Les différentes licences de distribution

Quand on parle de logiciels libre, on se réfère à un logiciel "open-source", c'est-à-dire que son code source est ouvert, consultable par n'importe qui. Ces logiciels et leur code source sont distribués sous certaines licences décrivant les droits exerçables par les auteurs et les utilisateurs. Ces licences logicielles possède chacune des caractéristiques qui les rendent plus ou moins attirantes pour les utilisateurs et les entreprise. Ici il est traité des deux familles de licences open-source les plus utilisées.

4.1.1. GPL : GNU Public License ¹²

C'est la licence sous laquelle est distribué Linux, cette licence viens de la communauté GNU qui milite pour la gratuité des systèmes informatique. Cette licence peut se résumer en quelques principes :

- Le logiciel peut être utilisé à n'importe quelle fin, les utilisateurs son libres d'en disposer comme bon leur semble.
- N'importe qui peut modifier le programme comme il le souhaite, il n'y a pas de restriction, tout le monde peut l'étendre à sa volonté.
- Le logiciel (modifié ou non) peut être redistribué librement à n'importe qui qui le souhaite, de manière commerciale ou non.
- Le logiciel restera libre, c'est-à-dire que n'importe quelle distribution du logiciel il doit y avoir son code source distribué également, cela évite la privatisation du programme.

Cette licence a permit le développement collaboratif de Linux sans qu'aucune entreprise ne se l'attribue afin d'en distribuer des licence commerciales, cela a aussi permit à Linux et sa communauté de se développer, chacun a pu contribuer au projet en proposant sa modification. On parle aussi de "copyleft" dans le cadre de cette licence, un terme dérivé du "copyright" qui montre une opposition à ce dernier.

4.1.2. Licences permissives : BSD/MIT

Les licence BSD et MIT sont aussi des licences open-source, elles permettent à tous ceux qui le souhaitent de distribuer du code source. Néanmoins ce type de licences est dit permissif, cela signifie qu'on peut distribuer un logiciel sous cette licence sans avoir l'obligation d'en distribuer son code source. Les logiciels sous cette licence sont plus appréciés des entreprises car elles n'ont pas l'obligation de

¹¹ Prakash, Abhishek, "Linux Runs all of the Top 500 Fastest Supercomputers" in <https://itsfoss.com/linux-runs-top-supercomputers/>.

¹² Communauté GNU, "GNU Public License v3" (fr) (2007). <https://www.gnu.org/licenses/quick-guide-gplv3.html>.

redistribuer les modifications apportées à ce logiciel.

On peut, pour n'en citer qu'une, l'histoire de Andrew Tanenbaum, le créateur de Minix a distribué son logiciel sous licence BSD. Intel a silencieusement pris son logiciel (grâce à sa licence) pour l'utiliser pour "Intel Management Engine", un micro système tournant sous tous les ordinateurs ayant un processeur Intel.

Cela a été révélé à Andrew seulement plusieurs années plus tard par des articles de journaux. Il est partagé entre un sentiment de dégoût, de regret et un sentiment de fierté : ¹³*Je suppose que ça fait de Minix le système d'exploitation le plus utilisé au monde, encore plus que Windows, Linux, ou Mac Os.* Il ne s'attendait pas à ça même s'il aurait aimé de la reconnaissance, ou au moins une information de la part de Intel. Mais c'est la licence BSD, elle n'oblige ni la redistribution du code, ni le crédit de l'auteur.¹⁴ On peut donc penser que cette histoire est tragique, mais il est fortement possible que si la licence aurait été GPL, Intel n'aurait pas utilisé Minix car il aurait fallu aussi distribuer le code source modifié qui serait "offert" à la concurrence.

Pour résumer sur les licences open-sources, il y en a deux types répandus, l'une **GPL** est restrictive dans le sens où elle oblige à redistribuer le code source des logiciels, elles ne sont pas forcément appréciées des entreprises. Les autres, telles que **BSD / MIT** sont dites "permissives", les utilisateurs distribuant des versions modifiées ne sont pas obligés de distribuer le code source, plus apprécié des entreprises qui peuvent monétiser ou vendre ces services sans avoir l'obligation d'en dévoiler les sources.

4.2. Le rôle de l'open source

Linux est de loin le projet open-source le plus ambitieux de l'histoire, il regroupe des milliers de développeurs à travers le monde. Ce projet serait impossible à refaire pour une seule entreprise ou organisation, il est maintenu grâce à plusieurs contributeurs et associations de contributeurs. Certains sont bénévoles mais la majorité sont des entreprises, on peut noter Intel, IBM, Red Hat...

