Engineer

Linux

Noyau utilisé par la plupart des serveurs

Ce que c’est

Linux est un noyau de système d’exploitation libre de type Unix : le programme qui démarre en premier, pilote le matériel, partage le temps processeur et la mémoire entre tous les autres programmes et décide lequel peut accéder à quel fichier ou port réseau. Dans le langage courant, le mot s’étend à tout le système assemblé autour de ce noyau ; certains écrivent GNU/Linux pour reconnaître l’apport des outils GNU qui l’accompagnent généralement.

Linus Torvalds en a commencé le développement en 1991 comme clone d’Unix écrit de zéro ; un réseau informel de développeurs travaillant sur Internet le développe depuis. Il vise les standards POSIX et Single UNIX Specification : les logiciels écrits pour un autre système de type Unix peuvent donc généralement être construits pour Linux avec peu de changements. Il a commencé sur l’Intel 386 à 32 bits et fonctionne désormais sur une longue liste d’architectures à 32 et 64 bits. Le code est sous GPL version 2 uniquement, avec une note explicite indiquant que cette licence ne couvre pas les programmes qui se contentent d’appeler le noyau. Cette note permet aux logiciels propriétaires de fonctionner sous Linux sans question de licence.

Le noyau seul n’est pas un système auquel se connecter. Un système utilisable exige aussi une bibliothèque C, un shell, un gestionnaire de services, un gestionnaire de paquets et des milliers de programmes. Les projets qui assemblent le tout en système installable s’appellent distributions. Debian, Ubuntu, Red Hat Enterprise Linux et ses reconstructions, SUSE, Fedora et Alpine en sont. Chacune livre sa propre construction du noyau, ses réglages initiaux et son propre engagement sur la durée pendant laquelle elle corrigera les problèmes de sécurité.

Ses concurrents varient selon l'usage. Sur serveur, on le compare à Windows Server et à la famille BSD, FreeBSD et OpenBSD, systèmes complets de type Unix développés d'un seul tenant plutôt qu'assemblés à partir de pièces. Sur les ordinateurs portables, il est en concurrence avec Windows et macOS.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce que Linux ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. L'entretien quotidien des serveurs, de la messagerie, des comptes et des mises à jour, pour que les outils indispensables restent sans histoire.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. 10 serveurs PostgreSQL et MS SQL sur Ubuntu — 10 serveurs de bases de données déployés et maintenus

10 serveurs PostgreSQL et MS SQL sur Ubuntu Contactez-nous →

Pourquoi c'est important

Pour une entreprise exécutant des logiciels sur ses propres serveurs ou dans un cloud, le système d'exploitation est généralement choisi avant même qu'on se pose la question. Serveurs web, bases de données, moteurs de conteneurs et outils de supervision sont construits d'abord pour Linux et y sont testés le plus intensément. La décision qui reste à prendre a une portée plus limitée, et il importe de bien choisir.

Le choix entre Linux et Windows Server dépend du logiciel plutôt que du système. Une entreprise reposant sur Active Directory, IIS ou d'anciennes applications .NET a de vraies raisons de rester sous Windows. Presque tout le reste d'une pile web moderne, de PostgreSQL et Nginx à Kubernetes, suppose Linux ; même Microsoft SQL Server possède une édition Linux.

L'argument économique est simple. Le noyau et les distributions communautaires comme Debian et Ubuntu s'installent gratuitement sur autant de machines que vous le souhaitez, sans décompte de licences à auditer ; Windows Server est vendu sous licence commerciale. Avec Linux, l'argent va aux personnes et, si vous en voulez un, à un abonnement d'assistance auprès d'un fournisseur comme Canonical, Red Hat ou SUSE.

Les conteneurs Linux renforcent encore ce lien. Chaque image regroupe un espace utilisateur Linux dans un fichier, et l'exécuter revient à faire fonctionner des processus ordinaires sur un noyau Linux. Sur Mac ou sur un portable Windows, les outils exécutant ces conteneurs démarrent donc d'abord une petite

machine virtuelle Linux ; l'outil `container` d'Apple exécute chacun d'eux dans sa propre machine virtuelle légère. Choisir des conteneurs Linux fait entrer Linux dans la pile, quel que soit le système des portables des développeurs.

Ce qui distingue réellement deux serveurs Linux est la distribution et sa durée de maintenance. Le projet du noyau publie une nouvelle version principale toutes les neuf à dix semaines et maintient quelques branches à long terme pendant des années : la 6.12, par exemple, devrait recevoir des correctifs jusqu'en décembre 2028. Les distributions ajoutent leur propre maintenance ; leur noyau est pris en charge par elles plutôt que par les développeurs du noyau. Choisissez la distribution selon sa période d'assistance et les compétences déjà présentes dans l'équipe.

Une règle du projet rend les mises à jour moins risquées que ce rythme ne le suggère : un changement qui casse un logiciel fonctionnant auparavant constitue une régression, et les régressions sont corrigées, souvent en annulant le changement. Les programmes construits longtemps auparavant continuent ainsi à fonctionner sur de nouveaux noyaux.

Le prix de ce choix est que « nous utilisons Linux » décrit peu de choses. Deux distributions diffèrent par le gestionnaire de paquets, le pare-feu, la configuration réseau et le rythme des versions : une équipe qui en connaît une doit réapprendre certaines parties de l'autre.

