

Architectures des systèmes UNIX, Linux et *BSD

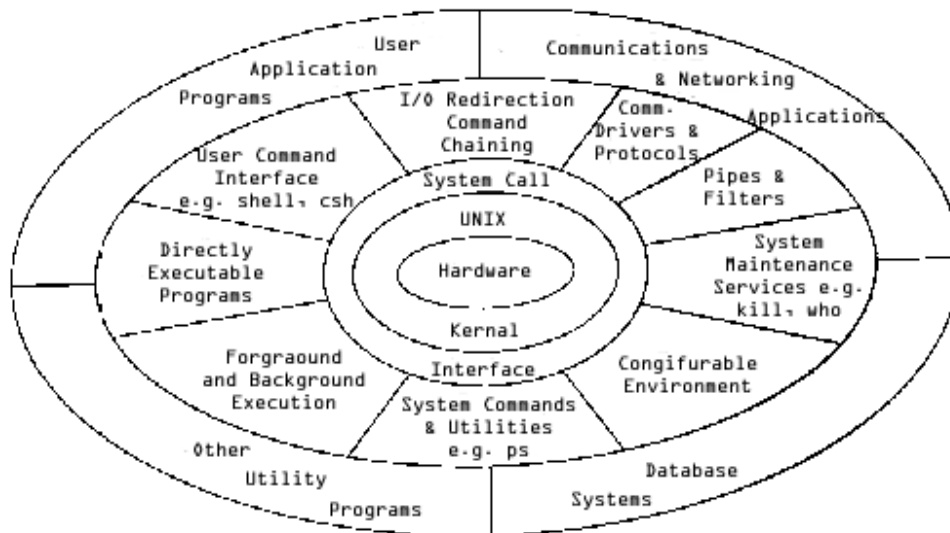
Principes de base qui ont guidé son développement (UNIX)

- Tout est fichier.
- Gestion des processus.
- Un programme doit faire au mieux ce pour quoi il est écrit.
- Tout ce qui n'est pas explicitement autorisé est interdit.
- Redirection des entrées sorties.
- Les sockets ont été ajoutés plus tard pour TCP/IP.

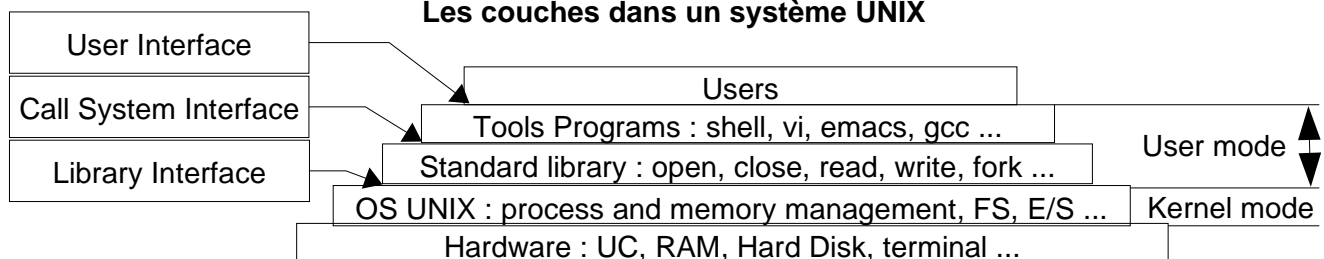
Les concepts fondamentaux d'UNIX (*)

- + Système multiutilisateurs : plusieurs utilisateurs peuvent être actifs en même temps
- + Système multitâches : exécute plusieurs programmes en même temps
- + Permet la répartition des ressources (mémoire, processeurs, espace disque, imprimantes, programmes et utilitaires) entre les utilisateurs et les tâches.
- + Chaque utilisateur peut exécuter plusieurs programmes simultanément.
- + Fournit des primitives pour construire des applications complexes à partir d'autres plus simples
- + Il est possible de rediriger les entrées et sorties des processus
- + Un mécanisme de communication par tubes permet de synchroniser des processus et de leur faire échanger des informations.
- + Un système UNIX est administré par un *super utilisateur* ou administrateur système.

