Engineer

macOS

Appleのデスクトップ用OS

どのようなものか

macOSは、AppleがすべてのMacに載せるOSです。デスクトップとFinder、付属のアプリ、その下に、ターミナル、シェル、使い慣れたコマンドラインの道具を持つUnixシステムがあります。Appleは、自社のハードウェア用に作って両方を一緒に売るため、実際には、macOSを選ぶことはMacを選ぶことです。

中核は、Appleがソースを公開するDarwinです。そのカーネルXNUは、Carnegie Mellon UniversityのMachカーネル、FreeBSD由来のコード、C++のドライバーフレームワークIOKitを合わせたハイブリッドです。名前は「X is Not Unix」の略です。ユーザー空間の多くもFreeBSD由来で、Macの `sed` と `date` のマニュアルは、FreeBSD版を説明します。macOS 10.15 Catalina以降、既定のシェルはzshです。

OSを支えるハードウェアにも、近年変化がありました。Appleは2020年末、MacをIntelのプロセッサーから、自社チップApple siliconに移し始めました。今ではMacBook AirからMac Proまで、すべてのMacの製品系列にApple siliconモデルがあります。それらはarm64であり、Macで作ったソフトウェアをx86-64サーバーで動かすときに重要になります。

ノートPCとデスクトップでは、macOSはWindowsやLinuxと比較して選ばれます。

概念を判断に生かす

「macOS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。ノートパソコン、プリンター、ログイン、メールの問題を、利用者が気後れしない平易な説明で、速やかに解決します。

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。 技術サポート部門を設立 — 10,000 ゼロから設立したサポート部門が対応した顧客数

技術サポート部門を設立 お問い合わせ →

なぜ重要なのか

ソフトウェアを作る企業にとって、macOSには混同しやすい二つの役割があります。多くの開発者が仕事に使うマシンであり、顧客がソフトウェアを動かすプラットフォームでもあります。

開発用マシンとしてのMacは、ソフトウェアの配備先となるLinuxサーバーと、三つの点で違います。コマンドラインの道具がGNUではなくFreeBSD由来なので、同じフラグでも、スクリプト内で意味が違うことがあります。Macのプロセッサーはarm64でも、サーバーはx86-64かもしれません。また、MacにはLinuxカーネルがありません。Linuxコンテナーはホストのカーネルを共有するため、それをMacで動かす道具は、最初にLinux仮想マシンを起動します。Apple自身の `container` は、Apple siliconとmacOS 26を必要とし、各コンテナーに、それぞれ軽量な仮想マシンを用意します。どれもMacを使えない理由ではありません。ノートPC上での成功を信用するのではなく、配備先のシステムで試す必要があるということです。

実行先としては、macOS向けにどう作るかが問題になります。ブラウザーエンジンを埋め込んだ一つのコードベースを、Windows、Linux、Mac用に配布する複数OS共通のアプリケーション基盤なら、少ない費用で、複数のシステム向けの版を同じ状態に保てます。ただしMacでも、ネイティブアプリには不要なブラウザー実行環境を持ち、システムの部品ではなく独自の操作部品を描きます。SwiftとSwiftUIのネイティブアプリは、システムの部品とマテリアルを使い、追加のコードなしに、利用者による外観、アクセントカラー、コントラスト、文字サイズ、動きを減らす設定に従います。その分、専用のクライアントと、プラットフォームに詳しい人材が必要です。顧客がMacの利用者であるときに、ネイティブ化は割に合います。

Macへの配布にも、通過すべき検査があります。Mac App Store外で配るソフトウェアは、Developer ID証明書で署名し、Appleの公証サービスに送ります。悪意ある内容とコード署名の問題を自動検査するもので、App Reviewとは別です。macOS 10.15以降、2019年6月1日より後にビルドし、Developer IDで配布するソフトウェアは、すべて公証が必要です。Appleは、ソフトウェアに添付するチケットを返し、同じチケットをオンラインにも公開します。利用者が初めて開

くとき、Gatekeeperがそこで探します。証明書は、年額$99のApple Developer Programに含まれます。加入と署名の手順は、最初のリリースから計画に入れます。

実際の活用例

あるメッセージ製品のmacOS版は、Macアプリらしい見た目も動作もない、複数OS共通のアプリケーション基盤でした。十分にテストされたRustのコアがCインターフェースを公開し、SwiftはCを呼べるため、SwiftとSwiftUIのネイティブクライアントを、そのインターフェースの上に直接作りました。コアは、上流の固定した版から静的アーカイブとしてリンクし、ビルドの再現性を保ちます。言語間を渡るすべてのポインターは、薄いラッパーが所有します。クライアントは、Macアプリとして起動し、プラットフォーム自身のマテリアルと操作部品を使い、ブラウザーを埋め込みません。ページは不足も明示します。公証済みの配布も更新経路もなく、動くアプリではあっても、他者が実際に使う製品にはなっていません。両者の間にあるのが、前述の公証手順です。