À quoi cela ressemble en pratique

L'exemple le plus proche est un hôte unique de 464 Mo hébergeant une forge et un serveur web, dont l'alarme de manque de mémoire ou d'espace disque est un dispositif d'alerte par absence de signal déclenché par minuterie. Toutes les quinze minutes, avec une variation aléatoire, un script mesure la mémoire disponible et l'espace disque libre, puis contacte un service extérieur seulement si les deux dépassent leurs seuils : le silence est donc l'alerte. Le script avait toujours lu le bon chiffre, la mémoire disponible, mais six noms de variables autour disaient « free ». Le noyau utilise la mémoire inutilisée comme cache : la mémoire libre d'une machine Linux saine est donc proche de zéro. Face à un seuil de dix pour cent, un lecteur aurait vu douze mégaoctets libres sur les 464 Mo, et une « correction » d'un caractère aurait transformé une alarme fonctionnelle en alarme qui se déclenche en permanence. Les noms ont été corrigés. La réception d'un signal confirme que le script s'est exécuté et que ses contrôles des seuils de mémoire disponible et d'espace disque libre ont réussi ; elle ne certifie ni le bon

état général de l'hôte ni que le site sert des pages. La supervision peut révéler des signes avant-coureurs, dont le moment d'apparition dépend toutefois de la charge de travail et des conditions observées.

Dans un rôle antérieur, la fondatrice a construit le premier tableau de bord d'observabilité d'une organisation, qui collectait par SSH les mesures de CPU, RAM, HDD et de température, ainsi que d'autres métriques de serveurs Linux distants, et supervisait les conteneurs Docker de la même façon. Une deuxième version est passée à Grafana et Prometheus. Il s'affichait sur un grand téléviseur au bureau, avec des alertes de dépassement de seuil, et les équipes détectaient et résolvaient les incidents 40 % plus vite.

À l'échelle d'un parc entier, l'accès est devenu un ensemble de règles écrites plutôt que mémorisées. Les serveurs Linux faisaient partie des systèmes régis par 1 000 règles RBAC, aux côtés des environnements de développement, instances applicatives et bases PostgreSQL et MS SQL. Ansible les appliquait de façon cohérente entre environnements. Les développeurs ne pouvaient accéder qu'aux serveurs qui leur étaient attribués. Le résultat livré était un ensemble documenté de règles fondées sur les rôles, avec une méthode reproductible d'application et de revue.

Où les problèmes surviennent

Le tableau de bord affiche une mémoire presque pleine sur un serveur parfaitement sain. Linux considère la mémoire inutilisée comme gaspillée et la remplit de données de fichiers en cache, puis libère cette mémoire dès qu'un programme en a besoin. Le chiffre « free » ne compte que la mémoire ne faisant rien du tout et tend donc vers zéro sur une machine occupée. La réponse à « ce serveur peut-il accepter plus de travail ? » est MemAvailable, l'estimation par le noyau de ce que de nouveaux programmes pourraient utiliser sans recourir au swap. Déclenchez l'alerte sur cette valeur :

```
grep -E '^(MemFree|MemAvailable):' /proc/meminfo
```

Un service disparaît, et l'explication est dans le journal du noyau plutôt que dans le sien. Sans configuration contraire, Linux permet de réserver davantage de mémoire que la machine n'en possède, en pariant qu'une grande part ne sera jamais utilisée. Si le pari échoue et que la mémoire manque réellement, le mécanisme d'arrêt des processus en cas de manque de mémoire choisit un

processus, généralement gourmand en mémoire, et l'arrête. L'application n'a aucune occasion d'écrire ; la trace se trouve dans le journal du noyau, affiché par `journalctl -k`. Donnez à chaque service son plafond mémoire via `MemoryMax=` de systemd ou une limite de conteneur : un programme incontrôlé sera arrêté avant que le noyau doive choisir, et surveillez la mémoire disponible sans attendre que le noyau arrête un processus.

Un rapport de bogue du noyau envoyé en amont reste sans réponse. La machine exécute le noyau de sa distribution, avec ses correctifs propres, alors que les développeurs du noyau ne prennent en charge que leurs versions. `uname -r` affiche la version, et ce qui suit le tiret, comme dans `6.8.0-45-generic`, désigne une construction de distribution. Signalez d'abord le problème à celle-ci ou reproduisez-le sur un noyau actuel de kernel.org avant de le remonter en amont.

Les correctifs de sécurité cessent d'arriver alors que le serveur continue comme avant. Chaque version de distribution et chaque noyau à long terme a une date de fin que la machine n'annonce pas. Les branches à long terme 5.10 et 5.15, par exemple, devraient l'atteindre en décembre 2026. Gardez cette date de fin d'assistance dans le même inventaire que le nom du serveur et prévoyez la mise à jour une version à l'avance plutôt qu'à l'échéance.

Pour aller plus loin

- À propos du noyau Linux — la présentation du projet sur Linux et l'intérêt d'une distribution pour débuter.
- Versions — noyaux principaux, stables et à long terme, avec la fin prévue de chaque branche à long terme.
- Le système de fichiers /proc — les définitions de MemFree, MemAvailable et des autres compteurs mémoire.
- Comptabilisation de la surréservation — les trois modes de surréservation et leurs promesses aux programmes.
- Signaler les régressions — la règle « pas de régressions » et la façon de signaler une rupture pour qu'elle soit reconnue comme telle.
- Règles de licence du noyau Linux — GPL-2.0 uniquement et la note sur les appels système qui ne soumet pas les programmes utilisateurs à cette licence.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le parcours de déploiement, les risques opérationnels et les personnes d'astreinte. Nous pouvons définir un changement avec des vérifications, une procédure de reprise et une responsabilité claire.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce que Linux ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/linux/