Architecture du système UNIX



Les couches dans un système UNIX



Le système comprend :

- * un ensemble de fonctions de base appel ;
- * l'enchaînement des tâches ou processus ;
- * un système de gestion de fichiers hiérarchisé ;
- * des interpréteurs de commandes (shell, C-shell, Korn-shell, ...) disposant d'instructions et de structures de contrôle ;
- * des utilitaires : compilateur C, éditeurs, logiciels réseau, générateurs d'analyseurs lexicaux et syntaxiques, ... ;
- * un mécanisme d'interactions entre processus : les signaux ;
- * un système de messagerie électronique.

(*) Note : quelques points négatifs demeurent dans UNIX, tels que les coûts de développement encore élevés. Il reste cantonné dans le monde des mini-ordinateurs jusqu'au milieu des années 90, où la puissance est devenue suffisante sur PC et stations à base de μ RISC.

Différences et évolutions dans Linux et *BSD

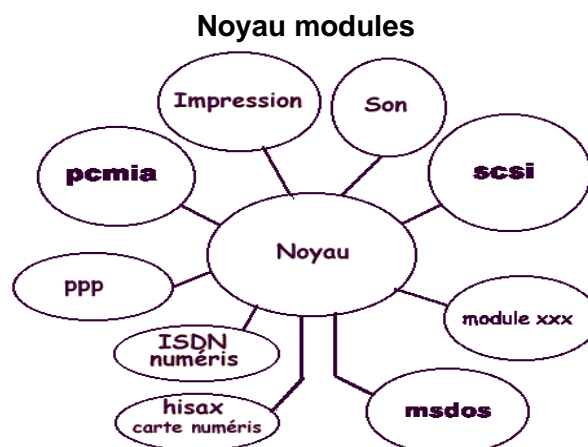
Le noyau Linux :

Architecture d'origine complètement monolithique (noyau en un seul fichier exécutable) :

- obsolète, du fait de l'intégration d'un nombre de plus en plus important de *pilotes* (drivers),
- souffre d'obésité en plus d'être confronté à des limites de maintenabilité et d'évolutivité.

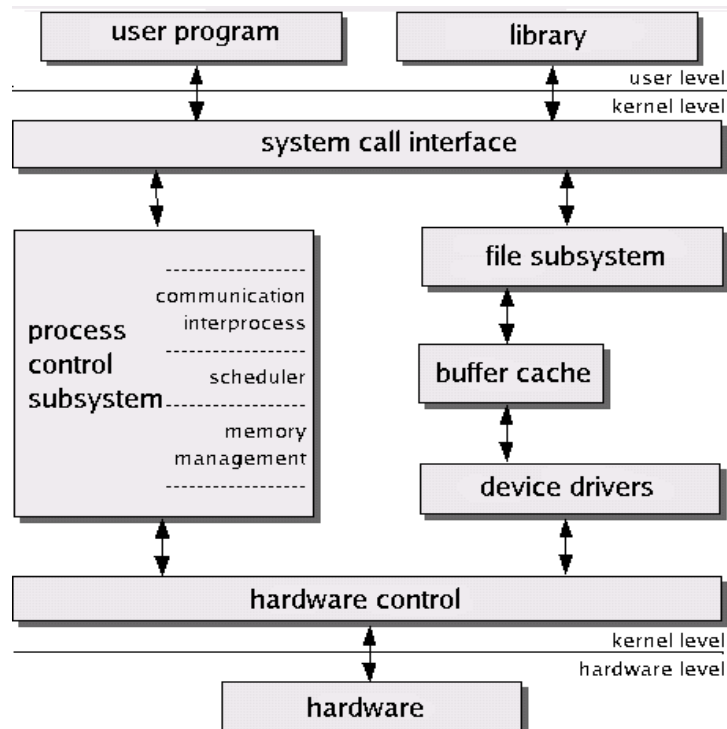
Evolution vers un modèle d'architecture hybride à partir du noyau 2.0 :

- le noyau monolithique/hybride est constitué du strict nécessaire pour démarrer, s'initialiser et a la capacité de charger des modules, il est de type LKM, *Loadable Kernel Module*,
- caractéristiques : interaction directe avec du matériel, encombrement mémoire faible, portabilité importante, fiabilité, souplesse, sécurité limitée.
- multiplate-formes : fonctionne avec différents types de processeurs (Intel, Sparc, Alpha, PPC ...)



