Engineer

カーネル

OSの中核となるプログラム

どのようなものか

カーネルは、OSの中心にあり、プロセッサーの全権限で動き、ハードウェアへのアクセスを制御するプログラムです。次に動かすプログラムを決め、メモリーを割り当て、各プログラムのメモリーをほかから分離します。ファイルシステムを通してディスクを読み書きし、ネットワークスタックでパケットを運び、誰がどのファイル、ポート、機器にアクセスできるかを制限します。それ以外は、より少ない権限のユーザー空間で動き、代わりの処理をカーネルに求めます。

その要求が、システムコールです。Linuxのマニュアルは、「アプリケーションとLinuxカーネルの間の基本的なインターフェース」と呼びます。プログラムが直接呼ぶことは、めったにありません。glibcなどのCライブラリーのラッパーを呼び、ラッパーが引数を渡し、カーネルの応答を戻り値とエラーコードに変えます。「Permission denied」（権限がない）でプログラムが止まるとき、その文言は通常、システムコールへのカーネルの応答を、ライブラリーが文字列にしたものです。

カーネルは、内部に何を含めるかで異なります。Linuxは、デバイスドライバー、ファイルシステム、ネットワークスタックをカーネル内に持ち、実行中もモジュールとして追加できます。通常、モノリシックと呼ぶ設計です。この場合、各ドライバーもカーネルの全権限で動きます。マイクロカーネルは、最小限だけを残します。seL4は、物理アドレス空間へのアクセス、割り込み、プロセッサー時間を制御する仕組みを用意し、ドライバー、ファイルシステム、ネットワークスタックは、通常のユーザーモードのプログラムとして動かします。検証済みの64ビットRISC-V版は、約10,000行のコードからなり、C実装が仕様どおり動くことを形式的に証明しています。macOSのカーネルXNUは、Machマイクロカーネルと FreeBSDの一部を組み合わせる、ハイブリッドです。

カーネルの更新は、通常、マシンが新しいカーネルで起動して初めて有効になります。6.12などのカーネルのバージョン番号は、そのカーネルを含むディストリビューションのリリース番号とは別です。

概念を判断に生かす

「カーネルとは？」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

企業にとって、カーネルが決める問いは三つあります。ワークロードを実際にどれだけ分離できるか、セキュリティー修正にどれだけの停止時間がかかるか、マシンの実態を示す情報がどこにあるかです。

第一は、分離です。コンテナーは、カーネルが名前空間と制御グループに置いた、通常のプロセスの集合です。名前空間は、各グループに、プロセスID、ネットワークインターフェース、マウント、利用者について、独自の見え方を与えます。制御グループは、プロセッサー時間、メモリー、ディスク入出力を配分します。ホスト上の全コンテナーは、一つのホストカーネルで動くため、そのカーネルの欠陥は、全コンテナーに同時に関わる共通の土台の欠陥です。独自のライブラリーを持つイメージも、ホストのカーネル機能に依存します。仮想マシンは、自身のカーネルを起動します。一つのチームのサービスなら、コンテナーは妥当な境界です。互いに信頼しない顧客には、テナントごとに独立したカーネルを与えます。

次は、修正です。ディスク上に修正済みのカーネルパッケージがあっても、そのカーネルで起動するまで何も変わらず、サーバーでは、その再起動が代償になります。ライブパッチがあるのは、カーネル文書の言葉でいえば、利用者が「再起動したがらない」ためです。稼働中に、個々の関数の呼び出し先を修正版へ切り替えます。その方法で修正できるのは、カーネルがトレースできる関数だけです。そのため、ライブパッチは、緊急の修正で時間を稼ぐもので、再起動に代わるものではありません。再起動を定期的に行い、手順もリハーサルしておく計画は、再起動が一度も要らないと仮定する計画より、負担が小さくなります。

最後に、カーネルは稼働中の状態を持ちます。ファイアウォールのファイルは規則を記述しますが、パケットの扱いを決めるのは、カーネルのパケットフィルター内の規則です。 `/etc/sysctl.d` のファイルは設定を記述し、実際に有効なのは `/proc/sys` の値です。サービスは、操作がカーネルに届いたかどうかに関係なく、正常と報告することがあります。カーネル自身の状態を読む検査は、設定ファイルのレビューでは合格してしまう不具合を見つけます。

実際の活用例

2026年8月の本番ホストの強化レビューは、SSHの総当たり攻撃対策が機能するかを調べました。どれも設定済みで、すべての報告に出ていたため、あるべき動作を記したファイルではなく、カーネルがパケットに実際に行うことと照合しました。カーネルから見ると、三つとも、ホストを構築して以来、一度も作用していませんでした。SSHをポート1986へ移した後も、jailの遮断は、待ち受けソケットのないポート22を対象にしていました。レート制限は、パケットフィルターには届いていました。ただし、規則は上から評価し、最初の一致で止まるため、上にある広い規則に、制限を調べる前に、すべてのパケットが一致していました。遮断の処理は、パッケージが必須依存ではなく推奨依存とする、パケットフィルターのプログラムをシェル経由で呼んでいました。そのプログラムがないホストでも、jailは起動し、数え、判断した後、カーネルへ規則を渡すべき瞬間に失敗しました。レビューから、二つの原則が得られました。セキュリティー制御の動作を確かめるには、コメントではできない検証を、アサーションによって行うこと、ファイアウォールは、規則の有無ではなく、カーネルが保持する規則の順序で検査することです。

