Engineer

macOS

настільна операційна система Apple

Що це таке

macOS — це операційна система, яку Apple постачає на кожному Mac: робочий стіл і Finder, вбудовані програми, а під ними система Unix із терміналом, оболонкою та звичними інструментами командного рядка. Apple створює її для власного обладнання й продає їх разом, тож на практиці обрати macOS означає обрати Mac.

Основа системи — Darwin, вихідний код якого Apple публікує. Її ядро, XNU, — гібрид, що поєднує ядро Mach з Університету Карнегі-Меллона з компонентами FreeBSD і драйверним фреймворком на C++ під назвою IOKit; назва розшифровується як «X is Not Unix». Значна частина користувацького простору теж походить із FreeBSD, і сторінки довідки для `sed` і `date` на Mac описують саме версії цих інструментів із FreeBSD. Починаючи з macOS 10.15 Catalina, оболонкою за замовчуванням є zsh.

Нещодавно змінилося й обладнання. Наприкінці 2020 року Apple почала переводити Mac із процесорів Intel на власні чипи, Apple silicon, і тепер кожна лінійка Mac, від MacBook Air до Mac Pro, має моделі на Apple silicon. Ці чипи виконують код arm64, і це стає важливим, щойно програму, зібрану на Mac, треба запустити на сервері x86-64.

На ноутбуках і настільних комп'ютерах macOS порівнюють із Windows і Linux.

Застосуйте поняття до рішення

Застосуйте «Що таке macOS?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Ноутбук, принтер, вхід у систему і поштова скринька — швидко, зрозумілою мовою і без зверхності.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Заснувала відділ технічної підтримки — 10,000 клієнтів обслуговує відділ підтримки, створений із нуля

Заснувала відділ технічної підтримки Зв'язатися з нами →

Чому це важливо

Для компанії, що пише програмне забезпечення, macOS відіграє дві ролі, які легко сплутати: це машина, на якій працює багато розробників, і платформа, на якій клієнти запускають програми.

Як робоча станція Mac відрізняється від серверів Linux, на які розгортають код, за трьома ознаками. Інструменти командного рядка походять із FreeBSD, а не з GNU, тож той самий прапорець у скрипті може означати різне. Процесор — arm64, тоді як сервер може бути x86-64. І ядра Linux на Mac немає, а контейнери Linux використовують ядро свого хоста, тому інструменти для їх запуску на Mac спершу запускають віртуальну машину з Linux; власний інструмент Apple, `container`, якому потрібні Apple silicon і macOS 26, дає кожному контейнерові окрему легку віртуальну машину. Це не причина відмовлятися від Mac, а причина тестувати на тій системі, куди розгортаєте, замість того, щоб довіряти зеленому запуску на ноутбуці.

Як платформа macOS ставить питання, як під неї будувати. Кросплатформна оболонка, тобто одна кодова база з вбудованим браузерним рушієм, яку постачають для Windows, Linux і Mac, дешево тримає кілька систем узгодженими, але на Mac вона тягне за собою середовище виконання браузера, непотрібне нативній програмі, і малює власні елементи керування замість системних. Нативна програма на Swift і SwiftUI використовує системні елементи керування й матеріали та без додаткового коду йде за налаштуваннями користувача: виглядом, акцентним кольором, контрастом, розміром тексту і зменшенням руху. Ціна — окремий клієнт і люди, які знають платформу. Нативний шлях окупається, коли клієнти — це користувачі Mac.

Постачання на Mac проходить ще й через контроль. Програми, які поширюють поза Mac App Store, підписують сертифікатом Developer ID і надсилають до служби нотаризації Apple — автоматичної перевірки на шкідливий вміст і проблеми з підписом коду, яка є окремою від App Review. Починаючи з macOS 10.15, усе програмне забезпечення, зібране після 1 червня 2019 року й поширюване з Developer ID, має бути нотаризоване. Apple повертає квиток, який прикріплюють до програми, і публікує той

самий квиток онлайн, де Gatekeeperег шукає його, коли користувач уперше відкриває програму. Сертифікат дає членство в Apple Developer Program за 99 доларів на рік. Закладіть членство й крок підписування в перший реліз.

Як це виглядає на практиці

Настільна версія одного продукту для обміну повідомленнями на macOS була кросплатформною оболонкою, яка й виглядала, і поводилася інакше, ніж програма для Mac. Добре випробувана основа продукту, написана на Rust, має C-інтерфейс, а Swift уміє викликати C, тож нативний клієнт на Swift і SwiftUI збудували безпосередньо на цьому інтерфейсі. Основу вкомпоновано як статичний архів із зафіксованої ревізії основного проєкту, що робить збирання відтворюваним, а тонка обгортка володіє кожним вказівником, який перетинає межу між мовами. Клієнт запускається як програма для Mac, використовує власні матеріали й елементи керування платформи і не містить вбудованого браузера. Сторінка чесно називає прогалину: немає ні нотаризованого розповсюдження, ні каналу оновлень, тож це робоча програма, а не продукт, яким користуються інші. Саме описаний вище крок нотаризації стоїть між цими двома станами.