Le kernel est toujours constitué d'un seul fichier « noyau » hybride ou modulaire contenant le code pouvant s'exécuter en kernel space et peut-être lié statiquement ou dynamiquement à d'autres fichiers « son, ppp, impression ... ».

Kernel Linux 2.0



Le noyau 2.6 a été conçu pour fonctionner sur des architectures diamétralement opposées :

- optimisé pour les très grosses architectures massivement multiprocesseurs (énorme serveur ultra-puissant et possédant beaucoup de mémoire) à l'électronique embarquée. Il peut ainsi travailler sur une plus *grande échelle de matériel* (scalability).
- préemptif, c'est à dire qu'il peut arrêter une tâche en cours d'exécution pour en exécuter une autre considérée prioritaire.
- interactivité améliorée notamment avec l'*ordonnanceur* (scheduler), la partie qui choisit d'attribuer un temps processeur, à un programme.
- bibliothèque optimisée de gestion de threads POSIX : NPTL, *Native Posix Threads Linux*.
- support de beaucoup plus de périphériques, c'est à dire le *branchement à chaud de périphériques* (hotplug). Le système d'accès aux disques durs permet de meilleures performances.
- support de plusieurs nouveaux systèmes de fichiers, des systèmes de fichier réseau non distribués (comme NFSv4) et distribués (comme Intermezzo).
- support du multimédia en général notamment avec le son : un système de pilotes pour les cartes, plus performants, offrant plus de possibilités : ALSA, *Abstract Layer Sound Architecture*.

Constations et améliorations du noyau 2.6 :

Un nombre croissant d'amélioration du *noyau* (kernel) et son adaption à différentes architectures matérielles permet à Linux d'être intégré dans de nombreux appareils. (ordinateur, calculateur, PDA, téléphone, satellite, robot, etc ...). Parmi les domaines dans lesquels excelle le noyau Linux, il y a incontestablement le réseau et il est un des systèmes d'exploitation le plus interopérable du marché.

Linux 2.6 inclut également un support amélioré pour le domaine relativement nouveau des systèmes de fichiers réseau distribués, système où les fichiers sur un volume logique unique peuvent être découpés sur de multiples noeuds. En plus du système de fichiers CODA introduit dans Linux 2.4, Linux possède maintenant un support pour deux autres systèmes de fichiers distribués : AFS et InterMezzo. AFS, le système de fichiers d'Andrew (ainsi nommé parce qu'il était originellement développé à CMU), est pour l'instant très limité et restreint à des opérations de lecture seule (une version plus complète d'AFS est disponible en dehors du noyau). Le second système de fichiers supporté, InterMezzo (également développé à CMU) est aussi nouvellement supporté sous Linux 2.6 et il possède des caractéristiques plus avancées telles que l'opération de déconnexion (donc vous travaillez sur des fichiers locaux mis en cache) et est adapté à des applications nécessitant une haute disponibilité, où vous avez besoin de garantir que le stockage n'est jamais indisponible (où simulé si absent). Il présente également des applications pour garder des données synchronisées entre plusieurs ordinateurs, tels qu'un portable ou un PDA et un ordinateur de bureau. Plusieurs des projets fournissant un support pour ces nouveaux types de systèmes de fichiers sont initialement développés sur Linux, permettant à Linux d'être très en avance sur le support de ces nouvelles fonctionnalités.

Fonctionnalités diverses et sécurité :

