Engineer

Linux

大半のサーバーが利用するカーネル

どのようなものか

Linuxは、自由に使えるUnix系のオペレーティングシステムのカーネルです。最初に起動してハードウェアを制御し、ほかのすべてのプログラムにCPU時間とメモリーを分け、どのプログラムが、どのファイルやネットワークポートにアクセスできるかを決めます。日常の言い方では、そのカーネルを中心に組み立てたOS全体まで含みます。通常一緒に使うGNUのツールの貢献を認め、組み合わせをGNU/Linuxと書く人もいます。

Linus Torvaldsが、1991年に、ゼロから書くUnixの互換システムとして始めました。それ以来、インターネットを通じた緩やかな開発者の集まりが作り続けています。POSIXとSingle UNIX Specificationへの準拠を目指しているため、別のUnix系システム用のソフトウェアも、通常は少し変えるだけでビルドできます。32ビットのIntel 386で始まり、今では、多くの32ビットと64ビットのアーキテクチャーで動きます。コードのライセンスはGPLの版2のみで、カーネルを呼ぶだけのプログラムには適用しないと、明示した注記があります。プロプライエタリーソフトウェアを、ライセンスの問題なくLinux上で動かせるのは、そのためです。

カーネルだけでは、ログインできるシステムにはなりません。動くシステムには、Cライブラリー、シェル、サービス管理ツール、パッケージ管理ツール、数千のプログラムも必要です。それらをまとめ、インストールできる形にするプロジェクトを、ディストリビューションと呼びます。Debian、Ubuntu、Red Hat Enterprise Linuxとその再ビルド版、SUSE、Fedora、Alpineなどです。どれも、自身でビルドしたカーネル、既定の設定、いつまでセキュリティーの問題を直すかという約束を持ちます。

競合は、用途で変わります。サーバーではWindows Serverや、BSD系のFreeBSD、OpenBSDと比較します。BSD系は、部品を集めるのでなく、OS全体を一体として開発するUnix系システムです。ノートパソコンでは、Windowsや

macOSと競います。

概念を判断に生かす

「Linuxとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 サーバー、メール、アカウント、更新を日々管理し、誰もが頼る仕組みを平穏に動かし続けます。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 UbuntuでPostgreSQL・MS SQLサーバー10台を運用 — 10 導入し正常に保ったDBサーバー

UbuntuでPostgreSQL・MS SQLサーバー10台を運用 お問い合わせ →

なぜ重要なのか

自社サーバーやクラウドでソフトウェアを動かす企業にとって、OSは、尋ねる前にほぼ決まっています。ウェブサーバー、データベース、コンテナエンジン、監視ツールは、まずLinux向けに作り、最も厳しくそこで試験します。残る判断は、より狭い範囲ですが、適切に決める価値があります。

LinuxかWindows Serverかは、システム自体でなく、ソフトウェアで決まります。Active Directory、IIS、古い.NETアプリケーションを基盤にした事業には、Windowsを使い続ける実際の理由があります。PostgreSQL、Nginxから Kubernetesまで、現代のウェブ構成のほぼすべてはLinuxを前提とし、Microsoft SQL ServerにもLinux版があります。

費用の考え方は明快です。カーネルやDebian、Ubuntuのようなコミュニティーのディストリビューションは、何台に入れても無料で、ライセンス数の監査もありません。Windows Serverは、商用ライセンスで販売します。Linuxにかける費用は、人と、必要ならCanonical、Red Hat、SUSEなどの保守契約に向かいます。

Linuxコンテナは、その結び付きをさらに強めます。各イメージは、Linuxのユーザー空間を一つのファイルにまとめ、その実行は、Linuxカーネル上で普通のプロセスを動かすことです。そのため、MacやWindowsのノートパソコンで、これらのコンテナを動かすツールは、まず小さなLinux仮想マシンを起動します。Apple自身の `container` は、一つずつ、専用の軽量な仮想マシンで動かします。Linuxコンテナを選べば、開発者のノートパソコンが何を動かしていても、構成にLinuxが加わります。

二台のLinuxサーバーを実際に分けるのは、ディストリビューションと保守期間です。カーネルのプロジェクトは、九週から十週ごとに新しいmainline（主開発系列）版を出し、少数のlongterm（長期保守）ブランチを何年も保守します。例えば6.12は、2028年12月まで修正する予定です。ディストリビューションは、その上に独自の保守を加えます。ディストリビューションのカーネルを支えるのは、そのディストリビューションで、カーネルの開発者ではありません。保守期間と、チームがすでに持つ技能で選びます。

カーネルのプロジェクトの一つの規則が、更新のリスクを、その頻度から想像するよりも小さくしています。以前は動いたソフトウェアを壊す変更は、リグレッション、つまり後退とみなし、しばしば変更を戻すことで直します。そのため、昔ビルドしたプログラムも、新しいカーネルで動き続けます。

選択肢の多さには、代償もあります。「Linuxで動かしている」だけでは、ほとんど何も分かりません。二つのディストリビューションでは、パッケージ管理、ファイアウォールのツール、ネットワーク設定、リリースの周期が異なり、一方を知るチームでも、他方では一部を学び直す必要があります。

