

České vysoké učení technické v Praze FIT

Programování v Pythonu

Jiří Znamenáček

Příprava studijního programu Informatika je podporována projektem financovaným z Evropského sociálního fondu a rozpočtu hlavního města Prahy.

Praha & EU: Investujeme do vaší budoucnosti



Python - NetPBM

Úvod

Rodinu grafických formátů **PBM** (někdy též souhrnně nazývanou **PNM**, *portable anymap format*) vymyslel v 80. letech Jef Poskanzer. Jeho původním záměrem bylo vytvořit monochromatický grafický formát, který by se dal přenášet v těle emailu jako obyčejné ASCII.

Později byl formát rozšířen o možnost zobrazení škály šedé a klasické 8-bitové RGB-spektrum a nakonec i o 16-bitové (které je ale dosti nestandardizované). Je sice velmi nešikovný na uchovávání větších objemů dat, protože jde v podstatě pouze o nekomprimovanou sekvenci údajů o jednotlivých pixelech obrázku, ale na druhou stranu se s ním z úplně stejného důvodu velmi snadno pracuje, a proto je často používán jako pracovní grafický formát.

Dnes se o jeho udržování stará projekt [Netpbm](#).

→ Příklady ve slajdech převzaty a případně mírně upraveny podle [Wikipedie](#).

Rozdělení

Rodina formátů **Netpbm** je extrémně jednoduchá na zpracovávání (a ruku v ruce s tím extrémně nevhodná na uchovávání :). Obsahuje celkem 6 různých formátů lišících se barevnou „hloubkou“ a typem kódování:

identifikátor pro kódování ASCII / binary	typ
P1 / P4	PBM = portable bitmap (černobílý)
P2 / P5	PGM = portable graymap (odstíny šedé)
P3 / P6	PPM = portable pixmap (RGB)

ASCII se bezvadně čte, ale pochopitelně je oproti *binary* pekelně nafouknuté – kde v binárním P6 pro zápis RGB-pixelu stačí 3 bajty, tam ASCII P3 potřebuje bajtů až 11 (a to počítám pouze mezery mezi barvami jednoho pixelu, nikoli mezi jednotlivými pixely).

Struktura PNM-formátů je obecně následující:

```
identifikátor  # identifikátor podle tabulky výše
[# komentář]  # nepovinný; často je zde název souboru
šířka výška  # rozměry zadány jako dekadické číslo v ASCII
maximální-počet-barev  # zadán jako dekadické číslo v ASCII; >0 & <=255 (resp. 65536)
data-obrázku  # „mřížka“ pixelů o daném počtu řádků a sloupců
```

→ představuje libovolný počet bílých znaků (*whitespace*, tj. mezery, TABy, CR, LF), zastupuje právě jeden bílý znak.

Dejte si pozor, kam a jak píšete komentáře – obecně všechno za znakem # je pokládáno za komentář, to znamená i uprostřed řádky (a hlavně odřádkování na konci komentáře), což může někdy udělat nečekanou paseku. Podobně není (u ASCII-kódování) doporučováno mít datové řádky delší než 70 znaků. A nakonec data obrázku by také měla končit na *whitespace*, protože některé programy si s tím jinak neporadí.

Struktura PBM

Struktura PBM je obecně následující (nyní už zjednodušeně oproti obecnému popisu z předchozího slajdu):

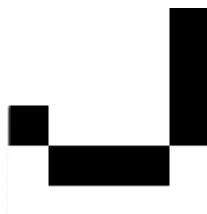
```
identifikátor
šířka výška
data-obrázku
```

I. Pro ukázkou PBM-souboru (tj. černobílého) si ukažme jednoduché písmeno J ve variantě P1, tj. ASCII:

```
P1
# an example bitmap of the letter "J"
7 10
0 0 0 0 0 0 0
0 0 0 0 0 1 0
0 0 0 0 0 1 0
0 0 0 0 0 1 0
0 0 0 0 0 1 0
0 0 0 0 0 1 0
0 0 0 0 0 1 0
0 1 0 0 0 1 0
0 0 1 1 1 0 0
0 0 0 0 0 0 0
```

V tomto souboru je zaznamenán následující obrázek (dvacetkrát zvětšené):





II. Binární podoba, tedy varianta P4, se z pochopitelných důvodů úspornosti liší od binárních variant vícebarevných formátů PGM a PPM: Po hlavičce, která je společná ASCII i binární variantě, následují (za povinným právě jedním prázdným znakem) data obrázku v podobě – *co 8 pixelů, to jeden bajt*. Každých osm sloupečků je tedy sloučeno do jednoho bajtu, ve kterém jejich barvové hodnoty (černá/bílá) představuje nastavení příslušných bitů (0/1). „Přebytečné“ sloupečky jsou „shrnuty“ do posledního bajtu.

