

<http://www.coincoin.fr.eu.org/?Genese-de-Linux>



Génèse de Linux

- 3- Logiciels Libres - Historique et analyses -

Date de mise en ligne : mercredi 18 janvier 2006

Copyright © L'Imp'Rock Scénette (by @_daffyduke_) - Tous droits réservés

HISTOIRE DE LINUX

L'histoire d'Unix est liée étroitement à celle des grandes entreprises et l'utilisateur moyen, tel que vous ou moi n'a pas voix au chapitre. Mais nombreux sont ces utilisateurs, principalement dans les universités, qui ont envie d'en savoir plus et qui aimeraient pénétrer un peu au coeur du système, trop profondément au goût des sociétés qui se sont appropriées le produit.

Andrew S. Tanenbaum de la Vrije Universiteit d'Amsterdam aux Pays-Bas, développa, avec ses élèves, le système d'exploitation Minix pour PC qui devait s'articuler autour de la même logique de base qu'Unix. Grâce à ce modèle simplifié, les étudiants devaient apprendre le fonctionnement d'un système d'exploitation. La première version de Minix sortit en 1987 et remporta un franc succès en quelques mois.

Minix fut proposé sans licence à des réseaux de distribution non commerciaux et c'est ainsi que naquit l'idée d'un système Unix accessible sans droit de licence. Et déjà, des passionnés consacraient tous leurs loisirs pour transposer Minix à d'autres plates-formes matérielles (par exemple Atari ST, Commodore, Amiga, Macintosh, etc.).

Linus Torvalds met le feu aux poudres

C'est sur cette idée que s'appuie l'étudiant finlandais Linus Torvalds pour ses travaux à l'université de Helsinki. Partant de Minix, il édifie les murs porteurs de des versions adaptées ce qui deviendra Linux tel que nous le connaissons aujourd'hui. De nombreux amis et bénévoles l'ont aidé dans sa tâche en mettant leurs travaux à sa disposition pour ce projet.

La collaboration avec le projet GNU de la Free Software Foundation de l'université de Cambridge, Massachussets, permit notamment d'étendre considérablement l'éventail des fonctions et programmes. La règle de base qui a prévalu tout le temps voulait qu'il n'y ait jamais de droits de licence pour aucun code de programme. De fait, toutes les parties du système d'exploitation Unix ont ainsi été réécrites et améliorées en pas mal de points.

A la fin de l'année 1991, Linus Torvalds présente la première version opérationnelle de Linux. Il lui donna le numéro de version 0.99 pour bien montrer que quelques améliorations étaient encore nécessaires pour que l'on puisse parler d'un produit fini. Il était important, à ce moment, comme pendant toutes les phases précédentes de l'histoire d'Unix, d'obtenir un maximum de remontées de la part de la communauté des utilisateurs de Linux. Les erreurs furent ainsi vite débusquées et corrigées. Des modifications du produit furent cependant nécessaires à des intervalles rapprochés et le système d'exploitation ainsi que les utilitaires devaient ainsi être révisés quasiment chaque mois à l'aide de programmes de mise à jour que l'on nomme patches.

Il était important, pendant ce temps, que tout le monde puisse profiter de Linux de la même manière. Personne ne devait gagner de l'argent sur ce produit au titre d'un quelconque droit de propriété et personne ne devait être exclu des nouveaux développements. C'est pourquoi Linux est distribué sous les conditions du GNU Public License (GPL). Chaque utilisateur a le droit d'en modifier le code source, de l'améliorer ou de corriger d'éventuelles erreurs. Mais il n'a pas le droit d'utiliser Linux à des fins commerciales et ses programmes doivent toujours être mis gratuitement à la disposition de la communauté Linux.

Premières versions majeures de Linux.

A la fin de l'année 1993, on en était ainsi à la version 0.99 avec le patch level 14 [en abrégé : 0.99pl 14). Au début de l'année 1994, la version 1.0 était disponible et l'on pouvait dire alors qu'il s'agissait bien d'un système complet et abouti. A l'heure actuelle, on en est à la version 2.3.12 et ce numéro de version sera probablement à la lecture de cette page, un nouveau patch sortant presque toutes les semaines.

Toujours est-il que Linux est devenu un système d'exploitation stable et il vaut la peine que l'on s'y intéresse. Il n'existe pas seulement sur PC. On trouve également des versions adaptées pour d'autres ordinateurs tels que Hewlett Packard ou Sun. D'autres sociétés rassemblent les logiciels Linux existants dans un but commercial, les rabaissent et les polissent à divers endroits puis proposent le produit ainsi obtenu comme étant un nouveau système d'exploitation. C'est par exemple ainsi qu'est apparu le système d'exploitation Caldera.

Aujourd'hui, de nombreuses entreprises mettent en oeuvre pour des applications professionnelles la version la plus récente de Linux. Ainsi, beaucoup de serveurs de données et de gestionnaires de domaine reposent sur Linux. Cela prouve la confiance qui lui est accordée.