Вигляд платформи можна закріпити правилом, а не наслідуванням. Система дизайну з 48 іменованих токенів у шести групах будує колір і матеріал на семантичних кольорах платформи та її скляному матеріалі замість сталих значень, тож інтерфейс іде за системним виглядом, акцентним кольором і налаштуваннями контрасту без коду, який би за ними стежив, а текстові стилі масштабуються з розміром, який обрав користувач. Лінтер відхиляє голе число там, де має бути токен, жорстко заданий розмір шрифту й анімацію, що ігнорує налаштування зменшеного руху. Сторінка називає й свою межу: токени роблять інтерфейс узгодженим, а це не те саме, що зробити його добрим.

Стиснення пам’яті й підкачування дають застосунку macOS змогу вирости далеко за межі фізичної пам’яті машини. Один запис про інцидент документує діагностичні звіти про 111 ГБ стиснутих сторінок на машині з 36 ГБ від 2026-09-01 та окремий замір обсягу зайнятої пам’яті зі зростанням приблизно на 41 МБ за секунду під час циклу повторних спроб підключення. Шлях обробки подій утримував події журналу ядра програми в необмеженому основному потоці й копіював їх підписникам, які цих подій не використовували. Буфери обмежили, підписки відфільтрували за типом події, а журналюванню надали бюджет частоти записів. Порівняння повних тестових запусків зафіксувало скорочення 610 996 рядків журналу ядра

програми до 1 411; це вимірювання тестових запусків, а не виміряний показник використання пам’яті після виправлення чи бенчмарк у виробничому середовищі. Компроміс названо прямо: повторювані діагностичні події можна відкидати, а відповідний стан — оновлювати, замість того щоб зберігати необмежену кількість подій. Окремі тайм-аути під час завершення роботи потребували окремого виправлення.

Де виникають проблеми

Монітор активності показує майже повністю зайняту пам’ять на Mac, який працює цілком нормально. macOS кешує нещодавно використані файли в пам’яті, яка інакше простоювала б, а коли пам’ять заповнюється, стискає неактивні програми, щоб звільнити місце для активних. Тому використана пам’ять, близька до фізичної, і великий обсяг стиснутої — це норма. Сама Apple пише, що вільна пам’ять ще не означає вищої продуктивності комп’ютера. Дивіться натомість на графік «Навантаженість пам’яті»: зелений означає, що оперативна пам’ять використовується ефективно, жовтий — що згодом комп’ютеру може знадобитися більше, червоний — що її потрібно більше. Найпростіший сигнал — підкачування:

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

Лічильники `vm_stat` ідуть від завантаження системи, тож порівнюйте два зчитування з інтервалом у хвилину; якщо swapouts і далі зростають, роботі вже затісно в оперативній пам’яті.

Скрипт, який працює на Mac, падає з дивною помилкою на сервері Linux, або навпаки. Похідний від FreeBSD `sed` на Mac сприймає слово після `-i` як суфікс резервної копії, тож звичний для GNU виклик `sed -i 's/a/b/' file` бере вираз за суфікс, а ім’я файлу — за скрипт. Звичка з Mac, `sed -i '' …`, змушує GNU `sed` читати вираз як ім’я файлу. Суфікс, записаний разом із прапорцем, працює в обох системах:

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

Надійне рішення — запускати скрипти в CI на тій системі, з якою вони зустрінуться в робочому середовищі.

Збірку, яку передали за межі команди, блокує Gatekeeper, щойно отримувач її відкриває. Програму підписано для розробки або підписано, але не нотаризовано. Підпишіть її сертифікатом Developer ID з увімкненим hardened runtime і захищеною позначкою часу, надішліть до служби нотаризації й прикріпіть квиток, щоб Gatekeeper знаходив його навіть без мережі. ZIP-архів можна нотаризувати, але прикріпити до нього квиток не вийде, тож прикріплюйте його до програми всередині архіву:

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

Образ контейнера працює на Mac розробника й падає на сервері з `exec format error` **.** Docker за замовчуванням збирає під платформу демона BuildKit, який виконує збирання, а на Mac з Apple silicon цей демон працює у віртуальній машині arm64. Контейнер використовує ядро хоста, тож без емуляції хост x86-64 виконує лише код x86-64. Перевірте сервер командою `uname -m`, а потім збирайте під нього з `docker build --platform linux/amd64` або збирайте в CI на платформі сервера.

Що почитати далі

- Mac User Guide — власний посібник Apple з macOS, який оновлюють для кожного актуального випуску.
- Check if your Mac needs more RAM — графік навантаженості пам’яті і що означають три його кольори.
- Notarizing macOS software before distribution — Developer ID, hardened runtime і як Gatekeeper використовує квиток.
- Multi-platform builds — чому хостам arm64 і amd64 потрібні різні образи і як одне збирання дає обидва.
- container — інструмент Apple для запуску контейнерів Linux на Mac, кожного в окремій легкій віртуальній машині.
- XNU — вихідний код ядра macOS із коротким поясненням, як поєднуються Mach, FreeBSD та IOKit.

Словник

Обирайте за фактичними умовами

Опишіть пристрій або сервіс, спостережувані симптоми та дані, які слід зберегти. Ми можемо визначити найбезпечніший корисний діагностичний крок.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке macOS?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

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

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/macos/