

# Fractions64

Ondrej Nedojedly

18. Března, 2023

## 1 Úvod

Tato dokumentace obsahuje podrobný přehled knihovny Fractions64, což je knihovna jazyka C (používajícího standard 99) pro práci se zlomky s 64bitovou přesností celých čísel. Knihovna obsahuje funkce pro vytváření, manipulaci a provádění operací se zlomky.

Knihovna je navržena tak, aby se snadno používala a integrovala do projektů v jazyce C a poskytuje celou řadu funkcí, umožňující vývojářům pracovat se zlomky v prostředí flexibilním a efektivním způsobem. Kromě toho knihovna obsahuje funkce pro zjednodušení zlomků, převody zlomků a další funkce. zlomků na čísla s pohyblivou řádovou čárkou a další.

Tato dokumentace poskytuje přehled o knihovně včetně příkladů použití jednotlivých funkcí, a popisuje datové struktury knihovny, a rozhraní API. Je určena vývojářům, kteří hledají pracovat se zlomky ve svých projektech v jazyce C a kteří chtějí pracovat se zlomky. porozumět tomu, jak používat knihovnu Fractions64, aby k dosažení tohoto cíle.

## 2 Stažení

Tento projekt k dispozici ke stažení na adrese mého githubu<sup>1</sup>. Projekt můžete naklonovat přímo do svého vývojového prostředí

<sup>1</sup><https://www.github.com/ondranedo/fractions>

IDE<sup>2</sup>, nebo můžete naklonovat projekt pomocí příkazu `git clone`. Pokud však chcete jen použít knihovnu, můžete si složku se zdrojovými kódy stáhnout ručně, jsou tam koneckonců jen dva zdrojové soubory.

*fractions64.h*

*fractions64.c*

## 3 Jak nastavit projekt

Pokud jste si zvolili jinou variantu sestavení, dovloste mi podotknou že že je třeba projekt sestavit jako `.lib`. Celé toto může udělat nástroj CMake<sup>3</sup>, který vytvoří `.lib` soubor za vás, ale klidně použijte vlastní nástroje pro sestavení.

Pro ty, kteří si zvolili sestavovací nástroj CMake, stačí přímo naklonovat tuto knihovnu do navržené složky pro externí nástroje. knihovny (něco jako `/vendor`). A do své `root` `CMakeLists.txt` přidejte tento příkaz: `addSubdirectory([your_specified_path])`. Musíte také zahrnout přidruženou hlavičku, která se nachází ve složce `/include`. A nakonec musíte do svého spustitelného souboru zahrnout soubor `.lib`. název souboru `lib` je `frac`.

Pro ty, kteří používají jiný způsob sestavování, mějte na paměti, že vaše verze překladače a OS<sup>4</sup> nemusí být podporovány,

<sup>2</sup>Integrované vývojové prostředí

<sup>3</sup><https://cs.wikipedia.org/wiki/CMake>

<sup>4</sup>Operační systém

překladače na kterých byl testován, jejich přidružená verze a operační systém jsou uvedeny v seznamu zde:

- **Windows**

Nebylo testováno.

- **MacOS**

Úspěšně otestováno a vyvinuto na Clangu<sup>5</sup> verze *Apple clang verze 14.0.0 (clang-1400.0.29.202)* pro target *arm64-apple-darwin22.2.0* pomocí verze CMake: *3.25.1: 3.25.1*.

- **Linux**

Nebylo testováno.

Existuje však alternativa pro ty, kteří se nesnaží knihovnu vyvíjet. Pro tyti lidi je zde již předvytvořen projekt pro různá IDE. Tyto projekty můžete nalézt ve složce */build*

1. **RedPanda**

2. **Visual Studio 2023**

V těchto projektech najdete i předvytvořené demo soubory, pro ujasnění funkcí knihovny. Tyto spustitelné soubory je třeba ale předem zkompileovat.

## 4 API

API této knihovny je dodržováno striktním dodržováním pojmenovávací konvencí. Ta je následující:

1. Všechna funkce a datové typy začínají jménem **fracXxx**.
2. Vrací li funkce paměť vytvořenou na haldě<sup>6</sup> má ve svém názvu nakonci **H**. Příklad: *fracAddFraction64H*
3. Za název **frac** je vždy operace kterou chceme vykonávat.