STANDARDS RESPECTÉS PAR LINUX

Connaissant l'histoire de Linux, l'on ne s'étonnera pas d'apprendre que le système ne fait pas la part belle aux standards existants. En fait, la plupart des programmeurs qui ont contribué à son développement n'avaient que peu d'intérêt ou de respect pour les normes officielles. Ils ont cependant adapté leur produit aux standards chaque fois que quelques personnes jugeaient que c'était nécessaire. Au bout du compte, Linux respecte tout de même un certain nombre de spécifications officielles.

L'interface iBCS2

C'est ainsi que Linux gère par exemple l'interface permettant d'exécuter des programmes binaires écrits pour d'autres systèmes Unix. Ces programmes doivent (voir plus haut) respecter l'Intel Binary Compatibility Standard (iBCS2) et le pilote iBCS2 doit être chargé sous Linux. On peut alors exécuter sous Linux des applications qui ont été écrites pour Unix SCO et compilées sous ce système.

il est, par exemple, possible de faire fonctionner Oracle sous Linux. Il convient de noter que cette base de données est fortement dépendante du système d'exploitation concerné, et qu'elle est très exigeante, sur le plan du système d'exploitation mais aussi sur celui du matériel.

Le format de fichier ELF

Par ailleurs, le format interne des programmes exécutables est construit suivant la définition ELF (External Link Format). Le passage à ce nouveau format, qui a été développé à l'origine pour Unix System V Release 4, n'est cependant pas encore implémenté dans toutes les versions Linux. Mais à l'avenir, ce sera certainement le format standard d'après lequel seront structurés les fichiers binaires et les bibliothèques.

Linux et les autres systèmes d'exploitation

Linux peut aussi fonctionner avec d'autres systèmes d'exploitation en usage sur les PC et être intégré dans un réseau existant. Ceci s'applique aussi bien pour les réseaux purement Unix que pour les réseaux locaux comportant

des ordinateurs sous MS-DOS et/ou Windows. Il est même possible d'utiliser un ordinateur Linux comme serveur pour des machines MS-DOS.

Sur un PC, Linux peut cohabiter avec MS-DOS, Windows x.xx / 9x / NT ou OS/2. Linux sait lire et écrire les fichiers des autres systèmes d'exploitation, à condition cependant qu'il s'agisse de fichiers texte et de données. En ce qui concerne l'appel de programmes, Linux a réalisé d'importants progrès mais il reste encore quelques restrictions dans ce domaine. Il est possible, dans une certaine mesure de lancer sous Linux des programmes DOS (avec l'émulateur DOS) ou Windows (avec WinE)

Linux, X/OPEN et POSIX

La prudence est cependant de mise si l'on veut appliquer les spécifications X/OPEN. On risque d'être déçu dans bien des cas. Seules les définitions POSIX entrent en ligne de compte si l'on veut appliquer des normes officielles. L'interface de programmation de Linux est entièrement compatible avec la norme POSIX. Cela constitue un petit progrès mais il faut néanmoins rester très prudent lorsqu'il s'agit de porter simplement sur d'autres systèmes Unix des programmes développés sous Linux. Là aussi, la déception sera au rendez-vous dans la plupart des cas.

La plupart des commandes et programmes de Linux permettent de spécifier des options et des paramètres comme dans les autres systèmes Unix. En plus, il est également possible d'indiquer ces options et paramètres avec la méthode conforme à POSIX. Pour de nombreuses commandes Unix, on finira assez rapidement par mélanger complètement les différentes abréviations d'options, tout simplement parce qu'elles varient énormément d'une commande à l'autre.

Dans ce cas, la méthode POSIX de spécification d'options constitue une aide. On a moins de difficultés à mémoriser les options car elles sont composées de mots entiers et non plus d'abréviations.

Compilateurs et autres outils de développement

Le compilateur C (le GNU C) supporte le C ANSI et il constitue également une bonne base pour obtenir des programmes portables. Mais les fonctionnalités décrites par les spécifications ANSI sont souvent insuffisantes pour créer des programmes pouvant être transférés sans trop de complications d'un système à un autre.

Le compilateur C++ (également développé par le GNU), bien qu'il ne respecte pas toutes les définitions de la norme ANSI actuelle, constitue une bonne initiation en matière de programmation orientée objet. D'autres langages de programmation

sont également disponibles, mais qui ne respectent pas les normes internationales (Smalltalk, Lisp, Objective C, etc.).

Linux et les autres systèmes Unix

De nombreuses commandes de Linux et d'autres systèmes Unix présentent de très fortes ressemblances et les racines communes sont ici évidentes. Les commandes *ls* et *who* existent aussi sous Linux, par exemple. Mais de nombreuses options sont parfois ajoutées et les possibilités des commandes s'en sont également trouvées modifiées. Les manuels Unix que l'on trouve dans le commerce constituent en général une initiation de premier niveau satisfaisante mais qui ne saurait prétendre exploiter les particularités de Linux.

Génèse de Linux

Sa compatibilité avec les réseaux permet de raccorder Linux avec d'autres systèmes UNIX. Les données circulent ainsi de manière transparente entre Linux et les systèmes UNIX, autrement dit sans que l'utilisateur ne s'en aperçoive.