

## Princíp zpracování rastrových obrázků BMP & PCX

Vávra David · Elektrotechnika, Informačné technológie

09.03.2012



První část článku se zabývá principem zpracování rastrových obrázků BMP a PCX. Popisuje tyto formáty, způsob práce s barvami a u PCX formátu popisuje i způsob komprese. Těchto poznatků bylo využito k vytvoření vlastního programu demonstrujícího zpracování obrázků BMP a PCX a umožňujícího několik základních funkcí, jako např. rotaci o 90° vlevo i vpravo, vertikální i horizontální převrácení, inverzi všech i jednotlivých barev. Zjednodušeným popisem programu se pak zabývá druhá část článku.

### 1. Grafický formát BMP

Bitmapový soubor se skládá z následujících částí:

- Bitmap-File header (bitmapová hlavička)
- Bitmap-Information header (informační hlavička o bitmapě)
- Paleta barev (u 24 bitového obrázku se paleta barev nenachází)
- Pole bytů definující bitmapu

```

00000000: 42 4D 82 00 00 00 00 00|00 00 76 00 00 00 28 00
00000010: 00 00 05 00 00 00 03 00|00 00 01 00 04 00 00 00
00000020: 00 00 0C 00 00 00 00 00|00 00 00 00 00 00 00 00
00000030: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00
00000040: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00
00000050: 00 00 00 00 00 00 C0 C0|C0 00 00 00 FF 00 00 FF
00000060: 00 00 00 FF FF 00 FF 00|00 00 FF 00 FF 00 FF FF
00000070: 00 00 FF FF FF 00 72 34|50 00 67 69 60 00 0C DE
00000080: EB 00
  
```

■ Bitmap-file header      ■ Bitmap-Information header  
■ Paleta barev              ■ Obrazová data

Obrázek 1. Ukázka souboru 3×5.bmp v hexadecimálním tvaru

Bitmap-File header				
Offset	Délka	Význam	3×5.bmp (hex)	3×5.bmp (dec)
0	2	identifikace BMP souboru	42 4D	BM
2	4	celková velikost souboru v bytech	82 00 00 00	130 bytů
6	2	musí být 0	00 00	0
8	2	musí být 0	00 00	0
10	4	začátek obrazových dat v bytech	76 00 00 00	118 bytů
Bitmap-Information header				
Offset	Délka	Význam	3×5.bmp (hex)	3×5.bmp (dec)

14	4	určuje počet bytů hlavičky (vždy 40)	28 00 00 00	40 bytů
18	4	šířka obrazu v bodech	05 00 00 00	5 px
22	4	výška obrazu v bodech	03 00 00 00	3 px
26	2	počet bitových rovin	01 00	1
28	2	počet bitů na obrazový bod	04 00	4
30	4	komprese: 0 = žádná, 1 = 4b RLE, 2 = 8b RLE	00 00 00 00	0
34	4	velikost obrázku v bytech (nesmí být 0)	0C 00 00 00	12 Byte
38	4	horizontální rozlišení v pixelech na metr	00 00 00 00	0 px/m
42	4	vertikální rozlišení v pixelech na metr	00 00 00 00	0 px/m
46	4	aktuální počet barev v paletě, 0 = všechny	00 00 00 00	0
50	4	počet významných barev: 0 = všechny barvy	00 00 00 00	0

## Paleta barev

Definuje pole - RGB struktury skládající se z barev použitých v bitmapě. Barvy v tabulce jsou uspořádané s rostoucím významem. Toto je výhodné není-li zobrazovací zařízení schopné zobrazit všechny barvy použité v bitmapě. Paleta obrázku BMP má velikost jako počet barev\*4 byty. Paleta má význam, pokud je počet bitů 1, 4 nebo 8. Při použití 24 bitů na bod je barva uložena přímo v datech jako postupnost hodnot jednotlivých barevných složek RGB. Takže například data 10, 60, 30 znamenají, že červená je 10, zelená 60 a modrá 30. Na jeden bod je třeba uložit vždy 3 byty. Protože se v paletě ukládá i byte navíc, je pro 256 barev potřeba uložit namísto 786 bytů 1024.