Pour réaliser ce projet de grande ampleur, il a fallu trouver un moyen de collaborer, là Linus Torvalds, en bon développeur Linux, s'est enfermé dans sa cave à l'abri de la lumière et a inventé "git". Le logiciel de versionnage, conçu originellement pour Linux, permet de gérer des versions de code source, différentes branches, différents collaborateurs... Il est encore utilisé aujourd'hui par de nombreuses entreprises et particuliers, il possède certaines surcouches logicielles facilitant l'utilisation et le partage de codes sources comme : GitHub, GitLab (pour l'hébergement en ligne), GitKraken (client local pour la visualisation de dépôts git) ou d'autres clients ou extensions pour des logiciels de développement...

Un autre avantage indéniable de l'open-source pour les logiciels est que tout le monde peut consulter et vérifier le code source, il est donc impossible de placer des portes dérobées, des dispositifs d'espionnage, de traçage d'utilisateurs, de récupération de données personnelles...

5. Conclusion

5.1. Conclusion générale

Pour donner une conclusion, on peut admettre que Unix et ensuite Linux sont des œuvres majeures de ces 50 derniers siècles qui ont fait progresser l'informatique sur un plan mondial. Une part non négligeable des systèmes informatiques dans le monde utilisent Linux ou un autre système d'exploitation "Unix-like".

Son développement open-source permet qu'il soit toujours maintenu à jour par sa communauté, de plus cet aspect offre une liberté et une transparence logicielle que certaines personnes et organisations recherchent de plus en plus aujourd'hui. De plus les systèmes libres (GNU/Linux) pour poste utilisateur deviennent de plus en plus accessibles, ils sont parfaitement utilisables par le grand public, voire même par certaines entreprises. Aujourd'hui, un non initié peut utiliser un ordinateur de bureau sous Linux facilement.

¹³ Andrew, Tanenbaum, "An Open Letter to Intel" (en). <https://www.cs.vu.nl/~ast/intel/>.

¹⁴ Smith, Luke, "Why I use GPL and not Cuck Licenses" (en). <https://lukesmith.xyz/articles/cucklicenses>.

5.2. Note personnelle

Sur un plan personnel, je cotoie Linux depuis plusieurs années sur ordinateurs de bureau, raspberry pi, sur serveur depuis 1 an et depuis 3 mois comme système d'exploitation principal pour mon ordinateur personnel. Et Linux correspond à mes besoins qui sont plutôt divers (programmation, navigation web, jeu, impression 3D...), l'adaptation à Linux sur un ordinateur de bureau est très rapide et permet de se débarrasser de logiciels et système propriétaires.

Aussi cette veille a été mise en page grâce à groff, l'adaptation par GNU de roff/troff, un langage de formatage de texte par balises, il est plus léger que LaTeX.¹⁵ C'est un programme Unix existant depuis 1971, il est typique de la philosophie Unix : petit, léger et simple chaque fonction avancée est fournie par un autre programme nommé préprocesseur. Ces programmes sont interfacés par des pipes (symbole "|") entre chaque commande puis la sortie du dernier dans un fichier pdf. On peut citer : "refer" pour les notes bibliographiques, "pic" pour les images, "eqn" pour les équations... Ce programme est aussi utilisé pour mettre en forme les pages manuelles ("man") sur les systèmes de type Unix.

5.3. Plan de diffusion

Cette veille s'adresse aux personnes traitant de près ou de loin l'informatique, les informations peuvent être comprises avec un minimum de connaissances de l'informatique. Elle peut s'adresser aux membres d'une entreprise et être présentée sous forme orale lors de réunions. Ce document est aussi consultable sur mon site web personnel.

¹⁵ Corderoy, Ralph, "troff.org - the Text Processor for Typesetters" (en). <https://www.troff.org/>.

"

Table des Matières

1. Introduction	2
2. Historique	2
2.1. Unix	2
2.1.1. Les origines d'Unix	2
2.1.2. La philosophie Unix	3
2.1.3. La suite d'Unix	4
2.2. Linux	5
2.3. Introduction générale	5
2.4. Les premiers pas de Linux	5
2.5. GNU, la montée	6
3. Linux	6
3.1. Le noyau, les plateformes	6
3.1.1. Les noyaux monolithiques	6
3.1.2. Les noyaux monolithiques modulaires	6
3.1.3. Les micro-noyaux	7
3.1.4. Les noyaux hybrides	7
3.2. Les distributions	9
3.2.1. L'origine des distributions	9
3.2.2. Les composants d'une distribution	9
3.2.3. Les familles de distributions	10
3.3. Les utilisations de Linux	11
3.3.1. Bureau	11
3.3.2. Serveur	11
3.3.3. Embarqué	12
4. Les logiciels libres	12
4.1. Les différentes licences de distribution	12
4.1.1. GPL : GNU Public License ¹²	12
4.1.2. Licences permissives : BSD/MIT	12
4.2. Le rôle de l'open source	13
5. Conclusion	13
5.1. Conclusion générale	13
5.2. Note personnelle	14
5.3. Plan de diffusion	14