その後、遮断の経路には、設定を適用するたびに走るテストを加えました。実行ごとに、文書用に予約したブロックのアドレスを遮断し、カーネルのパケットフィルターから、生成した規則を読み戻し、成否にかかわらず動く後処理で、遮断を解除します。読み戻すからこそ、テストになります。遮断サービス自身の状態なら、規則を一つも作らない遮断でも、成功と報告します。予約アドレスは、実在するネットワークに属さないため、失敗した実行が遮断を残しても、誰も締め出しません。代償は、毎回数秒の時間と、本番にファイアウォール規則を書き、取り除くことです。

メトリクス基盤を置く余裕のない464 MBホストには、稼働中のマシンの状態をMarkdownで報告する、七つの読み取り専用ロールを用意しました。状態のスナップショットは、カーネル自身のカウンターを二回採取し、プロセッサー使用率を計算します。セキュリティー報告は、実際の遮断規則と並べて、ファイアウォール規則を照合順に表示します。七つのロールは、それぞれ、自身の実行が読ん

だ数値だけを出します。カーネルのカウンターも含み、穴を埋める推測値は使いません。報告から、機能していなかった二つのセキュリティー制御と、誰も宣言していない待ち受けポートが見つかりました。

問題が起きるところ

手で変更したカーネル設定が、再起動で戻ります。 `sysctl -w` は、稼働中のカーネルだけに値を書くため、再起動までしか残りません。逆の見落としもあります。 `/etc/sysctl.d` のファイルは起動時に読むため、変更は、後の再起動まで、気付かれずに待ちます。ファイルに設定を書き、今読み込み、実際の値を読み戻します。

```
echo 'vm.swappiness = 10' | sudo tee /etc/sysctl.d/90-local.conf
sudo sysctl --system
sysctl vm.swappiness
```

負荷が急増しても、CPUグラフが平らです。 `cpu` という `/proc/stat` の行は、起動からの時間をUSER_HZ単位、大半のアーキテクチャーでは百分の一秒単位で数えます。そのため、一回の読み取りは、起動以来の平均を示し、数か月稼働するマシンでは、ほとんど動きません。二回読み、差分から比率を求めます。

```
head -1 /proc/stat; sleep 5; head -1 /proc/stat
```

CPUが処理していた時間は、user、nice、system、irq、softirqの差分の合計で、待機時間は、idleとiowaitの差分の合計です。使用率は、処理時間を、処理時間と待機時間の和で割った値です。マニュアルは、iowaitを信頼できない値としているため、警報ではなく、手掛かりとして扱います。

クラッシュ報告に、汚染状態（taint）のないカーネルで再現するよう求める返答が来ます。 カーネルは、後の調査に関係し得ることが起きると、汚染状態を示すtaintedという印を付けます。たとえば、プロプライエタリーなモジュール、ツリー外のモジュール、署名なしのモジュールを読み込んだ場合です。文書は、開発者が、taintのあるカーネルのバグ報告を、しばしば無視すると警告しています。モジュールを外しても、印は再起動まで残ります。報告を出す前に読みます。

```
cat /proc/sys/kernel/tainted
```

ゼロならtaintなしで、ほかの数値はフラグの和です。たとえば、`1` はプロプライエタリーなモジュール、`4096` はツリー外のモジュール、`8192` は署名なしのモジュールを表します。フラグを立てたモジュールなしで起動し直し、不具合を再現します。

新しいネットワークカードが、機器一覧にあっても、インターフェースがありません。 ドライバーはカーネルの一部で、ハードウェアより古いカーネルには、対応するものがないことがあります。`lspci -k` は、各機器で使用中のドライバーと、扱えるモジュールを表示します。driver-in-useの行がない機器には、ドライバーが結び付いていません。ベンダーのモジュールを手でビルドするより、ディストリビューションの新しいカーネル系列を導入します。ツリー外のモジュールは、新しいカーネルごとに再ビルドが必要で、読み込むとカーネルにtaintが付きます。

参考資料

- Linuxカーネルの利用者・管理者ガイド — Linuxを運用するための、カーネルプロジェクト自身のマニュアルです。
- syscalls(2) — 各Linuxシステムコールと、それが導入されたカーネルのバージョンです。
- namespaces(7) — 名前空間の種類と、それらからコンテナーを作る方法です。
- proc_stat(5) — iowaitの警告を含む、`/proc/stat` の各フィールドです。
- 汚染状態（taint）のカーネル — 各taintフラグと、数値の読み解き方です。
- seL4 FAQ — マイクロカーネルが含めないものと、seL4の証明が前提とする条件です。

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

「カーネルとは？」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/kernel/