

# Vulkan API: Moderní standard pro grafiku

**Vulkan** je revoluční grafické API navržené tak, aby minimalizovalo režii procesoru (CPU) a poskytlo vývojářům přímou kontrolu nad grafickým hardwarem. Na rozdíl od svého předchůdce, OpenGL, Vulkan předpokládá, že programátor přesně ví, co dělá, a nese plnou zodpovědnost za správu paměti a synchronizaci.

## 1. Proč vznikl Vulkan? (Konec "Black Box" éry)

Tradiční API jako OpenGL fungovala jako „černá skříňka“. Ovladač dělal mnoho rozhodnutí za programátora (správa paměti, kontrola chyb), což vedlo k nekonzistentnímu výkonu a vysoké zátěži jednoho jádra CPU.

Vulkan toto mění:

- **Explicitní kontrola:** Programátor explicitně spravuje alokaci paměti, vytváření vláken a synchronizaci mezi CPU a GPU.
- **Nižší CPU režie:** Ovladač je velmi tenký a nedělá žádné skryté kontroly chyb během vykreslování.
- **Multi-core Friendly:** Vulkan je od základu navržen pro efektivní škálování na vícejádrové procesory.

## 2. Architektura a klíčové koncepty

Vulkan vyžaduje mnohem více kódu (tzv. boilerplate) pro inicializaci, ale odměnou je předvídatelný výkon.

### Instance a Device

- **Instance:** Spojení mezi aplikací a knihovnou Vulkan.
- **Physical Device:** Reprezentuje konkrétní GPU v systému.
- **Logical Device:** Softwarové rozhraní, přes které aplikace komunikuje s fyzickým zařízením.

### Command Buffers a Fronty (Queues)

Ve Vulkanu se příkazy (vykresli, zkopíruj, vypočítej) neposílají na GPU okamžitě. Zapisují se do **Command Bufferů**, které lze připravovat v několika vláknech současně. Tyto buffery se následně odesílají do fronty (**Queue**) k vykonání na GPU.

### Pipeline State Objects (PSO)

Všechny stavy (shadery, míchání barev, hloubkový test) se definují dopředu při startu aplikace do

objektů typu Pipeline. To eliminuje „záškuby“ (stuttering) během hry, které v OpenGL vznikaly kompilací shaderů za běhu.

### 3. Shaders a SPIR-V

Vulkan nepoužívá shadery v textové podobě (jako GLSL), které by musel ovladač kompilovat. Místo toho používá **SPIR-V** - binární mezikód (podobně jako LLVM IR pro procesory).

- **Výhoda:** Rychlejší načítání aplikací a menší šance na chyby v různých ovladačích (NVIDIA, AMD, Intel).

### 4. Vulkan vs. Konkurence

Vulkan soupeří s dalšími moderními API, ale jeho hlavní silou je otevřenost.

API	Výrobce	Platformy
<b>Vulkan</b>	Khronos Group	Windows, <a href="#">Linux</a> , Android, Nintendo Switch.
<b>DirectX 12</b>	Microsoft	Windows, Xbox.
<b>Metal</b>	Apple	macOS, iOS.

**Poznámka k Apple:** Apple Vulkan nativně nepodporuje, ale komunita vyvinula vrstvu **MoltenVK**, která překládá volání Vulkanu do Metal API.

### 5. Využití v praxi

- **Herní enginey:** Unreal Engine, Id Tech 7 (DOOM Eternal), Unity.
- **Emulátory:** Rpcs3 (PS3), Yuzu/Ryujinx (Switch) - zde Vulkan drasticky zvyšuje výkon.
- **Profesionální software:** [Adobe Photoshop](#) využívá Vulkan pro akceleraci filtrů na Windows a Androidu.
- **Ray Tracing:** Vulkan obsahuje rozšíření pro hardwarově akcelerovaný Ray Tracing (VK\_KHR\_ray\_tracing).

—

**Zajímavost:** Vulkan je evolucí API zvaného **Mantle**, které původně vyvinula společnost AMD. AMD následně darovalo kód skupině Khronos, aby se stal základem pro otevřený průmyslový standard.

*Související:* [GPU](#), [CPU](#), [C++](#), [Rust](#), [Linux](#)

From:  
<https://serviceit.cz/> - IT ENCYKLOPEDIÉ

Permanent link:  
[https://serviceit.cz/doku.php?id=vulkan\\_api](https://serviceit.cz/doku.php?id=vulkan_api)

Last update: 2025/12/31 18:12



