

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Dokumentace ke 2. projektu IVS

Hrdza tým

Obsah

1	Úvod	2
2	Instalace a odinstalace programu	2
2.1	Instalace pomocí DEB balíčku	2
2.2	Odinstalace pomocí DEB balíčku	2
3	Manuální instalace	2
4	Manuální odinstalace	3
5	Návod k použití	4
6	Matematický Výpočtový Engine	5
6.1	Tokenizace	5
6.2	Konverze do Postfix Notace	5
6.2.1	Využití Bottom-Up Parsingu	5
6.3	Výpočet Postfix Výrazu	5
7	Implementační Detaily	5
7.1	Bezpečnost a Robustnost	5
8	Závěr	5

1 Úvod

Projekt IVS je zaměřen na vývoj desktopové aplikace kalkulačky s využitím technologií Rust a Tauri. Cílem je poskytnout uživateli grafické rozhraní pro výpočet matematických operací.

2 Instalace a odinstalace programu

2.1 Instalace pomocí DEB balíčku

Instalace aplikace probíhá pomocí DEB balíčku, který obsahuje všechny potřebné soubory a závislosti. Uživatel si stáhne balíček a nainstaluje ho následujícím způsobem:

```
sudo dpkg -i calculator_1.0.0_amd64.deb
```

Tento příkaz nainstaluje aplikaci a automaticky přidá ikonu do startovacího menu, což uživatelům umožní snadno aplikaci spouštět.

2.2 Odinstalace pomocí DEB balíčku

Pro odinstalaci aplikace nainstalované z DEB balíčku se použije následující příkaz:

```
sudo dpkg -r calculator_1.0.0_amd64
```

Tento příkaz odstraní všechny soubory aplikace z počítače, včetně ikon.

3 Manuální instalace

Pro uživatele, kteří preferují manuální instalaci, je nutné nejprve získat zdrojové soubory. Po jejich stažení se použijí následující příkazy pro sestavení aplikace:

Získáme zdrojové soubory aplikace a aktualizujeme si systémové knihovny:

```
cd calculator-main/src
sudo apt update
sudo apt install libwebkit2gtk-4.0-dev \
    build-essential \
    curl \
    cargo \
    wget \
    file \
    libssl-dev \
    libgtk-3-dev \
    libayatana-appindicator3-dev \
    librsvg2-dev
```

Následně můžeme vytvořit DEB balíček, který umožní přidání aplikace do systému.

```
make dependencies
make install
```

Příkaz "make install" přidá aplikaci do knihovny aplikací operačního systému Ubuntu, přesněji:

```
sudo dpkg -i target/release/bundle/deb/calculator_1.0.0_amd64.deb
sudo dpkg -i target/debian/profiling_0.1.0_amd64.deb
```

4 Manuální odinstalace

Pro odstranění aplikace nainstalované manuálně je třeba odinstalovat všechny komponenty a nástroje, které byly použity při instalaci. To zahrnuje odinstalaci balíčků Rust a Tauri CLI a odstranění jakýchkoli závislostí nainstalovaných v rámci procesu. Příkazy mohou zahrnovat:

```
cargo uninstall create-auri-app  
cargo uninstall tauri-cli  
cargo uninstall flamgraph  
cargo uninstall cargo-deb  
make uninstall
```

Příkaz "make uninstall" odstraní aplikaci z knihovny aplikací operačního systému Ubuntu, přesněji:

```
sudo dpkg -r calculator  
sudo dpkg -r profiling
```

5 Návod k použití

Pro zapnutí kalkulačky je potřeba najít a spustit program 'calculator' v menu. Po otevření už lze kalkulačku libovolně používat pro výpočet příkladů.

Funkcionalita kalkulačky:

- **C** – smaže veškerý text v okně – klávesová zkratka `Escape`
- **←** – smaže poslední znak – klávesová zkratka `Backspace`
- **0-9** – tlačítka pro zadání číslic
- **,** – udává desetinnou čárku v čísle
- **π** – číselná konstanta pí – klávesová zkratka `p` (`pi`)
- **%** – operace procentuálního podílu
- **+** – operace součtu
- **-** – operace rozdílu
- ***** – operace součinu
- **/** – operace podílu
- **^** – operace mocniny – klávesová zkratka `e` (`exponent`)
- **$\sqrt{}$** – operace odmocniny – klávesová zkratka `r` (`root`)
- **!** – operace faktoriálu
- **()** – závorky (je potřeba, aby každá otevírací závorka k sobě měla i zavírací závorku)
- **=** – vypočítá a nahradí okno výsledkem – klávesová zkratka `Enter`

Není li uvedeno, klávesová zkratka je totožná se znakem popisujícím klávesu.

Pro rychlé zjištění funkčnosti některých z tlačítek je možné zavítat do nápovědy umístěné v levém horním rohu.

6 Matematický Výpočtový Engine

V rámci aplikace je integrován matematický výpočtový engine založený na Rustu, který zpracovává vstupní výrazy a konvertuje je do postfixové notace pro jednodušší výpočet. Tento engine používá tokenizaci pro rozpoznání čísel, operátorů a závorek v matematickém výrazu.

6.1 Tokenizace

V prvním kroku tokenizačního procesu engine čte vstupní řetězec a rozdělí ho na jednotlivé tokeny (čísla a operátory). Každý token je identifikován a přidělen do vhodné kategorie, což umožňuje další zpracování.

6.2 Konverze do Postfix Notace

Po tokenizaci následuje konverze z infixové notace (standardní matematická notace) do postfixové notace. Tento krok je zásadní pro zjednodušení výpočtového procesu, protože umožňuje vyhodnocení výrazů bez nutnosti zohledňovat prioritu operátorů při samotném výpočtu.

6.2.1 Využití Bottom-Up Parsingu

V našem projektu je implementován bottom-up parsing s využitím precedenční tabulky pro transformaci matematických výrazů do postfixové notace. Během analýzy jsou operátory podle jejich priority v precedenční tabulce buď vkládány na zásobník nebo redukovány. Tento postup zaručuje, že operace jsou ve výsledném vyhodnocení prováděny ve správném pořadí.

6.3 Výpočet Postfix Výrazu

V posledním kroku engine vyhodnotí postfixový výraz pomocí zásobníku. Pro každý operátor v postfixovém výrazu se získají potřebné hodnoty z vrcholu zásobníku, provede se operace, a výsledek je vrácen zpět do zásobníku. Výsledek celého výrazu je poté hodnota, která zůstane na vrcholu zásobníku.

7 Implementační Detaily

Pro implementaci matematického engine byl použit jazyk Rust. Toto rozhodnutí bylo motivováno potřebou vysokého výkonu a efektivity při zpracování matematických operací a vyzkoušením si něčeho nového než je jazyk C nebo python.

7.1 Bezpečnost a Robustnost

Velký důraz byl kladen na bezpečnost a robustnost aplikace. Všechny komponenty byly pečlivě testovány, aby byly odolné vůči chybám a nevalidním vstupům.

8 Závěr

Projekt usiluje o vytvoření robustní a uživatelsky přívětivé aplikace. Díky technologii Tauri je aplikace multiplatformní, běžící na systémech s Ubuntu a potenciálně dalších.