*Tabulka demonstrující příklad uložení palety 16 barevného obrázku:*

	B	G	R	-		B	G	R	-
[0]	84	252	84	0	[8]	0	168	0	0
[1]	252	252	84	0	[9]	168	168	0	0
[2]	84	84	252	0	[A]	0	0	168	0
[3]	252	84	252	0	[B]	168	0	168	0
[4]	84	252	252	0	[C]	0	168	168	0
[5]	252	252	252	0	[D]	168	168	168	0
[6]	0	0	0	0	[E]	84	84	84	0
[7]	168	0	0	0	[F]	252	84	84	0

## Pole bytů definující bitmapu

Je určené podle hlavičky a tabulky barev. Data, které následují hned po tabulce barev, jsou tvořena z pole bytů reprezentující postupnost řádků bitmapy (scan line). Každý scan line se skládá z postupnosti bytů reprezentujících postupnost pixelů ve scan line, v postupnosti zleva doprava. Velikost bitů reprezentujících scan line-novou hloubku barevného formátu a šířku v pixelech bitmapy. Pokud je nutné, tak scan line musí

vynulovat nepoužité prázdné bity po 32-bitový okraj. Ale segment okraje se může vyskytnout leckde v bitmapě. Scan lines v bitmapě jsou uloženy zespoda nahoru. To znamená, že první byte v poli reprezentuje pixel v levém dolním rohu bitmapy a poslední reprezentuje pixel v pravém horním rohu.

## 2. Grafický formát PCX

PCX soubor se skládá z následujících částí:

- Hlavička PCX souboru
- Pole bytů definující bitmapu
- Paleta barev (u 8 bitového obrázku)

```

00000000: 0A 05 01 04 00 00 00 00 00 00 76 00 00 00 28 00
00000010: 00 00 05 00 00 00 03 00 00 00 01 00 04 00 00 00
00000020: 00 00 0C 00 00 00 00 00 00 00 00 00 00 00 00 00
00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000050: 00 00 80 80 80 80 C0 C0 C0 00 00 00 FF 00 00 FF
00000060: 00 00 00 FF FF 00 FF 00 00 00 FF 00 FF 00 FF FF
00000070: 00 00 FF FF FF 00 32 30 50 00 67 89 80 00 8C DE
00000080: FE 00
  
```

■ Bitmap-file header      ■ Bitmap-Information header  
■ Paleta barev            ■ Obrazová data

Obrázek 2. Ukázka souboru 3×5.pcx v hexadecimálním tvaru

Hlavička souboru PCX				
Offset	Délka	Význam	3×5.pcx (hex)	3×5.pcx (dec)
0	1	identifikace PCX souboru	0A	10
1	1	číslo verze	05	5
2	1	kompresce: 0 = žádná, 1 = RLE	01	1
3	1	počet bitů na pixel v obrazové rovině	04	4
4	8	souřadnice obrazu X1, Y1, X2, Y2	00 00 00 00 04 00 02 00	0,0,4,2
12	2	horizontální rozlišení	2C 01	300
14	2	vertikální rozlišení	2C 01	300
16	48	paleta barev (maximálně 16 barev)	-	-
64	1	rezervovaná položka	00	1
65	1	počet obrazových (bitových) rovin	01	1
66	2	počet bytů na obrazový řádek (sudý počet)	04 00	4
68	2	interpretace palety: 1 = barvy, 2 = odstíny šedi	01	1
70-127	58	nepoužito, možné využití aplikací	-	-

### Paleta barev

U 1 a 4 bitových obrázků je paleta uložena v hlavičce souboru. U 8 bitových je uložena na konci souboru a od obrazových dat ji odděluje byte „0c“. Barvy jsou uloženy jako jednotlivé barevné složky RGB.

