

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



## Konverze obrazového formátu GIF na BMP (Projekt č. 3)

Projekt do předmětu Kódování a komprese dat (KKO)

# 1 Úvod

Cílem projektu bylo vytvořit knihovnu a aplikaci (v jazyce C/C++) pro převod souboru grafického formátu GIF na soubor grafického formátu BMP.

## 2 LZW algoritmus

LZW algoritmus<sup>1</sup> je slovníková kompresní metoda, jejíž modifikace se používá pro zakódování obrazových dat GIF souboru. Pomocí algoritmu jsou zakódovány indexy do tabulky barev, kterými jsou reprezentovány všechny pixely v GIF souboru, včetně své barvy, případně i průhlednosti.

Modifikace algoritmu LZW spočívá v přidání dvou speciálních kódů do slovníku – inicializačního kódu (*Clear Code*) a kódu indikujícího konec obrazových dat (*End Of Information Code*). Položky slovníku jsou inicializovány na hodnoty indexů tabulky barev a doplněny speciálními kódy. V každém kroku algoritmu je určeno, zda je, nebo není, aktuálně načítaný kód již ve slovníku, a podle toho se změní položka, které se následně přidají do výstupní posloupnosti indexů tabulky barev. Pseudokód modifikovaného LZW algoritmu je uveden níže.

---

**Algorithm 1** Modifikovaný LZW algoritmus

---

```
1: procedure DECODEIMAGEDATA
2:   ▷ CC = CurrentCode ▷ CC.First = index první barvy CC ▷ PC = PreviousCode ▷ PC.First = index první barvy PC
3:   loop:
4:     načti ze souboru další kód (CC)
5:     if CC = inicializační kód then
6:       inicializuj slovník hodnotami z tabulky barev a doplň speciální kódy
7:     else if CC = kód indikující konec obrazových dat then
8:       break
9:     else
10:      if PC = inicializační kód then
11:        ulož index barvy z indexu PC na výstup
12:      else if CC není ve slovníku then
13:        vlož nový záznam do slovníku z indexů PC a PC.First()
14:        vlož nové indexy barev do výstupu z indexů PC a PC.First()
15:      else if CC je ve slovníku then
16:        vlož nový záznam do slovníku z indexů PC a CC.First()
17:        vlož nové indexy barev do výstupu z indexu CC
18:      PC = CC
```

---

## 3 Popis implementace

Knihovna byla implementována v jazyce C++. Rozhraní knihovny tvoří funkce `int gif2bmp(tGIF2BMP* gif2bmp, FILE* inputFile, FILE* outputFile)`, která má jako parametry:

- velikosti jednotlivých souborů (GIF a BMP) v bytech (datový typ `tGIF2BMP`),
- ukazatel na otevřený vstupní soubor GIF, který se bude převádět,
- ukazatel na otevřený výstupní soubor BMP, který se bude vytvářet.

Funkce vrátí 0 v případě, že byl převod úspěšný, a -1, pokud se při převodu vyskytly chyby. Funkce po zavolání zjistí velikost GIF souboru a začne jej zpracovávat po jednotlivých částech<sup>2</sup>. Funkce postupně:

- ověří, zda se jedná o GIF soubor, poté zkontroluje jeho verzi (podporované verze jsou 87a a 89a),
- načte *Logical Screen Descriptor*, který obsahuje parametry obrázku,
- načte *Global Color Table*, kde jsou uloženy jednotlivé barvy obrázku,
- zpracuje rozšíření,

---

<sup>1</sup><https://www.w3.org/Graphics/GIF/spec-gif89a.txt>

<sup>2</sup>[http://giflib.sourceforge.net/whatsinagif/bits\\_and\\_bytes.html](http://giflib.sourceforge.net/whatsinagif/bits_and_bytes.html)

- *Graphics Control Extension*, který definuje průhlednost a animace,
- *Plain Text Extension*, *Application Extension* a *Comment Extension*, které jsou pouze přeskočeny (neobsahují žádné relevantní informace pro BMP soubor),
- načte *Image Descriptor*, *Local Color Table* a *Image Data*; obrázek je definován obrazovými daty (*Image Data*), která jsou komprimována modifikovaným algoritmem LZW. Algoritmus dekódování je popsán v části 2.

Aplikace podporuje animované GIF obrázky, ale převádí pouze první rámeček (uživatel je na tuto skutečnost upozorněn výpisem na standardní výstup), převod prokládaných obrázků je také podporován.

Vzorová aplikace je implementována v jazyce C++ a používá implementovanou knihovnu pro převod grafického formátu GIF na BMP. Pro její přeložení a sestavení je nutné zadat příkaz `make`. Aplikaci lze spustit pomocí příkazu `./gif2bmp`, za kterým následují parametry. Popis podporovaných parametrů příkazové řádky je v tabulce 3.

Přepínač	Parametr	Popis přepínače
<b>-i</b>	<b>&lt;ifile&gt;</b>	Název vstupního souboru v GIF formátu, pokud není zadáný, použije se <code>stdin</code> .
<b>-o</b>	<b>&lt;ofile&gt;</b>	Název výstupního souboru v BMP formátu, pokud není zadáný, použije se <code>stdout</code> .
<b>-l</b>	<b>&lt;logfile&gt;</b>	Název výstupního souboru se statistickými údaji (obsahuje login, velikost GIF a BMP souborů), pokud není zadáný, výpis statistických údajů bude ignorován.
<b>-h</b>	<b>-</b>	Výpis nápovědy na standardní výstup.

Tabulka 1: Parametry příkazové řádky vzorové aplikace

## 4 Testování

Funkčnost knihovny a vzorové aplikace byla testována na sadě obrázků formátu GIF s různými parametry (verze, velikost, prokládání, animace, komentáře a další rozšíření).

## 5 Závěr

Cílem projektu bylo implementovat knihovnu pro převod souboru v grafickém formátu GIF na soubor v grafickém formátu BMP. Funkčnost knihovny i vzorové aplikace byla ověřena pomocí testů na sadě obrázků. Zadání bylo zcela splněno a byla implementována i rozšíření podporující animované GIF soubory a prokládání.