実際の活用例

最も身近な例は、コードのフォージとウェブサーバーを動かす、一台の464 MBのホストです。メモリーやディスクの枯渇を知らせるのは、タイマーで動くデッドマンズスイッチです。十五分ごとに、時間に揺らぎを加えて、スクリプトが利用可能メモリーと空きディスク容量を測り、両方がしきい値を上回る間だけ外部サービスに通知します。通知がないことが警報です。スクリプトは、最初から正しい利用可能メモリーを読んでいましたが、周囲の六つの変数名は「free」でした。カーネルは使っていないメモリーをキャッシュにするため、正常なLinuxマシンでも、完全に空きのメモリーはゼロに近づきます。十パーセントのしきい値と比べる読者には、464 MBのマシンで空きが十二メガバイトに見え、一文字の「修正」で、正常な警報が、常にしきい値を割るものになり得ました。名前を直しました。信号の受信は、スクリプトが動作し、利用可能メモリーと空きディスク容量が設定した閾値の条件を満たしたことを確認します。ホスト全体の健全性や、サイトが配信できていることを証明するものではありません。監視によって兆候を捉えられますが、その時期は負荷や観測する条件によって異なります。

創業者は、以前の職務で、ある組織の最初のオブザーバビリティーダッシュボードを構築しました。遠隔のLinuxサーバーから、CPU、RAM、HDD、温度などの指標をSSH経由で集め、Dockerコンテナも同じ方法で監視しました。第二版では

GrafanaとPrometheusへ移りました。オフィスの大きなテレビで動かし、しきい値を超えると警報を出しました。チームのインシデント検出と解決は、40%速くなりました。

インフラ全体では、アクセス権が、記憶するものから、記述するものに変わりました。Linuxサーバーは、開発者環境、アプリケーションのインスタンス、PostgreSQLとMS SQLのデータベースとともに、1,000件のRBACルールが管理するシステムの一つでした。Ansibleは、環境をまたいで一貫して適用しました。開発者が接続できるのは、割り当てられたサーバーだけでした。納品したのは、文書化したロールベースのルール群と、それを繰り返し適用し、確認できる方法です。

問題が起きるところ

正常なサーバーで、ダッシュボードがメモリーをほぼ満杯と示します。 Linuxは、使わないメモリーを無駄と考え、ファイルのキャッシュで埋めますが、プログラムが求めれば、すぐにそのメモリーを返します。「free」は、何もしていないメモリーだけなので、稼働するマシンではゼロに近づきます。「このサーバーは仕事を追加できるか」に答えるのは、MemAvailableです。スワップせず、新しいプログラムが使える量を、カーネル自身が推計します。この値で警報を出します。

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

サービスが消え、理由は自身のログでなく、カーネルのログに残ります。 Linuxは、使われない予約が多いだろうという見込みで、既定では、実際の容量を超えたメモリーの予約を許します。その見込みが外れ、実際にメモリーが尽きると、メモリー不足時にプロセスを終了する仕組み（out-of-memory killer）が、通常はメモリーを多く使うプロセスを選んで終了します。アプリケーションには、何かを書く機会がありません。記録は、`journalctl -k` で表示するカーネルのログにあります。systemdの `MemoryMax=` やコンテナの上限を使って、各サービスにメモリーの上限を与えます。暴走したプログラムを、カーネルが選ばなければならなくなる前に止め、終了を待たず、利用可能メモリーを監視します。

カーネルのバグを上流に報告しても、返事が来ません。 動いているのは、ディストリビューション独自のパッチを持つカーネルで、カーネル開発者が支えるのは、自身のリリースだけです。版は `uname -r` で分かり、`6.8.0-45-generic` のよう

に、ハイフン以降の文字が、ディストリビューションのビルドを示します。まずディストリビューションに報告するか、kernel.orgの現行カーネルで再現してから、上流へ持ち込みます。

サーバーは動き続けても、セキュリティー修正が届かなくなります。 各ディストリビューションのリリースにも、各longtermカーネルにも終了日があり、マシン自身は知らせません。例えば、longtermの5.10と5.15は、2026年12月に終了する予定です。保守終了日を、サーバー名と同じ台帳に記録し、期限当日でなく、一リリース前に更新を計画します。

参考資料

- Linuxカーネルについて — カーネルのプロジェクト自身による、Linuxとは何か、初心者がディストリビューションを必要とする理由の短い説明です。
- リリース — mainline（主開発）、stable（安定）、longterm（長期保守）のカーネルと、各longtermブランチの予定終了日です。
- /procファイルシステム — MemFree、MemAvailableなどのメモリーカウンターの定義です。
- オーバーコミットの管理 — 三つのオーバーコミットモードと、各モードがプログラムに約束するものです。
- リグレッションの報告 — 「後退させない」という規則と、問題として扱われる報告の方法です。
- Linuxカーネルのライセンス規則 — GPL-2.0のみという条件と、その上のプログラムを制限しないシステムコールの注記です。

用語集

実際の制約に合わせて選ぶ

デプロイの流れ、運用上のリスク、緊急対応の担当者をお知らせください。検査、復旧手段、明確な責任の所在を伴う変更の範囲を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「Linuxとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

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

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/linux/