## Komprimace rastrových dat v souborech typu PCX

Používá se modifikovaný algoritmus RLE, který má u PCX stále stejnou podobu - bez ohledu na typ komprimovaného obrázku. Nemusíme rozlišovat, zda se jedná o 1, 4, 8 či 24 bitový obrázek. Pracujeme s proudem bytů, kde maximální délka proudu odpovídá délce obrazového řádku. Postupně se načítá obrazový řádek a zjišťuje se počet bytů se stejnou hodnotou. Blok za sebou jdoucích bytů se stejnou hodnotou je zapsán jako dvojice bytů - první byte udává počet opakování, druhý byte hodnotu. Počítadlo je inicializováno na hodnotu „c0“ - pokud dekomprimační program narazí na byte větší než „c0“, ví, že se jedná o dvoubytový komprimovaný blok.

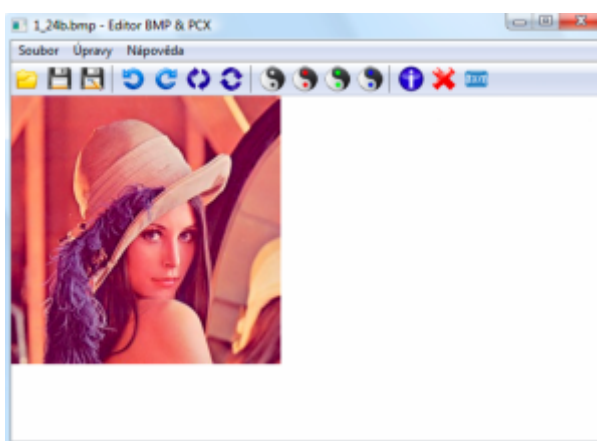
Jednotlivé byty, které nejsou součástí bloku s hodnotou menší než „c0“, jsou do komprimovaného souboru zapsány v původní podobě. Aby nenastala kolize s počítadlem, musí byty, které mají hodnotu větší než „c0“ uložit jako dvojice bytů „c1“ a „xx“ - tj. jako blok o délce jednoho bytu s barvou pixelu uloženou ve druhém bytu. V tomto případě nastává prodloužení výstupního souboru. Pouhým přeindexováním bytů a úpravou palety barev je možné měnit komprimační poměr u PCX souborů (ideální je, aby paleta byla setříděna tak, že nejčastěji používané barvy jsou uloženy na začátku).

Příklad komprimace tří posloupností bytů:

```
01 01 01 01 01 => C5 01
01 01 01 01 01 04 01 01 => C5 01 04 C2 01
01 01 01 01 01 FF 01 01 => C5 01 C1 FF C2 01
```

### 3. Vlastní program pro zpracování rastrových obrázků BMP & PCX

„Editor BMP & PCX“ byl naprogramován jako praktická ukázka principu prohlížení a úpravy BMP a PCX obrázků. Pro vytvoření tohoto programu bylo použito programovacího jazyka „Python“ a knihovny wxWidgets, které umožňují multiplatformní programování.



Obrázek 3. Ukázka vlastního programu

Program obsahuje 2 třídy. Jedna obsahuje grafické rozhraní programu a druhá pak samotnou třídu s funkcemi pro veškerou práci s obrázky.

**Základní proměnné:**

- `nazev` - obsahuje název souboru
- `cesta` - obsahuje celou cestu k souboru
- `data` - pole všech dat načtených ze souboru (v 16 soustavě po bytech)
- `hlavicka1` - u BMP Bitmap-File header, u PCX celá hlavička (v 16 soustavě po bytech)
- `hlavicka2` - u BMP Bitmap-Information header (v 16 soustavě po bytech)
- `obrazovaData` - samotná obrazová data z načteného souboru (v 16 soustavě po bytech)
- `obraz` - čistá data obrazu (pro každý bod obsahuje x, y, r, g, b, odkaz do palety)

### **Proměnné obsahující jednotlivé parametry získané z hlaviček:**

BMP: `identifikace`, `delkaSouboru`, `zacatekObrazu`, `velikostHlavicky`, `sirka`, `vyska`, `bitovychRovin`, `pocetBitu`, `komprese`, `velikost`, `rozliseniH`, `rozliseniV`, `pocetBarev`, `dulezitychBarev`

