

# Développer sous Linux

## Comment faire ?

## Comment choisir ?\*

---

*Pour assurer la réussite d'un développement applicatif sous Linux, il est important de bien cibler les objectifs. Ceux-ci conditionneront à la fois les outils et les langages de programmation à utiliser.*

Le choix d'effectuer le développement d'un logiciel en langage C ou C++ sous Linux recouvre en réalité deux prises de décision. En premier lieu, on fait le choix d'un système d'exploitation. L'utilisation de Linux comme environnement de développement est une option nouvelle dans le milieu industriel. La seconde décision concerne le langage et les bibliothèques employées. D'autres langages sont en effet disponibles, et il faut être en adéquation avec le but final de l'application ainsi que le public visé. Nous allons examiner ici les possibilités qu'offrent les outils disponibles sous Linux pour une bonne réalisation logicielle en langage C ou C++.

### Développement sous Linux

Nous avons indiqué que le choix de la plateforme Linux pour effectuer un développement logiciel est déjà une décision importante. Plusieurs motivations peuvent être à l'origine de ce choix. Tout d'abord, Linux constitue probablement le meilleur système d'exploitation pour la diffusion de logiciels libres. L'esprit qui anime la communauté Linux est tel qu'un nouveau projet est généralement bien accueilli, et l'enthousiasme des participants permet la progression rapide d'un logiciel, pour peu que l'équipe soit suffisamment motivée.

Ce type de choix n'est plus seulement le fait d'amateurs ou d'étudiants, mais certaines entreprises ont déjà compris l'intérêt immédiat que présente pour elles l'adhésion à la logique

des logiciels libres. C'est le cas, par exemple, de certains fabricants de périphériques pour ordinateurs : ils diffusent les spécifications techniques de leur matériel et fournissent une assistance pour le développement de drivers libres permettant le fonctionnement sous Linux.

Le choix du système Linux comme environnement de développement peut aussi être motivé par la qualité des outils présents « librement » et la richesse de la documentation disponible sur Internet. Dès son installation, toute distribution Gnu / Linux comprend tous les outils logiciels nécessaires au programmeur – éditeurs, compilateurs, débogueurs – et ce pour l'ensemble des langages informatiques couramment employés.

De plus le portage des applications vers d'autres environnements Unix est souvent très rapide, puisqu'il ne nécessite qu'une simple recompilation des fichiers source. Il est alors possible de disposer de stations de développement de type PC, puissantes et économiques. Notons également que Linux, loin d'être limité aux PC, est disponible sur un large éventail de machines. Les applications tournant sous ce système disposent ainsi automatiquement d'une portabilité « interne » sur une gamme complète d'architectures et de processeurs différents.

Cependant, la motivation la plus fréquente pour le développement sous Linux, est tout simplement le choix de ce système comme plate-forme applicative finale. Cette solution est depuis plusieurs années retenue dans de nombreux laboratoires, centres d'études, etc. L'utilisation de Linux comme environnement

---

\* Cet article est paru dans le numéro 189 de « L'Informatique Professionnelle », au mois de Décembre 2000.

principal dans le domaine des applications industrielles est plus récente, mais une fois les réticences initiales vaincues, on assiste généralement au sein d'une entreprise à une progression lente mais continue du parc de machines fonctionnant sous Linux.

Pour assurer la réussite d'un développement applicatif sous Linux, il est important de bien cibler dans quelle catégorie s'inscrit le produit : logiciel libre diffusé sur Internet ou réservé à un usage interne ; application destinée à être largement portée sur d'autres systèmes d'exploitation ou au contraire limitée à une utilisation uniquement sous Linux. Ceci influe à la fois sur les outils et langages de programmation à employer, et sur les buts prioritaires que l'équipe de développeurs devra se fixer.

## Choix du langage

Dans le monde industriel, l'un des points forts les plus appréciés du système Linux est sa stabilité remarquable. Les stations configurées correctement peuvent présenter des temps de fonctionnement ininterrompu très longs. Les seuls arrêts du système sont alors dus aux composants matériels, et pas au logiciel. D'où des taux de disponibilité du système particulièrement élevés, ce qui est très précieux dans certains types d'application.

L'exemple le plus fréquemment cité est celui du serveur web. Le logiciel libre Apache est très utilisé pour constituer des serveurs performants et peu coûteux sous Linux. Pour configurer et personnaliser un serveur réseau, on emploie souvent le langage Perl, qui permet d'écrire rapidement de petits scripts puissants, pour dynamiser le contenu des pages web, programmer des robots logiciels gérant des listes de diffusion ou encore trier et répartir le courrier ou les articles Usenet.