プラットフォームの見た目は、まねるだけでなく、規則で強制できます。六群の名前付き48トークンによるデザインシステムは、色とマテリアルを固定値ではなく、プラットフォームの意味に応じた色とガラスのマテリアルから組み立てます。変化を監視するコードなしに、外観、アクセントカラー、コントラストの設定に従い、文字のスタイルは、利用者が選んだ大きさに合わせて伸縮します。リンターは、トークンが必要な場所の生の数値、固定したフォントサイズ、動きを減らす設定を無視するアニメーションを拒否します。ページは限界も述べます。トークンが生むのは一貫性であり、良いインターフェースにすることとは別です。

圧縮メモリーとスワップによって、macOSアプリは、マシンの物理メモリーをはるかに超えて増大できます。あるインシデントの記録は、2026-09-01に36 GBのマシンで圧縮ページが111 GBになった診断報告と、接続の再試行ループで、毎秒約41 MB増えた別のメモリー使用量の採取結果を記載します。イベント経路は、コアのログイベントを、上限のない主ストリームに保持し、それを使わない購読先にもコピーしていました。バッファーに上限を設け、購読をイベント種別で絞り、ログ頻度に予算を与えました。全テスト実行の比較では、コアのログが610,996行から1,411行に減りました。テスト実行の測定であり、修正後のメモリー測定でも、本番のベンチマークでもありません。その代償も明示します。繰り返す診断イベントを落とし、無制限にイベントを保持する代わりに、影響を受けた状態を取り直す場合があります。終了時の別のタイムアウトには、別の修正が必要でした。

問題が起きるところ

正常に動くMacなのに、Activity Monitorには、空きメモリーがほとんどありません。 macOSは、使っていないメモリーに最近のファイルをキャッシュし、満ちてくると、動作中のアプリのために、非アクティブなアプリを圧縮します。したがって、Memory Used（使用中のメモリー）がPhysical Memory（物理メモリー）に近く、Compressed（圧縮済み）が大きくても、正常です。Apple自身の説明では、空きメモリーが多ければ必ず速くなるわけではありません。代わりに、Memory Pressure（メモリーへの圧力）のグラフを見ます。緑ならRAMを効率的に使い、黄なら、いずれ追加が必要になる可能性があり、赤なら追加が必要です。最も明白な兆候は、スワップです。

```
sysctl vm.swapusage
vm_stat | grep -E '^Swap(ins|outs)'
```

`vm_stat` のカウンターは、起動時からの累積です。一分間隔の二回の値を比べます。スワップアウトが増え続けるなら、仕事量がRAMの容量を超えています。

Macでは動くスクリプトが、Linuxサーバーで奇妙なエラーになったり、その逆が起きたりします。 MacのFreeBSD由来の `sed` は、 `-i` の次の語をバックアップの接尾辞と解釈します。そのため、GNUでよく使う `sed -i 's/a/b/' file` では、式が接尾辞に、ファイル名がスクリプトになります。Macで使う `sed -i '' …` は、GNUの `sed` に、式をファイル名として読ませます。フラグに直接付けた接尾辞なら、両方で動きます。

```
sed -i.bak 's/^port=80$/port=8080/' app.conf && rm app.conf.bak
```

恒久的な対処は、本番と同じシステム環境で、CIからスクリプトを実行することです。

チーム外に渡したアプリを受取人が開くと、Gatekeeperに止められます。 アプリが開発用に署名されているか、署名しただけで公証していません。Developer ID証明書で署名し、強化ランタイムを有効にし、安全なタイムスタンプを付け、公証サービスに送ります。オフラインでもGatekeeperが見つけられるよう、チケットを添付します。ZIPは公証できますが、チケットは添付できないため、中のアプリに添付します。

```
ditto -c -k --keepParent App.app App.zip
xcrun notarytool submit App.zip --keychain-profile notary --wait
xcrun stapler staple App.app
```

開発者のMacで動くコンテナーイメージが、サーバーでは `exec format error` で止まります。 Dockerは、既定でBuildKitデーモンのプラットフォーム向けにビルドします。Apple siliconのMacでは、そのデーモンはarm64仮想マシンで動きます。コンテナーは、ホストのカーネルを共有するため、エミュレーションなしのx86-64ホストは、x86-64コードしか実行できません。 `uname -m` でサーバーを確認し、 `docker build --platform linux/amd64` でその環境用に作るか、サーバーと同じCPUの種類を持つマシン上のCIでビルドします。

参考資料

- Macユーザーガイド — 現行の各リリース向けに維持する、Apple自身のmacOSガイドです。
- MacにRAMの追加が必要か確認する — Memory Pressureグラフと三色の意味です。
- 配布前のmacOSソフトウェアの公証 — Developer ID、強化ランタイム、Gatekeeperによるチケットの利用です。
- 複数プラットフォーム向けのビルド — arm64とamd64のホストで別のイメージが必要な理由と、一度に両方作る方法です。
- container — MacでLinuxコンテナーを動かすAppleの道具です。それぞれが軽量仮想マシンで動きます。
- XNU — macOSカーネルのソースです。Mach、FreeBSD、IOKitの組み合わせを短く説明します。

用語集

実際の制約に合わせて選ぶ

機器やサービス、観測された症状、保持すべきデータをご説明ください。安全性を最も重視しながら、役に立つ診断の次の一手を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「macOS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/macos/