Un autre gros changement dans Linux 2.6 qui n'a pas reçu assez d'attention est la richesse des nouvelles modifications liées à la sécurité. Plus fondamentalement, la totalité de la sécurité basée sur le noyau (la puissance du super-utilisateur sous les systèmes d'exploitation de la famille UNIX) a été modularisée pour n'être plus qu'un module de sécurité parmi une multitude d'alternatives potentielles (cependant le seul modèle de sécurité offert à ce jour est celui par défaut et un exemple pour faire le vôtre). Pour ce changement, toutes les parties du noyau ont été mises à jour afin de baser tous les accès utilisateur sur un système fini de capacités, plutôt que le vieux système superuser. Presque tous les systèmes Linux continueront d'avoir un compte `root` avec un accès complet, mais ceci permet à un système Linux-like d'ignorer cette supposition fondamentale. Un autre changement connexe à la sécurité est que les modules binaires (par exemple, les pilotes vendus par un fabricant de matériel) ne peuvent plus surcharger des appels système avec les leurs et ne peuvent plus ni lire ni modifier la table d'appels système. Ceci limite de manière significative la quantité d'accès que les modules non open source peuvent faire dans le noyau et règle probablement quelques failles légales autour de la GPL. Le dernier changement quelque peu lié à la sécurité est que le nouveau noyau Linux peut maintenant utiliser des générateurs de nombres aléatoires matériels (tels que ceux présents dans quelques nouveaux processeurs), plutôt que de compter sur le pool d'entropie (tout à fait bon il est vrai) basé sur des fluctuations aléatoires de matériel.

Le noyau BSD :

Le noyau BSD a une typologie différente de Linux, et il hérite directement des versions d'UNIX (noyau monolithique). Néanmoins le noyau de la distribution FreeBSD est comparable à celui de Linux et présente des similitudes avec son architecture hybride (noyau modulaire).

Le système BSD, *Berkeley Software Distribution*, également appelé BSD/OS, fut développé à la fin des années 70 par le Computer System Research Group à l'Université américaine de Berkeley. Il se situe comme l'un des descendants de l'UNIX d'AT&T. Une partie de ce BSD/OS est toujours utilisée par les différents systèmes UNIX dont Linux (ex : API socket BSD). L'essor des logiciels libres a en effet permis à divers groupes de développeurs de nouveaux systèmes basés sur BSD.

Ces systèmes sont tous de véritables UNIX, ils sont les descendants directs de BSD 4.4. Ces OS sont connus pour être très stable et très rapide. Tout comme Linux, ils sont disponibles librement.

FreeBSD, le plus répandu :

Disponible pour plate-formes i386 et Alpha, FreeBSD dispose de nombreux atouts, et notamment de performances très impressionnantes lorsqu'il est utilisé comme serveur Web.

Il gère en partie ou totalité certains sites générant beaucoup de trafic, comme Yahoo!, Hotmail et le serveur du site Apache. De plus, FreeBSD supporte bien mieux les lourdes charges (abondances de requêtes) que Linux. Par exemple, un site hébergé par une seule machine a transféré plus de 2 Téra octets en une journée. FreeBSD peut être utilisé comme station de travail grâce aux environnements GNOME et KDE. Il est actuellement le seul des 3 *BSD à supporter le clustering et le SMP. Cette situation devrait cependant bientôt changer car le support de ces fonctions sont actuellement en développement pour NetBSD et OpenBSD.

NetBSD, le plus hétérogène :

Le principal objectif de NetBSD : la portabilité du système. Il existe pas moins d'une trentaine d'architectures supportées par NetBSD des VAX au Alpha en passant par l'Amiga, l'Atari ou le i386 et même les processeur MIPS des PocketPC. Si vous avez des machines "exotiques" ou relativement anciennes, que vous désirez malgré tout faire fonctionner sous un UNIX de type BSD. Si vous avez, dans vos placards, plusieurs stations de travail Sun ou de vieux modèles Silicon Graphics ou encore des Macintosh, alors NetBSD est certainement la meilleure solution pour vous ! Vous pourrez ainsi donner un second souffle à vos vieilles machines.

OpenBSD, les plus sûr des OS libres :

OpenBSD est avant tout axé sur la sécurité et la cryptographie, mais se révèle plus lent que les autres BSD. Respectant les normes de sécurité de la NSA et n'ayant connu aucunes failles depuis 3 ans, OpenBSD s'avère le plus intéressant lorsque l'on souhaite créer un Firewall, ou toute autre application assurant des fonctionnalités critiques. OpenBSD s'appuie sur l'algorithme de cryptage Blowfish ainsi que sur les tunnels OpenSSH et SFTP permettant une administration à distance sûre. Il est en réalité issu d'une scission de l'équipe NetBSD et possède donc des similitude avec celui-ci.

Points communs et différences avec Linux :

Les installations des différents BSD sont toujours efficaces, mais elles sont généralement destinées aux personnes que le mode console ne rebute pas. Nous sommes ici très loin de la convivialité des dernières distributions Linux.

