Wine (to nie jest emulator) — uruchamianie aplikacji Windows na innych systemach

Ten materiał porównuje wirtualizację, emulację oraz warstwę kompatybilności Wine. Zawiera praktyczne przykłady (QEMU, VMware/VirtualBox, DOSBox) i wskazówki, kiedy użyć którego podejścia. Na końcu znajdziesz krótką notatkę o numeracji wersji Windows (np. 10.0.22000 oznacza Windows 11).

1. Kluczowe pojęcia

- - Wirtualizacja: uruchamianie systemu gościa bezpośrednio na sprzęcie gospodarza z pomocą hipernadzorcy (np. KVM, VMware ESXi, Hyper-V). Minimalna warstwa pośrednia zapewnia wydajność zbliżoną do natywnej.
- - Emulacja: pełne odwzorowanie sprzętu w oprogramowaniu (CPU, urządzenia). Pozwala uruchomić system/aplikacje dla innej architektury, kosztem wydajności.
- - Wine: warstwa kompatybilności — tłumaczy wywołania API Windows na wywołania POSIX systemu gospodarza. Nie emuluje całego Windows, dlatego „Wine Is Not an Emulator".

2. QEMU: wirtualizacja i emulacja

QEMU potrafi działać w dwóch trybach:

- - Sprzętowo przyspieszona wirtualizacja (np. KVM/hvf/whpx) — wysokia wydajność, wspólna architektura host/guest.
- - TCG (Tiny Code Generator) — emulacja CPU dla innej architektury (wolniejsze, ale uniwersalne).

Przykład wirtualizacji (x86_64 + KVM):

qemu-system-x86_64 -enable-kvm -m 4096 -hda windows.qcow2

Przykład emulacji (inna architektura, bez KVM):

qemu-system-aarch64 -cpu cortex-a53 -M virt -m 2048 -drive file=win.img,if=none,format=qcow2,id=disk -device virtio-blk-device,drive=disk

3. VMware/VirtualBox/Hyper-V: wirtualizacja

Produkty te korzystają ze wirtualizacji, nie pełnej emulacji: gość uruchamia kod bezpośrednio na CPU gospodarza, a hipernadzorca przechwytuje tylko wybrane zdarzenia/operacje uprzywilejowane.

4. DOSBox: emulator MS-DOS

DOSBox emuluje środowisko MS-DOS (CPU, grafika, dźwięk) — idealny do uruchamiania starszych gier i aplikacji DOS/Windows 3.x. To emulator, nie wirtualizator.

5. Wine: warstwa kompatybilności dla aplikacji Windows

Wine tłumaczy wywołania WinAPI na POSIX i integruje aplikacje Windows bez uruchamiania pełnego systemu gościa. Dla komponentów HTML wymagany jest Wine Gecko (x86 i x86_64), a dla aplikacji .NET — Wine Mono lub właściwe runtime'y.

6. Wine w kontenerach (Docker/Podman)
Konteneryzacja pozwala izolować aplikacje uruchamiane przez Wine. Każdy kontener może mieć własny WINEPREFIX i zestaw bibliotek, co ułatwia dystrybucję oraz powtarzalność środowiska.

7. Numeracja wersji Windows (przydatne w Wine)
- - 10.0.19041–19045 → Windows 10 (20H1–21H2).
- - 10.0.22000 → Windows 11 RTM (2021).
- - 10.0.26100.x → Windows 11 24H2 (nowsze kompilacje).

Sprawdzenie w Wine: uruchom `wine cmd`, a następnie `ver`. Zmiana wersji: `winecfg` → zakładka Aplikacje → Wersja Windows.

8. Kiedy wybrać które podejście?
- - Wirtualizacja (VMware/VirtualBox/QEMU+KVM): pełne środowisko Windows, najwyższa zgodność; kosztem zasobów.
- - Emulacja (QEMU/TCG, DOSBox): gdy host/guest mają różne architektury albo celem jest retro/legacy.
- - Wine: gdy chcemy szybko uruchomić pojedyncze aplikacje Windows bez VM; wymaga dobrania komponentów.