<sup>5</sup><https://cs.wikipedia.org/wiki/Clang>

<sup>6</sup><https://cs.wikipedia.org/wiki/Halda>

### 4.1 Vytváření zlomku

Zlomky je možno vytvořit dvěma způsoby, a to:

1. Skrze funkce **Create**

2. Skrze funkce **GetCpy**

Metoda **Create** se dá rozdělit na dvě varianty, a to na vytváření pomocí jmenovatele s čitatele, a nebo pomocí desetinného čísla pomocí funkce **CreateFloat**.

Jak metoda **GetCpy** tak metoda **Create** podporují **H** variantu, která vrátí adresu zlomku, který byl vytvořen na haldě

### 4.2 Ničení zlomku

Pokud zlomek již nebude potřebován je jej třeba zničit skrze funkce **Destroy**. A jeli zlomek uchovávan jako adresa na haldě, je třeba zavolat funkci **Destroy** s **H** variantou.

### 4.3 Getters a Setters

Nastavují hodnotu zlomku, či získají hodnotu ze zlomku.

- **fracGetA** vrací čitatele zlomku.
- **fracGetB** vrací jmenovatele zlomku.
- **fracSet** nastaví jmenovatel a čitatele zlomku.
- **fracSetA** nastaví čitatele zlomku.
- **fracSetB** nastaví jmenovatele zlomku.

### 4.4 Matematické operace

Každá funkce má **H** variantu, která vrátí adresu zlomku, který byl vytvořen na haldě, a variantu **Overwrite**, jež přepíše první zlomek, který byl dán jako parametr pro volání funkce s novými hodnotami.

- **fracAdd** sečte zlomky.
- **fracSub** odečte zlomky.
- **fracMul** vynásobí zlomky.
- **fracDiv** vydělí zlomky.
- **fracPow** dá zlomek na mocninu jiného zlomku.

#### 4.5 Záměna jmenovatele a čitatele

Chceme li zamenit jmenovatele a čitatele zlomku, udělat takzvanou obrácenou hodnotu voláme funkci **fracSwitchAB**, chceme li zaměnit čitatele či jmenovatele u dvou zlomků, voláme funkce **fracSwitchA** pro čitatele a **fracSwitchB** pro jmenovatele.

#### 4.6 Výpis zlomku do souboru

Chceme li vypsát zlomek do souboru konzole či do námi otevřeného souboru, můžeme využít funkce **Dump** jejíž první parametr je onen soubor. Druhý parametr je adresa zlomku, a třetím parametrem je formát. Je-li formát nastaven na hodnotu **NULL** je využit defaultní formát:  $a/b$ . Chceme li nastavit náš vlastní formát použijeme pro čitatele **%a** a jmenovatele **%b** př.:

$$\frac{5}{12} \xrightarrow{ "[%a, %b] \%b\ ano"} [5, 12] 12\ ano$$

#### 4.7 Další funkce

- **fracCom** porovná dva zlomky. Vrací 0 jsou li stejné, 1 je-li první zlomek větší, a  $-1$  je-li druhý zlomek větší.
- **Simplify** převede zlomek na jeho zjednoduší variantu.
- **Float** vrátí zlomek v desetinném čísle.
- **LCD** vrátí nejmenší společný jmenovatel pro dva zlomky.

- **SetLogLevel** nastaví úroveň výpisu. Defaultní úroveň je nastavena na **FRAC\_ERROR**.

1. **FRAC\_ALL** vypíše všechny zprávy.
2. **FRAC\_WARNING** vypíše všechny varování a chybové stavy.
3. **FRAC\_ERROR** vypíše pouze chybové stavy.
4. **FRAC\_NONE** nevypíše nic.

- **SetDefaultSimplification** defaultní stav automatického zjednodušování zlomků je **FRAC\_TRUE**, možno vypnout defaultní zjednodušování zlomků posláním parametru **FRAC\_FALSE**.
- **PowDouble** vypočte desetinnou hodnotu na mocninu zlomku.