Un utilisateur de n'importe quel UNIX pour se servir immédiatement d'un BSD dans le cadre d'une installation standard, en mode console ou graphique, grâce à la présence du serveur XFree86. Il est tout à fait possible d'utiliser GNOME ou KDE sous BSD. En mode console, le shell est `sh` au lieu du traditionnel `bash` configuré sous Linux. Ce shell est moins convivial car vous ne disposez pas de fonctions comme les flèches de déplacement pour retrouver les dernières commandes, ou la pression sur la touche [Tab] pour compléter les noms. Il est facile d'installer `bash` mais il est fortement conseillé de laisser l'utilisateur `root` avec `sh`.

Comparaisons dans les phases de développements de BSD et Linux :

Les systèmes BSD ont comme avantages, la centralisation des projets de développements, contrairement à GNU/Linux et sa centaine de distributions. Aussi, la sécurité, la facilité d'installation et l'ensemble homogène fourni.

Les utilisateurs du monde Open Source, des UNIX libres, reprochent à GNU/Linux : pas assez de prestataires de services compétents, administration complexe.

Autre noyau (le micro noyau) :

Si le *noyau* (kernel) est toujours un fichier unique, il se limite au code du noyau et n'a pas d'extension du kernel space. L'architecture des *micro-noyaux* (micro-kernels) est indépendante de la plate-forme matérielle et elle est orientée services. C'est à dire qu'il ne fournit ni systèmes ou gestionnaires de fichiers, de répertoires et de processus. L'idée sous-jacente à l'architecture des micro-noyaux est un kernel très léger. L'intégralité du système d'exploitation est conçu pour ne fonctionner qu'avec seul type d'architecture comme une sorte de machine virtuelle. Le matériel étant géré par un ensemble de services minimaux et d'une API, *Application Programming Interface* normalisée. Cette abstraction standard interagit sur un modèle deux tiers ralentissant le système.

Le micro-noyau Mach (prononcé mâar) gère cinq abstractions fondamentales : les tâches, les threads, les ports, les messages, et les objets mémoire. Les tâches et les brins sont l'environnement et les moyens d'exécuter un programme. Les ports et les messages permettent aux brins d'interagir entre eux et avec le noyau.

Un mot sur Mac OS X : ce noyau, basé sur BSD UNIX et sur le noyau Mach 3.0 réunit depuis sa sortie en 1999 des caractéristiques des mondes Apple et UNIX. Darwin est le nom donné par Apple au noyau du système Mac OS X. Le système d'exploitation OPENSTEP développé par NeXT et la suite de NEXTSTEP, qui se basait sur BSD 4.3. Apple s'implique activement dans la communauté, et Darwin est finalement compatible avec la distribution BSD de référence FreeBSD et permet d'utiliser un grand nombre de projets OpenSource. MacOS X allie les performances et la stabilité d'UNIX (protection de l'espace mémoire; à la facilité d'utilisation de MacOS).

Tableau comparatif des noyaux :

Type de noyau	Caractéristiques clés
<p>Noyau monolithique :</p> <p>UNIX</p> <p>*BSD</p> <p>Linux version < 2.0</p>	<ul style="list-style-type: none"> ● Interaction directe avec le matériel ● Code optimisé pour architecture hardware spécifique : performances. ● Encombrement limitant les fonctionnalités. ● Portabilité faible.
<p>Noyau hybride :</p> <p>Linux version >= 2.0</p> <p>FreeBSD</p>	<ul style="list-style-type: none"> ● Interaction directe avec le matériel ● Code optimisé pour architecture hardware spécifique : performances ● Encombrement mémoire faible ● Portabilité importante ● Fiabilité ● Souplesse ● Sécurité limitée
<p>Micro noyau :</p> <p>Amoeba</p> <p>Mach (MACOSX, MKLinux)</p> <p>GNU/Hurd</p>	<ul style="list-style-type: none"> ● Encombrement très faible ● La plupart des services de bas niveau du système d'exploitation ne sont pas des constituants du noyau ● Interaction de ces services avec une abstraction virtuelle normalisée de la machine, via une interface de communication standard ● Portabilité importante ● Sécurité renforcée ● Performances limitées (noyaux généralistes)