

# Konverze obrázkového formátu GIF na BMP

Michal Šrubař

xsruba03@stud.fit.vutbr.cz

13. března 2016

Tato dokumentace popisuje knihovnu `libgif2bmp.a` a aplikaci `gif2bmp` implementovanou v rámci projektu do předmětu *Kódování a komprese dat*. Celá knihovna včetně aplikace je implementována v jazyce C bez použití externích knihoven. Knihovna poskytuje funkce, pomocí kterých je možné načíst GIF obrázek do paměti, vypsát jeho interní strukturu, převést jej do 24-bitového binárního RGB formátu a převést obrázek na bitmapový obrázek ve formátu BMP. Knihovna umí zpracovat statické<sup>1</sup> 8-bitové GIF obrázky, průhledné pixely jsou konvertovány na pixely bílé barvy a animované GIF obrázky nejsou podporovány. Výstupní bitmap obrázky jsou poté nekomprimované.

## Použití aplikace gif2bmp

1. Převod GIF obrázku `logo.gif` na BMP obrázek `logo.bmp`

```
gif2bmp -i logo.gif > logo.bmp
```

2. Převod GIF obrázku na 24-bitové RGB hodnoty

```
gif2bmp -i logo.gif -r > logo.rgb
```

3. Zobrazení interní struktury GIF obrázku

```
gif2bmp -i logo.gif -p
```

Další volby a popis fungování je možné získat v nápovědě, kterou je možné zobrazit pomocí přepínače `-h`, tj. `gif2bmp -h`.

## Popis modulů knihovny

Knihovna se skládá z několika modulů, které popisuje tabulka [1](#). Slovník, který je reprezentován hašovací tabulkou je implementován jako dynamický kruhový seznam, který je implementován moduly `list` a `gif_dict`. Samotná aplikace `gif2bmp` je poté implementována v souboru `main.c`

---

<sup>1</sup>GIF obrázky bez animací

Modul	Popis
list	lineární seznam
gif_bits	základní bitové a bajtové operace
gif	načtení GIF obrázku do paměti
gif_print	interní strukturu GIF souboru
gif_dict	operace pro práci se slovníkem použitým při LZW dekompresi
gif_interlance	převod prokládaného obrázku na neprokládaný
gif_lzw	implementace LZW algoritmu pro dekompresi dat
gif_rgb	převod GIF obrázku na RGB hodnoty
bmp	mapování GIF obrázku na bitmapu
gif2bmp	převod GIF obrázku na bitmapu

Tabulka 1: Tabulka popisující moduly knihovny `libgif2bmp`

## Dekomprese pomocí algoritmu LZW

Implementační detaily dekomprese, jehož algoritmus popisuje pseukód 1, je možné nalézt v modulu `gif_lzw.c`. `code` reprezentuje kódové slovo a `{code}` hodnotu ve slovníku.

---

### Algorithm 1: LZW Dekomprese

---

**Input:** Posloupnost kodových slov  
**Output:** Posloupnost indexu do tabulky barev

- 1 Inicializace slovníku;  
    Preci Clear Code;  
    code = precí další kod;  
    vytiskni {code};
- 2 LOOP;  
    code-1 = code code = precí další kod;  
    Je kod v tabulce?;
- 3 ANO;;  
    vytiskni {code};  
    k = první prvek z {code};
- 4 NE;;  
    k = první prvek z {code-1};  
    vytiskni {code-1}+k;  
    ;
- 5 do slovníku přidej {code-1}+k;  
    zpět na LOOP;

---

Pokud je přečten speciální kód *Clear Code*, pak nastane reinicializace slovníku a resetování velikosti délky kódového slova. Pokud je přečten kód *End of Information*, pak je čtení dat u konce.