PCX: `identifikace`, `verze`, `komprese`, `bouroviny`, `souradniceX1`, `souradniceY1`, `souradniceX2`, `souradniceY2`, `rozliseniH`, `rozliseniV`, `paletaBarev`, `pocetRovin`, `bytuNaRadek`, `paletaInfo`, ostatní

### **Funkce pro inverzi všech barev:**

Volá funkci pro vytvoření obrazu s parametrem „`inverze = [255, 255, 255]`“, což při vytváření obrazu zajistí inverzi všech barev. Hodnoty v závorce jsou [R, G, B] nastaveny na 255. Při vytváření obrazu se v absolutní hodnotě od těchto hodnot odečítají skutečné hodnoty RGB - pracuje se buďto přímo s daty obrazu nebo s paletou barev (inverze palety).

### **Funkce pro inverzi červené barvy:**

Volá funkci pro vytvoření obrazu s parametrem „`inverze = [255, 0, 0]`“, což při vytváření obrazu zajistí inverzi červené barvy. Hodnoty v závorce jsou [R, G, B], kde R je nastaveno na 255 a GB na 0. Při vytváření obrazu se v absolutní hodnotě od těchto hodnot odečítají skutečné hodnoty RGB - pracuje se buďto přímo s daty obrazu nebo s paletou barev (inverze palety).

### **Funkce pro inverzi zelené barvy:**

Volá funkci pro vytvoření obrazu s parametrem „`inverze = [0, 255, 0]`“, což při vytváření obrazu zajistí inverzi zelené barvy. Hodnoty v závorce jsou [R, G, B], kde G je nastaveno na 255 a RB na 0. Při vytváření obrazu se v absolutní hodnotě od těchto hodnot odečítají skutečné hodnoty RGB - pracuje se buďto přímo s daty obrazu nebo s paletou barev (inverze palety).

### **Funkce pro inverzi modré barvy:**

Volá funkci pro vytvoření obrazu s parametrem „`inverze = [0, 0, 255]`“, což při vytváření obrazu zajistí inverzi modré barvy. Hodnoty v závorce jsou [R, G, B], kde B je nastaveno na 255 a RG na 0. Při vytváření obrazu se v absolutní hodnotě od těchto hodnot odečítají skutečné hodnoty RGB - pracuje se buďto přímo s daty obrazu nebo s paletou barev (inverze palety).

## **Funkce pro otočení obrazu doleva a doprava o 90°, vertikální a horizontální převrácení:**

Volá funkci pro vytvoření obrazu s patřičnými parametry, což při vytváření obrazu zajistí otočení nebo převrácení. V podstatě se jen mění způsob načítání obrazu.

### **4. Závěr**

Grafický formát BMP patří mezi velmi používané grafické formáty, což je z technologického pohledu docela paradoxní, protože je poměrně složitý na zpracování a přitom nabízí pouze minimum užitečných vlastností. Důvodem je to, že formát byl navržen firmami IBM a Microsoft jako základní rastrový obrazový formát pro jejich operační systémy. U formátu PCX jsou data uložena buď v nekomprimované podobě nebo v komprimovaném tvaru. Nekomprimované PCX se však prakticky nepoužívají.

Samotný program, jako praktická ukázka prohlížení a úpravu BMP a PCX obrázků, byl vytvořen v programovacím jazyce „Python“ za použití knihoven wxWidgets, které umožňují multiplatformní programování. Pro úpravu těchto obrázků poskytuje několik základních funkcí: rotaci o 90° vlevo i vpravo, vertikální i horizontální převrácení, inverzi všech i jednotlivých barev.

### **Zdroje**

1. ROOT.CZ: Grafický formát BMP – používaný a přitom neoblíbený [online]. [cit. 2011-1-04]. Dostupný z WWW:  
<http://www.root.cz/clanky/graficky-format-bmp-pouzivany-a-pritom-neoblíbeny>
2. ROOT.CZ: PCX prakticky – implementace komprimace RLE [online]. [cit. 2011-10-04]. Dostupný z WWW:  
<http://www.root.cz/clanky/pcx-prakticky-implementace-komprimace-rle>