Pour des applications nécessitant une implication technique plus importante, les langages les plus utilisés sont le C et le C++. La rapidité et la puissance des applications produites constituent des arguments de choix. En contrepartie, la portabilité est un peu plus restreinte, essentiellement en ce qui concerne l'interface utilisateur. En fait, le noyau Linux, ainsi que tous les utilitaires système, sont écrits en C. L'emploi de ce langage permet donc d'accéder facilement aux ressources de bas niveau indispensables dans certaines applications (interfaçage, écriture de pilote de périphérique, etc.) Le langage C offre des avantages techniques en termes de puissance et

de rapidité convenant tout particulièrement aux applications scientifiques et industrielles. De plus, en tant que langage compilé, il fournit un fichier exécutable autonome, ce qui assure la confidentialité des algorithmes utilisés dans les sources. Cet aspect constitue, dans certaines circonstances, un avantage réel sur les langages interprétés.

L'environnement graphique utilisé sous Linux, comme pour les autres Unix, est le système X-Window. Ce système très puissant est particulièrement intéressant sur des machines en réseau, car il permet de déporter des affichages sur des stations distantes. X-Window n'impose que très peu d'obligations en terme d'aspect ou de comportement des applications. C'est pourquoi de nombreuses bibliothèques graphiques différentes coexistent, compliquant parfois quelque peu le portage des applications.

Sous Linux, deux environnements graphiques principaux existent actuellement : le système Gnome, issu du projet Gnu, et le système KDE. L'un comme l'autre sont installés automatiquement avec la plupart des distributions actuelles, et c'est l'utilisateur qui choisit son environnement favori lors de la connexion. Ces environnements jouent sur l'aspect et le comportement des fenêtres (menu système, bordures, réaction à la souris, etc.), mais ont aussi un rôle important dans la communication entre les applications.

Choisir d'utiliser l'environnement Gnome ou KDE pour l'interface utilisateur d'une application n'est donc pas une action innocente. D'autant qu'il faut alors être conscient que la portabilité de l'application hors de l'univers Linux est largement diminuée. Une autre possibilité est d'employer une bibliothèque portable sur l'ensemble des systèmes Unix. L'environnement Motif s'est imposé comme standard de fait dans les systèmes informatiques industriels. Il en existe plusieurs implémentations commerciales sous Linux, ainsi qu'une implémentation libre nommée Lesstif. L'utilisation de ces bibliothèques permet d'assurer une certaine portabilité des applications graphiques, tout en conservant l'efficacité et les performances d'un langage compilé proche de la machine.

## Outils de développement

Les outils de développement disponibles librement sous Linux sont étonnamment puissants. Au cœur de la programmation en langage C ou C++ se trouve naturellement le compilateur. Celui que l'on emploie sous Linux

se nomme GCC. Il s'agit d'un compilateur produit par le projet Gnu, dont le but est de fournir librement des logiciels de remplacement pour tous les utilitaires système et outils de développement sous Unix. Le compilateur GCC n'a rien de spécifique à Linux, il est disponible sur de nombreux autres Unix, et un portage a même été réalisé sous Dos. La qualité de GCC est telle que certains le préfèrent aux compilateurs natifs vendus sur les stations Unix classiques. Il représente également une alternative intéressante aux compilateurs très coûteux, puisqu'il permet d'équiper facilement et à moindre frais toutes les stations des développeurs d'une équipe. GCC se décline en plusieurs versions installées conjointement, selon que l'on préfère utiliser le langage C, le C++, voire le dialecte Objective C. Les très nombreuses options disponibles pour ajuster la configuration du compilateur permettent d'adapter son comportement aux désirs du programmeur, notamment en ce qui concerne les respects des normes les plus courantes (Posix, Iso C99...)

Le compilateur GCC est un outil fonctionnant en ligne de commande, qu'on peut donc lancer directement « à la main » en indiquant les options désirées et les fichiers sources à compiler. Cet utilitaire n'effectue qu'une seule tâche, mais il la réalise au mieux, sans que des considérations d'interface utilisateur ne viennent le distraire de son rôle. Pour améliorer l'ergonomie du programmeur, il existe toutefois un certain nombre d'environnements de développement intégré, qui regroupent un éditeur de texte pour la saisie des fichiers source, et des composants graphiques (boutons, menus...) pour invoquer directement le compilateur en arrière-plan, en s'assurant de l'emploi des options correctes. Ces environnements permettent aussi l'appel du débogueur Gnu, nommé GDB. À l'instar du compilateur, cet outil fonctionne par une invocation depuis une ligne de commande, puis il offre une interface utilisateur minimale. Les environnements de développement intégré l'encadrent et présentent une interface graphique plus attrayante, en permettant l'exécution d'un programme instruction par instruction, l'inspection des variables, etc.