Struktura PGM

Struktura PGM je obecně následující (opět zjednodušeně oproti obecnému popisu):

```
identifikátor
šířka výška
maximální-počet-stupňů-šedi
data-obrázku
```

I. Jako příklad PGM (tj. obrázek ve stupních šedi) ve variantě P2, tj. ASCII, si ukažme následující:

```
P2
# Shows the word "FEEP" (example from Netpbm man page on PGM)
24 7
15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 1
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 1
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 1
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

V tomto souboru je zaznamenán následující obrázek (asi třicetkrát zvětšené):





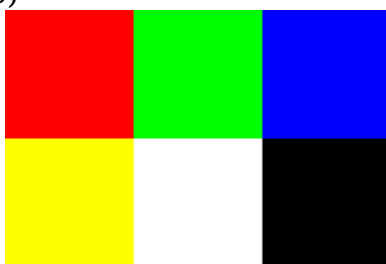
II. Binární podoba, tedy varianta P5, je zcela průhledná: Po hlavičce, která je společná ASCII i binární variantě, následují (za povinným právě jedním prázdným znakem) data obrázku v podobě – *co pixel, to jeden bajt*.

Struktura PPM

Struktura PPM je obecně následující (opět zjednodušeně oproti obecnému popisu):

identifikátor
šířka výška
maximální-počet-barev
data-obrázku

I. PPM se od PGM liší tím, že na každý pixel musí zaznamenat tři barvy, a to postupně v pořadí R-G-B (tj. červená, zelená, modrá). Použijeme-li podobně přehledný způsob zaznamenání jako dříve, následující obrázek (šedesátčtyři-krát zvětšeno)



bude v ASCII-variantě P3 zapsán takto:

```
P3
# The P3 means colors are in ASCII, then 3 columns and 2 rows, the
3 2
255
255 0 0 0 255 0 0 0 255
255 255 0 255 255 255 0 0 0
```

→ Jak jsou pixely „rozházené“ po řádcích není podstatné, protože příslušnou informaci si zpracovávající software načte z druhého významového řádku.

II. Binární podoba, tedy varianta P6, je opět zcela průhledná a prakticky stejná, jako u P5, pouze na každý pixel připadají nyní tři barvové údaje: Po hlavičce, která je společná ASCII i binární variantě, následují (za povinným právě jedním prázdným znakem) data obrázku v podobě – *co pixel, to právě tři bajty (postupně R, G a B)*.

Poznámky

I. ASCII-kódování znamená, že to jsou řetězcové údaje v kódování ASCII! V základním nastavení by to mělo být v pořádku, protože spodní polovina jednobajtových kódování (ve většině případů) i UTF8 bez BOMu jsou v podstatě právě ASCII.

II. Rozložení pixelů do řádek v datech obrázku není podstatné – zpracovávající program ví vše potřebné už z hlavičky. Pouze se doporučuje, aby u ASCII-formátů nebyla žádná z řádek delší než 70 znaků.

→ Pro vlastní ruční tvorbu (nebo čtení) je samozřejmě vizuální rozdělení po pixelech velmi přínosné.

III. Komprese do bajtů mezi ASCII-formátem a binárním formátem se projevuje až na úrovni dat obrázků – celá hlavička je zaznamenána dekadicky v ASCII!

IV. Rozšíření na 16-bitovou barevnou hloubku s sebou nese potřebu rozhodnout o pořadí příslušných dvou bajtů. Nepřekvapivě se na tom jednotlivé implementace neshodnou, ale *de facto* standard *Netpbm* bere jako nejvýznamější bajt ten první.