Le projet Gnu regroupe une panoplie complète d'outils destinés à assister le travail du développeur. Citons par exemple les utilitaires d'archivage permettant de créer des bibliothèques de fonctions personnalisées, les enjoliveurs de code source pour uniformiser l'aspect des fichiers créés par les différents membres d'une équipe, les outils RCS ou CVS qui assurent le contrôle de version et permettent

ainsi le travail simultané par plusieurs développeurs sur le même projet en minimisant les risques de modifications concurrentielles.

## Documentation et exemples

La richesse de la documentation disponible concernant l'environnement Linux fait partie des caractéristiques de ce système qui sont les plus attrayantes pour les programmeurs. On peut trouver sur Internet des sites regroupant des informations de grande qualité dans tous les domaines intéressant le développeur.

En ce qui concerne l'environnement Linux en général, l'installation et la configuration des applications ou l'administration du système on consultera les documents HOW-TO, traduits en français, et qui regroupent des informations sur la plupart des sujets importants pour l'utilisateur. Pour vérifier la syntaxe d'appel d'une routine de la bibliothèque C, pour connaître la signification d'un appel-système (c'est à dire un service offert par le noyau Linux lui-même) ou encore pour étudier le détail de toutes les options proposées par une application, il existe une commande MAN dont l'invocation est très commode. Elle donne accès à des fichiers indépendants, que l'on nomme habituellement « pages de manuel » par analogie avec les classeurs de documentation livrés avec les Unix traditionnels. Chaque page (qui peut représenter parfois une dizaine de pages imprimées) contient la documentation pour une routine de la bibliothèque C, un appel-système ou un utilitaire. Il existe environ mille pages de manuel pour Linux actuellement traduites en français.

Le programmeur pourra toujours consulter à son gré les sources du noyau ou celles de la bibliothèque C, ce qui représente non seulement une source d'information exacte en ce qui concerne une fonctionnalité précise, mais également des exemples performants et souvent élégants, d'implémentation d'algorithmes<sup>1</sup>.

Les fonctionnalités offertes par le noyau Linux et les routines de plus haut-niveau proposées par la bibliothèque C standard ne sont que rarement suffisantes pour la réalisation d'une application

---

<sup>1</sup> Ceci est d'ailleurs vrai pour l'ensemble des applications disponibles sous Linux, la plupart des distributions fournissant d'emblée plusieurs centaines de mégaoctets de fichiers sources immédiatement consultables.

complète. Nous avons déjà abordé la question de l'interface graphique généralement indispensable. Mais il est souvent nécessaire de faire appel à d'autres bibliothèques réalisant des tâches diverses. Citons par exemple l'existence de bibliothèques intégrant des algorithmes de compression, de cryptage, de lecture de fichiers graphiques dans la plupart des formats usuels. Citons aussi la possibilité d'accéder aux objets disponibles sur un bus Corba ou encore d'utiliser des passerelles vers d'autres langages de développement comme Python, Perl ou Tcl/Tk. Toutes ces bibliothèques sont disponibles sous forme de code source librement consultable et adaptable à des besoins spécifiques.

## Conclusion

Pour conclure cette présentation rapide, j'aimerais insister sur l'aspect intellectuellement motivant que représente, pour un programmeur, la possibilité de travailler dans un environnement où il est possible d'avoir accès au fonctionnement interne du système dans ses moindres détails, où rien ne reste dissimulé à qui désire obtenir des informations, et où la qualité des outils et l'assistance disponible par Internet font que l'on ne reste jamais bloqué face à un dysfonctionnement d'un utilitaire système.

**Christophe Blaess** <[ccb@club-internet.fr](mailto:ccb@club-internet.fr)>

## Pour aller plus loin

« *Programmation système en C sous Linux* » Christophe Blaess, éditions Eyrolles.

Le site de l'auteur, et les pages de manuels Linux traduites en français : <http://perso.club-internet.fr/ccb>

Liens de documentation française sur Linux : <http://www.linux-center.org/fr>

Ensemble de liens internationaux sur Linux : <http://www.linux.org/>

Le projet Gnu : <http://www.gnu.org/home.fr.html>